SIAM J. Discrete Math.
Vol. 16, No. 1, pp. 1–19

ON THE STRUCTURE OF SOME SPACES OF TILINGS[*]

ERIC RÉMILA[†]

Abstract. We study the structure of the set of tilings of a polygon P with bars of fixed length. We obtain an undirected graph connecting two tilings if one can pass from one tile to the other one by a flip (i.e., a local replacement of tiles).

Using algebraic tools (such as tiling groups and their quotients and subgroups), we give a formula to compute the distance in this graph (i.e., the minimal number of necessary flips) between two tilings. Moreover, we prove that, for each pair (T, T') of tilings, the set $\Upsilon_{T,T'}$ consisting of tilings which are in a path of minimal length from T to T' canonically has a structure of distributive lattice.

Key words. tilings, lattice, tiling group, height function

AMS subject classification. 05B45

PII. S0895480100373297

1. Introduction.

In 1990, Conway and Lagarias [2] introduced the notion of tiling groups, which is a very powerful tool for studying tiling problems. Using tiling groups, a lot of necessary conditions for a simply connected figure to be tileable (see [5], [6]) were discovered and unified.

This work has been prolonged by Thurston [11], who especially studied tilings formed with dominoes (i.e., rectangles 2×1) and tilings formed with calissons (i.e., lozenges formed with two equilateral triangles of unit side, with a common edge). For these examples, Thurston [11] introduced the notion of height function associated with a tiling of polygon P. Such a height function permits us to encode each tiling as a mapping from vertices in P to the set $\mathbb{Z}$ of integer numbers.

Using this new notion, a linear algorithm which, given a polygon as input, produces a tiling of this polygon as output (if there exists one tiling; otherwise, the claim of the impossibility is the output) is exhibited.

Thurston's ideas have been taken again by Kenyon and Kenyon [4] and Rémila [7], who obtained some tiling algorithms for the case when tiles are m-bars (rectangles of length m and unit width) and when tiles are equilateral triangles with sides of length 2 or "leaning dominoes" (parallelograms formed with four equilateral triangles of sides of unit length). For these authors, height functions appear as heights on special trees.

In each of the papers cited above, local transformations on tilings, called flips, are introduced. (For example, for tilings with dominoes, a flip is the replacement of two tiles covering a 2×2 square by the other pair of tiles covering the same square.) The space of tilings of P is the (undirected) graph $G_P = (V_P, E_P)$ whose vertices are the tilings of P, and two tilings are linked by an edge if and only if one can pass from a tiling to the other one using only one flip.

This space has been precisely studied for the cases of dominoes and calissons [10], [8]. The main result is that edges can be directed in such a way that the space of tilings becomes the Hasse diagram of a distributive lattice (see, for example, [1] or [3] for definitions about orders and lattices). On the opposite side, before the present

[*] Received by the editors June 5, 2000; accepted for publication (in revised form) April 17, 2002; published electronically November 6, 2002.

http://www.siam.org/journals/sidma/16-1/37329.html

[†] LIP, umr 5668 CNRS INRIA ENS-Lyon, 46 allée d'Italie, 69364 Lyon cedex 07, France and GRIMA, IUT Roanne, Université Jean Monnet Saint-Etienne, 20 avenue de Paris, 42334 Roanne cedex, France (Eric.Remila@ens-lyon.fr).

work, the only result known about spaces of tilings introduced in [4] and [7] was the connectivity of these spaces.

In this paper (which is an extended and improved version of [9]), we first study the space of tilings with m-bars. After recalling general notions about tiling groups and their applications (section 2), we focus on the structure of the tiling group used for m-bars (section 3). We especially show the importance of a special normal subgroup of the main quotient of the tiling group. Using this subgroup, we are able to give an algebraic characterization of functions which encode tilings and give the definition of a distance between tilings. We prove (section 4) that this distance is (up to a multiplicative constant) the distance in the space of tilings, i.e., the minimal number of necessary flips to transform a tiling into another one. This fact gives the flip formula, which permits us to compute this number of flips, and an algorithm to find a shortest path between two tilings given as input. From the algebraic characterization of functions which encode tilings, we also prove (section 5) that, for each pair (T, T') of tilings, the subgraph of the space of tilings induced by tilings which are in a path of minimal length from T to T' canonically has a structure of distributive lattice. Afterwards, we prove that the same method gives similar results for examples in spaces of tilings induced by tilings in the triangular lattice.

2. Tiling groups and tiling functions.

2.1. Tilings. let Λ be the square lattice of the Euclidean plane. A (finite) figure F of Λ is a (finite) union of closed square cells of Λ. A figure F is simply connected if F and its complement $\mathbb{R}^2 \setminus F$ both are connected. A finite simply connected figure F is called a polygon of Λ. The boundary of a polygon P canonically induces a cycle in Λ, which is called the boundary cycle of P. A set S of prototiles is a fixed finite set of polygons of Λ. A tile is a translated copy of a prototile. A tiling T of a figure F is a set of tiles included in F, with pairwise disjoint interiors, such that the union of the tiles of T equals F.

2.2. Groups and their representation. Let Σ be the set $\{a, b, a^{-1}, b^{-1}\}$, let $F_{a,b}$ be the free group generated by a, b, and let π denote the canonical surjection from the language Σ^* of words with letters in Σ to $F_{a,b}$.

Let $R = \{r_1, r_2, \dots, r_p\}$ be a finite set of words of Σ^*. The group N_R denotes the normal group of $F_{a,b}$ generated by the elements of $\pi(R)$ and $\langle a, b | r_1, r_2, \dots, r_p\rangle$ denotes the quotient group $F_{a,b}/N_R$. The group $\langle a, b | r_1, r_2, \dots, r_p\rangle$ has a classical graphic representation: the Cayley graph C_R is the directed graph with labeled edges with labels in $\{a, b\}$ such that

- vertices of C_R are elements of $\langle a, b | r_1, r_2, \dots, r_p\rangle$;
- the set of labels is $\{a, b\}$;
- for each 3-uple (g, g', u) of elements of $(\langle a, b | r_1, r_2, \dots, r_p\rangle)^2 \times \{a, b\}$, $gu = g'$ if and only if there exists an edge of C_R from g to g', labeled by u.

Hence, the underlying graph of Λ can be seen as the Cayley graph $C_{cell} = C_{R_0}$, with $R_0 = \{aba^{-1}b^{-1}\}$, each element of $\{a, b\}$ being associated with a unit move (a for a horizontal rightward move, b for a vertical upward move). In this way, each vertex of Λ is identified to an element of $\langle a, b | aba^{-1}b^{-1}\rangle$.

2.3. Tiling groups. Let $\mu = (v_0, v_1, \dots, v_{p'})$ be a path of Λ, i.e., a sequence of vertices such that, for each integer i, with $0 \leq i < p'$, there exists an element of Σ such that $v_i u_i = v_{i+1}$. The path word $w(\mu)$ is the word $u_0 u_1 \dots u_{p'-1}$. Moreover, if μ is a boundary cycle of a polygon P, we say that $w(\mu)$ is a contour word of P.

Let $S = \{t_1, t_2, \ldots, t_p\}$ be a set of prototiles, and let $R = \{r_1, r_2, \ldots, r_p\}$ be a set of words such that for each integer i such that $1 \leq i \leq p$, r_i is a contour word of t_i. The tiling group of S is the group $G_{tile} = \langle a, b | r_1, r_2, \ldots, r_p \rangle$, and the tiling Cayley graph of S is the graph $C_{tile} = C_R$.

Remark that tiling groups and tiling Cayley graphs depend only on the set S, and not on the contour words chosen for each prototile.

For $S_0 = \{c_0\}$, where c_0 denotes the unit cell of Λ, the tiling group G_{cell} of S_0 is isomorphic to $\mathbb{Z}^2$ and can be identified with Λ.

2.4. Tiling functions. Let T be a tiling of a figure F. The graph G_T of T is the subgraph of $\Lambda (= G_{cell} = \mathbb{Z}^2)$ generated by the set of edges which are on boundaries of tiles of T (i.e., which cut no tile of T).

Definition 2.1. *Let T be a tiling of a figure F. A tiling function induced by T is a mapping f_T from the set V_T of vertices of G_T to G_{tile} such that, for each pair (v, u) of $V_T \times \{a, b\}$, if the edge outgoing from v labeled by u is an edge of G_T, then the equality $f_T(vu) = f_T(v)u$ holds.*

Proposition 2.2 (J. H. Conway). *Let F be a figure of Λ, T be a tiling of F, v_0 be a vertex of G_T, and g_0 be an element of G_{tile}.*

If F is connected (respectively, is a polygon), then there exists at most one (respectively, exactly one) tiling function f_T induced by T such that $f_T(v_0) = g_0$.

Proof (sketch). A function f_T can easily be constructed successively exploring the contour of each tile: f_T is first defined on the vertices of a tile t_0 which has v_0 on its boundary. Afterwards, f_T is defined on the vertices of a tile t_1 which has a common vertex with t_0 and so on.

This method gives the uniqueness of f_T for F connected. Nevertheless, a conflict (i.e., a vertex v such that two distinct values of $f_T(v)$) can arise if F has some holes, which yields that there is no tiling function. □

Remark 2.3. *let P be a polygon and v_0 be a vertex of the boundary of P. If f and f' are tiling functions such that $f(v_0) = f'(v_0)$, then, for each vertex v of the boundary of P, $f(v) = f'(v)$.*

The use of tiling functions is one of the main methods for studying tilings. Interesting examples are developed in [2], [4], [5], [6], [7], [11].

We apply the theoretical notions on our special case below.

3. Groups for tilings with bars . Let m and n be fixed positive integers such that $m \geq 2$ and $n \geq 2$. The first sets of prototiles on which we apply notions of the previous section is (as in [4]) the set $S_{m,n} = \{h_m, v_n\}$, where

- the prototile h_m denotes an $m \times 1$ horizontal rectangle, which admits $a^m b a^{-m} b^{-1}$ for a contour word, and
- the prototile v_n denotes a $1 \times n$ vertical rectangle which admits $b^n a b^{-n} a^{-1}$ for a contour word.

Thus, a set R_{bars} of contour words of prototiles is $\{a^m b a^{-m} b^{-1}, b^n a b^{-n} a^{-1}\}$, which gives a group $G_{bars} = \langle a, b | a^m b a^{-m} b^{-1}, b^n a b^{-n} a^{-1} \rangle$. Since this group has a complex structure, quotient groups of G_{bars} will be used in order to have groups that can be easily described. This is an indirect way to understand the structure of G_{bars}.

3.1. Quotient groups. To obtain such quotient groups, it suffices to exhibit a set $R' = \{r'_1, r'_2, \ldots, r'_{p'}\}$ of words such that the words of R_{bars} are null in $\langle a, b | r'_1, r'_2, \ldots, r'_{p'} \rangle$. In this case, we have a natural surjection from G_{bars} to $\langle a, b | r'_1, r'_2, \ldots, r'_{p'} \rangle$.

3.1.1. The cycle group. As in [4], the principal quotient group used is constructed from the auxiliary set $R' = \{a^m, b^n\}$. Obviously, the elements of R_{bars} are null in $\langle a, b | a^m, b^n \rangle$, which guarantees that we have a canonical surjection s from G_{bars} to $\langle a, b | a^m, b^n \rangle$.

The structure of $\langle a, b | a^m, b^n \rangle$ is rather simple: it is isomorphic to the free product of a cyclic group of m elements and a cyclic group of n elements. The associated Cayley graph $C_{R'}$ is formed with directed cycles of length m with edges labeled by a and directed cycles of length n with edges labeled by b, each vertex being element of exactly two cycles, one of each type (see Figure 1). Moreover, $C_{R'}$ is a tree of cycles: the only cycles of $C_{R'}$ are those described above. Thus we pose $C_{R'} = C_{cycles}$ and $\langle a, b | a^m, b^n \rangle = G_{cycles}$.

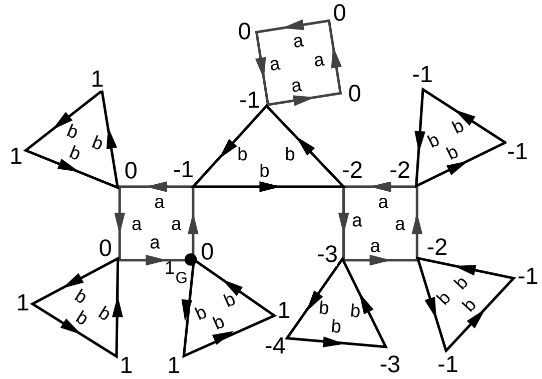


FIG. 1. *The Cayley graph of the group G_{cycles} (example with $m = 4$ and $n = 3$).*

We introduce some definitions which permit us to have a geometrical understanding of G_{cycles}.

DEFINITION 3.1 (canonical expression, length, distance, order in the cycle group). *Each element w of G_{cycles} has a canonical expression $w = x_1^{i_1} x_2^{i_2} \dots x_p^{i_p}$, where,*

- *for each integer j, x_j is an element of $\{a, b\}$ and (for $j < p$) $x_j \neq x_{j+1}$;*
- *if $x_j = a$ (respectively, $x_j = b$), then $1 \leq i_j < m$ (respectively, $1 \leq i_j < n$).*

With these notations, we say that the integer p is the length of w (denoted by $l(w)$). For $p \geq 2$, the element $init(w) = x_1^{i_1} x_2^{i_2}$ of G_{cycles} is called the initial part of w, and the element $fin(w) = x_{p-1}^{i_{p-1}} x_p^{i_p}$ is called the final part of w.

The distance $d(w', w'')$ between two elements of G_{cycles} is equal to $l(w'^{-1} w'')$. Moreover, we say that $w' \leq_{cycles} w''$ if $l(w'^{-1} w'') = l(w'') - l(w')$.

The relation $\leq_{cycles}$ is obviously an order relation. Each element w of G (such that $w \neq 1_{G_{cycles}}$) has a unique immediate predecessor. In other words, the relation $\leq_{cycles}$ induces a structure of tree on G_{cycles}. Thus, the order $\leq_{cycles}$ has the following infimum property: for each pair (w', w'') of elements of G_{cycles}, there exists an element $inf_{cycles}(w', w'')$ in G_{cycles} such that $inf_{cycles}(w', w'') \leq_{cycles} w'$, $inf_{cycles}(w', w'') \leq_{cycles} w'$, and for each element w''', if $w''' \leq_{cycles} w'$ and $w''' \leq_{cycles} w''$, then $w''' \leq_{cycles} inf_{cycles}(w', w'')$.

3.1.2. The cell group. Another quotient group of G_{bars} is the group G_{cell} defined from the set $R_0 = \{aba^{-1}b^{-1}\}$. As we have seen before, this group is isomorphic to $\mathbb{Z}^2$.

3.1.3. The torus group. The third quotient group which will be used is $G_{torus} = \langle a, b | a^m, b^n, aba^{-1}b^{-1} \rangle$, constructed using the set $R'' = R' \cup R_0 = \{a^m, b^n, aba^{-1}b^{-1}\}$. This group is isomorphic to $\mathbb{Z}_m \times \mathbb{Z}_n$, i.e., the direct product of a cyclic group of m elements and a cyclic group of n elements; each element can be seen as a pair (i, j) of $\mathbb{Z}_m \times \mathbb{Z}_n$. The associated Cayley graph $C_{R''}$ is formed with directed cycles of length m with edges labeled by a and directed cycles of length n with edges labeled by b in such a way that a and b commute. It has the structure of a torus $T_{m \times n}$. Notice that G_{torus} is also a quotient group of both G_{cycles} and G_{cell}.

3.2. Bar tiling projections. Let T be a tiling of a polygon P, and let f_T be a tiling function induced by T. Function g_T, defined by $g_T = \pi' \circ f_T$ (where π' denotes the canonical surjection from G_{bars} to G_{cycles} and $\circ$ denotes the composition of functions), is called a tiling projection of T (see Figure 2). Notice that, for a set of bars as the set of prototiles, the set of vertices of G_T is the set of vertices which are elements of P (since each vertex of P is on the boundary of a bar of T).

$$\begin{tikzcd}
 & & F_{a,b} \arrow[d] & \\
\text{Polygon} \arrow[rr, "f_T"] \arrow[dr, "g_T"'] & & G_{bars} \arrow[dl] \arrow[dr] & \\
 & G_{cycles} \arrow[dr] & & G_{cell} \arrow[dl] \\
 & & G_{torus} &
\end{tikzcd}$$

FIG. 2. *Mappings used (dark arrows represent canonical surjective group morphisms).*

Fix a vertex v_0 of the boundary of P and assume that $g_T(v_0) = 1_{G_{cycles}}$. Let v be any vertex of P; how can we compute $g_T(v)$? We have to find a path of P from v_0 to v which cuts no tile of T, and, from the definitions of f_T and g_T, the word associated with this path, seen as an element of G_{cycles}, is equal to $g_T(v)$ (see Figure 3).

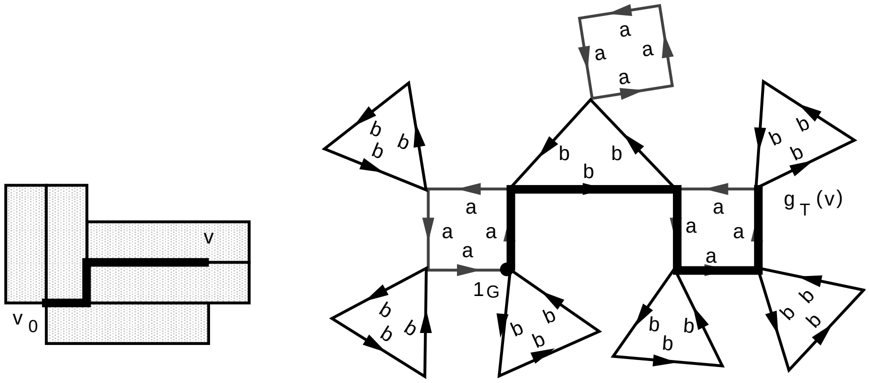


FIG. 3. *Computation of $g_T(v)$.*

Let (v', v'') be a pair of neighbor vertices of P. If the line segment $[v', v'']$ is on the boundary of a bar of T, then $d(g_T(v'), g_T(v'')) = 1$. If, otherwise, this line segment cuts a bar of T, then $d(g_T(v'), g_T(v'')) = 3$. Thus function g_T encodes the tiling T.

Also, notice that, for each vertex v of the boundary of P, the value $g_T(v)$ does not depend on the tiling T, from Remark 2.3.

Definition 3.2. *For each element v of the planar grid, there exists a unique pair (i_v, j_v) of $\mathbb{Z}^2$ such that $v = v_0 a^{i_v} b^{j_v}$. We define $cong_{torus}(v)$ as the element of G_{torus} such that*

$$cong_{torus}(v) = a^{i_v} b^{j_v}.$$

Remark 3.3. *Let s denote the canonical surjective morphism from G_{cycles} to G_{torus}. One obviously sees (by induction on the length of a shortest path, from v_0 to v, which cuts no tile) that, for each tiling T and each vertex v of P, we have $s(g_T(v)) = cong_{torus}(v)$.*

Thus, for each pair (T, T') of tilings of P and each vertex v of P, we have

$$g_T(v)^{-1} g_{T'}(v) \in ker(s).$$

The proposition below gives a characterization of tiling projections, from a congruence condition and a local Lipschitz condition. It gives an algebraic interpretation of the geometrical regularity of a tiling.

Proposition 3.4. *Let g be a function from the set of vertices of P to G_{cycles} such that*

- $g(v_0) = 1_{G_{cycles}}$;
- *for each vertex v of P, $s(g(v)) = cong_{torus}(v)$;*
- *for each pair (v, v') of neighbor vertices of P, $d(g(v), g(v')) \leq 3$, and if, moreover, the line segment $[v, v']$ is included on the boundary of P, then $d(g(v), g(v')) = 1$.*

There exists a tiling T of P such that $g = g_T$.

Proof. Let v and v' be vertices of the polygon P such that $v' = va$ (respectively, $v' = vb$). Let u be the element of G_{cycles} such that $g(v') = g(v)u$. We necessarily have $s(u) = a$ (respectively, $s(u) = b$). Thus, since $l(u) \leq 3$, there exists an integer j, with $0 \leq j < n$, such that $u = b^j a b^{-j}$ (respectively, there exists an integer i, with $0 \leq i < m$, such that $u = a^i b a^{-i}$).

We claim the following fact. (We can also claim the symmetric fact.)

Fact. Assume that $v' = va$ and there exists an integer j, with $0 < j < n$, such that $g(v') = g(v) b^j a b^{-j}$. Then, for each integer j' such that $j - n \leq j' \leq j$, we have $g(vb^{j'}) = g(v) b^{j'}$ and $g(v' b^{j'}) = g(v') b^{j'}$.

We prove this fact as follows: first notice that the cells an edge of which is the line segment $[v, v']$ are included in P, since $d(g(v), g(v')) \neq 1$. Thus, the vertices vb and $v'b$ are both in P.

Now let u' and u'' such that $g(vb) = g(v)u'$ and $g(v'b) = g(v')u''$. There exists a pair (i, i') of $\{0, 1, \ldots, m-1\}^2$ such that $u' = a^{i'} b a^{-i'}$ and $u'' = a^{i''} b a^{-i''}$. With these notations, using the path $(vb, v, v', v'b)$, we have $g(v'b) = g(vb)(u')^{-1} b^j a b^{-j} u''$. From our hypothesis, we have $l((u')^{-1} b^j a b^{-j} u'') \leq 3$, which necessarily yields $i' = i'' = 0$, since $j \neq 0$.

We have obtained the claim for $j' = 1$, and, moreover, $g(v'b) = g(vb) b^{j-1} a b^{-j+1}$. Thus, if $j - 1 \neq 0$, we can repeat the same argument for $g(v'b^2)$ and $g(vb^2)$, and so on for $g(vb^{j'})$ and $g(vb^{j'})$, while $j' \leq j$. We can also use the same argument in the other direction (for $j' < 0$) while $j' \geq j - n$, which concludes the proof of the fact.

We now introduce the set T of tiles defined as follows: a vertical (respectively, horizontal) bar is in T if and only if there exists a vertex v of P and an integer j with $0 < j < n$ (respectively, an integer i with $0 < i < m$) such that $g(va) = g(v)b^j ab^{-j}$ (respectively, $g(vb) = g(v)a^i ba^{-i}$).

There is no overlap, from the fact above. Let $(v_0, v_1, v_2, v_3, v_4 = v_0)$ be a contour cycle of a cell, in the trigonometric sense, such that v_0 is the southwest corner of the cell. If we have $d(g(v_0), g(v_1)) = d(g(v_1), g(v_2)) = d(g(v_2), g(v_3)) = d(g(v_3), g(v_4)) = 1$, then we have $g(v_0) = g(v_0)aba^{-1}b^{-1}$, which is a contradiction. Thus there exists an integer i of $\{0, 1, 2, 3\}$ such that $d(g(v_i), g(v_{i+1})) = 3$, which guarantees that the cell is covered by a bar of T. Thus there is no gap and T is actually a tiling of P.

The fact that $g_T = g$ is obvious, from the definition of T and the fact above, which concludes the proof. □

3.3. The kernel of the morphism from G_{cycles} to G_{torus}.

Let s denote the canonical surjective morphism from G_{cycles} to G_{torus}. We will see that the subgroup $ker(s)$ of G_{cycles} has a fundamental importance in the comparison of tilings. This is a consequence of Remark 3.3. We thus have to explore the structure of this group.

Let S_{rect} be the subset of G_{cycles} defined by $S_{rect} = \{a^ib^ja^{-i}b^{-j}, b^ja^ib^{-i}a^{-j}$, for $1 \leq i < m$ and $1 \leq j < n\}$. One easily verifies that S_{rect} is closed by inverse (i.e., if $t \in S_{rect}$, then $t^{-1} \in S_{rect}$) and $S_{rect} \subset ker(s)$.

Remark 3.5. *Let w be an element of $ker(s)$. By projection (since G_{torus} is commutative), one obviously verifies that $l(w) \leq 4$ if and only if $w \in S_{rect} \cup \{1_{G_{cycles}}\}$.*

We need the following result to study the space of tilings of P with bars.

Proposition 3.6. *Let w be an element of $ker(s)$. There exists a unique finite sequence $(t_1, t_2, \dots, t_p)$ of elements of S_{rect} (called the decomposition of w) such that $w = \Pi_{i=1}^p t_i$ and, for each integer i, such that $0 < i < p$, $t_it_{i+1} \neq 1_{G_{cycles}}$.*

Moreover, the decomposition of w can be computed in $O(l(w))$ time units, from the canonical expression of w given as input.

We decompose the proof of the above proposition into two lemmas.

Lemma 3.7 (existence and computation of a decomposition). *Let w be an element of $ker(s)$ (different from $1_{G_{cycles}}$) such that the initial part of w is a^ib^j (respectively, b^ja^i). We state $w = a^ib^ja^{-i}b^{-j}w'$ (respectively, $w = b^ja^ib^{-j}a^{-i}w'$).*

We have $l(w') \leq l(w) - 1$.

Proof. Assume that the initial part of w is a^ib^j. From Remark 3.5, we have $l(w) \geq 4$. Thus, we can state $w = a^ib^ja^ku$, with $a^ib^ja^k \leq_{cycles} w$. Thus $w' = b^ja^{k-i}u$, which gives $l(w') \leq l(u) + 2 = l(w) - 3 + 2 = l(w) - 1$.

The symmetric case can be treated in a symmetric way. □

Lemma 3.8 (uniqueness of the decomposition). *Let $(t_1, t_2, \dots, t_p)$ be a (nonempty) sequence of elements of S_{rect} such that, for each integer i such that $0 < i < p$, $t_it_{i+1} \neq 1$. Let w be defined by $w = \Pi_{i=1}^p t_i$. Then*

- *$l(w) \geq p + 3$,*
- *$t_p = a^{-i}b^{-j}a^ib^j$ (respectively, $t_p = b^{-j}a^{-i}b^ja^i$) if and only if the final part of w is a^ib^j (respectively, b^ja^i).*

Proof. The proof is by induction on the integer p. The result is obvious if $p = 1$. Assume that the result is true for each element w such that w is a product of p elements of S_{rect}. Let w' be a product of $p + 1$ elements of S_{rect}. We state $w' = t_1t_2 \dots t_pt_{p+1}$. By induction hypothesis, if $t_p = a^{-i}b^{-j}a^ib^j$, the canonical expression of element $w = t_1t_2 \dots t_p$ is of type ua^ib^j, with u such that $l(u) \geq p + 1$, and the canonical expression of u finishes by b.

If $t_{p+1} = a^{-k}b^{-l}a^kb^l$, then the canonical expression of w' is $ua^ib^{j'}a^{m-k}b^{n-l}a^kb^l$, which gives the first item and the direct part of the second item of the lemma.

If $t_{p+1} = b^{-l}a^{-k}b^la^k$, then we have $w' = ua^ib^{j-l}a^{-k}b^la^k$. If $j \neq l$, then the results of the lemma hold. If $j - l = 0$, then we have $w' = ua^{i-k}b^la^k$. Notice that if $i = k$, then $t_pt_{p+1} = 1$, which is a contradiction. Thus $a^{i-k} \neq 1$, which gives the first item and the direct part of the second item of the lemma.

Conversely, if the final part of w' is a^lb^k, then we necessarily have $t_{p+1} = b^{-l}a^{-k}b^la^k$, since, otherwise, the direct part of the second item of the lemma would be contradicted.

The symmetric cases can be treated in a similar way. □

The proposition proved above allows the definitions below.

Definition 3.9 (decomposition number of an element of $ker(s)$). *The decomposition number of an element w of $ker(s)$ (denoted by $num(w)$) is the number of factors of its decomposition.*

Definition 3.10 (order on $ker(s)$). *Let w and w' be elements of $ker(s)$ whose decompositions are $w = \Pi_{i=1}^p t_i$ and $w' = \Pi_{i=1}^{p'} t'_i$. We say that $w \leq_{decomp} w'$ if $p \leq p'$, and, for each integer i such that $1 \leq i \leq p$, we have $t_i = t'_i$*

The relation $\leq_{decomp}$ is an order relation on the set $ker(s)$. Each element w of $ker(s)$ (such that $w \neq 1_{G_{cycles}}$) has a unique immediate predecessor (denoted by $pr_{decomp}(w)$), which induces a structure of a directed tree on $ker(s)$.

The order $\leq_{decomp}$ has the infimum property. The infimum of a pair (w, w') of elements of $ker(s)$ is denoted by $inf_{decomp}(w, w')$. Notice that, from the tree structure, $inf_{decomp}(w, w')$ is the unique element w'' of $ker(s)$ such that

$$inf_{decomp}(w''^{-1}w, w''^{-1}w') = inf_{decomp}(w''^{-1}, w''^{-1}w) = inf_{decomp}(w''^{-1}, w''^{-1}w') = 1_{G_{cycles}}.$$

From Lemma 3.7, we have the following characterization of the predecessor.

Remark 3.11. *For each element w of $ker(s)$ (such that $w \neq 1_{G_{cycles}}$), there exists a unique element t_i of S_{rect} such that $l(wt_i) \leq l(w) - 1$. For any other element t_j of S_{rect}, we have $l(wt_j) \geq l(w) + 1$.*

We have the equality $pr_{decomp}(w) = wt_i$.

4. Distance between tilings. We introduce a distance in the space of tilings by the definition below.

Definition 4.1. *Let (T, T') be a pair of tilings of P, and let g_T and $g_{T'}$ be the associated projections (such that $g_T(v_0) = g_{T'}(v_0) = 1_{G_{cycles}}$). The distance $\Delta(T, T')$ is defined by the equality*

$$\Delta(T, T') = \sum_{v \in P} num(g_{T'}(v)^{-1}g_T(v)).$$

We will prove that the distance defined above has a geometric interpretation, using the local flips defined below.

4.1. Local flips. Let T be a tiling of P. Assume that there exists an $m \times n$ rectangle R_0 such that T contains a tiling T_0 of R_0. (In this case, tiles of T_0 are copies of the same prototile.) Another tiling T_{flip} of P is obtained by replacing tiles of T_0 by tiles of T_1, where T_1 denotes the only tiling of R_0 different from T_0. We say that T_{flip} is deduced from T by a local flip whose support is R_0 (see Figure 4).

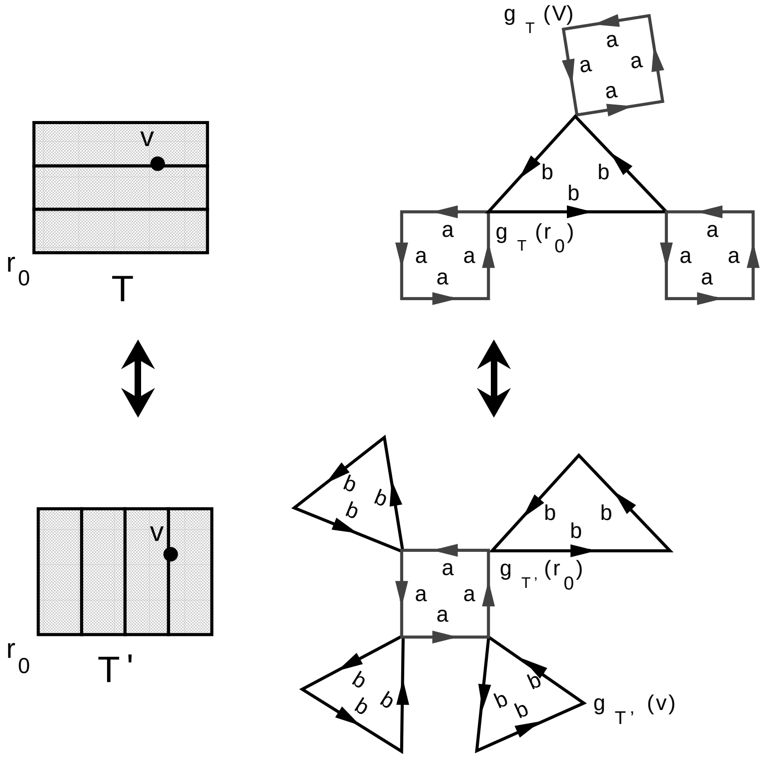


Fig. 4. *Local flips for bars.*

If v is a vertex of P which is not in the interior of R_0, then there exists a path from v_0 to v which cuts no tile of T and does not go through the interior part of R_0. This path also cuts no tile of T_{flip}. Thus, $g_T(v) = g_{T_{flip}}(v)$.

If v is an interior vertex of R_0, let r_0 be the lower left corner of R_0. Let us denote $v = r_0a^ib^j (= r_0b^ja^i)$. Assume that T_0 consists of h_m tiles. Then we have $g_T(v) = g_T(r_0)b^ja^i$ and $g_{T_{flip}}(v) = g_T(r_0)a^ib^j$. Thus

$$g_{T_{flip}}(v) = g_T(v)a^{-i}b^{-j}a^ib^j.$$

This equality means that a local flip induces a multiplication of the tiling projection of each interior vertex of R_0 by an appropriate element of S_{rect}. Thus

$$\Delta(T, T_{flip}) = (m-1)(n-1).$$

4.2. The flip formula. It has been proved, as a consequence of the algorithm of tiling [4], that, for each pair (T, T') of tilings of P, there exists a sequence of local flips which permit us to deduce T' from T. We will improve this result, giving a formula for the minimal number $minflip(T, T')$ of necessary flips.

Proposition 4.2. *For each pair (T, T') of tilings of P, we have*

$$\Delta(T, T') \leq (m-1)(n-1)minflip(T, T').$$

Proof. Let $(T = T_0, T_1, \dots, T_p = T')$ be a sequence of tilings such that, for $0 \leq i < p$, T_{i+1} is deduced from T_i by a local flip. We have $\Delta(T, T') \leq \Sigma_{i=0}^{p-1} \Delta(T_i, T_{i+1})$.

From the study of local flips, we see that, for $0 \leq i < p$, $\Delta(T_i, T_{i+1}) = (m-1)(n-1)$: If v is not in the interior part of the rectangle on which the flip is done, then $g_{T_i}(v) = g_{T_{i+1}}(v)$, thus $num(g_{T_i}(v)^{-1} g_{T_{i+1}}(v)) = 0$; if v is in the interior part of the rectangle on which the flip is done (we have $(m-1)(n-1)$ such interior vertices), then $g_{T_{i+1}}(v)$ is obtained by multiplying $g_{T_i}(v)$ by an element of S_{rect}, thus $num(g_{T_i}(v)^{-1} g_{T_{i+1}}(v)) = 1$.

Thus $\Delta(T_i, T_{i+1}) = (m-1)(n-1)$, which yields that $\Delta(T, T') \leq (m-1)(n-1)p$, which gives the result. □

We will prove that the inequality of the previous proposition is actually an equality. To do it, we have to exhibit a local flip in T which decreases $\Delta(T, T')$ of $(m-1)(n-1)$ units. This flip will be done in the neighborhood of a special point which will be called a maximal vertex.

4.2.1. Maximal vertex.

Definition 4.3. *A maximal vertex for a pair (T, T') of distinct tilings is a vertex v of P such that*

- $g_T(v) \neq g_{T'}(v)$,
- $max[l(g_T(v)), l(g_{T'}(v))]$ *is maximal with the previous condition.*

There exists such a maximal vertex (since otherwise $g_T = g_{T'}$, which yields that $T = T'$). Let v_1 be a maximal vertex. It can be assumed without loss of generality that $l(g_{T'}(v_1)) \leq l(g_T(v_1))$.

Lemma 4.4. *We have $l(g_T(v_1)) \geq 2$ and, moreover, $fin(g_T(v_1)) = fin(g_{T'}(v_1)^{-1} g_T(v_1))$.*

Proof. We state $g_T(v_1) = uw$ and $g_{T'}(v_1) = uw'$, where u denotes $inf_{cycles}(g_T(v_1), g_{T'}(v_1))$. Thus, $g_{T'}(v_1)^{-1} g_T(v_1) = w'^{-1} w$. Since it is assumed that $l(g_{T'}(v_1)) \leq l(g_T(v_1))$, we have $l(w') \leq l(w)$. Moreover, $4 \leq l(w'^{-1} w) \leq l(w') + l(w)$, which yields that $2 \leq l(w)$.

If $w = a^i b^j$, then we necessarily have $w' = b^j a^i$, since $l(w') \leq l(w)$, $w' \neq w$, and $w'^{-1} w = g_{T'}(v_1)^{-1} g_T(v_1)$ is an element of $ker(s)$. This fact gives the result. (The same argument can be used in the symmetric case, when $w = b^j a^i$.)

If $l(w) \geq 3$, then the result is obvious, since $inf_{cycles}(w, w') = 1_{G_{cycles}}$. □

Proposition 4.5. *Assume that the final part of $g_T(v_1)$ is $a^i b^j$ (respectively, $b^j a^i$). Let r_0 be the vertex of the plane defined by $v_1 = r_0 a^i b^j$ (respectively, $v_1 = r_0 b^j a^i$). This vertex is the lower left corner of an $m \times n$-rectangle R_0 such that T contains a tiling T_{R_0} of R_0 consisting of vertical (respectively, horizontal) tiles.*

Proof. We treat the case when the final part of $g_T(v_1)$ is $a^i b^j$. Let u_1 be the element of G_{cycles} such that $u_1 a^i b^j = g_T(v_1)$. By the definition of g_T, there exists a unique integer l such that $0 \leq l < n$ and $g_T(v_2) = g_T(v_1) b^{-l} a b^l = u_1 a^i b^j b^{-l} a b^l$. If we have $l \neq j$, then $l(g_T(v_2)) = l(g_T(v_1)) + 2$, thus $l(g_T(v_2)) > l(g_T(v_1))$.

Moreover, from the previous lemma, the canonical expression of $g_{T'}(v_1)^{-1} g_T(v_1)$ is $w_1 a^i b^j$, with w_1 finishing by b, thus $g_{T'}(v_1)^{-1} g_T(v_2) = w_1 a^i b^j b^{-l} a b^l$. Thus, if $l \neq j$, then we have $d(g_T(v_2), g_{T'}(v_1)) = d(g_T(v_1), g_{T'}(v_1)) + 2$, which gives $d(g_T(v_2), g_{T'}(v_1)) \geq 6$.

On the other hand, we have $d(g_{T'}(v_2), g_{T'}(v_1)) \leq 3$, which yields that $g_T(v_2) \neq g_{T'}(v_2)$. The previous facts contradict the maximality of v_1. Thus we necessarily have $j = l$ and, consequently, $g_T(v_2) = g_T(v_1) b^{-j} a b^j$. This last equality implies (using the same kind of argument used in the proof of the fact of Proposition 3.4) that the vertical tile whose lower left corner is $r_0 a^i$ is an element of T.

If, moreover, v_2 is an interior vertex of R_0, then $i+1 \neq m$, thus $l(g_T(v_2)) = l(g_T(v_1))$ and $d(g_T(v_2), g_{T'}(v_1)) = d(g_T(v_1), g_{T'}(v_1)) \geq 4$, which gives $g_T(v_2) \neq g_{T'}(v_2)$. Hence, v_2 is also a maximal vertex, and we can repeat the argument for v_3, the right neighbor of v_2, and so on. The same kind of argument can also be used leftward. This gives the tiling T_{R_0} of R_0. □

Proposition 4.6. *Let T_{flip} be the tiling deduced from T by a flip on R_0. We have the equality $\Delta(T_{flip}, T') = \Delta(T, T') - (m-1)(n-1)$.*

Proof. Let us denote, for any vertex v of the interior of R_0, $v = r_0 a^{i'} b^{j'} (= r_0 b^{j'} a^{i'})$, and $g_{T'}(v)^{-1} g_T(v) = \Pi_{i=1}^{p} t_i$. The final part of $g_T(v)$ is $a^{i'} b^{j'}$, which yields that the last factor of $g_{T'}(v)^{-1} g_T(v)$ is $t_p = a^{-i'} b^{-j'} a^{i'} b^{j'}$. On the other hand,

$$g_T(v)^{-1} g_{T_{flip}}(v) = (g_T(v)^{-1} g_T(v_0))(g_T(v_0)^{-1} g_{T_{flip}}(v)) = b^{-j'} a^{-i'} b^{j'} a^{i'} = (t_p)^{-1}.$$

Thus,

$$g_{T'}(v)^{-1} g_{T_{flip}}(v) = (g_{T'}(v)^{-1} g_T(v))(g_T(v)^{-1} g_{T_{flip}}(v)) = (\Pi_{i=1}^{p} t_i)(t_p)^{-1} = \Pi_{i=1}^{p-1} t_i,$$

which yields that $num(g_{T'}(v)^{-1} g_{T_{flip}}(v)) = num(g_{T'}(v)^{-1} g_T(v)) - 1$, which gives the result. □

Corollary 4.7 (flip formula). *For each pair (T, T') of tilings of P, we have*

$$\Delta(T, T') = (m-1)(n-1) minflip(T, T').$$

Proof. The proof is obvious, by induction on $\Delta(T, T')$. □

4.3. Algorithm.

4.3.1. Presentation. The notion of maximal vertex permits us to give an algorithm which, given a pair (T, T') of tilings of a polygon, gives a sequence $(R_1, R_2, \ldots, R_{minflip(T,T')})$ of rectangles on which flips can successively be done, to go from T to T'. Informally, such a sequence is a space economic way to encode a shortest path of tilings from T to T'. The algorithm is presented below.

Input: a pair (T, T') of tilings of a same polygon P.

Initialization: Construct a spanning tree rooted in a fixed vertex v_0. When a new vertex v is reached, compute the canonical expressions of $g_T(v)$, $g_{T'}(v)$, $g_{T'}(v)^{-1} g_T(v)$, compute $max(l(g_T)(v)), l(g_{T'}(v)))$, and place v in a "list of lists" such that each vertex v' is in a basic list corresponding to the value $max(l(g_T(v')), l(g_{T'}(v')))$, and those basic lists are ordered according to their decreasing corresponding values.

A variable list L of rectangles stores the sequence of rectangles used. The beginning of this list consists of rectangles which are supports of flips deduced from T, and the end of the list contains the rectangles which are supports of flips deduced from T'. For initialization, L is empty and each insertion is done just between both parts.

We also need a variable vertex v_1, which, for initialization, is the first element of the "list of lists" of vertices.

Repeat: Take the first element of the "list of lists" for v_1.

If $g_T(v_1) = g_{T'}(v_1)$, then delete the value of v_1 from the "list of lists."

Otherwise, v_1 is a maximal vertex, which (in the case when $l(g_T(v_1)) \geq l(g_{T'}(v_1))$, the symmetric case being treated in a symmetric way) is in a rectangle R on which a flip of tiling T can be done. (R is defined by the final part of $g_T(v_1)$.)

Insert this rectangle in L and update replacing T by T_{flip}: for each vertex v of the interior of R, update $g_T(v)$, $max(l(g_T(v)), l(g_{T'}(v)))$ and the place of v in the "list of lists" of vertices.

Notice that $l(g_T(v)) - 4 \leq l(g_{T_{flip}}(v)) \leq l(g_T(v))$, which permits us to update the place of v in a constant time.

Until: the "list of lists" is empty.

4.3.2. Analysis. Correctness. Just before the $i+1$st passage through the loop, a pair $(T_{1,i}, T_{2,i})$ of tilings is stored. At the initialization, $(T_{1,0}, T_{2,0}) = (T, T')$; at each passage through the loop $(T_{1,i}, T_{2,i})$ is replaced by a pair $(T_{1,i+1}, T_{2,i+1})$ such that

- $minflip(T_{1,i+1}, T_{2,i+1}) = minflip(T_{1,i}, T_{2,i}) - 1$;
- either $T_{1,i+1} = T_{1,i}$ and $T_{2,i+1}$ is deduced from $T_{2,i}$ by a flip, or $T_{2,i+1} = T_{2,i}$ and $T_{1,i+1}$ is deduced from $T_{1,i}$ by a flip.

The algorithm stops for the integer i_0 such that $T_{1,i_0} = T_{2,i_0}$.

We define a finite subsequence of $(T_{1,0}, T_{1,1}, \ldots, T_{1,i_0})$ constructed extracting different tilings: precisely $(T'_{1,0}, T'_{1,1}, \ldots, T'_{1,p})$ is the sequence of tilings such that $T'_{1,0} = T_{1,0}$, $T'_{1,p} = T_{1,i_0}$, and, for each integer i such that $i < p$, $T'_{1,i+1}$ equals the first element $T_{1,j}$ of the sequence $(T_{1,0}, T_{1,1}, \ldots, T_{1,i_0})$ such that $T_{1,j} \neq T'_{1,i}$. We similarly define a subsequence $(T'_{2,0}, T'_{2,1}, \ldots, T'_{2,p'})$ of $(T_{2,0}, T_{2,1}, \ldots, T_{2,i_0})$.

By this way, the sequence $(T'_{1,0}, T'_{1,1}, \ldots, T'_{1,p} = T'_{2,p'}, T'_{2,p'-1}, \ldots, T'_{2,0})$ is a shortest path of tilings from T to T'. The sequence of supports of flips necessary to pass from a tiling of this sequence to its successor is exactly the final list L, obtained at the end of the execution of the algorithm. This proves the correctness.

Time complexity. The characteristic values of a vertex v can be deduced from those of its father in the spanning tree in $O(1)$ time units. Thus, the initialization can be done in $O(A(P))$ time units, where $A(P)$ denotes the area (i.e., the number of cells) of P.

Each passage through the loop costs $O(1)$ time units (for m and n been fixed) and reduces the distance between the current tiling and T' from $(m-1)(n-1)$. Thus the second part of the algorithm costs at most $O(\Delta(T, T'))$ time units.

Thus the complete time cost is $O(A(P) + \Delta(T, T'))$, which is optimal since $O(A(P))$ time units are necessary to read the input and $O(\Delta(T, T'))$ time units are necessary to write the output.

5. Structures of lattices and semilattices.

5.1. Order relations on the set of tilings. In this section, a tiling T_0 of P is fixed and the tiling projection induced by T_0 is denoted by g_0.

Definition 5.1. *Let (T, T') be a pair of tilings of P. We say that $T \leq_{T_0} T'$ if, for each vertex v of P, $g_0(v)^{-1} g_T(v) \leq_{decomp} g_0(v)^{-1} g_{T'}(v)$.*

The relation defined in this way is obviously an order relation on the set Υ_P of tilings of P. The proposition below gives a geometrical interpretation of this order.

Proposition 5.2. *Let (T, T') be a pair of tilings of P. We have $T \leq_{T_0} T'$ if and only if there exists a sequence $(T_0, T_1, \ldots, T_p)$ of tilings of P such that $T_p = T'$, $p = minflip(T_0, T')$, for each integer i such that $0 \leq i < p$; $T_i + 1$ is deduced from T_i by a local flip, and there exists an integer i_0 such that $0 \leq i_0 \leq p$ and $T = T_{i_0}$.*

Proof. $T \leq_{T_0} T'$ if and only if we have the equality

$$\Delta(T_0, T') = \Delta(T_0, T) + \Delta(T, T')$$

from the definition of the distance between tilings. Moreover, from the flip formula, the above equality is equivalent to

$$minflip(T_0, T') = minflip(T_0, T) + minflip(T, T'),$$

which means that there exists a sequence of tilings as described in the proposition. □

5.2. The infimum property.

Proposition 5.3. *Let (T, T') be a pair of tilings of P. We define the function $g_{inf(T,T')}$ by, for each vertex v of P,*

$$g_{inf(T,T')}(v) = g_0(v) inf_{decomp}(g_0(v)^{-1} g_T(v), g_0(v)^{-1} g_{T'}(v)).$$

There exists a tiling T'' of P such that $g_{T''} = g_{inf(T,T')}$.

The proof of the above proposition is based on the following lemma.

Lemma 5.4. *Let (w, w') be a pair of elements of $ker(s)$ such that $inf_{decomp}(w, w') = 1_{G_{cycles}}$, and let (j, j') be a pair of integers. We state $u = a^{-1}wb^jab^{-j}$ and $u' = a^{-1}wb^{j'}ab^{-j'}$. (Notice that u and u' both are elements of $ker(s)$.)*

- *If there exists an integer j'' such that $0 < j'' < n$ and $b^{j''} \leq_{cycles} inf_{cycles}(w, w')$, then $inf_{decomp}(u, u') = a^{-1}b^{j''}ab^{-j''}$;*
- *otherwise, $inf_{decomp}(u, u') = 1_{G_{cycles}}$.*

Proof. First assume that there exists an integer j'' such that $0 < j'' < n$ and $b^{j''} \leq_{cycles} inf_{cycles}(w, w')$. Thus, we can state $w = b^{j''}w_1$ with the canonical expression of w_1 beginning by a and $l(w_1) \geq 3$ (since $l(w) \geq 4$)

If $l(w) = 4$, then w is an element of S_{rect}, thus $l(wb^jab^{-j}) = 7$ and $init(wb^jab^{-j}) = init(w)$. If $l(w) \geq 5$, we obviously have $l(wb^jab^{-j}) \geq 2$ and $init(wb^jab^{-j}) = init(w)$. Thus, in any case, we can state $w = b^{j''}a^iw_1$ with $0 < i < m$ and either $w_1 = 1_{G_{cycles}}$ or the canonical expression of w_1 begins by b.

In a similar way, we can state $w'b^{j'}ab^{-j'} = b^{j''}a^{i'}w'_1$. Notice that $i \neq i'$, from Lemma 3.7. Thus, $u = (a^{-1}b^{j''}ab^{-j''})(b^{j''}a^{-1+i}w_1)$ and $u' = (a^{-1}b^{j''}ab^{-j''})(b^{j''}a^{-1+i'}w'_1)$, which gives the result, since $init(b^{j''}a^{-1+i}w_1) \neq init(b^{j''}a^{-1+i'}w_1)$.

Now we treat the second alternative of the lemma: if $w = ab^ja{-}1b^{-j}$, then $u = 1_{G_{cycles}}$, which obviously gives the result. The same argument can be used if $w' = ab^{j'}a{-}1b^{-j'}$. In any remaining case, one can remark as it has been done for the first alternative that $init(wb^jab^{-j}) = init(w)$ and $init(w'b^jab^{-j}) = init(w')$. Thus $init(wb^jab^{-j}) \neq init(w'b^jab^{-j})$, which yields that $init(u) \neq init(u')$, which gives the result from Lemma 3.7. □

Proof of Proposition 5.3. We prove this proposition, proving that $g_{inf(T,T')}$ satisfies the hypothesis of Proposition 3.4. The only nontrivial point is the verification that, for each pair (v, v') of neighbor vertices of P, $d(g_{inf(T,T')}(v), g_{inf(T,T')}(v')) \leq 3$.

We will prove it assuming, moreover, that $v' = va$ (which can be done without loss of generality, since the case when $v' = vb$ can be treated in a symmetric way).

We need some notations: we state $g_0(v') = g_0(v)b^{j_0}ab^{-j_0}$, $g_T(v') = g_T(v)b^{j_T}ab^{-j_T}$, and $g_{T'}(v') = g_{T'}(v)b^{j_{T'}}ab^{-j_{T'}}$.

We also state $g_{inf(T,T')}(v)^{-1}g_0(v) = w_0$, $g_{inf(T,T')}(v)^{-1}g_T(v) = w_T$, $g_{inf(T,T')}(v)^{-1}g_{T'}(v) = w_{T'}$. With these notations, we have $inf_{decomp}(w_0, w_T) = inf_{decomp}(w_0, w_{T'}) = inf_{decomp}(w_T, w_{T'}) = 1_{G_{cycles}}$.

Afterwards, we state $u_0 = a^{-1}w_0b^{j_0}ab^{-j_0}$, $u_T = a^{-1}w_Tb^{j_T}ab^{-j_T}$, $u_{T'} = a^{-1}w_{T'}b^{j_{T'}}ab^{-j_{T'}}$.

(a) If there exists an integer j'' such that $0 < j'' < n$ and $b^{j''} \leq_{cycles} inf_{cycles}(w_0, w_T, w_{T'})$, then, from the previous lemma, we have

$$inf_{decomp}(u_0, u_T) = inf_{decomp}(u_0, u_{T'}) = inf_{decomp}(u_T, u_{T'}) = a^{-1}b^{j''}ab^{-j''}.$$

Thus, if we state $t = a^{-1}b^{j''}ab^{-j''}$, we have

$$\begin{aligned} inf_{decomp}(t^{-1}u_0, t^{-1}u_T) &= inf_{decomp}(t^{-1}u_0, t^{-1}u_{T'}) = inf_{decomp}(t^{-1}u_T, t^{-1}u_{T'}) \\ &= 1_{G_{cycles}} \end{aligned}$$

which means that $g_{inf(T,T')}(va) = g_{inf(T,T')}(v)b^{j''}ab^{-j''}$ and gives the result.

(b) If there exists an integer j'' such that $0 < j'' < n$ and $b^{j''} \leq_{cycles} inf_{cycles}(w_0, w_T)$ (which yields that $inf_{decomp}(u_0, u_T) = a^{-1}b^{j''}ab^{-j''} = t$), and $inf_{decomp}(u_0, u_{T'}) = inf_{decomp}(u_T, u_{T'}) = 1_{G_{cycles}}$, then

$$\begin{aligned} inf_{decomp}(t^{-1}u_0, t^{-1}u_T) &= inf_{decomp}(t^{-1}u_0, t^{-1}u_{T'}) = inf_{decomp}(t^{-1}u_T, t^{-1}u_{T'}) \\ &= 1_{G_{cycles}}, \end{aligned}$$

which means that $g_{inf(T,T')}(va) = g_{inf(T,T')}(v)b^{j''}ab^{-j''}$ and gives the result.

(c) If $inf_{decomp}(u_0, u_T) = inf_{decomp}(u_0, u_{T'}) = inf_{decomp}(u_T, u_{T'}) = 1_{G_{cycles}}$, then we have $g_{inf(T,T')}(va) = g_{inf(T,T')}(v)a$, which gives the result.

We have treated all the cases (up to symmetry) from the previous lemma. □

COROLLARY 5.5. *For each tiling T_0 of P, the order relation $(\Upsilon_P, \leq_{T_0})$ is an inferior semilattice.*

Proof. The proof is obvious. □

PROPOSITION 5.6. *For each pair (T_0, T_0') of tilings of P, we define the set $\Upsilon_{T_0,T_0'}$ consisting of tilings T such that $T_0 \leq_{T_0} T \leq_{T_0} T_0'$.*

The order relation $(\Upsilon_{T_0,T_0'}, \leq_{T_0})$ is a distributive lattice.

Proof. We have seen that $\Upsilon_{T_0,T_0'}$ has the infimum property. Notice that $T_0 \leq_{T_0} T \leq_{T_0} T_0'$ if and only if $T_0' \leq_{T_0'} T \leq_{T_0'} T_0$. Thus $\Upsilon_{T_0,T_0'}$ has the supremum property, since Υ_{T_0',T_0} has the infimum property. We have proven that $\Upsilon_{T_0,T_0'}$ has a lattice structure.

For each vertex v of P, we state $g_{T_0'}(v) = g_{T_0}(v)\Pi_{i=1}^{p(v)}t_i(v)$, where $\Pi_{i=1}^{p(v)}t_i(v)$ is the decomposition of $g_{T_0}(v)^{-1}g_{T_0'}(v)$. Let T be a tiling of $\Upsilon_{T_0,T_0'}$. There exists a unique integer $q_T(v)$ such that $0 \leq q(v) \leq p(v)$ and $g_T(v) = g_{T_0}(v)\Pi_{i=1}^{q_T(v)}t_i(v)$. Thus, one can define the injective mapping Q from $\Upsilon_{T_0,T_0'}$ to $\mathbb{Z}^V$ (where V denotes the set of vertices of P) such that $Q(T)$ is the vector consisting of values $q_T(v)$.

By definition of the order on $\Upsilon_{T_0,T_0'}$, the mapping Q is a lattice morphism (i.e., $Q(inf(T,T')) = inf(Q(T), Q(T'))$ and $Q(sup(T,T')) = sup(Q(T), Q(T')))$, which yields that $\Upsilon_{T_0,T_0'}$ is isomorphic to a sublattice of $\mathbb{Z}^V$ and, consequently, is a distributive lattice. □

6. Tilings with leaning dominoes and triangles. From a similar method, we will see that similar results and algorithms can be obtained working with sets of prototiles of the triangular lattice. In this section, we limit ourselves to present the general framework and the main tools used, and do not give proofs, since most of them are very similar to those of previous sections, about tilings with bars.

The triangular lattice Γ induces three unit moves in the plane: rightward, denoted by a, and b and c, such that $angle(a,b) = angle(b,c) = angle(c,a) = 2\pi/3$. Let v be a vertex of Γ and u denote an element of $\{a, b, c, a^{-1}, b^{-1}, c^{-1}\}$. The element of Γ which is reached from v with a u-move is denoted by vu.

We now study the set of prototiles $S = \{ld_1, ld_2, ld_3, ld_4, ld_5, ld_6, tr_1, tr_2\}$ (previously studied in [7]), where ld_i (respectively, tr_i) denotes a parallelogram (respectively, an equilateral triangle) formed with four cells of Γ (see Figure 5). Each prototile ld_i is called a leaning domino.

A set of contour words of S is $R = \{a^2ba^{-2}b^{-1}$, $a^2ca^{-2}c^{-1}$, $b^2ab^{-2}a^{-1}$, $b^2cb^{-2}c^{-1}$, $c^2ac^{-2}a^{-1}$, $c^2bc^{-2}b^{-1}$, $a^2b^2c^2, a^2c^2b^2\}$, thus the tiling group G_{tile} is the group generated by $\{a, b, c\}$ whose set of relators is R (i.e., the quotient group $F_{a,b,c}/N_R$).

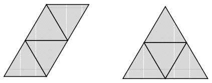


FIG. 5. *A leaning domino and a triangular prototile. Counterclockwise, starting from the lower left corners, the contour words, respectively, are $ac^{-2}a^{-1}c^2$ and $a^2b^2c^2$.*

6.1. Quotient groups.

6.1.1. The tricolored group. Since G_{tile} is complex, we use quotients of it. The main quotient group used is $\langle a, b, c | a^2, b^2, c^2\rangle$. This group is isomorphic of the free product of three groups, each of them with only two elements. If we identify opposite arcs with the same label, the induced Cayley graph is a tree (see Figure 6).

Each element w of $G_{tricolored}$ has a canonical expression: w can be written in a unique way as $w = \Pi_{i=1}^{p} x_i$ with, for each integer i, $x_i \in \{a, b, c\}$ and, for $i < p$, $x_i \neq x_{i+1}$. This permits us to define the initial (respectively, final part) of w (the word formed by the two first (respectively, last) letters of the canonical expression of w), the length $l(w)$ of w by $l(w) = p$, and the distance $d(w', w'')$ between elements of $G_{tricolored}$ by $d(w', w'') = l(w'^{-1}w'')$. One can also canonically define an order relation (denoted by $\leq_{tricolored}$) on $G_{tricolored}$. This relation obviously has the infimum property.

Let P be a fixed polygon formed with cells of the triangular lattice, and let v_0 be a fixed vertex of the boundary of P. As in section 3, for each tiling T of P, one can define a tiling projection g_T, which associates an element of $G_{tricolored}$ to each vertex v of the polygon. Such a tiling projection encodes the tiling T.

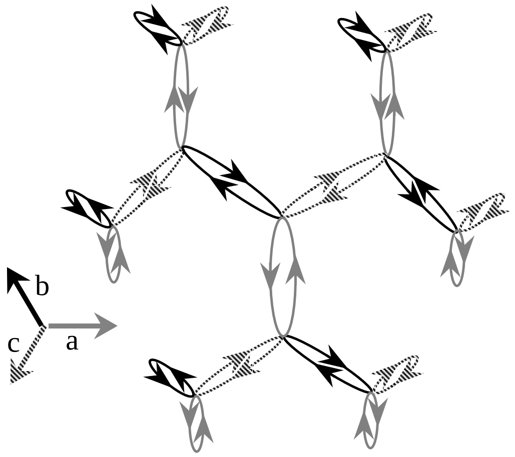


FIG. 6. *The Cayley graph of $G_{tricolored}$.*

6.1.2. The cell group and the tetrahedron group. Another quotient group of G_{tile} is $G_{cell} = \langle a, b, c | abc, acb\rangle$ (i.e., the set of relators is a set of contour words of cells). This group is isomorphic to $\mathbb{Z}^2$, and the induced Cayley graph is the planar triangular grid.

The third quotient group is $G_{tetrahedron} = \langle a, b, c | a^2, b^2, c^2, abc, acb\rangle$. This group has four elements, and the induced Cayley graph is a tetrahedron. Since there exists a canonical morphism from G_{cell} (which can be seen as the planar triangular grid) to $G_{tetrahedron}$, we can define (after an origin vertex v_0 has been fixed), for each vertex v of the grid, the element $cong_{tetrahedron}(v)$ of $G_{tetrahedron}$.

Given a tiling T of a polygon P with a fixed vertex v_0 of its boundary, one can define a tiling projection g_T from vertices of P to $G_{tricolored}$. We have, in a similar way as in section 3, the proposition below.

Proposition 6.1. *Let g be a function from the set of vertices of P to $G_{tricolored}$. There exists a tiling T of P such that $g = g_T$ if and only if the following constraints are satisfied:*

- *$g(v_0) = 1_{G_{tricolored}}$;*
- *for each vertex vertex v of P, $s(g(v)) = cong_{tetrahedron}(v)$;*
- *for each pair (v, v') of neighbor vertices of P, $d(g(v), g(v')) \leq 3$, and if, moreover, the line segment $[v, v']$ is included on the boundary of P, then $d(g(v), g(v')) = 1$.*

Proof (sketch). The direct part of the proposition is very easy. Conversely, assume that g satisfies the constraint above and let $[v, va]$ be a line segment included in P. We necessarily have $g(v)^{-1}g(va) \in \{a, bc, cb, cac, bab\}$. (We have a similar fact for line segments $[v, vb]$ and $[v, vc]$.)

Moreover, we have a large amount of information about the values of $g(vb^{-1})$ and $g(vc^{-1})$: Precisely, if $g(v)^{-1}g(va) = bab$, then $g(v)^{-1}g(vc^{-1}) = ba$ and $g(v)^{-1}g(vb^{-1}) = b$; if $g(v)^{-1}g(va) = bc$, then $g(v)^{-1}g(vb^{-1}) = b$ and $g(v)^{-1}g(vb^{-1}) \in \{c, ba, bcb\}$. (We also have a lot of symmetric equalities.)

The above equalities imply that the set of edges $[v, v']$ such that $d(g(v), g(v')) = 1$ draw a tiling T of P: precisely, for each cell C of P, the set of cells C', such that there exists a path starting in C and finishing in C' which cuts no edge $[v, v']$ such that $d(g(v), g(v')) = 1$, form a tile. The set of those tiles form a tiling T of P, and we obviously have $g = g_T$. □

As a corollary, we obtain that, for each pair (T, T') of tilings of P and for each vertex v of P, $g_{T'}(v)^{-1}g_T(v)$ is an element of the kernel of the canonical morphism from $G_{tricolored}$ to $G_{tetrahedron}$. For the following, this kernel is denoted by N'_{cell}.

6.1.3. Structures that are induced by N'_{cell}.

We state $S_{triangle} = \{abc, acb, bac, bca, cab, cba\}$ (i.e., the set of possible contour words of triangular cells). As in section 3, one can prove that each element w of N'_{cell} can be written in a unique way as $w = \Pi_{i=1}^{p'} x_i$ with, for each integer i, $x_i \in S_{triangle}$ and, for $i < p$, $x_i \neq x_{i+1}$.

Moreover, the canonical expression of w finishes by ab (respectively, ac, ba, bc, ca, cb) if and only if $x_{p'} = cab$ (respectively, bac, cba, abc, bca, acb).

Thus, as in section 3, one can define the order relation $\leq_{decomp}$ on N'_{cell}, the decomposition number $num(w)$ of w by $num(w) = p'$, and, afterwards, the distance $\Delta(T, T')$ between two tilings of a same polygon P.

6.2. Local flips.

We have two kinds of local flips (see Figure 7): a lozenge L_0, formed with eight triangular cells of P, admits three tilings. Two of those tilings consist of two leaning dominoes, and the third one consists of two triangles. The replacement of a tiling of L_0 consisting of parallelograms by a tiling consisting of triangles (or the inverse) is our first kind of local flip (the lozenge flips).

An isosceles trapezoid Tr_0 formed with eight triangular cells of P admits two tilings, each of them consisting of a parallelogram and a triangle. The replacement of one of these two tilings by the other one is our second kind of local rotations (the trapezoid flips).

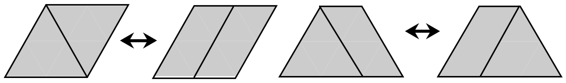


FIG. 7. *Flips for leaning dominoes and triangles.*

Let T be a tiling and T_{flip} be a tiling deduced by a flip of support Sup (which is either a lozenge or a trapezoid). Let v_{Sup} denote the only vertex which is in the interior part of Sup. For each vertex v of P such that $v \neq v_{Sup}$, we have $g_T(v) = g_{T_{flip}}(v)$, and $g_T(v_{Sup})^{-1} g_{T_{flip}}(v_{Sup})$ is an element of $S_{triangle}$.

6.2.1. Maximal vertex. A maximal vertex for a pair (T, T') of tilings can be defined exactly as in section 3, but the use of maximal vertices is a little different.

Let v_1 be a maximal vertex for a pair (T, T') of distinct tilings. One can assume without loss of generality that $l(g_{T'}(v_1)) \leq l(g_T(v_1))$.

From the tree structure of $G_{tricolored}$, one easily proves that $l(g_T(v_1)) \geq 2$. Moreover, if the final part of $g_T(v_1)$ is ab, then, since $l(g_{T'}(v_1)) \leq l(g_T(v_1))$, the last factor of the decomposition of $g_{T'}(v_1)^{-1} g_T(v_1)$ is necessarily cab.

PROPOSITION 6.2. *Consider the line segment $[v_1 b^{-1}, v_1 b]$. This line segment is the common side of two tiles of T. Moreover, each leaning domino of T, a large side of which is $[v_1 b^{-1}, v_1 b]$, admits $b^2 a^{-1} b^{-2} a$ as contour word.*

Proof (sketch). We first claim that the line segment $[v_1, v_1 c]$ necessarily cuts a tile of T; otherwise, we have $l(g_T(v_1 c)) = l(g_T(v_1)) + 1$.

On the other hand, since the final part of $g_{T'}(v_1)^{-1} g_T(v_1)$ is ab, we have

$$d(g_{T'}(v_1), g_T(v_1 c)) = d(g_{T'}(v_1), g_T(v_1)) + 1 \geq 3 + 1 = 4.$$

These inequalities prove that $g_T(v_1 c) \neq g_{T'}(v_1 c)$, since $d(g_{T'}(v_1), g_{T'}(v_1 c)) \leq 3$. Thus $v_1 c$ contradicts the maximality of v_1.

The same argument can also be used for $v_1 a$, $v_1 c^{-1}$ and $v_1 a^{-1}$. This gives the first part of the proposition.

If we assume that a leaning domino, whose sides issued from $v_1 b^{-1}$ are $[v_1 b^{-1}, v_1 b]$ and $[v_1 b^{-1}, v_1 b^{-1} c]$, is an element of T, then one proves as above that $l(g_T(v_1 c)) = l(g_T(v_1)) + 1$ and $g_T(v_1 c) \neq g_{T'}(v_1 c)$, which contradicts the maximality of v_1. This gives the second part of the proposition. □

From the above proposition, it follows that a flip can be done around such an extremal vertex v_1. This flip decreases the distance between T and T'. Thus, by a similar study as in section 3, we obtain

- a flip formula: $minflip(T, T') == \sum_{v \in P} num(g_{T'}(v)^{-1} g_T(v))$;
- an algorithm which, given a pair of tilings, produces a sequence of minimal length of necessary flips to transform the first tiling into the second one.

6.3. Lattice structures. For each tiling T_0, we can define, as in section 5, an order relation $\leq_{T_0}$, which can be geometrically interpreted by the following: for each pair (T, T') of tilings of P, $T \leq_{T_0} T'$ if and only if there exists a sequence $(T_0, T_1, \ldots, T_p)$ of tilings of P such that $T_p = T'$, $p = minflip(T_0, T')$, for each integer i such that $0 \leq i < p$; $T_i + 1$ is deduced from T_i by a local flip, and there exists an integer i_0 such that $0 \leq i_0 \leq p$ and $T = T_{i_0}$.

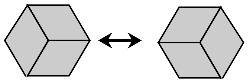


FIG. 8. *Flips for calissons.*

LEMMA 6.3. *Let (w, w') be a pair of elements of N'_{cell} such that $inf_{decomp}(w, w') = 1_{G_{tricolored}}$ and (s, s') be a pair of elements of $\{a, bc, cb, cac, bab\}$. We state $u = aws$ and $u' = aws'$. (Notice that u and u' both are elements of N'_{cell}.)*

- *If $b \leq_{tricolored} inf_{tricolored}(w, w')$, then $inf_{decomp}(u, u') = abc$.*
- *If $c \leq_{tricolored} inf_{tricolored}(w, w')$, then $inf_{decomp}(u, u') = acb$.*
- *Otherwise, $inf_{decomp}(u, u') = 1_{G_{tricolored}}$.*

From this lemma, as in section 5, one deduces that the order $\leq_{tricolored}$ induces a structure of inferior semilattice on the set of tilings of P, with T_0 as minimum element, and a structure of distributive lattice for each interval of tilings.

7. Tilings with calissons. The same method can easily be applied to the set of prototiles $S = \{cal_1, cal_2, cal_3\}$ (the calissons, studied in [11]), each element of S being formed with two cells of Γ with a common edge. A set of contour words is $\{aba^{-1}b^{-1}, aca^{-1}c^{-1}, bcb^{-1}c^{-1}\}$.

The main quotient group used is $G_{line} = \langle a, b | ab^{-1}, ac^{-1} \rangle$. Each element w of G_{line} has a canonical expression: there exists a unique relative integer such that $w = a^p$. This permits us to define the length $l(w)$ of w by $l(w) = |p|$.

Given a tiling T of a polygon P, one can define a tiling projection g_T from vertices of P to G_{line}. One can prove that, for each vertex v of P, $g_{T'}(v)^{-1} g_T(v)$ is an element of the normal group N''_{cell} of G_{line} generated by $\{abc, acb\} = \{a^3\}$. Thus, each element w of N''_{cell} can be written in a unique way as $w = a^{3p'}$. This permits us to define the decomposition number $num(w)$ of w by $num(w) = |p'|(= l(w)/3)$ and, afterwards, the distance $\Delta(T, T')$ between two tilings of a same polygon P.

The local flips are induced by the two possible tilings or a hexagon formed with six cells of Γ (see Figure 8). In this case, the end of the study is very simple, since G_{line} is isomorphic to $\mathbb{Z}$. We obtain

- a flip formula: $minflip(T, T') = \sum_{v \in P} |l(g_{T'})(v) - l(g_T(v))|/3$;
- an algorithm which, given a pair of tilings, produces a sequence of minimal length of necessary flips to transform the first tiling into the second one;
- a structure of distributive lattice in the set of tilings (the addition of an artificial maximum is not needed because of the structure of line of the quotient group).

As for the dominoes, results about calissons were previously obtained using elementary methods [8], [10], but here we explain them with a general framework.

REFERENCES

[1] G. BIRKHOFF, *Lattice Theory*, 3rd ed., AMS, Providence, RI, 1967.

[2] J. H. CONWAY AND J. C. LAGARIAS, *Tiling with polyominoes and combinatorial group theory*, J. Combin. Theory Ser. A, 53 (1990), pp. 183–208.

[3] B. A. DAVEY AND H. A. PRIESTLEY, *Introduction to Lattices and Orders*, Cambridge University Press, Cambridge, UK, 1990.

[4] C. KENYON AND R. KENYON, *Tiling a polygon with rectangles*, in Proceedings of the 33rd Symposium on Foundations of Computer Science, Pittsburgh, PA, 1992, pp. 610–619.

[5] J. C. LAGARIAS AND D. S. ROMANO, *A polyomino tiling of Thurston and its configurational entropy*, J. Combin. Theory Ser. A, 63 (1993), pp. 338–358.

[6] J. G. Propp, *Lattice Structure for Orientations of Graphs*, preprint, 1993.

[7] E. Rémila, *Tiling groups: New applications in the triangular lattice*, Discrete Comput. Geom., 20 (1998), pp. 189–204.

[8] E. Rémila, *On the lattice structure of the set of tilings of a simply connected figure with dominoes*, in Proceedings of the 3rd International Conference on Orders, Algorithms and Applications (ORDAL), L. Nourine and M. Habib, eds., Montpellier, France, 1999.

[9] E. Rémila, *An algebraic method to compute a shortest path of local flips between two tilings*, in Proceedings of the Eleventh Annual ACM-SIAM Symposium On Discrete Algorithms, San Francisco, CA, 2000, ACM, New York, 2000, pp. 646–653.

[10] N. C. Saldanha, C. Tomei, M. A. Casarin Jr, and D. Romualdo, *Spaces of domino tilings*, Discrete Comput. Geom., 14 (1995), pp. 207–233.

[11] W. P. Thurston, *Conway's tiling group*, Amer. Math. Monthly, 97 (1990), pp. 757–773.

SIAM J. DISCRETE MATH.
Vol. 16, No. 1, pp. 20–46

TESTING BASIC BOOLEAN FORMULAE[*]

MICHAL PARNAS[†], DANA RON[‡], AND ALEX SAMORODNITSKY[§]

Abstract. We consider the problem of determining whether a given function $f : \{0,1\}^n \to \{0,1\}$ belongs to a certain class of Boolean functions $\mathcal{F}$ or whether it is *far* from the class. More precisely, given query access to the function f and given a distance parameter ϵ, we would like to decide whether $f \in \mathcal{F}$ or whether it differs from every $g \in \mathcal{F}$ on more than an ϵ-fraction of the domain elements. The classes of functions we consider are singleton ("dictatorship") functions, monomials, and monotone disjunctive normal form functions with a bounded number of terms. In all cases we provide algorithms whose query complexity is independent of n (the number of function variables), and linear in $1/\epsilon$.

Key words. property testing, Boolean functions, randomized algorithms, approximation algorithms

AMS subject classifications. 68Q25, 68W20, 68W25, 68W40

PII. S0895480101407444

1. Introduction.

The newly founded country of Eff is interested in joining the international organization Pea. This organization has one rule: it does not admit dictatorships. Eff claims it is not a dictatorship but is unwilling to reveal the procedure by which it combines the votes of its government members into a final decision. However, it agrees to allow Pea's special envoy, Tee, to perform a small number of experiments with its voting method. Namely, Tee may set the votes of the government members (using Eff's advanced electronic system) in any possible way and obtain the final decision given these votes. Tee's mission is not to actually identify the dictator among the government members (if one exists) but only to discover *whether* such a dictator exists. Most importantly, she must do so by performing as few experiments as possible. Given this constraint, Tee may decline Eff's request to join Pea even if Eff is not exactly a dictatorship but behaves like one most of the time.

The above can be formalized as a *property testing problem*: Let $f : \{0,1\}^n \to \{0,1\}$ be a fixed but unknown function, and let $\mathcal{P}$ be a fixed property of functions. We would like to determine, by querying f, whether f has the property $\mathcal{P}$ or whether it is ϵ-*far* from having the property for a given distance parameter ϵ. By ϵ-*far* we mean that more than an ϵ–fraction of its values should be modified so that it obtains the property $\mathcal{P}$. For example, in the above setting we would like to test whether a given function f is a "dictatorship function," that is, whether there exists an index $1 \le i \le n$ such that $f(x) = x_i$ for every $x \in \{0,1\}^n$.

[*] Received by the editors August 2, 2001; accepted for publication (in revised form) July 1, 2002; published electronically November 6, 2002. An extended abstract of this work (under the slightly different title "Proclaiming Dictators and Juntas or Testing Boolean Formulae") appeared in *Proceedings of the* 5th *International Workshop on Randomization and Approximation Techniques in Computer Science* (RANDOM01), Berkeley, CA, 2001. The results in this version of the paper improve on those presented in RANDOM01.

http://www.siam.org/journals/sidma/16-1/40744.html

[†] The Academic College of Tel-Aviv-Yaffo, 4 Antokolsky St., Tel-Aviv, Israel (michalp@mta.ac.il).

[‡] Department of EE–Systems, Tel-Aviv University, Ramat Aviv, Israel (danar@eng.tau.ac.il). This author's research was supported by the Israel Science Foundation (grant 32/00-1).

[§] School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Israel (salex@cs.huji.ac.il). This author's work was done while at the Institute for Advanced Study, Princeton, NJ. This author's research was supported by NSF grant CCR–9987845 and by a State of New Jersey grant.

Previous work on testing properties of functions mainly focused on algebraic properties (e.g., [6, 21, 20]), or on properties defined by relatively rich families of functions such as the family of all monotone functions [12, 9]. Here we are interested in studying the most basic families of Boolean functions: singletons, monomials, and disjunctive normal form (DNF) functions.

One possible approach to testing whether a function f has a certain property $\mathcal{P}$ is to try and actually *find* a good approximation for f from within the family of functions $\mathcal{F}_{\mathcal{P}}$ having the tested property $\mathcal{P}$. For this task we would use a *learning algorithm* that performs queries and works under the uniform distribution. Such an algorithm ensures that if f has the property (that is, $f \in \mathcal{F}_{\mathcal{P}}$), then with high probability the learning algorithm outputs a *hypothesis* $h \in \mathcal{F}_{\mathcal{P}}$ such that $\Pr[f(x) \neq h(x)] \leq \epsilon$, where ϵ is a given distance (or error) parameter. The testing algorithm would run the learning algorithm, obtain the hypothesis $h \in \mathcal{F}_{\mathcal{P}}$, and check that h and f in fact differ only on a small fraction of the domain. This last step is performed by taking a sample of size $\Theta(1/\epsilon)$ from $\{0,1\}^n$ and comparing f and h on the sample. Thus, if f has the property $\mathcal{P}$, then it will be accepted with high probability, and if f is ϵ-far from having $\mathcal{P}$, so that $\Pr[f(x) \neq h(x)] > \epsilon$ for every $h \in \mathcal{F}_{\mathcal{P}}$, then it will be rejected with high probability.

Hence, provided that there exists a learning algorithm for the tested family $\mathcal{F}_{\mathcal{P}}$, we obtain a testing algorithm whose complexity is of the same order of that of the learning algorithm. To be more precise, the learning algorithm should be a *proper* learning algorithm. That is, the hypothesis h it outputs must belong to $\mathcal{F}_{\mathcal{P}}$.[1]

A natural question that arises is whether we can do better by using a different approach. Recall that we are not interested in actually finding a good approximation for f in $\mathcal{F}_{\mathcal{P}}$, but we want only to know whether such an approximation *exists*. Therefore, perhaps we can design a different and more efficient testing algorithm than the one based on learning. In particular, the complexity measure we would like to improve is the *query complexity* of the algorithm.

As we show below, for all the properties we study, we describe algorithms whose query complexity is linear in $1/\epsilon$, where ϵ is the given distance parameter, and *independent* of the input size n.[2] As we discuss shortly, the corresponding proper learning algorithms have query complexities that depend on n, though only polylogarithmically. We believe that our results are of interest both because they completely remove the dependence on n in the query complexity, and also because in certain aspects they are inherently different from the corresponding learning algorithms. Hence they may shed new light on the structure of the properties studied.

1.1. Our results.

We present the following testing algorithms:

- An algorithm that tests whether f is a singleton function, that is, whether there exists an index $1 \leq i \leq n$ such that $f(x) = x_i$ for every $x \in \{0,1\}^n$ or $f(x) = \bar{x}_i$ for every $x \in \{0,1\}^n$. This algorithm has query complexity $O(1/\epsilon)$.
- An algorithm that tests whether f is a monomial with query complexity $O(1/\epsilon)$.

[1] This is as opposed to *nonproper* learning algorithms that given query access to $f \in \mathcal{F}_{\mathcal{P}}$ are allowed to output a hypothesis h that belongs to a more general hypothesis class $\mathcal{F}' \supset \mathcal{F}_{\mathcal{P}}$. Nonproper learning algorithms are not directly applicable for our purposes.

[2] The running times of the algorithms are all linear in the number of queries performed and in n. This dependence on n in the running time is clearly unavoidable, since even writing down a query takes time n.

- An algorithm that tests whether f is a monotone DNF having at most ℓ terms, with query complexity $\tilde{O}(\ell^2/\epsilon)$.

We note that the above results improve on those presented in an extended abstract of this work [17].

Techniques. Our algorithms for testing singletons and for testing monomials have a similar structure. In particular, they combine two tests. One test is a "natural" test that arises from an exact logical characterization of these families of functions. In the case of singletons, this test uniformly selects pairs $x, y \in \{0,1\}^n$ and verifies that $f(x \wedge y) = f(x) \wedge f(y)$, where $x \wedge y$ denotes the bitwise "and" of the two strings. The corresponding test for monomials performs a slight variant of this test. The other test in both cases is a seemingly less evident test with an algebraic flavor. In the case of singletons it is a linearity test [6], and in the case of monomials it is an affinity test. This test ensures that if f passes it, then it has or is close to having a certain structure. This structure aids us in analyzing the logical test. We note that our current analysis of the affinity test differs from the one presented in previous versions of this work [17, 18]. In particular, in previous versions we used the discrete Fourier transform, while here we build on basic probabilistic arguments.

The testing algorithm for monotone DNF functions uses the test for monomials as a subroutine. Recall that a DNF function is a disjunction of monomials (the terms of the function). If f is a DNF function with a bounded number of monotone terms, then the test will isolate the different terms of the function and test that each is in fact a monotone monomial. If f is far from being such a DNF function, then at least one of these tests will fail with high probability.

1.2. Related work.

1.2.1. Property testing. Property testing was first defined and applied in the context of algebraic properties of functions [21], and has since been extended to various domains, perhaps most notably those of graph properties (e.g., [13, 14, 1]). (For surveys see [19, 10].) The relation between testing and learning is discussed at length in [13]. In particular, that paper suggests that testing may be applied as a preliminary stage to learning. Namely, efficient testing algorithms can be used in order to help in determining what hypothesis class should be used by the learning algorithm.

Linearity testing and its variants. As noted above, we use linearity testing [6] in our test for singletons and affinity testing, which can be viewed as an extension of linearity testing, for testing monomials. Other works in which improvements and variants of linearity testing are analyzed include [4, 3].

Testing the long code. We note that a test which is very similar to our test for singletons was applied to testing the *long code* [5]. Specifically, Bellare, Goldreich, and Sudan [5] considered the following task. For an integer ℓ, let $\mathcal{F}_\ell$ be the set of all Boolean functions over $\{0,1\}^\ell$. Let G be a function from $\mathcal{F}_\ell$ to $\{0,1\}$. The goal is to test whether G has the following property: there exists some fixed $a \in \{0,1\}^\ell$ such that $G(f) = f(a)$ for every $f \in \mathcal{F}_\ell$. In such a case G is the code word of length 2^{2^ℓ} corresponding to the plaintext a.

In order to better understand the relation to our problem, we view each string $a \in \{0,1\}^\ell$ as an index between 0 and $2^\ell - 1$, and each function $f \in \mathcal{F}_\ell$ as a string of length 2^ℓ that corresponds to the truth table of f. Then the test should accept G if there exists some index a such that for every f, $G(f) = f(a)$. The test should reject f if for every a, $G(f) \neq f(a)$ on more than an ϵ-fraction of the strings (functions) f. In other words, testing the long code is equivalent to testing monotone singletons over $\{0,1\}^n$ when $n = 2^\ell$. Thus we extend the long code/singletons test to any n, while simplifying the analysis.

Testing k-juntas. Following the publication of the extended abstract of this work [17], a recent work [11] addresses the problem of testing whether a Boolean function depends on at most k variables for a given parameter k. In [11] it is shown that this "k-junta" property can be tested using a number of queries that is linear in $1/\epsilon$ and polynomial in k. It is also noted that the k-junta testing algorithm can be applied as a subroutine to testing monomials using a number of queries that is $\tilde{O}(1/\epsilon)$.

1.2.2. Learning Boolean formulae.

Singletons, and more generally monomials, can be easily learned under the uniform distribution. The learning algorithm uniformly selects a sample of size $\Theta(\log n/\epsilon)$ and queries the function f on all sample strings. It then searches for a monomial that is consistent with f on the sample. Finding a consistent monomial, if one exists, can be done in time linear in the sample size and in n. A simple probabilistic argument, which is a slight variant of Occam's Razor [7],[3] can be used to show that a sample of size $\Theta(\log n/\epsilon)$ is sufficient to ensure that with high probability any monomial that is consistent with the sample is an ϵ-good approximation of f.

There is a large variety of results on learning DNF functions, and in particular monotone DNF, in several different models. We restrict our attention to the model most relevant to our work, namely when membership queries are allowed and the underlying distribution is uniform. The best known algorithm results from combining the works of [8] and [16], and builds on Jackson's celebrated Harmonic Sieve algorithm [15]. This algorithm has query complexity $\tilde{O}(r \cdot (\frac{\log^2 n}{\epsilon} + \frac{\ell^2}{\epsilon^2}))$, where r is the number of variables appearing in the DNF formula, and ℓ is the number of terms. However, this algorithm does not output a DNF formula as its hypothesis. On the other hand, Angluin [2] describes a proper learning algorithm for monotone DNF formulae that uses membership queries and works under arbitrary distributions. The query complexity of her algorithm is $\tilde{O}(\ell \cdot n + \ell/\epsilon)$. Using the same preprocessing technique as suggested in [8], if the underlying distribution is uniform, then the query complexity can be reduced to $\tilde{O}(\frac{r \cdot \log^2 n}{\epsilon} + \ell \cdot (r + \frac{1}{\epsilon}))$. Recall that the query complexity of our testing algorithm has similar dependence on ℓ and $1/\epsilon$ but does not depend on n.

1.3. Organization.

We start with some necessary preliminaries in section 2. In section 3 we present our algorithm for testing singleton functions. The algorithm for testing monomials is presented in section 4, and the algorithm for testing monotone DNF in section 5. In section 6 we discuss a possible simpler alternative to the singleton test.

2. Preliminaries.

We shall use the following definitions.

DEFINITION 1. *Let $x, y \in \{0,1\}^n$, and let $[n] \stackrel{\text{def}}{=} \{1, \ldots, n\}$.*

- *We denote by $|x|$ the number of ones in the vector x.*
- *We write $y \succeq x$ if in each coordinate $y_i \geq x_i$.*
- *Let $2^x \stackrel{\text{def}}{=} \{z \in \{0,1\}^n : z \preceq x\}$. Hence, $|2^x| = 2^{|x|}$.*
- *Let $x \wedge y$ denote the string $z \in \{0,1\}^n$ such that for every $i \in [n]$, $z_i = x_i \wedge y_i$.*
- *Let $x \oplus y$ denote the string $z \in \{0,1\}^n$ such that for every $i \in [n]$, $z_i = x_i \oplus y_i$.*

[3] Applying the theorem known as Occam's Razor would give a stronger result in the sense that the underlying distribution may be arbitrary (that is, not necessarily uniform). This however comes at a price of a linear, as opposed to logarithmic, dependence of the sample/query complexity on n.

Definition 2 (singletons, monomials, and DNF functions). *A function $f : \{0,1\}^n \rightarrow \{0,1\}$ is a* singleton *function if there exists an $i \in [n]$ such that $f(x) = x_i$ for every $x \in \{0,1\}^n$ or $f(x) = \bar{x}_i$ for every $x \in \{0,1\}^n$.*

We say that f is a monotone k-monomial *for $1 \leq k \leq n$ if there exist k indices $i_1, \ldots, i_k \in [n]$ such that $f(x) = x_{i_1} \wedge \cdots \wedge x_{i_k}$ for every $x \in \{0,1\}^n$. If we allow some of the x_{i_j}'s above to be replaced with $\bar{x}_{i_j}$, then f is a* k-monomial. *The function f is a* monomial *if it is a k-monomial for some $1 \leq k \leq n$.*

A function f is an ℓ-term DNF *function if it is a disjunction of ℓ monomials. If all monomials are monotone, then it is a* monotone DNF *function.*

When the identity of the function f is clear from the context, we may use the following notation.

Definition 3. *Define $F_0 \overset{\text{def}}{=} \{x | f(x) = 0\}$ and $F_1 \overset{\text{def}}{=} \{x | f(x) = 1\}$.*

Definition 4 (distance between functions). *The distance according to the uniform distribution between two functions $f, g : \{0,1\}^n \rightarrow \{0,1\}$ is denoted by $\text{dist}(f,g)$ and is defined as follows: $\text{dist}(f,g) \overset{\text{def}}{=} \Pr_{x \in \{0,1\}^n}[f(x) \neq g(x)]$.*

The distance between a function f and a family of functions $\mathcal{F}$ is $\text{dist}(f,\mathcal{F}) \overset{\text{def}}{=} \min_{g \in \mathcal{F}} \text{dist}(f,g)$. If $\text{dist}(f,\mathcal{F}) > \epsilon$ for some $0 < \epsilon < 1$, then we say that f is ϵ-far *from $\mathcal{F}$. Otherwise, f is* ϵ-close *to $\mathcal{F}$.*

Definition 5 (testing algorithms). *A testing algorithm for a family of Boolean functions $\mathcal{F}$ over $\{0,1\}^n$ is given a distance parameter ϵ, $0 < \epsilon < 1$, and is provided with query access to an arbitrary function $f : \{0,1\}^n \rightarrow \{0,1\}$.*

If $f \in \mathcal{F}$, then the algorithm must output accept *with probability at least $2/3$, and if f is ϵ-far from $\mathcal{F}$, then it must output* reject *with probability at least $2/3$.*

3. Testing singletons. We start by presenting an algorithm for testing singletons. The testing algorithm for k-monomials will generalize this algorithm. More precisely, we present an algorithm for testing whether a function f is a *monotone* singleton. In order to test whether f is a singleton we can check whether either f or $\bar{f}$ pass the monotone singleton test. For the sake of succinctness, in what follows we refer to monotone singletons simply as singletons.

The following characterization of monotone k-monomials motivates our tests. We later show that the requirement of monotonicity can be removed.

Claim 1. *Let $f : \{0,1\}^n \rightarrow \{0,1\}$. Then f is a monotone k-monomial if and only if the following two conditions hold:*

1. $\Pr[f = 1] = 1/2^k$.
2. *For all x, y, $f(x \wedge y) = f(x) \wedge f(y)$.*

Proof. If f is a k-monomial, then clearly the conditions hold. We turn to prove the other direction. We first observe that the two conditions imply that $f(x) = 0$ for all $|x| < k$, where $|x|$ denotes the number of ones in x. In order to verify this, assume in contradiction that there exists some x such that $|x| < k$ but $f(x) = 1$. Now consider any y such that $y_i = 1$ whenever $x_i = 1$. Then $x \wedge y = x$, and therefore $f(x \wedge y) = 1$. By the second item, since $f(x) = 1$, it must also hold that $f(y) = 1$. However, since $|x| < k$, the number of such points y is strictly greater than 2^{n-k}, contradicting the first item.

Next let $y = \bigwedge_{x \in F_1} x$. Using the second item in the claim we get

$$f(y) = f\left(\bigwedge_{x \in F_1} x\right) = \bigwedge_{x \in F_1} f(x) = 1.$$

However, we have just shown that $f(x) = 0$ for all $|x| < k$, and thus $|y| \geq k$. Hence, there exist k indices $i_1, \ldots, i_k$ such that $y_{i_j} = 1$ for all $1 \leq j \leq k$. However, $y_{i_j} = \bigwedge_{x \in F_1} x_{i_j}$. Hence, $x_{i_1} = \cdots = x_{i_k} = 1$ for every $x \in F_1$. The first item now implies that $f(x) = x_{i_1} \wedge \cdots \wedge x_{i_k}$ for every $x \in \{0,1\}^n$. □

Definition 6. *We say that $x, y \in \{0,1\}^n$ are a* violating pair *with respect to a function $f : \{0,1\}^n \to \{0,1\}$ if $f(x) \wedge f(y) \neq f(x \wedge y)$.*

Given the above definition, Claim 1 states that a basic property of monotone singletons, and more generally of monotone k-monomials, is that there are no violating pairs with respect to f. A natural candidate for a testing algorithm for singletons would take a sample of uniformly selected pairs x, y and for each pair verify that it is not violating with respect to f. In addition, the test would check that $\Pr[f = 0]$ is roughly $1/2$ (or else any monotone k-monomial would pass the test).

As we discuss in section 6, we were unable to give a complete proof for the correctness of this test. Somewhat counterintuitively, the difficulty with the analysis lies in the case when the function f is *very far* from being a singleton. More precisely, the analysis is quite simple when the distance δ between f and the closest singleton is bounded away from $1/2$. However, the argument does not directly apply to δ arbitrarily close to $1/2$. We believe it would be interesting to prove that this simple test is in fact correct (or to come up with an example of a function f that is almost $1/2$-far from any singleton but passes the test).

In the algorithm described below we circumvent the above difficulty by "forcing more structure" on f. Specifically, we first perform another test that accepts only functions that have or, more precisely, that are close to having a certain structure. In particular, every singleton will pass the test. We then perform a slight variant of our original test. Provided that f passes the first test, it will be easy to show that f passes the second test with high probability only if it is close to a singleton function. Details follow.

The algorithm begins by testing whether the function f belongs to a larger family of functions that contains singletons as a subfamily. This is the family of *parity functions*.

Definition 7. *A function $f : \{0,1\}^n \to \{0,1\}$ is a* parity function *(a linear function over* GF(2)*) if there exists a subset $S \subseteq [n]$ such that $f(x) = \oplus_{i \in S} x_i$ for every $x \in \{0,1\}^n$.*

The test for parity functions is a special case of the linearity test over general fields due to Blum, Luby, and Rubinfeld [6]. If the tested function f is a parity function, then the test always accepts, and if f is ϵ-far from any parity function, then the test rejects with probability at least $9/10$. The query complexity of this test is $O(1/\epsilon)$. Specifically, the test uniformly picks $O(1/\epsilon)$ pairs $x, y \in \{0,1\}^n$ and checks that $f(x) \oplus f(y) = f(x \oplus y)$.

Assuming this test passes, we still need to verify that f is actually close to a singleton function and not to some other parity function. If the parity test accepted only proper parity functions, then the following claim would suffice. It shows that if f is a nonsingleton parity function, then a constant size sample of pairs x, y would, with high probability, contain a violating pair with respect to f.

Claim 2. *Let $g = \oplus_{i \in S} x_i$ for $S \subseteq [n]$. If $|S|$ is even, then*

$$\Pr[g(x \wedge y) = g(x) \wedge g(y)] = \frac{1}{2} + \frac{1}{2^{|S|+1}},$$

and if $|S|$ is odd, then

$$\Pr[g(x \wedge y) = g(x) \wedge g(y)] = \frac{1}{2} + \frac{1}{2^{|S|}}.$$

Proof. Let $s = |S|$, and let x, y be two strings such that (i) x has $0 \le i \le s$ ones in S, that is, $|\{\ell \in S : x_\ell = 1\}| = i$; (ii) $x \wedge y$ has $0 \le k \le i$ ones in S; and (iii) y has a total of $j + k$ ones in S, where $0 \le j \le s - i$.

If $g(x \wedge y) = g(x) \wedge g(y)$, then either (1) i is even and k is even, or (2) i is odd and j is even. Let $Z_1 \subset \{0,1\}^n \times \{0,1\}^n$ be the subset of pairs x, y that obey the first constraint, and let $Z_2 \subset \{0,1\}^n \times \{0,1\}^n$ be the subset of pairs x, y that obey the second constraint. Since the two subsets are disjoint,

$$\Pr[g(x \wedge y) = g(x) \wedge g(y)] = 2^{-2n} \cdot (|Z_1| + |Z_2|). \tag{3.1}$$

It remains to compute the sizes of the two sets. Since the coordinates of x and y outside S do not determine whether the pair x, y belongs to one of these sets, we have

$$|Z_1| = 2^{n-s} \cdot 2^{n-s} \cdot \left(\sum_{i=0, i \ even}^{s} \binom{s}{i} \sum_{k=0, k \ even}^{i} \binom{i}{k} \sum_{j=0}^{s-i} \binom{s-i}{j} \right) \tag{3.2}$$

and

$$|Z_2| = 2^{n-s} \cdot 2^{n-s} \cdot \left(\sum_{i=0, i \ odd}^{s} \binom{s}{i} \sum_{k=0}^{i} \binom{i}{k} \sum_{j=0, j \ even}^{s-i} \binom{s-i}{j} \right). \tag{3.3}$$

The first expression equals

$$2^{2n-2s} \cdot (2^{2s-2} + 2^{s-1}) = 2^{2n-2} + 2^{2n-s-1} = 2^{2n} \cdot (2^{-2} + 2^{-(s+1)}).$$

The second sum equals $2^{2n} \cdot (2^{-2} + 2^{-(s+1)})$ if s is odd and 2^{2n-2} if s is even. The claim follows by combining (3.2) and (3.3) with (3.1). □

Hence, if f is a parity function that is not a singleton, that is, $|S| \ge 2$, then the probability that a uniformly selected pair x, y is violating with respect to f is at least $1/8$. In this case, a sample of 16 such pairs will contain a violating pair with probability at least $1 - (1 - 1/8)^{16} \ge 1 - e^{-2} > 2/3$.

However, what if f passes the parity test but is only close to being a parity function? Let g denote the parity function that is closest to f, and let δ be the distance between them. (Note that g is unique, given that f is sufficiently close to a parity function.) What we would like to do is check whether g is a singleton, by selecting a sample of pairs x, y and checking whether it contains a violating pair with respect to g. Observe that, since the distance between functions is measured with respect to the uniform distribution, then for a uniformly selected pair x, y, with probability at least $(1 - \delta)^2$, both $f(x) = g(x)$ and $f(y) = g(y)$. However, we cannot make a similar claim about $f(x \wedge y)$ and $g(x \wedge y)$, since $x \wedge y$ is *not* uniformly distributed. Thus it is not clear that we can replace the violation test for g with a violation test for f. In addition, we would like to verify that g is not the all-0 function.

The solution is to use a *self-corrector* for linear (parity) functions [6]. Given query access to a function $f : \{0,1\}^n \to \{0,1\}$, which is strictly closer than $1/4$ to some parity function g, and an input $x \in \{0,1\}^n$, the procedure Self-Correct(f, x)

returns the value of $g(x)$, with probability at least $9/10$. The query complexity of the procedure is constant.

The above discussion suggests the following testing algorithm.

ALGORITHM 1. Test for singleton functions.

```
1. Apply the parity test to $f$ with distance parameter $\min(1/5, \epsilon)$. If the parity
   test rejects, then reject.
2. If Self-Correct($f, \vec{1}$) $= 0$, then reject (where $\vec{1}$ is the all-1 vector).
3. Uniformly and independently select $m = 64$ pairs of points $x, y$.
   • For each such pair, let $b_x =$ Self-Correct($f, x$), $b_y =$ Self-Correct($f, y$),
     and $b_{x \wedge y} =$ Self-Correct($f, x \wedge y$).
   • Check that $b_{x \wedge y} = b_x \wedge b_y$.
4. If one of the checks fails, then reject. Otherwise, accept.
```

THEOREM 1. *Algorithm 1 is a testing algorithm for monotone singletons. Furthermore, it has a one-sided error. That is, if f is a monotone singleton, the algorithm always accepts. The query complexity of the algorithm is $O(1/\epsilon)$.*

Proof. Since the testing algorithm for parity functions has a one-sided error, if f is a singleton function, then it always passes the test. In this case the self-corrector always returns the value of f on every given input point. In particular, Self-Correct$(f, \vec{1}) = f(\vec{1}) = 1$, since every monotone singleton has value 1 on the all-1 vector. Similarly, no violating pair can be found in Step 1. Hence, the test always accepts a singleton.

Assume, without loss of generality, that $\epsilon \leq 1/5$. Consider the case in which f is ϵ-far from any singleton. If it is also ϵ-far from any parity function, then it will be rejected with probability at least $9/10$ in the first step of the algorithm. Otherwise, there exists a unique parity function g such that f is ϵ-close to g. If g is the all-0 function, then f is rejected with probability at least $9/10$. Otherwise, g is a parity function of at least two variables. By Claim 2, the probability that a uniformly selected pair x, y is a violating pair with respect to g is at least $1/8$. Given such a pair, the probability that the self-corrector returns the value of g on all the three calls (that is, $b_x = g(x)$, $b_y = g(y)$, and $b_{x \wedge y} = g(x \wedge y)$), is at least $(1 - 1/10)^3 > 7/10$. The probability that Algorithm 1 obtains a violating pair with respect to g *and* all calls to the self-corrector return the correct value is greater than $1/16$. Therefore, a sample of 64 pairs will ensure that a violation $b_{x \wedge y} \neq b_x \wedge b_y$ will be found with probability at least $9/10$. The total probability that f is accepted, despite being ϵ-far from any singleton, is hence at most $3 \cdot (1/10) < 1/3$.

The query complexity of the algorithm is dominated by the query complexity of the parity tester which is $O(1/\epsilon)$. The second stage takes a constant time. □

4. Testing monomials. In this section we describe an algorithm for testing *monotone* k-monomials, where k is provided to the algorithm. We discuss later how to extend this to testing monomials when k is not specified. As for the monotonicity requirement, the following observation and corollary show that this requirement can be easily removed, if desired.

OBSERVATION 3. *Let $f : \{0,1\}^n \to \{0,1\}$, and let $z \in \{0,1\}^n$. Consider the function $f_z : \{0,1\}^n \to \{0,1\}$ that is defined by $f_z(x) = f(x \oplus z)$. Then the following are immediate:*

1. *The function f is a k-monomial if and only if f_z is a k-monomial.*
2. *Let $y \in F_1$. If f is a (not necessarily monotone) k-monomial, then $f_{\bar{y}}$ is a monotone k-monomial.*

COROLLARY 4. *If f is ϵ-far from every (not necessarily monotone) k-monomial, then for every $y \in F_1$, $f_{\bar{y}}$ is ϵ-far from every monotone k-monomial.*

We next observe that we can also assume, without loss of generality, that $\epsilon < 2^{-k+2}$, or else the testing problem is trivial.

OBSERVATION 5. *Suppose that $\epsilon \geq 2^{-k+2}$. Then,*

1. *if $\Pr[f = 1] \leq \frac{\epsilon}{2}$, then f is ϵ-close to every k-monomial and in particular to every monotone k-monomial;*
2. *if $\Pr[f = 1] > \frac{\epsilon}{4}$, then f is not a k-monomial.*

Proof. If $\Pr[f = 1] \leq \frac{\epsilon}{2}$, then for every k-monomial g,

$$\mathrm{dist}(f, g) = \Pr[f = 1 \wedge g = 0] + \Pr[f = 0 \wedge g = 1] \leq \Pr[f = 1] + \Pr[g = 1] \leq \frac{\epsilon}{2} + 2^{-k} \leq \epsilon.$$

Since $\epsilon \geq 2^{-k+2}$, if $\Pr[f = 1] > \frac{\epsilon}{4}$, then $\Pr[f = 1] > 2^{-k}$, while by the definition of a k-monomial, $\Pr[f = 1] = 2^{-k}$. □

By Observation 5, if the algorithm receives parameters ϵ and k such that $\epsilon \geq 2^{-k+2}$, then it simply needs to obtain an estimate α for $p = \Pr[f = 1]$ such that the following holds with probability of at least 2/3: if $p > \epsilon/2$, then $\alpha > 3\epsilon/8$, and if $p \leq \epsilon/4$, then $\alpha \leq 3\epsilon/8$. By a multiplicative Chernoff bound, such an estimate can be obtained using a sample of size $O(1/\epsilon)$. The algorithm accepts if $\alpha \leq 3\epsilon/8$ and rejects otherwise.

Hence, from this point on we can assume that $\epsilon < 2^{-k+2}$.

We now present the algorithm for testing monotone k-monomials. The first two steps of the algorithm are an attempt to generalize the parity test in Algorithm 1. Specifically, we test whether F_1 is an *affine subspace*.

DEFINITION 8 (affine subspaces). *A subset $H \subseteq \{0,1\}^n$ is an* affine subspace *of $\{0,1\}^n$ if and only if there exist an $x \in \{0,1\}^n$ and a linear subspace V of $\{0,1\}^n$ such that $H = V \oplus x$. That is,*

$$H = \{y \mid y = v \oplus x, for\ some\ v \in V\}.$$

The following is a well-known alternative characterization of affine subspaces, which is a basis for our test.[4]

FACT 6. *H is an affine subspace if and only if for every $y_1, y_2, y_3 \in H$ we have $y_1 \oplus y_2 \oplus y_3 \in H$.*

Note that the above fact implies that for every $y_1, y_2 \in H$ and $y_3 \notin H$ we have $y_1 \oplus y_2 \oplus y_3 \notin H$.

ALGORITHM 2. Test for monotone k-monomials.

1. Size Test: *Uniformly and independently select a sample of $\Theta(2^k)$ strings in $\{0,1\}^n$. For each x in the sample, obtain $f(x)$. Let α be the fraction of sample strings x such that $f(x) = 1$. If $|\alpha - 2^{-k}| > 2^{-(k+5)}$, then* reject*; otherwise, continue.*
2. Affinity Test:
   (a) *Set $\delta = 1/36$.*
   (b) *Uniformly and independently select $m = 2^5/(\epsilon \cdot \delta)$ points $a_1, \ldots, a_m \in \{0,1\}^n$.*
   (c) *Uniformly and independently select $m' = 4/\delta$ pairs of points $(x_1, y_1), \ldots, (x_{m'}, y_{m'}) \in F_1 \times F_1$.*

[4] Here and in most of what follows we use y_i (similarly, x_i) to denote strings in $\{0,1\}^n$, and not single bits. We find this notation easier to read than the alternative notation y^i. We use the latter only when necessary, that is, when we need to refer to particular coordinates y^i_j of the string.

(d) *If for some* $1 \le i \le m$, $1 \le j \le m'$, *the equality* $f(a_i \oplus x_j \oplus y_j) = f(a_i)$ *does not hold, then* reject.

As we show in our analysis, passing this step with sufficiently high probability ensures that f *is close to some function* g *for which* $g(x) \oplus g(y) \oplus g(z) = g(x \oplus y \oplus z)$ *for all* $x, y, z \in G_1 = \{x | g(x) = 1\}$. *That is,* G_1 *is an affine subspace.*

3. Closure-Under-Intersection Test:
   (a) *Uniformly and independently select* 32 *points* $x \in F_1$.
   (b) *Uniformly and independently select* 2^{k+3} *points* $y \in \{0,1\}^n$.
   (c) *If for some pair* x, y *selected, Self-Correct*$(f, x \wedge y) \neq$ *Self-Correct*(f, y), *then* reject. *Here Self-Correct is a procedure that given any input* z *and oracle access to* f *asks a constant number of queries and returns with high constant probability, the value* $g(z)$, *where* g *is as described in Step* 2.
4. *If no step caused rejection, then* accept.

In both the affinity test and the closure-under-intersection test, we need to select strings in F_1 uniformly. This is simply done by sampling from $\{0,1\}^n$ and using only x's for which $f(x) = 1$. Since in both tests the number of strings selected from F_1 is a constant, the total number of queries required is $O(2^k) = O(1/\epsilon)$. (Recall that we can assume that $\epsilon < 2^{-k+2}$.)

We now embark on proving the correctness of the algorithm.

THEOREM 2. *Algorithm* 2 *is a testing algorithm for monotone* k*-monomials. The query complexity of the algorithm is* $O(1/\epsilon)$.

The proof of Theorem 2 is based on the following two lemmas whose proofs are provided in subsections 4.1 and 4.2, respectively.

LEMMA 7. *Let* f *be a function for which* $|\Pr[f = 1] - 2^{-k}| < 2^{-k-3}$. *If the probability that the affinity test accepts* f *is greater than* $1/10$, *then there exists a function* $g : \{0,1\}^n \to \{0,1\}$ *for which the following holds:*

1. $\mathrm{dist}(f, g) \le \epsilon/2^5$.
2. $G_1 \stackrel{\text{def}}{=} \{a : g(a) = 1\}$ *is an affine subspace of dimension* $n - k$.
3. *There exists a procedure Self-Correct that given any input* $a \in \{0,1\}^n$ *and oracle access to* f *asks a constant number of queries and returns the value* $g(a)$ *with probability at least* $1 - 1/40$.

Furthermore, if F_1 *is an affine subspace, then the affinity tests always accepts,* $g = f$, *and Self-Correct*$(f, a) = f(a)$ *with probability* 1 *for every* $a \in \{0,1\}^n$.

LEMMA 8. *Let* $f : \{0,1\}^n \to \{0,1\}$ *be a function for which* $|\Pr[f = 1] - 2^{-k}| < 2^{-k-3}$. *Suppose that there exists a function* $g : \{0,1\}^n \to \{0,1\}$ *such that the following hold:*

1. $\mathrm{dist}(f, g) \le 2^{-k-3}$.
2. $G_1 \stackrel{\text{def}}{=} \{x : g(x) = 1\}$ *is an affine subspace of dimension* $n - k$.
3. *There exists a procedure Self-Correct that given any input* $a \in \{0,1\}^n$ *and oracle access to* f *returns the value* $g(a)$ *with probability at least* $1 - 1/40$.

If g *is not a monotone* k*-monomial, then the probability that the closure-under-intersection test rejects is at least* $9/10$.

Proof of Theorem 2. If f is a monotone k-monomial, then $\Pr[f = 1] = 2^{-k}$. By a multiplicative Chernoff bound, for the appropriate constant in the $\Theta(\cdot)$ notation, the probability that it is rejected in the first step of Algorithm 2 is less than a $1/3$. By the definition of k-monomials, f always passes the affinity test and the closure-under-intersection test.

Suppose that f is ϵ-far from any monotone k-monomial. We show that it is rejected with probability greater than $2/3$.

1. If $|\Pr[f=1]-2^{-k}| \geq 2^{-k-3}$, then by a multiplicative Chernoff bound f is rejected in the first step of the algorithm with probability at least $9/10$.
2. Otherwise, $|\Pr[f=1]-2^{-k}| < 2^{-k-3}$. If f is $(\epsilon/2^5)$-far from every function g such that $G_1 = \{x : \, g(x)=1\}$ is an affine subspace of dimension $n-k$, then by Lemma 7 it is rejected in the second step of the algorithm (the affinity test) with probability at least $9/10$.
3. Otherwise, both $|\Pr[f=1]-2^{-k}| < 2^{-k-3}$ and f is $(\epsilon/2^5)$-close to a function g as described in the previous item. Since f is assumed to be ϵ-far from any monotone k-monomial, the function g cannot be a monotone k-monomial. Since $\epsilon \leq 2^{-k+2}$ then $\epsilon/2^5 \leq 2^{-k-3}$ and therefore f is 2^{-k-3}-close to g. Hence, by Lemma 8, f will be rejected with probability at least $9/10$ in the third step of the algorithm (the closure-under-intersection test).

Summing up, we get that the probability that f is accepted by the algorithm is less than a $1/3$, as required. □

4.1. Analysis of the affinity test. In this subsection we prove Lemma 7. To this end we define the function g as follows:

$$(4.1) \quad g(a) \overset{\text{def}}{=} 1 \quad \text{if} \quad \Pr_{x,y\in F_1}[f(a\oplus x\oplus y)=1] \geq 1/2 \quad \text{and } g(a) \overset{\text{def}}{=} 0 \text{ otherwise.}$$

We shall prove two lemmas from which Lemma 7 follows.

LEMMA 9. *If the probability that the affinity test accepts f is greater than* $1/10$*, then* $\mathrm{dist}(f,g) \leq \epsilon/2^5$.

LEMMA 10. *If the probability that the affinity test accepts f is greater than* $1/10$*, then for every $a,b,c \in G_1$ we have $(a\oplus b\oplus c) \in G_1$.*

Proof of Lemma 9. By definition,

$$\mathrm{dist}(f,g) = 2^{-n} \cdot \left(|F_1 \setminus G_1| + |G_1 \setminus F_1|\right).$$

We observe that for any particular $a \in F_1 \setminus G_1$

$$\Pr_{x,y\in F_1}[f(a\oplus x\oplus y) \neq f(a)] = \Pr_{x,y\in F_1}[f(a\oplus x\oplus y)=0] > 1/2,$$

where the equality follows from $a \in F_1$, and the inequality from $a \notin G_1$. Similarly, for any particular $a \in G_1 \setminus F_1$,

$$\Pr_{x,y\in F_1}[f(a\oplus x\oplus y) \neq f(a)] = \Pr_{x,y\in F_1}[f(a\oplus x\oplus y)=1] \geq 1/2.$$

Assume, contrary to the claim, that $\mathrm{dist}(f,g) > \epsilon/2^5$. Then with probability at least $1-(1-\epsilon/2^5)^{2^5/(\epsilon\cdot\delta)} > 1-e^{-1/\delta}$, one of the a_i's selected in Step 2(b) of Algorithm 2 is in the symmetric difference $(F_1 \setminus G_1) \cup (G_1 \setminus F_1)$. For that a_i, with probability at least $1-(1-1/2)^{4/\delta} > 1-e^{-2/\delta}$, Algorithm 2 selects, in Step 2(c), a pair (x_j, y_j) such that $f(a_i) \neq f(a_i\oplus x_j\oplus y_j)$. Hence, the probability that f is rejected in this case is greater than $(1-e^{-1/\delta})(1-e^{-2/\delta}) > 9/10$ for $\delta \leq 1/5$. However, this contradicts the premise of the lemma by which f is accepted with probability greater than $1/10$. □

In order to prove Lemma 10 we introduce some notation and prove a few claims.

For any $a \in \{0,1\}^n$ let

$$(4.2) \qquad WP(a) \overset{\text{def}}{=} \{(x,y) \in F_1 \times F_1 : \, f(a\oplus x\oplus y) \neq f(a)\}$$

be the set of *witness pairs* (x, y) that together with a constitute evidence against the affinity of F_1. Let

$$H = \{a \in \{0,1\}^n : |WP(a)| > \delta \cdot |F_1|^2\} \tag{4.3}$$

be the set of *heavy* points $a \in \{0,1\}^n$ for which there are relatively many pairs $(x, y) \in F_1 \times F_1$ that together with a constitute evidence against the affinity of F_1.

The claim below follows from the definition of H and the test, similarly to Lemma 9, where we take into account that $\epsilon \le 2^{-k+2}$. (See Observation 5 and the discussion following it.)

CLAIM 11. *If the probability that the affinity test accepts f is greater than $1/10$, then $|H| \le \delta \cdot 2^{n-k-1}$.*

In all that follows we assume that the affinity test accepts f with probability greater than $1/10$. Since we assume that $|F_1| \ge 2^{n-k-1}$, it directly follows from Claim 11 that $|H| \le \delta \cdot |F_1|$.

By the definition of g we know that for every $a \in \{0,1\}^n$, $\Pr_{x,y\in F_1}[g(a) = f(a\oplus x\oplus y)] \ge 1/2$. In the claim below we show that this agreement probability is actually higher.

CLAIM 12. *For every $a \in \{0,1\}^n$, $\Pr_{x,y\in F_1}[g(a) = f(a\oplus x\oplus y)] \ge 1 - 4\delta$.*

Proof. We fix a and let $\gamma \stackrel{\text{def}}{=} \Pr_{x,y\in F_1}[g(a) = f(x\oplus y\oplus a)]$. By the definition of $g(\cdot)$, $\gamma \ge \frac{1}{2}$. We are interested in strengthening this bound. Consider the following equality and inequality for $\Pr_{x_1,x_2,y_1,y_2\in F_1}[f(a\oplus x_1\oplus y_1) = f(a\oplus x_2\oplus y_2)]$:

$$\begin{aligned}
&\Pr_{x_1,x_2,y_1,y_2\in F_1}[f(a\oplus x_1\oplus y_1) = f(a\oplus x_2\oplus y_2)]\\
&\quad = \Pr_{x_1,x_2,y_1,y_2\in F_1}[(f(a\oplus x_1\oplus y_1) = g(a)) \wedge (f(a\oplus x_2\oplus y_2) = g(a))]\\
&\qquad + \Pr_{x_1,x_2,y_1,y_2\in F_1}[f(a\oplus x_1\oplus y_1) \ne g(a)) \wedge (f(a\oplus x_2\oplus y_2) \ne g(a))]\\
&\quad = \gamma^2 + (1-\gamma)^2 \qquad (4.4)
\end{aligned}$$

and

$$\begin{aligned}
&\Pr_{x_1,x_2,y_1,y_2\in F_1}[f(a\oplus x_1\oplus y_1) = f(a\oplus x_2\oplus y_2)]\\
&\ge \Pr_{x_1,x_2,y_1,y_2\in F_1}[\, f(a\oplus x_1\oplus y_1) = f(a\oplus x_1\oplus x_2\oplus y_1\oplus y_2))\\
&\qquad\qquad \wedge (f(a\oplus x_2\oplus y_2) = f(a\oplus x_1\oplus x_2\oplus y_1\oplus y_2))\,]\\
&= 1 - \Pr_{x_1,x_2,y_1,y_2\in F_1}[\, (f(a\oplus x_1\oplus y_1) \ne f(a\oplus x_1\oplus x_2\oplus y_1\oplus y_2))\\
&\qquad\qquad \vee (f(a\oplus x_2\oplus y_2) \ne f(a\oplus x_1\oplus x_2\oplus y_1\oplus y_2))\,]\\
&\ge 1 - 2\cdot \Pr_{x_1,x_2,y_1,y_2\in F_1}[\, f(a\oplus x_1\oplus y_1) \ne f(a\oplus x_1\oplus x_2\oplus y_1\oplus y_2)\,]. \qquad (4.5)
\end{aligned}$$

Subclaim 12.1. $\Pr_{x_1,x_2,y_1,y_2\in F_1}[\,(f(a\oplus x_1\oplus y_1) \ne f(a\oplus x_1\oplus x_2\oplus y_1\oplus y_2))\,] \le 2\delta$.

Proof. We bound $\Pr_{x_1,x_2,y_1,y_2\in F_1}[\,(f(a\oplus x_1\oplus y_1) \ne f(a\oplus x_1\oplus x_2\oplus y_1\oplus y_2))\,]$ by the sum of two terms:

$$\begin{aligned}
&\Pr_{x_1,x_2,y_1,y_2\in F_1}[\, f(a\oplus x_1\oplus y_1) \ne f((a\oplus x_1\oplus y_1)\oplus x_2\oplus y_2)\,]\\
&\quad \le\ \Pr_{x_1,y_1\in F_1}[(a\oplus x_1\oplus y_1) \in H]\cdot \max_{a'\in H}\{\Pr_{x_2,y_2\in F_1}[f(a') \ne f(a'\oplus x_2\oplus y_2)]\}\\
&\quad + \Pr_{x_1,y_1\in F_1}[(a\oplus x_1\oplus y_1) \notin H]\cdot \max_{a'\notin H}\{\Pr_{x_2,y_2\in F_1}[f(a') \ne f(a'\oplus x_2\oplus y_2)]\}. \qquad (4.6)
\end{aligned}$$

By the definition of H and since $\Pr_{x_1,y_1\in F_1}[(a\oplus x_1\oplus y_1) \notin H] \le 1$, the second term in the above sum is bounded by δ. It remains to bound the first term by δ as well. We shall use the trivial bound of 1 for $\max_{a'\in H}\{\Pr_{x_2,y_2\in F_1}[f(a')\neq f(a'\oplus x_2\oplus y_2)]\}$ and show that $\Pr_{x_1,y_1\in F_1}[(a\oplus x_1\oplus y_1)\in H]\leq\delta$.

For each choice of $x_1\in F_1$, consider the set $B(a,x_1)\overset{\text{def}}{=}\{y_1\in F_1:\ a\oplus x_1\oplus y_1\in H\}$. Since for each pair $y_1,y_1'\in F_1$ such that $y_1\neq y_1'$ we have $a\oplus x_1\oplus y_1\neq a\oplus x_1\oplus y_1'$, then there is a one-to-one mapping from $B(a,x_1)$ into H. Therefore, $|B(a,x_1)|\leq|H|$ for every $x_1\in F_1$. It follows that

$$\begin{aligned}\Pr_{x_1,y_1\in F_1}[(a\oplus x_1\oplus y_1)\in H] &= \frac{1}{|F_1|^2}\sum_{x_1\in F_1}|B(a,x_1)|\\ &\leq \frac{|H|}{|F_1|}\quad\leq\quad\frac{\delta\cdot|F_1|}{|F_1|}\ =\ \delta, \qquad (4.7)\end{aligned}$$

where we have used Claim 11 in the last inequality. □

Subclaim 12.2. *Let $\frac{1}{2}\leq\gamma\leq 1$ and suppose that $\gamma^2+(1-\gamma)^2\geq 1-\beta$ for some $0<\beta<1$. Then $\gamma\geq 1-\beta$.*

Proof. If $\gamma^2+(1-\gamma)^2\geq 1-\beta$, then $2\gamma(1-\gamma)\leq\beta$. Since $\gamma\geq 1/2$, this implies that $1-\gamma\leq\beta$ or, equivalently, that $\gamma\geq 1-\beta$. □

By combining (4.4) and (4.5) with Subclaim 12.1, we obtain that $\gamma^2+(1-\gamma)^2\geq 1-4\delta$. Claim 12 follows by applying Subclaim 12.2. □

Recall that in order to prove Lemma 10 we need to show that for every $a,b,c\in G_1$, $g(a\oplus b\oplus c)=1$. That is, by the definition of g, we need to show that $\Pr_{x,y\in F_1}[f((a\oplus b\oplus c)\oplus x\oplus y)=1]\geq 1/2$ for every $a,b,c\in G_1$. To this end we first prove the following related claim.

CLAIM 13. *For every $a,b,c\in G_1$*

$$\Pr_{x_1,y_1,x_2,y_2,x_3,y_3\in F_1}[f((a\oplus x_1\oplus y_1)\oplus(b\oplus x_2\oplus y_2)\oplus(c\oplus x_3\oplus y_3))=1]\geq 1-14\delta.$$

Proof. We first observe that by the definition of $WP(\cdot)$ (see (4.2)) we have

$$\begin{aligned}&\Pr_{x_1,y_1,x_2,y_2,x_3,y_3\in F_1}[f((a\oplus x_1\oplus y_1)\oplus(b\oplus x_2\oplus y_2)\oplus(c\oplus x_3\oplus y_3))=1]\\ &\geq \Pr_{x_1,y_1,x_2,y_2,x_3,y_3\in F_1}[(a\oplus x_1\oplus y_1)\in F_1\setminus H\ \text{ and}\\ &\qquad\qquad ((b\oplus x_2\oplus y_2),(c\oplus x_3\oplus y_3))\in(F_1\times F_1)\setminus WP(a\oplus x_1\oplus y_1)]\\ &\geq \Pr_{x_1,y_1\in F_1}[(a\oplus x_1\oplus y_1)\in F_1\setminus H]\\ &\quad\times\min_{a'\in F_1\setminus H}\Pr_{x_2,y_2,x_3,y_3\in F_1}[((b\oplus x_2\oplus y_2),(c\oplus x_3\oplus y_3))\in(F_1\times F_1)\setminus WP(a')]. \qquad (4.8)\end{aligned}$$

By Claim 12 and since $a\in G_1$, we know that $\Pr_{x_1,y_1\in F_1}[(a\oplus x_1\oplus y_1)\in F_1]\geq 1-4\delta$. By (4.7) we know that $\Pr_{x_1,y_1\in F_1}[(a\oplus x_1\oplus y_1)\in H]\leq\delta$. Hence,

$$\Pr_{x_1,y_1\in F_1}[(a\oplus x_1\oplus y_1)\in F_1\setminus H]\geq 1-5\delta. \qquad (4.9)$$

It remains to bound the second term in the product above in (4.8). By Claim 12, and since $b,c\in G_1$, we know that $\Pr_{x_2,y_2\in F_1}[(b\oplus x_2\oplus y_2)\in F_1]\geq 1-4\delta$ and similarly that $\Pr_{x_3,y_3\in F_1}[(c\oplus x_3\oplus y_3\in F_1]\geq 1-4\delta$. Hence

$$\begin{aligned}\Pr_{x_2,y_2,x_3,y_3\in F_1}[((b\oplus x_2\oplus y_2),(c\oplus x_3\oplus y_3))\in(F_1\times F_1)] &\geq (1-4\delta)^2\\ &\geq 1-8\delta. \qquad (4.10)\end{aligned}$$

Let us fix some $a'\in F_1\setminus H$. By the definition of H we know that $|WP(a')|\leq\delta\cdot|F_1|^2$. For any fixed $x_2,x_3\in F_1$ let

$$B(a',b,c,x_2,x_3)\overset{\text{def}}{=}\{(y_2,y_3)\in F_1\times F_1:\ ((b\oplus x_2\oplus y_2),(c\oplus x_3\oplus y_3))\in WP(a')\}.$$

Observe that for every two different pairs $(y_2, y_3), (y_2', y_3') \in F_1 \times F_1$ (i.e., such that either $y_2 \neq y_2'$ or $y_3 \neq y_3'$) the pair $((b \oplus x_2 \oplus y_2), (c \oplus x_3 \oplus y_3))$ differs from $((b \oplus x_2 \oplus y_2'), (c \oplus x_3 \oplus y_3'))$. That is, there is a one-to-one mapping from $B(a', b, c, x_2, x_3)$ into $WP(a')$. It follows that for every $x_2, x_3 \in F_1$,

$$|B(a', b, c, x_2, x_3)| \leq |WP(a')| \leq \delta \cdot |F_1|^2$$

and so

$$\begin{aligned} \mathrm{Pr}_{x_2,y_2,x_3,y_3 \in F_1}[((b \oplus x_2 \oplus y_2), (c \oplus x_3 \oplus y_3)) \in WP(a')] &= \frac{1}{|F_1|^4} \sum_{x_2,x_3 \in F_1} |B(a', b, c, x_2, x_3)| \\ &\leq \frac{\delta \cdot |F_1|^2}{|F_1|^2} \quad = \delta. \end{aligned} \tag{4.11}$$

The claim follows by combining (4.8)–(4.11). □

We are now ready to prove Lemma 10.

Proof of Lemma 10. Consider any fixed choice of $a, b, c \in G_1$. If $a = b$, then $(a \oplus b \oplus c) = c$, so that $(a \oplus b \oplus c) \in G_1$ and the claim holds by definition, and similarly for the case $a = c$ or $b = c$. Hence we may assume from now on that all three points a, b, c are different.

Assume contrary to the claim that there exist $a, b, c \in G_1$ such that $g(a \oplus b \oplus c) \neq 1$. By the definition of g this means that $\mathrm{Pr}_{x,y \in F_1}[f((a \oplus b \oplus c) \oplus x \oplus y) = 1] < 1/2$. In other words, if we let

$$O(a, b, c) \stackrel{\text{def}}{=} \{(x, y) \in F_1 \times F_1 : \ f((a \oplus b \oplus c) \oplus x \oplus y) = 1\},$$

then $|O(a, b, c)| < |F_1|^2/2$. We shall show that this contradicts Claim 13.

For every fixed choice of $x_1, x_2, y_1, y_2 \in F_1$, let

$$O(x_1, x_2, y_1, y_2) \stackrel{\text{def}}{=} \{(x_3, y_3) \in F_1 \times F_1 : \ ((x_1 \oplus x_2 \oplus x_3), (y_1 \oplus y_2 \oplus y_3)) \in O(a, b, c)\}.$$

(In order to be consistent with previous notation, we should have let $O(x_1, x_2, y_1, y_2)$ be denoted by $O(a, b, c, x_1, x_2, y_1, y_2)$, but we have chosen the above notation in consideration of the reader.) Then similarly to what we have argued before for similar subsets, $|O(x_1, x_2, y_1, y_2)| \leq |O(a, b, c)|$. Hence,

$$\begin{aligned} &\mathrm{Pr}_{x_1,x_2,x_3,y_1,y_2,y_3 \in F_1}[((x_1 \oplus x_2 \oplus x_3), (y_1 \oplus y_2 \oplus y_3)) \in O(a, b, c)] \\ &= \frac{1}{|F_1|^6} \sum_{x_1,x_2,y_1,y_2 \in F_1} |O(x_1, x_2, y_1, y_2)| \\ &\leq \frac{|O(a, b, c)|}{|F_1|^2} \quad < 1/2. \end{aligned} \tag{4.12}$$

Recalling that $O(a, b, c) \subset F_1 \times F_1$, this implies that

$$\begin{aligned} &\mathrm{Pr}_{x_1,x_2,x_3,y_1,y_2,y_3 \in F_1}[f((a \oplus x_1 \oplus y_1) \oplus (b \oplus x_2 \oplus y_2) \oplus (c \oplus x_3 \oplus y_3)) = 1] \\ &\quad \leq \ \mathrm{Pr}_{x_1,x_2,x_3,y_1,y_2,y_3 \in F_1}[((x_1 \oplus x_2 \oplus x_3), (y_1 \oplus y_2 \oplus y_3)) \in O(a, b, c)] \\ &\qquad + \mathrm{Pr}_{x_1,x_2,x_3,y_1,y_2,y_3 \in F_1}[((x_1 \oplus x_2 \oplus x_3), (y_1 \oplus y_2 \oplus y_3)) \notin F_1 \times F_1]. \end{aligned} \tag{4.13}$$

By (4.12), the first term in the sum above is less than 1/2. We turn to bound the second term.

$$\begin{aligned}
&\mathrm{Pr}_{x_1,x_2,x_3,y_1,y_2,y_3\in F_1}[((x_1 \oplus x_2 \oplus x_3),(y_1 \oplus y_2 \oplus y_3)) \notin F_1 \times F_1] \\
&\quad\leq\quad 2\cdot \mathrm{Pr}_{x_1,x_2,x_3\in F_1}[(x_1 \oplus x_2 \oplus x_3) \notin F_1] \\
&\quad\leq\quad 2\cdot \mathrm{Pr}_{x_1\in F_1}[x_1 \in H] \\
&\qquad\quad + 2\cdot \mathrm{Pr}_{x_1\in F_1}[x_1 \notin H]\cdot \max_{x_1'\notin H} \mathrm{Pr}_{x_2,x_3\in F_1}[(x_1' \oplus x_2 \oplus x_3) \notin F_1] \\
&\quad\leq\quad 2\cdot(\delta+\delta),
\end{aligned} \tag{4.14}$$

where in the last inequality we have applied Claim 11 and the definition of H. Thus we get that

$$\mathrm{Pr}_{x_1,x_2,x_3,y_1,y_2,y_3\in F_1}[f((a \oplus x_1 \oplus y_1) \oplus (b \oplus x_2 \oplus y_2) \oplus (c \oplus x_3 \oplus y_3)) = 1] < (1/2) + 4\delta\ .$$

However, for $\delta \leq 1/36$ we get a contradiction to Claim 13. □

Proof of Lemma 7. Suppose that the affinity test accepts with probability greater than 1/10, and let g be as defined in (4.1). By Lemma 9 we have that $\mathrm{dist}(f,g) \leq \epsilon/2^5$ as required. By applying Lemma 10 we get that G_1 is an affine subspace. Furthermore, its dimension must be $n-k$ given the premise of the lemma concerning the size of F_1. The last item in the lemma follows from the definition of g: given any input a, the procedure Self-Correct simply selects a sufficiently large (but constant) number of pairs $x, y \in F_1$ and returns the majority value obtained for $f(a \oplus x \oplus y)$. By Claim 12 it returns the correct value $g(a)$ with high probability.

If F_1 is an affine subspace, then by Fact 6 the test always accepts f, and, by definition of g in (4.1), $g = f$. Finally, the procedure Self-Correct as defined above always returns $f(a)$ for every $a \in \{0,1\}^n$. □

4.2. Analysis of the closure-under-intersection test.

In this subsection we prove Lemma 8.

4.2.1. Properties of affine subspaces.

We first recall several simple properties of affine subspaces.

Claim 14. *Let H be an affine subspace such that $H = V \oplus x$, where $x \in \{0,1\}^n$ and $V \subseteq \{0,1\}^n$ is a linear subspace. Then, we have the following:*

1. *$x \in H$.*
2. *For every $z \in H$ we have that $H = V \oplus z$. By the definition of the $\oplus$ operator, we thus also have that $V = H \oplus z$ for every $z \in H$.*
3. *$|H| = |V| = 2^{dimV}$.*

Claim 15. *Let H, H' be two affine subspaces of $\{0,1\}^n$ such that $H \not\subseteq H'$. Then*

$$\frac{|H \cap H'|}{|H|} \leq \frac{1}{2}.$$

Proof. The claim follows from the corresponding property of linear subspaces, namely, $V \not\subseteq V'$ implies that $|V \cap V'|/|V| \leq 1/2$. □

The following corollary is immediate.

Corollary 16. *Let H, H' be two affine subspaces of $\{0,1\}^n$ such that $H' \subseteq H$. Then either $H' = H$ or $|H'| \leq |H|/2$.*

Claim 17. *Let H, H' be two affine subspaces of $\{0,1\}^n$ such that $H' \subseteq H$, and let $y \in H'$. Denote by V' the linear subspace such that $H' = V' \oplus y$, and by V the linear subspace such that $H = V \oplus y$. Then we have the following:*

1. *$V' \subseteq V$.*
2. *For any $x \in V$ we have $(H' \oplus x) \subseteq H$, and for any $x \notin V$ we have $(H' \oplus x) \cap H = \emptyset$.*

Proof. By definition, $V' = H' \oplus y \subseteq H \oplus y = V$. This proves the first part of the lemma.

Now let $x \in V$. Since V and V' are linear subspaces and $V' \subseteq V$, then $(V' \oplus x) \subseteq V$. Thus, $H' \oplus x = (V' \oplus y) \oplus x = (V' \oplus x) \oplus y \subseteq V \oplus y = H$. On the other hand, let $x \notin V$. Observe that $(V' \oplus x) \cap V = \emptyset$. Since $H' \oplus x = (V' \oplus y) \oplus x = (V' \oplus x) \oplus y$, we get that $(H' \oplus x) \cap H = (V' \oplus x) \oplus y \cap (V \oplus y) = \emptyset$. This concludes the proof of the claim. □

4.2.2. Auxiliary claims. In order to prove Lemma 8 we will need several auxiliary claims. The first claim relates affine spaces that correspond to k-monomials and monotonicity.

CLAIM 18. *Let H be an affine subspace of $\{0,1\}^n$ of size 2^{n-k}. Assume also that H is monotone. Namely, if $x \in H$ and $y \succeq x$, then $y \in H$. Then $H = \{x : x_{i_1} = 1, \dots, x_{i_k} = 1\}$ for some subset $i_1, \dots, i_k$ of coordinates.*

Proof. Let V be an $n-k$ dimensional linear subspace, and let $y \in \{0,1\}^n$ be such that $H = V \oplus y$. Let $v_1, \dots, v_{n-k}$ be a basis of V. Consider an $(n-k) \times n$ matrix with rows $v_1, \dots, v_{n-k}$. Its rank is $n-k$, and therefore it has $n-k$ linearly independent columns. Without loss of generality, these are the first $n-k$ columns. Therefore the restriction of the rows to the first $n-k$ coordinates is a basis of $\{0,1\}^{n-k}$, and thus it spans all the vectors in $\{0,1\}^{n-k}$ and in particular the first $n-k$ coordinates of y. It follows that there is a vector $v \in V$, namely a linear combination of the rows, that is equal to y on the first $n-k$ coordinates. Therefore, $z = (v \oplus y) \in H$ is 0 on the first $n-k$ coordinates.

Since H is monotone, if $|z| < k$, or there exists a $z' \not\succeq z$ such that $z' \in H$, then $|H| > 2^{n-k}$, contradicting our assumption on H. Hence $H = \{x : x_{i_1} = 1, \dots, x_{i_k} = 1\}$, where $i_1, \dots, i_k$ are the coordinates on which z is 1. □

Recall that by the premise of Lemma 8 there exists a function g such that $\mathrm{dist}(f,g) \le 2^{-k-3}$, and $G_1 \overset{\text{def}}{=} \{x : g(x) = 1\}$ is an affine subspace of dimension $n-k$. Claim 18 implies that if g is not a k-monomial, then the affine subspace G_1 cannot be monotone. We shall use this, together with the fact that f and g are close, to prove that there are many pairs $x \in F_1$, $y \in \{0,1\}^n$ such that $f(y) \ne f(x \wedge y)$. To this end we define the following subsets.

DEFINITION 9. *Let $x \in \{0,1\}^n$ and $z \in 2^x$. Define $G(x,z) \overset{\text{def}}{=} \{y |\ x \wedge y = z\}$.*

We shall show that for many pairs (x,z), with $x \in G_1$ and $z \in 2^x$, the function g is far from constant on $G(x,z)$. Since the functions f and g are close to each other, this will imply the existence of many violating pairs, as desired. First, we prove some properties of the subsets $G(x,z)$.

CLAIM 19. *For every $x \in \{0,1\}^n$ and $z \in 2^x$, $G(x,z)$ is an affine subspace of $\{0,1\}^n$ of size $2^{n-|x|}$. Furthermore, for every $x \in \{0,1\}^n$, the affine subspaces $\{G(x,z)\}_{z \in 2^x}$ partition $\{0,1\}^n$.*

Proof. These facts about $G(x,z)$ follow easily from the following observation: for a fixed x, the map $m_x : y \to x \wedge y$ is a linear map from $\{0,1\}^n$ to 2^x, and $G(x,z) = m_x^{-1}(z)$. □

CLAIM 20. *Let $x \in G_1$ be such that there exists $z \in 2^x$ for which $G(x,z) \subseteq G_1$. Then $G(x,x) \subseteq G_1$.*

Proof. We first show that $G(x,z) \oplus x \oplus z \subseteq G_1$. Since G_1 is an affine subspace, by Fact 6, it is enough to show that x and z lie in G_1 and that $G(x,z)$ is a subset of G_1. Taking into account the assumptions of the claim, we need only to show that $z \in G_1$. Since $z \preceq x$, we have $z \wedge x = z$. Hence, $z \in G(x,z) \subseteq G_1$.

Next, we show that $G(x,x) \subseteq G(x,z) \oplus x \oplus z$. Take $y \in G(x,x)$. Now define y' as follows. If $z_i = 1$, then $y'_i = 1$ (in this case always $x_i = 1$). If $z_i = 0$ and $x_i = 1$, then $y'_i = 0$, and if $z_i = 0$ and $x_i = 0$, then $y'_i = y_i$. Thus, $y' \wedge x = z$ and so $y' \in G(x,z)$. It is also easy to verify that $y' \oplus x \oplus z = y$. (Note that $y \succeq x$, and therefore $x_i = 1$ implies that $y_i = 1$.) Hence, $y \in G(x,z) \oplus x \oplus z$. Since we have shown that $G(x,z) \oplus x \oplus z \subseteq G_1$, the claim follows. □

We shall be interested in the following set:

$$\mathcal{X} \stackrel{\text{def}}{=} \{x \in G_1 : \ G(x,x) \subseteq G_1\}. \tag{4.15}$$

Thus $\mathcal{X}$ consists of those $x \in G_1$ for which every $y \succeq x$ is in G_1. Since, by Claim 18, the set G_1 is not monotone, then necessarily $\mathcal{X} \neq G_1$. As we show momentarily, $\mathcal{X}$ is actually significantly smaller than G_1, and we shall exploit this in our proof.

CLAIM 21. *The set $\mathcal{X}$ is an affine subspace of G_1. Furthermore, if g is not a k-monomial, then $|\mathcal{X}| \leq \frac{1}{2}|G_1|$.*

Proof. By Fact 6, in order to prove the first part of the lemma it suffices to show that for every $x^1, x^2, x^3 \in \mathcal{X}$, we have $x^1 \oplus x^2 \oplus x^3 \in \mathcal{X}$. Let us fix $x^1, x^2, x^3 \in \mathcal{X}$, and let $x = x^1 \oplus x^2 \oplus x^3$. To show that $x \in \mathcal{X}$ we have to show that $G(x,x) \subseteq G_1$, namely, that for every $y \succeq x$ we have $y \in G_1$. Let $y \succeq x$. Then there exist y^1, y^2, y^3 such that $y = y^1 \oplus y^2 \oplus y^3$, where $y^j \succeq x^j$ for $j = 1, \ldots, 3$. (To verify this, choose a coordinate i: (1) If $y_i = x_i$, set $y^j_i = x^j_i$ for all j. (2) If $y_i = 1$ and $x_i = 0$, set $y^j_i = 1$ for all j.) That is, $y^j \in G(x^j, x^j) \subseteq G_1$. Therefore $y^j \in G_1$ for all j, and so $y = y^1 \oplus y^2 \oplus y^3 \in G_1$.

By Corollary 16, since $\mathcal{X}$ is an affine subspace of G_1, either $\mathcal{X} = G_1$ or $|\mathcal{X}| \leq \frac{1}{2}|G_1|$. If $\mathcal{X} = G_1$, then for any $x \in G_1$ we have $G(x,x) = \{y : \ y \succeq x\} \subseteq G_1$, namely, G_1 is monotone. By Claim 18, g is a k-monomial, which contradicts our assumptions. Therefore, $|\mathcal{X}| \leq \frac{1}{2}|G_1|$. □

In the next claim we show that for every $x \in G_1 \setminus \mathcal{X}$, the function g is far from constant on $G(x,z)$ for many $z \in 2^x$. Observe that this is trivially true if g is a monotone monomial, since in this case the set $G_1 \setminus \mathcal{X}$ is empty.

CLAIM 22. *For every $x \in G_1 \setminus \mathcal{X}$, and for any fixed function $h : \{0,1\}^n \to \{0,1\}$,*

$$\frac{1}{2^{|x|}} \cdot \sum_{z \in 2^x} \Pr_{y \in G(x,z)}[g(y) \neq h(z)] \geq 2^{-k} .$$

Proof. Let us fix $x \in G_1 \setminus \mathcal{X}$ and a function h. For every $z \in 2^x$, if $h(z) = 0$, then

$$\Pr_{y \in G(x,z)}[g(y) \neq h(z)] = \frac{|G(x,z) \cap G_1|}{|G(x,z)|},$$

and if $h(z) = 1$, then

$$\Pr_{y \in G(x,z)}[g(y) \neq h(z)] = \frac{|G(x,z) \setminus G_1|}{|G(x,z)|} = 1 - \frac{|G(x,z) \cap G_1|}{|G(x,z)|}.$$

Hence,

$$\Pr_{y \in G(x,z)}[g(y) \neq h(z)] \geq \min\left\{\frac{|G(x,z) \cap G_1|}{|G(x,z)|}, 1 - \frac{|G(x,z) \cap G_1|}{|G(x,z)|}\right\}. \tag{4.16}$$

However, for all $z \in 2^x$, $G(x,z) \not\subseteq G_1$. (Otherwise, by Claim 20, we would have $G(x,x) \subseteq G_1$, and so $x \in \mathcal{X}$.) Thus, by Claim 15, $\frac{|G(x,z) \cap G_1|}{|G(x,z)|} \leq \frac{1}{2}$. Combining this with (4.16), we get

$$\frac{1}{2^{|x|}} \cdot \sum_{z \in 2^x} \Pr_{y \in G(x,z)}[g(y) \neq h(z)] \geq \frac{1}{2^{|x|}} \cdot \sum_{z \in 2^x} \frac{|G(x,z) \cap G_1|}{|G(x,z)|}$$

$$= 2^{-n} \cdot \sum_{z \in 2^x} |G(x,z) \cap G_1| \;=\; 2^{-n} \cdot |G_1| \;=\; 2^{-k}. \tag{4.17}$$

In the last sequence of steps we have used the following: (1) $|G(x,z)| = 2^{n-|x|}$ for every z (Claim 19); (2) For every x, the subsets $G(x,z)$ form a partition of $\{0,1\}^n$ (Claim 19); (3) G_1 is of size 2^{n-k}. □

4.2.3. Proof of Lemma 8. Let g be the function ensured by the premise of Lemma 8. If g is not a k-monomial, then by Claim 21 we know that $|\mathcal{X}| \leq |G_1|/2$. In other words,

$$|G_1 \setminus \mathcal{X}| \;\;\geq\;\; \frac{1}{2}|G_1| \;\;\geq\;\; 2^{n-k-1},$$

where the second inequality follows from the fact that G_1 is an affine subspace of dimension $n-k$. Since $\mathrm{dist}(f,g) \leq 2^{-k-3}$, where $|\;|F_1| - 2^{n-k}\;| \leq 2^{n-k-3}$, we know that

$$|(F_1 \cap G_1) \setminus \mathcal{X}| \geq 2^{n-k-2}.$$

Hence, with probability of at least $1-(1-1/4)^{32} > 1-e^{-8}$, Algorithm 2 obtains in Step 3(a) a point $x \in (F_1 \cap G_1) \setminus \mathcal{X}$. Let us denote this point by x^1.

Since $x^1 \in G_1 \setminus \mathcal{X}$ we know by Claim 22 that

$$\frac{1}{2^{|x^1|}} \cdot \sum_{z \in 2^{x^1}} \Pr_{y \in G(x^1,z)}[g(y) \neq g(z)] \geq 2^{-k}\;. \tag{4.18}$$

By Claim 19, $|G(x^1,z)| = 2^{n-|x^1|}$, and so we have that

$$\frac{1}{2^{|x^1|}} \cdot \sum_{z \in 2^{x^1}} \Pr_{y \in G(x^1,z)}[g(y) \neq g(z)] = \frac{1}{2^{|x^1|}} \sum_{z \in 2^{x^1}} \frac{1}{2^{n-|x^1|}} \left|\{y \in G(x^1,z) :\; g(y) \neq g(z)\}\right|$$

$$= \frac{1}{2^n} \left|\{y \in \{0,1\}^n :\; g(y) \neq g(x^1 \wedge y)\}\right|$$

$$= \Pr_{y \in \{0,1\}^n}[g(y) \neq g(x^1 \wedge y)]. \tag{4.19}$$

By combining (4.18) and (4.19) we have that

$$\Pr_{y \in \{0,1\}^n}[g(y) \neq g(x^1 \wedge y)] \geq 2^{-k}\;. \tag{4.20}$$

Hence, with probability of at least $1-(1-2^{-k})^{2^{k+3}} > 1-e^{-8}$, Algorithm 2 will select, in Step 3(b), such a point $y^1 \in \{0,1\}^n$ for which $g(y^1) \neq g(x^1 \wedge y^1)$. Finally, if both calls to Self-Correct, in Step 3(c), return correct values, which occurs with probability of at least $1-1/20$, then the algorithm will reject as desired. The lemma follows by combining all error probabilities. □

4.3. Testing monomials when k is unspecified. Suppose that we want to test whether a function f is a monomial without the size of the monomial, k, being specified. In this case we start by finding k. We obtain an estimate α to $\Pr[f = 1]$ by taking a sample of size $\Theta(1/\epsilon)$. By a multiplicative Chernoff bound, such a sample ensures that, with high probability, if $\Pr[f = 1] \geq \epsilon/2$, then $\alpha \geq \epsilon/4$, while if $\Pr[f = 1] < \epsilon/8$, then $\alpha < \epsilon/4$. Hence, if $\alpha < \epsilon/4$, then we can immediately accept. This is true, since we may assume that $\Pr[f = 1] < \epsilon/2$, and so f is close to every monomial that contains at least $\log(2/\epsilon)$ literals.

Otherwise, we may assume that $\Pr[f = 1] \geq \epsilon/8$, and a multiplicative Chernoff bound implies that, with high probability, $(1-1/4)\cdot\Pr[f = 1] < \alpha < (1+1/4)\cdot\Pr[f = 1]$. Now we look for an integer k for which $4/5\alpha \leq 2^{-k} \leq 4/3\alpha$. If there is no such integer, we reject. If there is, there is at most one, and we choose it as our estimate for k. If f is in fact a monomial, then this estimate of k is correct with high probability. Given this k, we proceed as before.

5. Testing monotone DNF formulae. In this section we describe an algorithm for testing whether a function f is a monotone DNF formula with at most ℓ terms for a given integer ℓ.

In other words, we test whether $f = T_1 \vee T_2 \vee \cdots \vee T_{\ell'}$, where $\ell' \leq \ell$, and each term T_i is a monotone monomial. Note that we allow the size of the terms to vary. We assume, without loss of generality, that no term contains the set of variables of any other term (or else we can ignore the more specific term), though the same variable can of course appear in several terms.

The basic idea underlying the algorithm is to test whether the set $F_1 \stackrel{\text{def}}{=} \{x : f(x) = 1\}$ can be "approximately covered" by at most ℓ terms (monomials). To this end, the algorithm finds strings $x_i \in \{0,1\}^n$ and uses them to define functions f_i that are tested for being monomials. If the original function f is in fact an ℓ-term DNF, then, with high probability, each such function f_i corresponds to one of the terms of f.

The following notation will be useful. Let f be a monotone ℓ-term DNF, and let its terms be $T_1, \ldots, T_\ell$. Then, for any $x \in \{0,1\}^n$, we let $S(x) \subseteq \{1, \ldots, \ell\}$ denote the subset of indices of the terms satisfied by x. That is,

$$S(x) \stackrel{\text{def}}{=} \{i : \ T_i(x) = 1\} \ .$$

In particular, if $f(x) = 0$, then $S(x) = \emptyset$. This notion extends to a set $R \subseteq F_1$, where $S(R) \stackrel{\text{def}}{=} \bigcup_{x \in R} S(x)$. We observe that if f is a monotone ℓ-term DNF, then for every $x, y \in \{0,1\}^n$

$$S(x \wedge y) = S(x) \cap S(y).$$

We shall also need the following definitions.

Definition 10 (single-term representatives). *Let f be a monotone ℓ-term DNF. We say that $x \in F_1$ is a* single-term representative *for f if $|S(x)| = 1$. That is, x satisfies only a single term in f.*

Definition 11 (neighbors). *Let $x \in F_1$. The* set of neighbors of *x, denoted by $N(x)$, is defined as follows:*

$$N(x) \stackrel{\text{def}}{=} \{y \mid f(y) = 1 \textit{ and } f(x \wedge y) = 1\}.$$

The notion of neighbors extends to a set $R \subseteq F_1$, where $N(R) \stackrel{\text{def}}{=} \bigcup_{x \in R} N(x)$.

Note that the above definition of neighbors is very different from the standard notion (that is, strings at Hamming distance 1), and in particular depends on the function f.

Consider the case in which x is a single-term representative of f, and $S(x) = \{i\}$. Then, for every neighbor $y \in N(x)$, we must have $i \in S(y)$ (or else $S(x \wedge y)$ would be empty, implying that $f(x \wedge y) = 0$). Notice that the converse statement holds as well; that is, $i \in S(y)$ implies that x and y are neighbors. Therefore, the set of neighbors of x is exactly the set of all strings satisfying the term T_i. The goal of the algorithm will be to find at most ℓ such single-term representatives $x \in \{0,1\}^n$, and for each such x to test that its set of neighbors $N(x)$ satisfies some common term. We shall show that if f is in fact a monotone ℓ-term DNF, then all these tests pass with high probability. On the other hand, if all the tests pass with high probability, then f is close to some monotone ℓ-term DNF.

We start with a high-level description of the algorithm and then show how to implement its main step of finding single-term representatives.

Algorithm 3. Test for monotone ℓ-term DNF.

1. $R \leftarrow \emptyset$. *R is designated to be a set of single-term representatives for f.*
2. *For $i = 1$ to $\ell + 1$ (try to add ℓ single-term representatives to R):*
   (a) *Take a uniform sample U_i of size $m_1 = \Theta(\ell \log \ell / \epsilon)$ strings. Let $W_i = (U_i \cap F_1) \setminus N(R)$. That is, W_i consists of strings x in the sample such that $f(x) = 1$, and x is not a neighbor of any string already in R.*
   Observe that if the strings in R are in fact single-term representatives, then every $x \in W_i$ satisfies only terms not satisfied by the representatives in R.
   (b) *If $i = \ell + 1$ and $W_i \neq \emptyset$, then* reject.
   If there are more than ℓ single term representatives for f, then necessarily f is not an ℓ-term DNF.
   (c) *Else, if $\frac{|W_i|}{m_1} < \frac{\epsilon}{4}$ then* go to Step 3.
   The current set of representatives already "covers" almost all of F_1.
   (d) *Else ($\frac{|W_i|}{m_1} \geq \frac{\epsilon}{4}$ and $i \leq \ell$), use W_i in order to find a string x_i that is designated to be a single-term representative of a term not yet represented in R. This step will be described subsequently.*
3. *For each string $x_i \in R$, let the function $f_i : \{0,1\}^n \mapsto \{0,1\}$ be defined as follows: $f_i(y) = 1$ if and only if $y \in N(x_i)$.*
   As observed previously, if x_i is in fact a single-term representative, then f_i is a monomial.
4. *For each f_i, test that it is monomial, using distance parameter $\epsilon' = \frac{\epsilon}{2\ell}$ and confidence $1 - \frac{1}{6\ell}$ (instead of $\frac{2}{3}$—this can simply be done by $O(\log \ell)$ repeated applications of each test).*
   Note that we do not specify the size of the monomial, and so we need to apply the appropriate variant of our test, as described in subsection 4.3.
5. *If any of the tests fail then* reject, *otherwise* accept.

The heart of the algorithm lies in finding a new representative in each iteration of Step 2(d). This procedure will be described and analyzed shortly. In particular, we shall prove the following lemma.

Lemma 23. *Suppose that f is a monotone ℓ-term DNF, and let $R \subset \{0,1\}^n$ be a subset of single-term representatives for f such that $\Pr[x \in (F_1 \setminus N(R))] \geq \epsilon/8$. Let U_i be a uniformly selected sample of $m_1 = \Theta(\ell \log \ell / \epsilon)$ strings, and let $W_i = (U_i \cap F_1) \setminus N(R)$. Then there exists a procedure that receives W_i as input, for which the following holds:*

1. *With probability at least $1 - \frac{1}{6\ell}$, taken over the choice of U_i and the internal coin flips of the procedure, the procedure returns a string x_i that is a single-term representative for f of a term not yet represented in R. That is, $|S(x_i)| = 1$ and $S(x_i) \cap S(R) = \emptyset$.*
2. *The query complexity of the procedure is $O(\ell \log^2 \ell/\epsilon)$.*

Conditioned on the above lemma we can prove the following theorem.

THEOREM 3. *Algorithm 3 is a testing algorithm for monotone ℓ-term DNF. The query complexity of the algorithm is $\tilde{O}(\ell^2/\epsilon)$.*

Proof. We shall use the following notation: for any set $R \subset \{0,1\}^n$, let

$$\bar{p}(R) \stackrel{\text{def}}{=} \Pr[x \in (F_1 \setminus N(R))].$$

Suppose that f is a monotone ℓ-term DNF, and consider each iteration of Step 2. By Lemma 23, if all strings in R are single-term representatives for f and $\bar{p}(R) \geq \epsilon/8$, then with probability of at least $1 - \frac{1}{6\ell}$ the procedure for finding a single-term representative in fact returns a new representative (of a term not yet represented in R). Hence, the probability that, for some iteration i, the string x_i returned by the procedure is not a single-term representative is at most $1/6$. Conditioned on such an event not occurring, Algorithm 3 completes Step 2 with a set R that contains at most ℓ single-term representatives for f.

In such a case, by the definition of single-term representatives, each f_i defined in Step 3 of Algorithm 3 is a monotone monomial. For each fixed f_i, the probability that it fails the monomial test is at most $\frac{1}{6\ell}$. By applying a union bound, the probability that any one of the f_i's fail is at most $\frac{1}{6}$. Adding up the error probabilities, we obtain that f is accepted with probability of at least $2/3$.

We now turn to the case in which f is ϵ-far from being a monotone ℓ-term DNF. Consider the value of $\bar{p}(R)$ at the start of each iteration i of Step 2. Observe that $\bar{p}(R)$ does not increase with i. If $\bar{p}(R) > \epsilon/2$, then, by a multiplicative Chernoff bound, the probability that $\frac{|W_i|}{m_1} \leq \epsilon/4$ (causing Algorithm 3 to exit Step 2) is smaller than $\frac{1}{6\ell}$. Hence, the probability that Algorithm 3 completes Step 2 without rejecting and with a set R for which $\bar{p}(R) > \epsilon/2$ is at most $1/6$.

Conditioned on such an event not occurring, consider the functions f_i defined in Step 3 of Algorithm 3. We claim that at least one of these functions is $\frac{\epsilon}{2\ell}$-far from being a monomial. To verify this, assume in contradiction that all these $|R| \leq \ell$ functions are $\frac{\epsilon}{2\ell}$-close to being monomials. For each such function f_i, let g_i be a closest monomial to f_i, and let $g = g_1 \vee g_2 \vee \cdots \vee g_{|R|}$. Then $\text{dist}(f, g) \leq |R| \cdot \frac{\epsilon}{2\ell} + \bar{p}(R) \leq \epsilon$, contradicting the fact that f is ϵ-far from any ℓ-term DNF. Thus, let f_t be one of the f_i's that is $\frac{\epsilon}{2\ell}$-far from being a monomial. The probability that the monomial test does not reject f_t is at most $\frac{1}{6\ell}$. Adding up the error probabilities, f is rejected with probability of at least $2/3$.

Finally, we bound the query complexity of Algorithm 3. There are at most $\ell + 1$ iterations in Step 2 of the algorithm. In each iteration, $m_1 = O(\ell \log \ell/\epsilon)$ strings are queried in Step 2(a). By Lemma 23, $O(\ell \log^2 \ell/\epsilon)$ strings are queried by the procedure for finding a new representative that is called in Step 2(d). By Theorem 2, testing that each of the at most ℓ functions f_i is a monomial requires a total of $\ell \cdot O(1/\epsilon') \cdot O(\log \ell) = \tilde{O}(\ell^2/\epsilon)$ queries. Therefore, the total number of queries performed by Algorithm 3 is $\tilde{O}(\ell^2/\epsilon)$. □

5.1. Finding new representatives. Suppose that f is a monotone ℓ-term DNF with terms $T_1, \ldots, T_\ell$, and consider an arbitrary iteration i in Step 2 of Algorithm 3.

Assume that $R \subset \{0,1\}^n$ is a subset of single-term representatives for f, such that $\Pr[x \in (F_1 \setminus N(R))] \geq \epsilon/8$. Let $\overline{N}(R) \overset{\text{def}}{=} F_1 \setminus N(R)$ be the set of all the strings that are not neighbors of any string in R, and let $\overline{S}(R) \overset{\text{def}}{=} \{1,\ldots,\ell\} \setminus S(R)$ be the set of indices of terms not yet represented in R. By definition, $W_i \subseteq \overline{N}(R)$, and for every $x \in W_i$ we have $S(x) \subseteq \overline{S}(R)$.

Given a string $x_0 \in W_i$, we shall try to "remove" terms from $S(x_0)$ until we are left with a single term. More precisely, we produce a sequence of strings $x_0,\ldots,x_r$, where $x_0 \in W_i$, such that $\emptyset \neq S(x_{j+1}) \subseteq S(x_j)$, and in particular $|S(x_r)| = 1$. The aim is to decrease the size of $S(x_j)$ by a constant factor for most j's. This will ensure that for $r = \Theta(\log \ell)$ the final string x_r is a single-term representative as desired.

How is such a sequence obtained? Given a string $y_j \in N(x_j)$, define $x_{j+1} = x_j \wedge y_j$. Then $f(x_{j+1}) = 1$ (i.e., $S(x_{j+1}) \neq \emptyset$), and $S(x_{j+1}) = S(x_j) \cap S(y_j) \subseteq S(x_j)$. The string y_j is acquired by uniformly selecting a sufficiently large sample from $\{0,1\}^n$, and picking the first string in the sample that belongs to $N(x_j)$, if one exists. The exact procedure follows.

Procedure for finding a new representative, given $W_i \subseteq \overline{N}(R)$.

```
1. Let the strings in $W_i$ be denoted $w_1,\ldots,w_{|W_i|}$.
2. Uniformly and independently select $r = \Theta(\log \ell)$ samples, $Y_0,\ldots,Y_{r-1}$, each
   consisting of $m_2 = O(\ell \log \ell/\epsilon)$ strings from $\{0,1\}^n$.
3. $found \leftarrow FALSE$; $t \leftarrow 0$;
4. While $found \neq TRUE$ and $t < |W_i|$ do:
   (a) $t \leftarrow t+1$; $x_0 \leftarrow w_t$.
   (b) For $j = 1$ to $r$
       (i) If $Y_{j-1} \cap N(x_{j-1}) = \emptyset$, then exit the "for" loop and go to Step 4(a).
       (ii) Otherwise, pick the first string $y_{j-1} \in Y_{j-1} \cap N(x_{j-1})$, and let
            $x_j = x_{j-1} \wedge y_{j-1}$.
   (c) If $j = r$, then $found \leftarrow TRUE$.
5. If $found = TRUE$, then return $x_r$. Otherwise, return an arbitrary string.
```

We first prove that if Y_j intersects $N(x_j)$, then the probability that the size of $S(x_{j+1})$ is significantly smaller than that of $S(x_j)$ is at least $1/3$. Observe that since the sample Y_j is uniformly distributed in $\{0,1\}^n$, then $Y_j \cap N(x_j)$ is uniformly distributed in $N(x_j)$.

CLAIM 24. *Let x_j be a fixed string. With probability of at least $1/3$ over the uniform choice of a string $y_j \in N(x_j)$, $|S(x_j \wedge y_j)| \leq 1 + \frac{3}{4} \cdot (|S(x_j)| - 1)$.*

Proof. Without loss of generality, let $S(x_j) = \{1,\ldots,t\}$. We partition the set of neighbors $N(x_j)$ into disjoint subsets $N_i(x_j)$, for $1 \leq i \leq t$, where

$$N_i(x_j) = \{y \in N(x_j) : \ i \in S(y) \text{ and for every } i' < i,\ i' \notin S(y)\}.$$

Since y_j is uniformly distributed in $N(x_j)$, we can view it as being selected by first choosing i with probability $\frac{|N_i(x_j)|}{|N(x_j)|}$ and then selecting y uniformly in $N_i(x_j)$.

Consider the case $y_j \in N_1(x_j)$. In order to select a string uniformly in $N_1(x_j)$, we first set to 1 all the bits corresponding to the variables in T_1 and then set the remaining bits to 0 or 1 with equal probability. Since for every $i \neq 1$ there is at least one variable that appears in T_i and not in T_1, we have that

$$\Pr\left[T_i(y_j) = 0 \mid y_j \in N_1(x_j)\right] \geq \frac{1}{2}.$$

It follows that the expected number of indices $i \in S(x_j)$, $i \neq 1$, for which $T_i(y_j) = 1$ is at most $(t-1)/2$. By Markov's inequality, the probability that there are more than

$(1-\alpha)(t-1)$ terms T_i, $i \neq 1$, satisfied by a uniformly selected $y_j \in N_1(x_j)$ is at most $\frac{1}{2(1-\alpha)}$. Setting $\alpha = 1/4$, we get that, with probability of at least $\frac{1}{3}$, over the choice of a uniformly selected $y_j \in N_1(x_j)$, we have that $|S(x_{j+1})| \leq 1 + \frac{3}{4} \cdot (|S(x_j)| - 1)$. It is easy to see that for any $N_i(x_j)$, $i > 1$, this probability is at least as large. In particular, note that for $i = t$, for any $y_j \in N_t(x_j)$, $|S(x_{j+1})| = 1$. □

The next corollary follows directly from Claim 24 and the fact that $|S(x_0)| \leq \ell$.

COROLLARY 25. *Let $r = c \cdot \log \ell$, where c is a sufficiently large constant, and let x_0 be a fixed string in W_i. Consider the following process, consisting of r steps, where in the j's step we uniformly and independently select a string $y_{j-1} \in N(x_{j-1})$ and set $x_j = x_{j-1} \wedge y_{j-1}$. Then, with probability of at least $1 - \frac{1}{18\ell}$ over the choice of $y_0, \ldots, y_{r-1}$, we obtain $|S(x_r)| = 1$.*

Finally, we bound the size of a sample Y_j sufficient for acquiring a string $y_j \in N(x_j)$ with high probability. We first define a "good initial string" x_0. This is a string that satisfies only relatively "large" terms.

DEFINITION 12. *A string x_0 will be called a* good initial string *if for every $i \in S(x_0)$, $\Pr[T_i = 1] \geq \frac{\epsilon}{16\ell}$.*

Recall that $\overline{N}(R) = F_1 \setminus N(R)$ and define the set

$$\text{Good} \stackrel{\text{def}}{=} \left\{ x \in \overline{N}(R) \mid x \text{ is a good initial string} \right\}.$$

CLAIM 26. *Suppose that $\Pr[x \in \overline{N}(R)] \geq \frac{\epsilon}{8}$. Then the probability, taken over the choices of U_i, that W_i does not contain any good initial strings is at most $\frac{1}{18\ell}$.*

Proof. Recall that $\bar{p}(R) \stackrel{\text{def}}{=} \Pr[x \in \overline{N}(R)]$ and that $\overline{S}(R) \stackrel{\text{def}}{=} \{1, \ldots, \ell\} \setminus S(R)$. For any $i \in \overline{S}(R)$, consider the event

$$E_i \stackrel{\text{def}}{=} \left\{ x \in \overline{N}(R) \text{ and } T_i(x) = 1 \right\}.$$

By definition, $\bar{p}(R) = \Pr[\bigcup_{i \in \overline{S}(R)} E_i]$. Let

$$\overline{S}_{\text{small}}(R) = \left\{ i \in \overline{S}(R) \text{ and } \Pr[E_i] \leq \frac{\bar{p}(R)}{2\ell} \right\}.$$

Clearly, for any term i, $\Pr[T_i = 1] \geq \Pr[E_i]$. Therefore, if $x \in (\bigcup_{i \in \overline{S}(R)} E_i) \setminus (\bigcup_{i \in \overline{S}_{\text{small}}(R)} E_i)$, then $S(x) \subseteq \overline{S}(R) \setminus \overline{S}_{\text{small}}(R)$, and therefore for all $i \in S(x)$ we have that $\Pr[T_i(x) = 1] \geq \Pr[E_i] \geq \frac{\bar{p}(R)}{2\ell} \geq \frac{\epsilon}{16\ell}$. Thus, $x \in \text{Good}$. Therefore,

$$\begin{aligned} \Pr[\text{Good}] &\geq \Pr\left[\left(\bigcup_{i \in \overline{S}(R)} E_i \right) \setminus \left(\bigcup_{i \in \overline{S}_{\text{small}}(R)} E_i \right) \right] \\ &\geq \Pr\left[\bigcup_{i \in \overline{S}(R)} E_i \right] - \Pr\left[\bigcup_{i \in \overline{S}_{\text{small}}(R)} E_i \right] \\ &\geq \bar{p}(R) - \ell \cdot \frac{\bar{p}(R)}{2\ell} \quad = \quad \frac{\bar{p}(R)}{2}. \end{aligned}$$

Since $\bar{p}(R) \geq \frac{\epsilon}{8}$ and the size of the sample U_i is $\Theta(\ell \log \ell / \epsilon)$, then the probability that W_i does not contain *any* good initial string is, for a sufficiently large constant in the $\Theta(\cdot)$ notation, smaller than $\frac{1}{18\ell}$. □

The next claim follows from the definition of a good initial string.

Claim 27. *Let $m_2 = c_1 \cdot \ell \log \ell/\epsilon$ and $r = c_2 \cdot \log \ell$, where c_1, c_2 are sufficiently large constants. Let $Y_1, \ldots, Y_r$ be samples of m_2 strings each, and suppose that x_0 is a good initial string. Then, for each $1 \leq j \leq r$, the probability that $Y_j \cap N(x_j) \neq \emptyset$ is at least $1 - \frac{1}{18\ell^2}$.*

We can now complete the proof of Lemma 23, and hence the correctness of Theorem 3.

Proof of Lemma 23. By the premise of the lemma, $\Pr[x \in \overline{N}(R)] \geq \epsilon/8$. By Claim 26, the set W_i contains a good initial string with probability at least $1 - \frac{1}{18\ell}$. Conditioned on this event, let us fix such a string x_0, and consider the execution of Step 4(a) in the procedure. By Claim 27, the probability that there exists a $j \leq r$ for which the sample Y_j does not contain a string in $N(x_j)$ is at most $\frac{1}{18\ell}$. Since the strings in Y_j are uniformly selected from $\{0,1\}^n$, the strings in $Y_j \cap N(x_j)$ are uniformly distributed in $N(x_j)$. Hence, conditioned on each Y_j containing a string from $N(x_j)$, we can apply Corollary 25 and get that with probability of at least $1-\frac{1}{18\ell}$, $|S(x_r)| = 1$. Since $x_0 \in \overline{N}(R)$, necessarily $x_r \in \overline{N}(R)$. Therefore, with probability of at least $1 - 3 \cdot \frac{1}{18\ell} = 1 - \frac{1}{6\ell}$, taken over the choices of U_i and the samples Y_j, the procedure returns a string x_r that is a single-term representative for f of a term not yet represented in f.

The number of queries performed by the procedure is $r \cdot m_2 = O(\ell \log^2 \ell/\epsilon)$ as promised. □

6. Testing singletons without testing linearity. Recall that by Claim 1 an alternative characterization of singletons is that $\Pr[f = 1] = 1/2$, and furthermore that there are no violating pairs $x, y \in \{0,1\}^n$. That is, there are no x, y such that $f(x \wedge y) \neq f(x) \wedge f(y)$. We show that the following simple algorithm that checks these properties is a testing algorithm for singletons if f is not too far from a singleton function. Let $\mathcal{F}_{sing}$ denote the class of singletons. The algorithm will receive a value γ_0 such that $\min_{g \in \mathcal{F}_{sing}} \text{dist}(f, g) \leq \frac{1}{2} - \gamma_0$. That is, γ_0 is a lower bound on the difference between $1/2$ and the distance of f to the closest singleton. We shall think of γ_0 as a constant.

Algorithm 4. Test for singletons with lower bound γ_0.

1. Size Test: *Uniformly select a sample of $m = \Theta(1/\epsilon^2)$ strings in $\{0,1\}^n$. For each x in the sample, obtain $f(x)$. Let α be the fraction of sample strings x such that $f(x) = 1$. If $|\alpha - 1/2| > \epsilon/4$, then* reject*; otherwise, continue.*
2. Closure-Under-Intersection Test: *Repeat the following $\Theta(\epsilon^{-1}\gamma_0^{-1})$ times: Uniformly select $x, y \in \{0,1\}^n$. If x and y are a violating pair, then reject.*
3. *If no step caused rejection, then* accept.

Theorem 4. *If f is a singleton, then Algorithm 4 accepts with probability of at least 2/3. If f is ϵ-far from any singleton where ϵ is bounded away from 1/2, then the algorithm rejects with probability of at least 2/3. The query complexity of the algorithm is $O(1/\epsilon^2)$.*

Proof. If f is a singleton, then $\Pr[f = 1] = 1/2$. By an additive Chernoff bound, and for the appropriate constant in the $\Theta(\cdot)$ notation, the probability that it is rejected in the first step of Algorithm 4 is less than 1/3. By the definition of singletons, f always passes the closure-under-intersection test.

Suppose that f is ϵ-far from any singleton, and let δ be its distance to the closest singleton. Thus $\epsilon < \delta \leq 1/2 - \gamma_0$. We show that f is rejected with probability greater than 2/3.

1. If $|\Pr[f = 1] - 1/2| > \frac{\epsilon}{2}$, then f is rejected in the first step of Algorithm 4 with probability of at least 5/6.

2. Otherwise, $|\Pr[f = 1] - 1/2| \leq \frac{\epsilon}{2} < \frac{\delta}{2}$. In this case, as we show shortly in Lemma 28, the probability of obtaining a violating pair is at least $\frac{\delta}{4}(\frac{1}{2} - \delta) \geq \frac{\epsilon}{4} \cdot \gamma_0$. Therefore, f will be rejected with probability of at least 5/6 in the second step of the algorithm (the closure-under-intersection test).

Thus, the probability that f is accepted by the algorithm is at most a 1/3, as required. □

LEMMA 28. *Let δ be the distance of f to the closest singleton. If* $\Pr[f(x) = 1] \geq \frac{1}{2} - \frac{\delta}{2}$, *then the probability of obtaining a violating pair is at least* $\frac{\delta}{4}(\frac{1}{2} - \delta)$.

Proof. Let x_i be the closest singleton to f, so that $\Pr[f(x) \neq x_i] = \delta$. Define

$$G_1 = \{x | f(x) = 1, x_i = 1\}, \quad B_1 = F_1 \setminus G_1,$$

$$G_0 = \{x | f(x) = 0, x_i = 0\}, \quad B_0 = F_0 \setminus G_0.$$

A simple counting argument shows that there are $(\frac{1}{2} - \delta)2^n$ disjoint pairs x, x', such that (1) $x \in G_1$, $x' \in G_0$; (2) x and x' differ only on the ith bit. To see why this is true, simply match each $x \in G_1$ to a point x', which differs with x only on the ith bit. Thus, there are at least $|G_1| - |B_1|$ points $x \in G_1$ that must be matched to points $x' \in G_0$. However, $|G_1| + |B_0| = 2^{n-1}$, and $|B_1| + |B_0| = \delta 2^n$, and therefore $|G_1| - |B_1| = (\frac{1}{2} - \delta)2^n$.

Now consider any point $y \in B_1$, and let $x \in G_1$, $x' \in G_0$ be a matched pair as defined above. Then $x \wedge y = x' \wedge y$, but $f(x) \wedge f(y) = 1$ while $f(x') \wedge f(y) = 0$. Therefore, either $f(x \wedge y) \neq f(x) \wedge f(y)$ or $f(x' \wedge y) \neq f(x') \wedge f(y)$, and so either y and x are a violating pair, or y and x' are a violating pair.

Since $\Pr[f(x) = 1] \geq \frac{1}{2} - \frac{\delta}{2}$, then $|G_1| + |B_1| \geq 2^n(\frac{1}{2} - \frac{\delta}{2})$. Using again the fact that $|G_1| - |B_1| = (\frac{1}{2} - \delta)2^n$, we get that $|B_1| \geq \delta 2^{n-2}$. It follows that the probability of obtaining a violating pair is at least $\frac{\delta}{4}(\frac{1}{2} - \delta)$. □

The above analysis breaks when f is actually almost $1/2 - far$ from every singleton, since in this case δ is close to 1/2, and the probability $\frac{\delta}{4}(\frac{1}{2} - \delta)$ of obtaining a violating pair is not bounded from below. Another disadvantage of Algorithm 4 is the two-sided error probability for testing singletons, as opposed to the one-sided error we achieved in Algorithm 1 when we added the parity test.

Algorithm 4 can be generalized to testing k-monomials, with a query complexity of $O(1/\epsilon^2)$. The probability of choosing a violating pair can be shown to be at least $\frac{\delta}{4}(2^{-k} - \delta)$. Thus the requirement here is that δ will be strictly smaller than 2^{-k}. Notice that requiring that $\delta = O(1/2^k)$ is not really a restriction: every function f for which $\Pr[f(x) = 1]$ is approximately 2^{-k} is $O(2^{-k})$-close to being a monomial, and our algorithm first verifies that in fact $\Pr[f(x) = 1]$ is approximately 2^{-k}. The restriction is in requiring that it be *strictly* smaller than 2^{-k}.

Another alternative test for singletons is to replace the relatively expensive test of checking whether $\Pr[f(x) = 1]$ is approximately 1/2, by extending the notion of a violating pair. We will say that $x, y \in \{0,1\}^n$ are a violating pair if $f(x \wedge y) \neq f(x) \wedge f(y)$ or if $f(x \vee y) \neq f(x) \vee f(y)$. Then in a similar way to the proof of Lemma 28, it can be shown that the probability of obtaining a violating pair is at least $\frac{\delta}{2}(\frac{1}{2} - \delta)$. (In this case the size of either B_0 or B_1 is at least $\delta 2^{n-1}$. Therefore choosing $y \in B_1$ or $y \in B_0$ and x, x' as before will result in a violating pair either to the $\wedge$ test or to the $\vee$ test.) The query complexity of this algorithm will be only $O(1/\epsilon)$, and it will have a one-sided error. Unfortunately, this algorithm does not extend to testing monomials.

Further research. Our results raise several questions that we believe may be interesting to study.

- Our algorithms for testing singletons and, more generally, monomials, apply two tests. The role of the first test is essentially to facilitate the analysis of the second, natural test (the closure-under-intersection test). The question is whether the first test is necessary.
- The query complexity of our algorithm for testing ℓ-term monotone DNF has a quadratic dependence in ℓ. While some dependence on ℓ seems necessary, we conjecture that a lower dependence is achievable. In particular, suppose we slightly relax the requirements of the testing algorithm and ask only that it rejects functions that are ϵ-far from any monotone DNF with at most $c \cdot \ell$ (or possibly ℓ^c) terms for some constant c. Is it possible, under this relaxation, to devise an algorithm that has only polylogarithmic dependence on ℓ?
- Finally, can our algorithm for testing monotone DNF functions be extended to testing general DNF functions?

Acknowledgments. We would like to thank Luca Trevisan for bringing to our attention the relation between testing the long code and testing singletons. We would also like to thank Uri Feige and Adi Shamir for useful comments which led to an improvement in the complexity of the monomial test.

REFERENCES

[1] N. Alon, E. Fischer, M. Krivelevich, and M. Szegedy, *Efficient testing of large graphs*, in Proceedings of the 40th Annual Symposium on Foundations of Computer Science, New York, NY, 1999, pp. 645–655.

[2] D. Angluin, *Queries and concept learning*, Machine Learning, 2 (1988), pp. 319–342.

[3] Y. Aumann, J. Håstad, M. Rabin, and M. Sudan, *Linear consistency testing*, in Proceedings of the 3rd International Workshop on Randomization and Approximation Techniques in Computer Science, Berkeley, CA, 1999, Springer-Verlag, Berlin, 1999, pp. 109–120.

[4] M. Bellare, D. Coppersmith, J. Håstad, M. Kiwi, and M. Sudan, *Linearity testing in characteristic two*, in Proceedings of the 36th Annual Symposium on Foundations of Computer Science, Milwaukee, WI, 1995, IEEE Computer Society Press, 1995, pp. 432–441.

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

[6] M. Blum, M. Luby, and R. Rubinfeld, *Self-testing/correcting with applications to numerical problems*, J. Comput. System. Sci., 47 (1993), pp. 549–595.

[7] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, *Occam's razor*, Inform. Process. Lett., 24 (1987), pp. 377–380.

[8] N. Bshouty, J. Jackson, and C. Tamon, *More efficient PAC-learning of DNF with membership queries under the uniform distribution*, in Proceedings of the Twelfth Annual Conference on Computational Learning, Santa Cruz, CA, 1999, pp. 286–295.

[9] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky, *Improved testing algorithms for monotonocity*, in Proceedings of the 3rd International Workshop on Randomization and Approximation Techniques in Computer Science, Berkeley, CA, 1999, Springer-Verlag, Berlin, 1999, pp. 97–108.

[10] E. Fischer, *The art of uninformed decisions: A primer to property testing*, Bull. Eur. Assoc. Theor. Comput. Sci. EATCS, 75 (2001), pp. 97–126.

[11] E. Fischer, G. Kindler, D. Ron, S. Safra, and A. Samorodnitsky, *Testing juntas*, in Proceedings of the 43rd Annual Symposium on Foundations of Computer Science, Vancouver, BC, Canada, 2002, to appear.

[12] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, and A. Samorodnitsky, *Testing monotonicity*, Combinatorica, 20 (2000), pp. 301–337.

[13] O. Goldreich, S. Goldwasser, and D. Ron, *Property testing and its connection to learning and approximation*, J. ACM, 45 (1998), pp. 653–750.

[14] O. Goldreich and D. Ron, *Property testing in bounded degree graphs*, in Algorithmica, 32 (2002), pp. 302–343.

[15] J. Jackson, *An efficient membership-query algorithm for learning DNF with respect to the uniform distribution*, J. Comput. System Sci., 55 (1997), pp. 414–440.

[16] A. Klivans and R. Servedio, *Boosting and hard-core sets*, in Proceedings of the 40th Annual Symposium on Foundations of Computer Science, New York, NY, 1999, pp. 624–633.

[17] M. Parnas, D. Ron, and A. Samorodnitsky, *Proclaiming dictators and juntas or testing boolean formulae*, in Proceedings of the 5th International Workshop on Randomization and Approximation Techniques in Computer Science, Berkeley, CA, 2001, pp. 273–284.

[18] M. Parnas, D. Ron, and A. Samorodnitsky, *Testing Boolean Formulae*, http://www.eccc.uni-trier.de/eccc/, TR01-063, 2001.

[19] D. Ron, *Property testing*, in Handbook of Randomized Computing, Volume II, S. Rajasekaran, P. Pardalos, J. Reif, and J. Rolim, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001, pp. 597–649.

[20] R. Rubinfeld, *On the robustness of functional equations*, SIAM J. Comput., 28 (1999), pp. 1972–1997.

[21] R. Rubinfeld and M. Sudan, *Robust characterization of polynomials with applications to program testing*, SIAM J. Comput., 25 (1996), pp. 252–271.

SIAM J. DISCRETE MATH.
Vol. 16, No. 1, pp. 47–64

ENUMERATION OF MATCHINGS IN THE INCIDENCE GRAPHS OF COMPLETE AND COMPLETE BIPARTITE GRAPHS*

NICHOLAS PIPPENGER†

Abstract. If $G = (V, E)$ is a graph, the *incidence graph* $I(G)$ is the graph with vertices $V \cup E$ and an edge joining $v \in V$ and $e \in E$ when and only when v is incident with e in G. For G equal to K_n (the complete graph on n vertices) or $K_{n,n}$ (the complete bipartite graph on $n + n$ vertices), we enumerate the matchings (sets of edges, no two having a vertex in common) in $I(G)$, both exactly (in terms of generating functions) and asymptotically. We also enumerate the equivalence classes of matchings (where two matchings are considered equivalent if there is an automorphism of G that induces an automorphism of $I(G)$ that takes one to the other).

Key words. generating functions, asymptotic enumeration

AMS subject classifications. 05A15, 05A16, 05C05

PII. S0895480101395695

1. Introduction.

One goal of this paper is the enumeration of matchings in the incidence graphs of certain graphs. There are of course many standard combinatorial results that can be interpreted as counting matchings in a graph. Indeed, for the graphs we consider, the method of inclusion-exclusion yields a summation from which the asymptotic behavior can be obtained by elementary means. We shall also be interested, however, in enumerating equivalence classes of matchings (where two matchings are considered equivalent if there is an automorphism of the underlying graph that induces an automorphism of the incidence graph that takes one matching into the other). For this problem, these standard methods do not serve, and we have had to adopt a different strategy, using Pólya's theory of enumeration [P2, P3] to derive generating functions, and in the bipartite case an analytic method for diagonalizing a bivariate power series introduced by Pippenger [P1]. This new strategy, however, works only for certain highly symmetric graphs. For reasons we will explain later, we are particularly interested in the incidence graphs of complete graphs and of complete bipartite graphs.

We shall denote by K_n the complete graph on n vertices, and by $K_{n,m}$ the complete bipartite graph on $n + m$ vertices. If $G = (V, E)$ is a graph, the *incidence graph* $I(G)$ is the graph with edges $V \cup E$ and an edge joining $v \in V$ and $e \in E$ when and only when v is incident with e in G. If G is a graph, we shall denote by $M(G)$ the set of matchings in G. (These matchings need not be maximum, or even maximal. Thus $M(G)$ is never empty, since G always has an empty matching. All of the enumerations we present can easily be extended to enumerate matchings by cardinality, simply by inserting an additional indeterminate into the generating functions.)

In section 2 we shall enumerate the matchings $M\big(I(K_n)\big)$ in the incidence graph $I(K_n)$ of the complete graph K_n. We first do this by inclusion-exclusion, then (as background to what will follow) in a more cumbersome way by Pólya's method.

In section 3 we shall enumerate the equivalence classes $\tilde{M}\big(I(K_n)\big)$ of matchings

*Received by the editors September 28, 2001; accepted for publication (in revised form) June 20, 2002; published electronically November 6, 2002. This work was supported by an NSERC Research Grant and a Canada Research Chair.

http://www.siam.org/journals/sidma/16-1/39569.html

†Department of Computer Science, The University of British Columbia, 2366 Main Mall, Vancouver, BC V6T 1Z4, Canada (nicholas@cs.ubc.ca).

in the incidence graph $I(K_n)$ of the complete graph K_n. Here, only Pólya's method is applicable. The result is closely related to the enumeration of "functional digraphs" by Harary [H] and Read [R].

In section 4 we shall enumerate the matchings $M\big(I(K_{n,n})\big)$ in the incidence graph $I(K_{n,n})$ of the complete bipartite graph $K_{n,n}$. Again we use both inclusion-exclusion and Pólya's method. To obtain the asymptotic behavior from the generating function, we use the method of Pippenger [P1].

In section 5 we shall enumerate the equivalence classes $\tilde{M}\big(I(K_{n,n})\big)$ of matchings in the incidence graph $I(K_{n,n})$ of the complete bipartite graph $K_{n,n}$. This result requires combining almost all the techniques introduced in earlier sections.

Most of the methods used in this paper were also used by Pippenger [P1], and many of the calculations done here are along lines similar to ones in that paper. Accordingly, we shall give fewer details for such calculations, referring the reader to that paper when appropriate.

The problems considered in this paper originally arose from the study of "concentrators" for communication switching (see Beneš [B1, B2]). Here, the vertices of $I(G)$ representing edges of G model "clients," while those representing vertices of G model "servers." A "state" of the system, in which some clients are connected in a one-to-one fashion to some servers, then corresponds to a matching in $I(G)$. Enumeration of the matchings thus gives information about the amount of storage required to keep track of the state of the system, while enumeration of the equivalence classes of matchings gives information about the number of essentially different situations that must be considered in formulating a control policy for the system. The 12 elements of $\tilde{M}\big(I(K_4)\big)$ are listed by Beneš [B1, B2].

2. Enumerating $M\big(I(K_n)\big)$.

Let A_n denote the cardinality of $M\big(I(K_n)\big)$.

THEOREM 2.1. *We have*

$$A_n = \sum_{j\geq 0} \frac{(n)_{2j}}{2^j\, j!}(-1)^j n^{n-2j}.$$

(Here $(n)_k = n(n-1)\cdots(n-k+1)$.)

Proof. Consider a matching $X \in M\big(I(K_n)\big)$. For each edge $\{e,v\} \in X$ (where $e \in E$ is an edge of $K_n = (V,E)$ and $v \in V$ is a vertex incident with e), we shall direct the edge $e = \{v,w\}$ out of v and into w. In this way we direct some of the edges of K_n. These directed edges form the graph of a map $\sigma : D \to V$ from a subset D of V to V. Furthermore, this map does not have any fixed points ($\sigma(v) = v$) or exchanged pairs of points ($\sigma(v) = w$ and $\sigma(w) = v$). Conversely, every map $\sigma : D \to V$ with $D \subseteq V$ having no fixed points or exchanged pairs arises in this way from unique matching in $M\big(I(K_n)\big)$.

The number of maps from a subset of V to V is $(n+1)^n$. We can count the number of these having no fixed points or exchanged pairs by using the principle of inclusion-exclusion. There are n possible fixed points, and the fraction of maps having k of them is $\binom{n}{k}(n+1)^{-k}$. There are $\binom{n}{2}$ possible exchanged pairs, and the fraction of maps having j of them is

$$\frac{1}{j!}\binom{n}{2}\binom{n-2}{2}\cdots\binom{n-2j+2}{2} = \frac{(n)_{2j}}{2^j\, j!}.$$

Thus, by inclusion-exclusion, we have

$$A_n = \sum_{j\geq 0}\sum_{k\geq 0} \frac{(n)_{2j}}{2^j\, j!}\binom{n-2j}{k}(-1)^{j+k}(n+1)^{n-2j-k}.$$

By the binomial theorem, $\sum_{k\geq 0}\binom{n-2j}{k}(-1)^k(n+1)^{n-2j-k} = n^{n-2j}$. Thus

$$A_n = \sum_{j\geq 0} \frac{(n)_{2j}}{2^j\, j!}(-1)^j n^{n-2j}.$$

This last formula can be interpreted by considering vertices of K_n unmatched in X to be represented by fixed points, rather than undefined points, of f, so that A_n is the number of maps from V to V with no exchanged pairs. □

Corollary 2.2. *As $n \to \infty$,*

$$A_n \sim \frac{n^n}{e^{1/2}}.$$

Proof. The result of Theorem 2.1 can be rewritten as

$$A_n = n^n \sum_{j\geq 0} \frac{(n)_{2j}}{n^{2j}} \frac{(-1)^j}{2^j\, j!}.$$

Thus it will suffice to show that

$$\sum_{j\geq 0} \frac{(n)_{2j}}{n^{2j}} \frac{(-1)^j}{2^j\, j!} \to \frac{1}{e^{1/2}}$$

as $n \to \infty$. Using

$$\begin{aligned}\frac{(n)_{2j}}{n^{2j}} &= \prod_{0\leq i<2j}\left(1-\frac{i}{n}\right)\\ &= \left\{1+O\left(\frac{j^2}{n}\right)\right\}\end{aligned}$$

for $j \leq \log_2 n$ and

$$\left|\frac{(n)_{2j}}{n^{2j}} \frac{(-1)^j}{2^j\, j!}\right| = O\left(\frac{1}{n^2}\right)$$

for $j > \log_2 n$, we obtain

$$\sum_{j\geq 0} \frac{(n)_{2j}}{n^{2j}} \frac{(-1)^j}{2^j\, j!} = \frac{1}{e^{1/2}}\left\{1+O\left(\frac{(\log n)^2}{n}\right)\right\},$$

since $\sum_{j\geq 0}(-1)^j/2^j\, j! = e^{-1/2}$. □

Let

$$A(z) = \sum_{n\geq 1} \frac{A_n\, z^n}{n!}$$

be the exponential generating function for the sequence $\{A_n\}_{n\geq 1}$. Let R_n denote the number of rooted labelled trees on n vertices. Cayley [C] showed that $R_n = n^{n-1}$.

Let

$$R(z) = \sum_{n\geq 1} \frac{R_n\, z^n}{n!} = \sum_{n\geq 1} \frac{n^{n-1}\, z^n}{n!}$$

be the exponential generating function for rooted labelled trees. Pólya [P2] and Pólya and Read [P3] showed that $R(z)$ satisfies the functional equation

$$R(z) = z\, \exp R(z).$$

Theorem 2.3. *We have*

$$A(z) = \frac{\exp\bigl(-\frac{1}{2} R(z)^2\bigr)}{1 - R(z)}.$$

Proof. Using the interpretation at the end of the proof of Theorem 2.1, we enumerate maps from V (the vertices of K_n) to V having no exchanged pairs. The graph of such a map comprises a number of components. Each component contains a directed cycle, where each vertex of the cycle is the root of a tree in which all edges are directed toward the root. If $R(z)$ is the exponential generating function for labelled rooted trees, then $R(z)^m/m$ is the exponential generating function for components containing a cycle of length m. Since exchanged pairs correspond to cycles of length 2, the exponential generating function for components is

$$C(z) = \sum_{m\geq 1} \frac{R(z)^m}{m} - \tfrac{1}{2} R(z)^2 = \log \frac{1}{1 - R(z)} - \tfrac{1}{2} R(z)^2.$$

Applying Pólya's component principle (if the exponential generating function $U(z)$ enumerates labelled components, then the exponential generating function $\exp U(z)$ enumerates labelled structures comprising zero or more components), we obtain

$$A(z) = \exp C(z) = \frac{\exp\bigl(-\frac{1}{2} R(z)^2\bigr)}{1 - R(z)},$$

which completes the proof of the theorem. □

We note that Theorem 2.3 can be used to provide an alternative derivation of Corollary 2.2. The singularity of $R(z)$ closest to the origin is at $z = 1/e$, and $R(z)$ has a branch point of order 2 with the expansion

$$R(z) = 1 - 2^{1/2}(1 - ez)^{1/2} + O(1 - ez)$$

about this point (see Pippenger [P1, p. 96]). Furthermore, we have

$$|R(z)| \leq \sum_{n\geq 1} \frac{R_n\, |z|^n}{n!} < \sum_{n\geq 1} \frac{R_n\, e^{-n}}{n!} = R(1/e) = 1$$

for $|z| < 1/e$. Thus $A(z)$ also has $z = 1/e$ as its singularity closest to the origin, with the expansion

$$A(z) = \left(\frac{e}{2}\right)^{1/2} \frac{1}{(1-ez)^{1/2}} + O(1)$$

about this point. Applying Darboux's lemma (see Darboux [D] or Knuth and Wilf [K]), we obtain

$$\frac{A_n}{n!} \sim \left(\frac{e}{2}\right)^{1/2} (-1)^n \binom{-\frac{1}{2}}{n} e^n.$$

Since $n! \sim (2\pi n)^{1/2} e^{-n} n^n$ and $(-1)^n \binom{-\frac{1}{2}}{n} = \binom{2n}{n}/4^n \sim 1/(\pi n)^{1/2}$, we obtain Corollary 2.2.

3. Enumerating $\tilde{M}\big(I(K_n)\big)$. Let a_n denote the cardinality of $\tilde{M}\big(I(K_n)\big)$. Let

$$a(z) = \sum_{n\ge 1} a_n\, z^n$$

be the ordinary generating function for the sequence $\{a_n\}_{n\ge 1}$. Let r_n denote the number of rooted unlabelled trees on n vertices. Let

$$r(z) = \sum_{n\ge 1} r_n\, z^n$$

be the ordinary generating function for rooted unlabelled trees. Otter [O] showed that $r(z)$ satisfies the functional equation

$$r(z) = z\, \exp \sum_{h\ge 1} \frac{r(z^h)}{h}.$$

THEOREM 3.1. *We have*

$$a(z) = \prod_{m\ge 1} \frac{\exp\left(-\frac{1}{2m}\big(r(z^m)^2 + r(z^{2m})\big)\right)}{1 - r(z^m)}.$$

Proof. We proceed as in the proof of Theorem 2.3, with three differences. First, we are enumerating unlabelled, rather than labelled, structures, so we use the ordinary generating function $r(z)$, rather than the exponential generating function $R(z)$, for trees. Second, we use the cycle index $\frac{1}{m}\sum_{ij=m} \phi(j)\, r(z^j)^i$ (where $\phi(j)$ is Euler's function, the number of elements of $\{0, 1, \ldots, j-1\}$ relatively prime to j), rather than $R(z)^m/m$, to enumerate unlabelled cycles of length m. This gives

$$\begin{aligned} c(z) &= \sum_{m\ge 1} \frac{1}{m} \sum_{ij=m} \phi(j)\, r(z^j)^i - \tfrac{1}{2}\big(r(z)^2 + r(z^2)\big) \\ &= \sum_{j\ge 1} \frac{\phi(j)}{j} \sum_{i\ge 1} \frac{r(z^j)^i}{i} - \tfrac{1}{2}\big(r(z)^2 + r(z^2)\big) \\ &= \sum_{j\ge 1} \frac{\phi(j)}{j} \log \frac{1}{1 - r(z^j)} - \tfrac{1}{2}\big(r(z)^2 + r(z^2)\big) \end{aligned}$$

for the ordinary generating function enumerating unlabelled components. Third, we use Pólya's component principle for unlabelled, rather than labelled, structures. (If the ordinary generating function $u(z)$ enumerates unlabelled components, then the ordinary generating function $\exp \sum_{h\geq 1} \frac{1}{h} u(z^h)$ enumerates unlabelled structures comprising zero or more components.) Using $\sum_{j|m} \phi(j) = m$ we obtain

$$\begin{aligned} a(z) &= \exp\left(\sum_{h\geq 1} \frac{1}{h} \left(\sum_{j\geq 1} \frac{\phi(j)}{j} \log \frac{1}{1-r(z^{hj})}\right) - \tfrac{1}{2}\big(r(z^h)^2 + r(z^{2h})\big)\right) \\ &= \exp\left(\sum_{m\geq 1} \log \frac{1}{1-r(z^m)} - \tfrac{1}{2}\big(r(z^m)^2 + r(z^{2m})\big)\right) \\ &= \prod_{m\geq 1} \frac{\exp\big(-\frac{1}{2m}\big(r(z^m)^2 + r(z^{2m})\big)\big)}{1-r(z^m)}, \end{aligned}$$

which completes the proof of the theorem. □

We note that the generating function given in Theorem 3.1 differs merely by the factor of $\prod_{m\geq 1} \exp\big(-\frac{1}{2m}\big(r(z^m)^2 + r(z^{2m})\big)\big)$ from the generating function

$$v(z) = \prod_{m\geq 1} \frac{1}{1-r(z^m)}$$

derived by Read [R] for the number of unlabelled functional digraphs.

Our next result requires the definition of some constants associated with the generating function $r(z) = \sum_{n\geq 1} r_n\, z^n$ for rooted unlabelled trees. We define the function

$$\begin{aligned} \Psi(z) &= \sum_{h\geq 2} \frac{r(z^h)}{h} \\ &= \sum_{n\geq 1} r_n \left(\log \frac{1}{1-z^n} - z^n\right). \end{aligned}$$

The singularity of $r(z)$ closest to the origin is at $z = z_0$, where z_0 is the unique positive real solution of the equation $z = \exp -\big(1 + \Psi(z)\big)$. Numerical computation yields $z_0 = 0.3383\ldots$. We also define the constant $A = 1 + z_n\, \Psi'(z_0)$. Using the expansion

$$z\Psi(z) = \sum_{n\geq 1} n r_n \left(\frac{z^n}{1-z^n} - z^n\right),$$

numerical computation yields $A = 1.215\ldots$.

Corollary 3.2. *As $n \to \infty$,*

$$a_n \sim \frac{c}{n^{1/2}} \left(\frac{1}{z_0}\right)^n,$$

where

$$c = \frac{\exp\big(-\frac{1}{2} r(z_0^2)\big)}{(2A\pi e)^{1/2}} \prod_{h\geq 2} \frac{\exp\big(\frac{1}{2h}\big(r(z_0^h)^2 + r(z_0^{2h})\big)\big)}{1-r(z_0^h)}.$$

Proof. The singularity of $r(z)$ closest to the origin is at $z = z_0$, and $r(z)$ has a branch point of order 2 with the expansion

$$r(z) = 1 - (2A)^{1/2}(1 - z/z_0)^{1/2} + O(1 - z/z_0)$$

about this point (see Pippenger [P1, p. 104]). Furthermore, we have

$$|r(z)| \le \sum_{n\ge1} r_n\,|z|^n < \sum_{n\ge1} r_n\, z_0^{-n} = r(z_0) = 1$$

for $|z| < z_0$. Thus $a(z)$ also has $z = z_0$ as its singularity closest to the origin, with the expansion

$$a(z) = \frac{\exp\bigl(-\frac{1}{2}r(z_0^2)\bigr)}{(2Ae)^{1/2}(1 - z/z_0)^{1/2}} \prod_{h\ge2} \frac{\exp\bigl(\frac{1}{2h}\bigl(r(z_0^h)^2 + r(z_0^{2h})\bigr)\bigr)}{1 - r(z_0^h)} + O(1)$$

about this point. Applying Darboux's lemma, we obtain

$$a_n \sim \frac{(-1)^n}{z_0^n}\binom{-\frac{1}{2}}{n}\frac{\exp\bigl(-\frac{1}{2}r(z_0^2)\bigr)}{(2Ae)^{1/2}} \prod_{h\ge2} \frac{\exp\bigl(\frac{1}{2h}\bigl(r(z_0^h)^2 + r(z_0^{2h})\bigr)\bigr)}{1 - r(z_0^h)}.$$

Since $(-1)^n\binom{-\frac{1}{2}}{n} \sim 1/(\pi n)^{1/2}$, we obtain Corollary 3.2. □

The argument used to prove this corollary can also be used to derive the asymptotic behavior of the number v_n of unlabelled functional digraphs on n vertices:

$$v_n \sim \frac{1}{(2A\pi en)^{1/2} z_0^n} \prod_{h\ge2} \frac{1}{1 - r(z_0^h)}.$$

4. Enumerating $\boldsymbol{M\bigl(I(K_{n,n})\bigr)}$. Let B_n denote the cardinality of $M\bigl(I(K_{n,n})\bigr)$.

THEOREM 4.1. *We have*

$$B_n = \sum_{j\ge0} (-1)^j\, j!\binom{n}{j}^2 (n+1)^{2n-2j}.$$

Proof. Consider a matching $X \in M\bigl(I(K_{n,n})\bigr)$. For each edge $\{e, v\} \in X$ (where $e = \{v, w\} \in E$ is an edge of $K_{n,n} = (V, W, E)$ and $v \in V \cup W$ is a vertex incident with e), we shall direct the edge $e = \{v, w\}$ out of v and into w. In this way we direct some of the edges of $K_{n,n}$. These directed edges form the graph of a map $\sigma : D \to V \cup W$ from a subset D of the vertices of $V \cup W$ to $V \cup W$. This map exchanges V and W. (That is, it takes vertices in V to vertices in W, and vertices in W to vertices in V.) Furthermore, this map does not have any exchanged pairs of points ($\sigma(v) = w$ and $\sigma(w) = v$). Conversely, every map $\sigma : D \to V \cup W$ with $D \subseteq V \cup W$ that exchanges V and W and has no exchanged pairs arises in this way from unique matching in $M\bigl(I(K_{n,n})\bigr)$.

The number of maps from a subset of $V \cup W$ to $V \cup W$ that take vertices in V to vertices in W, and vertices in W to vertices in V, is $(n+1)^{2n}$. We can count the number of these having no exchanged pairs by using the principle of inclusion-exclusion. There are n^2 possible exchanged pairs, and the fraction of maps having j of them is

$$\frac{j!}{(n+1)^{2j}}\binom{n}{j}^2.$$

Thus, by inclusion-exclusion, we have

$$B_n = \sum_{j\geq 0} (-1)^j\, j!\, \binom{n}{j}^2 (n+1)^{2n-2j},$$

which completes the proof of the theorem. □

Corollary 4.2. *As $n \to \infty$,*

$$B_n \sim e\, n^{2n}.$$

Proof. The result of Theorem 4.1 can be rewritten as

$$B_n = (n+1)^{2n} \sum_{j\geq 0} \frac{(-1)^j}{j!} \frac{(n)_j^2}{(n+1)^{2j}}.$$

As in the proof of Corollary 2.2, we have

$$\frac{(-1)^j}{j!} \frac{(n)_j^2}{(n+1)^{2j}} = \frac{(-1)^j}{j!} \left\{ 1 + O\left(\frac{j^2}{n}\right) \right\},$$

so that

$$\sum_{j\geq 0} \frac{(-1)^j}{j!} \frac{(n)_j^2}{(n+1)^{2j}} = \frac{1}{e} \left\{ 1 + O\left(\frac{(\log n)^2}{n}\right) \right\}.$$

Using $(n+1)^{2n} \sim e^2\, n^{2n}$, we obtain the result of the corollary. □

Let

$$B(z) = \sum_{n\geq 1} \frac{B_n\, z^n}{n!}$$

be the exponential generating function for the sequence $\{B_n\}_{n\geq 1}$. Let $B_{n,m}$ denote the cardinality of $M\big(I(K_{n,m})\big)$. Let

$$B(x,y) = \sum_{n,m\geq 1} \frac{B_{n,m}\, x^n\, y^m}{n!\, m!}$$

be the exponential generating function for the sequence $\{B_{n,m}\}_{n,m\geq 1}$. Our strategy will be to derive the bivariate generating function $B(x,y)$ and then obtain $B(z)$ from it by a method of diagonalization.

Let $R_{n,m}$ denote the number of bicolored rooted labelled trees (that is, the number of rooted labelled trees that, when bicolored, have n vertices with the color of the root and m vertices with the other color). Austin [A] showed that $R_{n,m} = n^m\, m^{n-1}$. Let

$$\begin{aligned} R(x,y) &= \sum_{n\geq 1, m\geq 0} \frac{R_{n,m}\, x^n\, y^m}{n!\, m!} \\ &= \sum_{n\geq 1, m\geq 0} \frac{n^m\, m^{n-1}\, x^n\, y^m}{n!\, m!} \end{aligned}$$

be the exponential generating function for the sequence $\{R_{n,m}\}_{n,m\geq 1}$. Austin [A] showed that $R(x,y)$ satisfies the functional equation

$$R(x,y) = x \exp R(y,x).$$

Proposition 4.3. *We have*

$$B(x,y) = \frac{\exp\bigl(R(x,y) + R(y,x) - R(x,y)\,R(y,x)\bigr)}{1 - R(x,y)\,R(y,x)}.$$

Proof. Using the interpretation in the proof of Theorem 4.1, we enumerate maps σ from subsets $D \subseteq V \cup W$ to $V \cup W$ that exchange V and W and have no exchanged pairs. The graph of such a map comprises a number of components. Each component either is a rooted tree (where the root is a vertex in $(V \cup W) \setminus D$ at which σ is undefined) or contains a directed cycle of even length, where each vertex of the cycle is the root of a tree in which all edges are directed toward the roots. If $R(x,y)$ is the exponential generating function for bicolored rooted labelled trees, then $R(x,y) + R(y,x)$ is the exponential generating function for components that are trees, and $R(x,y)^m\,R(y,x)^m/m$ is the exponential generating function for components that contain a cycle of length $2m$. Since exchanged pairs correspond to cycles of length 2, the exponential generating function for components is

$$\begin{aligned} C(x,y) &= \sum_{m\ge 1} \frac{R(x,y)^m\,R(y,x)^m}{m} + R(x,y) + R(y,x) - \tfrac{1}{2}R(x,y)\,R(y,x) \\ &= \log\frac{1}{1 - R(x,y)\,R(y,x)} + R(x,y) + R(y,x) - \tfrac{1}{2}R(x,y)\,R(y,x). \end{aligned}$$

Applying Pólya's component principle (if the exponential generating function $U(x,y)$ enumerates labelled components, then the exponential generating function $\exp U(x,y)$ enumerates labelled structures comprising zero or more components), we obtain

$$\begin{aligned} B(x,y) &= \exp C(x,y) \\ &= \frac{\exp\bigl(R(x,y) + R(y,x) - R(x,y)\,R(y,x)\bigr)}{1 - R(x,y)\,R(y,x)}, \end{aligned}$$

which completes the proof of the theorem. □

Theorem 4.4. *We have*

$$B(z) = \frac{1}{2\pi}\int_{-\pi/2}^{3\pi/2} \frac{\exp\bigl(R_\vartheta(z) + R_{-\vartheta}(z) - R_\vartheta(z)\,R_{-\vartheta}(z)\bigr)}{1 - R_\vartheta(z)\,R_{-\vartheta}(z)}\,d\vartheta,$$

where

$$R_\vartheta(z) = R(ze^{i\vartheta}, ze^{-i\vartheta}).$$

Proof. Each term of the form $x^n\,y^n$ in $B(x,y)$ contributes a term of the form z^{2n} to $B(z)$, whereas each term of the form $x^n\,y^m$ (with $n \ne m$) in $B(x,y)$ contributes nothing to $B(z)$. □

We note that Theorem 4.4 can be used to provide an alternative derivation of Corollary 4.2. Following Pippenger [P1, pp. 97–102], we define

$$C_\vartheta(z) = \frac{R_\vartheta(z) + R_{-\vartheta}(z)}{2}.$$

From the functional equation

$$R_\vartheta(z) = z\exp\bigl(i\vartheta + R_{-\vartheta}(z)\bigr),$$

we have

$$R_\vartheta(z)\, R_{-\vartheta}(z) = z^2 \exp\bigl(2C_\vartheta(z)\bigr).$$

This allows the integrand in Theorem 4.4 to be written as

$$T_\vartheta(z) = \frac{\exp\bigl(2C_\vartheta(z) - z^2\exp\bigl(2C_\vartheta(z)\bigr)\bigr)}{1 - z^2\exp\bigl(2C_\vartheta(z)\bigr)}.$$

As before, the singularities of the integrand are those of $C_\vartheta(z)$. There are two such singularities. One of these, at

$$Z_\vartheta^+ = \exp -\mathrm{cyc}\,\vartheta,$$

is closest to the origin when ϑ is near 0, and we have the expansion

$$C_\vartheta(z) = \mathrm{cyc}\,\vartheta - (1 + \mathrm{cyc}\,\vartheta)^{1/2}(1 - z/Z_\vartheta^+)^{1/2} + O(z - Z_\vartheta^+)$$

about this point. Here cyc ϑ denotes a cycloid function having the expansion cyc $\vartheta = 1 - \vartheta^2/8 + O(\vartheta^4)$ for ϑ near 0. The other singularity, at

$$Z_\vartheta^- = -\exp -\mathrm{cyc}(\vartheta - \pi),$$

is closest to the origin when ϑ is near π, and we have the expansion

$$C_\vartheta(z) = \mathrm{cyc}(\vartheta - \pi) - \bigl(1 + \mathrm{cyc}(\vartheta - \pi)\bigr)^{1/2}(1 - z/Z_\vartheta^-)^{1/2} + O(z - Z_\vartheta^-)$$

about this point. From Theorem 4.4 we have

$$\frac{B_n}{n!^2} = \frac{1}{2\pi}\int_{-\pi/2}^{3\pi/2} [z^{2n}]\, T_\vartheta(z)\, d\vartheta.$$

We set

$$\varepsilon(n) = \left(\frac{48\log n}{n}\right)^{1/2}$$

and break the interval $I = [-\pi/2, 3\pi/2)$ into three parts: $J^+ = [-\varepsilon(n), \varepsilon(n)]$, $J^- = [\pi - \varepsilon(n), \pi + \varepsilon(n)]$, and $K = I \setminus (J^+ \cup J^-)$. For ϑ in K, Cauchy's theorem yields

$$[z^{2n}]\, T_\vartheta(z) = O\left(\frac{e^{2n}}{n^3}\right),$$

and the integral over K satisfies the same estimate. For ϑ in J^+, Darboux's lemma yields

$$[z^{2n}]\, T_\vartheta(z) = \frac{e^{2n+1}}{4(\pi n)^{1/2}}\left\{1 + O\left(\frac{(\log n)^2}{n}\right)\right\}\left\{1 + O\left(\vartheta^2\right)\right\}\exp -(n\vartheta^2/4).$$

Thus for the integral over J^+ we have

$$\frac{1}{2\pi}\int_{J^+} [z^{2n}]\, T_\vartheta(z)\, d\vartheta = \frac{e^{2n+1}}{4\pi n}\left\{1 + O\left(\frac{(\log n)^2}{n}\right)\right\}.$$

The integral over J^- satisfies the same estimate, and thus we obtain

$$\frac{B_n}{n!^2} = \frac{e^{2n+1}}{2\pi n}\left\{1 + O\left(\frac{(\log n)^2}{n}\right)\right\}.$$

Since $n! \sim (2\pi n)^{1/2} e^{-n} n^n$, we obtain Corollary 4.2.

5. Enumerating $\tilde{M}\big(I(K_{n,n})\big)$. Let b_n denote the cardinality of $\tilde{M}\big(I(K_{n,n})\big)$. Let

$$b(z) = \sum_{n\geq 1} b_n\, z^n$$

be the ordinary generating function for the sequence $\{b_n\}_{n\geq 1}$. Let $b_{n,m}$ denote the cardinality of $\tilde{M}\big(I(K_{n,m})\big)$. Let

$$b(x,y) = \sum_{n,m\geq 1} b_{n,m}\, x^n\, y^m$$

be the ordinary generating function for the sequence $\{b_{n,m}\}_{n,m\geq 1}$. Our strategy will be to derive the bivariate generating function $b(x,y)$ and then obtain $b(z)$ from it by a method of diagonalization.

Let $r_{n,m}$ denote the number of bicolored rooted unlabelled trees (that is, the number of rooted unlabelled trees that, when bicolored, have n vertices with the color of the root and m vertices with the other color). Let

$$r(x,y) = \sum_{n\geq 1, m\geq 0} r_{n,m}\, x^n\, y^m$$

be the ordinary generating function for the sequence $\{r_{n,m}\}_{n,m\geq 1}$. Pippenger [P1] showed that $r(x,y)$ satisfies the functional equation

$$r(x,y) = x \exp \sum_{h\geq 1} \frac{r(y^h, x^h)}{h}.$$

A positive integer m can be factorized as $m = v(m)\cdot w(m)$, where $v(m)$ is an integral power of 2 and $w(m)$ is an odd integer.

PROPOSITION 5.1. *We have*

$$b(x,y) = \frac{f(x,y)+g(x,y)}{2},$$

where

$$f(x,y) = \prod_{m\geq 1} \frac{\exp\!\big(\frac{1}{m}\big(r(x^m,y^m) + r(y^m,x^m) - r(x^m,y^m)\, r(y^m,x^m)\big)\big)}{1 - r(x^m,y^m)\, r(y^m,x^m)}$$

and

$$g(x,y) = \prod_{m\geq 1} \frac{\exp\!\big(\frac{1}{2m}\big(r(x^{2m}\, y^{2m}) - r(x^m\, y^m)^2\big)\big)}{1 - r(x^m\, y^m)}.$$

Proof. Using the interpretation in the proof of Theorem 4.1, we enumerate equivalence classes of maps σ from subsets $D \subseteq V\cup W$ to $V\cup W$ that exchange V and W and have no exchanged pairs, where now two maps σ and τ are considered equivalent if there is permutation π of $V\cup W$ that is (1) either *part-preserving* (that is, such that $\pi(V) = V$ and $\pi(W) = W$) or *part-exchanging* (that is, such that $\pi(V) = W$ and $\pi(W) = V$), and (2) such that $\pi\big(\tau(v)\big) = \sigma\big(\pi(v)\big)$ for all $v \in V\cup W$ (which means, in particular, that $\tau(v)$ is defined if and only if $\sigma\big(\pi(v)\big)$ is defined). We shall start by considering only part-preserving permutations. Let $f_{n,m}$ denote the number of equivalence classes of matchings in $I(K_{n,m})$ under part-preserving automorphisms of $K_{n,m}$ and $I(K_{n,m})$. Let

$$f(x,y) = \sum_{n,m\geq 1} f_{n,m}\, x^n\, y^m$$

be the ordinary generating function for the sequence $\{f_{n,m}\}_{n,m\geq 1}$. We shall show first that $f(x,y)$ is as given in the statement of the theorem.

Next we shall consider part-exchanging permutations. If a matching has no part-exchanging automorphism, then it, together with its mate obtained by exchanging V and W, are counted twice in $f(x,y)$. If, on the other hand, it has a part-exchanging automorphism (which can happen only when $n = m$), then it is counted just once. Let $g_{n,m}$ denote the number of equivalence classes (under part-preserving automorphisms) of matchings in $I(K_{n,m})$ that have at least one part-exchanging automorphism. Let

$$g(x,y) = \sum_{n,m\geq 1} g_{n,m}\, x^n\, y^m$$

be the ordinary generating function for the sequence $\{f_{n,m}\}_{n,m\geq 1}$. (We have $g_{n,m} = 0$ whenever $n \neq m$, so $g(x,y)$ is actually a power series in the product xy.) We shall show that $g(x,y)$ is as given in the statement of the theorem.

Finally, it follows that $b(x,y) = f(x,y)/2 + g(x,y)/2$, since a matching without a part-exchanging automorphism is counted with weight 1 by the first term, while one with a part-exchanging automorphism is counted with weight 1/2 by the first term, and again with weight 1/2 by the second term.

To derive $f(x,y)$, we proceed as in the proof of Proposition 4.3, with three differences. First, we are enumerating unlabelled, rather than labelled, structures, so we use the ordinary generating function $r(x,y)$, rather than the exponential generating function $R(x,y)$, for trees. Second, we use the cycle index

$$\frac{1}{m} \sum_{ij=m} \phi(j)\, r(x^j, y^j)^i\, r(y^j, x^j)^i,$$

rather than $R(x,y)^m\, R(y,x)^m/m$, to enumerate unlabelled cycles of length $2m$. This gives

$$\begin{aligned} c(x,y) &= \sum_{m\geq 1} \frac{1}{m} \sum_{ij=m} \phi(j)\, r(x^j, y^j)^i\, r(y^j, x^j)^i + r(x,y) + r(y,x) - r(x,y)\, r(y,x) \\ &= \sum_{j\geq 1} \frac{\phi(j)}{j} \sum_{i\geq 1} \frac{r(x^j, y^j)^i\, r(y^j, x^j)^i}{i} + r(x,y) + r(y,x) - r(x,y)\, r(y,x) \\ &= \sum_{j\geq 1} \frac{\phi(j)}{j} \log \frac{1}{1 - r(x^j, y^j)\, r(y^j, x^j)} + r(x,y) + r(y,x) - r(x,y)\, r(y,x) \end{aligned}$$

(where we have added the terms $r(x,y) + r(y,x)$ for the components that are trees) for the ordinary generating function enumerating unlabelled components. Third, we use Pólya's component principle for unlabelled, rather than labelled, structures. (If the ordinary generating function $u(x,y)$ enumerates unlabelled components, then the ordinary generating function $\exp \sum_{h \geq 1} \frac{1}{h} u(x^h, y^h)$ enumerates unlabelled structures comprising zero or more components.) We obtain

$$\begin{aligned}
f(x,y) &= \exp \sum_{h \geq 1} \frac{1}{h} \left(\sum_{j \geq 1} \frac{\phi(j)}{j} \log \frac{1}{1 - r(x^{hj}, y^{hj})\, r(y^{hj}, x^{hj})} \right) \\
&\qquad + \frac{r(x^h, y^h) + r(y^h, x^h) - r(x^h, y^h)\, r(y^h, x^h)}{h} \\
&= \exp \\
&\quad \left(\sum_{m \geq 1} \log \frac{1}{1 - r(x^m, y^m)} + \frac{r(x^m, y^m) + r(y^m, x^m) - r(x^m, y^m) r(y^m, x^m)}{m} \right) \\
&= \prod_{m \geq 1} \frac{\exp\bigl(\frac{1}{m}\bigl(r(x^m, y^m) + r(y^m, x^m) - r(x^m, y^m)\, r(y^m, x^m)\bigr)\bigr)}{1 - r(x^m, y^m)\, r(y^m, x^m)},
\end{aligned}$$

which completes the derivation of $f(x,y)$.

To derive $g(x,y)$ we proceed as for $f(x,y)$, but we observe that components that do not themselves have a part-exchanging automorphism must come in pairs, along with their mate obtained by exchanging V and W. Our goal then is to derive an ordinary generating function for components that have a part-exchanging automorphism. Such a component cannot be a tree, since a tree has its root in one part or the other. Thus it must contain a cycle of even length $2m$, and its part-exchanging automorphism must rotate this cycle by an odd number of vertices. This odd number of vertices is relatively prime to $v(2m)$, so the component must comprise $w(m)$ sets of trees, each of which contains $v(m)$ trees along with their $v(m)$ mates. The ordinary generating function for a tree along with its mate is $r(xy, xy) = r(xy)$. Thus the ordinary generating function for such components is

$$\frac{1}{w(m)} \sum_{ij = w(m)} \phi(j)\, r(x^{jv(m)}\, y^{jv(m)})^i.$$

Thus the ordinary generating function for all such components (except those associated with exchanged pairs) is

$$\begin{aligned}
d(x,y) &= \left(\sum_{m \geq 1} \frac{1}{w(m)} \sum_{ij = w(m)} \phi(j)\, r(x^{jv(m)}\, y^{jv(m)})^i \right) - r(xy) \\
&= \left(\sum_{v = 2^t \geq 1} \; \sum_{\text{odd } w \geq 1} \frac{1}{w} \sum_{ij = w} \phi(j)\, r(x^{jv}\, y^{jv})^i \right) - r(xy) \\
&= \left(\sum_{v = 2^t \geq 1} \; \sum_{\text{odd } j \geq 1} \frac{\phi(j)}{j} \sum_{\text{odd } i \geq 1} \frac{1}{i}\, r(x^{jv}\, y^{jv})^i \right) - r(xy) \\
&= \left(\sum_{v = 2^t \geq 1} \; \sum_{\text{odd } j \geq 1} \frac{\phi(j)}{2j} \log \left(\frac{1 + r(x^{jv}\, y^{jv})}{1 - r(x^{jv}\, y^{jv})} \right) \right) - r(xy).
\end{aligned}$$

The ordinary generating function for components that do not have a part-exchanging automorphism is thus $c(x,y) - d(x,y)$, and for pairs of these components along with their mates is $\big(c(xy,xy) - d(xy,xy)\big)/2$. Thus we obtain $g(x,y)$ by applying Pólya's component principle to $\big(c(xy,xy) - d(xy,xy)\big)/2 + d(x,y)$. For the first term, we have

$$\begin{aligned}
\sum_{h\geq 1} \frac{c(x^h y^h, x^h y^h)}{2h} &= \sum_{h\geq 1} \frac{1}{2h} \sum_{j\geq 1} \frac{\phi(j)}{j} \log \frac{1}{1 - r(x^{jh}\, y^{jh})^2} \\
&\qquad + \frac{r(x^h\, y^h)}{h} - \frac{r(x^h\, y^h)^2}{2h} \\
&= \sum_{m\geq 1} \frac{1}{2m} \sum_{j|m} \phi(j) \log \frac{1}{1 - r(x^m\, y^m)^2} \\
&\qquad + \frac{r(x^m\, y^m)}{m} - \frac{r(x^m\, y^m)^2}{2m} \\
&= \sum_{m\geq 1} \frac{1}{2} \log \frac{1}{1 - r(x^m\, y^m)^2} \\
&\qquad + \frac{r(x^m\, y^m)}{m} - \frac{r(x^m\, y^m)^2}{2m}.
\end{aligned}$$

For the last two terms, we have

$$\begin{aligned}
&\sum_{h\geq 1} \frac{d(x^m, y^m)}{h} - \frac{d(x^h y^h, x^h y^h)}{2h} \\
&\quad = \sum_{h\geq 1} \frac{1}{h} \sum_{u=2^t\geq 1} \sum_{\text{odd } j\geq 1} \frac{\phi(j)}{2j} \log \frac{1 + r(x^{huj}\, y^{huj})}{1 - r(x^{huj}\, y^{huj})} - \frac{r(x^h\, y^h)}{h} \\
&\qquad - \sum_{h\geq 1} \frac{1}{2h} \sum_{u=2^t\geq 1} \sum_{\text{odd } j\geq 1} \frac{\phi(j)}{2j} \log \frac{1 + r(x^{2huj}\, y^{2huj})}{1 - r(x^{2huj}\, y^{2huj})} + \frac{r(x^{2h}\, y^{2h})}{2h} \\
&\quad = \sum_{m\geq 1} \frac{1}{m} \sum_{u=2^t\geq 1} \sum_{\text{odd } j|m} \frac{\phi(j)}{2} \log \frac{1 + r(x^{mu}\, y^{mu})}{1 - r(x^{mu}\, y^{mu})} \\
&\qquad - \sum_{m\geq 1} \frac{1}{2m} \sum_{u=2^t\geq 1} \sum_{\text{odd } j|m} \frac{\phi(j)}{2} \log \frac{1 + r(x^{2mu}\, y^{2mu})}{1 - r(x^{2mu}\, y^{2mu})} - \sum_{\text{odd } k\geq 1} \frac{r(x^k\, y^k)}{k} \\
&\quad = \sum_{m\geq 1} \frac{1}{v(m)} \sum_{u=2^t\geq 1} \frac{1}{2} \log \frac{1 + r(x^{mu}\, y^{mu})}{1 - r(x^{mu}\, y^{mu})} \\
&\qquad - \sum_{m\geq 1} \frac{1}{v(vm)} \sum_{u=2^t\geq 1} \frac{1}{2} \log \frac{1 + r(x^{2mu}\, y^{2mu})}{1 - r(x^{2mu}\, y^{2mu})} - \sum_{\text{odd } k\geq 1} \frac{r(x^k\, y^k)}{k} \\
&\quad = \sum_{\text{odd } k\geq 1} \sum_{u=2^t\geq 1} \frac{1}{2} \log \frac{1 + r(x^{ku}\, y^{ku})}{1 - r(x^{ku}\, y^{ku})} - \sum_{\text{odd } k\geq 1} \frac{r(x^k\, y^k)}{k} \\
&\quad = \sum_{m\geq 1} \frac{1}{2} \log \frac{1 + r(x^m\, y^m)}{1 - r(x^m\, y^m)} - \sum_{\text{odd } k\geq 1} \frac{r(x^k\, y^k)}{k},
\end{aligned}$$

since $\sum_{\text{odd } j|m} \phi(j) = w(m)$. Combining these results, we obtain

$$g(x,y) = \exp \sum_{h\geq 1} \frac{c(x^h\, y^h, x^h\, y^h) - d(x^h\, y^h, x^h\, y^h) + 2d(x^h, y^h)}{2h}$$
$$= \prod_{m\geq 1} \frac{\exp\bigl(\frac{1}{2m}\bigl(r(x^{2m}\, y^{2m}) - r(x^m\, y^m)^2\bigr)\bigr)}{1 - r(x^m\, y^m)},$$

which completes the derivation of $g(x,y)$, and thus the proof of the proposition. □

Theorem 5.2. *We have*

$$b(z) = \frac{1}{2\pi} \int_{-\pi/2}^{3\pi/2} b(ze^{i\vartheta}, ze^{-i\vartheta})\, d\vartheta.$$

Proof. Each term of the form $x^n\, y^n$ in $b(x,y)$ contributes a term of the form z^{2n} to $b(z)$, whereas each term of the form $x^n\, y^m$ (with $n \neq m$) in $b(x,y)$ contributes nothing to $b(z)$. □

Our next result requires the definition of some constants associated with the generating function $r(x,y) = \sum_{n\geq 1, m\geq 0} r_{n,m}\, x^n\, y^m$ for bicolored rooted unlabelled trees. Define the power series $q(z) = \sum_{n\geq 1} q_n\, z^n$ by

$$q(z) = \sum_{n\geq 1, m\geq 0} (n-m)\, r_{n,m}\, z^{n+m}$$

and then define

$$B = 1 - \sum_{n\geq 1} q_n \left(\frac{z_0^n}{1 - z_0^n} - z_0^n \right).$$

Numerical computation yields $B = 0.8269\ldots$. Next define the power series $p(z) = \sum_{n\geq 1} p_n\, z^n$ by

$$p(z) = \sum_{n\geq 1, m\geq 0} (n-m)^2\, r_{n,m}\, z^{n+m}$$

and then define

$$C = -\sum_{n\geq 1} p_n \left(\frac{z_0^n}{(1 - z_0^n)^2} - z_0^n \right).$$

Numerical computation yields $C = -0.4450\ldots$.

Corollary 5.3. *As $n \to \infty$,*

$$b_n \sim \frac{d}{n} \left(\frac{1}{z_0} \right)^{2n},$$

where

$$d = \frac{e}{4\pi (B^2 - 4C)^{1/2}} \prod_{m\geq 2} \frac{\exp\bigl(\frac{1}{m}\bigl(2r(z_0^m) + r(z_0^m)^2\bigr)\bigr)}{1 - r(z_0^m)^2}.$$

Proof. As before, we shall apply Darboux's lemma to the integrand in Theorem 5.2, and thus we shall be concerned with singularities closest to the origin. Following Pippenger [P1, pp. 104–114], we define

$$c_\vartheta(z) = \frac{c_\vartheta(z) + c_{-\vartheta}(z)}{2}.$$

From the functional equation

$$r_\vartheta(z) = z \exp\left(i\vartheta + \sum_{h \geq 1} \frac{r_{-h\vartheta}(z^h)}{h} \right),$$

we have

$$r_\vartheta(z)\, r_{-\vartheta}(z) = z^2 \exp\left(2 \sum_{h \geq 1} \frac{c_{h\vartheta}(z^h)}{h} \right).$$

As before, the singularities of

$$t_\vartheta(z) = \frac{1}{1 - r_\vartheta(z)\, r_{-\vartheta}(z)}$$

are those of $c_\vartheta(z)$. One of these, at

$$z_\vartheta^+ = z_0 \left(1 + \frac{B^2 - 4C}{8A} \vartheta^2 + O(\vartheta^4) \right),$$

is closest to the origin when ϑ is near 0, and we have the expansion

$$c_\vartheta(z) = \bigl(1 + O(\vartheta^2)\bigr) - (2A)^{1/2}\bigl(1 + O(\vartheta^2)\bigr)(1 - z/z_\vartheta^+)^{1/2} + O(z - z_\vartheta^+)$$

about this point. Another singularity, at

$$z_\vartheta^- = -z_0 \left(1 + \frac{B^2 - 4C}{8A} \vartheta^2 + O(\vartheta^4) \right),$$

is closest to the origin when ϑ is near π, and we have the expansion

$$c_\vartheta(z) = \bigl(1 + O(\vartheta^2)\bigr) - (2A)^{1/2}\bigl(1 + O(\vartheta^2)\bigr)(1 - z/z_\vartheta^-)^{1/2} + O(z - z_\vartheta^-)$$

about this point.

Let us now estimate

$$t_n = \frac{1}{2\pi} \int_{-\pi/2}^{3\pi/2} [z^{2n}]\, t_\vartheta(z)\, d\vartheta.$$

We set

$$\varepsilon(n) = \left(\frac{48 \log n}{n} \right)^{1/2}$$

and break the interval $I = [-\pi/2, 3\pi/2)$ into three parts: $J^+ = [-\varepsilon(n), \varepsilon(n)]$, $J^- = [\pi - \varepsilon(n), \pi + \varepsilon(n)]$, and $K = I \setminus (J^+ \cup J^-)$. For ϑ in K, Cauchy's theorem yields

$$[z^{2n}]\, t_\vartheta(z) = O\left(\frac{z_0^{-2n}}{n^3} \right),$$

and the integral over K satisfies the same estimate. For ϑ in J^+, Darboux's lemma yields

$$[z^{2n}]\, t_\vartheta(z) = \frac{z_0^{-2n}}{4(A\pi n)^{1/2}} \left\{1 + O\left(\frac{(\log n)^2}{n}\right)\right\} \left\{1 + O\left(\vartheta^2\right)\right\} \exp - \left(\frac{n\vartheta^2(B^2-4C)}{4A}\right).$$

Thus for the integral over J^+ we have

$$\frac{1}{2\pi}\int_{J^+} [z^{2n}]\, t_\vartheta(z)\, d\vartheta = \frac{z_0^{-2n}}{4\pi n(B^2-4C)^{1/2}} \left\{1 + O\left(\frac{(\log n)^2}{n}\right)\right\}.$$

The integral over J^- satisfies the same estimate, and thus we obtain

$$t_n = \frac{z_0^{-2n}}{2\pi n(B^2-4C)^{1/2}} \left\{1 + O\left(\frac{(\log n)^2}{n}\right)\right\}.$$

Let us now estimate

$$f_n = \frac{1}{2\pi}\int_{-\pi/2}^{3\pi/2} f(ze^{i\vartheta}, ze^{-i\vartheta})\, d\vartheta.$$

Writing $f(x,y)$ as

$$\begin{aligned} f(x,y) = {} & \frac{\exp\bigl(\bigl(r(x,y) + r(y,x) - r(x,y)\, r(y,x)\bigr)\bigr)}{1 - r(x,y)\, r(y,x)} \\ & \times \prod_{m\ge 2} \frac{\exp\bigl(\frac{1}{m}\bigl(r(x^m,y^m) + r(y^m,x^m) - r(x^m,y^m)\, r(y^m,x^m)\bigr)\bigr)}{1 - r(x^m,y^m)\, r(y^m,x^m)}, \end{aligned}$$

we see that the asymptotic behavior of f_n is determined by that of the denominator of the first factor, which we have already analyzed as t_n, whereas the numerator of the first factor and all of the remaining factors merely contribute constant factors to the result. Thus we have

$$\begin{aligned} f_n & \sim e t_n \prod_{m\ge 2} \frac{\exp\bigl(\frac{1}{m}\bigl(2r(z_0^m) + r(z_0^m)^2\bigr)\bigr)}{1 - r(z_0^m)^2} \\ & \sim \frac{e\, z_0^{-2n}}{2\pi n(B^2-4C)^{1/2}} \prod_{m\ge 2} \frac{\exp\bigl(\frac{1}{m}\bigl(2r(z_0^m) + r(z_0^m)^2\bigr)\bigr)}{1 - r(z_0^m)^2}. \end{aligned}$$

Let us now estimate

$$g_n = \frac{1}{2\pi}\int_{-\pi/2}^{3\pi/2} g(ze^{i\vartheta}, ze^{-i\vartheta})\, d\vartheta.$$

Every term of $g(x,y)$ is of the form $x^n\, y^m$, so that we have $g(ze^{i\vartheta}, ze^{-i\vartheta}) = g(z,z)$, so that no integration is necessary. Furthermore, $g(z,z)$ has no singularity closer to the origin than $z_0^{1/2} > z_0$, so that we have $g_n = O(z_0^{-n})$, which is negligible as compared with f_n.

Thus we have

$$\begin{aligned} b_n & = \frac{f_n + g_n}{2} \\ & \sim \frac{e\, z_0^{-2n}}{4\pi n(B^2-4C)^{1/2}} \prod_{m\ge 2} \frac{\exp\bigl(\frac{1}{m}\bigl(2r(z_0^m) + r(z_0^m)^2\bigr)\bigr)}{1 - r(z_0^m)^2}, \end{aligned}$$

which completes the proof of the theorem. □

Acknowledgment. The author would like to thank Stephane Durocher; his enumeration of $\tilde{M}\big(I(K_5)\big)$ and $\tilde{M}\big(I(K_{3,3})\big)$ by hand led to the research reported in this paper.

REFERENCES

[A] T. L. Austin, *The enumeration of point-labelled chromatic graphs and trees*, Canad. J. Math., 12 (1960) pp. 535–545.

[B1] V. E. Beneš, *Programming and control problems arising from optimal routing in telephone networks*, Bell System Tech. J., 45 (1966) pp. 1373–1438.

[B2] V. E. Beneš, *Reduction of network states under symmetries*, Bell System Tech. J., 57 (1977) pp. 111–149.

[C] A. Cayley, *A theorem on trees*, Quart. J. Math., 23 (1889) pp. 376–378.

[D] G. Darboux, *Mémoire sur l'approximation des fonctions des très grands nombres, et sur une classe étendue des développements en série*, J. Math. Pures Appl., 4 (1878) pp. 5–56, 377–416.

[H] F. Harary, *The number of functional digraphs*, Math. Ann., 138 (1959) pp. 203–210.

[K] D. E. Knuth and H. S. Wilf, *A short proof of Darboux's lemma*, Appl. Math. Lett., 2 (1989) pp. 139–140.

[O] R. Otter, *The number of trees*, Ann. Math. (2), 49 (1948) pp. 583–599.

[P1] N. Pippenger, *Enumeration of equicolorable trees*, SIAM J. Discrete Math., 14 (2001) pp. 93–115.

[P2] G. Pólya, *Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen*, Acta Math., 68 (1937) pp. 145–254.

[P3] G. Pólya and R. C. Read, *Combinatorial Enumeration of Groups, Graphs and Chemical Compounds*, Springer-Verlag, New York, 1987.

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

SIAM J. Discrete Math.
Vol. 16, No. 1, pp. 65–73

COUNTING CLAW-FREE CUBIC GRAPHS[*]

EDGAR M. PALMER[†], RONALD C. READ[‡], AND ROBERT W. ROBINSON[§]

Abstract. Let H_n be the number of claw-free cubic graphs on $2n$ labeled nodes. Combinatorial reductions are used to derive a second order, linear homogeneous differential equation with polynomial coefficients whose power series solution is the exponential generating function for $\{H_n\}$. This leads to a recurrence relation for H_n which shows $\{H_n\}$ to be P-recursive and which enables the sequence to be computed efficiently. Thus the enumeration of labeled claw-free cubic graphs can be added to the handful of known counting problems for regular graphs with restrictions which have been proved P-recursive.

Key words. labeled graph counting, claw-free graph, cubic graph, exponential generating function, P-recursive sequence

AMS subject classifications. 05A15, 05C30

PII. S0895480194274777

1. Introduction.

The problem of generating cubic graphs, i.e., 3-regular graphs, has been studied for over 100 years using combinatorial reductions [6]. Read applied combinatorial reductions to the derivation of an efficient recurrence relation for counting the number of labeled connected cubic graphs on $2n$ nodes [12], in which the nodes are labeled but not the edges. He observed that expressing the recurrence relations in terms of an exponential generating function (EGF) resulted in substantial simplifications. This allowed him to derive a second order linear differential equation for the EGF of all labeled cubic graphs (not necessarily connected). Later, Wormald [16] incorporated EGFs directly into the reduction approach in order to obtain differential equations for the EGFs of cubic graphs of given k-connectivity ($k = 0$, 1, 2, and 3). He derived recurrence relations only at the end of the process. In the present paper we will follow this pattern in deriving a recurrence relation for the exact number H_n of labeled claw-free cubic graphs on $2n$ nodes. A graph is *claw-free* if and only if it contains no induced subgraph isomorphic to $K_{1,3}$. In a cubic graph, this is equivalent to the condition that every vertex lies on a triangle, i.e., on a 3-cycle.

Claw-free graphs have been studied in relation to independent sets, perfect graphs, Hamiltonicity, reconstruction, and matchings. References may be found in the introduction of [10]. In particular, claw-free graphs which are 3-regular or 4-regular have been amenable to analysis of extendibility of matchings [9]. Related questions and conjectures on Hamiltonicity arising from this work are presented in [11]. For cubic claw-free graphs, Plummer asks for the probabilistic behavior of Hamiltonicity in cubic claw-free graphs, in the planar case, and in general. The latter question was answered in [13] where it was determined that almost all claw-free cubic graphs are Hamiltonian. For 4-connected 4-regular claw-free graphs, Plummer conjectures that all are Hamiltonian [11, Conjecture 3.8]. The asymptotic behavior of the sequence

[*] Received by the editors September 23, 1994; accepted for publication (in revised form) June 24, 2002; published electronically November 6, 2002.

http://www.siam.org/journals/sidma/16-1/27477.html

[†] Department of Mathematics, Michigan State University, East Lansing, MI 48824-1027 (palmer@math.msu.edu).

[‡] Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Waterloo, ON N2L 3G1, Canada (rcread@math.uwaterloo.ca).

[§] Department of Computer Science, 415 GSRC, University of Georgia, Athens, GA 30602-7404 (rwr@cs.uga.edu).

$\{H_n\}$ was determined in [8], the results of which were essential for the Hamiltonicity result of [13] cited above. The enumeration of claw-free cubic graphs with given connectivity is treated in [1]. The method requires enumeration results for labeled general cubic graphs [2].

The recurrence relation obtained in section 3 for H_n allows the n numbers $H_1, \dots,$ H_n to be calculated with $O(n)$ arithmetic operations. It is a linear homogeneous recurrence of order 12 in which the coefficients are polynomials in n. These polynomials range in degree up to 23 and all have integer coefficients. This recurrence shows that $\{H_n\}$ belongs to the class of P-recursive sequences, first defined by Stanley [14]. It was later shown that a number of labeled graph enumeration problems, including cubic graphs, are P-recursive [5]. Gessel generalized those results considerably and showed that for any fixed r the number of labeled r-regular graphs is P-recursive [4]. However, Gessell commented on the lack of general methods for proving P-recursiveness of the number of r-regular graphs subject to restrictions on connectivity, girth, and the like. For restricted labeled cubic graphs there are two examples of P-recursive counting problems provided by Wormald; those rooted at a triangle [15], and those containing no triangle [17]. To these we can now add the enumeration of labeled claw-free cubic graphs. However, for labeled cubic and claw-free cubic graphs which are k-connected for $k = 1$, 2, or 3 the question of P-recursiveness is open, as the enumerations provided in [16] and [1] do not provide linear recurrences.

For general graph theoretic terminology and notation we follow [7], except for adopting the more modern names nodes and edges in place of points and lines. In particular, we assume a basic knowledge of labeled enumeration techniques using EGFs, such as is provided by Chapter 1 of [7].

2. Structural properties. All graphs to be considered will have nodes labeled but not edges. A *claw-free* graph is one with no induced subgraph isomorphic to $K_{1,3}$. We will deal exclusively with *cubic* graphs, i.e., 3-regular graphs. For cubic graphs, the claw-free condition is equivalent to requiring that every node should belong to a triangle. We will count the number H_n of labeled claw-free cubic graphs on $2n$ nodes.

In any cubic graph, the maximum number of triangles in which a node may lie is 3, and this can occur only in a component isomorphic to K_4. In our counting, we will account for such components at the end. A node may lie in exactly two triangles precisely if it is one of the nodes of degree 3 in an induced subgraph isomorphic to $K_4 - e$; we call such a subgraph a *diamond*. A maximal set of diamonds which are adjacent in series is called a *string of diamonds*. A connected graph in which every node is contained in a diamond is called a *ring of diamonds*. For the purposes of counting, we consider a single edge to be a trivial string of diamonds, provided it is not incident to a diamond. However, a ring of diamonds must contain at least two diamonds. Like copies of K_4, rings of diamonds will be accounted for explicitly at the end of the process.

Denote by *reduction* the operation of replacing each string of diamonds by a single edge. For any claw-free cubic graph with no component isomorphic to K_4 or a ring of diamonds, the reduced graph must be a cubic multigraph in which every node is contained in exactly one triangle (defined as a set of three mutually adjacent nodes). Clearly, none of the edges in these remaining disjoint triangles resulted from the reduction of a nontrivial string of diamonds unless it belongs to a double edge, the nodes of which are mutually adjacent to a third node. Such a configuration is termed a *trumpet*. In the double edge of a trumpet, exactly one of the two edges must have resulted from the reduction of a nontrivial string of diamonds. Since our edges are not labeled, for counting purposes it does not matter which edge is which in the double. Denote by *expansion* the operation which is inverse to reduction.

Now, in a reduced graph we can contract each of the disjoint triangles to a single node; denote this operation by *contraction* and its inverse by *dilation.* The contraction of a trumpet will be a loop. The contraction of a reduced graph is an arbitrary cubic general graph. We could, if we wished, contract an unreduced graph by contracting just those triangles which do not overlap any other triangle. Then reduction and contraction are easily seen to be commutative operations.

The approach that we will take to counting claw-free cubic graphs is to start with cubic general graphs, dilate and expand them, and then add in components isomorphic to K_4 or a ring of diamonds.

3. Labeled cubic general graphs. Let $g_{s,d,l}$ be the number of labeled cubic general graphs without triple edges having exactly s single edges, d double edges, and l loops. Note that the number $2n$ of nodes is just

$$2n = \frac{2s+4d+2l}{3}\ .$$

It is the nodes that are labeled. Also, trumpets are not distinguished from other double edges in this treatment. The graphs are not necessarily connected, so we let $g_{0,0,0} = 1$.

Now let $G(x, y, w)$ be the exponential generating function

$$G(x,y,w) = \sum_{s,d,l} g_{s,d,l}x^s y^d w^l/(2n)!\ .$$

The partial derivations with respect to x, y, and w will be denoted G_x, G_y, and G_w, and similarly for higher order derivatives. Clearly G_x is the exponential generating function for labeled cubic general graphs without triple edges which are rooted at a single edge, except that the root edge is not represented by a factor of x. The other first order partial derivatives have like interpretations, as do the higher order derivatives. To derive an expression for G_x, we can imagine removing a single edge from a general cubic graph, leaving two nodes of degree 2. These are then smoothed over, leaving edges which we think of as root edges. The possibilities for the latter are counted by appropriate partial derivatives of G, in general, depending upon whether the root edges are singles, doubles, triples, ordinary loops, or nodeless loops. The latter occurs when an edge incident to a loop is removed. One must also multiply by a monomial which accounts for the various edges which were deleted after the original root edge was removed.

If a cubic graph is originally rooted at a single edge, then after deleting the root we have 17 possibilities for the two new root edges, as shown in Table 1 along with the corresponding exponential generating function.

Hence we have

$$\begin{aligned} G_x &= \left(\frac{w^2}{2} + \frac{x^5}{4} + \frac{x^2yw}{2} + \frac{x^4y^2}{8}\right)G + \left(x^2w + \frac{x^4y}{2}\right)G_x \\ &+ \left(\frac{x^4}{2} + x^3w + \frac{x^5y}{2}\right)G_y + \left(yw + \frac{x^2y^2}{2}\right)G_w \\ &+ \frac{x^4}{2}G_{xx} + x^5G_{xy} + x^2yG_{xw} + \frac{x^6}{2}G_{yy} + x^3yG_{yw} + \frac{y^2}{2}G_{ww}\ . \end{aligned} \tag{1}$$

TABLE 1
Terms contributing to G_x.

EGF	Root edges
$\frac{w^2}{2}G$	two nodeless loops
$\frac{x^5}{4}G$	belong to same triple edge
$\frac{x^2yw}{2}G$	triple edge and nodeless loop
$\frac{x^4y^2}{8}G$	two triple edges
x^2wG_x	single edge and nodeless loop
$\frac{x^4y}{2}G_x$	single edge and triple edge
$\frac{x^4}{2}G_y$	belong to same double edge
x^3wG_y	double edge and nodeless loop
$\frac{x^5y}{2}G_y$	double edge and triple edge
ywG_w	ordinary loop and nodeless loop
$\frac{x^2y^2}{2}G_w$	ordinary loop and triple edge
$\frac{x^4}{2}G_{xx}$	two single edges
x^5G_{xy}	single edge and double edge
x^2yG_{xw}	single edge and ordinary loop
$\frac{x^6}{2}G_{yy}$	two double edges
x^3yG_{yw}	ordinary loop and double edge
$\frac{y^2}{2}G_{ww}$	two ordinary loops

If we wished a recurrence relation capable of determining all of the numbers $g_{s,d,l}$ starting with the initial condition $g_{0,0,0} = 1$, we would need only extract the coefficient of $x^{s-1}y^dw^l$ from both sides of (1) and set the values equal. This is because every nonempty cubic general graph without triple edges must contain at least one single edge. However, to compute the numbers corresponding to all such graphs on up to $2n$ nodes by way of this recurrence would require $O(n^3)$ arithmetic operations. As we shall see, the number of claw-free cubic graphs is P-recursive as a function of n and can therefore be calculated in $O(n)$ operations. This will require the use of separate equations for G_y, G_w, and each of the second order partial derivatives except for G_{xx}.

To obtain an equation for G_y similar to (1) for G_x, consider a cubic general graph rooted at a double edge. We then remove the double edge and splice the two edges which were adjacent to the root together into a new edge which we designate as the root for the reduced graph. The latter cannot form a nodeless loop, since the original root was not part of a triple edge. However, it can belong to a triple edge, be a single edge, belong to a double edge, or be an ordinary loop. These possibilities give, in order, the four terms on the right side of the next equation:

$$G_y = \frac{x^2y}{2}G + x^2G_x + x^3G_y + \frac{x^2}{2}G_w\ . \tag{2}$$

Finally, a cubic general graph rooted at a loop can be reduced by removing the loop and its adjacent edge. This leaves a vertex of degree 2, which we remove and splice the two incident edges into a new edge. The latter becomes the root of the reduced graph; the root can be a nodeless loop, belong to a triple edge, be a single edge, belong to a double edge, or be a loop. These possibilities correspond in that order to the five terms on the right side of this equation:

$$G_w = \left(xw + \frac{x^3y}{2}\right) G + x^3 G_x + x^4 G_y + xyG_w \ . \tag{3}$$

Finally, the differentiation of (2) and (3) with respect to x, y, and w is straightforward. Making use of the fact that the order of differentiation is immaterial, we obtain the following equations for the second order partial derivatives:

$$\begin{aligned}
&(4)\ \ G_{yw} = \frac{x^2y}{2} G_w + x^2 G_{xw} + x^3 G_{yw} + \frac{x^2}{2} G_{ww} \ ,\\
&(5)\ \ G_{xy} = xyG + \left(2x + \frac{x^2y}{2}\right) G_x + 3x^2 G_y + xG_w + x^2 G_{xx} + x^3 G_{xy} + \frac{x^2}{2} G_{xw} \ ,\\
&(6)\ \ G_{xw} = \left(w + \frac{3x^2y}{2}\right) G + \left(3x^2 + xw + \frac{x^3y}{2}\right) G_x + 4x^3 G_y + yG_w\\
&\qquad\quad + x^3 G_{xx} + x^4 G_{xy} + xyG_{xw} \ ,\\
&(7)\ \ G_{yy} = \frac{x^2}{2} G + \frac{x^2y}{2} G_y + x^2 G_{xy} + x^3 G_{yy} + \frac{x^2}{2} G_{yw} \ ,\\
&(8)\ \ G_{ww} = xG + \left(xw + \frac{x^3y}{2}\right) G_w + x^3 G_{xw} + x^4 G_{yw} + xyG_{ww} \ .
\end{aligned}$$

4. Claw-free cubic graphs. Let $H(z^2)$ be the exponential generating function for counting all labeled claw-free cubic graphs so that

$$H(z) = \sum_{n=0}^{\infty} \frac{H_n z^n}{(2n)!} \ .$$

Our objective is to derive a linear, homogeneous differential equation with coefficients rational in z which is satisfied by $H(z)$. This will imply that the coefficients form a P-recursive sequence, and hence that the n numbers $H_1, \ldots, H_n$ can be calculated in $O(n)$ operations.

The major portion of $H(z)$ is accounted for by the expansion and dilation of the triple-edge-free general cubic graphs counted by $G(x, y, w)$. The strings of diamonds which can reduce to a single edge are counted by

$$b(z) = (1 - z^2/2)^{-1} \ . \tag{9}$$

We leave $b = b(z)$ unexpanded as long as possible in order to simplify our equations. Then to count the graphs resulting from expansion and dilation we simply perform the substitutions

$$\begin{aligned}
x &= zb \ ,\\
y &= \frac{z^2b^2}{2} \ ,\\
w &= \frac{z^3b}{4} \ .
\end{aligned} \tag{10}$$

Note that after substitution of z^2 for z in the formula for w, the very first term is $180\frac{z^6}{6!}$. The exponent of z counts the two vertices of the trumpet horn and the four of the mandatory diamond. Since there are four automorphisms, the coefficient is $\frac{6!}{4} = 180$.

Now $G(z^2)$ counts everything in $H(z^2)$ except for components isomorphic to K_4 or a ring of diamonds, or which reduce to a triangular prism (since that contracts to a triple edge). These are counted, respectively, by $z^2/24$, $-z^2/4 + \ln(\sqrt{b})$, and $z^3b^3/12$. The second of these may require explanation; a ring of m diamonds has $2m2^m$ automorphisms, so the counting series for these components is

$$\sum_{m=2}^{\infty} \frac{z^{2m}}{2m2^m} = -\frac{z^2}{4} - \frac{1}{2}\ln(1 - z^2/2) \ .$$

We then exponentiate to count all graphs consisting entirely of components of these three types. Let $\varphi(z^2)$ be the resulting exponential generating function. Then

$$\varphi(z) = \sqrt{b}\exp\left(-\frac{5z^2}{24} + \frac{z^3b^3}{12}\right) \tag{11}$$

and

$$H(z) = \varphi(z)G(z) \ . \tag{12}$$

The differential equation satisfied by $H(z)$ is now determined by the set of equations (1)–(12). From (11) and (12) we have

$$\begin{aligned} H'(z) = {} & \frac{\varphi'(z)}{\varphi(z)} \cdot \varphi(z)G(z) + x'(z) \cdot \varphi(z)G_x(z) \\ & + y'(z) \cdot \varphi(z)G_y(z) + w'(z) \cdot \varphi(z)G_w(z) \ . \end{aligned} \tag{13}$$

Differentiating again with respect to z we find

$$\begin{aligned} H''(z) = {} & \frac{\varphi''(z)}{\varphi(z)} \cdot \varphi(z)G(z) + 2x'(z)\frac{\varphi'(z)}{\varphi(z)} \cdot \varphi(z)G_x(z) \\ & + 2y'(z)\frac{\varphi'(z)}{\varphi(z)} \cdot \varphi(z)G_y(z) + 2w'(z)\frac{\varphi'(z)}{\varphi(z)} \cdot \varphi(z)G_w(z) \\ & + x''(z) \cdot \varphi(z)G_x(z) + y''(z) \cdot \varphi(z)G_y(z) + w''(z) \cdot \varphi(z)G_w(z) \\ & + x'(z)^2 \cdot \varphi(z)G_{xx}(z) + 2x'(z)y'(z) \cdot \varphi(z)G_{xy}(z) \\ & + 2x'(z)w'(z) \cdot \varphi(z)G_{xw}(z) + y'(z)^2 \cdot \varphi(z)G_{yy}(z) \\ & + 2y'(z)w'(z) \cdot \varphi(z)G_{yw}(z) + w'(z)^2 \cdot \varphi(z)G_{ww}(z) \ . \end{aligned} \tag{14}$$

We now consider (13) and (14) as linear equations in the 12 unknown quantities $H''(z)$, $H'(z)$, $H(z) = \varphi(z)G(z)$, $\varphi(z)G_x(z)$, $\varphi(z)G_y(z)$, $\varphi(z)G_w(z)$, $\varphi(z)G_{xx}(z)$, $\varphi(z)G_{xy}(z)$, $\varphi(z)G_{xw}(z)$, $\varphi(z)G_{yy}(z)$, $\varphi(z)G_{yw}(z)$, and $\varphi(z)G_{ww}(z)$. The coefficients are polynomials in z and b. To see this, note that $b'(z) = zb^2(z)$. Thus all derivatives of x, y, and w can be expressed as polynomials in z and b. Moreover the ratios $\varphi'(z)/\varphi(z)$ and $\varphi''(z)/\varphi(z)$ are also polynomials in z and b. Equations (1)–(8) can all be converted to the same format by applying the substitutions in (10) and multiplying through by $\varphi(z)$. Thus we have 10 linear equations in these 12 unknowns. With the help of the symbolic Gaussian elimination procedure in Maple [3], we can eliminate all of the unknown quantities except for $H(z)$, $H'(z)$, and $H''(z)$. This leads to the linear differential equation

$$
\begin{aligned}
0 &= (144z^8 + 288z^7 - 576z^4)H''(z) \\
&\quad + (-36z^{10} - 96z^9 + 24z^8 + 144z^7 + 576z^6 + 384z^5 \\
&\qquad - 576z^4 - 2880z^3 - 576z^2 + 1152)H'(z) \\
&\quad + (-15z^{11} - 74z^{10} - 130z^9 - 96z^8 + 144z^7 + 368z^6 + 336z^5 - 288z^4 \\
&\qquad - 240z^3 - 288z^2 - 96z)H(z)\ .
\end{aligned} \tag{15}
$$

Here the substitution (9) has been applied to express the coefficients as rational functions of z, common factors have been removed from the three coefficients, and they have been multiplied by a suitable polynomial so that the three coefficients have all become polynomials in z with integer coefficients.

The power series $H(z)$ is the Taylor series about $z = 0$ of the unique solution to (15) which satisfies the initial conditions $H(0) = 1$ and $H^{'}(0) = 0$.

A recurrence relation for the coefficients of $H(z)$ is obtained by extracting the coefficient of $z^n/(2n)!$ in (15); this must be equal to 0. The term $1152H'(z)$ contributes $H_{n+1}/(4n+2)$. This has the maximum index in H, so we solve for H_{n+1} by equating it to $-(2n+1)/576$ times the sum of the other terms. In general, the term contributed by $2^kH(z)$ is $\binom{2n}{2k}(2k)!H_{n-k}$, which upon multiplying by $(2n+1)$ becomes $\binom{2n+1}{2k+1}(2k+1)!H_{n-k}$. For $k \geq 1$, the term contributed by $(2n+1)z^kH'(z)$ is $(n-k+1)\binom{2n+1}{2k+1}(2k+1)!H_{n-k+1}$. Finally, for $k \geq 2$ the term contributed by $(2n+1)z^kH''(z)$ is $(n-k+2)(n-k+1)\binom{2n+1}{2k+1}(2k+1)!H_{n-k+2}$. In this way we find the following relation, which is valid for $n \geq 1$:

$$
\begin{aligned}
H_{n+1} &= (6n-5)\binom{2n+1}{3}H_{n-1} + 60(2n^2-7)\binom{2n+1}{5}H_{n-2} \\
&\quad + 420(12n-31)\binom{2n+1}{7}H_{n-3} - 60480(4n-19)\binom{2n+1}{9}H_{n-4} \\
&\quad - 3326400(6n^2-54n+127)\binom{2n+1}{11}H_{n-5} \\
&\quad - 172972800(9n^2-108n+347)\binom{2n+1}{13}H_{n-6} \\
&\quad - 54486432000(n-1)\binom{2n+1}{15}H_{n-7} \\
&\quad + 59281238016000(n-7)\binom{2n+1}{17}H_{n-8} \\
&\quad + 422378820864000(18n-97)\binom{2n+1}{19}H_{n-9} \\
&\quad + 6563766876226560000\binom{2n+1}{21}H_{n-10} \\
&\quad + 6732296025751296000000\binom{2n+1}{23}H_{n-11}\ .
\end{aligned} \tag{16}
$$

Of course H_{n-j} is zero whenever $j > n$. With the initial conditions $H_0 = 1$ and $H_1 = 0$, (16) can be used to compute the values of $H_2, \ldots, H_{n+1}$ using just $O(n)$ arithmetic operations. In this way we computed the values shown in Table 2.

TABLE 2
Numbers of labeled cubic claw-free graphs.

H_n	n
1	2
60	3
2555	4
466200	5
62791575	6
14536021500	7
8381453705625	8
3284480337138000	9
1942832950684250625	10
2143745512307546647500	11
1743194710893176557891875	12
2022583790860881671548125000	13
3687297941048128552947911484375	14
5250396961636474882113432240187500	15
10270576798318031167485848746426640625	16
28247581137945084450497132391551830500000	17
63409618548369444745423852264233423897890625	18
189787893059957073451746036716319750214365937500	19
739731302424534941124199455315845613980976141796875	20
2436293022465856848407798760164672100623479345846875000	21
10433013033263780019056740194457690414996014419582021484375	22
55053013693844064927863480169144644331902982938883731835937500	23
252448493699621454815261719991354533831171674212674184547416015625	24
1472749695048011678818262827491781703308289147738221578121708593750000	25
10160314924243373000701474995668144304893902876648285295864422890087890625	26

Acknowledgment. The authors would like to thank M. D. Plummer for introducing them to his questions on Hamiltonicity of claw-free cubic graphs at the 1992 Kalamazoo conference. These questions motivated the research reported in the present paper and several subsequent papers.

REFERENCES

[1] G. CHAE, E. M. PALMER, AND R. W. ROBINSON, *Computing the Number of Claw-Free Cubic Graphs with Given Connectivity*, preprint.

[2] G. CHAE, E. M. PALMER, AND R. W. ROBINSON, *Computing the Number of Labeled General Cubic Graphs*, preprint.

[3] B. CHAR, K. O. GEDDES, G. H. GONNET, M. B. MONAGAN, AND S. M. WATT, *Maple Reference Manual*, 5th ed., WATCOM Publications, Waterloo, ON, Canada, 1988.

[4] I. M. GESSEL, *Symmetric functions and P-recursiveness*, J. Combin. Theory Ser. A, 53 (1990), pp. 257–285.

[5] I. P. GOULDEN AND D. M. JACKSON, *Labelled graphs with small vertex degrees and P-recursiveness*, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 60–66.

[6] H. GROPP, *Enumeration of regular graphs* 100 *years ago*, Discrete Math., 101 (1992), pp. 73–85.

[7] F. HARARY AND E. M. PALMER, *Graphical Enumeration*, Academic Press, New York, 1973.

[8] B. D. MCKAY, E. M. PALMER, R. C. READ, AND R. W. ROBINSON, *The asymptotic number of claw-free cubic graphs*, Discrete Math., to appear.

[9] M. D. PLUMMER, *Extending matchings in claw-free graphs*, Discrete Math., 125 (1994), pp. 301–307.

[10] M. D. PLUMMER, 2-*extendability in two classes of claw-free graphs*, in Graph Theory, Combinatorics, and Algorithms, Y. Alavi and A. Schwenk, eds., Wiley, New York, 1995, pp. 905–922.

[11] M. D. PLUMMER, *A note on Hamilton cycles in claw-free graphs*, Congr. Numer., 96 (1993), pp. 113–122.

[12] R. C. Read, *Some unusual enumeration problems*, Ann. New York Acad. Sci., 175 (1970), pp. 314–326.

[13] R. W. Robinson and N. C. Wormald, *Hamilton cycles containing randomly selected edges in random regular graphs*, Random Structures Algorithms, 19 (2001), pp. 128–147.

[14] R. P. Stanley, *Differentiably finite power series*, European J. Combin., 1 (1980), pp. 175–188.

[15] N. C. Wormald, *Triangles in labelled cubic graphs*, in Combinatorial Mathematics, Lecture Notes in Math. 686, Springer Verlag, Berlin, 1978, pp. 337–345.

[16] N. C. Wormald, *Enumeration of labelled graphs* II. *Cubic graphs with a given connectivity*, J. London Math. Soc. (2), 20 (1979), pp. 1–7.

[17] N. C. Wormald, *The number of labelled cubic graphs with no triangles*, Congr. Numer., 33 (1981), pp. 359–378.

SIAM J. Discrete Math.
Vol. 16, No. 1, pp. 74–80

CONSTRUCTIONS OF 3-COLORABLE CORES[*]

KAREN L. COLLINS[†] AND BENJAMIN SHEMMER[†]

Abstract. A finite graph G is a *core* if every endomorphism of G is an automorphism. The only 2-colorable core is K_2. Let G be a core with chromatic number at least 3. We construct an operation on G which yields a 3-colorable subdivision $C(G)$ of G, where $C(G)$ is also a core, and may have arbitrarily large girth. In addition, there is a homomorphism from $C(G)$ to G, in which every vertex of G is covered by two vertices of $C(G)$, and every edge of G is covered by three edges of $C(G)$. Thus, for every core $G \neq K_2$, there is a 3-colorable core $C(G)$ such that (a) $C(G)$ maps homomorphically onto G, (b) $C(G)$ is a topological subdivision of G, and (c) G is a minor of $C(G)$.

Let $\chi_c(G)$ be the circular chromatic number of G. Graphs which are χ_c-critical are cores. We show that G is χ_c-critical if and only if $C(G)$ is χ_c-critical.

Key words. core, 3-colorable, retract, homomorphism, minor, subdivision, circular chromatic number

AMS subject classification. 05C

PII. S0895480101390898

1. Introduction.

The inspiration for graph homomorphisms originally comes from topology. Let G, H be finite, simple graphs, with vertex sets $V(G)$, $V(H)$ and edge sets $E(G), E(H)$. A graph homomorphism from G to H is a map from the vertex set of G to the vertex set of H, say $\phi : V(G) \to V(H)$, such that whenever two vertices u, v of G are connected by an edge in G, then $\phi(u), \phi(v)$ are connected by an edge in H. We often write this as $G \to H$. An excellent beginning survey of graph homomorphisms appears in [4]; see [11] for a more advanced survey.

The *core* of a graph G is defined to be its smallest subgraph H such that $G \to H$. Every finite graph has a unique core up to isomorphism; in this paper we consider only finite graphs. A graph G is said to be a core if the core of G is G. Hence, G is a core if the only subgraph H such that $G \to H$ is G itself. Alternatively, a finite graph is a core if every endomorphism of G is an automorphism.

Another common name for a core is a retract-rigid graph, since a retract in topology is a subspace A of space B for which there is a topological map from B to A that fixes A. Let H be the core of G, where $\phi : G \to H$. Since H is a subgraph of G, there is an inclusion map $i : H \to G$. The restriction of ϕ to H, say ψ, is a map from H to H. Since H is the core of G, it must be its own core, so ψ is an automorphism of H. Thus the map $\phi' = \psi^{(-1)} \circ \phi : G \to H$ is a homomorphism from G to H whose restriction to H is the identity. For the basic properties of cores, see Hell and Nešetřil [5]. Imrich and Klavžar [8] show that the finite product of finitely many triangle-free graphs is a core if and only if every factor is a core. See [2] for results on infinite directed graphs.

There is a strong connection between graph coloring and graph homomorphism. A k-coloring of a graph G is an assignment of colors from $\{1, 2, 3, \ldots, k\}$ to the vertices of G so that whenever two vertices are connected by an edge they receive different colors. Let K_n be the complete graph on n vertices, that is, the graph where every

[*] Received by the editors June 14, 2001; accepted for publication (in revised form) July 24, 2002; published electronically November 22, 2002.

http://www.siam.org/journals/sidma/16-1/39089.html

[†] Department of Mathematics and Computer Science, Wesleyan University, Middletown, CT 06459-0128 (kcollins@wesleyan.edu, bshemmer@wesleyan.edu).

pair of vertices is connected. Then a graph is n-colorable if and only if $G \to K_n$. We can think of homomorphisms to a fixed graph T as a generalization of graph coloring. See, for example, [1, 7, 10, 12, 13, 15]. Let $\chi(G)$ be the chromatic number of G, that is, the smallest integer k such that G has a k-coloring. If $G \to H$, then $\chi(G) \leq \chi(H)$. A graph G is said to be χ-critical if all of the vertex-induced proper subgraphs of G have chromatic number less than $\chi(G)$. Thus, $G \not\to K_{\chi(G)-1}$, but $G - \{v\} \to K_{\chi(G)-1}$ for any vertex v in $V(G)$. Thus, any χ-critical graph is a core. Three simple examples of such cores are the complete graphs, the odd cycles, and the odd wheels. A generalization of the chromatic number is the circular chromatic number, or star chromatic number. Let G_k^t, with $t \leq k/2$, be the abelian Cayley graph on the vertex set $\{0, 1, 2, \ldots, k-1\}$ where i, j are connected if $|i-j| \in \{t, t+1, t+2, \ldots, k-t\}$. Then the circular chromatic number of G, called $\chi_c(G)$, is equal to $\inf\{\frac{k}{t} \mid G \to G_k^t\}$. Bondy and Hell [3] have shown that the infimum is, in fact, a minimum. See also Zhu's alternative definition [16] and his survey [14]. A graph G is said to be χ_c-critical if any of the vertex-induced subgraphs of G have circular chromatic number less than $\chi_c(G)$. Thus, once again, if G is χ_c-critical, then G is a core.

Since a 2-chromatic graph G is a core if and only if $G \cong K_2$, a natural question to ask is, What are the 3-chromatic cores? In this paper, we present a graph operation which transforms any core $G \not\cong K_2$ into a 3-chromatic graph $C(G)$. Moreover, we show that the graph $C(G)$ has the following properties:

1. $C(G)$ is a 3-chromatic core,
2. $C(G)$ maps homomorphically onto G,
3. $C(G)$ is a topological subdivision of G, hence
4. G is a minor of $C(G)$, and
5. G is χ_c-critical if and only if $C(G)$ is χ_c-critical.

The smallest core of G is the natural representative of G in the lattice of equivalence classes of graphs under homomorphism, where G and H are equivalent if $G \to H$ and $H \to G$. These equivalence classes form a lattice under the partial order given by the relation $G < H$ if $G \to H$. Thus, our main result shows that when we are interested in the topological embedding properties of cores in this lattice we need consider only cores with chromatic number less than or equal to 3.

2. The construction. Let $C(G)$ be G with each edge replaced by a path with three edges. Let $D(G)$ be the graph on the same vertex set as G, where $u \sim v$ in $D(G)$ if there is a homomorphism from the path with three edges to G such that one end of the path is vertex u and the other is vertex v; that is, there is a path $u \sim w_1 \sim w_2 \sim v$ in G, where w_2 may equal u and w_1 may equal v, if $u \sim v$ in G. These definitions are a special case of the ones given in the proof of Lemma 1 in Hell and Nešetřil's landmark paper [6].

LEMMA 2.1 (Hell and Nešetřil). *$C(G) \to H$ if and only if $G \to D(H)$.*

In the full lemma, the edges of G may be replaced by any graph I with fixed vertices i and j such that there is an automorphism of I which takes i to j.

Now $C(G)$ preserves cycles of G but multiplies their length by 3. Thus odd cycles remain odd and even cycles remain even. Hence if G is bipartite, then $C(G)$ will also be bipartite. If G is k-chromatic, where $k \geq 3$, then $C(G)$ will be 3-chromatic. We see this by observing that we can color all the vertices that correspond to vertices of G with 1, then color the new vertices, in pairs, with 2 and 3. For each vertex v of G let the corresponding vertex in $C(G)$ be $c(v)$. Label the two vertices between $c(v)$ and $c(u)$ as $c(v,u)$ and $c(u,v)$, where $c(v,u)$ is adjacent to $c(v)$ and $c(u,v)$ and $c(u,v)$ is adjacent to $c(u)$ and $c(v,u)$. We define the natural homomorphism $\phi : C(G) \to G$

by

$$\phi(w) = \begin{cases} u \text{ if } w = c(u), \\ v \text{ if } w = c(v), \\ u \text{ if } w = c(v,u), \\ v \text{ if } w = c(u,v). \end{cases} \tag{2.1}$$

For any two adjacent vertices u, v in G, this map identifies $c(u,v)$ and $c(v)$, and also identifies $c(v,u)$ and $c(u)$. Thus any two adjacent vertices in $C(G)$ have adjacent images under ϕ in G.

LEMMA 2.2. *$G \to H$ if and only if $C(G) \to C(H)$.*

Proof. In the first case, suppose that $G \to H$. Then it is straightforward to show that $C(G) \to C(H)$.

Conversely, suppose that $f : C(G) \to C(H)$. Let u be a vertex in $V(G)$ and define $g : G \to H$ by $g(u) = \phi(f(c(u)))$. We will show that g is a graph homomorphism. Clearly g is a map from the vertices of G to the vertices of H. Therefore we want to show that if $u \sim v$ in G, then $g(u) \sim g(v)$ in H. Now f and ϕ are both graph homomorphisms, but $c(u) \not\sim c(v)$ in $C(G)$; instead in $C(G)$ we have the path $c(u) \sim c(u,v) \sim c(v,u) \sim c(v)$. Consider what happens when f acts on this path.

Case 1. f does not act injectively on the path. Then f cannot identify $c(u)$ and $c(v)$, because $C(H)$ contains no triangles. Therefore any identification done by f causes $f(c(u))$ and $f(c(v))$ to be adjacent. Since ϕ is a homomorphism, $\phi(f(c(u))) \sim \phi(f(c(v)))$.

Case 2. f acts injectively on the path so that the image under f of this path with three edges is a path with three edges in $C(H)$: namely, $f(c(u)) \sim f(c(u,v)) \sim f(c(v,u)) \sim f(c(v))$. Now any path of length 3 in $C(H)$ must contain at least one vertex corresponding to a vertex in H and can contain at most two such vertices.

If the path in $C(H)$ contains two vertices corresponding to vertices in H, they must be the two end vertices, $f(c(u))$ and $f(c(v))$, because the distance between any two such vertices is at least 3 in $C(H)$. Thus, ϕ identifies $f(c(u))$ and $f(c(v,u))$; since $f(c(v,u))$ is adjacent to $f(c(v))$, and ϕ is a homomorphism, we have that $\phi(f(c(u)))$ is adjacent to $\phi(f(c(v)))$.

If only one vertex in the path corresponds to a vertex w in H, then w corresponds to either $f(c(u,v))$ or $f(c(v,u))$. Without loss of generality, suppose $w = f(c(u,v)$. Then ϕ identifies $f(c(u,v))$ and $f(c(v))$, and since $f(c(u,v))$ is adjacent to $f(c(u))$, we have that $\phi(f(c(u)))$ is adjacent to $\phi(f(c(v)))$. □

We point out that Lemmas 2.1 and 2.2 are not the same statement in disguise. In Lemma 2.1, the graphs H and $D(H)$ have the same number of vertices, but, in Lemma 2.2, $C(H)$ has three times as many vertices as H.

THEOREM 2.3. *Let G be a connected graph on three or more vertices. Then G is a core if and only if $C(G)$ is a core.*

Proof. Suppose first that $C(G)$ is a core. If $G \to X$, where X is a proper subgraph of G, then, by Lemma 2.2, $C(G) \to C(X)$ and clearly $C(X)$ is a proper subgraph of $C(G)$. This contradicts the fact that $C(G)$ is a core, so G must be a core.

Conversely, suppose that G is a core. Let $f : C(G) \to Y$, where Y is a proper subgraph of $C(G)$. We may assume that f fixes Y in $C(G)$ and that Y is the core of $C(G)$. Let X be the induced graph in G on $\{u \in G \mid f(c(u)) \in Y\}$. We show that X must be a proper subgraph of G, and that $C(G) \to C(X)$, hence $G \to X$ by Lemma 2.2.

First, if Y is a proper subgraph of $C(G)$, then Y must be missing some vertex $w \in C(G)$. If $w = c(u)$ for some $u \in G$, then X is also missing u, so X is a proper subgraph of G. Suppose that X contains $c(u)$ for all $u \in G$ so that f fixes $c(u)$ for all $u \in G$. Now, for any $u \sim v$, f must act injectively on the path $c(u) \sim c(u,v) \sim c(v,u) \sim c(v)$, because any identification will mean that $c(u) \sim c(v)$, and these are not adjacent in Y. Since G is simple, there is only one path from $c(u)$ to $c(v)$ with three edges in $C(G)$, and hence in Y, so f fixes $c(u,v)$ and $c(v,u)$ as well as $c(u)$ and $c(v)$. Thus if Y contains $c(v)$ for every v in G, then $Y \cong C(G)$.

Next we show that $C(G) \to C(X)$, hence $G \to X$. We have seen that if $u \sim v$ in G and Y contains both $c(u)$ and $c(v)$, then Y contains $c(u,v)$ and $c(v,u)$ as well. If, in contrast, for $u \sim v$ in G, Y contains $c(u)$ but not $c(v)$, then $c(v,u)$ can be identified with $c(u)$, and $c(u,v)$ can be identified with any neighbor of $c(u)$. If $c(u)$ has no neighbors in Y, then Y is not connected, but if G is connected, then $C(G)$ is connected, so its core Y must also be connected. Thus, since Y is the smallest subgraph that $C(G)$ maps to, Y must be $C(X)$. □

It is worthwhile to observe that the same proofs for Lemma 2.2 and Theorem 2.3 will work if all edges of G are replaced by paths of any fixed odd length.

THEOREM 2.4. *Let G be a connected graph, and let $C_i(G)$ be G with each edge replaced by a path with $2i+1$ edges. Then G is a core if and only if $C_i(G)$ is a core.*

3. χ_c-criticality. In this section we show that if G is χ_c-critical, then $C(G)$ is χ_c-critical. It is not true that if G is χ-critical, then $C(G)$ is also χ-critical. Take, for example, the 5-wheel, W_5. We know $C(W_5)$ is 3-colorable and contains, but does not equal, an odd cycle, which are the only 3-critical graphs. Thus $C(G)$ is χ-critical only if G is an odd cycle.

Recall that G_k^t, with $t \leq k/2$, is the abelian Cayley graph on the vertex set $\{0,1,2,\ldots,k-1\}$ where i,j are connected if $|i-j| \in \{t,t+1,t+2,\ldots,k-t\}$. The circular chromatic number of G, $\chi_c(G)$, is equal to $\min\{\frac{k}{t} \mid G \to G_k^t\}$. It is a well-known fact that $\chi(G)-1 < \chi_c(G) \leq \chi(G)$. Thus, if $G \ncong K_2$ is a core, then $2 < \chi_c(C(G)) \leq 3$.

LEMMA 3.1. *Let $3t > k$. Then $D(G_k^t) \cong G_k^{3t-k}$.*

Proof. Let the vertices of $D(G_k^t)$ be $\{0,1,2,\ldots,(k-1)\}$. Suppose that $i \sim j$ in $D(G_k^t)$; this is equivalent to the fact that there exists a path $i \sim w_1 \sim w_2 \sim j$ in G_k^t. For fixed i, the set of vertices j such that j can be reached from i by a path with three edges is the set $\{i+3t, i+3t+1, i+3t+2, \ldots, i+3k-3t\}$, reduced modulo k. Without loss of generality, since G_k^t is vertex-transitive, we may choose i to be 0. Now $3t > k > 2t$, hence $2k > 3t > k$, so $0 < 3t-k < k$ and $0 < 2k-3t < k$. Thus the neighbors of 0 in $D(G_k^t)$ are $\{3t-k, 3t-k+1, \ldots, 2k-3t\}$; hence $D(G_k^t) \cong G_k^{3t-k}$. □

THEOREM 3.2. *Let $3t > k$. Then $C(G) \to G_k^t$ if and only if $G \to G_k^{3t-k}$.*

Proof. Apply Lemmas 2.1 and 3.1. □

COROLLARY 3.3. *Let $3t > k$. Then $\chi_c(C(G)) = \frac{k}{t}$ if and only if $\chi_c(G) = \frac{k}{(3t-k)}$.*

Proof. The proof follows immediately from Theorem 3.2 and the definition of the circular chromatic number. □

THEOREM 3.4. *G is χ_c-critical if and only if $C(G)$ is χ_c-critical.*

Proof. First, in order to apply Corollary 3.3, we prove that $\chi_c(C(G)) < 3$ for any G. Let $\chi_c(G) = \frac{a}{b}$, hence $G \to G_b^a$. It is easy to check that $G_b^a \to G_{3b}^{3a}$ by the map i in G_b^a goes to $3i$ in G_{3b}^{3a}. Now, by Theorem 3.2, since $3(a+b) > 3b$, $C(G) \to G_{3b}^{a+b}$. Hence $\chi_c(C(G)) \leq \frac{3b}{a+b} < 3$.

Second, the following numerical fact follows from algebraic manipulation of the ratios. Let a,b,c,d be positive integers. Then

$$\frac{a}{b} < \frac{c}{d} \text{ if and only if } \frac{a}{3b-a} < \frac{c}{3d-c} .$$

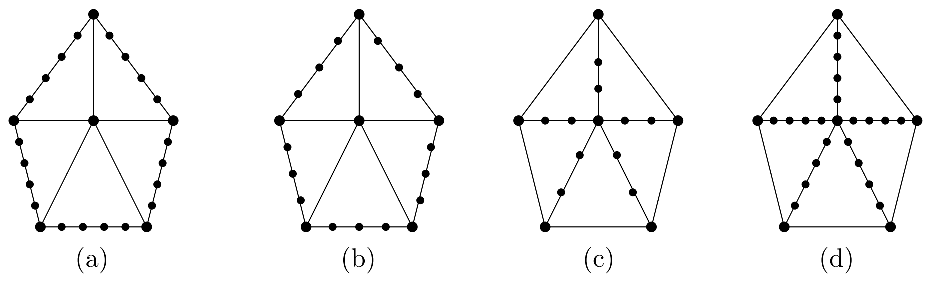


FIG. 4.1.

The theorem follows from Corollary 3.3 and the observation that if e is an edge in G and f_1, f_2, f_3 are the edges in $C(G)$ that replaced e, then

$$C(G) - f_i* \longleftrightarrow C(G-e) \text{ for } i = 1, 2, 3. \qquad \square$$

These results generalize to the case when all edges of G are replaced by paths of any fixed odd length, say $2i+1$. Then $\chi_c(C_i(G)) = \frac{k}{t}$ if and only if $\chi_c(G) = \frac{k}{((2i+1)t-ik)}$.

COROLLARY 3.5. *Let G be a connected graph, and let $C_i(G)$ be G with each edge replaced by a path with $2i+1$ edges. Then $C_i(G)$ is χ_c-critical if and only if G is χ_c-critical.*

4. Contrasting constructions. Our construction $C(G)$ is not the only way to create a subdivision of a graph which is a core. In $C(G)$, we replace every edge by a path of fixed odd length. In Theorem 4.1 we show that we can create a core by replacing all the edges in the outside edge orbit of an odd wheel by a path of odd length. If these edges are replaced by paths of varying lengths, or even lengths, the result is not a core. Also, if we replace the spoke edges of the odd wheel by a path of odd length, we sometimes get a core, and sometimes we do not. In particular, Figure 4.1(a) is a core, but Figure 4.1(b) is bipartite and hence not a core; Figure 4.1(c) is a core, but Figure 4.1(d) is not a core.

Define $W(s,m)$ as a cycle C with $s\,(1+m)$ vertices, labeled as $\{1, 2, 3, \ldots, s\,(1+m)\}$, plus an extra center vertex v which is connected to s evenly spaced vertices on C, say $\{1, 2+m, 1+2(1+m), \ldots, 1+(s-1)(1+m)\}$. Thus $W(s,m)$ has $s\,(1+m)+1$ vertices and $s\,(2+m)$ edges. The large cycle is odd when s is odd and m is even; the small cycles from the center vertex to consecutive neighbors on C are odd when m is even and even when m is odd. Figure 4.1(a) shows $W(5,4)$ and Figure 4.1(b) is $W(5,3)$.

THEOREM 4.1. *$W(s,m)$ is a core if and only if s is odd and m is even. Further, any graph G which is a cycle C plus an extra vertex v is a core if and only if $G \cong W(2k+1, 2j)$ for some positive integers k and j.*

Proof. Let $G = W(s,m)$. Suppose that m is odd. Then C is even, and the small cycles are even, so G is bipartite, and not a core. Suppose that m is even and s is even. Then C is even and the small cycles are odd. In this case, G maps to one of the small cycles. Then the set $\{v, 1, 2, 3, \ldots, 1+m, 1+(1+m)\}$ is a small cycle. We map the neighbors of v to 1 and 1+(1+m), and let the small cycles follow in the natural way. Send $1+j(m+1)$ to 1 if j is even and to $1+(1+m)$ if j is odd for $2 \le j \le s-1$.

Suppose that m is even and s is odd. Then C is odd, and the small cycles are odd. The girth of G is the same as the size of a small cycle. In any map of G to itself, the center vertex v cannot be identified with any vertex on the cycle, because the resulting graph has smaller girth than G. Thus the vertices in C must map within C, but C is an odd cycle and hence a core itself. Therefore G is a core.

Now suppose that G is a cycle C plus an extra vertex v. Let the neighbors of v in consecutive order relative to C be $w_1, w_2, \dots, w_t$. Then $w_2 \sim v \sim w_1$ and the path from w_1 to w_2 is a cycle. If this cycle is even, we can map it to the path $w_1 \sim v \sim w_2$. Therefore, assume that all small cycles in G are odd. Suppose that the small cycle formed by v and w_1, w_2 and the path from w_1 to w_2 is a smallest cycle of G such that the adjacent small cycle composed of $w_3 \sim v \sim w_2$ and the path from w_2 to w_3 is not a smallest cycle of G. Since both cycles are odd, the larger one has at least two more vertices than the smaller one. Let the vertices in the path from w_1 to w_2, excluding w_1, w_2, be $x_1 \sim x_2 \sim \cdots \sim x_m$ and let the vertices in the path from w_2 to w_3, excluding w_2, w_3, be $y_1 \sim y_2 \sim \cdots \sim y_n$. We fix w_3 and map y_n to the center vertex v, y_{n-1} to w_1, and the rest of the path from y_{n-2} to y_1 onto the path from x_1 to x_m, ending with y_1 maps to x_m. Since n is odd and $n \geq m + 2$, the path from y_{n-2} to y_1 has the same parity and at least as many vertices as the path from x_1 to x_m. Since x_m is adjacent to w_2, so is the image of y_1. Thus G maps to a subgraph of itself and is not a core. □

Note that it matters which edges we choose to replace by paths. If we replace only the spoke edges of W_5 by paths of length 2, shown in Figure 4.1(c), the resulting graph is a core, because the outside 5-cycle is the unique smallest cycle of the graph and hence remains fixed, and the center vertex cannot be identified with any vertex in that cycle without making a triangle. If we replace the spoke edges of W_5 by paths of length 4 (Figure 4.1(d)), however, the resulting graph maps to the 5-cycle by sending the center vertex to a vertex in the outside cycle and wrapping all the resulting 5-cycles around the outside 5-cycle. Thus this graph is not a core.

REFERENCES

[1] 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.

[2] B. L. Bauslaugh, *Cores and compactness of infinite directed graphs*, J. Combin. Theory Ser. B, 68 (1996), pp. 255–276.

[3] J. A. Bondy and P. Hell, *A note on the star chromatic number*, J. Graph Theory, 14 (1990), pp. 479–482.

[4] G. Hahn and C. Tardif, *Graph homomorphisms: Structure and symmetry*, in Graph Symmetry, NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci. 497, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997, pp. 107–166.

[5] P. Hell and J. Nešetřil, *The core of a graph*, Discrete Math., 109 (1992), pp. 117–126.

[6] P. Hell and J. Nešetřil, *On the complexity of H-coloring*, J. Combin. Theory Ser. B, 48 (1990), pp. 92–110.

[7] P. Hell, J. Nešetřil, and X. Zhu, *Duality of graph homomorphisms*, in Combinatorics, Paul Erdös is Eighty, Vol. 2, Bolyai Soc. Math. Stud. 2, János Bolyai Math. Soc., Budapest, 1996, pp. 271–282.

[8] W. Imrich and S. Klavžar, *Retracts of strong products of graphs*, Discrete Math., 109 (1992), pp. 147–154.

[9] J. Kratochvíl and P. Mihók, *Hom-properties are uniquely factorizable into irreducible factors*, Discrete Math., 213 (2000), pp. 189–194.

[10] D. D. Liu, *T-graphs and the channel assignment problem*, Discrete Math., 161 (1996), pp. 197–205.

[11] J. Nešetřil, *Graph homomorphisms and their structure*, in Graph Theory, Combinatorics, and Algorithms, Vol. 1, 2, Wiley-Interscience, New York, 1995, pp. 825–832.

[12] J. Nešetřil and A. Raspaud, *Colored homomorphisms of colored mixed graphs*, J. Combin. Theory Ser. B, 80 (2000), pp. 147–155.

[13] J. Nešetřil, A. Raspaud, and E. Sopena, *Colorings and girth of oriented planar graphs*,

Discrete Math., 165/166 (1997), pp. 519–530.

[14] X. Zhu, *Circular chromatic number: A survey. Combinatorics, graph theory, algorithms and applications*, Discrete Math., 229 (2001), pp. 371–410.

[15] X. Zhu, *Uniquely H-colorable graphs with large girth*, J. Graph Theory, 23 (1996), pp. 33–41.

[16] X. Zhu, *Star chromatic numbers and products of graphs*, J. Graph Theory, 16 (1992), pp. 557–569.

SIAM J. Discrete Math.
Vol. 16, No. 1, pp. 81–98

THE SPECIAL FUNCTION FIELD SIEVE*

OLIVER SCHIROKAUER†

Abstract. Let p be a prime number and n a positive integer, and let $q = p^n$. Adleman and Huang [*Inform. and Comput.,* 151 (1999), pp. 5–16] have described a version of the function field sieve which is conjectured to compute a logarithm in the field of q elements in expected time $L_q[1/3; (32/9)^{1/3} + o(1)]$, where $L_q[s; c] = \exp(c(\log q)^s(\log\log q)^{1-s})$ and the $o(1)$ is for $q \to \infty$ under the constraint that $p^6 \leq n$. In this paper, we present a modification of their method which runs conjecturally in expected time $L_q[1/3; (32/9)^{1/3} + o(1)]$ so long as $q \to \infty$ with $p \leq n^{o(\sqrt{n})}$. The technique we use can also be applied to the special number field sieve and results in an algorithm which, in expected time $L_p[1/3; (32/9)^{1/3} + o(1)]$, is conjectured to compute a logarithm in a prime field whose cardinality p is of the form $r^e - s$, with r and s small in absolute value.

Key words. finite field, discrete logarithm, function field sieve

AMS subject classifications. 11Y40, 11Y16, 14H05

PII. S0895480100372668

1. Introduction.

Let $q = p^n$, where p is prime and n is a positive integer. Let $\mathbb{F}_q$ denote the field of q elements, and let $\mathbb{F}_q^*$ denote the multiplicative group of $\mathbb{F}_q$. The discrete logarithm problem in $\mathbb{F}_q$ is to compute, upon input of $t, u \in \mathbb{F}_q^*$ with $u \in \langle t \rangle$, the least nonnegative integer x such that $t^x = u$. In this case, we write $x = \log_t u$. In addition to being of intrinsic interest as a fundamental computational problem, the discrete logarithm problem has become important in cryptography as various schemes depend on its intractability for their security. In recent years, the methods used to compute discrete logarithms have improved significantly, both in theory and practice. Some of these developments are discussed in [18] and [23]. Particularly noteworthy is the application of the number field sieve to the discrete logarithm problem [7], [8], [20], [21], [22], [29], [30] and the development of the analogous function field sieve [1], [2], [9], [22], [28], whose asymptotic complexity is the subject of the present paper.

Adleman and Huang [2] describe a version of the function field sieve algorithm for computing discrete logarithms in finite fields which is conjectured to run in expected time

$$L_q[1/3; (32/9)^{1/3} + o(1)], \tag{1.1}$$

where q is the cardinality of the field, $L_q[s; c] = \exp(c(\log q)^s(\log\log q)^{1-s})$, and the $o(1)$ is for $q \to \infty$ subject to the restriction that $p \leq n^{1/6}$. This running time is an improvement over the time $L_q[1/3; (64/9)^{1/3} + o(1)]$ conjectured for Adleman's original function field sieve [1]. The reader who is familiar with the history of the number field sieve will recognize (1.1) as the running time of the special number field sieve factoring algorithm designed to factor integers of the form $r^e - s$, where r and s are small in absolute value [11]. The appearance here of the same quantity is not a coincidence. Following Coppersmith's lead in [6], Adleman and Huang represent the finite field $\mathbb{F}_q$ as the quotient of $\mathbb{F}_p[X]$ by an ideal generated by an element of the form $r^e - s$, where r and s are polynomials of small degree. As a result they are

*Received by the editors May 24, 2000; accepted for publication (in revised form) June 6, 2002; published electronically November 22, 2002.

http://www.siam.org/journals/sidma/16-1/37266.html

†Department of Mathematics, Oberlin College, Oberlin, OH 44074 (oliver.schirokauer@oberlin.edu).

able to make use of a "small" extension of $\mathbb{F}_p(X)$ in much the same way that the special number field sieve takes advantage of a "small" number field. Because of this similarity, we refer to their version of the function field sieve as the special function field sieve.

The small extension of $\mathbb{F}_p(X)$ which is at the heart of the special function field sieve is only well-suited to the computation of the logarithm of an element which is small in the sense that it is represented by a polynomial in $\mathbb{F}_p[X]$ of small degree. To compute the logarithm of a general element in time (1.1), Adleman and Huang employ a reduction algorithm which produces a representation of this element as a product of powers of small elements. It is in the design of this reduction, however, that they encounter the obstruction which forces them to restrict the values of q for which (1.1) is valid. The contribution we make in this paper is an improved reduction technique. With it, the special function field sieve conjecturally runs in expected time (1.1) for $q \to \infty$ with

$$p \le n^{o(\sqrt{n})}. \tag{1.2}$$

This constraint on q is only slightly stronger than what is necessary to obtain the primary constant of 1/3 in (1.1).

The reduction method we provide can also be used in conjunction with the special number field sieve. In [7], Gordon describes a version of the special number field sieve for discrete logarithms which is of practical interest but is asymptotically slower than the general number field sieve because of the lack of a sufficiently fast reduction method. With the technique of this paper, however, the special number field sieve computes a discrete logarithm in a prime field of size $p = r^e - s$ in expected time $L_p[1/3; (32/9)^{1/3} + o(1)]$ for $p \to \infty$ subject to the restriction that $|r|$ and $|s|$ do not grow too quickly. In particular, this running time is valid for $|r|$ and $|s|$ bounded. We note that Semaev has provided a description of a special number field sieve discrete logarithm algorithm which achieves the same conjectural running time as ours [24]. Presumably, his ideas can be used to design an alternative special function field sieve as well.

We begin in the next section with the description of our reduction algorithm. We then analyze its running time in section 3. Finally, in section 4, we show how to compute the logarithms of the small elements in $\mathbb{F}_q$ and provide a running time analysis of the entire special function field sieve, not only under the assumption that (1.2) holds but for $q \to \infty$ in general. Since descriptions of the special number field sieve, both for factoring and discrete logarithms, can be found elsewhere [7], [11], [12], [22], [30], and since we anticipate that the reader will have little trouble translating the methods of this paper into the number field setting, we do not do so here.

We do not expect our reduction technique to be of practical significance at this time, and in what follows we do not consider issues of implementation. However, it should be noted that both the number field sieve for discrete logarithms and the function field sieve are practical. Indeed, the number field sieve has been used to compute logarithms in the field whose cardinality is the special 129-digit prime $(739 \cdot 7^{149} - 736)/3$ [30], as well as in a field whose cardinality is a "general" prime of 120 digits [8]. The function field sieve, in a special form somewhat different from that in [2], has been used to compute logarithms in $\mathbb{F}_{2^{521}}$ [9], and in the form of Coppersmith's algorithm, has succeeded in computing logarithms in $\mathbb{F}_{2^{607}}$ [28]. All of these implementations, regardless of the algorithm, compute the logarithms of small elements first and then employ a reduction step to find the logarithm of an arbitrary element. We refer the reader to [22] for a survey of the various reduction techniques in use and note that these methods work well for fields which are currently tractable.

2. Reduction method. Let p be a prime and n a positive integer, and let $q = p^n$. Following Adleman and Huang [2], let $s \in \mathbb{F}_p[X]$ be a polynomial of small degree such that $X^n - s$ is irreducible and such that at least one of the roots of s is nonzero and simple. We adopt as a model for the finite field $\mathbb{F}_q$ the set $\{f \in \mathbb{F}_p[X] \mid \deg f < n\}$ with addition and multiplication taken modulo $X^n - s$. Since we think of the elements in $\mathbb{F}_q$ as polynomials, we do not hesitate to speak of the degree of an element in $\mathbb{F}_q$. Our goal in this section is to describe an algorithm which reduces the computation of a general logarithm in $\mathbb{F}_q$ to the computation of the logarithms of many polynomials in $\mathbb{F}_q$ of small degree. In what follows, we assume that $n > 1$.

Our reduction algorithm proceeds in stages indexed by a variable j. For $j \geq 0$, let

$$a_j = \frac{1}{3} - \frac{1}{3 \cdot 2^j},$$

and for $j > 0$, let B_j be the positive integer closest to $M_q[2/3 - a_{j-1}; 3/4]$, where

$$M_q[s; c] = \log_p(L_q[s; c]) = cn^s \left(\frac{\log\log q}{\log p}\right)^{1-s}.$$

In addition, let B be the positive integer closest to $M_q[1/3; (4/9)^{1/3}]$. For any constant β, let S_β denote the set of elements in $\mathbb{F}_q$ which are monic and of degree $\leq \beta$ and which, when considered as elements in $\mathbb{F}_p[X]$, are irreducible.

ALGORITHM 2.1. Let j be a fixed, positive integer. This algorithm takes as input a set W of elements in S_{B_j} and three parameters E_1, E_2, and E_3. It is designed to output, for each $w \in W$, a multiplicative relation of the form

$$w = \eta \prod y_i^{c_i}, \tag{2.1}$$

where η is in $\mathbb{F}_p^*$, the y_i are in $S_{B_{j+1}}$, and the c_i are integers.

Let e be the integer closest to

$$\left(\frac{7}{8}\right)\left(\frac{\log q}{\log\log q}\right)^{a_j}, \tag{2.2}$$

let $\mu = \lceil n/e \rceil$, and let

$$H(X, Y) = Y^e - X^{\mu e - n}s.$$

By Eisenstein's criterion and our assumption that s has a root of multiplicity one, $H(X, Y)$ is absolutely irreducible. Let

$$\mathcal{O} = \mathbb{F}_p[X, Y]/(H(X, Y)),$$

denote by y the image of Y in $\mathcal{O}$, and let F be the field of fractions of $\mathcal{O}$. Observe that $H(X, X^\mu) \equiv 0 \bmod X^n - s$. We set $m = X^\mu$ and let $\phi : \mathcal{O} \to \mathbb{F}_q$ be the ring homomorphism which is the canonical surjection on $\mathbb{F}_p[X]$ and which sends $y \mapsto m$. For an element $\gamma \in F$, let $N(\gamma)$ be the norm of γ in $\mathbb{F}_p(X)$. Finally, recall that an element in $\mathbb{F}_p[X]$ is said to be β-smooth if it factors into a product of polynomials of degree $\leq \beta$ and that an element in F with norm in $\mathbb{F}_p[X]$ is said to be β-smooth if its norm is β-smooth.

Step 1. Let C be the positive integer closest to $M_q[1/3; 1/2]$. Let T be the union of the set of places of F at infinity and the set of places Q of F for which there exists a place P of $\mathbb{F}_p(X)$ of degree $\leq C$ such that Q lies over P. Use sieving techniques to find all pairs of relatively prime polynomials $c, d \in \mathbb{F}_p[X]$ such that $c - dy$ is C-smooth, $c - dm$ is B_{j+1}-smooth, and the degrees of c and d are less than E_1. If the number of pairs found is less than $|T|$, the algorithm terminates.

Step 2. For $Q \in T$, let v_Q denote the discrete valuation associated to Q. For each of the elements $c - dy$ found in Step 1, construct a valuation vector $V_{c,d}$ of length $|T|$ containing the values $v_Q(c - dy)$, with Q ranging over all the places in T. Since $c - dy$ has no zeros or poles outside of T, the vector $V_{c,d}$ is simply a representation of the divisor $\operatorname{div}(c - dy)$.

Step 3. Let $L = \mathbb{F}_p[X] \times \mathbb{F}_p[X]$ and define the length of a vector $v = (v_1, v_2) \in L$ by the equation $|v| = \max\{\deg(v_1), \deg(v_2)\}$. For each $w \in W$, proceed as follows.

(i) Let

$$L' = \{(v_1, v_2) \in L \mid v_1 - v_2 m \equiv 0 \bmod w\},$$

and note that L' contains the vectors $(w, 0)$ and $(m, 1)$. Apply the extended Euclidean algorithm to w and m. As shown in [17], doing so produces a sequence

$$(\alpha_1, \beta_1), \ldots, (\alpha_N, \beta_N)$$

of pairs of polynomials in $\mathbb{F}_p[X]$ with the property that for every pair of integers ψ and ζ satisfying $\psi + \zeta = \deg(w) - 1$ there exists a pair (α_i, β_i) such that

$$\deg(\beta_i) \leq \psi,$$
$$\deg(\alpha_i w + \beta_i m) \leq \zeta.$$

It follows that the vector $\alpha_i(w, 0) + \beta_i(m, 1)$ has length at most $\max\{\psi, \zeta\}$. Setting $\psi = \lfloor \deg(w)/2 \rfloor$, we then see that the extended Euclidean algorithm produces a vector $\theta = (\theta_1, \theta_2) \in L'$ of length at most $\deg(w)/2$. Setting $\psi = |\theta| - 1$, we find that it also yields a vector $\tau = (\tau_1, \tau_2) \in L'$ such that $|\tau| \leq \deg(w) - |\theta|$. Since the pairs (α_i, β_i) produced by the extended Euclidean algorithm are pairwise linearly independent over $\mathbb{F}_p[X]$ [17], the vectors θ and τ are also linearly independent.

We proceed to use a sieve to look for a pair of polynomials r and s such that

$$\frac{r\theta_1 + s\tau_1 - (r\theta_2 + s\tau_2)m}{w}$$

and

$$r\theta_1 + s\tau_1 - (r\theta_2 + s\tau_2)y$$

are both B_{j+1}-smooth. We consider two cases. If $E_2 \leq |\tau| - |\theta|$, let $s = 1$ and test all polynomials r of degree less than E_2. Otherwise, let δ and ϵ be integers such that

$$\delta + \epsilon = E_2,$$
$$\Big|(\delta + |\tau|) - (\epsilon + |\theta|)\Big| \leq 1,$$

and test pairs r, s such that $\deg(r) < \delta$ and $\deg(s) < \epsilon$. If no pairs satisfying the smoothness conditions are found, the algorithm terminates. Otherwise, fix a pair r, s that does satisfy the conditions, let $a = r\theta_1 + s\tau_1$ and $b = r\theta_2 + s\tau_2$, and write

$$N(a - by) = \prod_{i=1}^{l} g_i^{\epsilon_i},$$

where the g_i are distinct irreducible polynomials in $\mathbb{F}_p[X]$ of degree $\leq B_{j+1}$.

(ii) For each irreducible factor g_i of $N(a - by)$, let $\mathfrak{q}_i$ be the ideal in $\mathcal{O}$ generated by g_i and the element $a - by$, and let

$$L_i = \{(v_1, v_2) \in L \,\big|\, v_1 - v_2 y \equiv 0 \bmod \mathfrak{q}_i\}.$$

Then L_i contains the vectors $(g_i, 0)$ and $(a\bar{b}, 1)\}$, where $\bar{b}$ is the inverse of b modulo g_i. Apply the extended Euclidean algorithm to the pair $g_i, a\bar{b}$ to find two vectors $\theta_i = (\theta_{i,1}, \theta_{i,2})$ and $\tau_i = (\tau_{i,1}, \tau_{i,2})$, which are linearly independent over $\mathbb{F}_p[X]$ and such that

$$|\theta_i| \leq \frac{\deg(g_i)}{2},$$
$$|\tau_i| \leq \deg(g_i) - |\theta_i|.$$

Next sieve for r_i and s_i such that

$$r_i\theta_{i,1} + s_i\tau_{i,1} - (r_i\theta_{i,2} + s_i\tau_{i,2})m$$

is B_{j+1}-smooth and

$$\frac{N(r_i\theta_{i,1} + s_i\tau_{i,1} - (r_i\theta_{i,2} + s_i\tau_{i,2})y)}{g_i}$$

is C-smooth. Choose which elements to test in the same way as was done in substep (i), with E_2 replaced by E_3. The algorithm terminates if no pairs are found. Otherwise, let r_i and s_i be a pair satisfying the required smoothness conditions, and let $a_i = r_i\theta_{i,1} + s_i\tau_{i,1}$ and $b_i = r_i\theta_{i,2} + s_i\tau_{i,2}$.

(iii) Let

$$\sigma = \frac{\prod_{i=1}^{l}(a_i - b_i y)^{\epsilon_i}}{a - by}$$

and observe that σ is C-smooth. Construct a valuation vector V_σ of the same sort described in Step 2, containing the values $v_Q(\sigma)$ for all places Q in T.

(iv) Use the algorithm of [31] to solve the matrix congruence

$$Ax \equiv -V_\sigma \bmod (q-1)/(p-1), \tag{2.3}$$

where A is given as follows. If w is the first member of W input into Step 3, let A be a square matrix whose columns are chosen from among the vectors $V_{c,d}$. Otherwise, let A be the matrix used to solve (2.3) in the previous run of Step 3. If no solution to (2.3) is found for the w at hand and there exists a vector from Step 2 which does not appear as a column in A, enlarge A by including an unused vector as an additional column and try to solve (2.3) again. If no vectors $V_{c,d}$ remain, the algorithm terminates.

In the case that (2.3) is solved, observe that the coordinates of a solution can be indexed by the pairs (c,d) used to form the column vectors $V_{c,d}$, and let $(\ldots, x_{c,d}, \ldots)$ be one such solution. Then for each place Q of F, we have

$$v_Q\left(\sigma \prod (c-dy)^{x_{c,d}}\right) \equiv 0 \bmod (q-1)/(p-1).$$

Equivalently, there is a divisor D in the degree 0 part of the divisor group of F such that

$$\operatorname{div}\left(\sigma \prod (c-dy)^{x_{c,d}}\right) = \left(\frac{q-1}{p-1}\right) D. \tag{2.4}$$

Let h be the class number of F and assume that h is prime to $(q-1)/(p-1)$. By (2.4), the order of D in the class group of F divides $(q-1)/(p-1)$. We conclude that this order is 1 and that D is principal. Thus there exists an element $\lambda \in F$ and a constant $\eta' \in \mathbb{F}_p^*$ such that

$$\sigma \prod (c-dy)^{x_{c,d}} = \eta' \lambda^{\frac{q-1}{p-1}}. \tag{2.5}$$

As explained in [2], the map ϕ can be extended to the localization of $\mathcal{O}$ at the kernel of ϕ. Since the place associated to this kernel is not in the set T, we can, in particular, apply ϕ to both sides of (2.5). We conclude that

$$\frac{\prod_{i=1}^{l}(a_i-b_im)^{\epsilon_i} \prod (c-dm)^{x_{c,d}}}{w[(a-bm)/w]\eta'} \tag{2.6}$$

is a $(q-1)/(p-1)$th power in $\mathbb{F}_q^*$ and hence is an element in $\mathbb{F}_p^*$. Since $(a-bm)/w$, each $a_i - b_i m$, and each $c - dm$ is B_{j+1}-smooth, we have succeeded in expressing w as a product of an element in $\mathbb{F}_p^*$ and numerous polynomials in $S_{B_{j+1}}$. This completes the description of Algorithm 2.1.

In the next section, we choose values for E_1, E_2, and E_3 so that the sieving in Step 1 is likely to produce a matrix A of full rank and the sieving in Step 3 is likely to yield the desired elements. If the algorithm terminates because (2.3) has no solution or the sieving in Step 3 fails, one obvious modification is to increase the size of these parameters. However, if (2.3) has no solution, it may also be the case that the class number h has a factor in common with $(q-1)/(p-1)$. Moreover, if h is not prime to $(q-1)/(p-1)$, the algorithm could run to the end but output an incorrect answer because (2.6) is not in $\mathbb{F}_p^*$. One response to a suspicion that the class number is obstructing the algorithm is to change the field F. This can be accomplished by using a different representation of $\mathbb{F}_q$ and requires rerunning the algorithm from the beginning. A second, less costly, alternative is to replace the modulus $(q-1)/(p-1)$ in (2.3) by $(q-1)/N$, where N is a factor of $q-1$ different than $p-1$. For instance, we might choose N to be the largest B'-smooth factor of $q-1$ for some smoothness bound B'. Indeed, since the Riemann hypothesis for curves over finite fields tells us that

$$h \le (\sqrt{p}+1)^{2g}, \tag{2.7}$$

where g is the genus of the function field F, we can eliminate the interference of the class group by taking B' equal to this upper bound for h. Of course, doing so may have its price. When $(q-1)/(p-1)$ is replaced by $(q-1)/N$, the algorithm is no longer guaranteed to produce the promised relations, since the element η appearing in (2.1) is in the subgroup of $\mathbb{F}_q^*$ of cardinality N and may not be in $\mathbb{F}_p^*$. As a consequence, it becomes necessary, in the general discrete logarithm algorithm of which Algorithm 2.1 is a part, to compute a logarithm in the subgroup of size N. The Pohlig–Silver–Hellman (see [16]) algorithm can perform such a calculation without increasing the complexity of the algorithm if and only if the largest prime factor of N is sufficiently small. We leave the details to the reader.

Finally, a third approach, which is applicable in the case that a solution to (2.3) is obtained but (2.6) is not in $\mathbb{F}_p^*$, is to solve (2.3) modulo $h'(q-1)/(p-1)$, where h' is any multiple of the largest factor of h composed of primes dividing $(q-1)/(p-1)$. With this change, (2.4) is valid with D replaced by $h'D$, and (2.5) follows. In particular, we can let $h' = ((q-1)/(p-1))^{\ell}$, where ℓ is chosen so that $\ell \geq \lfloor \log_2 h \rfloor$. Note that such an ℓ can be found using (2.7).

We present now the entire reduction algorithm.

Algorithm 2.9. This algorithm takes as input two elements $t, u \in \mathbb{F}_q^*$ and three parameters E_1, E_2, and E_3. Its goal is to output a multiplicative relation of the form

$$t^z u = \eta \prod y_i^{c_i}, \tag{2.8}$$

where z and the c_i are integers, η is in $\mathbb{F}_p^*$, and the y_i are in S_B.

Step 1. Randomly test integers $z \in \{0, \ldots, q-2\}$ until one is found such that the element $t^z u \in \mathbb{F}_q$ is B_1-smooth when considered as a polynomial in $\mathbb{F}_p[X]$ (see [22, section 4] for a sieve-based alternative). Factor $t^z u$ into the product of an element in $\mathbb{F}_p^*$ and elements in S_{B_1}.

Step 2. Let J be the smallest integer such that $B_{J+1} \leq B$. If $J = 0$, the algorithm has fulfilled its purpose and terminates. Otherwise, let $j = 1$ and continue.

Step 3. We have $t^z u$ written as a product of an element in $\mathbb{F}_p^*$ and powers of elements in S_{B_j}. Let W_j be the subset of S_{B_j} containing these elements and use Algorithm 2.1, with parameters E_1, E_2, and E_3, to obtain an expression for each member of W_j as a product of an element in $\mathbb{F}_p^*$ and powers of polynomials in $S_{B_{j+1}}$. Substituting into our earlier expression, we obtain a factorization of $t^z u$ as a product of an element in $\mathbb{F}_p^*$ and powers of elements in $S_{B_{j+1}}$.

Step 4. If $j = J$, the algorithm terminates. Otherwise, increase j by one and repeat Step 3. This concludes our description of Algorithm 2.9.

3. Analysis of reduction. In this section we consider particular values of E_1, E_2, and E_3 for which we believe Algorithm 2.9 succeeds in producing a relation of the form (2.8) and compute the asymptotic complexity of the algorithm for this choice of parameters. We emphasize at the outset that our analysis is heuristic and that our results are conjectural. One of the many assumptions that we adopt is that the polynomials which are tested for smoothness at various points in the algorithm behave like random polynomials with respect to the property of being smooth. We then determine the likelihood that they are smooth by means of the following result. For a given prime p and positive integers δ and β with $\delta \geq \beta$, let $N_p(\delta, \beta)$ be the number of monic polynomials in $\mathbb{F}_p[X]$ which are of degree δ and are β-smooth.

Theorem 3.1. *Assume $\delta \geq \beta \geq 2 \log \delta$ and let $w = \delta/\beta$. Then, as β and w tend to infinity,*

$$\frac{N_p(\delta, \beta)}{p^{\beta}} = w^{-w(1+o(1))},$$

uniformly for all primes p.

Theorem 3.1 is a slightly weaker version of Theorem 2.1 in [3], which in turn follows from a result in [26]. Since the author of [26] has withdrawn the article due to the presence of [14], [15], and [19], it is to these three works that we refer the reader interested in a proof of Theorem 3.1. Since we are interested in the behavior of the various quantities in Algorithm 2.9 as functions of q, we provide the following reformulation of Theorem 3.1. For the remainder of this section, all $o(1)$'s are for $q \to \infty$.

Corollary 3.2. *Let β be functions from the set of prime powers to the set of positive integers, and let $\omega(q) = \delta(q)/\beta(q)$. Assume that $\omega(q) \to \infty$ as $q \to \infty$ and that $\beta(q) \geq 2 \log \delta(q)$ for q sufficiently large. Let p denote the prime divisor of q. Then*

$$\frac{N_p(\delta(q), \beta(q))}{p^{\delta(q)}} = \omega(q)^{-\omega(q)(1+o(1))}.$$

We now consider the case that

$$\delta(q) = M_q[s; c + o(1)] \quad \text{and} \quad \beta(q) = M_q[s'; c' + o(1)],$$

where $0 \leq s' < s \leq 1$ and c' is nonzero. Let ρ be the probability that a randomly chosen polynomial in $\mathbb{F}_p[X]$ of degree $\delta(q)$ is $\beta(q)$-smooth. It follows from Corollary 3.2 that if $\beta(q) \to \infty$ as $q \to \infty$, then

$$\rho = L_q[s - s'; (s' - s)c/c' + o(1)]. \tag{3.1}$$

Writing $q = p^n$, we observe that $\beta(q)$ is unbounded as $q \to \infty$ if and only if p and n satisfy $p \leq o(n^{s'/(1-s')}\log n)$. In the analysis that follows, we are concerned only with the case that $s' \geq 1/3$ and thus ensure the applicability of (3.1) by imposing on q the restriction that $p \leq o(n^{1/2}\log n)$. In section 4, we consider the case that this inequality does not hold.

Turning to Algorithm 2.9, we see from Corollary 3.2 that the expected number of trials needed to find z in Step 1 is equal to $L_q[1/3; 4/9 + o(1)]$. Since the expected number of steps required to test a candidate using the algorithm in [4] is bounded by a power of $\log q$, we find that the expected running time of this step is $L_q[1/3; 4/9+o(1)]$. To analyze Steps 2–4 of Algorithm 2.9, we inspect each step of Algorithm 2.1. We continue with all the notation introduced in the description of this method in the previous section.

Step 1. Let E_1 be the least integer greater than or equal to $M_q[1/3; 3/4]$. The number of pairs of polynomials tested in this step is p^{2E_1}. Hence the running time of the step is bounded by $p^{2E_1(1+o(1))} = L_q[1/3; 3/2 + o(1)]$, where the limit implicit in the $o(1)$ converges uniformly for all j.

We consider whether the number of pairs c, d produced by the sieve is at least $|T|$. Observe that

$$N(c - dy) = d^e H(X, c/d) = c^e - X^{\mu e - n} s d^e.$$

Relying on the fact that a random polynomial of degree n in $\mathbb{F}_p[X]$ is irreducible with probability greater than $(1 - 2p^{-n/2})/n$, we expect that s can be found in $(1 + o(1))n$ trials and therefore adopt the assumption that there exists a constant κ such that

$\deg(s) \le \kappa \log n$ for all q. Since $\mu e - n \le e$, we see that when $\deg(c)$ and $\deg(d)$ are less than E_1, the degree of $N(c - dy)$ is bounded by

$$(E_1 + 1)e + \kappa \log n. \tag{3.2}$$

Let ρ_y be the probability that a random polynomial in $\mathbb{F}_p[X]$ of degree at most (3.2) is C-smooth. Using $(7/8)(\log q/\log\log q)^{1/3}$ as a bound for e, we find that (3.2) is bounded by $M_q[2/3; 21/32 + o(1)]$. Corollary 3.2 then implies that $\rho_y \ge L_q[1/3; -7/16 + o(1)]$. Since $\deg(m) = \lceil n/e \rceil$ we obtain

$$\lceil n/e \rceil + E_1 \tag{3.3}$$

as a bound for the degree of the elements $c - dm$. Let ρ_m be the probability that a random polynomial in $\mathbb{F}_p[X]$ of degree bounded by (3.3) is B_{j+1}-smooth, and let

$$\omega_j = \frac{\lceil n/e \rceil + E_1}{B_{j+1}} = M_q[1/3; 32/21 + o(1)].$$

Then, by Corollary 3.2, we have

$$\rho_m = \omega_j^{\omega_j(1+o(1))} = L_q[1/3; -32/63 + o(1)]. \tag{3.4}$$

Moreover, since, as $q \to \infty$, both B_j and ω_j tend to ∞ uniformly for all j, we find, using Theorem 3.1, that the limit implicit in the $o(1)$ in (3.4) converges uniformly for all j. We now make the assumption that, uniformly for all j,

$$\rho_y \rho_m E_1^2 = F_j^{1+o(1)}.$$

We find then that

$$F_j = L_q[1/3; 559/1008 + o(1)], \tag{3.5}$$

uniformly for all j. Since $|T| = L_q[1/3; 1/2 + o(1)]$, we see that for q sufficiently large this step will produce enough pairs satisfying the smoothness conditions.

Step 2. For most of the places $Q \in T$, the value $v_Q(c - dy)$ can be read off of the factorization of $N(c - dy)$ in the same way as is done in the number field sieve (see [5]). The places Q with the property that the localization of $\mathcal{O}$ at $\mathcal{O} \cap Q$ is not a discrete valuation ring require more work. In [2], the authors propose a method for computing the valuations in this case which makes use of Newton polygons and which runs in time bounded by $(\log q)^{O(1)}$.

Step 3.

(i) Let E_2 be the least integer greater than or equal to $M_q[1/3; 1]$. For each $w \in W$, the running time of this substep is equal to $p^{E_2(1+o(1))} = L_q[1/3; 1 + o(1)]$, where the $o(1)$ converges uniformly for all j . What must be checked is whether the search for r and s succeeds. We argue that for q sufficiently large it does.

Let θ and τ be the vectors obtained in this step by means of the Euclidean algorithm. We first consider the case that $E_2 \le |\tau| - |\theta|$. Let F_j be the number of polynomials r of degree less than E_2 such that both $N(r\theta_1 + \tau_1 - (r\theta_2 + \tau_2)y)$ and $(r\theta_1 + \tau_1 - (r\theta_2 + \tau_2)m)/w$ are B_{j+1}-smooth. We find that when $\deg(r) < E_2$, the degree of $N(r\theta_1 + \tau_1 - (r\theta_2 + \tau_2)y)$ is bounded by

$$\begin{aligned} (|\tau| + 1)e + \kappa \log n &\le (\deg(w) + 1)e + \kappa \log n \\ &\le (B_j + 1)e + \kappa \log n \end{aligned} \tag{3.6}$$

and the degree of $(r\theta_1 + \tau_1 - (r\theta_2 + \tau_2)m)/w$ is bounded by

$$|\tau| + \lceil n/e \rceil \leq B_j + \lceil n/e \rceil. \tag{3.7}$$

Let ρ be the probability that a polynomial of degree less than the sum of the bounds given in (3.6) and (3.7) is B_{j+1}-smooth. We determine, using Theorem 3.1 and Corollary 3.2, that $\rho = L_q[1/3; -403/504 + o(1)]$ uniformly for all j. We now adopt the assumption that $\rho\, p^{E_2} = F_j^{1+o(1)}$ uniformly for all j. Since $p^{E_2} = L_q[1/3; 1+o(1)]$, we see that $F_j = L_q[1/3; 101/504 + o(1)]$ and hence conclude that, for q sufficiently large, $F_j \geq 1$.

We turn to the case that $E_2 > |\tau| - |\theta|$. Recall that δ and ϵ are integers whose sum is E_2 and for which

$$\Big|(\delta + |\tau|) - (\epsilon + |\theta|)\Big| \leq 1.$$

Note that

$$\begin{aligned} \max\{\delta + |\tau|, \epsilon + |\theta|\} &\leq \frac{E_2 + |\tau| + |\theta| + 1}{2} \\ &= \frac{E_2 + \deg(w) + 1}{2} \\ &\leq \frac{E_2 + B_j + 1}{2}. \end{aligned}$$

Let F_j be the number of pairs of polynomials r, s such that $\deg(r) < \delta$ and $\deg(s) < \epsilon$ and such that $N(r\theta_1 + s\tau_1 - (r\theta_2 + s\tau_2)y)$ and $(r\theta_1 + s\tau_1 - (r\theta_2 + s\tau_2)m)/w$ are B_{j+1}-smooth. We observe that when $\deg(r) < \delta$ and $\deg(s) < \epsilon$, the polynomial $N(r\theta_1 + s\tau_1 - (r\theta_2 + s\tau_2)y)$ has degree bounded by

$$(\max\{(\epsilon + |\theta|, \delta + |\tau|\} + 1)e + \kappa \log n \leq \left(\frac{E_2 + B_j + 3}{2}\right) e + \kappa \log n \tag{3.8}$$

and the polynomial $(r\theta_1 + s\tau_1 - (r\theta_2 + s\tau_2)m)/w$ has degree bounded by

$$\max\{(\epsilon + |\theta|, \delta + |\tau|\} + \lceil n/e \rceil \leq \frac{E_2 + B_j + 1}{2} + \lceil n/e \rceil. \tag{3.9}$$

Let ρ be the probability that a polynomial of degree less than the sum of the bounds in (3.8) and (3.9) is B_{j+1} smooth. Using a bound of $M_q[2/3; (7/8) + o(1)]$ for the product of E_2 and e in (3.8), we find, by Theorem 3.1 and Corollary 3.2, that $\rho \geq L_q[1/3; -95/112 + o(1)]$ uniformly for all j. We make the assumption that $\rho\, p^{E_2} = F_j^{1+o(1)}$ uniformly for all j. It follows that $F_j \geq L_q[1/3; 17/112 + o(1)]$ and in particular that, for q sufficiently large, $F_j \geq 1$.

(ii) Let E_3 be the least integer greater than or equal to $M_q[1/3; 1.026]$. For each $w \in W$, the running time for this substep is equal to $p^{E_3(1+o(1))} = L_q[1/3; 1.026 + o(1)]$. We argue that, for q sufficiently large, the search for the elements r_i and s_i is successful. Our analysis proceeds in the same manner as the one given for substep (i). We leave it to the reader to make the appropriate changes.

(iii) The time required to compute one valuation vector is certainly bounded by the running time of Step 2.

(iv) The size of T, and hence the column length of the matrix A appearing in (2.3), is equal to $L_q[1/3; 1/2 + o(1)]$. Ignoring the concerns about the class number of

F addressed in section 2, we expect, and assume, that the number of columns required in order that A have full rank modulo $(q-1)/(p-1)$ is also $L_q[1/3; 1/2 + o(1)]$, where the limit implicit in the $o(1)$ converges uniformly for all j. Equation (3.5) then implies that, for q sufficiently large, enough vectors are available to ensure that (2.3) is solved for all w. Moreover, our assumption implies that, as we move through the elements of W, the total number of times we fail to find a solution to (2.3) and must add an additional vector to A is bounded by $L_q[1/3; 1/2 + o(1)]$.

An upper bound for the running time of the method described in [31] is the product of the maximum number of nonzero entries appearing in a column of the matrix A, the square of the maximum dimension of A, and a factor which is aymptotically insignificant in the present case. Using the fact that the number of entries in a column of A is asymptotically bounded by $(ne)^{O(1)}$, we obtain a running time of $L_q[1/3; 1 + o(1)]$, uniformly for all j, for each attempt to solve (2.3). Thus the total time spent on linear algebra is

$$L_q[1/3; 3/2 + o(1)] + |W| \cdot L_q[1/3; 1 + o(1)].$$

Combining steps, we conjecture that with parameters E_1, E_2, and E_3 as given, Algorithm 2.1 succeeds in time

$$L_q[1/3; 3/2 + o(1)] + |W| \cdot L_q[1/3; 1.026 + o(1)].$$

Returning to Algorithm 2.9, we therefore conjecture that with the same choice of parameters the algorithm succeeds and that Steps 2–4 require time at most

$$J(L_q[1/3; 3/2 + o(1)]) + \sum_{j=1}^{J} |W_j| \, L_q[1/3; 1.026 + o(1)], \tag{3.10}$$

where J is the parameter appearing in Step 2. To compute the size of W_j, we note that for a given value of j and for each $w \in W_j$ the number of B_{j+1}-smooth elements appearing in expression (2.6) is at most $l + 1$ more than the number of c, d used to form the matrix A in (2.3). According to our analysis of Steps 1 and 3 of Algorithm 2.1, both l and the number of factors of each smooth element appearing in (2.6) are asymptotically less than n. Since the set of pairs c, d depends on j and not on w, and since the number of such pairs is by assumption $L_q[1/3; 1/2 + o(1)]$, we have

$$|W_{j+1}| \leq n \, L_q[1/3; 1/2 + o(1)] + n^{O(1)} |W_j|.$$

Since $|W_1|$ is bounded by n, we find that $|W_j| \leq n^{O(j)} L_q[1/3; 1/2 + o(1)]$ and that (3.10) is equal to

$$n^{O(J)} L_q[1/3; 1.526 + o(1)].$$

It remains to determine the size of J as a function of q. Recall that J is the smallest integer such that $B_{J+1} \leq B$. Consequently, either $J = 0$ or we have $B_J > B$. In the latter case,

$$\begin{aligned} M_q\left[\frac{1}{3} + \frac{1}{3 \cdot 2^{J-1}}; 3/4\right] &= \frac{3}{4} \left(\frac{\log q}{\log\log q}\right)^{\frac{1}{3 \cdot 2^{J-1}}} n^{\frac{1}{3}} \left(\frac{\log\log q}{\log p}\right)^{\frac{2}{3}} \\ &> \left(\frac{4}{9}\right)^{\frac{1}{3}} n^{\frac{1}{3}} \left(\frac{\log\log q}{\log p}\right)^{\frac{2}{3}}, \end{aligned}$$

and we conclude that

$$\left(\frac{\log q}{\log\log q}\right)^{\frac{1}{2^{J-1}}} > \frac{256}{243}.$$

Solving for J, we find that $J \leq O(\log\log\log q)$. We thus arrive at the following conjecture.

CONJECTURE 3.13. *There exist input parameters E_1, E_2, and E_3 in Algorithm 2.9 so that in the case that the class numbers of the function fields encountered by the algorithm are prime to $(q-1)/(p-1)$, the algorithm succeeds in producing a multiplicative relation of the form* (2.8) *in an expected running time of at most*

$$L_q[1/3; 1.526 + o(1)]$$

for $q \to \infty$ with $p \leq n^{o(\sqrt{n})}$.

We will see in the next section that the time needed to compute the logarithms of the polynomials in $\mathbb{F}_q$ of degree $\leq B$ is $L_q[1/3; (32/9)^{1/3} + o(1)]$. Since $1.526 < (32/9)^{1/3}$, the choices we have made for the various quantities arising in the algorithms in section 2 are good enough to render the conjectural running time of our reduction inconsequential in the asymptotic running time of the discrete logarithm algorithm as a whole. However, these choices are not the best possible. To begin with, the values of E_1, E_2, and E_3 given in our analysis of Algorithm 2.1 are not optimal. In addition, replacing the 3/4 which appears as the secondary constant in the definition of B_j in Algorithm 2.1 with a value closer to $(4/9)^{1/3}$ reduces the optimized running time of the algorithm, though not dramatically. The interested reader can verify that it is possible through this modification to achieve a time arbitrarily close to $L_q[1/3; c + o(1)]$, where c is the real root of $x^3 - (9/4)^{1/3}x - 3/2$ and has a value approximately equal to 1.516. Finally, we have not investigated whether it is possible to lower the running time by using smoothness bounds B_j that are not taken to be the closest integers to numbers of the form

$$M_q\left[\frac{1}{3} + \frac{1}{3 \cdot 2^j}; c\right]$$

for fixed c. In this context, we note that in [6] Coppersmith uses a sequence of bounds that are not given in this manner, but which instead are defined recursively by the formula $B_{j+1} = \sqrt{BB_j}$, where B is, as in our case, the target bound of the entire reduction.

We conclude by observing that in the event that more than one logarithm in $\mathbb{F}_q$ is desired, the sieving in Step 1 of Algorithm 2.1 needs to be performed only once for a given value of j. In this case, it may be beneficial to lower the smoothness bound C, and hence the running time of the linear algebra in Step 3, at the expense of a more time consuming precomputation.

4. Logarithms of small elements. Let $q = p^n$, with p prime, and let t and u be elements in $\mathbb{F}_q^*$ such that u is in the subgroup generated by t. To compute $\log_t u$, we first use Algorithm 2.9 to produce a relation of the form (2.8). It then suffices to compute the logarithm of an element in $\mathbb{F}_p^*$, which can be accomplished with the number field sieve, and the logarithms of a collection of elements in the smoothness base S_B. In this section, we describe a method to compute the logarithms in the smoothness base. It is due to Adleman and Huang and can be found in [2]. Throughout this section we adopt the model for $\mathbb{F}_q$, as well as all the notation, introduced prior to Algorithm 2.1.

Algorithm 4.1. This algorithm takes as input two positive integers $e, \beta \le n$, a third parameter E, and a primitive element $t \in \mathbb{F}_q^*$ which, when considered as a polynomial in $\mathbb{F}_p[X]$, is β-smooth. Its purpose is to compute $\log_t u$ for all $u \in S_\beta$.

Let $\mu = \lceil n/e \rceil$, and $H(X,Y) = Y^e - X^{\mu e - n}s$. As in section 2, $H(X,Y)$ is absolutely irreducible by Eisenstein's criterion. Let $\mathcal{O} = \mathbb{F}_p[X,Y]/(H(X,Y))$, let y be the image of Y in $\mathcal{O}$, and denote by F the field of fractions of $\mathcal{O}$. Set $m = X^\mu$, in which case $H(X,m) \equiv 0 \bmod X^n - s$, and let $\phi : \mathcal{O} \to \mathbb{F}_q$ be the ring homomorphism which extends the usual projection map by sending $y \mapsto m$. Finally, let S be the set of finite places of $\mathbb{F}_p(X)$ of degree $\le \beta$, and let T be the union of the set of places of F at infinity and the set of places Q of F with the property that there exists $P \in S$ such that Q lies over P.

Step 1. Use a sieve to collect all pairs of relatively prime elements $c, d \in \mathbb{F}_p[X]$ such that $c - dy$ and $c - dm$ are both β-smooth, and $\deg(c)$ and $\deg(d)$ are less than E. If the number of pairs found is less than $|S| + |T| - 1$, the algorithm terminates.

Step 2. For $P \in S$, let v_P denote the discrete valuation associated to P, and similarly, for $Q \in T$, let v_Q denote the discrete valuation associated to Q. For each pair c, d found in Step 1, compute an exponent vector $V_{c,d}$ containing the values $v_P(c - dm)$, where P runs through the places in S, and the values $-v_Q(c - dy)$, where Q runs through the places in T. In addition, compute a vector V_t containing $v_P(t)$ for all $P \in S$ and 0's at the coordinates corresponding to the places of T.

Step 3. Let A be a square matrix whose first row is V_t and whose remaining rows are chosen from among the vectors $V_{c,d}$. Let V be the column vector of length $|S| + |T|$ whose first entry is 1 and whose remaining entries are 0. Use the method given in [31] to solve the matrix equation

$$Ax \equiv V \bmod (q-1)/(p-1). \tag{4.1}$$

If no solution is found, the algorithm terminates. If a solution is obtained, then it is likely to contain the logarithms we seek. To be more precise, for each $P \in S$, let x_P be the entry in the solution to (4.1) corresponding to the column of A containing the values of the discrete valuation associated to the place P. Let u_P be the monic, irreducible polynomial in $\mathbb{F}_p[X]$ of degree $\le \beta$ with the property that P is the place of the localization of $\mathbb{F}_p[X]$ at u_P. To simplify notation, we also use u_P to denote the element in $\mathbb{F}_q$ represented by u_P. With this convention, we see that $S_\beta = \{u_P \mid P \in S\}$. Finally, let h be the class number of F and assume that h is prime to $(q-1)/(p-1)$. We claim that in this case $x_P \equiv \log_t(u_P) \bmod (q-1)/(p-1)$ for all $P \in S$.

In support of this assertion, we give the following argument from [2]. Choose a place R of F of degree 1. Let Q be a place in T, and let f_Q be the degree of Q. Then the divisor $Q - f_Q R$ has degree 0, and so the divisor $hQ - hf_Q R$ is principal. In other words, there exists an element $\gamma_Q \in F$, determined up to a constant in $\mathbb{F}_p^*$, such that $\operatorname{div}(\gamma_Q) = hQ - hf_Q R$. We then see that for each c, d found in Step 1

$$\begin{aligned}\operatorname{div}\left(\prod_{Q \in T} \gamma_Q^{v_Q(c-dy)}\right) &= \sum_{Q \in T} v_Q(c-dy)(hQ - hf_Q R) \\ &= \sum_{Q \in T} v_Q(c-dy)hQ - \left(\sum_{Q \in T} v_Q(c-dy) f_Q\right) hR.\end{aligned}$$

The quantity $\sum_{Q \in T} v_Q(c-dy) f_Q$ is the degree of the divisor of $c - dy$ and is consequently equal to 0. Since $\operatorname{div}(c-dy) = \sum_{Q \in T} v_Q(c-dy)Q$, we conclude that $(c-dy)^h$

and $\prod_{Q\in T}\gamma_Q^{v_Q(c-dy)}$ have the same divisor and therefore that there exists some $\eta\in\mathbb{F}_p^*$ such that

$$(c-dy)^h=\eta\prod_{Q\in T}\gamma_Q^{v_Q(c-dy)}. \tag{4.2}$$

The element $c-dm$ is β-smooth, and so, for some $\eta'\in\mathbb{F}_p^*$,

$$c-dm=\eta'\prod_{P\in S}u_P^{v_P(c-dm)}. \tag{4.3}$$

Since $\phi(c-dy)=\phi(c-dm)$, we see from (4.2) and (4.3) that

$$\left(\eta'\prod_{P\in S}u_P^{v_P(c-dm)}\right)^h=\eta\prod_{Q\in T}\phi(\gamma_Q)^{v_Q(c-dy)}.$$

Taking the logarithm of both sides yields the linear relation

$$\sum_{P\in S}v_P(c-dm)\log_t(u_P)\equiv\sum_{Q\in T}v_Q(c-dy)h^{-1}\log_t(\phi(\gamma_Q)) \bmod (q-1)/(p-1),$$

where h^{-1} is the inverse of h modulo $(q-1)/(p-1)$. Thus, by solving (4.1) we obtain for all $P\in S$ the residue class of $\log_t(u_P)$ modulo $(q-1)/(p-1)$. As a side product, we also find, for all $Q\in T$ the residue class of $h^{-1}\log_t(\phi(\gamma_Q))$. Note that we have relied on the fact that ϕ can be extended multiplicatively to a subset of F containing the elements γ_Q for all $Q\in T$ (see [2]).

Step 4. For each $P\in S$, compute

$$\eta_P=u_P\,t^{-x_P}.$$

We expect that $\eta_P\in\mathbb{F}_p^*$. If it is not, the algorithm terminates, Otherwise, we use the number field sieve to compute $\log_{t'}\eta_P$, where $t'=t^{(q-1/(p-1)}$. We then obtain the logarithm of u_P by means of the formula

$$\log_t(u_P)=x_P+\left(\frac{q-1}{p-1}\right)\log_{t'}(\eta_P).$$

This concludes the description of Algorithm 4.1.

Though Algorithm 4.1 terminates, it may do so without producing the desired logarithms. If Step 3 reveals that A is singular, one option is to change A by replacing one or more rows with valuation vectors $V_{c,d}$ from Step 2 which do not appear in A. If the problem persists or in Step 4, if η_P is not in $\mathbb{F}_p^*$, it is likely that the difficulty is due to the existence of a nontrivial common divisor of h and $(q-1)/(p-1)$. For more on this obstruction, we refer the reader to the discussion following Algorithm 2.1 in section 2.

In the analysis of Algorithm 4.1 that follows, we rely on the assumption that the elements we check for smoothness in Step 1 behave like random polynomials with respect to smoothness and hence that we can use Corollary 3.2, as well as Theorem 4.7 stated below, to determine how many tests are needed in this step. In what follows, all $o(1)$'s are for $q\to\infty$.

Our results depend greatly on the behavior of p and n as $q \to \infty$. We begin by assuming that $M_q[1/3; (4/9)^{1/3}]$ is unbounded as $q \to \infty$ or, equivalently, that $p \le n^{o(\sqrt{n})}$. In this case, we conjecture that e, β, and E can be chosen so that

$$\begin{aligned} e &= \left(\frac{(3 + o(1))\log q}{2 \log\log q} \right)^{1/3}, \\ \beta &= M_q[1/3; (4/9)^{1/3} + o(1)], \\ E &= M_q[1/3; (32/9)^{1/3} + o(1)] \end{aligned} \tag{4.4}$$

and so that Step 1 of the algorithm produces at least $|S| + |T| - 1$ pairs c, d satisfying the smoothness conditions. Furthermore, we conjecture that the bound on E in (4.4) cannot be improved and hence that the optimal running time of Step 1 of Algorithm 4.1 is

$$L_q[1/3; (32/9)^{1/3} + o(1)]. \tag{4.5}$$

We refer the reader interested in the details to [2], where these values are established for the present algorithm, and [11], where they are obtained for the analogous special number field sieve. In Step 2, the entries in an exponent vector $V_{c,d}$ corresponding to the places in S and most of the places of T can be read off of the factorizations of $c - dm$ and $N(c - dy)$ obtained during sieving. As explained in [2], the valuations at the remaining places of T can be computed in time polynomial in $\log q$ using a method involving Newton polygons. Since the matrix A in Step 3 is sparse, Wiedemann's algorithm [31] runs in time $p^{2\beta(1+o(1))}$, which is equal to (4.5) when β satisfies the equation in (4.4). Finally, in Step 4, the number field sieve for prime fields computes the logarithm of each η_P in an expected running time of $L_p[1/3; (64/9)^{1/3} + o(1)]$. Thus, we conjecture that so long as the class number of F is prime to $(q-1)/(p-1)$, we can choose parameters so that Algorithm 4.1 succeeds in a running time of (4.5).

We consider now the case that $M_q[1/3; (4/9)^{1/3}]$ is bounded as $q \to \infty$. Then $n^{\sqrt{n}}$ is bounded by a constant power of p. We take e to be the positive integer closest to $\sqrt{n/2}$ and let $\beta = 1$. We observe immediately that computing the valuation vectors in Step 2, solving (4.1) in Step 3, and determining the logarithms of the elements η_P in Step 4 require at most time $p^{2\beta(1+o(1))} = p^{2+o(1)}$. To analyze Step 1, we rely on the following result which appears as Theorem 2.2 in [3] and is proven there using elementary techniques.

Theorem 4.7. *Let δ and β be integers with $\delta \ge \beta^2 \ge 1$. Let $\omega = \delta/\beta$. Then, for all primes p,*

$$\frac{N_p(\delta, \beta)}{p^\delta} \ge \delta^{-\omega}.$$

By our choice of e, there exists a constant k such that the degree of the polynomials being tested for smoothness is bounded by $kE\sqrt{n}$. According to Theorem 4.7, the probability that a random polynomial in $\mathbb{F}_p[x]$ of degree $\le kE\sqrt{n}$ is β-smooth is at least $(kE\sqrt{n})^{-kE\sqrt{n}}$. We therefore conjecture that in order to find at least $|S| + |T| - 1$

pairs c, d satisfying the given smoothness conditions, the number of pairs that need to be tested is $\le (|S| + |T| + 1)(kE\sqrt{n})^{kE\sqrt{n}} = p^{1+o(1)}(kE\sqrt{n})^{kE\sqrt{n}}$. Since $n^{\sqrt{n}}$ is bounded by a constant power of p, so is $\sqrt{n}^{\sqrt{n}}$. We conclude that E can be taken to be constant. That is, we can choose e, β, and E so that the sieving stage of Algorithm 4.1 produces enough pairs and so that the sieve runs in time bounded by a constant power of p. We consider two cases. If $p \le n^{O(\sqrt{n})}$ as $q \to \infty$, then any constant power of p is bounded by $L_q[1/3; c]$ for some constant c, and we conjecture that Algorithm 4.1 succeeds in a running time of this form. If for some constant $k' > k/2$ we have $p \ge n^{k'\sqrt{n}}$ as $q \to \infty$, then $k\sqrt{n}^{k\sqrt{n}} < p$ for sufficiently large q. Indeed, either n is unbounded, in which case $k\sqrt{n}^{k\sqrt{n}} < n^{k'\sqrt{n}} \le p$, for sufficiently large n, or n is bounded, in which case $k\sqrt{n}^{k\sqrt{n}} < p$ for sufficiently large p. Therefore, we conjecture that for q large we can restrict our search for pairs c, d to those which are linear and that the number of pairs that need to be tested is at most $p^{2+o(1)}$. We summarize our results in the following conjecture, which we organize to parallel the complexity results given in [3] for a variation of the standard index calculus method.

CONJECTURE 4.8. *There exist input parameters e, β, and E in Algorithm* 4.1 *and a constant k' so that in the case that the class number of the field F appearing in the algorithm is prime to $(q-1)/(p-1)$, the algorithm succeeds in computing $\log_t u$ for all $u \in S_\beta$ in time at most $L_q[1/3; (32/9)^{1/3} + o(1)]$ for $q \to \infty$ with $p \le n^{o(\sqrt{n})}$, in time at most $L_q[1/3; O(1)]$ for $q \to \infty$ with $p \le n^{O(\sqrt{n})}$, and in time at most $p^{2+o(1)}$ for $q \to \infty$ with $p > n^{k'\sqrt{n}}$.*

Conjectures 4.8 and 3.13 together yield the conjecture that for $q \to \infty$ with $p \le n^{o(\sqrt{n})}$ the special function field sieve requires expected time $L_q[1/3; (32/9)^{1/3} + o(1)]$ to compute a logarithm in $\mathbb{F}_q$, as long as it is with respect to a base t which is represented by a smooth polynomial. This smoothness requirement can easily be avoided by finding a primitive element τ which is represented by a smooth polynomial and using the identity

$$\log_t u \equiv \frac{\log_\tau u}{\log_\tau t} \bmod q - 1.$$

According to Theorem 1.1 in [25], once $q - 1$ is factored, we can obtain such a τ in time $p(\log q)^{O(1)}$. We leave it to the reader to verify that using the special number field sieve to factor $q - 1$ in the case that $p \le n^{o(\sqrt{n})}$ and the general number field sieve otherwise, we can complete the search for τ in sufficiently little time so as not to affect the asymptotic running times given in Conjecture 4.8. Unfortunately, the same cannot be said of Algorithm 2.9. In the case that $p \le n^{O(\sqrt{n})}$, the type of argument made above for Algorithm 4.1 leads to the conjecture that we can choose parameters for Algorithm 2.9 so that it succeeds in expected time at most $L_q[1/3; O(1)]$. However, in the case that $p > n^{k'\sqrt{n}}$, the optimal parameter choices yield a lower bound of p^{3C} for the running time of Algorithm 2.9, where C is the smoothness bound appearing in Step 1 of Algorithm 2.1. We conclude that in this range Adleman's original function field sieve [1] is a better choice than the special function field sieve since it runs conjecturally in expected time at most $p^{2+o(1)}$.

REFERENCES

[1] L.M. ADLEMAN, *The function field sieve*, in Algorithmic Number Theory, ANTS-I, Lecture Notes in Comput. Sci. 877, L.M. Adleman and M.-D. Huang, eds., Springer-Verlag, Berlin, 1994, pp. 108–121.

[2] L.M. Adleman and M-D. Huang, *Function field method for discrete logarithms over finite fields*, Inform. and Comput., 151 (1999), pp. 5–16.

[3] R.L. Bender and C. Pomerance, *Rigorous Discrete Logarithm Computations in Finite Fields via Smooth Polynomials*, AMS/IP Stud. Adv. Math., 7, AMS, Providence, RI, 1998, pp. 221–232.

[4] E. Berlekamp, *Factoring polynomials over large finite fields*, Math. Comp., 25 (1970), pp. 713–735.

[5] J.P. Buhler, H.W. Lenstra, Jr., and C. Pomerance, *Factoring integers with the number field sieve*, in The Development of the Number Field Sieve, Lecture Notes in Math. 1554, Springer-Verlag, Berlin, 1993, pp. 50–94.

[6] D. Coppersmith, *Fast evaluation of discrete logarithms in fields of characteristic two*, IEEE Trans. Inform. Theory, 30 (1984), pp. 587–594

[7] D.M. Gordon, *Discrete logarithms in GF(P) using the number field sieve*, SIAM J. Discrete Math. 6 (1993), pp. 124–138.

[8] A. Joux and R. Lercier, *Improvements on the General Number Field Sieve for Discrete Logarithms in Prime Fields*, preprint.

[9] A. Joux and R. Lercier, *The function field sieve is quite special*, in Algorithmic Number Theory, ANTS-V, Lecture Notes in Comput. Sci. 2369, C. Fieker and D.R. Kohel, eds., Springer-Verlag, Berlin, 2002, pp. 131–445.

[10] A.K. Lenstra and H.W. Lenstra, Jr., eds., *The Development of the Number Field Sieve*, Lecture Notes in Math. 1554, Springer-Verlag, Berlin, 1993.

[11] A.K Lenstra, H.W. Lenstra, Jr., M.S. Manasse, and J.M. Pollard, *The number field sieve*, in The Development of the Number Field Sieve, Lecture Notes in Math. 1554, Springer-Verlag, Berlin, 1993, pp. 11–42.

[12] A.K Lenstra, H.W. Lenstra, Jr., M.S. Manasse, and J.M. Pollard, *The factorization of the ninth Fermat number*, Math Comp., 61 (1993), pp. 319–349.

[13] D. Lorenzini, *An Invitation to Arithmetic Geometry*, Grad. Stud. Math. 9, AMS, Providence, RI, 1996.

[14] E. Manstavičius, *Semigroup elements free of large prime factors*, in New Trends in Probability and Statistics, 2: Analytic and Probabilistic Methods in Number Theory, F. Schweiger and E. Manstavicius, eds, VSP, Utrecht, The Netherlands, 1992, pp. 135–153.

[15] E. Manstavičius, *Remarks on elements in semigroups that are free of large prime factors*, Lithuanian Math. J., 132 (1993), pp. 400–409.

[16] K. McCurley, *The discrete logarithm problem*, in Cryptology and Computational Number Theory, Proc. Sympos. Appl. Math. 42, C. Pomerance, ed., AMS, Providence, RI, 1990, pp. 49–74.

[17] R.J. McEliece, *The Theory of Information and Coding: A Mathematical Framework for Communications*, Encyclopedia Math. Appl. 3, Addison-Wesley, Reading, MA, 1977.

[18] A.M. Odlyzko, *Discrete logarithms: The past and the future*, Des. Codes Cryptogr., 19 (2000), pp. 129–145.

[19] D. Panario, X. Gourdon, and P. Flajolet, *An analytic approach to smooth polynomials over finite fields*, in Algorithmic Number Theory, ANTS-III, Lecture Notes in Comput. Sci. 1424, J. Buhler, ed., Springer-Verlag, Berlin, 1998, pp. 237–246.

[20] O. Schirokauer, *Discrete logarithms and local units,* in Theory and Applications of Numbers Without Large Prime Factors, Philos. Trans. Roy. Soc. London Ser. A 345, R.C. Vaughan, ed., Royal Society, London, 1993, pp. 409–424.

[21] O. Schirokauer, *Using number fields to compute logarithms in finite fields*, Math. Comp., 69 (2000), pp. 1267–1283.

[22] O. Schirokauer, *The impact of the number field sieve on the discrete logarithm problem*, in Algorithmic Number Theory: Lattices, Number Fields, Curves, and Cryptography, Cambridge University Press, Cambridge, UK, to appear.

[23] O. Schirokauer, D. Weber, and T. Denny, *Discrete logarithms: The effectiveness of the index calculus method*, in Algorithmic Number Theory, ANTS-II, Lecture Notes in Comput. Sci. 1123, H. Cohen, ed., Springer-Verlag, Berlin, 1996, pp. 337–361.

[24] I.A. Semaev, *Special prime numbers and discrete logs in prime finite fields*, Math. Comp., 71 (2002), pp. 363–377.

[25] V. Shoup, *Searching for primitive roots in finite fields*, Math. Comp., 58 (1992), pp. 369–380.

[26] K. Soundararajan, *Smooth Polynomials: Analogies and Asymptotics*, preprint.

[27] H. Stichtenoth, *Algebraic Function Fields and Codes*, Springer-Verlag, Berlin, 1993.

[28] E. Thomé, *Computing discrete logarithms over* $\mathrm{GF}(2^{607})$, in Advances in Cryptology – Asiacrypt 2001, Lecture Notes in Comput. Sci. 2248, C. Boyd, ed., Springer-Verlag, Berlin, 2001, pp. 107–124.

[29] D. Weber, *Computing discrete logarithms with the number field sieve*, in Algorithmic Number Theory, ANTS-II, Lecture Notes in Comput. Sci. 1123, H. Cohen, ed., Springer-Verlag, Berlin, 1996, pp. 391–403.

[30] D. Weber and T. Denny, *The solution of McCurley's discrete log challenge*, in Advances in Cryptology – Crypto '98, Lecture Notes in Comput. Sci. 1462, H. Krawczyk, ed., Springer-Verlag, Berlin, 1998, pp. 458–471.

[31] D.H. Wiedemann, *Solving sparse linear equations over finite fields*, IEEE Trans. Inform. Theory, 32 (1986), pp. 54–62.

SIAM J. Discrete Math.
Vol. 16, No. 1, pp. 99–113

ASYMPTOTIC SIZE RAMSEY RESULTS FOR BIPARTITE GRAPHS[*]

OLEG PIKHURKO[†]

Abstract. We show that $\lim_{n\to\infty} \hat{r}(F_{1,n},\ldots,F_{q,n},F_{q+1},\ldots,F_r)/n$ exists, where the bipartite graphs $F_{q+1},\ldots,F_r$ do not depend on n while, for $1 \le i \le q$, $F_{i,n}$ is obtained from some bipartite graph F_i with parts $V_1 \cup V_2 = V(F_i)$ by duplicating each vertex $v \in V_2$ $(c_v + o(1))n$ times for some real $c_v > 0$.

In fact, the limit is the minimum of a certain mixed integer program. Using the Farkas lemma we show how to compute it when each forbidden graph is a complete bipartite graph, in particular answering the question of Erdős, Faudree, Rousseau, and Schelp [*Period. Math. Hungar.*, 9 (1978), pp. 145–161], who asked for the asymptotics of $\hat{r}(K_{s,n},K_{s,n})$ for fixed s and large n. Also, we prove (for all sufficiently large n) the conjecture of Faudree, Rousseau, and Sheehan in [*Graph Theory and Combinatorics*, B. Bollobas, ed., Cambridge University Press, Cambridge, UK, 1984, pp. 273–281] that $\hat{r}(K_{2,n},K_{2,n}) = 18n - 15$.

Key words. size Ramsey number, bipartite graphs, mixed integer programming, Farkas lemma

AMS subject classifications. 05C55, 05C35, 90C05, 90C11

PII. S0895480101384086

1. Introduction.

Let $(F_1,\ldots,F_r)$ be an r-tuple of graphs which are called *forbidden.* We say that a graph G *arrows* $(F_1,\ldots,F_r)$ if for any r-coloring of $E(G)$, the edge set of G, there is a copy of F_i of color i for some $i \in [r] := \{1,\ldots,r\}$. We denote this *arrowing property* by $G \to (F_1,\ldots,F_r)$.

The (ordinary) *Ramsey number* asks for the minimum order of such G. Here, however, we deal exclusively with the *size Ramsey number*

$$\hat{r}(F_1,\ldots,F_r) = \min\{e(G) : G \to (F_1,\ldots,F_r)\}$$

which is the smallest number of edges that an arrowing graph can have.

Size Ramsey numbers seem hard to compute, even for simple forbidden graphs. For example, the old conjecture of Erdős [6] that $\hat{r}(K_{1,n},K_3) = 3n(n+1)/2$ has only recently been disproved in [16], where it is shown that $\hat{r}(K_{1,n},F) = (1+o(1))n^2$ for any fixed 3-chromatic graph F. (Here, $K_{m,n}$ is the complete bipartite graph with parts of sizes m and n; K_n is the complete graph of order n.)

This research has been initiated as an attempt to find the asymptotics of $\hat{r}(K_{1,n},F)$ for a fixed graph F. The case $\chi(F) \ge 4$ is treated in [14] (and [16] deals with $\chi(F) = 3$). What can be said if F is a bipartite graph?

Faudree, Rousseau, and Sheehan [10] proved that

$$\hat{r}(K_{1,n},K_{2,m}) = 4n + 2m - 4$$

for every $m \ge 9$ if n is sufficiently large (depending on m) and stated that their method shows that $\hat{r}(K_{1,n},K_{2,2}) = 4n$, $n \ge 3$. They also observed that $K_{s,2n}$ arrows the pair

[*] Received by the editors January 24, 2001; accepted for publication (in revised form) July 22, 2002; published electronically January 3, 2003. This research was supported by a Research Fellowship, St. John's College, Cambridge. Part of this research was carried out during the author's stay at the Humboldt University, Berlin, sponsored by the German Academic Exchange Service (DAAD).

http://www.siam.org/journals/sidma/16-1/38408.html

[†] DPMMS, Centre for Mathematical Sciences, Cambridge University, Cambridge CB3 0WB, England (O.Pikhurko@dpmms.cam.ac.uk).

$(K_{1,n}, C_{2s})$ for $n \geq s$, where C_{2s} is the cycle of order $2s$; hence $\hat{r}(K_{1,n}, C_{2s}) \leq 2sn$ then.

Let P_s be the path with s vertices. Lortz and Mengersen [13] showed that $K_{k,2n-1} \to (K_{1,n}, P_{2k+1})$ and $K_k + \overline{K}_{2n-k-1} \to (K_{1,n}, P_{2k})$ and conjectured that this is sharp for any $s \geq 4$ provided n is sufficiently large; that is,

$$\hat{r}(K_{1,n}, P_s) = \begin{cases} 2kn - k & \text{if } s = 2k+1, \\ 2kn - k(k+3)/2 & \text{if } s = 2k, \end{cases} \qquad n \geq n_0(s). \tag{1.1}$$

The conjecture was proved for $4 \leq s \leq 7$ in [13].

Size Ramsey numbers $\hat{r}(F_1, F_2)$ for bipartite graphs F_1 and F_2 are also studied in [8, 5, 2, 3, 7, 9, 12, 11], for example.

It is not hard to see that, for fixed $s_1, \ldots, s_r \in \mathbb{N}$ and $t_1, \ldots, t_r \in \mathbb{R}_{>0}$, we have

$$\hat{r}(K_{s_1,\lfloor t_1 n\rfloor}, \ldots, K_{s_r,\lfloor t_r n\rfloor}) = O(n). \tag{1.2}$$

This follows, for example, by assuming that $s_1 = \cdots = s_r = s$, $t_1 = \cdots = t_r = t$ and considering K_{v_1,v_2}, where $v_1 = (s-1)r + 1$ and $v_2 = \lceil rtn\binom{v_1}{s}\rceil$. The latter graph has the required arrowing property. Indeed, for any r-coloring, each vertex of V_2 is incident to at least s edges of the same color; hence there are at least v_2 monochromatic $K_{s,1}$-subgraphs and some $S \in \binom{V_1}{s}$ appears in at least $v_2/\binom{v_1}{s} \geq rtn$ such subgraphs of which at least tn have the same color.

Here we will show that the limit $\lim_{n\to\infty} \hat{r}(F_{1,n}, \ldots, F_{r,n})/n$ exists if each forbidden graph is either a fixed bipartite graph or a subgraph of $K_{s,\lfloor tn\rfloor}$ which "dilates" uniformly with n. (The precise definition will be given in section 2.) In particular, $\hat{r}(K_{1,n}, F)/n$ tends to a limit for any fixed bipartite graph F.

The limit value can in fact be obtained as the minimum of a certain mixed integer program (which does depend on n). We have been able to solve the MIP when each $F_{i,n}$ is a complete bipartite graph. In particular, we answer the question of Erdős et al. [8, Problem B], who asked for the asymptotics of $\hat{r}(K_{s,n}, K_{s,n})$. Working harder on the case $s = 2$ we prove (for all sufficiently large n, $n \geq n_0$) the conjecture of Faudree, Rousseau, and Sheehan [10, Conjecture 15] that

$$\hat{r}(K_{2,n}, K_{2,n}) = 18n - 15, \tag{1.3}$$

where the upper bound is obtained by considering $K_{3,6n-5} \to (K_{2,n}, K_{2,n})$. The identity (1.3) is not true for all n: for example, it is stated in [10] that $\hat{r}(K_{2,2}, K_{2,2}) = 15$. The upper bound follows from $K_6 \to (K_{2,2}, K_{2,2})$, which is easy to verify. Our method could produce a concrete value for n_0 with extra tedious calculations, but this would probably be rather large.

Unfortunately, our MIP is not well suited for practical calculations, and we were not able to compute the asymptotics for any other nontrivial forbidden graphs; in particular, we had no progress on (1.1). However, we hope that the introduced method will produce more results: although the MIP is hard to solve, it may be possible that, for example, some manageable relaxation of it gives good lower or upper bounds.

Our method does not work if we allow both vertex classes of forbidden graphs to grow with n. In these settings, in fact, we do not know the asymptotics even in the simplest cases. For example, the best known bounds on $r = \hat{r}(K_{n,n}, K_{n,n})$ seem to be $r < \frac{3}{2}n^3 2^n$ for $n \geq 6$ (Erdős et al. [8]) and $r > \frac{1}{60}n^2 2^n$ for $n \geq n_0$ (Erdős and Rousseau [9]).

2. Main ideas and definitions. Let us briefly describe the main ideas behind our approach and how they came into existence. As an illustration, suppose we want to prove that $\hat{r}(K_{2,n}, K_{2,n}) \ge (18 + o(1))n$. Let n be large, and let $G \to (K_{2,n}, K_{2,n})$ be any graph with $e(G) \le (18 + o(1))n$. We try to get as much information about the structure of G as possible.

Let $L = \{x \in V(G) : d(x) \ge n\}$. Clearly, $|L| \le 18$. As no edge disjoint from L lies inside a $K_{2,n}$-subgraph of G, we can harmlessly remove all such edges from G; that is, we can assume that $V(G) \setminus L$ is an independent set in G.

Also, if we remove all edges within L, the arrowing property is only slightly impaired: the obtained graph G' arrows $(K_{2,n'}, K_{2,n'})$, where we can take $n' = n - 16$ (or even larger). So, replacing G by G' and n by n', we can assume that $G \subset K_{18,m}$ for some $m = m(n)$.

Also, we can assume that every vertex of L has degree at least $2n - 1$. (This is not crucial here, but this illustrates Lemma 3.1.) Indeed, if we remove any $x \in L$ of degree at most $2n - 2$, then the obtained graph G' arrows $(K_{2,n-1}, K_{2,n-1})$: any $(K_{2,n-1}, K_{2,n-1})$-free coloring of G' extends to a $(K_{2,n}, K_{2,n})$-free coloring of G by coloring the remaining edges without a monochromatic $K_{1,n}$ centered at x.

Thus, we can assume that $G \subset K_{9,m}$. How can we economically describe such a graph? This brings us to new definitions.

Let F be a bipartite graph. We assume that bipartite graphs come equipped with a fixed bipartition $V(F) = V_1(F) \cup V_2(F)$, although graph embeddings need not preserve it. We denote $v_i(F) = |V_i(F)|$, $i = 1, 2$; thus $v(F) = v_1(F) + v_2(F)$. Define

$$F^A = \{v \in V_2(F) : \Gamma_F(v) = A\}, \quad A \subset V_1(F),$$

where $\Gamma_F(v)$ denotes the neighborhood of v in F. (We will write $\Gamma(v)$, etc. when the encompassing graph F is clear from the context.) Clearly, in order to determine F (up to an isomorphism) it is enough to know $V_1(F)$ and $|F^A|$ for all $A \in 2^{V_1(F)}$.

Now, instead of dealing with $G \to (K_{2,n}, K_{2,n})$ we prefer to work with the numbers $|G^A|$. As $e(G) = O(n)$, we can let $n \to \infty$ over some sequence so that $|G^A|/n$ tends to a limit g_A for each $A \in 2^L$. The how we call it "weight" $\mathbf{g} = (g_A)_{A \in 2^L}$ cannot be arbitrary: the fact that $G \to (K_{2,n}, K_{2,n})$ imposes some restrictions on $\mathbf{g}$. The question arises whether we can rephrase the arrowing property for weights without appealing to the original graphs. This requires rewriting the notions of a subgraph, coloring, etc. For the sake of generality, one would also wish to allow constant (i.e., not depending on n) forbidden subgraphs, which prompts one to define the mixed relation "$F \subset \mathbf{g}$" as well, where F is a graph and $\mathbf{g}$ is a weight. This is the first part of the program, which culminates in Theorem 3.3, where it is shown that the "weight size Ramsey number" indeed gives the asymptotics of the ordinary number. However, the second part, to calculate the weight size Ramsey number, is not an easy task and we are able to carry it out for complete bipartite graphs only.

Let us give formal definitions. A *weight* $\mathbf{f}$ on a set $V(\mathbf{f})$ is a sequence $(f_A)_{A \in 2^{V(\mathbf{f})}}$ of nonnegative reals. A bipartite graph F *agrees* with $\mathbf{f}$ if $V_1(F) = V(\mathbf{f})$ and $F^A = \emptyset$ if and only if $f_A = 0$, $A \in 2^{V(\mathbf{f})}$. A sequence of bipartite graphs $(F_n)_{n \in \mathbb{N}}$ is a *dilatation* of $\mathbf{f}$ (or *dilates* $\mathbf{f}$) if each F_n agrees with $\mathbf{f}$ and

$$|F_n^A| = f_A n + o(n) \quad \forall A \in 2^{V(\mathbf{f})}.$$

(Of course, the latter condition is automatically true for all $A \in 2^{V(\mathbf{f})}$ with $f_A = 0$.) Clearly, $e(F_n) = (e(\mathbf{f}) + o(1))n$, where $e(\mathbf{f}) = \sum_{A \in 2^{V(\mathbf{f})}} f_A\,|A|$, so we call $e(\mathbf{f})$ the *size* of $\mathbf{f}$. Also, the *order* of $\mathbf{f}$ is $v(\mathbf{f}) = |V(\mathbf{f})|$ and the *degree* of $x \in V(\mathbf{f})$ is

$$d(x) = \sum_{\substack{A \in 2^{V(\mathbf{f})} \\ A \ni x}} f_A.$$

Clearly, $e(\mathbf{f}) = \sum_{x \in V(\mathbf{f})} d(x)$.

For example, given $t \in \mathbb{R}_{>0}$, the sequence $(K_{s,\lceil tn \rceil})_{n \in \mathbb{N}}$ is the dilatation of $\mathbf{k}_{s,t}$, where the symbol $\mathbf{k}_{s,t}$ will be reserved for the weight on $[s]$ which has value t on $[s]$ and zero otherwise. (We assume that $V_1(K_{s,m}) = [s]$.) It is not hard to see that any sequence of bipartite graphs described in the abstract is in fact a dilatation of some weight.

We write $F \subset \mathbf{f}$ if for some bipartition $V(F) = V_1(F) \cup V_2(F)$ there is an injection $h : V_1(F) \to V(\mathbf{f})$ such that for any $A \subset V_1(F)$ dominated by a vertex of $V_2(F)$ there is $B \subset V(\mathbf{f})$ with $B \supset h(A)$ and $f_B > 0$. This notation is motivated by the following easy lemma. In fact, we will implicitly prove a sharper version during the proof of Theorem 3.3, so we give no proof here.

LEMMA 2.1. *Let $(F_n)_{n \in \mathbb{N}}$ be a dilatation of $\mathbf{f}$. If $F \subset \mathbf{f}$, then F is a subgraph of F_n for all sufficiently large n. Otherwise, which is denoted by $F \not\subset \mathbf{f}$, no F_n contains F.* □

Next, we define the "$\subset$"-relation between two weights $\mathbf{f}$ and $\mathbf{g}$. Assume that $v(\mathbf{f}) \le v(\mathbf{g})$ by adding new vertices to $V(\mathbf{g})$ and letting $\mathbf{g}$ be zero on all new sets. We write $\mathbf{f} \subset \mathbf{g}$ if there is an injection $h : V(\mathbf{f}) \to V(\mathbf{g})$ and numbers $(w_{AB} \ge 0)_{A \in 2^{V(\mathbf{f})},\, B \in 2^{V(\mathbf{g})}}$ such that

$$\begin{aligned}
\forall A \in 2^{V(\mathbf{f})},\ \forall B \in 2^{V(\mathbf{g})} \quad & h(A) \not\subset B \Rightarrow w_{AB} = 0, \\
\forall A \in 2^{V(\mathbf{f})} \quad & \sum_{\substack{B \in 2^{V(\mathbf{g})} \\ B \supset h(A)}} w_{AB} \ge f_A, \\
\forall B \in 2^{V(\mathbf{g})} \quad & \sum_{\substack{A \in 2^{V(\mathbf{f})} \\ h(A) \subset B}} w_{AB} \le g_B.
\end{aligned}$$

This definition is a bit difficult to comprehend. In a sense, it corresponds to a graph embedding $F \subset G$ preserving the $V_1 \cup V_2$-partition: h embeds $V_1(F)$ into $V_1(G)$ and $w_{A,B}$ says how much of $F^A \subset V_2(F)$ is mapped into G^B. The motivation comes from the following lemma which, like Lemma 2.1, is not used later and so is stated without a proof.

LEMMA 2.2. *Let $(F_n)_{n \in \mathbb{N}}$ and $(G_n)_{n \in \mathbb{N}}$ be dilatations of $\mathbf{f}$ and $\mathbf{g}$, respectively. Then $\mathbf{f} \subset \mathbf{g}$ implies that for any $\epsilon > 0$ there is n_0 such that $F_n \subset G_m$ for any $n \ge n_0$ and $m \ge (1+\epsilon)n$. Otherwise, which is denoted by $\mathbf{f} \not\subset \mathbf{g}$, there is $\epsilon > 0$ and n_0 such that $F_n \not\subset G_m$ for any $n \ge n_0$ and $m \le (1+\epsilon)n$.* □

The weight $\subset$-relation enjoys many properties of the graph one. For example, $d(x) \le d(h(x))$ for any $x \in V(\mathbf{f})$:

$$d(x) = \sum_{\substack{A \in 2^{V(\mathbf{f})} \\ A \ni x}} f_A \le \sum_{\substack{A \in 2^{V(\mathbf{f})} \\ A \ni x}} \sum_{\substack{B \in 2^{V(\mathbf{g})} \\ B \supset h(A)}} w_{A,B} \le \sum_{\substack{B \in 2^{V(\mathbf{g})} \\ B \ni h(x)}} \sum_{\substack{A \in 2^{V(\mathbf{f})} \\ h(A) \subset B}} w_{A,B} \le \sum_{\substack{B \in 2^{V(\mathbf{g})} \\ B \ni h(x)}} g_B = d(h(x)). \tag{2.1}$$

An *r-coloring* $\mathbf{c}$ of $\mathbf{g}$ is a sequence $(c_{A_1,\dots,A_r})$ of nonnegative reals indexed by r-tuples of pairwise disjoint subsets of $V(\mathbf{g})$ such that

$$\sum_{A_1 \cup \dots \cup A_r = A} c_{A_1,\dots,A_r} > g_A \quad \forall A \in 2^{V(\mathbf{g})}. \tag{2.2}$$

The *ith color subweight* $\mathbf{c}_i$ is defined by $V(\mathbf{c}_i) = V(\mathbf{g})$ and

$$c_{i,A} = \sum_{\substack{A_1,\dots,A_r\\ A_i=A}} c_{A_1,\dots,A_r}, \quad A \in 2^{V(\mathbf{g})}. \tag{2.3}$$

The analogy is as follows: to define an r-coloring of G, it is enough to define, for all disjoint $A_1,\dots,A_r \subset V_1(G)$, how many vertices of $G^{A_1\cup\dots\cup A_r}$ are connected, for all $i \in [r]$, by color i precisely to A_i. We put the strict inequality in (2.2) so that Lemma 3.2 is true.

3. Existence of limit. Let $r \ge q \ge 1$. Consider a sequence $\mathbf{F} = (\mathbf{F}_1,\dots,\mathbf{F}_r)$, where $\mathbf{F}_i = \mathbf{f}_i$ is a weight for $i \in [q]$ and $\mathbf{F}_i = F_i$ is a bipartite graph for $i \in [q+1,r]$. Assume that $\mathbf{F}_i$ does not have an *isolated vertex* (that is, $x \in V(\mathbf{F}_i)$ with $d(x) = 0$), $i \in [r]$. We say that a weight $\mathbf{g}$ *arrows* $\mathbf{F}$ (denoted by $\mathbf{g} \to \mathbf{F}$) if for any r-coloring $\mathbf{c}$ of $\mathbf{g}$ we have $\mathbf{F}_i \subset \mathbf{c}_i$ for some $i \in [r]$. Define

$$\hat{r}(\mathbf{F}) = \inf\{e(\mathbf{g}) : \mathbf{g} \to \mathbf{F}\}. \tag{3.1}$$

The definition (3.1) imitates that of the size Ramsey number, and we will show that these are very closely related indeed. However, we need a few more preliminaries.

Observe that $\hat{r}(\mathbf{F}) < \infty$ by considering $\mathbf{k}_{a,b}$ which arrows $\mathbf{F}$ if, for example, $a = 1 + \sum_{i=1}^r (v(\mathbf{F}_i) - 1)$ and b is sufficiently large; cf. (1.2). Let l be an integer greater than $\hat{r}(\mathbf{F})/d_0$, where $d_0 = \sum_{i=1}^q d_i$ and

$$d_i = \min\{d_{\mathbf{f}_i}(x) : x \in V(\mathbf{f}_i)\} > 0, \quad i \in [q].$$

LEMMA 3.1. *Let $\mathbf{g} \to \mathbf{F}$ have no isolated vertices. If $d_{\mathbf{g}}(x) < d_0$ for some $x \in V(\mathbf{g})$ or if $v(\mathbf{g}) > l$, then there is $\mathbf{g}' \to \mathbf{F}$ with $e(\mathbf{g}') < e(\mathbf{g})$ and $v(\mathbf{g}') < v(\mathbf{g})$.*

It follows that $\hat{r}(\mathbf{F}) = \hat{r}_l(\mathbf{F})$, where $\hat{r}_l(\mathbf{F}) = \min\{e(\mathbf{g}) : \mathbf{g} \to \mathbf{F},\, v(\mathbf{g}) \le l\}$.

Proof. Let $d(x) < d_0$. Choose $\delta > 0$ with $\delta + d_i d(x)/d_0 < d_i$ for any $i \in [q]$. Define the weight $\mathbf{g}'$ on $V(\mathbf{g}) \setminus \{x\}$ by $g'_A = g_A + g_{A\cup\{x\}}$, $A \in 2^{V(\mathbf{g}')}$. Clearly, $e(\mathbf{g}') = e(\mathbf{g}) - d(x) < e(\mathbf{g})$.

We claim that $\mathbf{g}'$ arrows $\mathbf{F}$. Suppose that this is not true, and let $\mathbf{c}'$ be an $\mathbf{F}$-free r-coloring of $\mathbf{g}'$. We can assume that

$$\sum_{A_1\cup\dots\cup A_r = A} c'_{A_1,\dots,A_r} \le g'_A + \delta/2^{v(\mathbf{g}')} \quad \text{for any } A \in 2^{V(\mathbf{g}')}.$$

Define $\mathbf{c}$ by

$$c_{A_1,\dots,A_r} = \begin{cases} \frac{\lambda_{A\setminus\{x\}} d_i}{d_0} \cdot c'_{A_1,\dots,A_{i-1},A_i\setminus\{x\},A_{i+1},\dots,A_r}, & x \in A_i,\ i \in [q],\\ 0, & x \in A_{q+1}\cup\dots\cup A_r,\\ (1-\lambda_A)\cdot c'_{A_1,\dots,A_r}, & x \notin A, \end{cases}$$

where we denote $A = A_1 \cup \dots \cup A_r$, $\lambda_A = g_{A\cup\{x\}}/g'_A$ if $g'_A > 0$, and $\lambda_A = 1/2$ if $g'_A = 0$. The reader can check that $\mathbf{c}$ is an r-coloring of $\mathbf{g}$.

By the assumption on $\mathbf{g}$, we have $\mathbf{F}_i \subset \mathbf{c}_i$ for some $i \in [r]$. However, this embedding cannot use x because for $i \in [q+1,r]$ we have $d_{\mathbf{c}_i}(x) = 0$ while for $i \in [q]$

$$\begin{aligned} d_{\mathbf{c}_i}(x) &= \sum_{A_1,\dots,A_r \subset V(\mathbf{g}')} c_{A_1,\dots,A_{i-1},A_i\cup\{x\},A_{i+1},\dots,A_r} \;=\; \sum_{A\in 2^{V(\mathbf{g}')}} \frac{\lambda_A d_i}{d_0} \sum_{A_1\cup\dots\cup A_r=A} c'_{A_1,\dots,A_r}\\ &\le \sum_{A\in 2^{V(\mathbf{g}')}} \frac{\lambda_A d_i}{d_0}\left(g'_A + \delta/2^{v(\mathbf{g}')}\right) \;\le\; \frac{d_i\delta}{d_0} + \frac{d_i}{d_0}\sum_{A\in 2^{V(\mathbf{g}')}} g_{A\cup\{x\}} \;\le\; \delta + d_i\frac{d(x)}{d_0} \;<\; d_i \end{aligned}$$

is too small; see (2.1). However, $c_{i,A} \le c'_{i,A}$ for $A \in 2^{V(\mathbf{g}')}$; hence, $\mathbf{F}_i \subset \mathbf{c}'_i$, which is the desired contradiction proving the first claim.

Let $v(\mathbf{g}) > l$. If $e(\mathbf{g}) \ge \hat{r}(\mathbf{F}) + d_0$, replace $\mathbf{g}$ by any other arrowing weight with $e(\mathbf{g}) < \hat{r}(\mathbf{F}) + d_0$. As $e(\mathbf{g})/(l+1) < d_0$, we can eventually ensure that $v(\mathbf{g}) \le l$ by iterating the procedure which proved the first claim. □

Hence, to compute $\hat{r}(\mathbf{F})$ it is enough to consider $\mathbf{F}$-arrowing weights on $L = [l]$ only.

Lemma 3.2. *There exists $\mathbf{g} \to \mathbf{F}$ with $V(\mathbf{g}) \subset L$ and $e(\mathbf{g}) = \hat{r}(\mathbf{F})$. (We call such a weight* extremal.*)*

Proof. Choose $\mathbf{g}_n \to \mathbf{F}$ with $V(\mathbf{g}_n) \subset L$, $n \in \mathbb{N}$, such that $e(\mathbf{g}_n)$ approaches $\hat{r}(\mathbf{F})$. By choosing a subsequence, assume that $V(\mathbf{g}_n)$ is constant and $g_A = \lim_{n\to\infty} g_{n,A}$ exists for each $A \in 2^L$. Clearly, $e(\mathbf{g}) = \hat{r}(\mathbf{F})$ so it remains to show that $\mathbf{g} \to \mathbf{F}$.

Let $\mathbf{c}$ be an r-coloring of $\mathbf{g}$. Let δ be the smallest slack in inequalities (2.2). Choose sufficiently large n so that $|g_{n,A} - g_A| < \delta$ for all $A \in 2^L$. We have

$$\sum_{A_1\cup\cdots\cup A_r = A} c_{A_1,\ldots,A_r} \ge g_A + \delta > g_{n,A}, \quad A \in 2^L;$$

that is, $\mathbf{c}$ is a coloring of $\mathbf{g}_n$ as well. Hence, $\mathbf{F}_i \subset \mathbf{c}_i$ for some i, as required. □

Now we are ready to prove our general theorem. The proof essentially takes care of itself. We just exploit the parallels between weights and graphs, which, unfortunately, requires messing around with various constants.

Theorem 3.3. *Let $(F_{i,n})_{n\in\mathbb{N}}$ be a dilatation of $\mathbf{f}_i$, $i \in [q]$, and let F_i be a fixed bipartite graph, $i \in [q+1, r]$. Then, for all sufficiently large n,*

$$\hat{r}(\mathbf{F})n - M(1+f_0) \le \hat{r}(F_{1,n},\ldots,F_{q,n},F_{q+1},\ldots,F_r) \le \hat{r}(\mathbf{F})n + M(1+f_0), \tag{3.2}$$

where $f_0 = \max\{\,\big||F^A_{i,n}| - f_{i,A}n\big| : i \in [q],\ A \in V(\mathbf{f}_i)\}$ and $M = M(\mathbf{F})$ is some constant.

In particular, the limit $\lim_{n\to\infty} \hat{r}(F_{1,n},\ldots,F_{q,n},F_{q+1},\ldots,F_r)/n$ exists.

Proof. Let $v_0 = \max\{v(F_i) : i \in [r]\}$, $m_1 = 2^{v_0}(f_0+1)$, and $m_2 = r^l m_1 + 1$, where, as before, $l > \hat{r}(\mathbf{F})/d_0$.

We prove that

$$\hat{r}(F_{1,n},\ldots,F_{q,n},F_{q+1},\ldots,F_r) \le \hat{r}(\mathbf{F})n + 2^l l(m_2+1), \quad n \ge 1. \tag{3.3}$$

By Lemma 3.2 choose an extremal weight $\mathbf{g}$ on L. Define a bipartite graph G as follows. Choose disjoint from each other (and from L) sets G^A with $|G^A| = \lceil g_A n + m_2 \rceil$, $A \in 2^L$. Let $V(G) = L \cup (\cup_{A\in 2^L} G^A)$. In G we connect $x \in L$ to everything in G^A if $x \in A$. These are all the edges. Clearly,

$$e(G) = \sum_{A\in 2^L} |G^A|\,|A| \le 2^l l(m_2+1) + \sum_{A\in 2^L} g_A n\,|A| \le 2^l l(m_2+1) + \hat{r}(\mathbf{F})n,$$

as required. Hence, it is enough to show that G has the arrowing property.

Consider any r-coloring $c : E(G) \to [r]$. For every r-tuple of disjoint sets $B_1,\ldots,B_r \subset L$, let

$$C_{B_1,\ldots,B_r} = \{y \in G^B : \forall i \in [r]\ \forall x \in B_i\ \ c(\{x,y\}) = i\},$$
$$c_{B_1,\ldots,B_r} = \begin{cases} (|C_{B_1,\ldots,B_r}| - m_1)/n & \text{if } |C_{B_1,\ldots,B_r}| \ge m_1, \\ 0 & \text{otherwise,} \end{cases}$$

where $B = B_1 \cup \cdots \cup B_r$. In any case, $nc_{B_1,\dots,B_r} \geq |C_{B_1,\dots,B_r}| - m_1$; hence, for every $B \in 2^L$ we have

$$n \sum_{B_1 \cup \cdots \cup B_r = B} c_{B_1,\dots,B_r} \geq -r^{|B|} m_1 + \sum_{B_1 \cup \cdots \cup B_r = B} |C_{B_1,\dots,B_r}| \geq -r^l m_1 + |G^B| > ng_B;$$

that is, $\mathbf{c}$ is an r-coloring of $\mathbf{g}$. Hence, $\mathbf{F}_i \subset \mathbf{c}_i$ for some $i \in [r]$. Now we show that G contains a forbidden subgraph in the ith color.

Suppose that $i \in [q]$. By definition, we find appropriate $h : V(\mathbf{f}_i) \to L$ and $\mathbf{w}$. We aim at proving that $F_{i,n} \subset G_i$, where $G_i \subset G$ is the color-i subgraph. Partition $F_{i,n}^A = \cup_{B \supset h(A)} W_{A,B}$ so that $W_{A,B} = \emptyset$ if $w_{A,B} = 0$ and $|W_{A,B}| \leq \lfloor w_{A,B} n + f_0 + 1 \rfloor$, $A \in 2^{V(\mathbf{f}_i)}$, $B \in 2^L$. This is possible for any A: if $w_{A,B} = 0$ for all $B \in 2^L$ with $h(A) \subset B$, then $f_{i,A} = 0$ and $F_{i,n}^A = \emptyset$; if $w_{A,B} > 0$ for at least one B, then

$$\sum_{\substack{B \in 2^L \\ w_{A,B} > 0}} (w_{A,B} n + f_0) \geq f_0 + n \sum_{\substack{B \in 2^L \\ w_{A,B} > 0}} w_{A,B} \geq f_0 + f_{i,A} n \geq |F_{i,n}^A|.$$

Let $B \in 2^L$. If $c_{i,B} = 0$, then $w_{AB} = 0$ and $W_{A,B} = \emptyset$ for all $A \in 2^{V(\mathbf{f}_i)}$. Otherwise,

$$nc_{i,B} = n \sum_{\substack{B_1,\dots,B_r \\ B_i = B}} c_{B_1,\dots,B_i} \leq -m_1 + \sum_{\substack{B_1,\dots,B_r \\ B_i = B}} |C_{B_1,\dots,B_i}| = |G_i^B| - m_1,$$

and we have

$$\sum_{\substack{A \in 2^{V(F_{i,n})} \\ h(A) \subset B}} |W_{A,B}| \leq \sum_{\substack{A \in 2^{V(F_{i,n})} \\ h(A) \subset B}} (w_{A,B} n + f_0 + 1) \leq c_{i,B} n + 2^{v_0}(f_0 + 1) \leq |G_i^B|.$$

Hence, we can extend $h : V_1(F_{i,n}) \to L \subset V(G_i)$ to the whole of $V(F_{i,n})$ by mapping $\cup_{h(A) \subset B} W_{A,B}$ injectively into G_i^B, which proves that $F_{i,n} \subset G_i$.

Suppose that $i \in [q+1, r]$. The relation $F_i \subset \mathbf{c}_i$ means that there exist appropriate $V_1(F_i) \cup V_2(F_i) = V(F_i)$ and $h : V_1(F_i) \to L$. We view h as a partial embedding of F_i into G_i and extend h to the whole of $V(F_i)$.

Take consecutively $y \in V_2(F_i)$. There is $B_i \subset L$ such that $c_{i,B_i} > 0$ and $h(\Gamma(y)) \subset B_i$. The inequality $c_{i,B_i} > 0$ implies that there are disjoint B_j's, $j \in [r] \setminus \{i\}$, such that $c_{B_1,\dots,B_r} > 0$. Each vertex in $C_{B_1,\dots,B_r}$ is connected by color i to the whole of $B_i \supset h(\Gamma(y))$. The inequality $c_{B_1,\dots,B_r} > 0$ means that $|C_{B_1,\dots,B_r}| \geq m_1 \geq v(F_i)$, so we can always extend h to y; that is, we find an F_i-subgraph of color i.

Thus the constructed graph G has the desired arrowing property, which proves the upper bound.

Let $d' = \min_{i \in [q]} \min_{x \in V(\mathbf{f}_i)} d_{\mathbf{f}_i}(x) > 0$, $l' = 5ld_0/d'$, $m_3 = \max(r^{l'}, 2^{v_0}(f_0 + l'))$. As the lower bound, we show that, for all sufficiently large n,

$$\hat{r}(F_{1,n}, \dots, F_{q,n}, F_{q+1}, \dots, F_r) \geq \hat{r}(\mathbf{F})n - 2^{l'} l' m_3. \tag{3.4}$$

Choose any asymptotically minimum graph G with the arrowing property. Let $L \subset V(G)$ be the set of vertices of degree at least $d'n/2$ in G. From $d'n|L|/4 < e(G) < (1 + o(1))ld_0 n$, it follows that $|L| \leq l'$ (assuming that n is sufficiently large). For $A \in 2^L$, define $g_A = (|G^A| + m_3)/n$, where $G_A = \{x \in V(G) \setminus L : \Gamma(x) \cap L = A\}$.

Claim 1. $\mathbf{g} \to \mathbf{F}$.

Suppose, on the contrary, that there is an $\mathbf{F}$-free r-coloring $\mathbf{c}$ of $\mathbf{g}$. We are going to exhibit a contradictory r-coloring of $E(G)$.

For each $B \in 2^L$ choose any disjoint sets $C_{B_1,\dots,B_r} \subset G^B$ (indexed by r-tuples of disjoint sets partitioning B) such that they partition G^B and

$$|C_{B_1,\dots,B_r}| \le \lfloor c_{B_1,\dots,B_r} \cdot n \rfloor. \tag{3.5}$$

This is possible because

$$\sum_{B_1 \cup \dots \cup B_r = B} \lfloor c_{B_1,\dots,B_r} \cdot n \rfloor \ge g_B n - r^{l'} \ge |G^B|.$$

For $j \in [r]$, $x \in B_j$, and $y \in C_{B_1,\dots,B_r}$, color the edge $\{x, y\} \in E(G)$ with color j. All the remaining edges of G (namely, those lying inside L or inside $V(G) \setminus L$) are colored with color 1.

There is $i \in [r]$ such that $G_i \subset G$, the color-i subgraph, contains a forbidden subgraph.

Suppose that $i \in [q]$. Let $h : F_{i,n} \to G_i$ be an embedding. If n is large, then

$$d(x) \ge d'n + o(n) > d'n/2, \quad x \in V_1(F_{i,n}),$$

which implies that $h(V_1(F_{i,n})) \subset L$. Define, for $A \in 2^{V(\mathbf{f}_i)}$ and $B \in 2^L$ with $B \supset h(A)$ and $f_{i,A} \neq 0$,

$$w_{A,B} = \frac{|h^{-1}(G^B) \cap F_{i,n}^A| + f_0 + l'}{n}.$$

All other $w_{A,B}$'s are set to zero. For $A \in 2^{V(\mathbf{f}_i)}$ with $f_{i,A} \neq 0$, we have

$$\sum_{\substack{B \in 2^L \\ B \supset h(A)}} w_{A,B} \ge \frac{|F_{i,n}^A \cap h^{-1}(V(G) \setminus L)| + f_0 + l'}{n} \ge \frac{|F_{i,n}^A| + f_0}{n} \ge f_{i,A}.$$

For $B \in 2^L$ we have

$$\sum_{\substack{A \in 2^{V(\mathbf{f}_i)} \\ h(A) \subset B}} w_{A,B} \le \frac{2^{v_0}(f_0 + l')}{n} + \sum_{\substack{A \in 2^{V(\mathbf{f}_i)} \\ h(A) \subset B}} \frac{|h^{-1}(G^B) \cap F_{i,n}^A|}{n} \le \frac{2^{v_0}(f_0 + l')}{n} + \frac{|G^B|}{n} \le g_B;$$

that is, h (when restricted to $V(\mathbf{f}_i)$) and $\mathbf{w}$ demonstrate that $\mathbf{f}_i \subset \mathbf{c}_i$, which is a contradiction.

Suppose that $i \in [q+1, r]$. Let $V_1(F_i)$ consist of those vertices which are mapped by $h : F_i \to G_i$ into L, and let $V_2(F_i) = V(F_i) \setminus V_1(F_i)$. This is a legitimate bipartition of F_i because any color-i edge of G connects L to $V(G) \setminus L$. Let $y \in V_2(F_i)$. The sets $C_{B_1,\dots,B_r}$ partition $V(G) \setminus L$; let $y \in C_{B_1,\dots,B_r}$. By (3.5) we have $c_{B_1,\dots,B_r} > 0$. However, $h(\Gamma(y)) \subset B_i$, which shows that $F_i \subset \mathbf{g}_i$. This contradiction proves Claim 1.

Hence, $\mathbf{g} \to \mathbf{F}$ and we have

$$\hat{r}(\mathbf{F}) \le \sum_{A \in 2^L} g_A\,|A| \le \frac{2^{l'} l' m_3}{n} + \frac{1}{n} \sum_{A \in 2^L} |G^A|\,|A| \le \frac{2^{l'} l' m_3 + e(G)}{n},$$

which implies the desired inequality (3.4). □

A moment's thought on Claim 1 reveals the following "characterization" of extremal graphs.

THEOREM 3.4. *Let* $\mathbf{F}$ *and the* F*'s be as in Theorem* 3.3, *and let*

$$G_n \to (F_{1,n}, \dots, F_{q,n}, F_{q+1}, \dots, F_r), \quad n \in \mathbb{N},$$

be any sequence of asymptotically minimum graphs. Then there is an extremal weight $\mathbf{g} \to \mathbf{F}$ *and an increasing sequence* $(n_i)_{i\in\mathbb{N}}$ *such that, up to removing* $o(n_i)$ *edges and relabelling vertices,* G_{n_i} *can be made into a bipartite graph with* $V_1(G_{n_i}) = V(\mathbf{g})$ *and* $\lim_{i\to\infty} |G^A_{n_i}|/n_i = g_A$ *for each* $A \in 2^{V(\mathbf{G})}$.

In particular, if $\mathbf{g} \to \mathbf{F}$ *is the unique extremal weight, then we can take* $n_i = i$. □

4. Complete bipartite graphs. Here we will compute asymptotically the size Ramsey number if each forbidden graph is a complete bipartite graph.

THEOREM 4.1. *Let* $r \geq 2$ *and* $q \geq 1$. *Suppose that we are given* $t_1, \dots, t_q \in \mathbb{R}_{>0}$ *and* $s_1, \dots, s_r, t_{q+1}, \dots, t_r \in \mathbb{N}$ *with* $t_i \geq s_i$ *for* $i \in [q+1, r]$. *Then there exist* $s \in \mathbb{N}$ *and* $t \in \mathbb{R}_{>0}$ *such that* $\mathbf{k}_{s,t} \to \mathbf{F}$ *and* $\hat{r}(\mathbf{F}) = e(\mathbf{k}_{s,t}) = st$, *where*

$$\mathbf{F} = (\mathbf{k}_{s_1,t_1}, \dots, \mathbf{k}_{s_q,t_q}, K_{s_{q+1},t_{q+1}}, \dots, K_{s_r,t_r}).$$

Proof. Let us first describe an algorithm finding extremal s and t. Some by-product information gathered by our algorithm will be used in the proof of the extremality of $\mathbf{k}_{s,t} \to \mathbf{F}$.

Choose $l \in \mathbb{N}$ bigger than $\hat{r}(\mathbf{F})/t_0$, where $t_0 = \sum_{i=1}^q t_i$, which is the same definition as that before Lemma 3.1.

We claim that $l > \sigma$, where $\sigma = \sum_{i=1}^r (s_i - 1)$. Indeed, take any extremal $\mathbf{g} \to \mathbf{F}$ without isolated vertices. Lemma 3.1 implies that $d(x) \geq t_0$ for any $x \in V(\mathbf{g})$. Also, it is easy to see that $v(\mathbf{g}) > \sigma$. Hence, $l \geq \hat{r}(\mathbf{F})/t_0 \geq v(\mathbf{g}) > \sigma$, as claimed.

For each integer $s \in [\sigma+1, l]$ let $t'_s > 0$ be the infimum of $t \in \mathbb{R}$ such that $\mathbf{k}_{s,t} \to \mathbf{F}$. Also, let Π_s be the set of all sequences $\mathbf{a} = (a_1, \dots, a_r)$ of nonnegative integers with $a_i = s_i - 1$ for $i \in [q+1, r]$ and $\sum_{i=1}^r a_i = s$. For a sequence $\mathbf{a} = (a_1, \dots, a_r)$ and a set A of size $\sum_{i=1}^r a_i$, let $\binom{A}{\mathbf{a}}$ consist of all sequences $\mathbf{A} = (A_1, \dots, A_r)$ of sets partitioning A with $|A_i| = a_i$, $i \in [r]$.

We claim that t'_s is $sol(L_s)$, the extremal value of the following linear program L_s: "Find $sol(L_s) = \max \sum_{\mathbf{a}\in\Pi_s} u_{\mathbf{a}}$ over all sequences $(u_{\mathbf{a}})_{\mathbf{a}\in\Pi_s}$ of nonnegative reals such that

$$\sum_{\mathbf{a}\in\Pi_s} u_{\mathbf{a}} \binom{a_i}{s_i} \leq t_i \binom{s}{s_i} \quad \forall i \in [q].\text{''} \tag{4.1}$$

Claim 1. The weight $\mathbf{k}_{s,t}$ does not arrow $\mathbf{F}$ for $t < sol(L_s)$.

To prove this, let

$$\lambda = \frac{t + sol(L_s)}{2sol(L_s)} < 1 \quad \text{and} \quad \epsilon = \frac{1-\lambda}{2^{s+1}} \min\{t_i : i \in [q]\} > 0.$$

Let $V(\mathbf{k}_{s,t}) = [s]$. Define an r-coloring $\mathbf{c}$ of $\mathbf{k}_{s,t}$ by

$$c_{\mathbf{A}} = \frac{\lambda u_{|A_1|,\dots,|A_r|}}{\binom{s}{|A_1|,\dots,|A_r|}}, \quad \mathbf{a} \in \Pi_s,\ \mathbf{A} \in \binom{[s]}{\mathbf{a}}, \tag{4.2}$$

$c_{B,\emptyset,\dots,\emptyset} = \epsilon$, $B \subsetneq [s]$, while all other c's are zero. It is indeed a coloring of $\mathbf{k}_{s,t}$:

$$\sum_{\mathbf{a}\in\Pi_s} \sum_{\mathbf{A}\in\binom{[s]}{\mathbf{a}}} c_{\mathbf{A}} = \sum_{\mathbf{a}\in\Pi_s} \lambda u_{\mathbf{a}} = \lambda\, sol(L_s) > t.$$

We have $\mathbf{k}_{s_i,t_i} \not\subset \mathbf{c}_i$ for $i \in [q]$: for example, for $i = 1$ and any $S \in \binom{[s]}{s_1}$, we have

$$\sum_{\mathbf{a}\in\Pi_s} \sum_{\substack{\mathbf{A}\in\binom{[s]}{\mathbf{a}}\\ A_1\supset S}} c_{\mathbf{A}} = \sum_{\substack{\mathbf{a}\in\Pi_s\\ a_1\ge s_1}} \frac{\binom{s-s_1}{a_1-s_1,a_2,\dots,a_r}\lambda u_{\mathbf{a}}}{\binom{s}{a_1,\dots,a_r}} = \lambda \sum_{\mathbf{a}\in\Pi_s} \frac{\binom{a_1}{s_1}u_{\mathbf{a}}}{\binom{s}{s_1}} \le \lambda t_1 < t_1 - \sum_{\substack{B\subsetneq[s]\\ B\supset S}} c_{B,\emptyset,\dots,\emptyset}.$$

Also, $K_{s_i,t_i} \not\subset \mathbf{c}_i$ for $i \in [q+1,r]$ because $c_{A_1,\dots,A_r} = 0$ whenever $|A_i| \ge s_i$ for some $i \in [q+1,r]$. Claim 1 is proved.

Claim 2. $\mathbf{k}_{s,t} \to \mathbf{F}$ for any $t > sol(L_s)$.

Suppose that the claim is not true and we can find an $\mathbf{F}$-free r-coloring $\mathbf{c}$ of $\mathbf{k}_{s,t}$. By definition, $c_{A_1,\dots,A_r} = 0$ whenever $|A_i| \ge s_i$ for some $i \in [q+1,r]$. If some $c_{A_1,\dots,A_r} = c > 0$ with $|A_i| \le s_i - 2$ for some $i \in [q+1,r]$, then $A_j \ne \emptyset$ for some $j \in [q]$, so we can pick $x \in A_j$ and set $c_{A_1,\dots,A_r} = 0$ while increasing $c_{\dots,A_j\setminus\{x\},\dots,A_i\cup\{x\},\dots}$ by c. Clearly, $\mathbf{c}$ remains $\mathbf{F}$-free. Thus, we can assume that all the c's are zero except those of the form $c_{\mathbf{A}}$, $\mathbf{A} \in \binom{[s]}{\mathbf{a}}$ for some $\mathbf{a} \in \Pi_s$. Now, tracing back our proof of Claim 1, we obtain a feasible solution $u_{\mathbf{a}} = \sum_{\mathbf{A}\in\binom{[s]}{\mathbf{a}}} c_{\mathbf{A}}$, $\mathbf{a} \in \Pi_s$, to L_s with a larger objective function, which is a contradiction. The claim is proved.

Thus, $t'_s = sol(L_s)$ and $m_u = \min\{st'_s : s \in [\sigma+1,l]\}$ is an upper bound on $\hat{r}(\mathbf{F})$. Let us show that in fact $\hat{r}(\mathbf{F}) = m_u$.

We rewrite the definition of $\hat{r}(\mathbf{F})$ so that we can apply the Farkas lemma. The verification of the following easy claim is left to the reader.

Claim 3. $\hat{r}(\mathbf{F}) = \inf e(\mathbf{g})$ over all weights $\mathbf{g}$ on $L = [l]$ such that there do not exist nonnegative reals $(c_{\mathbf{A}})_{\mathbf{A}\in\binom{A}{\mathbf{a}},\,\mathbf{a}\in\Pi_{|A|},\,A\in 2^L}$ with the following properties:

$$\sum_{\mathbf{a}\in\Pi_{|A|}} \sum_{\mathbf{A}\in\binom{A}{\mathbf{a}}} c_{\mathbf{A}} \ge g_A, \quad A \in 2^L,$$

$$\sum_{A\in 2^L} \sum_{\mathbf{a}\in\Pi_{|A|}} \sum_{\substack{\mathbf{A}\in\binom{A}{\mathbf{a}}\\ A_i\supset S}} c_{\mathbf{A}} \le t_i, \quad i \in [q],\ S \in \binom{L}{s_i}.$$

Let $\mathbf{g}$ be any feasible solution to the above problem. By the Farkas lemma there exist $x_A \ge 0$, $A \in 2^L$, and $y_{i,S} \ge 0$, $i \in [q]$, $S \in \binom{L}{s_i}$, such that

$$\sum_{i=1}^{q} \sum_{S\in\binom{A_i}{s_i}} y_{i,S} \ge x_A, \quad A \in 2^L,\ \mathbf{a} \in \Pi_{|A|},\ \mathbf{A} \in \binom{A}{\mathbf{a}}, \tag{4.3}$$

$$\sum_{i=1}^{q} t_i \sum_{S\in\binom{L}{s_i}} y_{i,S} < \sum_{A\in 2^L} g_A x_A. \tag{4.4}$$

We deduce that $x_A \le 0$ (and hence $x_A = 0$) if $|A| \le \sigma$ by considering (4.3) for some $\mathbf{A}$ with $|A_i| \le s_i - 1$ for each $i \in [r]$.

For each A with $a := |A| > \sigma$ repeat the following. Let $(u_{\mathbf{a}})_{\mathbf{a}\in\Pi_a}$ be an extremal solution to L_a. For each $\mathbf{a}\in\Pi_a$, take the average of (4.3) over all $\mathbf{A}\in\binom{A}{\mathbf{a}}$, multiply it by $u_{\mathbf{a}}$, and add all these inequalities together to obtain the following:

$$\begin{aligned} x_A t'_a &= \sum_{\mathbf{a}\in\Pi_a} u_{\mathbf{a}} x_A \ \le \sum_{\mathbf{a}\in\Pi_a} \frac{u_{\mathbf{a}}}{\binom{a}{a_1,\dots,a_r}} \sum_{\mathbf{A}\in\binom{A}{\mathbf{a}}} \sum_{i=1}^{q} \sum_{S\in\binom{A_i}{s_i}} y_{i,S} \\ &= \sum_{i=1}^{q} \sum_{S\in\binom{A}{s_i}} y_{i,S} \sum_{\substack{\mathbf{a}\in\Pi_a\\ a_i\ge s_i}} \frac{u_{\mathbf{a}}\binom{a-s_i}{a_1,\dots,a_{i-1},a_i-s_i,a_{i+1},\dots,a_q}}{\binom{a}{a_1,\dots,a_r}} \\ &= \sum_{i=1}^{q} \sum_{S\in\binom{A}{s_i}} y_{i,S} \sum_{\substack{\mathbf{a}\in\Pi_a\\ a_i\ge s_i}} \frac{u_{\mathbf{a}}\binom{a_i}{s_i}}{\binom{a}{s_i}} \ \le \ \sum_{i=1}^{q} t_i \sum_{S\in\binom{A}{s_i}} y_{i,S}. \end{aligned}$$

(In the last inequality we used (4.1).)

Substituting the obtained inequalities on the x_A's into (4.4) we obtain

$$\sum_{i=1}^{q} t_i \sum_{S\in\binom{L}{s_i}} y_{i,S} < \sum_{\substack{A\in 2^L\\ |A|>\sigma}} \frac{g_A}{t'_{|A|}} \sum_{i=1}^{q} t_i \sum_{S\in\binom{A}{s_i}} y_{i,S}.$$

As the $y_{i,S}$'s are nonnegative, one of these variables has a larger coefficient on the right-hand side. Let it be $y_{i,S}$. We have

$$t_i < t_i \sum_{\substack{A\in\binom{L}{>\sigma}\\ A\supset S}} \frac{g_A}{t'_{|A|}} \le \frac{t_i}{m_u} \sum_{A\in 2^L} g_A|A|. \tag{4.5}$$

The last inequality follows, by comparing coefficients at each g_A, from the fact that for any integer $a > \sigma$ we have $1/t'_a \le a/m_u$ by the definition of m_u. Hence, $e(\mathbf{g}) = \sum_{A\in 2^L} a_A|A| > m_u$, as required. □

Corollary 4.2. *Let $r \ge q \ge 1$, $t_1,\dots,t_q \in \mathbb{R}_{>0}$ and $s_1,\dots,s_r,t_{q+1},\dots,t_r \in \mathbb{N}$ with $t_i \ge s_i$ for $i\in[q+1,r]$. For $i\in[q]$, let $(t_{i,n})_{n\in\mathbb{N}}$ be an integer sequence with $t_{i,n} = t_i n + o(n)$. Define*

$$\mathbf{F}_n = (K_{s_1,t_{1,n}},\dots,K_{s_q,t_{q,n}},K_{s_{q+1},t_{q+1}},\dots,K_{s_r,t_r}).$$

Let $l\in\mathbb{N}$ be larger than $\lim_{n\to\infty}\hat{r}(\mathbf{F}_n)/(t_0 n)$, where $t_0 = \sum_{i=1}^{q} t_i$. Then

$$\lim_{n\to\infty}\frac{\hat{r}(\mathbf{F}_n)}{n} = \lim_{n\to\infty}\frac{\min\{e(K_{s,t}) : s\le l,\ K_{s,t}\to\mathbf{F}_n\}}{n}. \tag{4.6}$$ □

In other words, in order to compute the limit in Corollary 4.2, it is sufficient to consider only complete bipartite graphs arrowing $\mathbf{F}_n$. It seems that there is no simple general formula, but the proof of Theorem 4.1 gives an algorithm for computing $\hat{r}(\mathbf{F})$. The author has realized the algorithm as a C program which calls the `lp_solve 3.2` library. (The latter is a freely available linear programming software, currently maintained by Berkelaar [4]). Later, Avis rewrote the program to be linked with his `lrslib 4.1` library [1]. The latter library has the advantage that its arithmetic is exact (while `lp_solve` operates with reals), so that any computed limit can be considered as proved. The reader is welcome to experiment with the program;

Table 4.1
Values of $\lim_{n\to\infty}\hat{r}(K_{s,n},K_{t,n})/n$ *obtained with the* `lrslib` *library of Avis.*

1	2							
2	6	18						
3	12	40	98					
4	20	75	$182\frac{14}{19}$	363				
5	30	$118\frac{10}{17}$	$310\frac{19}{62}$	$638\frac{44}{47}$	1156			
6	42	$172\frac{4}{5}$	$469\frac{6}{7}$	$1023\frac{23}{87}$	$1952\frac{15}{22}$	$3350\frac{1}{3}$		
7	56	$241\frac{7}{23}$	$678\frac{4}{11}$	$1538\frac{36}{55}$	$3030\frac{1}{2}$	$5456\frac{92}{209}$	$9120\frac{42}{55}$	
8	72	$320\frac{4}{7}$	$938\frac{2}{5}$	$2211\frac{579}{1573}$	$4517\frac{317}{504}$	$8426\frac{176}{221}$	$14523\frac{595}{4693}$	$23781\frac{7}{34}$
(s,t)	1	2	3	4	5	6	7	8

its source can be found in [15]. Here, in Table 4.1, we present the asymptotics of $\hat{r}(K_{s,n},K_{t,n})$ for $1 \le s \le t \le 8$. Unfortunately, the number of iterations (which is approximately $\frac{1}{2}\lim\hat{r}(K_{s,n},K_{t,n})/n$) increases rapidly with s and t.

For certain series of parameters we can get a more explicit expression. First, let us treat the case $q=1$, that is, when only the first forbidden graph dilates with n. We can assume that $t_1 = 1$ by scaling n.

Theorem 4.3. *Let $q=1$ and $r\ge 2$. Then for any $s_1,\dots,s_r,t_2,\dots,t_r\in\mathbb{N}$ with $t_i\ge s_i$, $i\in[2,r]$, we have*

$$\hat{r}(K_{s_1,n},K_{s_2,t_2},\dots,K_{s_r,t_r}) = n\cdot\min\left\{s\,\frac{(s)_{s_1}}{(s-s')_{s_1}} : s\in\mathbb{N}_{>\sigma}\right\}+O(1),$$

where $s'=\sigma-s_1+1$, $\sigma=\sum_{i=1}^{r}(s_i-1)$, and $(s)_k = s(s-1)\dots(s-k+1)$.

Proof. The problem L_s has only one variable $u_{s-s',s_2-1,\dots,s_r-1}$. Trivially, $t'_s = \binom{s}{s_1}/\binom{s-s'}{s_1} = (s)_{s_1}/(s-s')_{s_1}$, and the theorem follows. □

In the case $s_1=1$ we obtain the following formula.

Corollary 4.4. *For any $s_2,\dots,s_r,t_2,\dots,t_r\in\mathbb{N}$ with $t_i\ge s_i$, $i\in[2,r]$, we have*

$$\hat{r}(K_{1,n},K_{s_2,t_2},\dots,K_{s_r,t_r}) = 4\left(1-r+\sum_{i=2}^{r}s_i\right)n+O(1).$$

Proof. By Theorem 4.3, we have to compute $\min_{s>s'}\frac{s^2}{s-s'}$, where $s'=\sum_{i=2}^{r}(s_i-1)$. The differentiation $\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{s^2}{s-s'}\right) = \frac{s(s-2s')}{(s-s')^2}$ shows that the minimum is attained for $s=2s'$. □

Another case with a simple formula for $\hat{r}(\mathbf{F})$ is $q=2$, $s_1=s_2$, and $t_1=t_2$. Again, without loss of generality we can assume that $t_1=t_2=1$.

Theorem 4.5. *Let $q=2$ and $r\ge 2$. Then for any $s,s_3,\dots,s_r,t_3,\dots,t_r\in\mathbb{N}$ with $t_i\ge s_i$, $i\in[3,r]$, we have*

$$\hat{r}(K_{s,n},K_{s,n},K_{s_3,t_3},\dots,K_{s_r,t_r}) = n\cdot\min\left\{a\cdot f(a) : a\in\mathbb{N}_{>\sigma}\right\}+O(1), \tag{4.7}$$

where $\sigma = 2s-r+\sum_{i=3}^{r}s_i$ and

$$f(a) = \frac{2\binom{a}{s}}{\binom{\lfloor a'/2\rfloor}{s}+\binom{\lceil a'/2\rceil}{s}},$$

with $a' = a - \sum_{i=3}^{r}(s_i - 1)$.

Proof. Let $a \in \mathbb{N}_{>\sigma}$, and let $(u_{\mathbf{a}})_{\mathbf{a}\in\Pi_a}$ be an extremal solution to L_a (where we obviously define $s_1 = s_2 = s$ and $t_1 = t_2 = 1$). Excluding the constant indices in $u_{\mathbf{a}}$, we assume that the index set Π_a consists of pairs of integers (a_1, a_2) with $a_1 + a_2 = a'$.

Clearly, $(u'_{a_1,a_2})_{(a_1,a_2)\in\Pi_a}$ is also an extremal solution, where $u'_{a_1,a_2} = \frac{1}{2}(u_{a_1,a_2} + u_{a_2,a_1})$. Thus we can assume that $u_{a_1,a_2} = u_{a_2,a_1}$ for all $(a_1, a_2) \in \Pi_a$.

If $u_{a_1,a_2} = c > 0$ for some $a_1 < \lfloor a'/2 \rfloor$, then we can set $u_{a_1,a_2} = u_{a_2,a_1} = 0$ while increasing $u_{\lfloor a'/2\rfloor,\lceil a'/2\rceil}$ and $u_{\lceil a'/2\rceil,\lfloor a'/2\rfloor}$ by c. The easy inequality

$$\tbinom{b+1}{s} + \tbinom{a'-b-1}{s} - \tbinom{b}{s} - \tbinom{a'-b}{s} = \tbinom{b}{s-1} - \tbinom{a'-b-1}{s-1} < 0, \quad s - 1 \le b < \lfloor a'/2 \rfloor,$$

implies inductively that the left-hand side of (4.1) strictly decreases while the objective function $\sum_{\mathbf{a}\in\Pi_a} u_{\mathbf{a}}$ does not change, which clearly contradicts the minimality of $\mathbf{u}$.

Now we deduce that, for *any* extremal solution $(u_{\mathbf{a}})_{\mathbf{a}\in\Pi_a}$, we have $u_{a_1,a_2} = 0$ unless $\{a_1, a_2\} = \{\lfloor a'/2\rfloor, \lceil a'/2\rceil\}$; moreover, it follows that necessarily $u_{\lfloor a'/2\rfloor,\lceil a'/2\rceil} = u_{\lceil a'/2\rceil,\lfloor a'/2\rfloor}$. Hence, $t'_a = f(a)$, which proves the theorem. □

The special case $r = 2$ of Theorem 4.5 answers the question of Erdős et al. [8, Problem B], who asked for the value of

$$r_s = \lim_{n\to\infty} \frac{\hat{r}(K_{s,n}, K_{s,n})}{n}.$$

The formula (4.7), which now reads $r_s = \min_{a\ge 2s-1} af(a)$ with $f(a) = 2\binom{a}{s}/(\binom{\lfloor a/2\rfloor}{s} + \binom{\lceil a/2\rceil}{s})$, can be further simplified in this case as follows.

THEOREM 4.6. *For $s \ge 4$ we have $r_s = a_s f(a_s)$, where $a_s = 2\lfloor s(s+3)/4\rfloor - 3$.*

Proof. For any $b \ge s$ we have $f(2b) = f(2b-1)$; hence, the minimum of $af(a)$ is attained for an odd a:

$$r_s = \min_{b\ge s}\,(2b-1)f(2b-1) = 2\min_{b\ge s}\,(2b-1)\binom{2b-1}{s}\left(\binom{b-1}{s} + \binom{b}{s}\right)^{-1}.$$

We have $\binom{b-1}{s} + \binom{b}{s} = \frac{(b-1)!(2b-s)}{s!(b-s)!}$ and, as it is routine to check,

$$(2b+1)f(2b+1) - (2b-1)f(2b-1) = cp_s(b),$$

where $c = \frac{2(2b-1)!(b-s)!}{(2b-s+1)!(b-1)!(2b-s+2)}$ and

$$p_s(b) = 2(2b+1)^2(b-s+1) - (2b-1)(2b-s+1)(2b-s+2) = 8b^2 - 2bs^2 - 6bs + 12b + s^2 - 5s + 4.$$

The quadratic in b polymomial p_s has two roots: one is less than 1 (because $p_s(1) < 0$) and the other is bigger than s (because $p_s(s) < 0$). Thus, the function $(2b-1)f(2b-1)$, $b \ge s$, first decreases and then increases; its minimum is attained for b_s, the smallest integer $b \ge s$ with $p_s(b) > 0$. The value of b_s can be computed exactly:

$$b_s = \begin{cases} 4t^2 + 3t - 1, & s = 4t, \\ 4t^2 + 5t, & s = 4t+1, \\ 4t^2 + 7t + 1, & s = 4t+2, \\ 4t^2 + 9t + 3, & s = 4t+3. \end{cases}$$

For example, let us check the case $s \equiv 0 \pmod 4$:

$$p_s(4t^2 + 3t - 2) = -32t + 12 < 0 < p_s(4t^2 + 3t - 1) = 32t^2 - 8t.$$

Also, $2b_s - 1 = 2\lfloor s(s+3)/4 \rfloor - 3$ in each case, as required. □

Remark. For $4 \le s \le 8$, the values of r_s given by Table 4.1 and Theorem 4.6 coincide, which is reassuring.

The natural question of how to characterize all extremal weights in Theorem 4.1 arises. We have a partial answer as follows. Let $\mathbf{g} \to \mathbf{F}$ be extremal. We know by Lemma 3.1 that $v(\mathbf{g}) \le l$, so assume that $v(\mathbf{g}) \subset [l]$. It is easy to check that if we increase each g_A by some $\epsilon > 0$, then the obtained weight is a feasible solution to the system of Claim 3 from the proof of Theorem 4.1 and hence satisfies (4.5) for some i and S. As $\epsilon > 0$ is arbitrary and there are finitely many possible pairs (i, S), the weight $\mathbf{g}$ satisfies the *nonstrict* inequality (4.5) for some (i, S). As $\mathbf{g}$ is extremal, we have, in fact, an equality there. This implies that $g_A = 0$ unless $|A| \cdot t'_{|A|} = m_u$ and $A \supset S$.

However, in some cases we can get more precise information. As an example, consider $\mathbf{F} = (\mathbf{k}_{2,1}, \mathbf{k}_{2,1})$. Theorem 4.5 implies that $\hat{r}(\mathbf{F}) = 18$. However, we are able to show more.

Theorem 4.7. *$\mathbf{k}_{3,6} \to (\mathbf{k}_{2,1}, \mathbf{k}_{2,1})$ is the unique extremal weight. Also, there is n_0 such that, for all $n > n_0$, we have $\hat{r}(K_{2,n}, K_{2,n}) = 18n - 15$, and $K_{3,6n-5}$ and $K_3 + \overline{K}_{6n-6}$ are the only extremal graphs (up to isolated vertices).*

Proof. Let $\mathbf{g} \to (\mathbf{k}_{2,1}, \mathbf{k}_{2,1})$ have size 18 and no isolated vertices.

By Lemma 3.1 we have $v(\mathbf{g}) \le 9$. It is routine to check that $at'_a > 18$ for any $a \in [4, 9]$. Thus we know that, for some $S = \{x, y\} \subset L$, we have $g_A = 0$ whenever $|A| \ne 3$ or $A \not\supset S$. Let J be the set of those $j \in L$ with $g_{\{x,y,j\}} > 0$. We have $\sum_{j \in J} g_{\{x,y,j\}} = 6$. Suppose, on the contrary to the claim, that $\mathbf{g} \ncong \mathbf{k}_{3,6}$. Then we have $|J| \ge 2$.

Consider the 2-coloring $\mathbf{c}$ of $\mathbf{g}$ obtained by letting $c_{A_1,A_2} = 2^{-18}/10$ for all disjoint $A_1, A_2 \in 2^L$ except

$$\left.\begin{aligned} &c_{\{x,j\},\{y\}} = c_{\{y,j\},\{x\}} = c_{\{x\},\{y,j\}} = c_{\{y\},\{x,j\}} = 0.9, \\ &c_{\{x,y\},\{j\}} = c_{\{j\},\{x,y\}} = (g_{\{x,y,j\}} - 3.5)_+/2, \end{aligned}\right\} \quad j \in J,$$

where $f_+ = f$ if $f > 0$ and $f_+ = 0$ if $f \le 0$. This is a coloring of $\mathbf{g}$: for example,

$$\sum_{A_1 \cup A_2 = \{x,y,j\}} c_{A_1,A_2} > 4 \times 0.9 + 2 \times (g_{\{x,y,j\}} - 3.5)_+/2 > g_{\{x,y,j\}}.$$

Also, neither $\mathbf{c}_1$ nor $\mathbf{c}_2$ contains $\mathbf{k}_{2,1}$: for example,

$$\sum_{\substack{A \in 2^L \\ A \supset \{x,y\}}} c_{i,A} < (5 - 3.5)/2 + 0.1 < 1, \quad i = 1, 2,$$

as $d_{\mathbf{g}}(j) \ge 1$, $j \in J$. This contradiction proves that $\mathbf{g} \cong \mathbf{k}_{3,6}$.

Let G_n be a minimum $(K_{2,n}, K_{2,n})$-arrowing graph, and let $L_n = \{x \in V(G_n) : d(x) \ge n\}$. By Theorem 3.4 $|L_n| = 3$ for all large n. By the minimality of G_n, $V(G_n) \setminus L_n$ spans no edge and each $x \in V(G_n) \setminus L_n$ sends three edges to L_n.

If L spans one or two edges in G_n, then these edges can be removed without affecting the arrowing property. Thus $e(G_n[L_n])$ equals 0 or 3. Now the easy analysis completes the proof. □

Acknowledgments. The author is grateful to Martin Henk, Deryk Osthus, and Günter Ziegler for helpful discussions and to David Avis for help with programming (and for his useful `lrs` package).

REFERENCES

[1] D. Avis, `lrs` *Homepage*, http://cgm.cs.mcgill.ca/~avis/C/lrs.html.

[2] J. Beck, *On size Ramsey number of pathes, trees, and circuits,* I, J. Graph Theory, 7 (1983), pp. 115–129.

[3] J. Beck, *On size Ramsey number of stars, trees, and circuits,* II, in Mathematics of Ramsey Theory, J. Nešetril and V. Rödl, eds., Springer, Berlin, 1990, pp. 34–45.

[4] M. Berkelaar, `lp_solve` *Homepage*, ftp://ftp.ics.ele.tue.nl/pub/lp_solve/.

[5] S. A. Burr, P. Erdős, R. J. Faudree, C. C. Rousseau, and R. H. Schelp, *Ramsey-minimal graphs for multiple copies*, Nederl. Akad. Wetensch. Indag. Math., 40 (1978), pp. 187–195.

[6] P. Erdős, *Problems and results in graph theory*, in The Theory and Applications of Graphs, G. Chartrand, ed., John Wiley, New York, 1981, pp. 331–341.

[7] P. Erdős and R. J. Faudree, *Size Ramsey functions*, in Sets, Graphs and Numbers, North-Holland, Amsterdam, 1992, pp. 219–238.

[8] P. Erdős, R. J. Faudree, C. C. Rousseau, and R. H. Schelp, *The size Ramsey number*, Period. Math. Hungar., 9 (1978), pp. 145–161.

[9] P. Erdős and C. C. Rousseau, *The size Ramsey number of a complete bipartite graph*, Discrete Math., 113 (1993), pp. 259–262.

[10] R. J. Faudree, C. C. Rousseau, and J. Sheehan, *A class of size Ramsey problems involving stars*, in Graph Theory and Combinatorics, B. Bollobas, ed., Cambridge University Press, Cambridge, UK, 1984, pp. 273–281.

[11] P. E. Haxell and Y. Kohayakawa, *The size-Ramsey number of trees*, Israel J. Math., 89 (1995), pp. 261–274.

[12] X. Ke, *The size Ramsey number of trees with bounded degree*, Random Structures Algorithms, 4 (1993), pp. 85–97.

[13] R. Lortz and I. Mengersen, *Size Ramsey results for paths versus stars*, Australas. J. Combin., 18 (1998), pp. 3–12.

[14] O. Pikhurko, *Size Ramsey numbers of stars versus* 4*-chromatic graphs*, J. Graph Theory, to appear.

[15] O. Pikhurko, *Asymptotic Size Ramsey Results for Bipartite Graphs*, eprint arXiv:math.CO/0101197 (includes the C source code), 2001.

[16] O. Pikhurko, *Size Ramsey numbers of stars versus* 3*-chromatic graphs*, Combinatorica, 21 (2001), pp. 403–412.

SIAM J. Discrete Math.
Vol. 16, No. 1, pp. 114–126

ALGEBRAIC TECHNIQUES FOR CONSTRUCTING MINIMAL WEIGHT THRESHOLD FUNCTIONS*

VASKEN BOHOSSIAN† and JEHOSHUA BRUCK†

Abstract. A linear threshold element computes a function that is a sign of a weighted sum of the input variables. The best known lower bounds on the size of threshold circuits are for depth-2 circuits with small (polynomial-size) weights. However, in general, the weights are arbitrary integers and can be of exponential size in the number of input variables. Namely, obtaining progress in lower bounds for threshold circuits seems to be related to understanding the role of large weights. In the present literature, a distinction is made between the two extreme cases of linear threshold functions with polynomial-size weights, as opposed to those with exponential-size weights. Our main contributions are in devising two novel methods for constructing threshold functions with minimal weights and filling up the gap between polynomial and exponential weight growth by further refining the separation. Namely, we prove that the class of linear threshold functions with polynomial-size weights can be divided into subclasses according to the degree of the polynomial. In fact, we prove a more general result—that there exists a minimal weight linear threshold function for any arbitrary number of inputs and any weight size.

Key words. threshold functions, computational complexity, neural networks

AMS subject classifications. 03D15, 68Q15, 68Q17, 92B20

PII. S0895480197326048

1. Introduction.

The present paper focuses on the study of a single linear threshold gate with binary inputs and output as well as integer weights. Such a gate is mathematically described by a *linear threshold function*.

Definition 1.1 (linear threshold function). *A linear threshold function of n variables is a Boolean function $f : \{0,1\}^n \to \{0,1\}$ that can be written, for any $\mathbf{x} \in \{0,1\}^n$ and a fixed $\mathbf{w} \in Z^{n+1}$, as*

$$f(\mathbf{x}) = sgn(F(\mathbf{x})) = \begin{cases} 1 & \text{for } F(\mathbf{x}) \geq 0, \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{where } F(\mathbf{x}) = \mathbf{w} \cdot (-1, \mathbf{x}) = -w_0 + \sum_{i=1}^{n} w_i x_i.$$

Although we could allow the weights, w_i, to be real numbers, it is known [Muroga 71] that one needs only $O(n \log n)$ bits per weight, where n is the number of inputs. So in the rest of the paper, we will assume without loss of generality that all weights are integers. Also, notice that a linear threshold function can be implemented as

$$f : \{-1,1\}^n \to \{0,1\}.$$

We will address both the $\{0,1\}$ and the $\{-1,1\}$ representations.

Note that, given a function f, the weight vector $\mathbf{w}$ is not unique (see Example 1 below).

*Received by the editors August 13, 1997; accepted for publication (in revised form) July 22, 2002; published electronically January 3, 2003. Part of this work was presented in Proceedings of Neural Information Processing Systems 8, 1995, under the title *On Neural Networks with Minimal Weights*, by the authors of this paper.

http://www.siam.org/journals/sidma/16-1/32604.html

†California Institute of Technology, Mail Code 136-93, Pasadena, CA 91125 (vincent@paradise.caltech.edu, bruck@paradise.caltech.edu).

Definition 1.2 (weight space). *Given a linear threshold function f we define $\mathcal{W}$ as the set of all weights that satisfy Definition* 1.1*, that is,*

$$\mathcal{W} = \{\mathbf{w} \in Z^n : \quad \forall \mathbf{x} \in \{0,1\}^n,\ sgn(\mathbf{w} \cdot (-1, \mathbf{x})) = f(\mathbf{x})\}.$$

Here follows a measure of the size of the weights.

Definition 1.3 (minimal weight size). *We define the size of a weight vector as the sum of the absolute values of the weights. The minimal weight size of a linear threshold function is defined as*

$$S[f] = \min_{\mathbf{w} \in \mathcal{W}} \left(\sum_{i=0}^{n} |w_i| \right).$$

The particular vector that achieves the minimum is called a minimal weight vector.

Naturally, $S[f]$ is a function of n.

1.1. Motivation. Why do we care about the size of the weights in threshold circuits?

Threshold circuits have been shown to be surprisingly powerful. For example, integer division can be implemented by a polynomial-size threshold circuit of constant depth [Beame 84], [Siu 93]. It is also proved in [Allender 89] that any function in AC^0 can be computed by depth-3 majority circuits of quasi-polynomial size; in fact, it is true for all of ACC^0 [Yao 90]. For a general survey about the representation of Boolean functions by threshold functions, see [Saks 93].

Given the foregoing impressive upper bounds, it is not surprising that we face difficulties in obtaining lower bounds. In fact, the best general lower bound for threshold circuits is the result that the inner-product mod 2 (IP2) requires exponential size for depth 2 [Hajnal 93]. However, this lower bound assumes that the circuits involve small weights, and it is not known whether IP2 can be computed by a depth-2 polynomial size threshold circuit with arbitrary weights. Obtaining progress in lower bounds for threshold circuits therefore seems to be related to understanding the role of large weights.

Hence, it is natural to ask how limited the computational power of the circuit is if one limits oneself to threshold elements with only "small" growth in the size of the coefficients. It has been shown [Anthony 93], [Hampson 86], [Hastad 94], [Myhill 61], [Muroga 71], [Siu 91] that there exist linear threshold functions that can be implemented by a single threshold element with exponentially growing weights, $S[f] \sim 2^n$, but cannot be implemented by a threshold element with smaller polynomialy growing weights, $S[f] \sim n^d$, d constant. In light of that result, the above question was dealt with by defining a class within the set of linear threshold functions, the class of functions with "small" (i.e., polynomialy growing) weights [Siu 91]. Most of the recent research focused on the power of circuits with small weights, relative to circuits with arbitrary weights [Goldmann 92], [Goldmann 98]. In particular, it showed that increasing the depth of the circuit by one is sufficient to reduce all the weights to be of polynomial size. However, these impressive upper bounds were still not helpful in improving the lower bounds.

In this paper we take a different approach. Rather than dealing with circuits we focus on the modest task of studying a single threshold gate. The main contribution of the present paper is to further refine the division of small versus arbitrary weights. We separate the set of functions with small weights into classes indexed by d, the degree of polynomial growth, and show that all of them are nonempty. In particular, we develop a technique for proving that a weight vector is minimal. We use that technique to construct a function of size $S[f] = s$ for an arbitrary s. The natural future direction is to extend our techniques for constructing minimal weight threshold functions to circuits of depth 2. This might help in defining explicit functions that cannot be computed by depth-2, polynomial size threshold circuits with specific weight size.

1.2. Organization. Here follows a brief outline of the rest of the paper. In section 2 we show some of the difficulties one faces when minimizing the weights as well as how they are affected by the choice of input domain. In section 3 we consider functions defined over $\{-1, 1\}$. We limit ourselves to functions with no threshold (generalized majority function), and we show how to construct such functions with minimal weights. In section 4 we present another way of constructing minimal functions that allows us to deal with any threshold function defined over $\{0, 1\}$.

2. Preliminaries and examples.

In this section we illustrate some of the difficulties one faces when trying to minimize the weights of a threshold function. We also show how the input domain (i.e., $\{0, 1\}$ versus $\{-1, 1\}$) affects the size of the weights. See [Krause 95] for related results.

2.1. Minimizing the weights. The main difficulty in analyzing the size of the weights of a threshold element is due to the fact that a single linear threshold function can be implemented by different sets of weights as shown in the following example.

EXAMPLE 1 (a threshold function with minimal weights). *Let us consider the following two sets of weights (weight vectors):*

$$\mathbf{w}_1 = (4 \;\; 1 \;\; 2 \;\; 5), \; F_1(\mathbf{x}) = -4 + x_1 + 2x_2 + 5x_3,$$

$$\mathbf{w}_2 = (8 \;\; 2 \;\; 4 \;\; 10), \; F_2(\mathbf{x}) = -8 + 2x_1 + 4x_2 + 10x_3.$$

They both implement the same threshold function

$$f(\mathbf{x}) = sgn(F_2(\mathbf{x})) = sgn(2F_1(\mathbf{x})) = sgn(F_1(\mathbf{x})).$$

A closer look reveals that $f(\mathbf{x}) = sgn(-1 + x_3)$, implying that none of the above weight vectors has minimal size. Indeed, the minimal one is $\mathbf{w}_3 = (1 \;\; 0 \;\; 0 \;\; 1)$ and $S[f] = 2$.

To determine if a given set of weights is minimal is in general a difficult problem [Willis 63]. Our technique consists of constructing weight vectors whose minimality is easily established. We then show how to modify them, while keeping them minimal, in order to get to a larger set of functions.

2.2. $\{0, 1\}$ versus $\{-1, 1\}$. Suppose we implement the same function over $\{0, 1\}$ and over $\{-1, 1\}$. How are the weights affected? Let us look at an example.

EXAMPLE 2 (the *OR* function).

1. *Let $x_i \in \{0, 1\}$,*

$$OR(x_1, \ldots, x_n) = sgn(-1 + x_1 + \cdots + x_n).$$

The size of the weights is $s = n + 1$. Those weights are minimal.

Proof. The weights are integers. Reducing their size implies resetting one or more of them to 0, which will violate the definition of OR. □

2. *Now let $x_i \in \{-1, 1\}$,*

$$OR(x_1, \ldots, x_n) = sgn(n - 2 + x_1 + \cdots + x_n).$$

The size of the weights is $s = 2n - 2$. Those weights are minimal as well.

Proof. Any weights that implement OR have to be positive. Suppose there exist weights of size $s' < 2n - 2$. No weight can be 0, so $\sum_1^n w' \geq n$, implying that the threshold $-w_0 < (2n-2) - n = n-2$. Let w_i' be the smallest weight. Set $x_i = 1$ and all other inputs to -1. $\sum_1^n w' < -w_i(n-2)$ so that $F(\mathbf{x}) < 0$ violating the definition of OR. □

It appears from this example that the $\{0,1\}$ implementation has smaller weight size than the $\{-1,1\}$ representation. Is that true in general?

Example 3 (the majority (MAJ) function). *Let the number of variables, n, be odd. The majority function outputs true if more than half of its inputs are true.*

1. *Let $x_i \in \{0,1\}$,*

$$MAJ(x_1, \dots, x_n) = sgn\left(-\frac{n+1}{2} + x_1 + \cdots + x_n\right).$$

The size of the weights is $s = \frac{3n+1}{2}$. They can be shown to be minimal by a proof similar to case 2 in Example 2.

2. *Now let $x_i \in \{-1,1\}$,*

$$MAJ(x_1, \dots, x_n) = sgn(x_1 + \cdots + x_n).$$

Those weights are minimal, since reducing them would imply resetting one or more of them to 0, which will violate the definition of MAJ. The size of the weights is $s = n$.

Example 3 shows that in general we cannot tell which implementation $\{0,1\}$ or $\{-1,1\}$ will produce a function with smaller weights. However, the weight sizes for each of those functions are always within a constant factor of each other, since the $\{0,1\}$ weights are related to a set of $\{-1,1\}$ weights by a simple linear transformation.

3. Generalized majority function over $\{-1, 1\}$.

In this section we study the following model:

$$f : \{-1,1\} \to \{0,1\},$$

$$f(X) = sgn\left(\sum_1^n w_i x_i\right).$$

Notice that there is no threshold; we are looking at a majority function with arbitrary weights. We address the problem of constructing functions with minimal weights. In particular, our goal is that for a given number of inputs n and size s we find a function.

3.1. Mathematical setting.

We are interested in constructing functions for which the minimal weight is easily determined. Finding the minimal weight involves a search, and we are therefore interested in finding functions with constrained weight spaces. The following tools allow us to put constraints on $\mathcal{W}$.

Definition 3.1 (root space of a Boolean function). *A vector $\mathbf{v} \in \{-1,1\}^n$ such that $f(\mathbf{v}) = f(-\mathbf{v})$ is called a root of f. We define the root space, $\mathcal{R}$, as the set of all roots of f.* Note that a vector $\mathbf{v}$ is a root if and only if $\sum w_i v_i = 0$.

Definition 3.2 (root generator matrix). *For a given weight vector $\mathbf{w} \in \mathcal{W}$ and a root $\mathbf{v} \in \mathcal{R}$, the root generator matrix, $G = (g_{ij})$, is a $(k \times n)$-matrix, with entries in $\{-1,0,1\}$, whose rows $\mathbf{g}$ are orthogonal to $\mathbf{w}$ and equal to $\mathbf{v}$ at all nonzero coordinates, namely,*

1. $G\mathbf{w}^T = \mathbf{0}$*;*

2. *$g_{ij} = 0$ or $g_{ij} = v_j$ for all i and j.*

The root generator matrix is used to generate linearly independent root vectors for f. Each row of G corresponds to a new root vector.

Example 4 (root generator matrix). *Suppose that we are given a linear threshold function specified by a weight vector $\mathbf{w} = (1,1,2,4,1,1,2,4)$. By inspection we determine one root $\mathbf{v} = (1,1,1,1,-1,-1,-1,-1)$. Notice that $w_1 + w_2 - w_7 = 0$ which can be written as $\mathbf{g} \cdot \mathbf{w} = 0$, where $\mathbf{g} = (1,1,0,0,0,0,-1,0)$ is a row of G. Set $\mathbf{r} = \mathbf{v} - 2\mathbf{g}$. Since $\mathbf{g}$ is equal to $\mathbf{v}$ at all nonzero coordinates, $\mathbf{r} \in \{-1,1\}^n$. Also $\mathbf{r} \cdot \mathbf{w} = \mathbf{v} \cdot \mathbf{w} - 2\mathbf{g} \cdot \mathbf{w} = 0$. We have generated a new root: $\mathbf{r} = (-1,-1,1,1,-1,-1,1,-1)$.*

Lemma 3.3 (orthogonality of G and $\mathcal{W}$). *For a given weight vector $\mathbf{w} \in \mathcal{W}$ and a root $\mathbf{v} \in \mathcal{R}$, $G\mathbf{u}^T = \mathbf{0}$ holds for any weight vector $\mathbf{u} \in \mathcal{W}$.*

Proof. For an arbitrary $\mathbf{u} \in \mathcal{W}$ and an arbitrary row, $\mathbf{g}_i$, of G, let $\mathbf{v}' = \mathbf{v} - 2\mathbf{g}_i$. By definition of $\mathbf{g}_i$, $\mathbf{v}' \in \{-1,1\}^n$ and $\mathbf{v}' \cdot \mathbf{w} = 0$. This implies $f(\mathbf{v}') = f(-\mathbf{v}') : \mathbf{v}'$ is a root of f. For any weight vector $\mathbf{u} \in \mathcal{W}$, $sgn(\mathbf{u} \cdot \mathbf{v}') = sgn(-\mathbf{u} \cdot \mathbf{v}')$. Therefore $\mathbf{u} \cdot (\mathbf{v} - 2\mathbf{g}_i) = 0$ and finally, since $\mathbf{v} \cdot \mathbf{u} = 0$, we get $\mathbf{u} \cdot \mathbf{g}_i = 0$. □

Lemma 3.4 (minimality). *For a given weight vector $\mathbf{w} \in \mathcal{W}$ and a root $\mathbf{v} \in \mathcal{R}$ if $\operatorname{rank}(G) = n-1$ (i.e., G has $n-1$ independent rows) and $|w_i| = 1$ for some i, then $\mathbf{w}$ is the minimal weight vector.*

Proof. From Lemma 3.3 any weight vector $\mathbf{u}$ satisfies $G\mathbf{u}^T = \mathbf{0}$. $\operatorname{rank}(G) = n-1$ implies that $\dim(\mathcal{W}) = 1$; i.e., all possible weight vectors are integer multiples of each other. Since $|w_i| = 1$, all vectors are of the form $\mathbf{u} = k\mathbf{w}$ for $k \geq 1$. Therefore $\mathbf{w}$ has the smallest size. □

We complete Example 4 with an application of Lemma 3.4.

Example 5 (minimality). *Given the following weights $\mathbf{w}$ and a root $\mathbf{v}$,*

$$\mathbf{w} = (1,1,2,4,1,1,2,4), \; \mathbf{v} = (1,1,1,1,-1,-1,-1,-1),$$

we can construct G:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

It is easy to verify that $\operatorname{rank}(G) = n - 1 = 7$ and therefore, by Lemma 3.4, $\mathbf{w}$ is minimal and $S[f] = 16$.

3.2. Weight vectors.

In Example 5 we saw how, given a weight vector, one can show that it is minimal. In this section we present an example of a linear threshold function with minimal weight size, with an arbitrary number of input variables.

We would like to construct a weight vector and show that it is minimal. Let the number of inputs, n, be even. Let $\mathbf{w}$ consist of two identical blocks :

$$\mathbf{w} = (w_1, w_2, \ldots, w_{n/2}, w_1, w_2, \ldots, w_{n/2}).$$

Clearly, $\mathbf{v} = (1,1,\ldots,1,-1,-1,\ldots,-1)$ is a root and G is the corresponding generator matrix.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & -1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & -1 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & & & & & & & & & & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & -1 \end{pmatrix}$$

3.3. Construction. The following theorem states that given an integer s and a number of variables n there exists a function of n variables and minimal weight size s.

THEOREM 3.5 (main result). *For any pair (s, n), with both s and n even, satisfying $n \leq s \leq 2^{\frac{n}{2}}$, there exists a linear threshold function of n variables, f, with minimal weight size $S[f] = s$.*

Proof. Given a pair (s, n) that satisfies the above conditions we first construct a weight vector $\mathbf{w}$ that satisfies $\sum_{i=1}^{n} |w_i| = s$; then we show that it is the minimal weight vector of the function $f(x) = sgn(\mathbf{w} \cdot \mathbf{x})$. The proof is shown only for n even.

Construction.

1. Define $(a_1, a_2, \ldots, a_{n/2}) = (1, 1, \ldots, 1)$.

2. If $\sum_{i=1}^{n/2} a_i < s/2$, then increase by one the smallest a_i such that $a_i < 2^{i-2}$. (In the case of a tie take the a_i with smallest index i).

3. Repeat the previous step until $\sum_{i=1}^{n/2} a_i = s/2$ or $(a_1, a_2, \ldots, a_{n/2}) = (1, 1, 2, 4, \ldots, 2^{\frac{n}{2}-2})$.

4. Set $\mathbf{w} = (a_1, a_2, \ldots, a_{n/2}, a_1, a_2, \ldots, a_{n/2})$.

Because we increase the size by one unit at a time the algorithm will converge to the desired result for any integer s that satisfies $n \leq s \leq 2^{\frac{n}{2}}$. We have a construction for any valid (s, n) pair. Let us show that $\mathbf{w}$ is minimal.

Minimality. Given that $\mathbf{w} = (a_1, a_2, \ldots, a_{n/2}, a_1, a_2, \ldots, a_{n/2})$ we find a root $\mathbf{v}$,

$$\mathbf{v} = (1, 1, \ldots, 1, -1, -1, \ldots, -1),$$

and $n/2$ rows of the generator matrix G corresponding to the equations $w_i = w_{i+\frac{n}{2}}$. To form additional rows note that the first k a_i's are powers of two (where k depends on s and n). Those can be written as $a_i = \sum_{j=1}^{i-1} a_j$ and generate $k - 1$ rows. And finally note that all other a_i, $i > k$, are smaller than 2^{k+1}. Hence, they can be written as a binary expansion $a_i = \sum_{j=1}^{k} \alpha_{ij} a_j$, where $\alpha_{ij} \in \{0, 1\}$. There are $\frac{n}{2} - k$ such weights. G has a total of $n - 1$ independent rows. $\text{rank}(G) = n - 1$ and $w_1 = 1$; therefore, by Lemma 3.4, $\mathbf{w}$ is minimal and $S[f] = s$. □

EXAMPLE 6 (a function of 10 variables and size 26). *We start with* $\mathbf{a} = (1, 1, 1, 1, 1)$. *We iterate* $(1, 1, 2, 1, 1)$, $(1, 1, 2, 2, 1)$, $(1, 1, 2, 2, 2)$, $(1, 1, 2, 3, 2)$, $(1, 1, 2, 3, 3)$, $(1, 1, 2, 4, 3)$, $(1, 1, 2, 4, 4)$, *and finally the algorithm converges to* $\mathbf{a} = (1, 1, 2, 4, 5)$. *We claim that* $\mathbf{w} = (\mathbf{a}, \mathbf{a}) = (1, 1, 2, 4, 5, 1, 1, 2, 4, 5)$ *is minimal. Indeed,*

$\mathbf{v} = (1, 1, 1, 1, 1, -1, -1, -1, -1, -1)$ *and*

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

is a matrix of rank 9.

Example 7 (functions with polynomial size). *This example shows an application of Theorem* 3.5. *We define* $\widehat{LT}^{(d)}$ *as the set of linear threshold functions for which* $S[f] \leq n^d$. *The theorem states that for any even n there exists a function f of n variables and minimum weight* $S[f] = n^d$. *The implication is that for all d,* $\widehat{LT}^{(d-1)}$ *is a proper subset of* $\widehat{LT}^{(d)}$.

4. Arbitrary threshold function over $\{0, 1\}$.

In this section we present a different technique for constructing threshold functions with minimal weights. It allows us to construct functions with any weight size and number of variables. We consider functions with input domain $\{0, 1\}$, but, as mentioned below, the argument holds for an arbitrary input space $\{a, b\}$.

4.1. Approach.

The method we use is based on a result from [Willis 63]. We assume, without loss of generality, that the weights are strictly positive integers. Our goal is to minimize $s = \sum_0^n |w_i| = \sum_0^n w_i$. We know from [Muroga 71] that any other weights, $\mathbf{u}$, implementing the same function have to be strictly positive. We will show that under certain conditions on $\mathbf{w}$, $\sum_0^n w_i \leq \sum_0^n u_i$ for any $\mathbf{u}$.

Consider input vectors $\mathbf{x}$ and $\mathbf{y}$ for which the following equations hold:

$$F(\mathbf{x}) = -w_0 + \sum_1^n w_i x_i = 0, \qquad\qquad F(\mathbf{y}) = -w_0 + \sum_1^n w_i y_i = -1.$$

Let them define the rows of a matrix that we call A. Using p $\mathbf{x}$-type and q $\mathbf{y}$-type vectors we get

$$A = \begin{pmatrix} -1 & \mathbf{x}^{(1)} \\ -1 & \mathbf{x}^{(2)} \\ \vdots & \vdots \\ -1 & \mathbf{x}^{(p)} \\ 1 & -\mathbf{y}^{(1)} \\ 1 & -\mathbf{y}^{(2)} \\ \vdots & \vdots \\ 1 & -\mathbf{y}^{(q)} \end{pmatrix} = \begin{pmatrix} -1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ -1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_n^{(2)} \\ \vdots & & & & \vdots \\ -1 & x_1^{(p)} & x_2^{(p)} & \cdots & x_n^{(p)} \\ 1 & -y_1^{(1)} & -y_2^{(1)} & \cdots & -y_n^{(1)} \\ 1 & -y_1^{(2)} & -y_2^{(2)} & \cdots & -y_n^{(2)} \\ \vdots & & & & \vdots \\ 1 & -y_1^{(q)} & -y_2^{(q)} & \cdots & -y_n^{(q)} \end{pmatrix}$$

Example 8 (the matrix A). *Suppose we are given the following weights:*

$$\mathbf{w} = (16\ \ 1\ \ 2\ \ 4\ \ 8\ \ 1\ \ 2\ \ 4\ \ 8).$$

Our goal is to show they are minimal. We need to first construct the matrix A. Here follows a candidate:

$$A = \begin{pmatrix} -1 & \mathbf{x}^{(1)} \\ -1 & \mathbf{x}^{(2)} \\ 1 & -\mathbf{y}^{(1)} \\ 1 & -\mathbf{y}^{(2)} \end{pmatrix} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

There are many possible choices for A, depending on which of the **x**- *and* **y**-*type vectors are used. The one shown above is not a good one, as we will see. Additional* **x**-*type vectors need to be included in the construction of A in order to satisfy the requirements in Theorem* 4.1.

THEOREM 4.1 (condition for minimality). *Given a weight vector* **w**, *we construct A as described above. If there is a nonnegative row vector* **a** *(that is, $a_i \geq 0$ for all i) such that A satisfies*

$$\mathbf{a}A = (\overbrace{1\ \ldots\ 1}^{n+1}),$$

the weight vector **w** *is minimal.*

Proof. By definition of the **x**'s and the **y**'s the matrix A satisfies

$$A \cdot (w_0\ w_1\ w_2, \ldots, w_n)^T = (\overbrace{0\,0\ \ldots\ 0\,0}^{p}\ \overbrace{1\,1\ \ldots\ 1\,1}^{q})^T. \tag{4.1}$$

Because $sgn(0) = 1$ and $sgn(-1) = 0$, any other weight vector, **u**, implementing the same function has to verify the above equalities with "$\geq$" instead of "$=$":

$$A \cdot (u_0\ u_1\ u_2, \ldots, u_n)^T \geq (\overbrace{0\,0\ \ldots\ 0\,0}^{p}\ \overbrace{1\,1\ \ldots\ 1\,1}^{q})^T. \tag{4.2}$$

Let $\mathbf{v} = \mathbf{u} - \mathbf{w}$, and subtracting equations (4.1) from inequalities (4.2) we get

$$A \cdot (v_0\ v_1\ v_2, \ldots, v_n)^T \geq (\overbrace{0\,0 \ldots 0\,0}^{p+q})^T \tag{4.3}$$

Now suppose A is such that

$$(a_0\ a_1, \ldots, a_{p+q-1}) \cdot A = (\overbrace{1\,1 \ldots 1\,1}^{n+1}) \tag{4.4}$$

Where the a_i are strictly positive. We multiply inequalities (4.3) by **a** from the left and get

$$(a_0\ a_1, \ldots, a_{p+q-1}) \cdot A \cdot (v_0\ v_1\ v_2, \ldots, v_n)^T \geq (a_0\ a_1, \ldots, a_{p+q-1}) \cdot (\overbrace{0\,0 \ldots 0\,0}^{p+q})^T,$$

$$(\overbrace{1\,1\ \ldots\ 1\,1}^{n+1}) \cdot (v_0\ v_1\ v_2, \ldots, v_n)^T \geq 0,$$

$$\sum_0^n v_i \geq 0.$$

Since $w_i \geq 0$, $u_i \geq 0$ for all $i = 0, \dots, n$ we know that $\sum_0^n u_i \geq \sum_0^n w_i$. □

Notice that nowhere in the proof did we use the fact that the input domain is $\{0, 1\}$. Indeed, the above proof is valid for any input domain $\{a, b\}$. As you can see the proof relies on constructing A so that (4.4) holds. To construct A we need appropriate $\mathbf{x}$'s and $\mathbf{y}$'s which in turn depend on the choice $\mathbf{w}$.

4.2. Basic construction. In this section we introduce $\mathbf{w}$, the weight vector for the general construction, and prove it is minimal by finding an appropriate matrix A. We use a construction similar to the one in section 3, based on powers of two.

Construction. Given a pair (s, n), where $n+1 \leq s \leq 3 * 2^{\lfloor \frac{n}{2} \rfloor} - 2$, and $s = 3m - l$, with $l \in \{0, 1, 2\}$, we have the following:

1. Define $s' = 3m - 2$ and $n' = n - (s - s')$.
2. Define k as the largest integer such as $s' > 3 * 2^{k-1} - 2$.
3. Define $s_0 = \frac{1}{3}(s' - 3 * 2^{k-1} + 2)$.
4. Set

$$(w_0, w_1, \dots, w_{2k}) = (2^{k-1} + s_0, 1, 2, 4, \dots, 2^{k-2}, s_0, 1, 2, 4, \dots, 2^{k-2}, s_0).$$

At this point the size of $\mathbf{w}$ is s'. In the following two steps additional weights are added in order to get to the desired number of variables n and the exact weight size s.

5. For every w_i with $i \in \{2k+1, \dots, n'\}$ let $w_i = 1$ and subtract 1 from the largest weight w_j, $j \in \{1, \dots, 2k\}$. In case of a tie select the weight with largest index.
6. For every w_i with $i \in \{n'+1, \dots, n\}$ let $w_i = 1$. No subtraction is needed. (Notice that $n - n' \in \{0, 1, 2\}$.)

Let us look at two examples.

Example 9 (a function of 12 variables and size 35). *$s = 35 = 3 * 12 - 1$, therefore $s' = 34$, $n' = 11$, $k = 4$, $s_0 = 4$. The weight iterations are*

$$\mathbf{w} = (12, 1, 2, 4, 4, 1, 2, 4, 4),$$

$$\mathbf{w} = (12, 1, 2, 4, 4, 1, 2, 4, 3, 1),$$

$$\mathbf{w} = (12, 1, 2, 4, 4, 1, 2, 3, 3, 1, 1),$$

$$\mathbf{w} = (12, 1, 2, 4, 3, 1, 2, 3, 3, 1, 1, 1),$$

$$\mathbf{w} = (12, 1, 2, 4, 3, 1, 2, 3, 3, 1, 1, 1, 1).$$

Example 10 (base case: $n = 2k$, $s_0 = 2^{k-1}$). *Let us show that the weights of Example 8 are minimal. Using the above notation $n = 8$, $s_0 = 8$, and $k = 4$.*

$$\mathbf{w} = (16 \;\; 1 \;\; 2 \;\; 4 \;\; 8 \;\; 1 \;\; 2 \;\; 4 \;\; 8).$$

Here follow the X- and Y-type rows for A:

$$\left\{ \begin{array}{ccccccccc} -1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{array} \right\} \quad sumX_1 = (-2 \;\; 2 \;\; 1 \;\; 1 \;\; 1 \;\; 2 \;\; 1 \;\; 1 \;\; 1)$$

$$\left\{ \begin{array}{ccccccccc} -1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array} \right\} \quad sumX_2 = (-2 \;\; 0 \;\; 2 \;\; 1 \;\; 1 \;\; 0 \;\; 2 \;\; 1 \;\; 1)$$

$$\left\{ \begin{array}{ccccccccc} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ -1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{array} \right\} \quad sumX_3 = (-2 \;\; 0 \;\; 0 \;\; 2 \;\; 1 \;\; 0 \;\; 0 \;\; 2 \;\; 1)$$

$$\left\{ \begin{array}{ccccccccc} -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array} \right\} \quad sumX_4 = (-2 \;\; 0 \;\; 0 \;\; 0 \;\; 2 \;\; 0 \;\; 0 \;\; 0 \;\; 2)$$

$$\underbrace{\begin{matrix} 1 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 \\ 1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \end{matrix}}_{sumY_1=(2\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1)}$$

We replicate rows and add them in order to get to the all 1 *vector. Only the first five columns are shown.*

$$\begin{pmatrix} -2 & 2 & 1 & 1 & 1 \\ -2 & 0 & 2 & 1 & 1 \\ -2 & 0 & 0 & 2 & 1 \\ -2 & 0 & 0 & 0 & 2 \\ 2 & -1 & -1 & -1 & -1 \end{pmatrix} \quad \begin{pmatrix} -16 & 16 & 8 & 8 & 8 \\ -8 & 0 & 8 & 4 & 4 \\ -4 & 0 & 0 & 4 & 2 \\ -2 & 0 & 0 & 0 & 2 \\ 2 & -1 & -1 & -1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} -24 & 24 & 12 & 12 & 12 \\ -12 & 0 & 12 & 6 & 6 \\ -6 & 0 & 0 & 6 & 3 \\ -3 & 0 & 0 & 0 & 3 \\ 46 & -23 & -23 & -23 & -23 \end{pmatrix}$$

The last matrix was obtained by multiplying the first four rows by 3/2*, and the last row by* 23*. Its rows add up to the all* 1 *vector. Using the notation of Theorem* 4.1*, given the matrix A, as defined above,*

$$\mathbf{a} = \left(12, 12, 6, 6, 3, 3, \frac{3}{2}, \frac{3}{2}, 23, 23\right).$$

Theorem 4.2 (minimality of the construction). *For any pair (s, n) satisfying*

$$n + 1 \leq s \leq 3 * 2^{\lfloor \frac{n}{2} \rfloor} - 2$$

one can construct an n-variable threshold function with minimal weights of size s.

We will first show that steps 1–4 of the construction produce minimal weights. The second part of the proof focuses on adding a padding of unit weights in order to achieve the desired number of variables n.

Proof (part 1: no padding). As of step 4 of the construction,

$$\mathbf{w} = (2^{k-1} + s_0, 1, 2, 4, \ldots, 2^{k-2}, s_0, 1, 2, 4, \ldots, 2^{k-2}, s_0).$$

We are going to construct A, show that it satisfies $\mathbf{a}A = \mathbf{1}$, and apply Theorem 4.1. Only two Y-type vectors are needed for the construction of A:

$$\begin{pmatrix} 1 & 0 & \cdots & 0 & -1 & \cdots & -1 \\ 1 & -1 & \cdots & -1 & 0 & \cdots & 0 \end{pmatrix}$$

They add up to $(2\ \ -1\ \ \cdots\ \ -1)$. The X-type vectors, summed two by two, produce the following matrix (only the first $k+1$ columns are shown, the remaining k columns are identical to columns 2 to $k+1$):

$$A_X = \begin{pmatrix} -2 & 2 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 \\ -2 & 0 & 2 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 \\ -2 & 0 & 0 & 2 & 1 & 1 & \cdots & 1 & 1 & 1 \\ -2 & 0 & 0 & 0 & 2 & 1 & \cdots & 1 & 1 & 1 \\ \vdots & & & & & & & & & \vdots \\ -2 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 2 & 1 \\ -2 & t_0 & t_1 & t_2 & t_3 & t_4 & \cdots & t_{k-2} & t_{k-1} & 2 \end{pmatrix}$$

The t_i, ($t_i \in \{0,1\}$), are the binary expansion of $2^{k-1} - s_0$,

$$2^{k-1} - s_0 = \sum_{i=0}^{k-1} 2^i t_i.$$

One can verify that the last row is indeed the sum of two X-type vectors. Given the above choice of A we need to compute the a_i in the following set of equations:

$$\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{k-2} \\ a_{k-1} \\ a_k \end{pmatrix}^T \begin{pmatrix} -2 & 2 & 1 & 1 & 1 & \cdot\cdot & 1 & 1 & 1 & 1 \\ -2 & 0 & 2 & 1 & 1 & \cdot\cdot & 1 & 1 & 1 & 1 \\ -2 & 0 & 0 & 2 & 1 & \cdot\cdot & 1 & 1 & 1 & 0 \\ \vdots & & & & & & & & & \vdots \\ -2 & 0 & 0 & 0 & 0 & \cdot\cdot & 0 & 0 & 2 & 1 \\ -2 & t_0 & t_1 & t_2 & t_3 & \cdot\cdot & t_{k-4} & t_{k-3} & t_{k-2} & 2 \\ 2 & -1 & -1 & -1 & -1 & \cdot\cdot & -1 & -1 & -1 & -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 1 \end{pmatrix}^T$$

It is possible to get an explicit formula for a_i as a function of the t_i, but it is not necessary. All that is needed is to show that the a_i are nonnegative. Consider the following set of equations:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{k-2} \\ 1 \end{pmatrix}^T \begin{pmatrix} 2 & 1 & 1 & 1 & \cdot\cdot & 1 & 1 & 1 & 1 \\ 0 & 2 & 1 & 1 & \cdot\cdot & 1 & 1 & 1 & 1 \\ 0 & 0 & 2 & 1 & \cdot\cdot & 1 & 1 & 1 & 0 \\ \vdots & & & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdot\cdot & 0 & 0 & 2 & 1 \\ t_0 & t_1 & t_2 & t_3 & \cdot\cdot & t_{k-4} & t_{k-3} & t_{k-2} & 2 \end{pmatrix} = \begin{pmatrix} h \\ h \\ h \\ \vdots \\ h \\ h \end{pmatrix}^T$$

Notice that b_{k-1} is set to 1. This is a system of k equations with k unknowns. Solving for the b_i and h we get $2b_0 = h - t_0$, $2b_i = h - t_i - \sum_{j=0}^{i-1} b_j$, and $h = 2 + \sum_{j=0}^{k-2} b_j$. The last two equations can be combined into $b_{k-2} = 2 - t_{k-2}$. Using the recurrence formula, $2b_i = b_{i-1} - (t_i - t_{i-1})$, the remaining values are obtained:

$$b_{k-3} = 4 - t_{k-2} - t_{k-3},$$

$$b_{k-4} = 8 - 2t_{k-2} - t_{k-3} - t_{k-4},$$

$$b_{k-5} = 16 - 4t_{k-2} - 2t_{k-3} - t_{k-4} - t_{k-5},$$

$$\vdots$$

$$b_0 = 2^{k-1} - 2^{k-3}t_{k-2} - 2^{k-4}t_{k-3} - \cdots - t_1 - t_0,$$

$$h = 2^k - 2^{k-2}t_{k-2} - 2^{k-3}t_{k-3} - \cdots - 2t_1 - t_0.$$

Notice that all the b_i and h are nonnegative because $t_i \in \{0,1\}$.

Let $a_i = \alpha b_i$ for $i = 0, .., k-1$. We need to show that α and a_k are nonnegative. Going back to $\mathbf{a}A = \mathbf{1}$, the remaining two equations are

$$2a_k - 2\alpha \sum_{i=0}^{k-1} b_i = 1 \quad \text{and} \quad \alpha h - a_k = 1.$$

Solving for α and a_k we get $\alpha = 3/2(h - \sum b_i)$ and $a_k = (h + 2\sum b_i)/2(h - \sum b_i)$. Substituting for $h = 1 + \sum_{i=0}^{k-1} b_i$ we get

$$\alpha = \frac{3}{2} \quad \text{and} \quad a_k = \frac{1}{2} + \frac{3}{2}\sum_{i=0}^{k-1} b_i.$$

Since all b_i are nonnegative, $a_k \geq 0$, which completes the proof. □

Proof (part 2: extra padding of ones). The second part of the proof will focus on steps 5 and 6 of the construction. The following two lemmas are needed.

LEMMA 4.3 (splitting a weight). *Let* $\mathbf{w} = (w_0, w_1, \ldots, w_n)$ *be minimal. Then* $\tilde{\mathbf{w}} = (w_0, w_1, \ldots, w_{n-1}, a, b)$*, where* $a + b = w_n$ *is also minimal.*

Proof. Construct the matrix A while duplicating the last column. □

LEMMA 4.4 (adding an input with unit weight). *If* $\mathbf{w} = (w_0, w_1, \ldots, w_n)$ *is minimal, and* $w_0 > 0$*, then* $\tilde{\mathbf{w}} = (w_0, w_1, w_2, w_3, \ldots, w_{n+1})$*, where* $w_{n+1} = 1$*, is also minimal.*

Proof. Suppose it is not minimal, implying there exists a better choice for $\tilde{\mathbf{w}}$; let us call it $\mathbf{w}'$. There are two possibilities. Either $w'_{n+1} = 0$ or some of the w'_i for $i < n+1$ is smaller than the corresponding w_i. In the latter case, we set $x_{n+1} = 0$ and obtain the original function implemented with smaller weights, contradicting the hypothesis. Now suppose $w'_{n+1} = 0$, implying that $\tilde{f}$ does not depend on x_{n+1}. That in turn implies $\sum_0^n w_i x_i \geq 0$ or $\sum_0^n w_i x_i \leq -2$ for all inputs X. We can reduce w_0 by 1, implying the original function was not minimal. □

In step 5 of the construction, starting with the following weights,

$$\mathbf{w} = (2^{k-1} + s_0, 1, 2, 4, \ldots, 2^{k-2}, s_0, 1, 2, 4, \ldots, 2^{k-2}, s_0).$$

Lemma 4.3 is used to increase the number of weights while keeping their size constant. In step 6, a final adjustment is done for the cases $s = 3m - 1$ and $s = 3m$. Applying Lemma 4.4, an additional one, or two, unit weights are added to achieve the desired pair (s, n). The smallest weights achievable are $\mathbf{w} = (1 \ \ldots \ 1)$. Any smaller weights will produce a function of less variables. The upper bound $3 * 2^{\lfloor \frac{n}{2} \rfloor} - 2$ is achieved when $s_0 = 2^{k-1}$ and there is no padding of ones. □

EXAMPLE 11 (functions with polynomial size). *Just as in section* 3*, we can define* $\widehat{LT}^{(d)}$ *as the set of linear threshold functions for which* $S[f] \leq n^d$*. Theorem* 4.2 *states that for any* n *there exists a function* f *of* n *variables and minimum weight* $S[f] = n^d$*. The implication is that for all* d*,* $\widehat{LT}^{(d-1)}$ *is a proper subset of* $\widehat{LT}^{(d)}$.

5. Conclusions.

We presented two techniques for constructing minimal weight threshold functions of arbitrary weight size and number of inputs. We considered both the $\{0,1\}$ and $\{-1,1\}$ input domains. Using these techniques we further refined the separation between polynomialy and exponentially growing weights. The natural open problem is to find out if these new techniques are useful in extending the existing lower bounds [Hajnal 93] on circuit size to functions with arbitrary weights.

REFERENCES

[Allender 89] E. ALLENDER, *A note on the power of threshold circuits*, in Proceedings of the 30th IEEE Symposium on Foundations of Computer Science, Research Triangle Park, NC, 1989, pp. 580 – 584.

[Anthony 93] M. ANTHONY AND J. SHAWE-TAYLOR, *Using the perceptron algorithm to find consistent hypotheses*, Combin. Probab. Comput., 2 (1993), pp. 385–387.

[Beame 84] P.W. Beame, S.A. Cook, and H.J. Hoover, *Log depth circuits for division and related problems*, in Proceedings of the 25th IEEE Symposium on Foundations of Computer Science, Singer Island, FL, 1984, pp. 1–6.

[Goldmann 92] M. Goldmann, J. Hastad, and A. Razborov, *Majority gates vs. general weighted threshold gates*, Comput. Complexity, 2 (1992), pp. 277–300.

[Goldmann 98] M. Goldmann and M. Karpinski, *Simulating threshold circuits by majority circuits*, SIAM J. Comput., 27 (1998), pp. 230–246.

[Hajnal 93] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy, and G. Turan, *Threshold circuits of bounded depth*, J. Comput. System Sci., 46 (1993), pp. 129–154.

[Hampson 86] S.E. Hampson and D.J. Volper, *Linear function neurons: Structure and training*, Biol. Cybernet., 53 (1986), pp. 203–217.

[Hastad 94] J. Håstad, *On the size of weights for threshold gates*, SIAM. J. Discrete Math., 7 (1994), pp. 484–492.

[Krause 95] M. Krause and P. Pudlak, *On computing boolean functions by sparse real polynomials*, in Proceedings of the 36th IEEE Symposium on Foundations of Computer Science, Milwaukee, WI, 1995, pp. 682–691.

[Muroga 71] M. Muroga, *Threshold Logic and Its Applications*, Wiley-Interscience, New York, 1971.

[Myhill 61] J. Myhill and W. H. Kautz, *On the size of weights required for linear-input switching functions*, IRE Trans. Electronic Computers, EC10 (1961), pp. 288–290.

[Saks 93] M. Saks, *Slicing the hypercube*, in Surveys in Combinatorics, London Math. Soc. Lecture Note Ser. 187, 1, K. Walker, ed., Cambridge University Press, Cambridge, UK, 1993, pp. 211–256.

[Siu 93] K. Siu, J. Bruck, T. Kailath, and T. Hofmeister, *Depth efficient neural networks for division and related problems*, IEEE Trans. Inform. Theory, 39 (1993), pp. 423–435.

[Siu 91] K.-Y. Siu and J. Bruck, *On the power of threshold circuits with small weights*, SIAM J. Discrete Math., 4 (1991), pp. 423–435.

[Willis 63] D.G. Willis, *Minimum weights for threshold switches*, in Switching Theory in Space Techniques, Stanford University Press, Stanford, CA, 1963.

[Yao 90] A.C. Yao, *On ACC and threshold circuits*, in Proceedings of the 31th IEEE Symposium on Foundations of Computer Science, St. Louis, MO, 1990, pp. 619–627.

SIAM J. Discrete Math.
Vol. 16, No. 1, pp. 127–155

FACET OBTAINING PROCEDURES FOR SET PACKING PROBLEMS[*]

LÁZARO CÁNOVAS[†], MERCEDES LANDETE[‡], AND ALFREDO MARÍN[†]

Abstract. New results concerning the facial structure of set packing polyhedra are presented. In particular, new methods are given to obtain facets from the subgraphs of the intersection graph associated with a set packing polyhedron that properly subsume several other methods in the literature. A new class of facet defining graphs, termed fans, is also introduced, as well as a procedure to link any graph to a certain claw $K_{1,k}$ in order to obtain a new graph and an associated facet.

Key words. facet, set packing, independent set, stable set

AMS subject classifications. 05C75, 90C57

PII. S0895480100366786

1. Introduction.

Throughout this paper it is assumed that the graphs are finite, without loops, without multiple edges, undirected, and connected. Let $G = (V, E)$ be a graph with node set V and edge set E. G is said to be *odd* (resp., *even*) if $|V|$ is odd (resp., even). The *incidence vector* of a subset B of V is a binary vector $(x_1, \dots, x_{|V|})$ where $x_j = 1$ if and only if the jth node of V belongs to B, $j = 1, \dots, |V|$. A nonempty subset of V of mutually nonadjacent nodes in G is called a *packing* (*anticlique*, *stable set*, *independent set*). A *maximal* packing is a packing which is not a proper subset of another packing. A *maximum* packing is a packing of maximum cardinality. A *clique* in G is a maximal complete subgraph. A *hole* is a chordless cycle with more than three nodes. The *neighborhood* $N(v)$ of a node v is the set of nodes that are adjacent to v. The *incidence degree* $\delta(v)$ of a node v is the cardinality of its neighborhood. $P_I(G)$ is the set of incidence vectors of all packings of G, and the *polytope (polyhedron) associated with* G, $P(G)$ is the convex hull of $P_I(G)$. (It holds that $P(G)$ is a full dimensional polytope, and a vector x is a vertex of $P(G)$ if and only if $x \in P_I(G)$.) A *set packing problem* is a binary optimization problem

$$\text{SPP:} \quad \text{Opt}\,\{cx : \quad Ax \le \mathbf{1}_m, \quad x \in \{0,1\}^n\},$$

where $c \in \mathbf{R}^n$, $A \in \{0,1\}^{m \times n}$, and $\mathbf{1}_m$ is an m-vector of ones. The *graph associated with* (*intersection graph of)* SPP is $G = (V, E)$ with $|V| = n$ and $(v_i, v_j) \in E$ if and only if the ith and jth columns of A are not orthogonal. Then, if G is the graph associated with SPP, the feasible set of SPP is $P_I(G)$ and the optimal solutions of SPP can be obtained by solving the linear optimization problem

$$\text{Opt}\,\{cx : \quad x \in P(G)\}.$$

A linear inequality $\pi x \le \pi_0$ is said to be *valid* for $P(G)$ if it holds for all $x \in P(G)$ (if and only if it holds for all $x \in P_I(G)$). Given a valid inequality for $P(G)$, the set

[*] Received by the editors January 14, 2000; accepted for publication (in revised form) July 22, 2002; published electronically January 3, 2003. This work was partially supported by Plan Nacional de Investigación Científica, Desarrollo e Innovación Tecnológica (I+D+I), grant TIC2000-1750-C06-06.

http://www.siam.org/journals/sidma/16-1/36678.html

[†] Departamento de Estadística e Investigación Operativa, Facultad de Matemáticas, Campus de Espinardo, Universidad de Murcia, 30100 Murcia, Spain (lcanovas@um.es, amarin@um.es).

[‡] Centro de Investigación Operativa, Campus La Gal.lia, Universidad Miguel Hernández, 03202 Elche (Alicante), Spain (landete@umh.es).

$\{x \in P(G) : \pi x = \pi_0\}$ is called a *face* of $P(G)$. A *proper face* is a face different from the polyhedron and the empty set. A valid inequality for $P(G)$ is a *facet* of $P(G)$ if and only if it is satisfied as an equality (*exactly*) by $|V|$ linearly independent vertices of $P(G)$. Up to multiplication by a positive constant, there is a unique set of facets $\pi^i x \leq \pi_0^i$, $i = 1, \ldots, \ell$, such that $P(G) = \{x : \pi^i x \leq \pi_0^i,\ i = 1, \ldots, \ell\}$. A set of linear inequalities satisfying the last condition is called a *defining linear system* of $P(G)$. Since set packing problems have a large variety of practical applications, and linear optimization problems can be solved by means of several procedures, it is a matter of interest to contribute to the characterization of the defining linear system of $P(G)$, i.e., to obtain facets of $P(G)$. A facet of $P(G)$ is termed *nontrivial* if it is different from $x_j \geq 0$ for any $j = 1, \ldots, |V|$. All nontrivial facets of $P(G)$ are of the form $\pi x \leq \pi_0$ with $\pi_j \geq 0$, $j = 1, \ldots, |V|$, and $\pi_0 > 0$. A graph $G = (V, E)$ with $|V| = n$ is *facet defining* if there exists a facet $\pi x \leq \pi_0$ such that $\pi_j > 0$ for all $j \in V$.

Throughout the paper, sets of nodes are usually denoted by V or V_i, and the same node is denoted indifferently by v_j and j. In particular, j is used in the figures, summations, and subindices and v_j is used in the text. Frequently, the expression *facet of the graph* will be used instead of *facet of the polyhedron associated with the graph*, for brevity. Row and column vectors are not differentiated, since the orientation of the vectors employed in the paper should be clear from the context. X^i will denote the ith row of matrix X.

In the seminal papers [1, 6, 11, 12, 13, 14, 16] the basic principles to derive facets for set packing problems were given. One can find there the first families of facet defining graphs. It was proved in [11, 12, 14] that, if $G = (V, E)$ is a graph and B is a subset or V, the inequality $\sum_{j \in B} x_j \leq 1$ is a facet of $P(G)$ if and only if the subgraph induced by B is a clique in G. Consequently, a facet with right-hand side 1 and binary coefficients is called a *clique facet*. In [11, 12] it was shown that the inequality $\sum_{j \in V} x_j \leq (|V| - 1)/2$ is a facet of $P(G)$ if $G = (V, E)$ is an odd hole. Other results concerning *rank facets* $\sum_{j \in V} x_j \leq \pi_0$ were given in [1, 6, 14], and more recently in [8], [9] (where a complete characterization of the rank facets of $P(G)$ when G does not contain $K_{1,3}$ as an induced subgraph is given), [10], and [15]. In [11], Nemhauser and Trotter gave two procedures to construct facets of $P(G)$. Additional families of graphs and associated facets have been studied in the literature; see, for example, [2] and [4] for *wheels*, [2] and [3] for series-parallel and other special graphs.

In order to find new facets of the set packing polyhedron, it is useful to (i) identify families of graphs with associated known facets, and (ii) to devise methods for transforming known facets associated with smaller graphs into others containing those as subgraphs. Examples of these transformations in the literature can be consulted in [5, 14, 17]. We shall refer to these transformations as *liftings*, regardless of whether they keep the coefficients of the initial facet unchanged or not. Here the constructions in [14, 17] are revisited and extended, and new methods are developed.

2. Lifting procedures. The common techniques for proving that a given valid inequality induces a facet of a specified full dimensional polyhedron in $\mathbf{R}^n$ are (see, e.g., [7]) the following:

Proving necessity. The inequality must be included in any defining linear system of the polyhedron.

Direct construction. Display a set of n affinely (linearly, in our case) independent vectors in the polyhedron satisfying the inequality exactly.

Verifying maximality. Show that the face of the polyhedron defined by the inequality is not contained in any larger proper face.

Lifting procedures try to simplify the second method by using previously known information about facets of polyhedra of lower dimensions. In particular, when the polyhedron is $P(G')$, and G' can be obtained from a *smaller* graph G by means of some graph theoretic operations, a lifting method will transform the facet $\pi x \leq 1$ of $P(G)$ into the facet $\pi' x' \leq 1$ of $P(G')$ by constructing the independent vectors of $P(G')$ from those of $P(G)$. If the vectors of $P(G)$ are arranged as the rows of a matrix X, the facet of $P(G)$ is obtained by solving the regular equation system $X\pi = \mathbf{1}$. Then the lifting procedure can be seen as a way to obtain X' from X in such a way that the rows of X' are incidence vectors of $P(G')$ and the equation system $X'\pi' = \mathbf{1}$ can be solved to obtain π'.

Summarizing, five steps are necessary to implement the approach:

1. constructing X' from X,
2. proving the regularity of X',
3. solving the system $X'\pi' = \mathbf{1}$,
4. showing that $\pi'_j \geq 0$ for all j,
5. proving that $\pi' x' \leq 1$ is satisfied by all the packings in G'.

Sometimes it is necessary to impose algebraic conditions to guarantee that 2, 4, or 5 are satisfied.

Note that if the rows of X' include the incidence vectors of all the maximal packings of G, the nonnegativity of the solution π' of the system $X'\pi' = \mathbf{1}$ implies the validity of the inequality $\pi' x' \leq 1$. Moreover, if these rows include a nonmaximal packing p', the coefficients of the variables associated with the nodes in $p_i - p'$ will be zero for all the packings p_i such that $p' \subset p_i$.

The best known method to obtain facets of $P(G')$ from facets of the polyhedra associated with its subgraphs was simultaneously obtained by several authors ([12] for odd holes, [11, 13, 14] for the general case). This method will be called *usual lifting* throughout the paper. The usual lifting procedure fits as follows into our framework.

PROPOSITION 2.1 ([11, 12, 13, 14], usual lifting procedure). *Let $G' = (V', E')$ be a graph with $V' = \{v_1, \dots, v_n\}$. If the inequality $\sum_{j=1}^{n-1} \pi_j x_j \leq \pi_0$ is a facet of the subgraph of G' induced by $V' - \{v_n\}$, the inequality $\sum_{j=1}^{n-1} \pi_j x_j + \pi_n x_n \leq \pi_0$, where*

$$\pi_n = \pi_0 - \max \left\{ \sum_{j=1}^{n-1} \pi_j x_j : \ x \in P_I(G'), \ x_\ell = 0 \quad \forall v_\ell \in N(v_n) \right\}, \tag{2.1}$$

is a facet of $P(G')$.

Here

$$X' = \left(\begin{array}{ccc|c} & & & 0 \\ & X & & 0 \\ & & & 0 \\ \hline b_1 & \cdots & b_n & 1 \end{array} \right),$$

where X contains $n-1$ independent incidence vectors of packings of the subgraph induced by $V' - \{v_n\}$, and $(b_1, \dots, b_n)$ is the optimum of the maximization problem given in (2.1).

This paper analyzes more complicated combinatorial lifting situations, adding two or more variables at the same time, and involving less obvious extensions X' to the packings in X. Furthermore, the inverse way is also explored to obtain facets of a graph from facets associated with *greater* graphs.

Some technical results needed in the following sections are now given. All of them are quite straightforward, but to the best of our knowledge they do not appear explicitly in the literature.

PROPOSITION 2.2.

1. *Let $G = (V, E)$ be a graph. If $\pi_0 x_1 + \sum_{j=2}^{|V|} \pi_j x_j \leq \pi_0$ is a facet of $P(G)$, then v_1 is connected to every v_i such that $\pi_i > 0$.*

2. *Let $G = (V, E)$ be a graph, and let H be the graph $(V \cup \{v_0\}, E \cup \{(v_0, v_1), (v_0, v_2), \ldots, (v_0, v_{|V|})\})$. Then $\sum_{j=1}^{|V|} \pi_j x_j \leq \pi_0$ is a facet of $P(G)$ if and only if $\sum_{j=1}^{|V|} \pi_j x_j + \pi_0 x_0 \leq \pi_0$ is a facet of $P(H)$.*

Proof.

1. If not, a packing violating the inequality can be constructed.

2. The *only if* part directly follows applying the usual lifting to the first facet. To see the *if* part, note that $\{v_0\}$ is a maximal packing satisfying the second inequality exactly. The remaining $|V|$ independent packings do not contain v_0 and satisfy the first inequality (which is valid for $P(G)$) exactly. □

PROPOSITION 2.3. *Let $G = (V, E)$ be a graph and $\pi x \leq \pi_0$ a facet of $P(G)$ other than a clique facet. If a new node v_0 is added to G and connected to a set of nodes inducing a complete subgraph of G, then the coefficient of v_0 obtained by means of the usual lifting of $\pi x \leq \pi_0$ is null.*

Proof. The coefficient is null if and only if a vertex x of $P(G)$ verifying $\pi x = \pi_0$ and such that $x_j = 0$ for all $v_j \in N(v_0)$ exists. For if not, all the vertices x of $P(G)$ with $\pi x = \pi_0$ satisfy $\sum_{j \in N(v_0)} x_j \geq 1$. Since $N(v_0)$ induces a complete subgraph, it follows that $\sum_{j \in N(v_0)} x_j = 1$, and there cannot be enough independent points in $P(G)$ satisfying $\pi x = \pi_0$, unless both equalities are the same. □

PROPOSITION 2.4. *Let $G = (V, E)$ be a graph, let $C \subset V$ be a set of nodes inducing a clique in G, and let $v_1 \in C$ be a node such that $(v_1, v_j) \notin E$ for any $v_j \notin C$. Let $\pi x \leq \pi_0$ be a facet of $P(G)$ other than $\sum_{j \in C} x_j \leq 1$. Then $\pi_1 = 0$.*

Proof. Let P be any packing in G satisfying $\pi x = \pi_0$. If P does not include nodes of $C - \{v_1\}$ and $\pi_1 > 0$, P must include v_1. For, if not, $P \cup v_1$ is a packing which violates the inequality. In any case, if $\pi_1 > 0$, then $\sum_{j \in P} x_j = 1$, and there cannot exist enough independent packings satisfying $\pi x \leq \pi_0$ exactly. □

3. Replacing a node by $K_{1,p}$.

In this section a facet generating procedure given in [17] is generalized (in several ways). The generalization is worthwhile itself and is also employed in the forthcoming section. The arguments of the proofs have been adapted from those of [17].

3.1. Replacing a node by $K_{1,2}$.

Let us initially consider the following construction (see Figure 3.1). Given a graph $G = (V, E)$ and a selected node $v_n \in V$, a new graph G' is obtained by

(i) separating the nodes adjacent to v_n into two nonempty subsets V_1 and V_2,

(ii) introducing two new nodes v_{n+1} and v_{n+2} so that each vertex of V_i is joined to v_{n+i}, $i = 1, 2$, and

(iii) joining v_n to v_{n+1} and v_{n+2} only.

THEOREM 3.1. *Let $\sum_{j=1}^{n} \pi_j x_j \leq \pi_0$ be a facet of $P(G)$ and*

$$M_i := \max \left\{ \sum_{j=1}^{n} \pi_j x_j : x \in P_I(G), x_\ell = 0 \ \forall v_\ell \in V_i \cup \{v_n\} \right\}, \qquad i = 1, 2.$$

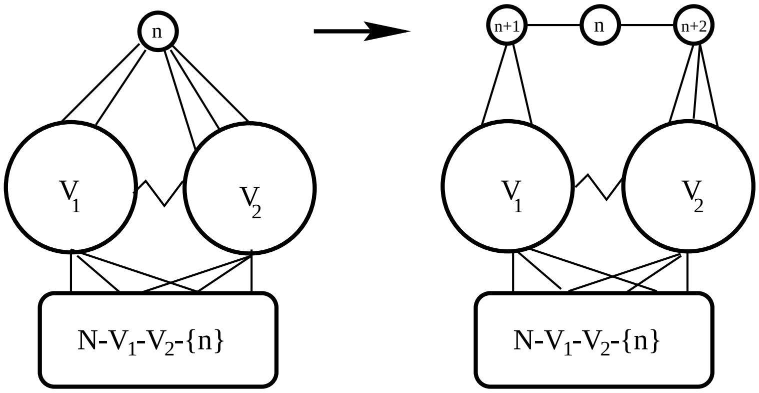


FIG. 3.1. *Construction of Theorem* 3.1.

If $M_1 + M_2 + \pi_n \geq 2\pi_0$, *then*

$$\begin{aligned} &\sum_{j=1}^{n-1} \pi_j x_j + (M_1 + M_2 + \pi_n - 2\pi_0)x_n \\ &+ (M_2 + \pi_n - \pi_0)x_{n+1} + (M_1 + \pi_n - \pi_0)x_{n+2} \leq M_1 + M_2 + \pi_n - \pi_0 \end{aligned} \tag{3.1}$$

is a facet of $P(G')$.

Proof. The proof is divided in five parts, according to the generic procedure given in section 1.

Part 1. *The matrix* X'. Let $\{(X^k, 0)\}_{k=1}^{s}$, $\{(X^k, 1)\}_{k=s+1}^{n}$ be n independent points of $P_I(G)$ satisfying $\sum_{j=1}^{n} \pi_j x_j = \pi_0$, and let $(X^{n+i}, 0)$, $i = 1, 2$, be two vertices of $P(G)$ satisfying $x_j^{n+i} = 0$, $j \in V_i \cup \{v_n\}$, and $\sum_{j=1}^{n} \pi_j x_j^{n+i} = M_i$. Let X' be the $(n+2) \times (n+2)$ matrix

$$\left(\begin{array}{c|ccc} X^1 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ X^s & 1 & 0 & 0 \\ \hline X^{s+1} & 0 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ X^n & 0 & 1 & 1 \\ \hline X^{n+1} & 0 & 1 & 0 \\ X^{n+2} & 0 & 0 & 1 \end{array}\right).$$

It can be rightly checked that the $n+2$ rows of X' are incidence vectors of vertices of $P(G')$.

Part 2. *Regularity of* X'. It is clear that the $n+1$ first rows of X' are independent, and assuming w.l.o.g. that $(X')^{n+2} = \sum_{k=1}^{n} \alpha_k (X')^k + \beta (X')^{n+1}$ it follows that

$\sum_{k=1}^{s}\alpha_k = 0$, $\sum_{k=s+1}^{n}\alpha_k = 1$, $\beta = -1$, $X^{n+2} = \sum_{k=1}^{n}\alpha_k X^k - X^{n+1}$, and

$$\begin{aligned} M_2 = \sum_{j=1}^{n-1}\pi_j x_j^{n+2} &= \sum_{j=1}^{n-1}\pi_j(\sum_{k=1}^{n}\alpha_k x_j^k - x_j^{n+1}) \\ &= \sum_{k=1}^{n}\alpha_k\sum_{j=1}^{n-1}\pi_j x_j^k - \sum_{j=1}^{n-1}\pi_j x_j^{n+1} \\ = \pi_0\sum_{k=1}^{s}\alpha_k + (\pi_0-\pi_n)&\sum_{k=s+1}^{n}\alpha_k - M_1 = \pi_0 - \pi_n - M_1, \end{aligned}$$

which is impossible by hypothesis.

Part 3. *Solution of* $X'\pi' = \mathbf{1}$. Since X' is regular, there is a unique solution of the system. It can be easily checked that the coefficients of (3.1), divided by its right-hand side, satisfy all the equations.

Part 4. *Nonnegativity of* π'. The proof follows from the assumptions of the theorem.

Part 5. *Validity of* (3.1). Let x^1 be a vertex of $P(G')$. If $x_n^1 = 1$, then $x_{n+1}^1 = x_{n+2}^1 = 0$ and the point x^2 given by $x_j^2 = x_j^1$ for $j = 1,\ldots,n-1$, $x_n^2 = 0$, is a vertex of $P(G)$; thus

$$\begin{aligned} &\sum_{j=1}^{n-1}\pi_j x_j^1 + (M_1+M_2+\pi_n-2\pi_0)x_n^1 \\ &= \sum_{j=1}^{n}\pi_j x_j^2 + M_1 + M_2 + \pi_n - 2\pi_0 \le M_1 + M_2 + \pi_n - \pi_0. \end{aligned}$$

If $x_n^1 = 0$ and $x_{n+1}^1 = x_{n+2}^1 = 1$, the point x^3 given by $x_j^3 = x_j^1$ for $j = 1,\ldots,n-1$, $x_n^3 = 1$, is a vertex of $P(G)$ and

$$\begin{aligned} &\sum_{j=1}^{n-1}\pi_j x_j^1 + (M_2+\pi_n-\pi_0)x_{n+1}^1 + (M_1+\pi_n-\pi_0)x_{n+2}^1 \\ &= \sum_{j=1}^{n}\pi_j x_j^3 + M_1 + M_2 + \pi_n - 2\pi_0 \le M_1 + M_2 + \pi_n - \pi_0. \end{aligned}$$

If $x_n^1 = x_{n+1}^1 = 0$ and $x_{n+2}^1 = 1$, using the above defined point x^2

$$\sum_{j=1}^{n-1}\pi_j x_j^1 + (M_1+\pi_n-\pi_0)x_{n+2}^1 = \sum_{j=1}^{n-1}\pi_j x_j^2 + M_1 + \pi_n - \pi_0 \le M_1 + M_2 + \pi_n - \pi_0,$$

and the remaining cases are treated in a similar way. □

Example. Let G be the left-hand graph of Figure 3.2, $v_n = v_{21}$, $V_1 = \{v_{16}, v_{19}, v_{20}\}$, and $V_2 = \{v_{17}, v_{18}\}$. Then G' is the right-hand graph of Figure 3.2 and the facet

$$\pi x = 2\sum_{i=1}^{5}x_i + \sum_{i=6}^{20}x_i + 2x_{21} \le 12$$

of $P(G)$ (see [4]) becomes the facet of $P(G')$,

$$2\sum_{i=1}^{5}x_i + \sum_{i=6}^{21}x_i + 2x_{22} + x_{23} \le 13,$$

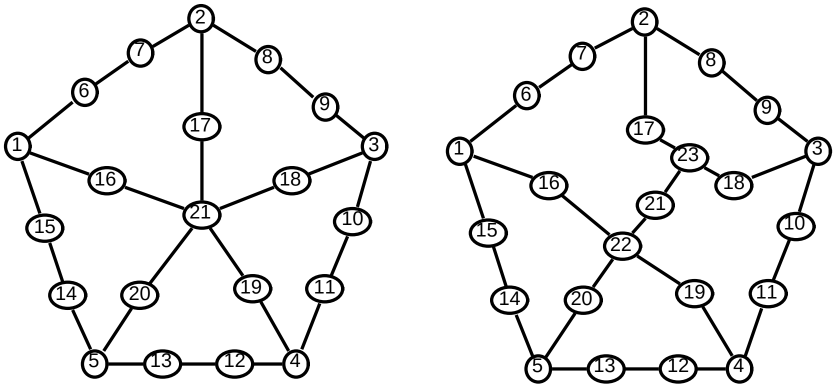


Fig. 3.2. *Example of Theorem* 3.1.

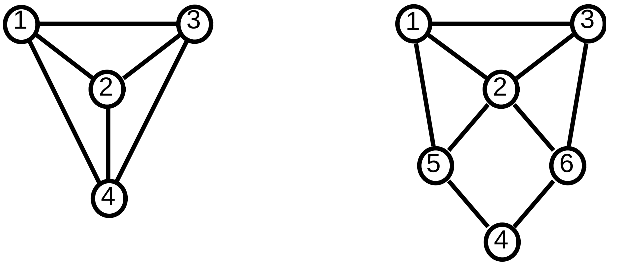


Fig. 3.3. *Example of Theorem* 3.6 *when* $V_1 \cap V_2 \neq \emptyset$.

since $M_1 = \max\{\pi x : x_{16} = x_{19} = x_{20} = x_{21} = 0, x \in P_I(G)\} = 11$ and $M_2 = \max\{\pi x : x_{17} = x_{18} = x_{21} = 0, x \in P_I(G)\} = 12$, $\pi_0 = 12$, and $\pi_{21} = 2$.

Remark 1. Proposition 3 in [17] is obtained as a particular case of Theorem 3.1 by taking $M_1 = M_2 = \pi_0$. As seen in the previous example, this is an unnecessarily restrictive condition.

Remark 2. The proof of Theorem 3.1 remains valid when $V_1 \cap V_2 \neq \emptyset$. A similar observation can be made about the assumptions of the forthcoming Theorem 3.6.

Example. Let G be the left-hand graph of Figure 3.3, $v_n = v_4$, $V_1 = \{v_1, v_2\}$, and $V_2 = \{v_2, v_3\}$. Then G' is the right-hand graph of Figure 3.3 and the facet $\pi x = \sum_{i=1}^{4} x_i \leq 1$ of $P(G)$ becomes the facet of $P(G')$ $\sum_{i=1}^{6} x_i \leq 2$, since $M_1 = \max\{\pi x : \ x_1 = x_2 = x_4 = 0, \ x \in P_I(G)\} = 1$ and $M_2 = \max\{\pi x : \ x_2 = x_3 = x_4 = 0, \ x \in P_I(G)\} = 1$, $\pi_0 = 1$, and $\pi_4 = 1$.

We will show next how Proposition 2 in [17] (see also [14]) can be obtained from Theorem 3.1. Consider the nodes adjacent to v_n subdivided into two subsets, V_1 containing a unique node (say v_1) and V_2 containing the remaining nodes (see Figure 3.4).

Corollary 3.2. *Let* $\sum_{j=1}^{n} \pi_j x_j \leq \pi_0$ *be a facet of* $P(G)$ *other than* $x_1 + x_n \leq 1$, *and let* $G^- := (N, E - \{(v_1, v_n)\})$ *be the graph obtained by removing the edge joining* v_1 *to* v_n *from* G. *Assume an optimal solution of the auxiliary problem*

$$\max\left\{\sum_{j=1}^{n} \pi_j x_j : x \in P_I(G^-)\right\}$$

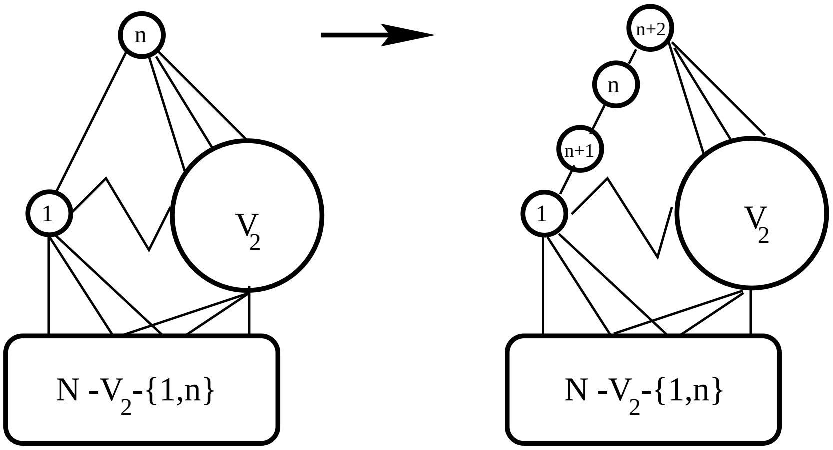


FIG. 3.4. *Inserting two nodes inside an edge.*

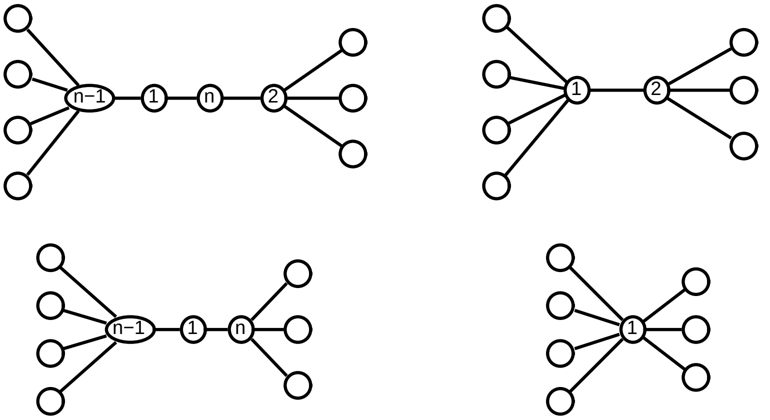


FIG. 3.5. *Transformations of Theorems* 3.3 *and* 3.4.

exists such that $x_1 = x_n = 1$, and let Z be its optimal value. Then

$$\sum_{j=1}^{n-1} \pi_j x_j + (Z - \pi_0)x_n + (Z - \pi_0)x_{n+1} + \pi_n x_{n+2} \leq Z \tag{3.2}$$

is a facet of $P(G')$.

Proof. Since the original facet is other than $x_1 + x_n \leq 1$, a vertex in $P(G)$ with $x_1 = x_n = 0$ satisfying $\sum_{j=1}^{n} \pi_j x_j = \pi_0$ will exist. Therefore $M_1 = \pi_0$.

Now, since the optimal value of the auxiliary problem is reached in some point with $x_1 = x_n = 1$, the maximum M_2 will verify $M_2 = Z - \pi_n \geq \pi_0 - \pi_n$, which is the condition of Theorem 3.1, and (3.2) follows. □

A converse transformation was shown in Theorem 2.5 of [2].

THEOREM 3.3 ([2], converse of Corollary 3.2). *Let $G = (V, E)$ be a graph with $|V| = n$, and let $\sum_{j=1}^{n} \pi_j x_j \leq \pi_0$ be a facet defined by $P(G)$. Suppose that (v_{n-1}, v_1), (v_1, v_n), $(v_n, v_2) \in E$, and v_1, v_n have degree two. Assume also that $\pi_{n-1} = \pi_1 = \pi_n = \beta$. Let G^b be the graph obtained from G by deleting v_{n-1} and v_n, and linking v_1 to v_2 and to the nodes in $N(v_{n-1})$ (see Figure* 3.5*). Then $\sum_{j=1}^{n-2} \pi_j x_j \leq \pi_0 - \beta$ is a* *facet of* $P(G^b)$.

The following result deals with the case when v_n does not have degree two. Although the proof is similar to that of [2], it is included here for the sake of completeness.

THEOREM 3.4. *Let* $G = (V, E)$ *be a graph with* $|V| = n$, *and let* $\pi x = \sum_{j=1}^{n} \pi_j x_j \leq \pi_0$ *be a facet of* $P(G)$. *Suppose that* (v_{n-1}, v_1), $(v_1, v_n) \in E$, $(v_{n-1}, v_n) \notin E$, *and* v_1 *has degree two. Assume also that* $\pi_{n-1} = \pi_1 = \pi_n = \beta > 0$. *Let* G^m *be the graph obtained from* G *by deleting* v_{n-1} *and* v_n, *and linking* v_1 *to the nodes in* $N(v_{n-1}) \cup N(v_n)$ *(see Figure* 3.5*). Then*

$$\sum_{j=1}^{n-2} \pi_j x_j \leq \pi_0 - \beta \tag{3.3}$$

is a facet of $P(G^m)$.

Proof.

Part 1. *The matrix* X'. Let $X = (x_{ij})$ be the $n \times n$ matrix whose rows are the incidence vectors of the independent packings associated with $\pi x \leq \pi_0$. Then X' is obtained from the $n \times (n-2)$ matrix

$$X'' = \begin{pmatrix} x_{11} + x_{1,n-1} + x_{1n} - 1 & x_{12} & \cdots & x_{1,n-2} \\ x_{21} + x_{2,n-1} + x_{2n} - 1 & x_{22} & \cdots & x_{2,n-2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} + x_{n,n-1} + x_{nn} - 1 & x_{n2} & \cdots & x_{n,n-2} \end{pmatrix}$$

by choosing $n-2$ rows of maximum rank. Note that all the rows of X'' are incidence vectors of vertices of $P_I(G^m)$.

Part 2. *Regularity of* X'. Consider the $(n+1) \times (n+1)$ linear equations system

$$X'''\lambda = \left(\begin{array}{ccccc|c} & & & & & 1 \\ & & X & & & \vdots \\ & & & & & 1 \\ \hline 1 & 0 & 0 & \cdots & 0 & 1 \end{array} \right) \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ \lambda_{n+1} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}.$$

It follows that $\lambda_{n+1} = -\lambda_1$ and

$$\begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix} = -\lambda_{n+1} X^{-1} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = -(\lambda_{n+1}/\pi_0)\pi$$

and then $\lambda_1(\pi_0 - \pi_1) = 0$. Since v_1 is not connected to the remaining nodes in V (Proposition 2.2), it follows that $\lambda_1 = 0$ and then $\lambda_j = 0$ for all j. Therefore X''' has range $n+1$. X'' is obtained from X''' by (i) adding $(X''')^{n-1} + (X''')^n - (X''')^{n+1}$ to $(X''')^1$ and (ii) deleting $(X''')^{n-1}$, $(X''')^n$, and $(X''')^{n+1}$; therefore the range of X'' must be $n-2$, there exist $n-2$ independent rows in X'', and the range of X' is $n-2$.

Part 3. *Solution of* $X'\pi' = \mathbf{1}$. The coefficients of (3.3), divided by the right-hand side, are the unique solution of the system $X''\pi' = \mathbf{1}$ and, in particular, of $X'\pi' = \mathbf{1}$.

Part 4. *Nonnegativity of* π'. The proof is obvious.

Part 5. *Validity of* (3.3). Consider $(x_1, \ldots, x_{n-2}) \in P_I(G^m)$. The point $(1 - x_1, x_2, \ldots, x_{n-2}, x_1, x_1)$ belongs to $P_I(G)$ and then $\beta(1-x_1) + \sum_{j=1}^{n-2} \pi_j x_j + 2\beta x_1 \leq \pi_0$, which implies $\sum_{j=1}^{n-2} \pi_j x_j + \beta x_1 \leq \pi_0 - \beta$. □

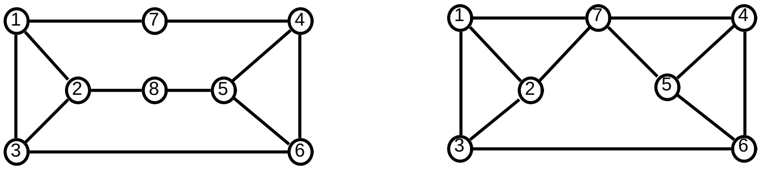


FIG. 3.6. *Example of Corollary* 3.5.

COROLLARY 3.5. *Let $G = (V, E)$ be a graph with $|V| = n$ and $(v_{n-1}, v_n) \notin E$, and let $\sum_{j=1}^{n} \pi_j x_j \le \pi_0$ be a facet of $P(G)$ satisfying $\pi_{n-1} = \pi_n = \pi_0 - \max\{\sum_{j=1}^{n-2} \pi_j x_j : x \in P_I(G)\}$. Then $\sum_{j=1}^{n-1} \pi_j x_j \le \pi_0 - \pi_n$ is a facet of the graph obtained from G by deleting v_n and linking v_{n-1} to the nodes in $N(v_n)$.*

Proof. It can be easily checked that this is equivalent to (i) adding a node to V linked to v_{n-1} and v_n only, (ii) making the usual lifting of the original facet, and (iii) applying Theorem 3.4. □

Example. Let G be the left-hand graph of Figure 3.6, $v_{n-1} = v_7$, and $v_n = v_8$. Then $\pi x = \sum_{i=1}^{8} x_i \le 3$ is a facet of $P(G)$. Applying Corollary 3.5, the facet $\sum_{i=1}^{7} x_i \le 2$ of the right-hand graph of Figure 3.6 is obtained.

3.2. Replacing a node by $K_{1,3}$. Theorem 3.1 can be extended by separating the nodes adjacent to n into more than two subsets. This is shown now by means of two results: Theorem 3.6, in which the separation is done into three subsets, and Theorem 3.11 in subsection 3.3, where separation occurs into m unitary subsets. To start with, the following construction is needed. Given a graph $G = (V, E)$ and a selected node $v_n \in E$, a new graph G'' is obtained by

(i) separating the nodes adjacent to v_n into three nonempty subsets V_i, $i = 1, 2, 3$,

(ii) introducing three new nodes v_{n+i} so that each vertex of V_i is joined to v_{n+i}, $i = 1, 2, 3$, and

(iii) joining v_n to v_{n+i}, $i = 1, 2, 3$, only.

THEOREM 3.6. *Let $\sum_{j=1}^{n} \pi_j x_j \le \pi_0$ be a facet of $P(G)$ and*

$$M_i := \max\left\{\sum_{j=1}^{n} \pi_j x_j : x \in P_I(G), x_\ell = 0 \ \forall v_\ell \in V_i \cup \{v_n\}\right\},$$

$$M_{ij} := \max\left\{\sum_{j=1}^{n} \pi_j x_j : x \in P_I(G), x_\ell = 0 \ \forall v_\ell \in V_i \cup V_j \cup \{v_n\}\right\}$$

for $i, j = 1, 2, 3$. (Note that $M_{ij} = M_{ji}$ for all i and j.) Then we have the following:

1. *If $M_{12} + M_{13} + M_{23} + 2\pi_n \ge 3\pi_0$ and $M_{i,i+1} + M_{i,i+2} + \pi_n \ge M_i + \pi_0$ for $i = 1, 2, 3$, then*

$$\sum_{j=1}^{n-1} \pi_j x_j + (M_{12} + M_{23} + M_{13} + 2\pi_n - 3\pi_0)x_n$$

$$+ \sum_{i=1}^{3} (M_{i+1,i+2} + \pi_n - \pi_0)x_{n+i} \le M_{12} + M_{23} + M_{13} + 2\pi_n - 2\pi_0 \tag{3.4}$$

is a facet of $P(G'')$.

2. *If* $M_1 + M_2 + M_3 + \pi_n \geq 3\pi_0$ *and* $M_i + M_{i+1} + \pi_0 \geq 2M_{i,i+1} + M_{i+2} + \pi_n$ *for* $i = 1, 2, 3$, *then*

$$\sum_{j=1}^{n-1} 2\pi_j x_j + (M_1 + M_2 + M_3 + \pi_n - 3\pi_0)x_n$$

$$+ \sum_{i=1}^{3}(M_{i+1} + M_{i+2} - M_i + \pi_n - \pi_0)x_{n+i} \leq M_1 + M_2 + M_3 + \pi_n - \pi_0$$

is a facet of $P(G'')$.

In parts 1 and 2, the index set $\{1, 2, 3\}$ *is considered to be cyclic; i.e.,* $4 = 1$, $5 = 2$, *and so on.*

Proof. Only the first part of the theorem is going to be proved. The proof of the second part is very similar and is left to the reader.

Part 1. *The matrix* X'. Let $\{(X^k, 0)\}_{k=1}^{s}$, $\{(X^k, 1)\}_{k=s+1}^{n}$ be n independent points of $P_I(G)$ satisfying $\sum_{j=1}^{n} \pi_j x_j = \pi_0$, and let $(X^{n+i}, 0)$, $i = 1, 2, 3$, be three vertices of $P(G)$ satisfying $x_j^{n+i} = 0$, $j \in V_{i+1} \cup V_{i+2} \cup \{v_n\}$, and $\sum_{j=1}^{n} \pi_j x_j^{n+i} = M_{i+1,i+2}$. Let X' be the $(n + 3) \times (n + 3)$ matrix

$$\left(\begin{array}{c|cccc}
X^1 & 1 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
X^s & 1 & 0 & 0 & 0 \\
\hline
X^{s+1} & 0 & 1 & 1 & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
X^n & 0 & 1 & 1 & 1 \\
\hline
X^{n+1} & 0 & 0 & 1 & 1 \\
X^{n+2} & 0 & 1 & 0 & 1 \\
X^{n+3} & 0 & 1 & 1 & 0
\end{array}\right).$$

It can be rightly checked that the $n + 3$ rows of X' are incidence vectors of vertices of $P(G'')$.

Part 2. *Regularity of* X'. It is clear that the $n+2$ first rows of X' are independent, and assuming w.l.o.g. that $(X')^{n+3} = \sum_{k=1}^{n} \alpha_k (X')^k + \beta_1 (X')^{n+1} + \beta_2 (X')^{n+2}$ it follows that $\sum_{k=1}^{s} \alpha_k = 0$, $\sum_{k=s+1}^{n} \alpha_k = 2$, $\beta_1 = \beta_2 = -1$, $X^{n+3} = \sum_{k=1}^{n} \alpha_k X^k - X^{n+1} - X^{n+2}$, and

$$\begin{aligned}
M_{12} = \sum_{j=1}^{n-1} \pi_j x_j^{n+3} &= \sum_{j=1}^{n-1} \pi_j (\sum_{k=1}^{n} \alpha_k x_j^k - x_j^{n+1} - x_j^{n+2}) \\
&= \sum_{k=1}^{n} \alpha_k \sum_{j=1}^{n-1} \pi_j x_j^k - \sum_{j=1}^{n-1} \pi_j x_j^{n+1} - \sum_{j=1}^{n-1} \pi_j x_j^{n+2} \\
= \pi_0 \sum_{k=1}^{s} \alpha_k + (\pi_0 - \pi_n) &\sum_{k=s+1}^{n} \alpha_k - M_{23} - M_{13} = 2\pi_0 - 2\pi_n - M_{23} - M_{13},
\end{aligned}$$

which is impossible by hypothesis.

Part 3. *Solution of* $X'\pi' = \mathbf{1}$. The coefficients of (3.4), divided by its right-hand side, satisfy all the equations.

Part 4. *Nonnegativity of* π'. Since $M_{ij} \geq \max\{\sum_{j=1}^{n} \pi_j x_j : x \in P_I(G),\ x_\ell = 0$ for all $v_\ell \in V_1 \cup V_2 \cup V_3 \cup \{v_n\}\} = \pi_0 - \pi_n$, the coefficients of (3.4) are nonnegative.

Part 5. *Validity of* (3.4). Let x^1 be a vertex of $P(G'')$. If $x^1_n = 1$, then $x^1_{n+i} = 0$, $i = 1, 2, 3$, and the point x^2 given by $x^2_j = x^1_j$ for $j = 1, \ldots, n-1$, $x^2_n = 0$, is a vertex of $P(G)$; thus

$$\begin{aligned} &\sum_{j=1}^{n-1} \pi_j x^1_j + (M_{12} + M_{23} + M_{13} + 2\pi_n - 3\pi_0)x^1_n \\ &= \sum_{j=1}^{n} \pi_j x^2_j + M_{12} + M_{23} + M_{13} + 2\pi_n - 3\pi_0 \leq M_{12} + M_{23} + M_{13} + 2\pi_n - 2\pi_0. \end{aligned}$$

If $x^1_n = 0$ and $x^1_{n+i} = 1$, $i = 1, 2, 3$, the point x^3 given by $x^3_j = x^1_j$ for $j = 1, \ldots, n-1$, $x^3_n = 1$, is a vertex of $P(G)$ and

$$\begin{aligned} &\sum_{j=1}^{n-1} \pi_j x^1_j + \sum_{i=1}^{3} (M_{i+1,i+2} + \pi_n - \pi_0)x^1_{n+i} \\ &= \sum_{j=1}^{n} \pi_j x^3_j + M_{12} + M_{23} + M_{13} + 2\pi_n - 3\pi_0 \leq M_{12} + M_{23} + M_{13} + 2\pi_n - 2\pi_0. \end{aligned}$$

If $x^1_n = x^1_{n+1} = 0$ and $x^1_{n+2} = x^1_{n+3} = 1$, using the above defined point x^2

$$\begin{aligned} &\sum_{j=1}^{n-1} \pi_j x^1_j + \sum_{i=2}^{3} (M_{i+1,i+2} + \pi_n - \pi_0)x^1_{n+i} \\ &= \sum_{j=1}^{n-1} \pi_j x^2_j + M_{13} + M_{12} + 2\pi_n - 2\pi_0 \leq M_{23} + M_{13} + M_{12} + 2\pi_n - 2\pi_0. \end{aligned}$$

If $x^1_n = x^1_{n+1} = x^1_{n+2} = 0$ and $x^1_{n+3} = 1$, using again the point x^2

$$\begin{aligned} &\sum_{j=1}^{n-1} \pi_j x^1_j + (M_{12} + \pi_n - \pi_0)x^1_{n+3} \\ &= \sum_{j=1}^{n-1} \pi_j x^2_j + M_{12} + \pi_n - \pi_0 \leq M_3 + M_{12} + \pi_n - \pi_0 \\ &\leq M_{13} + M_{23} + \pi_n - \pi_0 + M_{12} + \pi_n - \pi_0, \end{aligned}$$

the last inequality being by hypothesis. The remaining cases are treated in a similar way. □

Example.

1. Let G be the northwest graph of Figure 3.7, $v_n = v_5$, $V_1 = \{v_1\}$, $V_2 = \{v_2, v_3\}$, and $V_3 = \{v_4\}$. Then G'' is the northeast graph and $\pi x = \sum_{i=1}^{5} x_i \leq 1$ is a facet of $P(G)$, $M_i = M_{ij} = 1$ for all i, j, $\pi_0 = 1$ and $\pi_5 = 1$. Applying the first part of Theorem 3.6, the facet of $P(G'')$

$$x_1 + x_2 + x_3 + x_4 + 2x_5 + x_6 + x_7 + x_8 \leq 3$$

is obtained. The conditions of the second part of Theorem 3.6 are not satisfied.

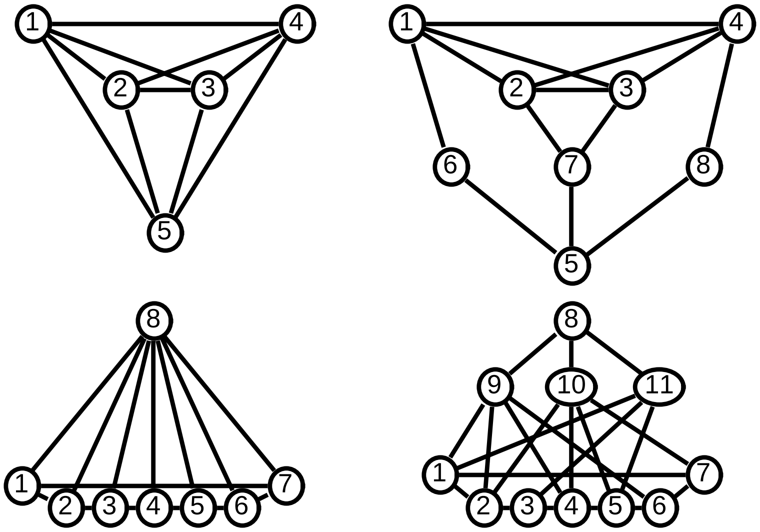


FIG. 3.7. *Examples of Theorem* 3.6.

2. Now let G be the southwest graph of Figure 3.7, $v_n = v_8$, $V_1 = \{v_1, v_2, v_4, v_6\}$, $V_2 = \{v_2, v_4, v_5, v_7\}$, and $V_3 = \{v_1, v_3, v_5\}$. Then G'' is the southeast graph and $\pi x = \sum_{i=1}^{7} x_i + 3x_8 \leq 3$ is a facet of $P(G)$, $M_i = 3$ for all i, $M_{ij} = 1$ for all i, j, $\pi_0 = 3$, and $\pi_8 = 3$. Applying the second part of Theorem 3.6, the facet of $P(G'')$

$$\sum_{j=1}^{7} 2x_j + \sum_{j=8}^{11} 3x_j \leq 9$$

is obtained. The conditions of the first part of Theorem 3.6 are not satisfied.

Consider now the special case of Theorem 3.6 where V_i, $i = 1, 2, 3$, are pairwise disjoint and $V_1 \cup V_2 \cup V_3 = V - \{v_n\}$. Then the coefficient of x_n in the original facet of $P(G)$ must be equal to π_0, and $\sum_{j=1}^{n-1} \pi_j x_j \leq \pi_0$ is a facet of the subgraph induced by $V - \{v_n\}$ (see Proposition 2.2). Thus the conditions of the first part of the theorem become

$$\begin{gathered} M_{12} + M_{13} + M_{23} \geq \pi_0, \\ M_{i,i+1} + M_{i,i+2} \geq M_i, \quad i = 1, 2, 3. \end{gathered} \tag{3.5}$$

Here

$$M_i = \max \left\{ \sum_{j \in V_{i+1} \cup V_{i+2}} \pi_j x_j \, : \, x \in P_I(G) \right\},$$

$$M_{i,i+1} = \max \left\{ \sum_{j \in V_{i+2}} \pi_j x_j \, : \, x \in P_I(G) \right\}.$$

Then

$$\pi_0 = \max\left\{ \sum_{j \in V_1 \cup V_2 \cup V_3} \pi_j x_j \,:\, x \in P_I(G) \right\}$$
$$\leq \sum_{i=1}^{3} \max\left\{ \sum_{j \in V_i} \pi_j x_j \,:\, x \in P_I(G) \right\} = M_{12} + M_{13} + M_{23}$$

and

$$M_i = \max\left\{ \sum_{j \in V_{i+1} \cup V_{i+2}} \pi_j x_j \,:\, x \in P_I(G) \right\}$$
$$\leq \sum_{\ell=1}^{2} \max\left\{ \sum_{j \in V_{i+\ell}} \pi_j x_j \,:\, x \in P_I(G) \right\} = M_{i,i+1} + M_{i,i+2}.$$

Therefore (3.5) holds, and the following results can be established.

Corollary 3.7. *Let $\sum_{j=1}^{n} \pi_j x_j \leq \pi_0$ be a facet of $P(G)$, and let V_i, $i = 1, 2, 3$, be disjoint subsets of $N(v_n)$ such that $V_1 \cup V_2 \cup V_3 \cup \{v_n\} = V$. Then*

$$\sum_{j=1}^{n-1} \pi_j x_j + \left(\sum_{i=1}^{3} M_{i,i+1} - \pi_0 \right) x_n + \sum_{i=1}^{3} M_{i+1,i+2} x_{n+i} \leq \sum_{i=1}^{3} M_{i,i+1}$$

is a facet of $P(G'')$.

Corollary 3.8. *Under the hypotheses of Corollary* 3.7*,*

1. *if each V_i, $i = 1, 2, 3$, contains a packing of G satisfying $\pi x = \pi_0$, then*

$$\sum_{j=1}^{n-1} \pi_j x_j + 2\pi_0 x_n + \sum_{i=1}^{3} \pi_0 x_{n+i} \leq 3\pi_0$$

is a facet of $P(G'')$;

2. *if the subgraphs of G induced by V_i, $i = 1, 2, 3$, are complete, then*

$$\sum_{j=1}^{n-1} \pi_j x_j + \left(\sum_{i=1}^{3} \Pi^i - \pi_0 \right) x_n + \sum_{i=1}^{3} \Pi^i x_{n+i} \leq \sum_{i=1}^{3} \Pi^i, \tag{3.6}$$

where $\Pi^i = \max_{j \in V_i} \{\pi_j\}$ is a facet of $P(G'')$.

Proof. Part 1 directly follows from Corollary 3.7, since $M_{i,i+1} = \pi_0$, $i = 1, 2, 3$, due to the presence in each V_i of a packing satisfying the inequality exactly. In part 2, since the subgraphs of G induced by V_i, $i = 1, 2, 3$, are complete, it holds that $M_{i,i+1} = \max_{j \in V_{i+2}} \{\pi_j\}$, and (3.6) follows from Corollary 3.7. □

Corollary 3.8 can be easily extended to the case of more than three subsets of nodes adjacent to v_n, $V_1, \ldots, V_k$.

Corollary 3.9. *Let $\sum_{j=1}^{n} \pi_j x_j \leq \pi_0$ be a facet of $P(G)$, and let V_i, $i = 1, \ldots, k$ be disjoint subsets of $N(v_n)$ such that $(\bigcup_{i=1}^{k} V_i) \cup \{v_n\} = V$. Then*

1. *if each V_i, $i = 1, \ldots, k$, contains a packing of G satisfying $\pi x = \pi_0$, then*

$$\sum_{j=1}^{n-1} \pi_j x_j + (k-1)\pi_0 x_n + \sum_{i=1}^{k} \pi_0 x_{n+i} \leq k\pi_0$$

is a facet of $P(G'')$;

2. *if the subgraphs of G induced by V_i, $i = 1, \dots, k$, are complete, then*

$$\sum_{j=1}^{n-1} \pi_j x_j + \left(\sum_{i=1}^{k} \Pi^i - \pi_0 \right) x_n + \sum_{i=1}^{k} \Pi^i x_{n+i} \leq \sum_{i=1}^{k} \Pi^i,$$

where $\Pi^i = \max_{j \in V_i} \{\pi_j\}$ is a facet of $P(G'')$.

Note that Proposition 2.2 along with the last statement of Corollary 3.9 leads to Theorem 2.3 of [5].

Moreover, consider Theorem 3.6 in the case where V_i, $i = 1, 2, 3$, are pairwise disjoint and complete, although v_n is not necessarily connected to the rest of the nodes of V. If a facet of G is obtained from a facet of its subgraph induced by $V - \{v_n\}$ by means of the usual lifting, the coefficient of x_n must be $\pi_0 - c$, where

$$c = \max \left\{ \sum_{j=1}^{n-1} \pi_j x_j : \ x \in P_I(G), x_\ell = 0 \ \forall v_\ell \in V_1 \cup V_2 \cup V_3 \cup \{v_n\} \right\}.$$

Assume also that $\pi x \leq \pi_0$ is not a clique inequality. Then for each $i = 1, 2, 3$ there will be a packing of G satisfying $\pi x = \pi_0$ and such that $x_j = 0$ for all $v_j \in V_i$ (Proposition 2.3). Therefore $M_i = \pi_0$, $i = 1, 2, 3$, and the conditions of the second part of Theorem 3.6 become $4\pi_0 - c \geq 3\pi_0$, which always holds, and $\pi_0 + c \geq 2 \max\{M_{12}, M_{13}, M_{23}\}$, leading to the following result.

Corollary 3.10. *Let $\sum_{j=1}^{n-1} \pi_j x_j \leq \pi_0$ be a facet of the polytope associated with the subgraph of G induced by $V - \{v_n\}$ other than a clique facet. If $N(v_n)$ can be subdivided into V_i, $i = 1, 2, 3$, disjoint subsets such that the subgraph induced by V_i is complete, and if $\pi_0 + c \geq 2 \max\{M_{12}, M_{13}, M_{23}\}$, then the inequality*

$$\sum_{j=1}^{n-1} 2\pi_j x_j + (\pi_0 - c) x_n + \sum_{i=1}^{3} (\pi_0 - c) x_{n+i} \leq 3\pi_0 - c$$

is a facet of $P(G'')$.

Note that $N(v_n)$ may always be subdivided into a number of disjoint subsets whose induced subgraphs are complete, say k subsets. Then a facet of $P(G'')$ of the form

$$\sum_{j=1}^{n-1} (k-1)\pi_j x_j + (\pi_0 - c) x_n + \sum_{i=1}^{k} (\pi_0 - c) x_{n+i} \leq k\pi_0 - c \tag{3.7}$$

is obtained if (3.7) is a valid inequality. This result matches Theorem 2.4 in [5]. The conditions for the validity of (3.7) extend that of Corollary 3.10 and can be consulted in the referenced paper.

3.3. Replacing a node v by $K_{1,|N(v)|}$.

Finally, consider the following construction. Given a graph $G = (V, E)$ and a selected node $v_n \in E$, assume w.l.o.g. that $N(v_n)$, the set of nodes adjacent to v_n, is given by $M = \{v_1, \dots, v_m\}$. Then a new graph G''' is obtained by

(i) introducing m new nodes v_{n+i} so that each vertex v_i in M is joined to v_{n+i} and

(ii) joining v_n to v_{n+i}, $i = 1, \dots, m$, only.

THEOREM 3.11. *Let $\sum_{j=1}^{n} \pi_j x_j \leq \pi_0$ be a facet of $P(G)$ and*

$$\mu(S) := \max \left\{ \sum_{j=1}^{n} \pi_j x_j : x \in P_I(G), x_\ell = 0 \ \forall v_\ell \in (M - S) \cup \{v_n\} \right\}$$

for $S \subseteq M$. Let $\mu_i := \mu(\{v_i\})$ for $i = 1, \ldots, m$. If $\sum_{i \in S} \mu_i - \mu(S) \geq (|S| - 1)(\pi_0 - \pi_n)$ for all $S \subseteq M$, then

$$\sum_{j=1}^{n-1} \pi_j x_j + \left(\sum_{i=1}^{m} \mu_i + (m-1)\pi_n - m\pi_0 \right) x_n$$
$$+ \sum_{i=1}^{m} (\mu_i + \pi_n - \pi_0) x_{n+i} \leq \sum_{i=1}^{m} \mu_i - (m-1)(\pi_0 - \pi_n) \tag{3.8}$$

is a facet of $P(G''')$.

Proof. Note that $\mu(\emptyset) = \pi_0 - \pi_n$ and $\mu(M) = \pi_0$.

Part 1. *The matrix X'.* Let $\{(X^k, 0)\}_{k=1}^{s}$, $\{(X^k, 1)\}_{k=s+1}^{n}$ be n independent points of $P_I(G)$ satisfying $\sum_{j=1}^{n} \pi_j x_j = \pi_0$, and let $(X^{n+i}, 0)$, $i = 1, \ldots, m$, be m vertices of $P(G)$ satisfying $x_n^{n+i} = 0$, $x_j^{n+i} = 0$ for all $j \neq i$ and $\sum_{j=1}^{n} \pi_j x_j^{n+i} = \mu_i$. Let X' be the $(n+m) \times (n+m)$ matrix

$$\left(\begin{array}{c|ccccc} X^1 & 1 & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ X^s & 1 & 0 & 0 & \ldots & 0 \\ \hline X^{s+1} & 0 & 1 & 1 & \ldots & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ X^n & 0 & 1 & 1 & \ldots & 1 \\ \hline X^{n+1} & 0 & 1 & 0 & \ldots & 0 \\ X^{n+2} & 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ X^{n+m} & 0 & 0 & 0 & \ldots & 1 \end{array} \right).$$

It can be rightly checked that the $n + m$ rows of X' are incidence vectors of vertices of $P(G''')$.

Part 2. *Regularity of X'.* The $n + m - 1$ first rows of X' are clearly independent, and assuming w.l.o.g. that

$$(X')^{n+m} = \sum_{k=1}^{n} \alpha_k (X')^k + \sum_{k=1}^{m-1} \beta_k (X')^{n+k}$$

it follows that $\sum_{k=1}^{s}\alpha_k = 0$, $\sum_{k=s+1}^{n}\alpha_k = m-1$, $\beta_k = -1$ for $k = 1,\dots,m-1$, $X^{n+m} = \sum_{k=1}^{n}\alpha_k X^k - \sum_{k=1}^{m-1} X^{n+k}$, and

$$\begin{aligned}\mu_m = \sum_{j=1}^{n-1}\pi_j x_j^{n+m} &= \sum_{j=1}^{n-1}\pi_j\left(\sum_{k=1}^{n}\alpha_k x_j^k - \sum_{k=1}^{m-1} x_j^{n+k}\right)\\ &= \sum_{k=1}^{n}\alpha_k\sum_{j=1}^{n-1}\pi_j x_j^k - \sum_{k=1}^{m-1}\sum_{j=1}^{n-1}\pi_j x_j^{n+k}\\ = \pi_0\sum_{k=1}^{s}\alpha_k + (\pi_0-\pi_n)\sum_{k=s+1}^{n}\alpha_k - \sum_{k=1}^{m-1}\mu_k &= (m-1)(\pi_0-\pi_n) - \sum_{k=1}^{m-1}\mu_k,\end{aligned}$$

which is not compatible with the hypothesis when $S = M$.

Part 3. *Solution of* $X'\pi' = \mathbf{1}$. The coefficients of (3.8), divided by its right-hand side, satisfy all the equations.

Part 4. *Nonnegativity of* π'. The conditions of the theorem imply the coefficients in (3.8) are nonnegative.

Part 5. *Validity of* (3.8). Let x^1 be a vertex of $P(G''')$. If $x_n^1 = 1$, then $x_{n+i}^1 = 0$, $i = 1,\dots,m$, and the point x^2 given by $x_j^2 = x_j^1$ for $j = 1,\dots,n-1$, $x_n^2 = 0$, is a vertex of $P(G)$; thus

$$\begin{aligned}&\sum_{j=1}^{n-1}\pi_j x_j^1 + \left(\sum_{i=1}^{m}\mu_i + (m-1)\pi_n - m\pi_0\right)x_n^1\\ &= \sum_{j=1}^{n}\pi_j x_j^2 + \sum_{i=1}^{m}\mu_i + (m-1)\pi_n - m\pi_0 \le \sum_{i=1}^{m}\mu_i + (m-1)\pi_n - (m-1)\pi_0.\end{aligned}$$

If $x_n^1 = 0$ and $x_{n+i}^1 = 1$, $i = 1,\dots,m$, the point x^3 given by $x_j^3 = x_j^1$ for $j = 1,\dots,n-1$, $x_n^3 = 1$, is a vertex of $P(G)$ and

$$\begin{aligned}&\sum_{j=1}^{n-1}\pi_j x_j^1 + \sum_{i=1}^{m}(\mu_i + \pi_n - \pi_0)x_{n+i}^1\\ &= \sum_{j=1}^{n-1}\pi_j x_j^3 + \sum_{i=1}^{m}(\mu_i + \pi_n - \pi_0) \le \pi_0 - \pi_n + \sum_{i=1}^{m}\mu_i + m\pi_n - m\pi_0.\end{aligned}$$

If $x_n^1 = x_{n+i}^1 = 0$ for $i \in S$ and $x_{n+i}^1 = 1$, for $i \in \bar{S} := M - S$, using the above defined point x^2

$$\begin{aligned}\sum_{j=1}^{n-1}\pi_j x_j^1 + \sum_{i=1}^{m}(\mu_i + \pi_n - \pi_0)x_{n+i}^1 &= \sum_{j=1}^{n-1}\pi_j x_j^2 + \sum_{i\in\bar{S}}(\mu_i + \pi_n - \pi_0)\\ \le \mu(S) + \sum_{i\in\bar{S}}\mu_i + |\bar{S}|(\pi_n - \pi_0) &= \mu(S) + \sum_{i=1}^{m}\mu_i + |\bar{S}|(\pi_n - \pi_0) - \sum_{i\in S}\mu_i\\ &\le \sum_{i=1}^{m}\mu_i - (m-1)(\pi_0 - \pi_n),\end{aligned}$$

the last inequality being by hypothesis. □

Example. Let G be the left-hand graph of Figure 3.8, $v_n = v_7$. Then G''' is the right-hand graph of Figure 3.8 and $\pi x = x_1 + x_2 + x_3 + 2x_4 + x_5 + x_6 + x_7 \le 3$ is a

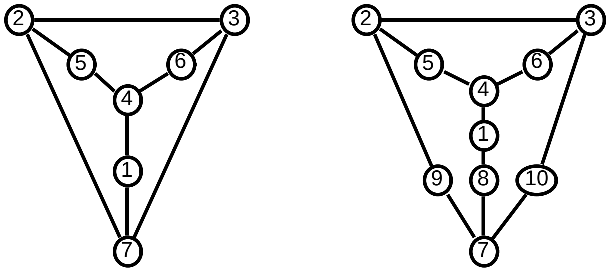


FIG. 3.8. *Example of Theorem* 3.11.

facet of $P(G)$, $m = 3$, $\mu_i = 3$ for $i = 1, 2, 3$, $\pi_0 = 3$, and $\pi_7 = 1$. Applying Theorem 3.11, the facet of $P(G''')$

$$x_1 + x_2 + x_3 + 2x_4 + x_5 + x_6 + 2x_7 + x_8 + x_9 + x_{10} \leq 5$$

is obtained.

Remark 3. Subdivision of a star (Theorem 2.3 of [2]) follows from Theorem 3.11. Moreover, it is not necessary to suppose (see [2]) that k and p are relatively prime.

COROLLARY 3.12. *Under the conditions of Theorem* 3.11*, assume* $|V| = n = m + 1$*; that is to say, the selected node* v_n *is adjacent to the rest of the nodes of* V *(and consequently* $\pi_n = \pi_0$*). Then*

$$\sum_{j=1}^{m} \pi_j x_j + \left(\sum_{i=1}^{m} \pi_i - \pi_0\right) x_n + \sum_{i=1}^{m} \pi_i x_{n+i} \leq \sum_{i=1}^{m} \pi_i \tag{3.9}$$

is a facet of $P(G''')$.

Proof. Here $\mu_i = \pi_i$ for all i, $\pi_n = \pi_0$ and (3.9) follows straightforward. □

Remark 4. Consider any graph $G = (V, E)$ with $|V| = n - 1$ and associated facet $\sum_{j=1}^{n-1} \pi_j x_j \leq \pi_0$. The graph and facet obtained by means of Proposition 1 of [17] (see also [14]) can be also obtained by (i) adding the node n, connecting it to the $n - 1$ nodes of V, and lifting the facet in the usual way to obtain $\sum_{j=1}^{n-1} \pi_j x_j + \pi_0 x_n \leq \pi_0$, and (ii) applying Corollary 3.12.

4. Fans.

In this section it is shown how the foregoing results can be employed to construct a new class of facet defining graphs, which are called *fans*. The details are given in the following.

A fan $A = (V_A, E_A)$ consists of

1. a complete graph $G = (V, E)$ with $V = \{v_1, v_2, \ldots, v_n\}$,
2. an odd path disjoint to G, going from node O to node D, and containing n so-called *connecting nodes* $O = u_1, u_2, \ldots, u_n = D$—not necessarily different—arranged in the given order, with odd distances when different,
3. an odd path between each node v_i and its associated connecting node u_i.

Example. Figure 4.1 shows a fan constructed from a complete 6-graph. Here $O = u_1 = u_2$, $D = u_4 = u_5 = u_6$, and the central black-filled node is u_3. Note that a fan can also be seen as a clique plus a collection of odd holes going through it in a certain way.

THEOREM 4.1. *Let* $A = (V_A, E_A)$ *be a fan. The inequality*

$$\sum_{i=1}^{n} x_i + \sum_{i=n+1}^{|V_A|} (\delta_i - 1) x_i \leq \frac{|V_A| - 1}{2}, \tag{4.1}$$

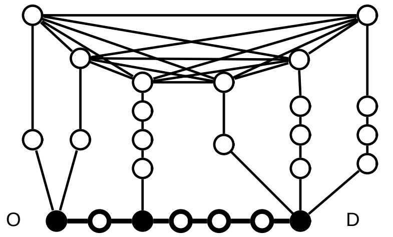


FIG. 4.1. *Fan. Connecting nodes are black-filled. The path through connecting nodes is the thick one.*

where δ_i denotes the incidence degree of node v_i, is a facet of $P(A)$.

Proof. We call d the number of different connecting nodes in A, $p_1,\dots,p_d$ the different connecting nodes arranged in the same order as above, n_i the number of nodes between v_i and u_i, and m_i the number of nodes in the path $O-D$ between p_i and p_{i+1}. A can be obtained from the complete graph G by using exclusively the transformations given in Theorem 3.1, Remark 4, and Corollary 3.2 as shown in the following iterative procedure.

Step 1. Consider the complete graph G with nodes $\{v_1,\dots,v_n\}$.

Step 2. Apply Remark 4 to G to obtain G^1. Let $v_{n+1},\dots,v_{2n}$ be the nodes adjacent to $v_1,\dots,v_n$ respectively, and let v_{2n+1} be the special node.

Step 3.

(a) Set $t=1$ and apply Theorem 3.1 to G^1, taking V_1 equal to the set of nodes v_k with $k\in\{n+1,\dots,n+\delta_{p_1}-1\}$, V_2 equal to the set of nodes v_k with $k\in\{n+\delta_{p_1},\dots,2n\}$, and special node v_{2n+1}, obtaining G^2 with two new nodes v_{2n+2} and v_{2n+3}.

(b) For $i=2,\dots,d-1$: Set $t=t+1$ and apply Theorem 3.1 to G^t, taking V_1 equal to the set of nodes v_k with $k\in\{n+\delta_{p_1}+\sum_{j=2}^{i-1}(\delta_{p_j}-2),\dots,n+\delta_{p_1}+\sum_{j=2}^{i}(\delta_{p_j}-2)-1\}\cup\{v_{|G^t|-2}\}$, V_2 equal to the set of nodes v_k with $k\in\{n+\delta_{p_1}+\sum_{j=2}^{i}(\delta_{p_j}-2),\dots,2n\}$, and special node $v_{|G^t|}$, obtaining G^{t+1} with two new nodes $v_{|G^t|+1}$ and $v_{|G^t|+2}$.

Step 4. For $i=1,\dots,n$: Set $t=t+1$ and apply Corollary 3.2, on the unique edge of the form (v_i,v_a) of G^t with v_a out of the clique, $(n_i-1)/2$ times, to obtain G^{t+1}.

Step 5. Let $(v_{q_1},\dots,v_{q_d})$ be the path linking the connecting nodes of G^{t+1}. For $i=1,\dots,d-1$: Set $t=t+1$ and apply Corollary 3.2 to G^t, on the edge of the subpath between v_{q_i} and $v_{q_{i+1}}$ which is incident to v_{q_i}, $(m_i-1)/2$ times.

Notice that the facet of $P(G^1)$

$$\sum_{i=1}^{2n}x_i+(n-1)x_{2n+1}=\sum_{i=1}^{n}x_i+\sum_{i=n+1}^{2n+1}(\delta_i-1)x_i\le n$$

is obtained in Step 2 from the facet $\sum_{i=1}^{n}x_i\le 1$ of $P(G)$, and the coefficients match those in (4.1). Moreover, when Step 3 is applied for the ith time, the maxima M_1 and M_2 needed in Theorem 3.1 are

$$M_1=n+i-|V_1|,\qquad M_2=n+i-|V_2|,$$

the coefficient of the node to be disconnected from $V_1\cup V_2$ and the right-hand side of the facet to be modified are (using the hypothesis of induction) $\pi^a=|V_1|+|V_2|-1$

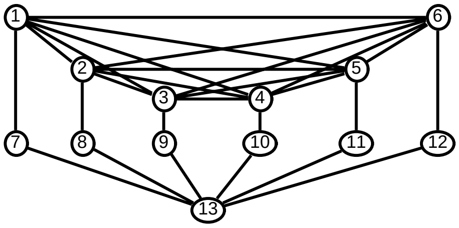


FIG. 4.2. *Graph G^1 with associated facet $\sum_{i=1}^{12} x_i + 5x_{13} \leq 6$.*

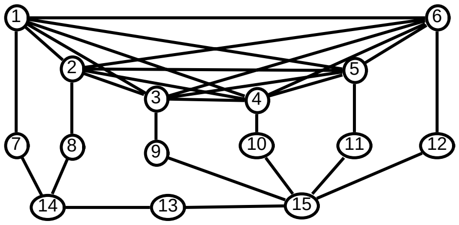


FIG. 4.3. *Graph G^2 with associated facet $\sum_{i=1}^{13} x_i + 2x_{14} + 4x_{15} \leq 7$.*

and $\pi^b = n + i - 1$, respectively. Then the new coefficient of the disconnected node is $M_1 + M_2 + \pi^a - 2\pi^b = 1$ and its new incidence degree is 2, the coefficient of the new node associated with V_1 is $M_2 + \pi^a - \pi^b = |V_1|$ and its incidence degree is $|V_1| + 1$, the coefficient of the new node associated with V_2 is $M_1 + \pi^a - \pi^b = |V_2|$ and its incidence degree is $|V_2| + 1$, and the new right-hand side of the inequality is $M_1 + M_2 + \pi^a - \pi^b = n + i$. That is to say, each time Step 3 is applied, two new nodes are added to the graph and the new facet still matches (4.1). In Steps 4 and 5, the optimum value of the auxiliary problem needed in Corollary 3.2 is always the right-hand side of the current facet plus one. Then two new nodes with coefficients 1 and incidence degree two are added to the graph, and the right-hand side is increased by one. Thus when $|V_A| - 2n - 1$ new nodes have been added to G^1, the right-hand side becomes

$$n + \frac{|V_A| - 2n - 1}{2} = \frac{|V_A| - 1}{2}$$

and the proof is complete. □

Example. Consider again the graph of Figure 4.1. Here $d = 3$, $m_1 = 1$, $m_2 = 3$, $n_1 = 1$, $n_2 = 1$, $n_3 = 3$, $n_4 = 1$, $n_5 = 3$, $n_6 = 3$. The initial facet, associated with the complete 6-graph, is $\sum_{i=1}^{6} x_i \leq 1$. Then G^1 is the graph given in Figure 4.2. In Step 3, for $i = 1$, we have $V_1 = \{v_7, v_8\}$ and $V_2 = \{v_9, v_{10}, v_{11}, v_{12}\}$ and then $M_1 = n + i - |V_1| = 6 + 1 - 2 = 5$, $M_2 = n + i - |V_2| = 6 + 1 - 4 = 3$, $\pi_{13} = 5$, and $\pi_0 = 6$. Therefore

$$\sum_{i=1}^{12} x_i + x_{13} + 2x_{14} + 4x_{15} \leq 7$$

is a facet associated with the new graph G^2 given in Figure 4.3. For $i = 2$, we have $V_1 = \{v_9, v_{13}\}$ and $V_2 = \{v_{10}, v_{11}, v_{12}\}$ and then $M_1 = n + i - |V_1| = 6 + 2 - 2 = 6$, $M_2 = n + i - |V_2| = 6 + 2 - 3 = 5$, $\pi_{15} = 4$, and $\pi_0 = 7$. Therefore

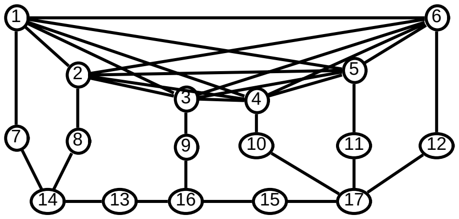


FIG. 4.4. *Graph G^3 with associated facet* $\sum_{i=1}^{13} x_i + 2x_{14} + x_{15} + 2x_{16} + 3x_{17} \le 8$.

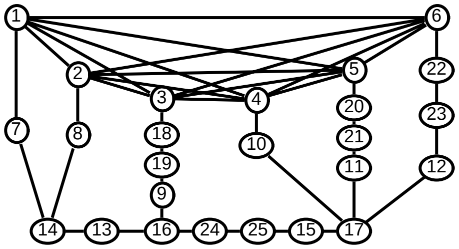


FIG. 4.5. *Graph G^{11} with associated facet* $\sum_{i=1}^{13} x_i + 2x_{14} + x_{15} + 2x_{16} + 3x_{17} + \sum_{i=18}^{25} x_i \le 12$.

$$\sum_{i=1}^{13} x_i + 2x_{14} + x_{15} + 2x_{16} + 3x_{17} \le 8$$

is a facet associated with the new graph G^3 given in Figure 4.4. By using Step 4 one time for edges (v_3, v_9), (v_5, v_{11}), and (v_6, v_{12}), and then Step 5 one time for edge (v_{16}, v_{15}), the fan A of the example is obtained (see Figure 4.5), and the associated facet of $P(A)$ is

$$\sum_{i=1}^{13} x_i + 2x_{14} + x_{15} + 2x_{16} + 3x_{17} + \sum_{i=18}^{25} x_i \le 12.$$

If a new node is added to the clique of a fan A to obtain a new fan, the facet can be lifted in the usual way and all the coefficients of the new nodes become zero. To see it, add a node v' and link it to all the nodes of the complete subgraph in A. By Proposition 2.3, v' is lifted with coefficient 0. Now add a node v'' linked only to a connecting node of A; since there must exist a packing not containing v'' and satisfying the inequality of the facet exactly, v'' is also lifted with coefficient 0. The rest of the new nodes are on the path which links v' and v'', and are also lifted with coefficient 0 since they are not linked to any node of A. Consequently, the following result can be established.

COROLLARY 4.2. *Let $A = (V_A, E_A)$ be a fan, and let N be the set of nodes of its associated clique. The inequalities*

$$\sum_{i \in H} x_i + \sum_{i \in V_H} (\delta_i - 1)x_i \le \frac{|V_H| - 1}{2}$$

are facets of $P(A)$ for all $H \subset N$, $|H| \ge 2$, where (V_H, E_H) is the unique subgraph of A containing the nodes of H which is a fan.

Example. Consider again the fan of Figure 4.5. By enumerating all the subsets of $\{v_1, \dots, v_6\}$ of cardinality at least 2, 56 facets of $P(A)$ are obtained. Some of them, corresponding to the subsets of N $\{v_1, v_2, v_3, v_4, v_6\}$, $\{v_2, v_3, v_5\}$, $\{v_4, v_5, v_6\}$, $\{v_1, v_2\}$, and $\{v_1, v_6\}$, respectively, are given in the following:

$$\begin{gathered}
\sum_{i=1}^{4} x_i + \sum_{i=6}^{10} x_i + x_{12} + x_{13} + 2x_{14} + x_{15} + 2x_{16} + 2x_{17} \\
\qquad + x_{18} + x_{19} + \sum_{i=22}^{25} x_i \leq 10, \\
x_2 + x_3 + x_5 + x_8 + x_9 + x_{11} + \sum_{i=13}^{15} x_i + 2x_{16} + \sum_{i=17}^{21} x_i + x_{24} + x_{25} \leq 8, \\
x_4 + x_5 + x_6 + x_{10} + x_{11} + x_{12} + 2x_{17} + x_{20} + x_{21} + x_{22} + x_{23} \leq 5, \\
x_1 + x_2 + x_7 + x_8 + x_{14} \leq 2, \\
x_1 + x_6 + x_7 + x_{12} + x_{13} + x_{14} + x_{15} + x_{16} + x_{17} + x_{22} + x_{23} + x_{24} + x_{25} \leq 6.
\end{gathered}$$

Note that the last two facets match odd holes traversing a pair of nodes of the clique.

Furthermore, it is not difficult to prove that the unique 2-connected subgraphs of a fan are either fans or fans plus additional nodes in the clique (only connected to the other nodes in the clique). Then, by Proposition 2.4, the unique subgraphs of a fan which are facet defining are fans.

5. Other results.

Two different methods to obtain facets of graphs which are combinations of other graphs are given in this section.

5.1. An alternative lifting procedure.

In this subsection an alternative facet lifting method is given. It can be used when two nodes are going to be added to the graph and the sets of nodes not to be connected to each of the new nodes are disjoint packings and one of them is maximal.

DEFINITION 5.1. *Given a graph $G = (V, E)$ with $V = \{v_1, \dots, v_n\}$ and two packings $E_1, E_2 \subset V$ such that $E_1 \cap E_2 = \emptyset$, we denote by $G^*(E_1, E_2)$ the graph obtained by adding to G* (i) *two new nodes v_{n+1} and v_{n+2} and* (ii) *$|V - E_1| + |V - E_2|$ edges connecting v_{n+1} to the nodes in $V - E_1$ and v_{n+2} to the nodes in $V - E_2$.*

THEOREM 5.2. *Let $E_1, \dots, E_m$ be the incidence vectors of all the maximal packings of the graph $G = (V, E)$ with $V = \{v_1, \dots v_n\}$ and assume they are independent. Let $\pi x \leq 1$ be a facet of $P(G)$ associated with $E_1, \dots, E_m$ along with $n - m$ nonmaximal packings $E_{m+1}, \dots, E_n$. Let X be the square matrix whose rows are the incidence vectors of the packings E_i, $i = 1, \dots, n$, let $M = X^{-1} = (m_{ij})$ be the inverse of X, and let $B \subset V$ be a packing of G such that for some i, $1 \leq i \leq m$, $E_i \cap B = \emptyset$ holds and*

$$\pi_B \leq \min \left\{ 1 + m_{Bi}, \min \left\{ (1 + m_{Bi}) \frac{\pi_\ell}{m_{\ell i}} : \quad m_{\ell i} > 0 \right\} \right\},$$

where $\pi_B = \sum_{j \in B} \pi_j$ and $m_{Bi} = \sum_{j \in B} m_{ji}$. Then the inequality

$$\sum_{j=1}^{n} \left(\pi_j - m_{ji} \frac{\pi_B}{1 + m_{Bi}} \right) x_j + \frac{\pi_B}{1 + m_{Bi}} x_{n+1} + \left(1 - \frac{\pi_B}{1 + m_{Bi}} \right) x_{n+2} \leq 1 \tag{5.1}$$

is a facet of $P(G^*(E_i, B))$.

Proof. For clarity of exposition, suppose that the packings $\{E_1, \ldots, E_n\}$ are reordered in such a way that E_n is a maximal packing such that $E_n \cap B = \emptyset$, i.e., such that $i = n$. Then let $B \subset V$, $E_n \cap B = \emptyset$, be a packing of G.

Part 1. *The matrix* X'. Let $(b_1, \ldots, b_n)$ be the incidence vector of packing B in the graph G, and let X' be the $(n+2) \times (n+2)$ matrix

$$\left(\begin{array}{ccc|cc} & & & 0 & 0 \\ & X & & \vdots & \vdots \\ & & & 0 & 0 \\ & & & 1 & 0 \\ \hline b_1 & \ldots & b_n & 0 & 1 \\ 0 & \ldots & 0 & 1 & 1 \end{array}\right).$$

It should be clear that the rows of X' are incidence vectors of vertices of $P(G^*(E_n, B))$.

Parts 2 *and* 3. *Regularity of* X' *and solution of* $X'\pi' = \mathbf{1}$. The system $X'\pi' = \mathbf{1}$ can be split into

$$X \begin{pmatrix} \pi'_1 \\ \vdots \\ \pi'_{n-1} \\ \pi'_n \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \pi'_{n+1} \end{pmatrix} = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 1 \end{pmatrix},$$

$$(b_1, \ldots, b_n, 0, 1) \cdot \begin{pmatrix} \pi'_1 \\ \vdots \\ \pi'_{n+2} \end{pmatrix} = 1,$$

$$\pi'_{n+1} + \pi'_{n+2} = 1.$$

Given that $\pi = M\mathbf{1}_n$, it follows that

$$\begin{pmatrix} \pi'_1 \\ \vdots \\ \pi'_{n-1} \\ \pi'_n \end{pmatrix} + M \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \pi'_{n+1} \end{pmatrix} = \begin{pmatrix} \pi_1 \\ \vdots \\ \pi_{n-1} \\ \pi_n \end{pmatrix},$$

$$\sum_{j \in B} \pi'_j + \pi'_{n+2} = 1,$$

$$\pi'_{n+1} + \pi'_{n+2} = 1,$$

and finally

$$\pi'_j = \pi_j - m_{jn} \frac{\pi_B}{1 + m_{Bn}}, j = 1, \ldots, n, \quad \pi'_{n+1} = \frac{\pi_B}{1 + m_{Bn}}, \quad \pi'_{n+2} = 1 - \frac{\pi_B}{1 + m_{Bn}}.$$

Part 4. *Nonnegativity of* X'. Under the conditions given in the theorem, the coefficients π'_j are nonnegative.

Part 5. *Validity of* (5.1). Consider the graph $G^1 = (V^1, E^1)$ obtained when the node v_{n+1} and its incident edges are added to G, i.e., $G^1 = (V \cup \{v_{n+1}\}, E \cup \{(v_j, v_{n+1}) : \ v_j \in V - E_n\})$. In particular, v_{n+1} will be connected to all the nodes of packing B. Hence, the $n+1$ first coefficients of each row $(X')^i$, which represented

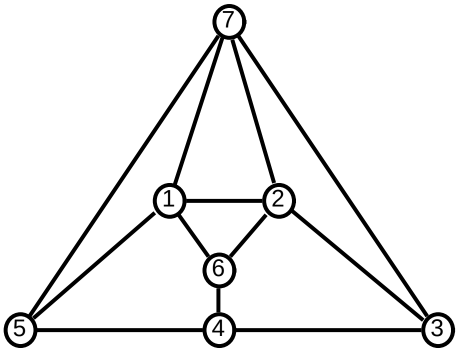


FIG. 5.1. *Graph for the example of Theorem* 5.2.

a maximal packing in G, define an incidence vector of a maximal packing of G^1. Moreover, these are all the maximal packings of G^1.

The graph $G^*(E_n, B)$ is obtained from G^1 by adding the node v_{n+2} to V^1 and the edges connecting v_{n+2} with all the nodes in $V^1 - (B \cup \{v_{n+1}\})$ to E^1. In particular, v_{n+2} will be connected to the nodes of packing E_n. Consequently, the n first rows of X' become incidence vectors of maximal packings of G^*, except the incidence vector of B, if it was one of them. In any case, rows $(X')^{n+1}$ and $(X')^{n+2}$ are the incidence vectors of the two unique new maximal packings of G^*. Therefore, given that any packing of G^* must be included in at least one of the $n+2$ maximal packings, (5.1) is a valid inequality. □

Example. Consider the graph of Figure 5.1. The subgraph G induced by nodes $\{v_1, \dots, v_5\}$ is a cycle of length 5 and therefore it has five maximal and independent packings $\{v_1, v_3\}$, $\{v_1, v_4\}$, $\{v_2, v_4\}$, $\{v_2, v_5\}$, and $\{v_3, v_5\}$ determining the facet $\pi x \leq 1$ with $\pi = \frac{1}{2}(1,1,1,1,1)$. The node v_6 is connected to all the nodes of the cycle except those in the last packing. The node v_7 is connected to all the nodes of the cycle except v_4 and is not connected to v_6, and so the graph of Figure 5.1 matches $G^*(E_5, B)$ with $E_5 = \{v_3, v_5\}$ and $B = \{v_4\}$. According to the established order of the packings, the inverse of X is

$$M = \frac{1}{2}\begin{pmatrix} 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 & 1 \\ -1 & 1 & -1 & 1 & 1 \end{pmatrix}.$$

Then it is obtained that $\pi_B = \frac{1}{2}$, $m_{B5} = m_{45} = \frac{1}{2}$, and $(\pi_j - \frac{1}{3}m_{j5})_{j=1}^5 = \frac{1}{3}(2,2,1,1,1)$. Since all the coefficients are nonnegative, the inequality

$$2x_1 + 2x_2 + x_3 + x_4 + x_5 + x_6 + 2x_7 \leq 3$$

is a facet of the graph of Figure 5.1. Note that the usual lifting of the original facet gives a different facet:

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_7 \leq 2.$$

COROLLARY 5.3. *Let $E_1, \dots, E_m$ be the incidence vectors of all the maximal packings of the graph $G = (V, E)$ with $V = \{v_1, \dots v_n\}$ and assume they are independent. Let $\pi x \leq 1$ be a facet of $P(G)$ associated with $E_1, \dots, E_m$ along with $n - m$*

nonmaximal packings $E_{m+1}, \dots, E_n$. Let X be the square matrix whose rows are the incidence vectors of packings E_i, $i = 1, \dots, n$, and let $M = X^{-1} = (m_{ij})$ be the inverse of X. Assume E_i and E_k are two maximal packings such that $E_i \cap E_k = \emptyset$ and also that for all $j = 1, \dots, n$, $\sum_{\ell \neq i} m_{j\ell} \geq 0$ holds. Then the inequality

$$\sum_{j \in V - E_i} (\pi_j - m_{ji}) x_j \leq 1 \tag{5.2}$$

is a facet of the subgraph of G induced by $V - E_i$.

Proof. The assumptions of the corollary are those of Theorem 5.2 when $B = E_k$. Reordering the packings in such a way that $E_n = E_i$, and given that B is, in this case, a maximal packing, it holds that

$$\pi_B = \sum_{j \in B} \pi_j = 1,$$
$$m_{Bn} = \sum_{j \in B} m_{jn} = 0,$$

and then the coefficients of x_{n+1} and x_{n+2} in the inequality (5.1) are 1 and 0, respectively. Since $E_n \cup \{v_{n+1}\}$ is a packing of $G^*(E_n, B)$ and the coefficient of x_{n+1} is 1, the coefficients of x_j for all $j \in E_n$ must be equal to zero. Furthermore,

$$0 \leq \pi_j - m_{jn} \frac{\pi_B}{1 + m_{Bn}} = \sum_{\ell=1}^{n} m_{j\ell} - m_{jn} = \sum_{\ell=1}^{n-1} m_{j\ell},$$

which is the assumption of the corollary, and this leads to the facet of $P(G^*(E_n, B))$

$$\sum_{j \in V - E_n} (\pi_j - m_{jn}) x_j + x_{n+1} \leq 1. \tag{5.3}$$

Consequently, (5.3) is also a facet of the graph induced by the nodes with positive coefficients. Finally, v_{n+1} is connected to all the other nodes in this graph (the reason for its coefficient to be 1). So, if this node is deleted only one packing satisfying (5.3) exactly is eliminated. Then (5.2) is a facet of the remaining subgraph and the proof is complete. □

Example. Consider the graph (a) of Figure 5.2, with associated facet $\sum_{j=1}^{5} x_j + 2x_6 \leq 2$, whose maximal packings are $\{v_1, v_3\}$, $\{v_1, v_4\}$, $\{v_2, v_4\}$, $\{v_2, v_5\}$, $\{v_3, v_5\}$, and $\{v_6\}$. For each of these packings, another maximal and disjoint packing exists. The inverse of the matrix whose rows are the incidence vectors of the maximal packings in the indicated order is

$$M = \frac{1}{2} \begin{pmatrix} 1 & 1 & -1 & 1 & -1 & 0 \\ 1 & -1 & 1 & 1 & -1 & 0 \\ 1 & -1 & 1 & -1 & 1 & 0 \\ -1 & 1 & 1 & -1 & 1 & 0 \\ -1 & 1 & -1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

Taking $E_i = \{v_6\}$, the difference between $\pi = \frac{1}{2}(1,1,1,1,1,2)$ and the last column of M is $\frac{1}{2}(1,1,1,1,1,0)$, and it follows that $\sum_{j=1}^{5} x_j \leq 2$ is a facet of the cycle given by the first five nodes. Taking E_i equal to each of the remaining packings, the facets $x_4 + x_5 + x_6 \leq 1$, $x_2 + x_3 + x_6 \leq 1$, $x_1 + x_5 + x_6 \leq 1$, $x_3 + x_4 + x_6 \leq 1$, and $x_1 + x_2 + x_6 \leq 1$, associated with the graphs (c)–(g), respectively, are obtained.

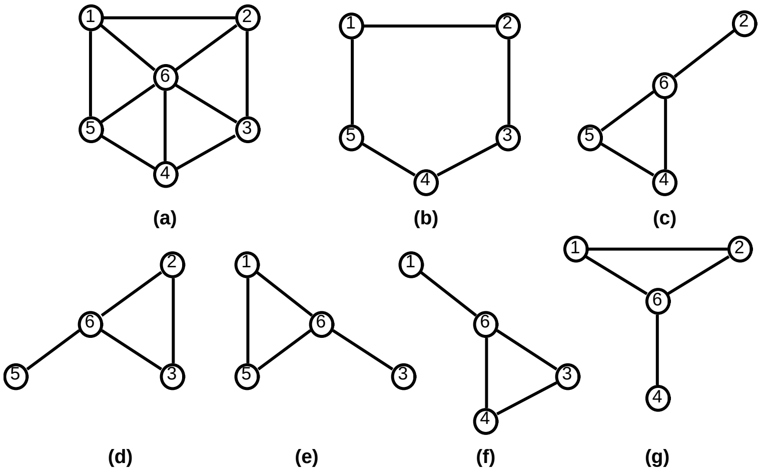


FIG. 5.2. *Graphs of the example of Corollary* 5.3.

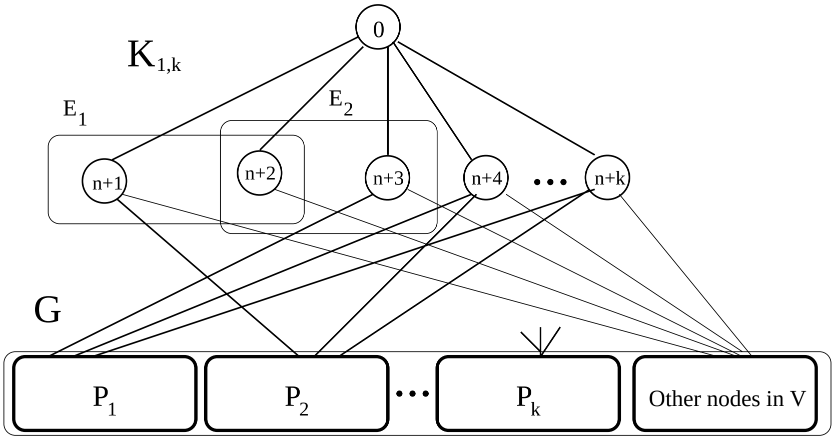


FIG. 5.3. *Cooking a facet.*

5.2. Cooked facets. Finally, a result is presented which relates those obtained in section 3, in the following sense. The nodes of an initial graph G are going to be connected to the degree one nodes of an additional graph $K_{1,k}$ in order to obtain a new graph and an associated facet, the so-called *cooked* facet. This facet will depend on the connections between the initial graph and $K_{1,k}$, connections which are based on the packings of a k-node graph to be freely chosen.

Let us consider the following construction (see Figure 5.3). Given a graph $G = (V, E)$ with $V = \{v_1, \dots, v_n\}$ and a facet of $P(G)$ $\pi x \leq 1$ associated with the n packings $P_1, \dots, P_n$, a new graph G^x is obtained by

(i) choosing k pairwise disjoint packings, say $P_1, \dots, P_k$,

(ii) choosing a graph $G^a = (V^a, E^a)$ with $|V^a| = k$ and a facet $\mu x \leq 1$ of $P(G^a)$ with associated packings $E_1, \dots, E_k$,

(iii) adding $k + 1$ nodes $\{v_{n+1}, \dots, v_{n+k}, v_0\}$ to V,

(iv) joining the nodes in P_i to the nodes in $\{v_{n+1}, \dots, v_{n+k}\} - E_i$, $i = 1, \dots, k$, and

(v) joining v_0 to $v_{n+1}, \ldots, v_{n+k}$.

THEOREM 5.4. *The inequality*

$$\left(\sum_{\ell=1}^{k} \mu_\ell - 1\right) \sum_{j=1}^{n} \pi_j x_j + \sum_{j=1}^{k} \mu_j x_{n+j} + x_0 \le \sum_{\ell=1}^{k} \mu_\ell \tag{5.4}$$

is a facet of $P(G^x)$.

Proof.

Part 1. *The matrix* X'. Let X' be the $(n+k+1) \times (n+k+1)$ matrix

$$\begin{pmatrix} X_1 & X_2 & \mathbf{0}_{k\times 1} \\ X_3 & \mathbf{0}_{n\times k} & \mathbf{1}_{n\times 1} \\ \mathbf{0}_{1\times n} & \mathbf{1}_{1\times k} & 0 \end{pmatrix},$$

where X_3 and X_2 are the matrices whose rows are the incidence vectors of the packings of G and G^a determining π and μ, respectively, and X_1 is the matrix whose rows are the incident vectors of the disjoint packings $P_1, \ldots, P_k$.

Part 2. *Regularity of* X'. Since each row of X_1 is equal to some row of X_3, subtracting the adequate rows of X' results in the matrix with the same rank

$$X'' = \begin{pmatrix} \mathbf{0}_{k\times n} & X_2 & -\mathbf{1}_{k\times 1} \\ X_3 & \mathbf{0}_{n\times k} & \mathbf{1}_{n\times 1} \\ \mathbf{0}_{1\times n} & \mathbf{1}_{1\times k} & 0 \end{pmatrix}.$$

Suppose, given a vector of multipliers $t = (t_1, \ldots, t_{n+k+1})$, that $X''t = \mathbf{0}$. Then

$$\begin{aligned} X_2(t_{n+1}, \ldots, t_{n+k})^T &= (t_{n+k+1}, \ldots, t_{n+k+1}), \\ X_3(t_1, \ldots, t_n)^T &= (-t_{n+k+1}, \ldots, -t_{n+k+1}), \end{aligned}$$

and $\sum_{j=n+1}^{n+k} t_j = 0$. Since $\pi = X_3^{-1}\mathbf{1}$ and $\mu = X_2^{-1}\mathbf{1}$, it follows that

$$(t_{n+1}, \ldots, t_{n+k}) = t_{n+k+1}\mu, \quad (t_1, \ldots, t_n) = -t_{n+k+1}\pi.$$

Now $\sum_{j=n+1}^{n+k} t_j = t_{n+k+1} \sum_{j=1}^{k} \mu_j = 0$, and thus $t_{n+k+1} = 0$, implying $t = \mathbf{0}$.

Part 3. *Solution of* $X'\pi' = \mathbf{1}$. The unique solution of the system is given by the coefficients of (5.4) divided by its right-hand side.

Part 4. *Nonnegativity of* π'. It is clear that the coefficients in (5.4) cannot be negative.

Part 5. *Validity of* (5.4). All the packings of G^x will be considered. Packings including v_0 are those of the form $P \cup \{v_0\}$, where P is a packing of G. For these, (5.4) becomes

$$\left(\sum_{\ell=1}^{k} \mu_\ell - 1\right) \sum_{j=1}^{n} \pi_j x_j + 1 \le \sum_{\ell=1}^{k} \mu_\ell - 1 + 1.$$

Now packings not including nodes of V are subsets of $\{v_{n+1}, \ldots, v_{n+k}\}$. Then (5.4) becomes

$$\sum_{j=1}^{k} \mu_j x_{n+j} \le \sum_{\ell=1}^{k} \mu_\ell.$$

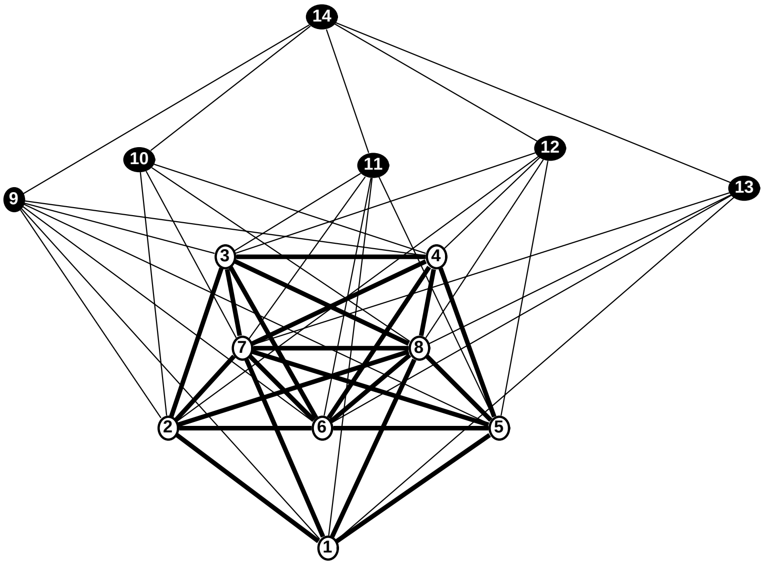


FIG. 5.4. *Graph for the example of Theorem* 5.4.

The rest of the packings contain nodes in V and, possibly, nodes in $\{v_{n+1},\ldots, v_{n+k}\}$. If a node of P_j, $j = 1,\ldots,k$, belongs to the packing, the nodes of $\{v_{n+1},\ldots,v_{n+k}\}$ out of E_j cannot belong to the packing, and then, since E_j is itself a packing of G^a determining the facet $\mu x \leq 1$,

$$\left(\sum_{\ell=1}^{k}\mu_\ell - 1\right)\sum_{j=1}^{n}\pi_j x_j + \sum_{j=1}^{k}\mu_j x_{n+j} \leq \sum_{\ell=1}^{k}\mu_\ell - 1 + 1.$$

Therefore the inequality holds for all the packings of G^x. □

Example. Consider the graph G^x of Figure 5.4. The subgraph G induced by the set of nodes $\{v_1,\ldots,v_8\}$, the facet of $P(G)$ $\sum_{j=1}^{6} x_j + 2x_7 + 2x_8 \leq 2$, and the five disjoint packings $P_1 = \{v_8\}$, $P_2 = \{v_7\}$, $P_3 = \{v_1, v_6\}$, $P_4 = \{v_2, v_4\}$, $P_5 = \{v_3, v_5\}$ have been chosen. (Notice that P_i, $i = 1,\ldots,5$, are independent and satisfy the inequality exactly.) Moreover, a cycle G^a of length 5, the facet of $P(G^a)$ $\sum_{j=1}^{5} x_j \leq 2$, and the associated maximal packings $E_1 = \{w_1, w_3\}$, $E_2 = \{w_1, w_4\}$, $E_3 = \{w_2, w_4\}$, $E_4 = \{w_3, w_5\}$, $E_5 = \{w_2, w_5\}$ have been employed to obtain G^x. For instance, P_3 consists of nodes v_1 and v_3; since E_3 contains nodes w_2 and w_4, nodes v_1 and v_3 in G^x are connected to all the nodes $v_9,\ldots,v_{13}$ except v_{10} and v_{12}. Theorem 5.4 gives the (cooked) facet of $P(G^x)$

$$\sum_{j=1}^{6} 3x_j + 6x_7 + 6x_8 + \sum_{j=9}^{13} 2x_j + 4x_{14} \leq 10.$$

Acknowledgment. We thank two anonymous referees for their help in the presentation of the article and for pointing out Remark 3.

REFERENCES

[1] E. Balas and E. Zemel, *Critical cutsets of graphs and canonical facets of set-packing polytopes*, Math. Oper. Res., 2 (1977), pp. 15–19.

[2] F. Barahona and A. R. Mahjoub,, *Compositions of graphs and polyhedra* II: *Stable sets*, SIAM J. Discrete Math., 7 (1994), pp. 359–371.

[3] F. Barahona and A. R. Mahjoub, *Compositions of graphs and polyhedra* III: *Graphs with no W_4 minor*, SIAM J. Discrete Math. 7 (1994), pp. 372–389.

[4] E. Cheng and W. H. Cunningham, *Wheel inequalities for stable set polytopes*, Math. Programming, 77 (1997), pp. 389–421.

[5] D. C. Cho, M. W. Padberg, and M. R. Rao, *On the uncapacitated plant location problem* II: *Facets and lifting theorems*, Math. Oper. Res., 8 (1983), pp. 590–612.

[6] V. Chvátal, *On certain polytopes associated with graphs*, J. Combin. Theory Ser. B, 18 (1975), pp. 138–154.

[7] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*, John Wiley & Sons, New York, 1998.

[8] G. Dahl, *Stable set polytopes for a class of circulant graphs*, SIAM J. Optim., 9 (1999), pp. 493–503.

[9] A. Galluccio and A. Sassano, *The rank facets of the stable set polytope for claw-free graphs*, J. Combin. Theory Ser. B, 69 (1997), pp. 1–38.

[10] R. Giles and L. E. Trotter, Jr., *On stable set polyhedra for $K_{1,3}$-free graphs*, J. Combin. Theory Ser. B, 31 (1981), pp. 313–326.

[11] G. L. Nemhauser and L. E. Trotter, Jr., *Properties of vertex packing and independence system polyhedra*, Math. Programming, 6 (1974), pp. 48–61.

[12] M. W. Padberg, *On the facial structure of set packing polyhedra*, Math. Programming, 5 (1973), pp. 199–215.

[13] M. W. Padberg, *A note on zero-one programming*, Oper. Res., 23 (1975), pp. 833–837.

[14] M. W. Padberg, *On the complexity of set packing polyhedra*, Ann. Discrete Math., 1 (1977), pp. 421–434.

[15] E. C. Sewell and L. E. Trotter, Jr., *Stability critical graphs and rank facets of the stable set polytope*, Discrete Math., 147 (1995), pp. 247–255.

[16] L. E. Trotter, *A class of facet producing graphs for vertex packing polyhedra*, Discrete Math., 12 (1975), pp. 373–388.

[17] L. A. Wolsey, *Further facet generating procedures for vertex packing polytopes*, Math. Programming, 11 (1976), pp. 158–163.

SIAM J. DISCRETE MATH.
Vol. 16, No. 1, pp. 156–171

IMPROVED INCLUSION-EXCLUSION IDENTITIES AND BONFERRONI INEQUALITIES WITH RELIABILITY APPLICATIONS[*]

KLAUS DOHMEN[†]

Abstract. This paper establishes a connection between the theory of convex geometries, the principle of inclusion-exclusion, and the topological concept of an abstract tube. In particular, it is shown that convex geometries give rise to improved inclusion-exclusion identities and improved Bonferroni inequalities. In this way, several known results from the literature are rediscovered in a concise and unified way. The results are applied in identifying a new class of hypergraphs for which the reliability covering problem can be solved in polynomial time.

Key words. convex geometry, inclusion-exclusion, Bonferroni inequalities, sieve formula, abstract tube, abstract simplicial complex, contractible, system reliability, consecutive k-out-of-n system, reliability covering problem

AMS subject classifications. 05A19, 52A01, 60C05, 60E15, 62N05, 90B25

PII. S0895480101392630

1. Introduction.

Undoubtedly, one of the most important tools in combinatorial probability theory and reliability theory is the principle of inclusion-exclusion and the associated Bonferroni inequalities; see Galambos and Simonelli [10] for a detailed account. For any finite family of sets $\{A_v\}_{v\in V}$ the classical principle of inclusion-exclusion states that

$$\chi\left(\bigcup_{v\in V} A_v\right) = \sum_{\substack{J\subseteq V\\ J\neq\emptyset}} (-1)^{|J|-1}\chi\left(\bigcap_{j\in J} A_j\right) \tag{1.1}$$

and the classical Bonferroni inequalities are

$$\chi\left(\bigcup_{v\in V} A_v\right) \geq \sum_{\substack{J\subseteq V, J\neq\emptyset\\ |J|\leq r}} (-1)^{|J|-1}\chi\left(\bigcap_{j\in J} A_j\right) \quad (r \text{ even}), \tag{1.2}$$

$$\chi\left(\bigcup_{v\in V} A_v\right) \leq \sum_{\substack{J\subseteq V, J\neq\emptyset\\ |J|\leq r}} (-1)^{|J|-1}\chi\left(\bigcap_{j\in J} A_j\right) \quad (r \text{ odd}), \tag{1.3}$$

where $\chi(A)$ is used to denote the indicator function of A; that is, $\chi(A)(\omega) = 1$ if $\omega \in A$, and $\chi(A)(\omega) = 0$ if $\omega \notin A$. There is no real restriction in using indicator functions rather than measures, since (1.1)–(1.3) can be integrated with respect to any measure (e.g., a probability measure) on the algebra generated by $\{A_v\}_{v\in V}$.

Since each sum on the right-hand sides of (1.1)–(1.3) ranges over a large number of terms, it is natural to ask whether fewer terms would give the same or an even better result. Partial answers to this question have been given by several authors,

[*] Received by the editors July 19, 2001; accepted for publication (in revised form) July 22, 2002; published electronically January 3, 2003.
http://www.siam.org/journals/sidma/16-1/39263.html

[†] Hochschule Mittweida, Technikumplatz 17, 09648 Mittweida, Germany (dohmen@htwm.de).

e.g., McKee [15], Naiman and Wynn [16, 17, 18], Narushima [19, 20], and the present author [3, 4, 5]. The problem naturally arises when assessing the reliability of a coherent system [11, 12, 22, 23] or when computing the volume of a union of spherical balls or other geometric objects in Euclidean space [8, 16, 17, 18].

This paper unifies some of the known results in the area by establishing a connection with the theory of convex geometries, which was initiated by Edelman and Jamison [6]. Section 2 reviews the concept of a convex geometry and provides some examples that will be used in later sections. In section 3 we review the topological concept of an abstract tube due to Naiman and Wynn [17] and state our main result, which under certain conditions provides an improved inclusion-exclusion identity and a series of improved Bonferroni inequalities for any finite collection of sets and any convex geometry on the index set of this collection. Then, in section 4 we give an elementary proof of the inclusion-exclusion identity corresponding to our main result in section 3. Sections 5 and 6 provide applications to system reliability analysis, reliability analysis of consecutive k-out-of-n systems, and reliability covering problems.

2. Convex geometries.

For any set V, we use $\mathcal{P}(V)$ to denote the set of subsets of V and $\mathcal{P}^*(V)$ to denote the set of nonempty subsets of V. A *closure operator* on V is a mapping c from $\mathcal{P}(V)$ into itself such that for any subsets X and Y of V,

$$\begin{aligned} &\text{(i)} \quad X \subseteq c(X) \quad \text{(extensionality)},\\ &\text{(ii)} \quad X \subseteq Y \Rightarrow c(X) \subseteq c(Y) \quad \text{(monotonicity)},\\ &\text{(iii)} \quad c(c(X)) = c(X) \quad \text{(idempotence)}. \end{aligned}$$

If c is a closure operator on V, then a subset X of V is referred to as *c-closed* if $c(X) = X$ and as *c-free* if all subsets of X are c-closed. A *c-basis* of X is a minimal subset B of X such that $c(B) = X$. If there are no ambiguities, we simply write *closed* instead of c-closed, *free* instead of c-free, and *basis* instead of c-basis.

A *convex geometry* is a pair (V, c) consisting of a finite set V and a closure operator c on V such that any closed set has a unique basis. For equivalent characterizations of convex geometries, see Edelman and Jamison [6]. Some examples follow.

Throughout, we assume that all graphs (including trees) are finite, undirected, simple, and loop-free. The empty graph is considered as connected.

Example 2.1 (see [6]). Let V be a finite set of points in $\mathbb{R}^d$, and for any subset $X = \{x_1, \ldots, x_n\}$ of V let $\operatorname{conv}(X)$ denote the convex hull of X, that is,

$$\operatorname{conv}(X) := \left\{ \sum_{i=1}^{n} t_i x_i \,\middle|\, t_1, \ldots, t_n \geq 0 \text{ and } \sum_{i=1}^{n} t_i = 1 \right\}.$$

Then, by $c(X) := \operatorname{conv}(X) \cap V$ a closure operator on V is defined. By the Minkowski–Krein–Milman theorem, any c-closed subset X of V has a unique c-basis, consisting of the vertices of the convex polytope $\operatorname{conv}(X)$. Thus, (V, c) is a convex geometry.

Example 2.2 (see [6]). For any tree $G = (V, E)$ and any subset X of V define

$$c(X) := \bigcup_{x,y \in X} \{ z \in V \mid z \text{ is on the unique path between } x \text{ and } y \}.$$

Then, a subset X of V is c-closed if and only if the vertex-induced subgraph $G[X]$ is a subtree of G, and c-free if and only if $X = \{v, w\}$ for some edge $\{v, w\} \in E$ or $X = \{v\}$ for some vertex $v \in V$ or $X = \emptyset$. Since the leaves of $G[X]$ constitute a unique c-basis of any c-closed subset X of V, we conclude that (V, c) is a convex geometry.

Example 2.3 (see [6]). Let $G = (V, E)$ be a connected block graph (i.e., a graph where each maximal 2-connected subgraph is complete), and for any subset X of V let $c(X)$ be the smallest (with respect to inclusion) superset of X that induces a connected subgraph of G. Then, a subset X of V is c-closed if and only if $G[X]$ is connected, and c-free if and only if X is a clique of G, that is, if $G[X]$ is complete. Since the vertices of $G[X]$ whose neighborhood induces a clique of $G[X]$ constitute a unique c-basis of any c-closed subset X of V, it follows that (V, c) is a convex geometry.

Example 2.4 (see [6]). Let V be a finite upper (resp., lower) semilattice, and for any subset X of V let $c(X)$ be the upper (resp., lower) subsemilattice of V which is generated by X (with respect to the join, resp., meet operation). Then, the c-closed subsets of V are the upper (resp., lower) subsemilattices of V, while the c-free subsets of V are the chains of V. Since any c-closed subset X of V has a unique c-basis, namely the set of its join-irreducibles (resp., meet-irreducibles), we are again faced with a convex geometry (V, c).

For any closure operator on a finite set, the following proposition, which will be used in the next two sections, characterizes the free sets by means of their bases.

Proposition 2.5. *Let V be a finite set, and let c be a closure operator on V. Then, any subset J of V is free if and only if it is a basis of itself.*

Proof. Trivially, if J is free, then J is a basis of itself. Subsequently, the opposite direction is proved by contraposition. Assume that J is not free; that is, $K \subset J$ for some nonclosed set K. If J is not closed, then it is not a basis of itself, and we are done. Thus, assume that J is closed. For each $k \in c(K) \setminus K$ we find that $k \in c(K) = c(K \setminus \{k\}) \subseteq c(J) = J$ and hence $J = c(J \setminus \{k\} \cup \{k\}) \subseteq c(c(J \setminus \{k\}) \cup \{k\}) = c(c(J \setminus \{k\})) = c(J \setminus \{k\}) \subseteq c(J) = J$. Therefore, $k \in J$ and $c(J \setminus \{k\}) = J$, whence J is not a basis of itself. □

3. Improved Bonferroni inequalities. The results of this section require some basic knowledge of combinatorial topology. For details, the reader is referred to Rotman [21].

An *abstract simplicial complex* $\mathcal{S}$ is a set of nonempty subsets of some finite set V such that $I \in \mathcal{S}$ and $\emptyset \neq J \subset I$ imply $J \in \mathcal{S}$. The elements of $\mathcal{S}$ are the *faces* or *simplices* of $\mathcal{S}$, whereas the elements of $\mathrm{Vert}(\mathcal{S}) := \bigcup_{I \in \mathcal{S}} I$ are the *vertices* of $\mathcal{S}$. The *dimension* of a face I, $\dim I$, is one less than its cardinality. The *dimension* of $\mathcal{S}$, $\dim \mathcal{S}$, is the maximum dimension of a face in $\mathcal{S}$. A *geometric realization* of $\mathcal{S}$ is any topological space homeomorphic to $\bigcup_{I \in \mathcal{S}} \mathrm{conv}\big(\{\mathbf{e}_{\pi i} \,|\, i \in I\}\big)$, where $\pi : \mathrm{Vert}(\mathcal{S}) \to \{1, \ldots, n\}$ is an injective mapping for some given n and $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ is the standard basis of $\mathbb{R}^n$. $\mathcal{S}$ is called *contractible* if it has a contractible geometric realization.

Example 3.1. Figure 1 shows a realization of the abstract simplicial complex

$$\begin{aligned} \mathcal{S} \;=\; &\{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\},\\ &\qquad\qquad\qquad\qquad \{3, 4\}, \{4, 5\}, \{4, 6\}, \{5, 6\}\}\,. \end{aligned}$$

Obviously, this complex is not contractible because of the unshaded hole on the right-hand side. However, if we fill in the hole (that is, if we attach the triangle $\{4, 5, 6\}$ to the complex), then a contractible abstract simplicial complex would result.

Following Naiman and Wynn [17], an *abstract tube* is a pair $(\mathcal{A}, \mathcal{S})$ consisting of a finite collection of sets $\mathcal{A} = \{A_v\}_{v \in V}$ and an abstract simplicial complex $\mathcal{S} \subseteq \mathcal{P}^*(V)$

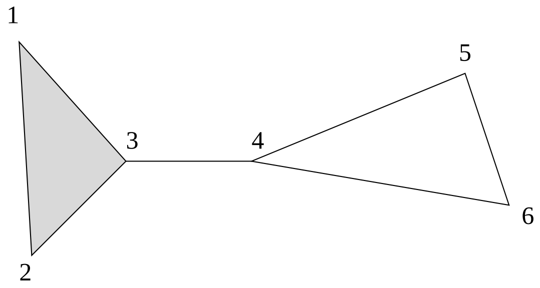


FIG. 1. *A geometric realization of an abstract simplicial complex.*

such that for any $\omega \in \bigcup_{v\in V} A_v$ the abstract simplicial complex

$$\mathcal{S}(\omega) \ := \ \left\{ I \in \mathcal{S} \,\middle|\, \omega \in \bigcap_{i\in I} A_i \right\}$$

is contractible. Given two abstract tubes $(\mathcal{A}_1, \mathcal{S}_1)$ and $(\mathcal{A}_2, \mathcal{S}_2)$, we say that $(\mathcal{A}_1, \mathcal{S}_1)$ is a *subtube* of $(\mathcal{A}_2, \mathcal{S}_2)$ if $\mathcal{A}_1 = \mathcal{A}_2$ and $\mathcal{S}_1 \subseteq \mathcal{S}_2$.

In the following, we restate the main results of abstract tube theory due to Naiman and Wynn [17] without proof.

PROPOSITION 3.2 (see [17]). *Let $(\{A_v\}_{v\in V}, \mathcal{S})$ be an abstract tube. Then, for $r \in \mathbb{N}$,*

$$\chi\left(\bigcup_{v\in V} A_v\right) \geq \sum_{\substack{I\in\mathcal{S}\\|I|\leq r}} (-1)^{|I|-1}\chi\left(\bigcap_{i\in I} A_i\right) \qquad (r\ \textit{even}),$$

$$\chi\left(\bigcup_{v\in V} A_v\right) \leq \sum_{\substack{I\in\mathcal{S}\\|I|\leq r}} (-1)^{|I|-1}\chi\left(\bigcap_{i\in I} A_i\right) \qquad (r\ \textit{odd}).$$

PROPOSITION 3.3 (see [17]). *Let $(\{A_v\}_{v\in V}, \mathcal{S})$ and $(\{A_v\}_{v\in V}, \mathcal{S}')$ be abstract tubes, where $(\{A_v\}_{v\in V}, \mathcal{S}')$ is a subtube of $(\{A_v\}_{v\in V}, \mathcal{S})$. Then, for any $r \in \mathbb{N}$,*

$$\sum_{\substack{I\in\mathcal{S}'\\|I|\leq r}} (-1)^{|I|-1}\chi\left(\bigcap_{i\in I} A_i\right) \geq \sum_{\substack{I\in\mathcal{S}\\|I|\leq r}} (-1)^{|I|-1}\chi\left(\bigcap_{i\in I} A_i\right) \qquad (r\ \textit{even}),$$

$$\sum_{\substack{I\in\mathcal{S}'\\|I|\leq r}} (-1)^{|I|-1}\chi\left(\bigcap_{i\in I} A_i\right) \leq \sum_{\substack{I\in\mathcal{S}\\|I|\leq r}} (-1)^{|I|-1}\chi\left(\bigcap_{i\in I} A_i\right) \qquad (r\ \textit{odd}).$$

Remarks. Since $(\{A_v\}_{v\in V}, \mathcal{P}^*(V))$ is an abstract tube for any finite collection of sets $\{A_v\}_{v\in V}$, the classical Bonferroni inequalities are a particular case of Proposition 3.2. Moreover, since any tube $(\{A_v\}_{v\in V}, \mathcal{S})$ is a subtube of $(\{A_v\}_{v\in V}, \mathcal{P}^*(V))$, Proposition 3.3 especially states that the bounds provided by Proposition 3.2 are at least as sharp as their classical counterparts, although less computational effort is needed to compute them. We further remark that the inequalities in Proposition 3.2 become an identity if $r \geq \dim \mathcal{S} + 1$. In particular, any abstract tube $(\{A_v\}_{v\in V}, \mathcal{S})$ gives rise to an improved inclusion-exclusion identity for the indicator function of $\bigcup_{v\in V} A_v$ which does not require intersections of more than $\dim \mathcal{S} + 1$ sets; that is, the most complicated intersection is $(\dim \mathcal{S}+1)$-fold. Thus, in the terminology of Naiman and Wynn [17], any abstract tube $(\mathcal{A}, \mathcal{S})$ gives rise to an inclusion-exclusion identity of depth $\dim \mathcal{S} + 1$.

Due to Naiman and Wynn [17], the definition of an abstract tube can be weakened by requiring contractibility of $\mathcal{S}(\omega)$ for almost every ω with respect to some dominating measure μ on the ambient space. In this case, the improved Bonferroni inequalities of Propositions 3.2 and 3.3 (and the associated inclusion-exclusion identities) hold almost everywhere with respect to μ, and the pair $(\mathcal{A}, \mathcal{S})$ is referred to as a *weak abstract tube*. If μ is a probability measure, then the mapping $\omega \mapsto \mathcal{S}(\omega)$ may be considered as a random abstract simplicial complex which is required to be almost surely contractible.

We now state our main result. Recall from the above that any abstract tube gives rise to an improved inclusion-exclusion identity and a series of improved Bonferroni inequalities. In the following, we do not mention these identities and inequalities explicitly, since they can easily be read from Proposition 3.2.

For any convex geometry (V, c), we use $\mathrm{Free}(V, c)$ to denote the set of all nonempty c-free subsets of V. Obviously, $\mathrm{Free}(V, c)$ is an abstract simplicial complex.

THEOREM 3.4. *Let (V, c) be a convex geometry, and let $\{A_v\}_{v\in V}$ be a finite family of sets such that for any nonempty and nonclosed subset X of V,*

$$\bigcap_{x\in X} A_x \subseteq A_v \quad \text{for some } v \notin X. \tag{3.1}$$

Then, $(\{A_v\}_{v\in V}, \mathrm{Free}(V, c))$ is an abstract tube.

The proof of Theorem 3.4 is based on the following observation of Björner and Ziegler [2, Exercise 8.23c]. For a rigorous proof of this observation the reader is referred to the more recent paper of Edelman and Reiner [7].

PROPOSITION 3.5 (see [2]). *$\mathrm{Free}(V, c)$ is contractible for any convex geometry (V, c).*

Proof of Theorem 3.4. Let $\omega \in \bigcup_{v\in V} A_v$, $V_\omega := \{v \in V \mid \omega \in A_v\}$ and $c_\omega(I) := c(I)$ for any $I \subseteq V_\omega$. By the definition of V_ω and the requirements of the theorem, V_ω is c-closed. Thus, (V_ω, c_ω) is a convex geometry. Since, moreover, $\mathrm{Free}(V, c)(\omega) = \mathrm{Free}(V_\omega, c_\omega)$, the contractibility of $\mathrm{Free}(V, c)(\omega)$ follows from Proposition 3.5. □

Remarks. Note that by setting $c(X) := X$ for any subset X of V, the abstract tube of Theorem 3.4 specializes to the trivial tube $(\{A_v\}_{v\in V}, \mathcal{P}^*(V))$. In this case, the associated Bonferroni inequalities coincide with the classical Bonferroni inequalities.

We further remark that the depth of the abstract tube of Theorem 3.4 is equal to $h(c) := \max\{|J| : J\ c\text{-free}\}$. As shown by Jamison-Waldner [14], $h(c)$ is the Helly number of the family of all c-closed subsets of V, that is, the smallest integer h such that any family of c-closed subsets of V whose intersection is empty has a subfamily of h or less sets whose intersection is also empty.

Note that condition (3.1) can be replaced by the more general condition

$$\bigcap_{x\in X} A_x \subseteq \bigcup_{v\notin X} A_v\,.$$

However, in all known applications the stronger condition (3.1) applies.

In view of the remarks following Proposition 3.3, it is equally easy to prove that $(\{A_v\}_{v\in V}, \mathrm{Free}(V, c))$ is a weak abstract tube with respect to any probability measure

μ on the algebra generated by $\{A_v\}_{v\in V}$ such that

$$\mu\left(\bigcap_{x\in X} A_x\right) > 0 \quad\text{and}\quad \mu\left(\bigcup_{v\notin X} A_v \,\middle|\, \bigcap_{x\in X} A_x\right) = 1$$

for any nonempty and nonclosed subset X of V.

We further remark that the abstract tube $(\{A_v\}_{v\in V}, \mathrm{Free}(V,c'))$ is a subtube of $(\{A_v\}_{v\in V}, \mathrm{Free}(V,c))$ if both c and c' satisfy the requirements of Theorem 3.4 and $c' \leq c$, where the partial ordering relation $\leq$ is defined by

$$c' \leq c \quad :\Leftrightarrow \quad c(I) \subseteq c'(I) \text{ for any subset } I \text{ of } V$$

or, equivalently,

$$c' \leq c \quad :\Leftrightarrow \quad \text{all } c'\text{-closed subsets of } V \text{ are } c\text{-closed.}$$

By this and Proposition 3.3, it follows that the improved Bonferroni inequalities associated with c' are at least as sharp as those associated with c if $c' \leq c$. In particular, since the closure operator $I \mapsto I$ on V is largest with respect to $\leq$, the new Bonferroni inequalities are at least as sharp as their classical counterparts.

From Theorem 3.4 we now deduce some particular results, which for the first time appear in a common context. As a first consequence of Theorem 3.4 we deduce the following.

Corollary 3.6 (see [4]). *Let $\{A_v\}_{v\in V}$ be a finite family of sets, where V is endowed with a linear ordering relation, and let $\mathcal{X} \subseteq \mathcal{P}^*(V)$ such that for any $X \in \mathcal{X}$,*

$$\bigcap_{x\in X} A_x \;\subseteq\; A_v \quad \textit{for some } v > \max X.$$

Then, $\big(\{A_v\}_{v\in V}, \{I \in \mathcal{P}^*(V) \,|\, I \not\supseteq X \text{ for all } X \in \mathcal{X}\}\big)$ is an abstract tube.

Proof. We apply Theorem 3.4. By the requirements of the corollary there is a family $\{v_X\}_{X\in\mathcal{X}} \subseteq V$ such that for any $X \in \mathcal{X}$, $\bigcap_{x\in X} A_x \subseteq A_{v_X}$ for some $v_X > \max X$. Now, for any subset I of V define $c(I) := I \cup \{v_X \,|\, X \in \mathcal{X},\, X \subseteq I\}$ as well as

$$c^*(I) \;:=\; c(I) \cup c(c(I)) \cup c(c(c(I))) \cup c(c(c(c(I)))) \cup \dots .$$

Then, c^* is a closure operator on V, where any c^*-closed subset I of V has a unique c^*-basis, namely $I \setminus \{v_X \,|\, X \in \mathcal{X},\, X \subseteq I\}$. Thus, we find that (V, c^*) is a convex geometry, where a subset I of V is c^*-free if and only if $I \not\supseteq X$ for any $X \in \mathcal{X}$. □

Remarks. Notice that Corollary 3.6 can be dualized by replacing $v > \max X$ with $v < \min X$ and that it yields the trivial abstract tube $(\{A_v\}_{v\in V}, \mathcal{P}^*(V))$ if $\mathcal{X} = \emptyset$.

As already noted in [4], the identity associated with Corollary 3.6 generalizes Whitney's broken circuit theorem [24] on the chromatic polynomial of a graph.

From Corollary 3.6 we now deduce an abstract tube generalization of Narushima's inclusion-exclusion identity [20]. For any partially ordered set V we use $\mathcal{C}(V)$ to denote the *order complex* of V, which consists of all nonempty chains of V.

Corollary 3.7 (see [4]). *Let $\{A_v\}_{v\in V}$ be a finite family of sets, where V is endowed with a partial ordering relation such that for any $x, y \in V$, $A_x \cap A_y \subseteq A_z$ for some upper bound z of x and y. Then, $(\{A_v\}_{v\in V}, \mathcal{C}(V))$ is an abstract tube.*

Proof. Corollary 3.7 follows from Corollary 3.6 by defining $\mathcal{X}$ as the set of all unordered pairs of incomparable elements of V and then considering an arbitrary linear extension of the partial ordering relation on V. □

Remarks. The requirements of Corollary 3.7 are weaker than the requirements by Narushima [20]. Namely, Narushima [20] requires that for any $x, y \in V$, $A_x \cap A_y \subseteq A_z$ for some *minimal* upper bound z of x and y. In Corollary 3.7, however, the minimality of z is not required. Evidently, the requirements of Corollary 3.7 are satisfied if V is a finite upper semilattice and $A_x \cap A_y \subseteq A_{x \vee y}$ for any $x, y \in V$. This particular case can also be deduced by applying Theorem 3.4 in connection with Example 2.4.

Corollary 3.7 specializes to the classical inclusion-exclusion identity if the partial ordering relation on V is linear or, in other words, if V is a chain. In the extreme case where V has a maximum $\hat{1}$ and any distinct $x, y < \hat{1}$ are incomparable and satisfy $A_x \cap A_y \subseteq A_{\hat{1}}$, Corollary 3.7 requires evaluation of only $2\,|V| - 1$ terms, whereas the traditional inclusion-exclusion principle would require evaluation of $2^{|V|} - 1$ terms.

For our next corollary, we must recall some terminology from graph theory.

Let G be a graph. By definition, a *chord* of a path P of G is an edge of G joining two vertices that are not adjacent in P and similarly for cycles. A graph G is called a *chordal graph* if any cycle of length greater than three has a chord. A *clique* of G is a subset X of the vertex-set of G such that the vertex-induced subgraph $G[X]$ is complete. The *clique complex* of G consists of all nonempty cliques of G.

Corollary 3.8. *Let $\{A_v\}_{v \in V}$ be a finite family of sets, and let $G = (V, E)$ be a connected chordal graph such that $A_x \cap A_y \subseteq A_z$ for any $x, y \in V$ and any z on any chordless path between x and y in G. Then, $\{A_v\}_{v \in V}$ and the clique complex of G constitute an abstract tube.*

Proof. We apply Theorem 3.4. For any subset X of V define

$$c(X) := \bigcup_{x, y \in X} \{z \in V \mid z \text{ is on a chordless path between } x \text{ and } y\} .$$

Then, (V, c) is a convex geometry, where a subset X of V is free if and only if X is a clique of G [6, 9]. Now, Theorem 3.4 immediately gives the result. □

Remarks. Note that Corollary 3.8 yields the trivial abstract tube $(\{A_v\}_{v \in V}, \mathcal{P}^*(V))$ if G is complete, since all subsets of the vertex-set are cliques in this case.

From Corollary 3.8 we now deduce the tree sieve of Naiman and Wynn [16].

Corollary 3.9 (see [16]). *Let $\{A_v\}_{v \in V}$ be a finite family of sets, where the indices form the vertices of a tree $G = (V, E)$ such that $A_x \cap A_y \subseteq A_z$ for any $x, y \in V$ and any z on the unique path between x and y in G. Then, $\{A_v\}_{v \in V}$ and the tree (considered as an abstract simplicial complex) constitute an abstract tube.*

Proof. Since trees are chordal, Corollary 3.9 follows from Corollary 3.8. □

We close this section with a further corollary.

Corollary 3.10. *Let $\{A_v\}_{v \in V}$ be a finite family of sets such that for any nonempty subset X of V there is a unique minimal nonempty subset Y of X such that $\bigcap_{x \in X} A_x = \bigcap_{y \in Y} A_y$. Then, $\{A_v\}_{v \in V}$ and the abstract simplicial complex consisting of all nonempty subsets I of V such that $\bigcap_{i \in I} A_i \neq \bigcap_{j \in J} A_j$ for all nonempty proper subsets and supersets J of I constitute an abstract tube.*

Proof. Again, we apply Theorem 3.4. It is straightforward to check that

$$c(X) := \left\{ v \in V \;\middle|\; \bigcap_{x \in X} A_x \subseteq A_v \right\} \quad (X \neq \emptyset)\,; \qquad c(\emptyset) := \emptyset$$

defines a closure operator on V. In order to check the unique basis property, let X be a nonempty closed subset of V and Y the unique minimal nonempty subset of X

such that $\bigcap_{x\in X} A_x = \bigcap_{y\in Y} A_y$, which exists by the requirements. Then,

$$X \subseteq \left\{ v\in V \;\middle|\; \bigcap_{x\in X} A_x \subseteq A_v \right\} = \left\{ v\in V \;\middle|\; \bigcap_{y\in Y} A_y \subseteq A_v \right\} = c(Y) \subseteq X$$

and hence $c(Y) = X$. Now, to show that Y is minimal with respect to $c(Y) = X$, suppose that $c(Y') = X$ for some nonempty subset Y' of X. Then, $\bigcap_{x\in X} A_x = \bigcap_{y\in Y'} A_y$ and hence $Y' \supseteq Y$ by the choice of Y. Thus, Y is the unique basis of X. The description of the free sets immediately follows from Proposition 2.5. □

4. Improved inclusion-exclusion identities. In this section, we give an elementary proof of the inclusion-exclusion identity associated with the abstract tube of Theorem 3.4. More precisely, we prove the following theorem, which generalizes and improves the classical inclusion-exclusion identity.

THEOREM 4.1. *Let (V, c) be a convex geometry, and let $\{A_v\}_{v\in V}$ be a finite family of sets such that for any nonempty and nonclosed subset X of V,*

$$\bigcap_{x\in X} A_x \subseteq A_v \quad \textit{for some } v \notin X.$$

Then,

$$\chi\left(\bigcup_{v\in V} A_v\right) = \sum_{\substack{J\in\mathcal{P}^*(V)\\ J \textit{ free}}} (-1)^{|J|-1}\,\chi\left(\bigcap_{j\in J} A_j\right).$$

The proof of Theorem 4.1 is facilitated by several propositions. The first one is due to Edelman and Jamison [6]. Although not mentioned by these authors, their result strongly generalizes a useful result of Narushima [19] on semilattices.

PROPOSITION 4.2 (see [6]). *For any closed set J in a convex geometry (V, c),*

$$\sum_{\substack{I\subseteq J\\ c(I)=J}} (-1)^{|I|} = \begin{cases} (-1)^{|J|} & \textit{if } J \textit{ is free},\\ 0 & \textit{otherwise.}\end{cases}$$

Subsequently, we give our own proof of Proposition 4.2. It generalizes Narushima's proof [19] for the semilattice case (Example 2.4).

Proof. Let J_0 be the unique basis of J. Then, $c(I) = J$ if and only if $J_0 \subseteq I \subseteq J$. Hence,

$$\sum_{\substack{I\subseteq J\\ c(I)=J}} (-1)^{|I|} = \begin{cases} (-1)^{|J|} & \text{if } J_0 = J,\\ 0 & \text{otherwise.}\end{cases}$$

From Proposition 2.5 it follows that $J_0 = J$ if and only if J is free. □

From the following proposition we derive two corollaries which are not needed for proving Theorem 4.1 but which are interesting in their own right.

PROPOSITION 4.3. *Let (V, c) be a convex geometry. Furthermore, let g be a mapping from the power set of V into an abelian group such that $g = g \circ c$. Then,*

$$\sum_{I\subseteq V} (-1)^{|I|}\, g(I) = \sum_{\substack{J\subseteq V\\ J \textit{ free}}} (-1)^{|J|}\, g(J)\,.$$

Proof. By the requirements, $g(I) = g(c(I))$ for any subset I of V. Therefore,

$$\sum_{I \subseteq V} (-1)^{|I|}\, g(I) \;=\; \sum_{I \subseteq V} (-1)^{|I|}\, g(c(I)) \;=\; \sum_{\substack{J \subseteq V \\ c(J)=J}} \; \sum_{\substack{I \subseteq J \\ c(I)=J}} (-1)^{|I|}\, g(J)\,.$$

Now, by applying Proposition 4.2, the statement immediately follows. □

Corollary 4.4. *The number of free sets in a convex geometry* (V, c) *is equal to*

$$\sum_{I \subseteq V} (-1)^{|c(I) \setminus I|}\,.$$

Proof. For any $I \subseteq V$ define $g(I) := (-1)^{|c(I)|}$ and apply Proposition 4.3. □

Corollary 4.5. *Let* (V, c) *be a convex geometry. Then,*

$$\sum_{I \subseteq V} (-1)^{|I|}\, |c(I)| \;=\; \sum_{\substack{J \subseteq V \\ J \text{ free}}} (-1)^{|J|}\, |J|\,. \tag{4.1}$$

Proof. For any $I \subseteq V$ define $g(I) := |c(I)|$ and apply Proposition 4.3. □

Remark. For the convex geometry of Example 2.1, where c is derived from the convex hull operator in $\mathbb{R}^d$, Corollary 4.5 specializes to a result of Gordon [13]. A recent result of Edelman and Reiner [7] states that either side of (4.1) agrees in absolute value with the number of points in V which are in the interior of the convex hull of V.

We continue with a further proposition.

Proposition 4.6. *Let* $\{A_v\}_{v \in V}$ *be a finite family of sets, and let* c *be a closure operator on* V *such that for any nonempty and nonclosed subset* X *of* V,

$$\bigcap_{x \in X} A_x \;\subseteq\; A_v \quad \textit{for some } v \notin X\,. \tag{4.2}$$

Then, for any nonempty subset I *of* V,

$$\bigcap_{i \in I} A_i \;=\; \bigcap_{i \in c(I)} A_i\,.$$

Proof. Fix $I \subseteq V$, $I \neq \emptyset$. If $\bigcap_{i \in I} A_i = \emptyset$, then since $c(I) \supseteq I$, $\bigcap_{i \in c(I)} A_i = \emptyset$, and we are done. Otherwise choose $\omega \in \bigcap_{i \in I} A_i$ and show that $\omega \in \bigcap_{i \in c(I)} A_i$. By the choice of ω, $I \subseteq V_\omega$, where $V_\omega := \{v \in V \,|\, \omega \in A_v\}$. By the definition of V_ω and (4.2), V_ω is closed and hence $c(I) \subseteq V_\omega$. Thus, $\omega \in \bigcap_{i \in c(I)} A_i$ and the proof is complete. □

We are now ready to prove Theorem 4.1.

Proof of Theorem 4.1. By the classical inclusion-exclusion principle we have

$$\chi\left(\bigcup_{v \in V} A_v\right) \;=\; \sum_{I \in \mathcal{P}^*(V)} (-1)^{|I|-1} g(I), \quad \text{where} \quad g(I) \;:=\; \chi\left(\bigcap_{i \in I} A_i\right).$$

By Proposition 4.6, $g = g \circ c$. Thus, the result follows from Proposition 4.3. □

5. Application to system reliability analysis. In this section, we describe some consequences of our results to system reliability analysis. Applications to network reliability analysis can be found in [4].

A *coherent binary system* is a couple $\Sigma = (E, \phi)$ consisting of a finite set E and a function ϕ from the power set of E into $\{0; 1\}$ such that $\phi(\emptyset) = 0$, $\phi(E) = 1$ and $\phi(X) \leq \phi(Y)$ for any $X, Y \subseteq E$ with $X \subseteq Y$. E and ϕ are, respectively, called the *component set* and the *structure function* of Σ.

At any instant of time, each component e of Σ assumes randomly and independently one of two states, *operating* or *failing*, with probabilities p_e and $q_e = 1 - p_e$, respectively. Σ is said to be *operating* (resp., *failing*) if ϕ applied to the set of operating components, which is also referred to as the *state* of Σ, gives 1 (resp., 0). The *reliability* of Σ is the probability that Σ is operating. Since this quantity is determined by Σ and the vector of operation probabilities $\mathbf{p} = (p_e)_{e \in E}$, it is abbreviated to $\mathrm{Rel}_\Sigma(\mathbf{p})$.

At this point let us consider a particular kind of coherent binary system. Let $k, n \in \mathbb{N}$, $1 \leq k \leq n$. A *consecutive k-out-of-n success (resp., failure) system* is a coherent binary system $\Sigma = (E, \phi)$, where E is a linearly ordered finite set of size n and where for any subset X of E, $\phi(X) = 1$ (resp., $\phi(E \setminus X) = 0$) if and only if X contains k consecutive elements of E. In other words, the system operates (resp., fails) whenever k or more consecutive components operate (resp., fail).

Consecutive k-out-of-n failure systems serve as a model for a particular type of communication networks. Consider, for instance, the network in Figure 2, where nodes 1–6 are assumed to fail randomly and independently with known probabilities and all other nodes and edges are assumed to be perfectly reliable. It is immediately clear that in this network a message can pass from s to t if and only if no three consecutive nodes among 1–6 simultaneously fail. Thus, the network is appropriately modeled as a consecutive 3-out-of-6 failure system. The reliability of this system is simply the probability that a message can pass from s to t.

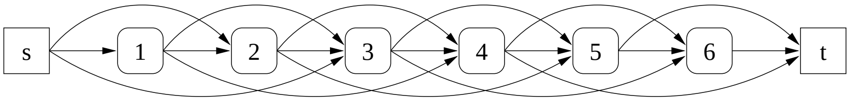


FIG. 2. *A consecutive* 3*-out-of-*6 *failure system.*

A key role in calculating the reliability of a general coherent binary system Σ is played by the minpaths and mincuts of Σ: A *minpath* of Σ is a minimal subset P of E such that $\phi(P) = 1$; that is, $\phi(P) = 1$ and $\phi(Q) = 0$ for any proper subset Q of P. A *mincut* of Σ is a minimal subset C of E such that $\phi(E \setminus C) = 0$; that is, $\phi(E \setminus C) = 0$ and $\phi(E \setminus D) = 1$ for any proper subset D of C.

Note that in the particular case of a consecutive k-out-of-n success (resp., failure) system, X is a minpath (resp., mincut) if and only if X is consecutive and $|X| = k$.

In general, if F is a set of components of a coherent binary system Σ, then F is said to *operate* (resp., *fail*) if all components in F operate (resp., fail). Thus, with $\mathcal{F}$ denoting the set of minpaths (resp., mincuts) of Σ, we have

$$\mathrm{Rel}_\Sigma(\mathbf{p}) = \Pr\left(\bigcup_{F \in \mathcal{F}} \{F \text{ operates}\}\right) \quad \left(\text{resp., } 1 - \mathrm{Rel}_\Sigma(\mathbf{p}) = \Pr\left(\bigcup_{F \in \mathcal{F}} \{F \text{ fails}\}\right)\right), \tag{5.1}$$

where Pr denotes the induced probability measure on the set of system states.

In connection with Proposition 3.2, the first part of the following theorem yields improved inclusion-exclusion identities and Bonferroni inequalities for the right-hand sides of (5.1) and thus for $\mathrm{Rel}_\Sigma(\mathbf{p})$. We do not mention the identities explicitly, since they are an immediate consequence of the corresponding inequalities.

THEOREM 5.1. *Let $\Sigma = (E, \phi)$ be a coherent binary system, whose set of minpaths (resp., mincuts) $\mathcal{F}$ is endowed with a closure operator c such that $(\mathcal{F}, c)$ is a convex geometry and such that $Y \subseteq \bigcup \mathcal{X}$ for any nonempty $\mathcal{X} \subseteq \mathcal{F}$ and any $Y \in c(\mathcal{X})$. Then,*

$$\Big(\big\{\{F \textit{ operates}\}\big\}_{F\in\mathcal{F}}, \mathrm{Free}(\mathcal{F}, c)\Big) \quad \Big(\textit{resp.,} \quad \Big(\big\{\{F \textit{ fails}\}\big\}_{F\in\mathcal{F}}, \mathrm{Free}(\mathcal{F}, c)\Big)\Big)$$

is an abstract tube. In particular, in the case where $\mathcal{F}$ denotes the set of minpaths,

$$\mathrm{Rel}_\Sigma(\mathbf{p}) \;\geq \sum_{\substack{\mathcal{I}\in\mathrm{Free}(\mathcal{F},c)\\ |\mathcal{I}|\leq r}} (-1)^{|\mathcal{I}|-1} \prod_{e\,\in\bigcup\mathcal{I}} p_e \quad \textit{(r even)},$$

$$\mathrm{Rel}_\Sigma(\mathbf{p}) \;\leq \sum_{\substack{\mathcal{I}\in\mathrm{Free}(\mathcal{F},c)\\ |\mathcal{I}|\leq r}} (-1)^{|\mathcal{I}|-1} \prod_{e\,\in\bigcup\mathcal{I}} p_e \quad \textit{(r odd)},$$

and in the case where $\mathcal{F}$ denotes the set of mincuts,

$$1-\mathrm{Rel}_\Sigma(\mathbf{p}) \;\geq \sum_{\substack{\mathcal{I}\in\mathrm{Free}(\mathcal{F},c)\\ |\mathcal{I}|\leq r}} (-1)^{|\mathcal{I}|-1} \prod_{e\,\in\bigcup\mathcal{I}} q_e \quad \textit{(r even)},$$

$$1-\mathrm{Rel}_\Sigma(\mathbf{p}) \;\leq \sum_{\substack{\mathcal{I}\in\mathrm{Free}(\mathcal{F},c)\\ |\mathcal{I}|\leq r}} (-1)^{|\mathcal{I}|-1} \prod_{e\,\in\bigcup\mathcal{I}} q_e \quad \textit{(r odd)}.$$

Proof. The first part follows from Theorem 3.4 with $V := \mathcal{F}$ and $A_F := \{F \text{ operates}\}$ (resp., $A_F := \{F \text{ fails}\}$) for any $F \in \mathcal{F}$. The second part is an immediate consequence of the first part and Proposition 3.2. □

Remark. Note that the inequalities of Theorem 5.1 specialize to the usual Bonferroni inequalities for system reliability if $c(\mathcal{X}) = \mathcal{X}$ for any $\mathcal{X} \subseteq \mathcal{F}$. For the convex geometry of Example 2.4, where $\mathcal{F}$ is a lower (resp., upper) semilattice, the requirements of Theorem 5.1 are equivalent to $X \wedge Y \subseteq X \cup Y$ (resp., $X \vee Y \subseteq X \cup Y$) for any $X, Y \in \mathcal{F}$. Note in this case the free sets are the chains of $\mathcal{F}$. We thus rediscover Shier's chain formula for the reliability of a coherent binary system [22, 23] as well as the corresponding improved Bonferroni inequalities, which are established in [4].

As an illustration of how to obtain such a semilattice structure (and thereby a closure operator), define for any k-subsets X and Y of some linearly ordered set E, $X \leq Y :\Leftrightarrow x \leq y$ for all $x \in X$, $y \in Y \setminus X$. In this case, $X \wedge Y$ consists of the k smallest elements of $X \cup Y$; in particular, $X \wedge Y \subseteq X \cup Y$ as required.

The remainder of this section is devoted to consecutive k-out-of-n systems.

In [22, 23], Shier describes a $O(n^3)$ method based on the disjoint products technique for computing the reliability of a consecutive k-out-of-n system for fixed k. The following theorem provides a $O(n^2)$ method under the requirement that $k \geq n/2$. By a suitable factoring, the expression in this theorem can be evaluated in $O(n)$ steps.

THEOREM 5.2. *Let Σ be a consecutive k-out-of-n success system whose component reliabilities are given by $\mathbf{p} = (p_1, \ldots, p_n)$. If $k \geq n/2$, then*

$$\mathrm{Rel}_\Sigma(\mathbf{p}) \;=\; \sum_{i=1}^{n-k} (1-p_{i+k}) \prod_{j=i}^{i+k-1} p_j \;+ \prod_{j=n-k+1}^{n} p_j\,.$$

Proof. For $i = 1, \ldots, n-k+1$ let A_i be the event that components $i, \ldots, i+k-1$ operate. Then, $\mathrm{Rel}_\Sigma(\mathbf{p}) = \Pr(A_1 \cup \cdots \cup A_{n-k+1})$. Since $k \geq n/2$, $A_x \cap A_y \subseteq A_z$ for $x, y = 1, \ldots, n-k+1$ and any z between x and y. Thus, by combining Theorem 4.1 with the convex geometry of Example 2.2 (or by applying Corollary 3.9) we obtain

$$\mathrm{Rel}_\Sigma(\mathbf{p}) \;=\; \sum_{i=1}^{n-k+1} \Pr(A_i) - \sum_{i=1}^{n-k} \Pr(A_i \cap A_{i+1}) \;=\; \sum_{i=1}^{n-k+1} \prod_{j=i}^{i+k-1} p_j - \sum_{i=1}^{n-k} \prod_{j=i}^{i+k} p_j$$

$$= \sum_{i=1}^{n-k} \prod_{j=i}^{i+k-1} p_j - \sum_{i=1}^{n-k} \prod_{j=i}^{i+k} p_j + \prod_{j=n-k+1}^{n} p_j \;=\; \sum_{i=1}^{n-k} \prod_{j=i}^{i+k-1} p_j - \sum_{i=1}^{n-k} p_{i+k} \prod_{j=i}^{i+k-1} p_j + \prod_{j=n-k+1}^{n} p_j\,,$$

which immediately gives the result. □

In the case of equal component reliabilities we even obtain a closed formula.

COROLLARY 5.3. *Let $\Sigma = (E, \phi)$ be a consecutive k-out-of-n success system whose component reliabilities are given by $\mathbf{p} = (p, \ldots, p)$. If $k \geq n/2$, then*

$$\mathrm{Rel}_\Sigma(\mathbf{p}) \;=\; p^k\,[(n-k)(1-p)+1]\,.$$

Proof. Corollary 5.3 is an immediate consequence of Theorem 5.2. □

Remark. Notice that the preceding results can easily be adapted to compute the reliability of a consecutive k-out-of-n failure system. In this way, the reliability of the consecutive 3-out-of-6 failure system in Figure 2 is easily seen to be

$$1 - (1-q_4)q_1q_2q_3 - (1-q_5)q_2q_3q_4 - (1-q_6)q_3q_4q_5 - q_4q_5q_6\,,$$

which equals $1 - 4q^3 + 3q^4$ if all failure probabilities are equal. The reader is invited to obtain the same result using the classical inclusion-exclusion method.

6. Application to reliability covering problems.

Reliability covering problems were introduced by Ball, Provan, and Shier [1] (see also [23]) in order to generalize several types of reliability problems. They serve, e.g., as a model for mass transit systems with reliable stops and unreliable routes. The overall reliability of such a system is the probability that each stop is served by an operating route. Further examples include evaluating the reliability of flight schedules for aircraft [1, 23] and determining the reliability of maintaining continuous surveillance of a critical point of a country's border [23].

Reliability covering problems can be adequately formulated using the terminology of hypergraphs. A *hypergraph* is a couple $H = (V, \mathcal{E})$, where V is a finite set and $\mathcal{E}$ is a set of subsets of V. The elements of V and $\mathcal{E}$ are the *vertices* and *edges* of H, respectively. Thus, in the case of a mass transit system, the vertices correspond to the stops and the edges to the routes of the system. Throughout, we assume that the vertices of the hypergraph are perfectly reliable, whereas the edges are subject to random and independent failure. The edge operation probabilities are given by a vector $\mathbf{p} = (p_E)_{E \in \mathcal{E}} \in [0,1]^{\mathcal{E}}$. A *covering* of V is a subset $\mathcal{X}$ of $\mathcal{E}$ such that $\bigcup \mathcal{X} = V$. Thus, in case of a mass transit system, the coverings correspond to sets of routes such that each stop is served by a route. The general objective is to compute $\mathrm{Cov}(H; \mathbf{p})$, the probability that the vertex-set of H is covered by the operating edges of H. With $\mathcal{E}(v) := \{E \in \mathcal{E} \mid v \in E\}$ $(v \in V)$, this coverage probability can be expressed as

$$\mathrm{Cov}(H; \mathbf{p}) \;=\; 1 \;-\; \Pr\left(\bigcup_{v \in V} \bigcap_{E \in \mathcal{E}(v)} \{E \text{ fails}\} \right). \tag{6.1}$$

The following theorem provides improved inclusion-exclusion identities and improved Bonferroni inequalities for the right-hand side of (6.1) and thus for $\mathrm{Cov}(H;\mathbf{p})$. Again, we do not mention the improved inclusion-exclusion identities explicitly, since they are an immediate consequence of the corresponding improved inequalities.

Theorem 6.1. *Let $H = (V,\mathcal{E})$ be a hypergraph whose edges fail randomly and independently and whose vertex-set V is endowed with a closure operator c such that (V,c) is a convex geometry and such that the complement of each edge is c-closed. Then,*

$$\left(\left\{\bigcap\nolimits_{E\in\mathcal{E}(v)}\{E \textit{ fails}\}\right\}_{v\in V}, \mathrm{Free}(V,c)\right)$$

is an abstract tube. In particular, for any $\mathbf{p} = (p_E)_{E\in\mathcal{E}} \in [0,1]^{\mathcal{E}}$ and any $r\in\mathbb{N}$,

$$\mathrm{Cov}(H;\mathbf{p}) \;\le\; \sum_{\substack{I\subseteq V, |I|\le r\\ I \textit{ is } c\textit{-free}}} (-1)^{|I|} \prod_{\substack{E\in\mathcal{E}\\ E\cap I\neq\emptyset}} q_E \quad (r \textit{ even}),$$

$$\mathrm{Cov}(H;\mathbf{p}) \;\ge\; \sum_{\substack{I\subseteq V, |I|\le r\\ I \textit{ is } c\textit{-free}}} (-1)^{|I|} \prod_{\substack{E\in\mathcal{E}\\ E\cap I\neq\emptyset}} q_E \quad (r \textit{ odd}),$$

where $q_E = 1 - p_E$ for any $E\in\mathcal{E}$.

Proof. We apply Theorem 3.4 with $A_v := \bigcap_{E\in\mathcal{E}(v)}\{E \text{ fails}\}$ for any $v\in V$. Evidently, the requirements of Theorem 3.4 are satisfied if $\bigcap_{x\in X} A_x \subseteq A_v$ for any nonempty subset X of V and any $v\in c(X)$. A sufficient condition for $\bigcap_{x\in X} A_x \subseteq A_v$ is that all edges containing v have a nonempty intersection with X. In order to show that this condition holds, assume that $v\in E$ and $E\cap X = \emptyset$ for some edge E of the hypergraph. Then X would be a subset of the complement $\overline{E}$ of E, and, since all complements of edges are required to be c-closed, we would also have $c(X)\subseteq\overline{E}$ and hence $v\in\overline{E}$, contradicting $v\in E$. Now, the first part of Theorem 6.1 follows from Theorem 3.4. The second part follows from the first part and Proposition 3.2. □

Due to Ball, Provan, and Shier [1] and Shier [23], the reliability covering problem, that is, the problem of computing $\mathrm{Cov}(H;\mathbf{p})$ for given H and $\mathbf{p}$, is $\#P$-hard, even when restricted to the class of hypergraphs whose vertices are the vertices of an undirected tree and whose edges are paths of cardinality 3 in the tree (viewing paths as sets of vertices). A careful reading of the $\#P$-hardness results in [1, 23] reveals that the restricted problem remains $\#P$-hard even if the tree is part of the input. Considering complements of paths instead of paths or, more generally, complements of subtrees instead of subtrees or paths, we obtain the following positive result.

Theorem 6.2. *For hypergraphs whose vertices are the vertices of an undirected tree and whose edges are complements of subtrees of the tree, the coverage probability can be computed in polynomial time from the hypergraph and the tree.*

Proof. Let $G = (V,T)$ be a tree and $H = (V,\mathcal{E})$ be a hypergraph where each edge of H is the complement of a subtree of G. By combining Theorem 6.1 with Example 2.2 (or by applying Corollary 3.9) we are led to the improved inclusion-exclusion identity

$$\mathrm{Cov}(H;\mathbf{p}) \;=\; 1 - \sum_{v\in V}\prod_{E\in\mathcal{E}(v)} q_E + \sum_{\{v,w\}\in T}\;\prod_{E\in\mathcal{E}(v)\cup\mathcal{E}(w)} q_E\,,$$

whose evaluation requires $O(|V|\cdot|\mathcal{E}|)$ time. □

An even more general result is the following. Recall that the *clique number* of a graph G is the maximum cardinality of a clique in G.

THEOREM 6.3. *For hypergraphs whose vertices are those of a connected block graph of bounded clique number and whose edges are complements of connected subgraphs of the connected block graph, the coverage probability can be computed in polynomial time from the hypergraph and the connected block graph.*

Proof. Let G be a connected block graph having clique number at most d, and let $H = (V, \mathcal{E})$ be a hypergraph where each edge of H is the complement of a connected subgraph of G. By applying Theorem 6.1 in connection with the convex geometry of Example 2.3 we obtain the improved inclusion-exclusion formula

$$\mathrm{Cov}(H; \mathbf{p}) \;=\; \sum_{\substack{I \text{ is a clique} \\ \text{of } G}} (-1)^{|I|} \prod_{\substack{E \in \mathcal{E} \\ E \cap I \neq \emptyset}} q_E \,,$$

whose evaluation requires $O(|V|^d \cdot |\mathcal{E}|)$ time, where d is a constant. □

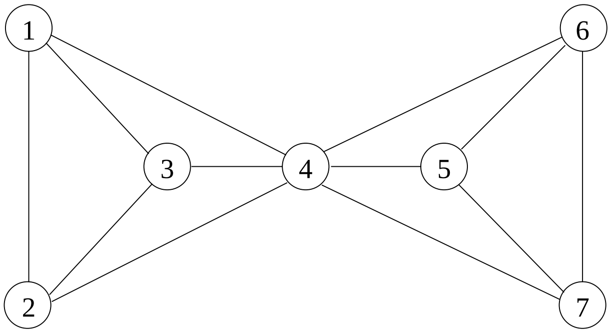


FIG. 3. *A connected block graph.*

TABLE 1
Bonferroni bounds for the coverage probability of the hypergraph in Example 6.4.

	Classical bounds		Improved bounds	
r	$f_r(q)$	# sets	$f_r^*(q)$	# sets
1	$1 - q^2 - q^3 - 3q^4 - 2q^5$	8	$1 - q^2 - q^3 - 3q^4 - 2q^5$	8
2	$1 - q^2 - q^3 - 2q^4 + 3q^5 + 5q^6 + 10q^7$	29	$1 - q^2 - q^3 - 2q^4 + 3q^5 + 3q^6 + 3q^7$	20
3	$1 - q^2 - q^3 - 2q^4 + 3q^5 + q^6 - 5q^7 - 16q^8$	64	$1 - q^2 - q^3 - 2q^4 + 3q^5 + q^6 - 3q^8$	28
4	$1 - q^2 - q^3 - 2q^4 + 3q^5 + q^6 + 14q^8$	99	$1 - q^2 - q^3 - 2q^4 + 3q^5 + q^6 - q^8$	30
5	$1 - q^2 - q^3 - 2q^4 + 3q^5 + q^6 - 7q^8$	120		
6	$1 - q^2 - q^3 - 2q^4 + 3q^5 + q^6$	127		
7	$1 - q^2 - q^3 - 2q^4 + 3q^5 + q^6 - q^8$	128		

TABLE 2
Numerical values of the bounds in Table 1.

q	$f_3(q)$	$f_3^*(q)$	$f_4^*(q)^\dagger$	$f_4(q)$	$f_2^*(q)$	$f_2(q)$
0.0	1.00000	1.00000	*1.00000*	1.00000	1.00000	1.00000
0.1	0.98883	0.98883	*0.98883*	0.98883	0.98883	0.98884
0.2	0.94972	0.94982	*0.94982*	0.94986	0.94999	0.95021
0.3	0.87268	0.87462	*0.87475*	0.87574	0.87693	0.87992
0.4	0.74094	0.75765	*0.75896*	0.76879	0.77272	0.79238
0.5	0.50781	0.59766	*0.60547*	0.66406	0.66406	0.75000
0.6	0.03603	0.39435	*0.42794*	0.67988	0.62203	0.91130
0.7	-1.02548	0.13572	*0.25101*	1.11573	0.79102	1.60280

†exact coverage probability

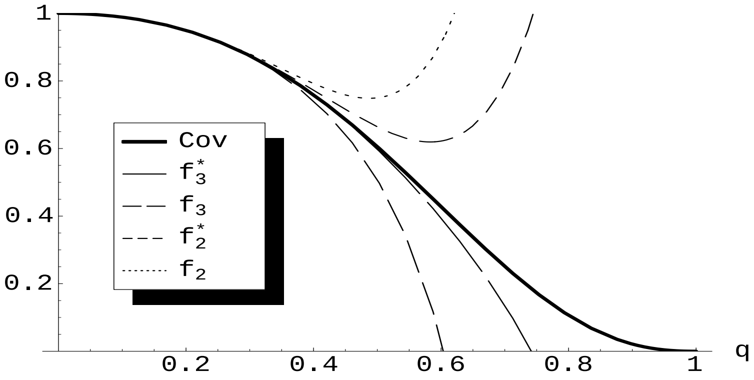


FIG. 4. *A plot of some of the bounds in Table* 1.

Remarks. The requirement that the connected block graph is of bounded clique number is essential, since otherwise we could take, e.g., the complete graph and thus reduce the problem to its unconstrained counterpart, which is #P-hard. Anyway, as in the following example, we can take full advantage of the improved Bonferroni inequalities associated with Theorem 6.1 and the convex geometry of Example 2.3.

EXAMPLE 6.4. Consider the hypergraph with vertices 1, 2, 3, 4, 5, 6, 7 and edges

$$\{1,2,3,4\},\{4,5,6,7\},\{1,6,7\},\{1,3,6\},\{2,3,5,7\},\{2,5,6\},\{2,6,7\},\{1,5,7\}.$$

Obviously, the edges of this hypergraph are complements of connected subgraphs of the connected block graph displayed in Figure 3. Therefore, we can apply Theorem 6.1 in connection with the convex geometry of Example 2.3 to obtain improved Bonferroni bounds on the coverage probability $\mathrm{Cov}(H;\mathbf{p})$ of this hypergraph. Under the assumption that the edges of the hypergraph fail randomly and independently with equal probability $q = 1 - p$, the results are shown in Table 1. Here, $f_r(q)$ (resp., $f_r^*(q)$) denotes the rth classical (resp., improved) Bonferroni bound, where even and odd values of r correspond to upper and lower bounds, respectively. As guaranteed by Proposition 3.3, the improved Bonferroni bounds are much sharper than the classical Bonferroni bounds, although much fewer sets are taken into account. Table 2 shows some numerical values and Figure 4 shows a plot of some of the bounds in Table 1.

REFERENCES

[1] M. O. Ball, J. S. Provan, and D. R. Shier, *Reliability covering problems*, Networks, 21 (1991), pp. 345–357.

[2] A. Björner and G. M. Ziegler, *Introduction to greedoids*, in Matroid Applications, N. White, ed., Cambridge University Press, Cambridge, UK, 1992, pp. 284–357.

[3] K. Dohmen, *An improvement of the inclusion-exclusion principle*, Arch. Math., 72 (1999), pp. 298–303.

[4] K. Dohmen, *Improved inclusion-exclusion identities and inequalities based on a particular class of abstract tubes*, Electron. J. Probab., 4 (1999), paper 5.

[5] K. Dohmen, *Improved Bonferroni inequalities via union-closed set systems*, J. Combin. Theory Ser. A, 92 (2000), pp. 61–67.

[6] P. H. Edelman and R. Jamison, *The theory of convex geometries*, Geom. Dedicata, 19 (1985), pp. 247–270.

[7] P. H. Edelman and V. Reiner, *Counting the interior points of a point configuration*, Discrete Comput. Geom., 23 (2000), pp. 1–13.

[8] H. Edelsbrunner and E. A. Ramos, *Inclusion-exclusion complexes for pseudodisk collections*, Discrete Comput. Geom., 17 (1997), pp. 287–306.

[9] M. Farber and R. E. Jamison, *Convexity in graphs and hypergraphs*, SIAM J. Alg. Disc. Meth., 7 (1986), pp. 433–444.

[10] J. Galambos and I. Simonelli, *Bonferroni-Type Inequalities with Applications*, Springer-Verlag, New York, 1996.

[11] B. Giglio, D. Naiman, and H. P. Wynn, *Abstract tube theory and Gröbner bases for improved inclusion-exclusion bounds*, Proceedings of the Second International Conference on Mathematical Methods in Reliability–MMR 2000, Bordeaux, France, 2000, pp. 451–454.

[12] B. Giglio, D. Naiman, and H. P. Wynn, *Gröbner bases, abstract tubes, and inclusion-exclusion reliability bounds*, IEEE Trans. Reliab., 58 (2002), pp. 358–366.

[13] G. Gordon, *A β invariant for greedoids and antimatroids*, Electron. J. Combin., 4 (1997), paper 13; printed version: J. Combin., 4 (1997), pp. 123–135.

[14] R. E. Jamison-Waldner, *Partition numbers for trees and ordered sets*, Pacific J. Math., 96 (1981), pp. 115–140.

[15] T. A. McKee, *Graph structure for inclusion-exclusion inequalities*, Congr. Numer., 125 (1997), pp. 5–10.

[16] D. Q. Naiman and H. P. Wynn, *Inclusion-exclusion-Bonferroni identities and inequalities for discrete tube-like problems via Euler characteristics*, Ann. Statist., 20 (1992), pp. 43–76.

[17] D. Q. Naiman and H. P. Wynn, *Abstract tubes, improved inclusion-exclusion identities and inequalities and importance sampling*, Ann. Statist., 25 (1997), pp. 1954–1983.

[18] D. Q. Naiman and H. P. Wynn, *Improved inclusion-exclusion inequalities for simplex and orthant arrangements*, JIPAM J. Inequal. Pure Appl. Math., 2 (2001), article 18.

[19] H. Narushima, *Principle of inclusion-exclusion on semilattices*, J. Combin. Theory Ser. A, 17 (1974), pp. 196–203.

[20] H. Narushima, *Principle of inclusion-exclusion on partially ordered sets*, Discrete Math., 42 (1982), pp. 243–250.

[21] J. J. Rotman, *An Introduction to Algebraic Topology*, Springer-Verlag, New York, 1988.

[22] D. R. Shier, *Algebraic aspects of computing network reliability*, in Applications of Discrete Mathematics, R. D. Ringeisen and F. S. Roberts, eds., SIAM, Philadelphia, 1988, pp. 135–147.

[23] D. R. Shier, *Network Reliability and Algebraic Structures*, Clarendon Press, Oxford, UK, 1991.

[24] H. Whitney, *A logical expansion in mathematics*, Bull. Amer. Math. Soc., 38 (1932), pp. 572–579.

SIAM J. Discrete Math.
Vol. 16, No. 2, pp. 173–178

THE EXISTENCE OF 2 × 4 GRID-BLOCK DESIGNS AND THEIR APPLICATIONS*

YUKIYASU MUTOH[†], TOSHIO MORIHARA[†], MASAKAZU JIMBO[†], AND HUNG-LIN FU[‡]

Abstract. Fu, Hwang, Jimbo, Mutoh, and Shiue [*J. Statist. Plann. Inference*, to appear] introduced the concept of a grid-block design, which is defined as follows: For a v-set V, let $\mathcal{A}$ be a collection of $r \times c$ arrays with elements in V. A pair $(V, \mathcal{A})$ is called an $r \times c$ *grid-block design* if every two distinct points i and j in V occur exactly once in the same row or in the same column. This design has originated from the use of DNA library screening. They gave some general constructions and proved the existence of 3×3 grid-block designs. Meanwhile, the existence of 2×3 grid-block designs was shown by Carter [*Designs on Cubic Multigraphs*, Ph.D. thesis, McMaster University, Hamilton, ON, Canada, 1989] by decomposing K_v into cubic graphs. In this paper, we show the existence of 2×4 grid-block designs.

Key words. graph decomposition, graph design, grid-block

AMS subject classifications. 05B05, 05C70

PII. S0895480101387364

1. Introduction.

A *graph* G is a pair of sets (V, E), where V is a finite set, and E is a set of unordered pairs of elements of V. The elements of V are called *vertices* of G and the elements of E are called *edges* of G. If x and y are vertices of a graph G, we say that x is *adjacent* to y if there is an edge between x and y. K_v is the graph with v vertices such that every vertex is adjacent to every other vertex. For a v-set V, let $\mathcal{A}$ be a collection of $r \times c$ arrays with elements in V. Each array in $\mathcal{A}$ is called a *grid-block*. For a graph $G = (V, E)$, a pair $(V, \mathcal{A})$ is called an $r \times c$ *grid-block design with respect to* G denoted by $D_{r\times c}(G)$ if every two distinct points i and j in V such that $\{i, j\} \in E$ occur exactly once in the same row or in the same column. We used the terminology "grid-block design" to avoid the confusion with the "grid design" defined by Lamken and Wilson [9]. Here we show an example of a $D_{3\times 3}(K_9)$.

Example 1. *The following two grid-blocks form a* $D_{3\times 3}(K_9)$.

$$\begin{array}{|ccc|}\hline 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ \hline \end{array} \qquad \begin{array}{|ccc|}\hline 1 & 6 & 8 \\ 9 & 2 & 4 \\ 5 & 7 & 3 \\ \hline \end{array}$$

A grid-block design was introduced by Fu et al. [7]. It is easy to show the following necessary conditions for the existence of a $D_{r\times c}(K_v)$.

Lemma 1.1. *Necessary conditions for the existence of a* $D_{r\times c}(K_v)$ *are*

(i) $(r + c - 2)|(v - 1)$ *and*

(ii) $rc(r + c - 2)|v(v - 1)$.

Combinatorial designs were used as an efficient way of group testing in fields such as medical science and pharmaceutical science (see Du and Hwang [6]). Recently, a

[*] Received by the editors April 25, 2001; accepted for publication (in revised form) July 24, 2002; published electronically February 20, 2003. This work was supported in part by the Institute of Statistical Mathematics in Japan.

http://www.siam.org/journals/sidma/16-2/38736.html

[†] Department of Mathematics, Keio University, Yokohama, Kanagawa, Japan 223-8522 (yukiyasu@jim.math.keio.ac.jp, toshio@jim.math.keio.ac.jp, jimbo@math.keio.ac.jp).

[‡] Department of Applied Mathematics, National Chiao Tung University, Hsin Chu, Taiwan, R.O.C. (hlfu@math.nctu.edu.tw).

combinatorial design has come to be applied to DNA library screening to discover the required DNA sequences by testing every row and every column in a microtiter plate at the same time.

In DNA library screening, a popular group testing method is a two-stage test. In this method, every row and every column in a microtiter plate is tested at the same time in the first stage, and each individual segment with positive response is tested in the second stage. See Figure 1.1 for demonstration. To reduce the number of tests and to improve the efficiency of experiments, several methods of screening have been studied by many authors.

Berger, Mandell, and Subrahmanya [1] evaluated the efficiency for the two-stage test from the point of view of information theory, while Fu et al. [7] introduced a combinatorial method based on a grid-block design.

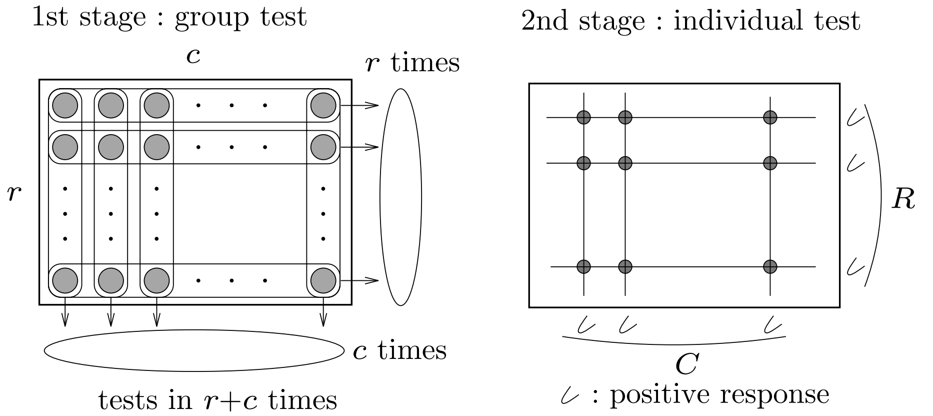


FIG. 1.1. *The demonstration of DNA library screening.*

In this paper, we start with a recursive construction for a grid-block design. Then, by utilizing this recursive construction together with those given by Fu et al. [7], we will prove the existence of 2×4 grid-block designs which satisfy the necessary condition $v \equiv 1 \pmod{32}$.

2. General constructions. In this section, we prepare a proposition and lemmas to use in the next section. First, we define a block design. For sets of positive integers K and M, let V be a set of v *points*, let $\mathcal{G}$ be a partition of V such that each G has m points for $m \in M$, and let $\mathcal{B}$ be a collection of k-subsets (*blocks*) of V for $k \in K$. A triple $(V, \mathcal{G}, \mathcal{B})$ is called a *group divisible design*, denoted by $GD[K, \lambda, M; v]$, if every two distinct points contained in different groups occur in exactly λ blocks and if every two distinct points contained in the same group do not occur together in any blocks. Especially, a $GD[\{k\}, \lambda, \{m\}; v]$ is written by $GD[k, \lambda, m; v]$ for simplicity of notation.

Suppose that the set of st vertices are partitioned into s subsets of size t each. Let $K_s(t)$ be the complete multipartite graph such that (i, j) is an edge if i and j are not in the same subset. A grid-block design $D_{r\times c}(K_s(t))$ is called a *group divisible grid-block design*. It is easy to see that the following lemma holds.

LEMMA 2.1. *Necessary conditions for a $D_{r\times c}(K_s(t))$ to exist are*

(i) $(r + c - 2)|(s - 1)t$ *and*

(ii) $rc(r + c - 2)|(s - 1)st^2$.

Fu et al. [7] proved the following construction.

Table 3.1
Table of the existence of group divisible designs.

v	K	Group type	u	Exceptions	Ref.
$0, 1 \pmod 4$	$\{4, 5\}$	1^u	$0, 1 \pmod 4$	12	[2]
12	4	3^4	–	–	[4]
$12 \pmod{12}$	4	2^u	$1 \pmod 3$	–	[4]
$3 \pmod{12}$	4	3^u	$1 \pmod 4$	–	[4]
$6 \pmod{12}$	4	6^u	Anything	18	[4]
$7 \pmod{12}$	4	$7^1 1^u$	$0 \pmod{12}$	19	[3]
$10 \pmod{12}$	4	$7^1 1^u$	$3 \pmod{12}$	–	[3]
$11 \pmod{12}$	4	$5^1 2^u$	$0 \pmod 3$	–	[3]

Proposition 2.2 (Fu et al. [7]). *A $D_{r\times c}(K_{st+1})$ exists if a $D_{r\times c}(K_{t+1})$ and a $D_{r\times c}(K_s(t))$ exist.*

We give a recursive construction by utilizing a group divisible design, group divisible grid-block designs, and grid-block designs.

Lemma 2.3. *A $D_{r\times c}(K_{vt+1})$ exists if a $GD[K, 1, M; v]$ exists and if a $D_{r\times c}(K_k(t))$ and a $D_{r\times c}(K_{mt+1})$ exist for any $k \in K$ and for any $m \in M$.*

Proof. For a v-set V, let a triple $(V, \mathcal{G}, \mathcal{B})$ be a $GD[K, 1, M; v]$, where $\mathcal{B} = \{B_1, B_2, \ldots, B_b\}$ is a collection of blocks and $\mathcal{G} = \{G_1, G_2, \ldots, G_n\}$ is a family of group sets. Let $T = \{0, 1, \ldots, t-1\}$ and $V^* = (V \times T) \cup \{\infty\}$. For each block B_i of size $k \in K$, let $(B_i \times T, \mathcal{H}_i, \mathcal{E}_i)$ be the ingredient design $D_{r\times c}(K_k(t))$, where $\mathcal{E}_i$ is a collection of grid-blocks and $\mathcal{H}_i$ is a family of group sets $\{\{b_{i1}\} \times T, \{b_{i2}\} \times T, \ldots, \{b_{ik}\} \times T\}$ for $b_{ij} \in B_i$. We define a collection of grid-blocks $\mathcal{A}_1 = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \cdots \cup \mathcal{E}_b$. Also, for each group G_i of size $m \in M$, let $((G_i \times T) \cup \{\infty\}, \mathcal{F}_i)$ be the ingredient design $D_{r\times c}(K_{mt+1})$, where $\mathcal{F}_i$ is a collection of grid-blocks. We define another collection of grid-blocks $\mathcal{A}_2 = \mathcal{F}_1 \cup \mathcal{F}_2 \cup \cdots \cup \mathcal{F}_n$ and let $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$. Then a pair $(V^*, \mathcal{A})$ is the desired $D_{r\times c}(K_{vt+1})$.

In fact, if two distinct elements x and y in V are not contained in the same group set G_j, then x and y occur together exactly once in a B_i. Hence (x, α_1) and (y, α_2) occur exactly once in the same row or in the same column of a grid-block in $\mathcal{A}_1$ and do not occur in $\mathcal{A}_2$ for any $\alpha_1, \alpha_2 \in T$. Otherwise, two elements x and y in V are contained in the same group set G_j including the case of $x = y$. In this case, (x, α_1) and (y, α_2) occur exactly once in the same row or in the same column of a grid-block in $\mathcal{A}_2$ and do not occur in $\mathcal{A}_1$. Finally, ∞ and (x, α) for any $x \in V$ and $\alpha \in T$ occur exactly once in the same row and in the same column of a grid-block in $\mathcal{A}_2$. ∎

3. The existence of a 2×4 grid-block design. In this section we apply the results obtained in the previous section to prove the following theorem.

Theorem 3.1. *The necessary condition $v \equiv 1 \pmod{32}$ for the existence of a $D_{2\times 4}(K_v)$ is also sufficient.*

This existence theorem is shown by utilizing a recursive construction. First, we give an existence of a group divisible design.

Lemma 3.2. *For any integer $v \geq 12$, there exists a $GD[K, 1, M; v]$, where $K = \{4, 5\}$ and $M = \{1, 2, \ldots, 7\}$.*

Proof. According to Brouwer [3], Brouwer, Schrijver, and Hanani [4], and Beth, Jungnickel, and Lenz [2], we know the existence of a $GD[K, 1; M; v]$ for any $v \geq 12$ except for $v = 18$ and 19 as is listed in Table 3.1 (see also Kreher and Stinson [8] and Mullin and Gronau [10]). In Table 3.1, the notation $t_1^{u_1} t_2^{u_2}$ of a group type implies that V is divided into u_1 groups with group size t_1 and u_2 groups with group size t_2.

TABLE 3.2
Table of the base grid-blocks of group divisible grid-block designs.

	Base grid-blocks		
$D_{2\times 4}(K_4(32))$	0 1 6 15 13 30 3 48	0 21 58 47 22 63 20 97	0 25 74 55 63 56 17 122
$D_{2\times 4}(K_5(32))$	0 1 7 3 11 27 48 39	0 31 17 63 22 73 129 30	0 66 47 133 13 149 105 51
	0 111 52 23 84 15 141 102		

TABLE 3.3
Table of the base grid-blocks of grid-block designs.

	Base grid-blocks		
$D_{2\times 4}(K_{33})$	0 1 3 9 12 5 23 28		
$D_{2\times 4}(K_{65})$	0 1 3 7 5 13 22 38	0 10 21 45 47 32 60 9	
$D_{2\times 4}(K_{97})$	0 1 3 7 5 13 22 33	0 10 23 41 33 65 86 3	0 15 37 61 39 55 84 12
$D_{2\times 4}(K_{193})$	0 36 65 60 89 155 152 153	0 46 180 153 186 23 71 169	0 55 108 73 114 77 133 81
	0 14 97 165 102 52 40 134	0 105 54 44 75 34 178 55	0 76 67 148 39 189 73 174
$D_{2\times 4}(K_{225})$	0 104 76 167 67 137 121 209	0 189 223 92 156 74 167 199	0 221 77 194 41 94 42 16
	0 122 177 140 212 190 106 67	0 15 220 111 95 76 55 46	0 7 10 24 38 82 206 32
	0 87 161 99 79 192 102 13		

Moreover, it is known that $GD[5,1,4;20]$ exists, which is obtained by deleting one parallel class of lines and five points on a line in the parallel class from $AG(2,5)$. By deleting a single point of a $GD[5,1,4;20]$, we can show the existence of a $GD[\{4,5\},1,\{3,4\};19]$. Similarly, by deleting two points from the same group of a $GD[5,1,4;20]$, we obtain a $GD[\{4,5\},1,\{2,4\};18]$, which proves the case of $v = 18$ and 19. Thus, the lemma is proved. □

Second, we give two group divisible grid-block designs which are obtained by computer.

LEMMA 3.3. *There exists a* $D_{2\times 4}(K_k(32))$ *for* $k = 4$ *and* 5.

Proof. For $V = \boldsymbol{Z}_{128}$, let

$$A_0 = \boxed{\begin{matrix} 0 & 1 & 6 & 15 \\ 13 & 30 & 3 & 48 \end{matrix}}, \qquad B_0 = \boxed{\begin{matrix} 0 & 21 & 58 & 47 \\ 22 & 63 & 20 & 97 \end{matrix}}, \quad \text{and}$$

$$C_0 = \boxed{\begin{matrix} 0 & 25 & 74 & 55 \\ 63 & 56 & 17 & 122 \end{matrix}},$$

which are listed in Table 3.2. Here A_0, B_0, and C_0 are called a *base grid-block* or a *starting grid-block*. For each base grid-block, let $A_i = A_0 + i \pmod{128}$, $B_i = B_0 + i \pmod{128}$, and $C_i = C_0 + i \pmod{128}$. Now we define

$$\mathcal{A} = \{A_0, A_1, \ldots, A_{127}, B_0, B_1, \ldots, B_{127}, C_0, C_1, \ldots, C_{127}\};$$

Table 3.4

Table of the base grid-blocks of grid-block designs (continued).

	Base grid-blocks											
$D_{2\times4}(K_{257})$	0	51	168	216	0	22	230	37	0	58	61	234
	148	147	81	37	30	211	187	193	200	118	101	154
	0	107	73	14	0	169	42	98	0	132	246	124
	50	79	202	176	63	61	96	216	20	41	72	162
	0	171	210	65	0	75	178	247				
	202	190	197	206	72	255	210	185				
$D_{2\times4}(K_{289})$	0	217	34	207	0	199	54	19	0	228	8	13
	28	188	253	168	105	282	236	183	86	35	165	189
	0	179	122	4	0	241	47	244	0	27	256	218
	209	37	211	284	124	191	110	98	248	182	225	98
	0	185	148	163	0	133	271	227	0	25	32	213
	128	186	216	180	166	14	150	206	77	255	266	164
$D_{2\times4}(K_{321})$	0	235	247	257	0	3	101	281	0	35	186	37
	310	101	228	133	76	105	212	309	244	138	264	16
	0	160	1	265	0	26	317	9	0	7	157	25
	158	66	291	221	269	178	228	315	23	205	143	74
	0	146	61	16	0	315	211	33	0	279	200	255
	283	288	174	115	206	78	146	254	34	105	272	308
	0	240	165	294								
	313	59	255	175								
$D_{2\times4}(K_{353})$	0	286	267	129	0	133	95	248	0	81	72	26
	198	149	219	118	22	20	275	113	82	257	147	261
	0	294	142	15	0	88	76	247	0	337	109	217
	34	173	198	1	71	222	144	194	66	150	2	211
	0	340	7	343	0	169	254	122	0	193	8	44
	195	5	234	264	316	229	17	59	352	103	127	76
	0	52	23	154	0	186	40	83				
	45	192	134	4	236	298	201	293				

then $(\boldsymbol{Z}_{128}, \mathcal{A})$ is the desired $D_{2\times4}(K_4(32))$. In fact, by calculating the differences of two elements in the same row or in the same column of A_0, B_0, and C_0, any difference except for multiples of 4 occurs exactly once.

Similarly, for $V = \boldsymbol{Z}_{160}$, by utilizing four base grid-blocks in Table 3.2, we obtain a $D_{2\times4}(K_5(32))$. In fact, by calculating the differences of two elements in the same row or in the same column of A_0, B_0, C_0, and D_0 any difference except for multiples of 5 occurs exactly once. $\square$

Third, we give some grid-block designs which are obtained by computer.

Lemma 3.4. *There exists a* $D_{2\times4}(K_{32m+1})$ *for any* $m = 1, 2, \ldots, 11$.

Proof. By utilizing the base grid-blocks in Tables 3.3 and 3.4, we obtain the desired $D_{2\times4}(K_{32m+1})$'s for $m = 1, 2, 3, 6, 7, \ldots, 11$. By applying Proposition 2.2 to a $D_{2\times4}(K_4(32))$ and a $D_{2\times4}(K_5(32))$ in Lemma 3.3 and a $D_{2\times4}(K_{33})$, $D_{2\times4}(K_{32m+1})$'s are obtained for $m = 4$ and 5. $\square$

Now we will show the main theorem.

Proof of Theorem 3.1. By Lemma 1.1, it is easy to show that the necessary condition for the existence of a $D_{2\times4}(K_v)$ is $v \equiv 1 \pmod{32}$. Now we write $v = 32w + 1$; then there exists a $D(K_{32w+1})$ for $w \leq 11$ by Lemma 3.4. By Lemma 3.2, a $GD[K, 1, M; w]$ exists for $w \geq 12$, where $K = \{4, 5\}$ and $M = \{1, 2, \ldots, 7\}$. And a $D(K_k(32))$ exists for $k = 4$ and 5 by Lemma 3.3. Thus by Lemma 2.3 a $D(K_{32w+1})$ exists for any $w \geq 12$, which prove the main theorem. $\square$

Acknowledgments. The authors would like to thank the referees for their careful readings of this article and for their many constructive comments. One of the referees suggested to us that the proof of the previous version can be shortened. The authors would also like to thank Professor R. C. Mullin for his valuable comments.

REFERENCES

[1] T. Berger, J. W. Mandell, and P. Subrahmanya, *Maximally efficient two-stage screening*, Biometrics, 56 (2000), pp. 833–840.

[2] T. Beth, D. Jungnickel, and H. Lenz, *Design Theory*, Cambridge University Press, Cambridge, UK, 1986.

[3] A. E. Brouwer, *Optimal packings of K_4's into a K_n*, J. Combin. Theory Ser. A, 26 (1979), pp. 278–297.

[4] A. E. Brouwer, A. Schrijver, and H. Hanani, *Group divisible designs with block size* 4, Discrete Math., 20 (1977), pp. 1–10.

[5] J. E. Carter, *Designs on Cubic Multigraphs*, Ph.D. thesis, McMaster University, Hamilton, ON, Canada, 1989.

[6] D.-Z. Du and F. K. Hwang, *Combinatorial Group Testing and Its Application*, World Scientific, River Edge, NJ, 2000.

[7] H. L. Fu, F. K. Hwang, M. Jimbo, Y. Mutoh, and C. L Shiue, *Decomposing complete graphs into $K_r \times K_c$'s*, J. Statist. Plann. Inference, to appear.

[8] D. L. Kreher and D. R. Stinson, *Small group-divisible designs with block size four*, J. Statist. Plann. Inference, 58 (1997), pp. 111–118.

[9] E. R. Lamken and R. M. Wilson, *Decompositions of edged-colored complete graphs*, J. Combin. Theory Ser. A, 89 (2000), pp. 149–200.

[10] R. C. Mullin and H.-D.O.F. Gronau, *PBDs and GDDs: The basics*, in The CRC Handbook of Combinatorial Designs, C. J. Colbourn and J. H. Dinitz, eds., CRC Press, Boca Raton, FL, 1996, pp. 185–193.

SIAM J. Discrete Math.
Vol. 16, No. 2, pp. 179–191

STIRLING NUMBERS FOR COMPLEX ARGUMENTS: ASYMPTOTICS AND IDENTITIES*

GRAEME KEMKES†, CHIU FAN LEE‡, DONATELLA MERLINI§, AND BRUCE RICHMOND†

Abstract. We derive asymptotic expansions for the Stirling numbers of real arguments as defined by Flajolet and Prodinger. We also generalize certain classical identities for Stirling numbers with integral arguments to real or complex arguments.

Key words. Stirling numbers, asymptotic enumeration

AMS subject classifications. 5A05, 5A16

PII. S0895480102401119

1. Introduction.

Recently Flajolet and Prodinger [4] defined the Stirling numbers of the second kind for complex arguments, solving a research problem of Graham, Knuth, and Patashnik [5]. They define *y set x* by

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

where $s! = \Gamma(s+1)$. The determination of $(e^z - 1)^x$ is the principal determination on the part of the contour $\Re z > 0$ extended by continuity to the whole of H. Here H is a Hankel contour (see [14]) that starts from $-\infty$ below the negative axis, goes around the origin counterclockwise, and returns to $-\infty$ in the half-plane $\Im z > 0$. The details of H are immaterial; we assume only that the singularities at $\pm k2\pi i, k = 1, 2, \dots$ are not inside H. This integral converges for $\Re y > 0$; however, if we integrate by parts we find that

$$\left\{ {y \atop x} \right\} = \frac{(y-1)!}{(x-1)!} \frac{1}{2\pi i} \int_H e^z (e^z - 1)^{x-1} \frac{1}{z^y} dz. \tag{2}$$

This integral converges for all values of x and y; thus $\left\{ {y \atop x} \right\}$ is a meromorphic function of y (for any fixed x) with poles at the nonpositive integers. As a function of x (for any fixed y) it is entire. We adopt this definition of $\left\{ {y \atop x} \right\}$.

Flajolet and Prodinger do not discuss in great detail the Stirling numbers of the first kind for complex arguments; however, their paper implies a possible definition of *y cycle x* by

$$\left[{y \atop x} \right] = \frac{y!}{x!} \frac{1}{2\pi i} \int_{H'} z^{-y-1} \ln^x \left(\frac{1}{1-z} \right) dz, \tag{3}$$

[*] Received by the editors January 18, 2002; accepted for publication (in revised form) October 14, 2002; published electronically February 20, 2003.
http://www.siam.org/journals/sidma/16-2/40111.html

[†] Department of Combinatorics and Optimization, University of Waterloo, Waterloo, ON N2L3G1, Canada (gdkemkes@student.math.uwaterloo.ca, lbrichmond@uwaterloo.ca).

[‡] Department of Physics, University College, Oxford OX1 4BH, UK (chiu.lee@university-college.oxford.ac.uk).

[§] Department di Sistemi e Informatica, Universita di Firenze, I-50100 Firenze, Italy (merlini@dsi.unifi.it).

where $H^{'}$ is a contour which starts at 1, circles the origin in the counterclockwise direction, and returns to 1. (Note H is the image of $H^{'}$ under the mapping which replaces z by $1-e^z$.) We adopt this definition of $\left[{y \atop x}\right]$.

These generalized Stirling number functions are interesting for several reasons. For example Flajolet and Prodinger show that

$$\frac{d}{dy}\left\{ {x \atop y} \right\}_{y=1} = \zeta(-x).$$

Thus finding the zeros of this derivative is equivalent to solving the famous Riemann hypothesis! We do not give the proof here; however, the argument of Sprugnoli and Del Lungo [13] can be adapted easily to give

$$\zeta(s) = \frac{1}{1-2^{1-s}} \sum_{k=1}^{\infty} \left\{ {-s \atop k} \right\} \frac{k!(-1)^{k-1}}{2^{k+1}}.$$

A celebrated identity between the first and second Stirling numbers is

$$\left\{ {x \atop y} \right\} = \left[{-y \atop -x} \right], \qquad x \text{ and } y \text{ integral.}$$

This is equation (6.33) of [5]; see [7] for the fascinating history of this identity. Richmond and Merlini [9] show that with the above definitions this identity holds when x and y are complex numbers such that x-y is an integer. In our last theorem of this paper we extend this identity to all complex x and y.

The best definition of the generalized Stirling numbers may not be the ones we give. Knuth [6] gives a definition of y *set* x when x is an integer and y is an integer plus 1/2. With his definition the value of 2.5 *set* 2 is 2; however, with the Flajolet–Prodinger definition, 2.5 *set* 2 equals $2\sqrt{2}-1$ (ask Maple). Chelluri, Richmond, and Temme [2] used a different definition of y *cycle* x which was very convenient to derive asymptotic estimates. The definition of [2] agrees with our present one for integral x and y. Flajolet and Prodinger derive the formula for integral k,

$$\left\{ {y \atop k} \right\} = \frac{1}{k!} \sum_{j=0}^{k} \binom{k}{j} (-1)^{k-j} j^y,$$

which is the definition of Sprugnoli and Del Lungo [13].

We aim to generalize certain well-known identities with integral arguments to complex arguments. For example the arguments of [9] when applied to the Flajolet–Prodinger definition of y *cycle* x very easily give

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

We let, using (2),

$$\left\{ {y \atop x} \right\} = \frac{(y-1)!}{(x-1)!} B_y(x)$$

and show the following.

THEOREM 1. *If $0 < d < 1$, then*

$$B_y(c+d) = \sum_{k \geq 0} \frac{d}{k+d} B_{k+d}(d) B_{y-k-d}(c) \tag{4}$$

converges absolutely for all y and c. If $d = 1$, the convergence is absolute for $y > 1$. Furthermore

$$\binom{c+d-1}{c} \left\{ {y \atop c+d} \right\} = \sum_{k \geq 0} \binom{y-1}{k+d} \left\{ {k+d \atop d} \right\} \left\{ {y-k-d \atop c} \right\}. \tag{5}$$

If c, d, and y are nonnegative integers, then the sum is finite and gives what seems to be a new identity for these Stirling numbers.

If we use (1) to define $B_y^{'}(x)$ ($B_y^{'}(x)$ is not a derivative) by

$$B_y^{'}(x) = \frac{x!}{y!} \left\{ {y \atop x} \right\},$$

then we find for x, y nonnegative integers that $B_y^{'}(x)$ is a convolution family as defined by Knuth [8]. A convolution family is also called a binomial function; see Olive [10]. These references show that convolution families or binomial functions have an extensive literature. One of the goals of Olive [10] is to generalize certain binomial functions and we shall generalize some of her binomial functions further by allowing certain of her parameters to be real instead of integral. We shall study $C_y(x)$ which is defined for $\left[{y \atop x} \right]$ as $B_y^{'}(x)$ is for $\left\{ {y \atop x} \right\}$ and find that it is completely analogous to a convolution family. The convolution family $B_y^{'}(x)$ gives a sequence, $B_y^{''}(x)$, of binomial type (see [12, p. 8])

$$B_y^{''}(x) = y! B_y^{'}(x).$$

THEOREM 2. *If c, d, and y are nonnegative integers, then*

$$B_y^{'}(c+d) = \sum_{k \geq 0} B_k^{'}(d) B_{y-k}^{'}(c)$$

and

$$\binom{c+d}{d} \left\{ {y \atop c+d} \right\} = \sum_{k \geq 0} \binom{y}{k+d} \left\{ {k+d \atop d} \right\} \left\{ {y-d-k \atop c} \right\}.$$

Remark. If c, d, and y are nonnegative integers, then this last sum is finite and questions of convergence do not arise. Theorem 2 becomes equation (6.28) of Graham, Knuth, and Patashnik [5]. Our proof of Theorem 2 is another proof of this identity.

We next give two theorems giving the asymptotic behavior of $\left\{ {y \atop x} \right\}$ as $y \to -\infty$, x fixed and as $x \to \infty$, y fixed (there is a paper of Chelluri, Richmond, and Temme [2] giving uniform asymptotic expansions of $\left\{ {y \atop x} \right\}, \left[{y \atop x} \right]$ as $y \to \infty$ for $0 < x < y$). The results in [2] correspond to the well-known asymptotic results for $\left\{ {n \atop k} \right\}$ and $\left[{n \atop k} \right]$. One simply replaces the integer variables n and k by the real variables y and x in the asymptotic formulas. The two theorems just referred to, however, do not have such analogies since in the integer variable case the Stirling numbers become zero as $k \to \infty$ or $n \to -\infty$. See the second paragraph before Theorem 3 for further discussion.

These results are used to prove that if $C_y(x) = \frac{x!}{y!}\left[{y \atop x}\right]$, then $\sum_{k\geq 0} C_{k+d}(d)C_{y-k-d}(c)$ converges to $C_y(c+d)$ provided $c+d<0$. This gives us a generalization of equation (6.29) in Graham, Knuth, and Patashnik [5], namely

$$\binom{c+d}{d}\left[{y \atop c+d}\right] = \sum_{k\geq 0}\binom{y}{k}\left[{k \atop d}\right]\left[{y-k \atop c}\right].$$

Equation (6.29) of Graham, Knuth, and Patashnik is the case when c, d, and y are nonnegative integers (since the sum is finite the condition $c+d<1$ implying convergence can be dropped).

The references [1], [3], [5], [7] contain many results concerning Stirling numbers of both kinds.

2. Proofs and further results. The contour in our definition (2) can be moved as long as we do not pass through a singularity of the integrand. We can deform H to a small circle at the origin so that the contour is inside the radius of convergence of any given power series analytic at the origin without changing the value of the integrand. We shall do this so that a term-by-term integration is justified, and then we will deform the circle back to H in the proof of Theorem 1. The contour H is very convenient for proving identities involving the Gamma function because of Hankels' formula so we adopt it in our definition (2) as Flajolet and Prodinger do.

First, for $k \in Z$, $k \geq 0$, it follows from Cauchy's integral formula that

$$[z^k]\left(\frac{e^z-1}{z}\right)^d = \frac{1}{2\pi i}\int_H \left(\frac{e^z-1}{z}\right)^d \frac{dz}{z^{k+1}} = \frac{1}{2\pi i}\int_H (e^z-1)^d \frac{dz}{z^{k+d+1}}.$$

Upon integrating by parts we have, since $d>0$,

$$= \left|\frac{-(e^z-1)^d}{(k+d)z^{k+d}}\right|_H + \frac{d}{k+d}\frac{1}{2\pi i}\int_H \frac{(e^z-1)^{d-1}}{z^{d+k}}e^z dz;$$

the first expression equals 0 so this equals $\frac{d}{k+d}B_{k+d}(d)$. Thus for small z

$$\sum_{k\geq 0}\frac{d}{k+d}B_{k+d}(d)z^k = \left(\frac{e^z-1}{z}\right)^d.$$

Proof of Theorem 1. Since $B_y(x) = (x-1)!\left\{{y \atop x}\right\}/(y-1)!$ we can rewrite sum (4) of Theorem 1 as

$$\frac{(c+d-1)!}{(y-1)!}\left\{{y \atop c+d}\right\} = \sum_{k\geq 0}\frac{d}{(k+d)}\frac{(d-1)!}{(k+d-1)!}\left\{{k+d \atop d}\right\}\frac{(c-1)!}{(y-d-k-1)!}\left\{{y-d-1 \atop c}\right\}$$

so

$$\binom{c+d-1}{d}\left\{{y \atop c+d}\right\} = \sum_{k\geq 0}\binom{y-1}{k+d}\left\{{k+d \atop d}\right\}\left\{{y-d-k \atop c}\right\};$$

hence sum (5) follows from sum (4). We now investigate the convergence of sum (4) of Theorem 1. We first bound $B_{y-k-d}(c)$. Suppose $c\geq 1$ first of all. We break up the contour of integration H into I, where $z=re^{-\pi i}$ below the real axis, and II, where $z=re^{i\pi}$ above the real axis. We may suppose k so large that $y-k-d<0$; then

$$\int_I = \int_\infty^0 (e^{-r}-1)^{c-1}e^{-r}e^{-i\pi(k+d-y)}r^{k+d-y}e^{-i\pi}dr$$

$$= -e^{-i\pi(k+d-y+c)} \int_0^\infty (1-e^{-r})^{c-1} e^{-r} r^{k+d-y} dr.$$

Also

$$\int_{II} = e^{i\pi(k+d-y+c)} \int_0^\infty (1-e^{-r})^{c-1} e^{-r} r^{k+d-y} dr.$$

Hence

$$B_{y-k-d}(c) = \frac{\sin \pi(k+d-y+c)}{\pi} \int_0^\infty (1-e^{-r})^{c-1} e^{-r} r^{k-y} dr.$$

When $c \geq 1$ this last integral is bounded in absolute value by

$$\int_0^\infty e^{-r} r^{k+d-y} dr = \Gamma(k+d-y+1).$$

Now $B_k(d)(k-1)!/(d-1)! = \left\{ {k \atop d} \right\}$. We use the asymptotic behavior of $\left\{ {k \atop d} \right\}$ derived in Chelluri, Richmond, and Temme [2]. There u_0 is defined by $1 - e^{-u_0} = du_0/k$, so we have $u_0 = k/d + O(ke^{-k/d}/d)$. The $H_0(u_0)$ of [2] is therefore equal to $1 + O(e^{-k/d})$ and Theorem 1 of [2] gives that $\left\{ {k \atop d} \right\} \sim d^k/d!$. Thus we have

$$B_{k+d}(d) \sim \frac{d^{k+d}}{(k+d-1)!d} = \frac{d^{k+d-1}}{(k+d-1)!}$$

and

$$\frac{d}{k+d} B_{k+d}(d) B_{y-k-d}(c) = O\left(\frac{d^k}{\Gamma(k+d)} \frac{\Gamma(k+d-y+1)}{k} \right) = O\left(\frac{d^k}{k^y} \right).$$

Hence, the first series in Theorem 1 converges for all y if $0 < d < 1$ and for $y > 1$ if $d = 1$.

Suppose now $c < 1$. In this case we break up the range of integration into I, where $z = re^{-i\pi}$, $1 \leq r \leq \infty$; III, where $z = re^{i\pi}$, $1 \leq r \leq \infty$; and II, where $z = e^{i\theta}$, $-\pi \leq \theta \leq \pi$. As before we deduce that

$$\int_I + \int_{II} = 2i \sin \pi(k+d-y-c) \int_1^\infty \frac{e^{-r} r^{k-x}}{(1-e^{-r})^{1-c}} dr$$

and

$$\int_{II} = \int_{-\pi}^{\pi} \frac{e^{e^{i\theta}} e^{i\theta(k+d-y+1)}}{(e^{e^{i\theta}} - 1)^{1-c}} d\theta$$

which is $O(1)$, where the O-constant depends on c. Now

$$\int_1^\infty \frac{e^{-r} r^{k+d-y}}{(1-e^{-r})^{1-c}} \leq \frac{1}{(1-e^{-1})^{1-c}} \int_1^\infty e^{-r} r^{k+d-y} dy \leq \frac{1}{(1-e^{-1})^{1-c}} \Gamma(k+d-y+1)$$

and the argument proceeds as before. This proves that the first sum in Theorem 1 converges under the assumptions of Theorem 1.

Now from the definitions of $B_y(x)$ and $\left\{ {y \atop x} \right\}$ we have

$$\frac{d}{k+d} B_{k+d}(d) B_{y-k-d}(c) = \frac{1}{2\pi i} \int_H \frac{k}{k+d} B_{k+d}(d) \frac{(e^z-1)^{c-1}}{z^{y-d-k}} e^z dz.$$

We can deform H to a small circle, C, centered at the origin without changing the value of the integral. If we sum over k, now our result for $[z^k](e^z - 1/z)^d$ gives

$$\sum \frac{d}{k+d} B_{k+d}(d) B_{y-k-d}(c) = \frac{1}{2\pi i} \int_C \left(\frac{e^z - 1}{z} \right)^d (e^z - 1)^{c-1} e^z dz.$$

We now deform the contour C into H, since the integral over H is $B_y(c+d)$, and Theorem 1 is proved. □

Proof of Theorem 2. The proof is quite similar to the proof of Theorem 1. We have

$$B_y^{'}(c+d) = \frac{1}{2\pi i} \int_H (e^z - 1)^c (e^z - 1)^d \frac{dz}{z^{y+1}}.$$

A nonnegative integral power of $e^z - 1$ is analytic so, for integral k,

$$[z^k](e^z - 1)^d = \frac{1}{2\pi i} \int_H (e^z - 1)^d \frac{dz}{z^{k+1}} = B_k^{'}(d)$$

so, we have as before

$$B_y^{'}(c+d) = \sum_{k \geq 0} B_k^{'}(d) B_{y-k}^{'}(c). \qquad \square$$

We now establish some further asymptotic relations that $B_y(x)$ satisfies. We use Laplace's method, a form of the saddle-point method. This method is used to derive the asymptotic behavior of a contour integral when the integrand has the form $f(x)^u$ as $u \to \infty$. The method works when $f(x)$ has a unique maximum at x_0 and decreases in absolute value as x moves away from x_0. In our applications $f(x)$ is log-concave.

We now discuss the asymptotic behavior of $B_{-y}(x)$ as y, then x, goes to ∞ in Theorem 3 and 4, respectively. We use only Theorem 4 to prove the identity in Theorem 5, so readers may skip Theorem 3 if their interest is not in asymptotics for its own sake.

In the next theorem the asymptotic relation for $B_{-y}(x)$ is given in terms of a function l defined implicitly as a function of x and y.

THEOREM 3. *Let l be defined by*

$$e^{ly} - x = \frac{e^{ly} - 1}{l}.$$

Then

$$B_{-y}(x) \sim \sin(\pi(y+x)) l^{y-1} \frac{y!}{e^{(l-1)y}} \ \text{ as } y \to \infty$$

and $l \sim 1$. Indeed if $x = 1$, then $l = 1$ and

$$B_{-y} \sim \sin(\pi(y+1)) y!$$

so

$$\left\{ {y \atop 1} \right\} \sim -\frac{\pi}{y+1}$$

as $y \to \infty$.

Proof. Note

$$B_{-y}(x) = \frac{1}{2\pi i}\int_H e^z \left(\frac{e^z-1}{z}\right)^{x-1} z^{x+y-1} dz.$$

As $y \to \infty$ the integrand goes to 0 as $z \to 0$; hence

$$\int_H = 2i\sin(\pi(x+y))\int_0^\infty e^{-r}(1-e^{-r})^{x-1} r^y dr.$$

We let

$$F(y) = \int_0^\infty e^{-r}(1-e^{-r})^{x-1} r^y dr$$

$$= \int_0^\infty \exp\left(y\left(\frac{-r}{y} + \frac{x-1}{y}\ln(1-e^{-r}) + \ln r\right)\right) dr = \int_0^\infty \exp(yS(r))dr.$$

We now apply Laplace's method; see [11, sect. 7]. Let

$$S(r) = \frac{-r}{y} + \frac{x-1}{y}\ln(1-e^{-r}) + \ln r$$

so that

$$S^{'}(r) = -\frac{1}{y} + \frac{x-1}{y}\frac{1}{e^r-1} + \frac{1}{r},$$

$$S^{''}(r) = -\frac{x-1}{y}\frac{e^r}{(e^r-1)^2} - \frac{1}{r^2}.$$

Since $S(r)$ is concave for any r we have that the unique maximum of $S(r)$ is at $S^{'}(r_0) = 0$ or

$$\frac{e^{r_0}-x}{y} = \frac{e^{r_0}-1}{r_0} \text{ or } r_0\left(\frac{e^{r_0}-x}{e^{r_0}-1}\right) = y$$

and that $r_0 \to \infty$ as $y \to \infty$. If we let $r_0 = ly$, then l is defined as in Theorem 3. Furthermore

$$S(r_0) = -l + \frac{x-1}{y}\ln(1-e^{-ly}) + \ln(ly) = -l + \ln(ly) + O\left(\frac{1}{y}\right),$$

$$yS^{''}(r_0) = -\frac{1}{l^2 y} + O(e^{-ly}),$$

so from Laplace's method we have

$$F(y) \sim \frac{\sqrt{2\pi y}}{l} e^{-ly + y\ln(ly)}$$

$$\sim \frac{\sqrt{2\pi y}}{l}\frac{(ly)^y}{e^{ly}} \sim l^{y-1}\frac{y!}{e^{(l-1)y}}$$

using Stirling's formula for $y!$. The first part of Theorem 3 follows. Finally

$$\left\{ \begin{matrix} -y \\ 1 \end{matrix} \right\} \sim (-y)!y! \sin \pi(y+1) \sim \frac{-\pi}{(y+1)\sin \pi(y+1)} \sin \pi(y+1).$$

It only remains to show that $l \sim 1$ as $y \to \infty$. We have seen that $r_0 = ly \to \infty$. If

$$e^{ly} - x = \frac{e^{ly}-1}{l}, \qquad \text{then } l = \frac{e^{ly}-1}{e^{ly}-x},$$

so $l \sim 1$ if $ly \to \infty$, and Theorem 3 is proved. □

THEOREM 4. *Suppose $y \geq 0$. Then*

$$B_{-y}(x) \sim \frac{\sin \pi(y+x)}{e} \sqrt{2/\pi} \frac{\ln^y x}{x}$$

as $x \to \infty$ and

$$\left\{ \begin{matrix} -y \\ x \end{matrix} \right\} \sim \frac{(-y)!}{x!} \frac{\sin(\pi(x+y))}{e} \sqrt{2/\pi} \frac{\ln^y x}{x-1}$$

as $x \to \infty$.

Proof. We have

$$B_{-y}(x) = \frac{1}{2\pi i} \int_H e^z (e^z - 1)^{x-1} z^y dz.$$

As $x \to \infty$ the integrand $\to 0$ as $x \to \infty$ for small z for any $y \geq 0$. Hence

$$\int_H = \int_I + \int_{II},$$

where

$$\int_I = \int_\infty^0 (e^{-r}-1)^{x-1} e^{-r} e^{-i\pi(y+1)} r^y dr$$

$$= -e^{-i\pi(x+y)} \int_0^\infty e^{-r}(1-e^{-r})^{x-1} r^y dr.$$

Similarly

$$\int_{II} = e^{i\pi(x+y)} \int_0^\infty e^{-r}(1-e^{-r})^{x-1} r^y dr$$

so

$$\int_H = 2i \sin(\pi(x+y)) \int_0^\infty e^{-r}(1-e^{-r})^{x-1} r^y dr.$$

We now proceed as in the proof of Theorem 3. Let us set

$$E(y) = \int_0^\infty e^{-r}(1-e^{-r})^{x-1} r^y dr$$

$$= \int_0^\infty e^{x[\frac{-r}{x} + \frac{x-1}{x} \ln(1-e^{-r}) + \frac{y}{x} \ln r]} dr$$

$$= \int_0^\infty e^{xS(r)} dr,$$

where

$$S^{'}(r) = \frac{-1}{x} + \frac{x-1}{x}\frac{1}{e^r-1} + \frac{y}{rx},$$

$$S^{''}(r) = -\frac{x-1}{x}\frac{e^r}{(e^r-1)^2} - \frac{y}{r^2x},$$

and we define r_0 by

$$-1 + \frac{x-1}{e^{r_0}-1} + \frac{y}{r_0} = 0.$$

We find that $r_0 = \ln x + O(1/(\ln x))$ as $x \to \infty$ and that

$$xS(r_0) = y \ln\ln x - \ln x - 1 + O(1/(\ln x)),$$

$$xS^{''}(r_0) = -1 + O(1/(\ln x)),$$

so again from Laplace's method we have

$$E(y) \sim \sqrt{\frac{-2\pi}{xS^{''}(r_0)}} e^{xS(r_0)} \sim \sqrt{2\pi} \ln^y x \left(\frac{1}{x}\left(1-\frac{1}{x}\right)^{x-1}\right)$$

$$\sim \frac{\sqrt{2\pi}}{e(x-1)} \ln^y x;$$

hence

$$B_{-y}(x) \sim \frac{\sin(\pi(x+y))}{e}\sqrt{\frac{2}{\pi}}\frac{\ln^y(y)}{x}$$

as $x \to \infty$ and

$$\left\{ \begin{matrix} -y \\ x \end{matrix} \right\} \sim \frac{(-y)!}{x!}\frac{\sin \pi(x+y)}{e}\sqrt{\frac{2}{\pi}} \ln^y x/x.$$

This proves Theorem 4. □

We conclude with a discussion of identities corresponding to Theorems 1 and 2 for the Stirling numbers of the first kind. Let us define $C_x(y)$ by

$$C_y(x) = \frac{1}{2\pi i}\int_{H'} \ln^x\left(\frac{1}{1-z}\right) z^{-y-1} dz$$

so that

$$\left[\begin{matrix} y \\ x \end{matrix} \right] = \frac{y!}{x!} C_y(x).$$

Then if we let $u = 1 - e^z$, $du = -e^z dz$, we have

$$C_y(x) = e^{i\pi(x-y)} \frac{1}{2\pi i} \int_H z^x \frac{e^z}{(e^z-1)^{y+1}} dz$$

$$= e^{i\pi(x-y)} B_{-x}(-y).$$

We then easily find that, as with $B_y^{'}(x)$,

$$C_y(c+d) = \sum_{k\geq 0} C_{k+d}(d) C_{y-k-d}(c)$$

provided the series converges. We determine the asymptotic behavior of $C_k(d)C_{y-k}(c)$. This is convenient because we can replace k by $k+d$ in our final estimate without affecting convergence. We have

$$C_{y-k}(c) = e^{i\pi(c-y+k)} B_{-c}(k-y).$$

From Theorem 4 we obtain

$$C_{y-k}(c) \sim \sqrt{2/\pi} e^{i\pi(c+k)} \frac{\sin(\pi(c+k-y))}{e} \frac{\ln^c(k-y)}{k-y}.$$

Now from Chelluri, Richmond, and Temme [2] we have

$$\begin{bmatrix} k \\ d \end{bmatrix} = \frac{k!}{d!} C_k(d) \sim \frac{k!(\log k)^{d-1}}{d!}$$

so $C_k(d) \sim (\ln k)^{d-1}$ and

$$C_k(d) C_{y-k}(c) \sim \frac{1}{e} \sqrt{2/\pi} e^{i\pi(c+k-y)} \sin(\pi(c+k-y)) \frac{\ln^c(k-y)(\ln k)^{d-1}}{k-y}.$$

Furthermore this series does not alternate with k. The series will not converge, therefore, unless $c+d-1 < -1$ or $c+d < 0$. Thus we have the following.

Theorem 5. *If $c+d<0$, then*

$$C_y(c+d) = \sum_{k\geq 0} C_k(d) C_{y-k}(c)$$

and, using $\left[\begin{smallmatrix} y \\ x \end{smallmatrix}\right] = y! C_y(x)/x!$,

$$\binom{c+d}{d} \begin{bmatrix} y \\ c+d \end{bmatrix} = \sum_{k\geq 0} \binom{y}{k} \begin{bmatrix} k \\ d \end{bmatrix} \begin{bmatrix} y-k \\ c \end{bmatrix}.$$

Remark. If c, d, y are nonnegative integers, the sum is finite and we have a proof of equation (6.26) of [5].

We now consider equation (6.15) of [5], namely

$$\left\{ \begin{matrix} n+1 \\ k+1 \end{matrix} \right\} = \sum_{j\geq 0} \binom{n}{j} \left\{ \begin{matrix} j \\ k \end{matrix} \right\}.$$

One way to generalize this identity is to follow Sprugnoli and Del Lungo [13]. Consider

$$\sum_{j\geq 0}\binom{y}{j}\left\{ {j \atop x} \right\};$$

however, as they show, $\left\{ {j \atop x} \right\} \sim x^j/j!$ for $0 < x$ so this series only converges for $0 \leq x < 1$. The identity can be generalized this way; however, we prefer the following path.

THEOREM 6. *If $y > 0$, then*

$$\sum_{j\geq 0}\binom{y}{j}\left\{ {y-j \atop x} \right\} = \left\{ {y+1 \atop x+1} \right\}.$$

Remark. If we set $y - j = k$, $y = n$, $z = k$, we recover equation (6.15) of [5].

Proof. Recall that we can deform H to a small circle C,

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

$$= \sum_{j\geq 0}\frac{y!}{j!x!}\frac{1}{2\pi i}\int_C \frac{(e^z-1)^x}{e^{y+1-j}}dz = \sum_{j\geq 0}\frac{y!}{j!(y-j)!}\frac{(y-j)!}{x!}\frac{1}{2\pi i}\int_C\frac{(e^z-1)^x}{z^{y+1-j}}dz$$

$$= \sum_{j\geq 0}\binom{y}{j}\left\{ {y-j \atop x} \right\},$$

provided the sum converges absolutely so that the interchange of summation and integration is justified. We rewrite this as

$$\frac{y!}{(x-1)!}\sum_{j\geq 0}\frac{1}{j!(y-j)!(x-1)!}B_{y-j}(x).$$

Case I. $x \geq 1$. Suppose j is large enough that $y - j < 0$. As before, we can break the integration into a range over r, where $z = re^{-\pi i}$ and $z = re^{\pi i}$. Then

$$B_{y-x}(x) = \frac{\sin\pi(j-y+x)}{\pi}\int_0^\infty (1-e^{-r})^{x-1}e^{-r}r^{j-y}dr.$$

This last integral is bounded by $\Gamma(j-y+1)$ so

$$\frac{1}{j!(y-j)!}B_{y-j}(x) = O\left(\frac{\Gamma(j-y+1)}{\Gamma(j+1)(y-j)}\right) = O(j^{-y-1});$$

hence we have convergence if $y > 0$.

Case II. $x < 1$. We break the range of integration up into I, where $z = re^{-i\pi}$, $1 \leq r \leq \infty$; II, where $z = e^{i\theta}$, $\pi < \theta < \pi$; and III, where $z = re^{i\pi}$, $1 \leq r < \infty$. As before

$$\int_I + \int_{III} = 2i\sin(\pi(j-y+x))\int_1^\infty e^{-r}r^{j-y}(1-e^{-r})^{x-1}dr$$

$$\leq (1-e^{-1})^{x-1}\int_1^\infty e^{-r}r^{j-y}dr \leq (1-e^{-1})^{x-1}\Gamma(j-y+1).$$

Clearly $\int_{II}$ is $O(1)$ and the O-constant depends on x. Thus as in Case I we have convergence for $y > 1$. This proves the theorem. □

We now prove the following.

Theorem 7. *We have*

$$\left\{\begin{matrix} -y \\ -x \end{matrix}\right\} = e^{i\pi(x-y)}\frac{\sin(\pi x)}{\sin(\pi y)}\left[\begin{matrix} x \\ y \end{matrix}\right].$$

Proof. By definition

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

Let $z = \log(1-w)$, so $w = 1-e^z$, $dz = -(1-w)^{-1}dw$. Then

$$\left\{\begin{matrix} -y \\ -x \end{matrix}\right\} = \frac{(-y-1)!}{(-x-1)!}\frac{1}{2\pi i}\int_{H'}\left(-\log\left(\frac{1}{1-w}\right)\right)^y(-w)^{-x-1} - dw,$$

where $H^{'}$ starts at 1, circles the origin counterclockwise, and returns to 1. Furthermore suppose $x < 0$; then we could start at $-\infty e^{i\pi}$, go to 0 along $re^{-i\pi}$, and return along $z = re^{i\pi}$ to $-\infty e^{i\pi}$. Then at first

$$(\log(1-w))^y = z^y = r^y e^{-i\pi y} = e^{-i\pi y}\left(\log\left(\frac{1}{1-w}\right)\right)^y.$$

Also w goes counterclockwise and so does $-w$. Thus

$$\frac{1}{(-w)^{x+1}} = -e^{i\pi(x-1)}\frac{1}{w^{x+1}}$$

so

$$\left\{\begin{matrix} -y \\ -x \end{matrix}\right\} = \frac{(-y-1)!}{(-x-1)!}e^{i\pi(x-y)}\int_{H'}\log^y\left(\frac{1}{1-w}\right)\frac{1}{w^{x+1}}dw$$

$$= \frac{\Gamma(x+1)\sin(\pi(x+1))}{\Gamma(y+1)\sin(\pi(y+1))}e^{i\pi(x-y)}\int_{H'}\log^y\left(\frac{1}{1-w}\right)\frac{1}{w^{x+1}}dw = e^{i\pi(x-y)}\frac{\sin\pi x}{\sin\pi y}\left[\begin{matrix} x \\ y \end{matrix}\right]$$

as required. We proved this for $x < 0$ but it holds wherever the expressions in the theorem are analytic. This proves the theorem. □

Acknowledgments. We wish to thank Renzo Sprugnoli for several helpful comments and we thank an anonymous referee for many comments which have improved our presentation.

REFERENCES

[1] V. Adamchik, *On Stirling numbers and Euler sums*, J. Comput. Appl. Math., 79 (1997), pp. 119–130.

[2] R. Chelluri, L. B. Richmond, and N. M. Temme, *Asymptotic estimates for generalized Stirling numbers*, Analysis, 20 (2000), pp. 1–13.

[3] L. Comtet, *Advanced Combinatorics*, D. Reidel, Dordrecht, 1974.

[4] P. Flajolet and H. Prodinger, *On Stirling numbers for complex arguments and Hankel contours*, SIAM J. Discrete Math., 12 (1999), pp. 155–159.

[5] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, 2nd ed., Addison-Wesley, Reading, MA, 1994.

[6] D. E. Knuth, *An analysis of optimum caching*, J. Algorithms, 6 (1985), pp. 181–199.

[7] D. E. Knuth, *Two notes on notation*, Amer. Math. Monthly, 99 (1992), pp. 403–422.

[8] D. E. Knuth, *Convolution polynomials*, Mathematica Journal, 2 (1992), pp. 67–78.

[9] B. Richmond and D. Merlini, *Stirling numbers for complex arguments*, SIAM J. Discrete Math., 10 (1997), pp. 73–82.

[10] G. Olive, *Binomial functions and combinatorial mathematics*, J. Math. Anal. Appl., 70 (1979), pp. 460–473.

[11] F. W. J. Olver, *Asymptotics and Special Functions*, A. K. Peters, Wellesley, MA, 1997.

[12] G. C. Rota, *Finite Operator Calculus*, Academic Press, New York, 1975.

[13] R. Sprugnoli and A. Del Lungo, *Semireal Stirling Numbers of the Second Kind*, Technical report, Dipartimento di Sistemi e Informatica, Università di Firenze, Italy, 1994.

[14] E. T. Whittaker and G. N. Watson, *A Course of Modern Analysis*, 4th ed., Cambridge Math. Library, Cambridge University Press, Cambridge, UK, 1996.

SIAM J. DISCRETE MATH.
Vol. 16, No. 2, pp. 192–207

MINIMIZING A CONVEX COST CLOSURE SET[*]

DORIT S. HOCHBAUM[†] AND MAURICE QUEYRANNE[‡]

Abstract. Many applications in the area of production and statistical estimation are problems of convex optimization subject to ranking constraints that represent a given partial order. This problem, which we call the convex cost closure (CCC) problem, is a generalization of the known maximum (or minimum) closure problem and the isotonic regression problem. For a CCC problem on n variables and m constraints we describe an algorithm that has the complexity of the minimum cut problem *plus* the complexity of finding the minima of up to n convex functions. Since the CCC problem is a generalization of both minimum cut and minimization of n convex functions, this complexity is the fastest one possible. For the quadratic problem the complexity of our algorithm is strongly polynomial, $O(mn \log \frac{n^2}{m})$. For the isotonic regression problem the complexity is $O(n \log U)$ for U the largest range for a variable value.

Key words. closure problem, nonlinear costs, Bayesian estimation, maximum flow, parametric minimum cut, convex optimization

AMS subject classifications. 68R10, 90C27, 90C30

PII. S0895480100369584

1. Introduction.

A common problem in statistical estimation is that observations do not satisfy preset ranking order requirements. In that case the problem is to find an adjustment of the observations that fits the ranking order constraints and minimizes the total deviation penalty. The deviation penalty is a convex function of the fitted values.

Formally, we define the problem for a directed graph $G = (V, A)$ and a convex function $f_j()$ associated with each node $j \in V$. The formulation of the convex cost closure (CCC) problem is then

$$\begin{array}{lll} \text{(CCC)} & \text{Min} & \sum_{j \in V} f_j(x_j) \\ & \text{subject to} & x_i - x_j \geq 0 \quad \forall (i,j) \in A, \\ & & \ell_j \leq x_j \leq u_j \;\; \text{integer} \quad j \in V. \end{array}$$

This problem generalizes the isotonic regression problem in which the graph is a partial order graph for linear order—the arcs of A are of the form $(i, i+1)$. Another well-known problem that CCC generalizes is the minimum closure problem. That problem is the binary case of CCC:

$$\begin{array}{lll} \text{(Minimum Closure)} & \text{Min} & \sum_{j \in V} w_j \cdot x_j \\ & \text{subject to} & x_i - x_j \geq 0 \quad \forall (i,j) \in A, \\ & & 0 \leq x_j \leq 1 \;\; \text{integer} \quad j \in V. \end{array}$$

[*] Received by the editors February 29, 2000; accepted for publication (in revised form) September 18, 2002; published electronically February 20, 2003. An extended abstract of this work appeared in *Algorithms—ESA* 2000, (Saarbrücken), Lecture Notes in Comput. Sci. 1879, Springer, Berlin, pp. 256–267.

http://www.siam.org/journals/sidma/16-2/36958.html

[†] Department of Industrial Engineering and Operations Research and Walter A. Haas School of Business, University of California, Berkeley, CA 94720 (hochbaum@ieor.berkeley.edu). This author's research was supported by NSF grants DMI-0085690 and DMI-0084857.

[‡] Faculty of Commerce and Business Administration, University of British Columbia, Vancouver, British Columbia V6T1W5, Canada (quey@commerce.ubc.ca).

Picard established in 1976 [17] that the closure problem is equivalent to a minimum cut problem on a graph associated with G to which we add a source and a sink. This construction is described in section 3. Solving the CCC problem is thus at least as hard as solving the minimum cut problem on the associated graph.

When the graph is empty the CCC problem reduces to the integer minimization of n convex functions, each in a given interval. Thus CCC generalizes the problem of convex functions integer minimization in bounded intervals.

The challenge of the convex optimization problem is that searching for a minimum of a convex function involves an unavoidable factor such as $\log U$ in the running time for U the length of the interval containing the optimal value of the variable. Although one can replace other parameters that depend on the variability of the functions, the running time cannot be made strongly polynomial using the arithmetic complexity model (see [11] for details on this result). The algorithm presented here differs from previous algorithms in that the search for the minima of the convex functions is separate from the rest of the algorithm. The main body of the algorithm identifies disjoint intervals that are guaranteed to contain the optimal values of each variable and satisfy the partial order constraints. The run time of our algorithm, $O(mn \log \frac{n^2}{m} + n \log U)$ for $U = \max_i \{u_i - \ell_i\}$, is the fastest known for the problem, and it either matches or improves the complexity of algorithms devised for special cases of CCC.

In dealing with nonlinear functions it is necessary to specify the complexity model used. We assume the unit cost model and no restriction on the structure of the convex functions; i.e., we assume the existence of an oracle returning function values for every polynomial length argument input in $O(1)$. Since we will search for an optimal solution among the integers we will be interested only in integer arguments. Any arithmetic operation or comparison involving function values is executed in unit time in this model. Derivatives, or rather subgradients, are required as finite differences, $f'_j(x) = f_j(x+1) - f_j(x)$.

In the next section we give an overview of the literature and applications of the problem. In section 3 the link of the closure problem to the maximum flow problem is reviewed. Section 4 discusses a linear time algorithm that is employed to verify whether an instance of CCC is feasible. Section 5 provides the main theoretical underpinning of the algorithm, the so-called *threshold theorem*. Section 6 describes the entire algorithm and its correctness and complexity. Section 7 details the implementations in strongly polynomial time for the quadratic case and the $O(n \log U)$ for the isotonic regression problem. In Section 8 we provide an algorithm for the continuous version of CCC. In Section 9 we conclude with several remarks and extensions.

2. Related applications and literature. We sketch first several classes of applications for the CCC problem.

In the problem of *selection of discrete contingent projects* a number of projects $j \in N$ can be undertaken, but only at discrete levels $l_j, l_j + 1, \ldots, u_j$. These projects are contingent in that, for specified pairs of projects $(i, j) \in A$, each unit of project i requires one unit of project j: $x_i \leq x_j$. Different projects i, k, …, however, can use the same units of project j (otherwise, we might consider j as part of project i). For instance, projects i and k may use j at different times. The objective is to maximize the total net profit associated with the selected project levels. Here, $-f_j(x_j)$ denotes the net profit associated with level x_j of project j. The convexity of $-f_j$ thus reflects decreasing returns of scale for project j. Additional information about this problem is provided by Picard and Queyranne [18].

Maxwell and Muckstadt [16] considered *nested power-of-two policies* in a *multistage production/inventory* problem. In this continuous-time deterministic model, demand for end-products arises at a constant rate. Intermediate products are consumed in the production of other products, as reflected by the directed graph (V, A). Given are positive inventory-related holding costs g_j and production setup costs K_j. The problem is to find production intervals $T_j = T_0 2^{k_j}$, with k_j integer, that are *nested*, that is, $T_i \leq T_j$ for $(i, j) \in A$. The objective is to minimize the average total cost per unit time,

$$c(T) = \sum_j g_j T_j + K_j / T_j.$$

Roundy [20] extends the Maxwell–Muckstadt model by considering joint setup costs and relaxing the nestedness condition. He shows that the total cost is now

$$c(T) = \sum_R G_R \max\{T_j : j \in R\} + \sum_F K_F / (\min\{T_j : j \in F\}),$$

where R and F are suitably defined subsets of products, and G_R and K_F are corresponding (nonnegative) costs. Although the constraints $T_i \leq T_j$ thus disappear, the modeling capabilities of variable upper bound constraints are reflected in the handling of joint setup costs and holding costs. For that, Roundy extends the product set N by adding all the R and F sets and "defines" corresponding variables T_R (T_F, respectively) by the inequalities $T_j \leq T_R$ ($T_F \leq T_j$, respectively) for all $j \in R$ (F, respectively). The resulting problem is thus recast into the Maxwell–Muckstadt form above. Roundy's major result is that *optimal power-of-two policies* thus constructed are 94% *effective*; that is, the cost of an optimal policy cannot be less than 94% of an optimum power-of-two policy. He also shows that searching for an optimal base interval T_0 yields a 98% *effective* solution. The present paper extends this approach to general convex average cost functions $f_j(k_j) = c_j(T_0 2^{k_j})$. The 94% and 98% effectiveness results, however, hold only for the specific functions c above.

Sokkalingam, Ahuja, and Orlin [23] discuss the *inverse spanning tree* problem. In this problem there is a spanning tree T given in an edge weighted graph. The problem is to modify the edge weights so that the given tree is a minimum spanning tree and the cost of the deviation is minimum. In order for a tree to be a minimum spanning tree each out-of-tree edge j must have a weight w_j greater than or equal to each of the edges in the tree on the unique path between its endpoints. That is, the constraints enforcing that T be a minimum spanning tree are of the form $w_j \geq w_i$ for $j \in E \setminus T$, $i \in T$. The corresponding graph is a bipartite graph—a structure that can be used to reduce the complexity of our algorithm for the resulting CCC. Sokkalingam, Ahuja, and Orlin devised algorithms for three specific convex deviation functions: sum of absolute differences, weighted sum of absolute differences, and maximum absolute differences. All these functions are convex for which the minima can be found in a single step. Our algorithm's run time is thus strongly polynomial and is better than $O(mn \log^2 n)$ for this type of function, and with an additional additive factor of $n \log C$ for general convex functions, where C is the maximum edge weight. The complexity reported in [23] for the weighted absolute deviation problem is weakly polynomial $O(n^2 m \log(nC))$. Our algorithm can be further adapted to provide substantial improvements for special cases as reported in [14].

Statistical problems of *partially ordered estimation* have been discussed extensively in the literature; see, e.g., Veinott [24] and Barlow et al. [4]. Let $p_1, \ldots, p_n$

denote parameters to be jointly estimated and let $f_j(x_j)$ denote the *loss* associated with estimating that $p_j = x_j$ for $j = 1, \dots, n$. A typical instance is when $f_j(x_j)$ is the negative of the logarithm of the likelihood, given $p_j = x_j$, that related random variables assume observed values. The model being estimated may specify a *partial order* on the parameters, as reflected by constraints $x_j \le x_i$ for a set A of pairs (i, j), as well as simple upper and lower bounds on the parameter values. If, in addition, the model requires the parameter values to be integer, then the problem of jointly estimating the parameter values so as to minimize total loss is precisely an instance of the CCC problem. If there is no such integrality restriction, then the problem is an instance of the continuous relaxation of CCC, which is discussed in section 8.

Algorithms for the CCC problem have been previously devised. Picard and Queyranne [19] proposed an algorithm solving the problem with a running time of $O(n(mn \log \frac{n^2}{m} + n \log U))$. Ahuja, Hochbaum, and Orlin [3] addressed a generalization of CCC—a convex cost dual of minimum cost network flow. Their algorithm for this convex cost dual of minimum cost flow has running time of $O(mn \log \frac{n^2}{m} \log(nU))$.

The method of Hochbaum and Naor [9] solves integer problems on monotone inequalities in at most two variables per inequality. A monotone inequality is of the form $ax - by \le c$, where the coefficients of x and y are of opposite signs. Obviously, the constraints of CCC are monotone inequalities. The algorithm of Hochbaum and Naor runs in pseudopolynomial time $O(\sum_i (u_i - \ell_i) mU \log \frac{n^2 U}{m})$. It is possible to combine that algorithm with a scaling approach implemented for CCC in time $O(mn \log U \log \frac{n^2 U}{m})$; see [2]. Hochbaum [12] generalized the concept of monotone inequality to include three variables, $ax - by \le c + z$, for $a, b \ge 0$. The run time for solving integer programming on such inequalities was shown in [12] to be solved in the same time as the algorithm in [2].

The algorithm described here solves CCC in time $O(mn \log \frac{n^2}{m} + n \log U)$. The first term in the complexity expression is the run time required to solve the minimum closure problem, and the second factor is the run time required to find the integer minima of n convex functions. Since CCC generalizes both these problems, as discussed above this is the best complexity achievable for CCC. It is likely that if a faster algorithm for the minimum closure problem is discovered, then the run time of the algorithm can be respectively improved. For the second term the factor of $\log U$ cannot be avoided, as any algorithm solving a constrained nonlinear and nonquadratic optimization problem may not run in strongly polynomial time [11]. When the functions f_i are quadratic convex, the algorithm runs in strongly polynomial time $O(mn \log \frac{n^2}{m} + n \log n)$. For the isotonic regression problem the running time improves to $O(n \log n + n \log U)$, and thus the complexity of our algorithm is $O(n \log(\max\{n, U\}))$.

There are efficient algorithms known for solving several special cases of CCC. Any maximum flow algorithm can be used to solve the minimum (or maximum) closure problem. The most efficient algorithm known to date, due to Goldberg and Tarjan [8], solves the maximum flow and thus the minimum cut, and the closure problems have complexity of $O(mn \log \frac{n^2}{m})$.

The isotonic regression problem is an instance of CCC defined on a linear order. Ahuja and Orlin report on an $O(n \log U)$ time algorithm for this problem [1]. The problem has been reviewed extensively in the statistical study of observations. Barlow et al. [4] provide an excellent review of applications and algorithms for the isotonic regression problem as well as the CCC problem.

3. Solving the minimum closure as a minimum cut problem. Recall that the minimum closure problem is a special case of CCC attained by setting the variables to be binary. We review here the procedure for solving the minimum closure problem with a minimum cut algorithm. Although CCC is a problem far more general than the minimum closure, our algorithm for CCC also solves the minimum closure problem in the most efficient complexity known.

A set of nodes $S \subseteq V$ in a directed graph $G = (V, A)$ is said to be *closed* if all predecessor nodes of S are also included in S; i.e., if $j \in S$ and $(i, j) \in A$, then $i \in S$. Equivalently, S is said to be closed if it has no incoming arcs.

We review here the reduction of Picard [17], demonstrating that the minimum closure problem is solved using a minimum cut procedure. We first define an s, t-graph that contains a source and a sink and that is associated with the minimum closure problem: the graph has a node j associated with each variable x_j. A source, and sink nodes s and t, are now added to the graph. If the weight of the variable w_j is positive, then node j has an arc from the source into it with capacity w_j. If the node has weight w_j which is negative, then there is an arc from j to t with capacity $-w_j$. Let V^+ be the set of nodes with positive weights and V^- be the set of nodes with negative weights.

Each inequality $x_i \geq x_j$ is associated with an arc (i, j) of infinite capacity. Consider any finite s, t-cut in the graph that partitions the set of nodes into two subsets commonly referred to as the *source set* of the cut and the *sink set* of the cut, $\{s\} \cup S$ and $\{t\} \cup \bar{S}$. It is easy to see that $\bar{S}$ is a closed set if there are no infinite capacity arcs from S to $\bar{S}$.

We denote by (A, B) the collection of arcs with tails at A and heads at B. The corresponding sum of capacities of these arcs is denoted by $C(A, B)$, $C(A, B) = \sum_{i \in A, j \in B} c_{ij}$, where c_{ij} is the capacity of arc (i, j) . Let $w(A) = \sum_{j \in A} w_j$.

Given a finite cut $(\{s\} \cup S, \bar{S} \cup \{t\})$, we have

$$\begin{aligned}\min_{\bar{S} \subseteq V}[C(\{s\} \cup S, \bar{S} \cup \{t\})] &= \min_{\bar{S} \subseteq V} \textstyle\sum_{j \in \bar{S} \cap V^+} w_j + \sum_{j \in S \cap V^-} (-w_j) \\ &= \min_{\bar{S} \subseteq V} \textstyle\sum_{j \in \bar{S} \cap V^+} w_j - (\sum_{i \in V^-} w_i - \sum_{i \in \bar{S} \cap V^-} w_i) \\ &= \min_{\bar{S} \subseteq V} \textstyle\sum_{j \in \bar{S}} w_j - w(V^-).\end{aligned}$$

In the last expression the term $w(V^-)$ is a constant. Thus the closed set $\bar{S}$ of minimum weight is also the sink set of a minimum cut and vice versa—the sink set of a minimum cut (without t), which has to be finite, also minimizes the weight of the closure.

4. Verifying feasibility in linear time. We define a graph associated with CCC that has one node representing each variable in the problem. We let the set of nodes be denoted by V. Each inequality $x_i \geq x_j$ is associated with an arc (i, j). We let the set of arcs be denoted by A. If the directed graph (V, A) has strongly connected components, then each node in the strongly connected component shares a directed cycle with each of the other nodes in the strongly connected component, and thus the values of the corresponding variables must be equal.

Finding the strongly connected components of a graph can be accomplished in $O(m)$ time; see, e.g., [6, Chap. 23]. The strongly connected components partition the nodes of the graph into $V_1 \cup \cdots \cup V_k$. In each strongly connected component V_i we let $\ell(V_i)$ be the tightest lower bound in V_i and let $u(V_i)$ be the tightest upper bound in V_i. That is,

$$\ell(V_i) = \max_{v \in V_i} \ell_v \quad \text{and} \quad u(V_i) = \min_{v \in V_i} u_v.$$

A necessary condition for feasibility is that all variables in the same strongly connected component assume the same value that falls in the interval range $[\ell(V_i), u(V_i)]$. Since the above recursion is performed in linear time, verifying this necessary condition amounts to checking that $\ell(V_i) \leq u(V_i)$ in $O(m+n)$ steps.

We now consolidate each strongly connected component into a single node V_i defined on the interval $[\ell(V_i), u(V_i)]$. The function associated with such a node (or variable) is the sum of the convex functions associated with all nodes in V_i, which is a convex function. The graph of strongly connected components is thus a directed acyclic graph (DAG).

Let V_i be a predecessor of V_j in different strongly connected components. Then the following updates are valid:

$$\ell(V_i) \leftarrow \max\{\ell(V_j), \ell(V_i)\}, \qquad u(V_j) \leftarrow \min\{u(V_i), u(V_j)\}.$$

All these updates can be performed in time $O(m)$. A necessary condition for feasibility is that for $i = 1, \ldots, k$, $\ell(V_i) \leq u(V_i)$. This condition is also sufficient since, if satisfied, there exists a feasible solution which is, say, to set all variables to the lower bounds of their corresponding intervals.

Our optimization algorithm runs faster if the feasibility preprocessing step is performed and the interval bounds are adjusted. This preprocessing step, however, is not essential and does not affect the worst case complexity.

5. The threshold theorem.

The threshold theorem is the cornerstone of our algorithm. The theorem is an extension of an earlier result of Picard and Queyranne [19].

Let α be a scalar in the interval $(\ell, u) = (\min_{i\in V} \ell_i, \max_{i \in V} u_i)$. Consider further the convex extension of the functions $f_i()$ on the real line by setting $f_i'(x)$ to be equal to M at values of $x > u_i$, and to $-M$ for values $x < \ell_i$, for M a suitably large value. We will comment after the statement of the theorem on how large M should be. The functions $f_i()$ are therefore defined for any real value x as follows:

$$f_i(x) = \begin{cases} f_i(u_i) + M(x - u_i) & \text{if } \ x > u_i, \\ f_i(x) & \text{if } \ \ell_i \leq x \leq u_i, \\ f_i(\ell_i) + M(\ell_i - x) & \text{if } \ x < \ell_i. \end{cases}$$

Consider now the minimum closure problem with variable weights $w_i = w_i(\alpha)$ that are the subgradients of f_i at α, $w_i = f_i'(\alpha) = f_i(\alpha+1) - f_i(\alpha)$. The theorem establishes that all elements i in the minimum weight closed set S^* satisfy that for any optimal solution $\mathbf{x}$, $x_i > \alpha$, and satisfy that for all elements j in the complement of S^*, $x_j \leq \alpha$. Consequently, the theorem allows for the reduction of CCC to a sequence of minimum closure problems.

In case there are several optimal minimum closed sets, we define a *minimal* minimum closed set as a minimum closed set that does not contain other minimum closed sets. Similarly, a *maximal* minimum closed set is defined as a minimum closed set that is not contained in another minimum closed set.

THEOREM 5.1. *Let $w_i = f_i'(\alpha)$ be the weight assigned to node i, $i = 1, \ldots, n$, in a minimum closure problem defined on the directed graph $G = (V, A)$. Let S^* be the minimal minimum weight closed set in this graph. Then an optimal solution $\mathbf{x}^*$ to the CCC problem satisfies $x_i^* > \alpha$ if $i \in S^*$ and $x_i^* \leq \alpha$ if $i \in \bar{S}^*$.*

Proof. The proof is by contradiction. Let S^* be the minimal minimum weight closed set, and suppose there is a nonempty subset $S^o \subseteq S^*$ such that at an optimal solution $\mathbf{x}^*$, $x_j^* \leq \alpha$ for all $j \in S^o$.

Since at the optimum $x_j^* > \alpha$ for $j \in S^* \setminus S^o$, the set $S^* \setminus S^o$ must be closed, as it has no predecessors (larger values) in S^o. But this set is not a minimal minimum closed set, as S^* is minimal. Thus the weight of nodes in S^o—the total sum of subgradients—must be negative, $\sum_{j \in S^o} f_j'(\alpha) < 0$. Furthermore, increasing the values of all x_j^* in this set to $\alpha + \epsilon \leq \min_{i \in S^* \setminus S^o} x_i^*$ for some $\epsilon > 0$ does not violate feasibility, since the values of their predecessors in $S^* \setminus S^o$ are all $\geq \alpha + \epsilon$. Thus replacing x_j^* for $j \in S^o$ by α is feasible and strictly reduces the weight of the closure compared to an optimal solution. This contradicts the assumption that $\mathbf{x}^*$ is optimal.

An analogous contradiction is reached if we assume that an optimal solution has in the set $\bar{S}^*$ a variable with value $> \alpha$. □

As a result of the theorem, we can decompose the set of nodes into subsets that imply a narrowing of the interval in which the optimal value of the respective variable is to be found: For a given value of α we solve the minimum closure problem with $w_i = f_i'(\alpha)$ for a minimal minimum closure S^*. For all $i \in S^*$ we conclude that $x_i^* \in (\max\{\ell_i, \alpha\}, u_i]$ and for all $j \in \bar{S}^*$, $x_j^* \in [\ell_j, \min\{\alpha, u_j\}]$.

Concerning the value of M, it is sufficient to set $M = \sum_i \max\{f_i'(u_i), |f_i'(\ell_i)|\}$. We claim that for a feasible problem a node with weight M is never in a minimum weight closed set, and a node with weight $-M$ is always in a minimum weight closed set. If that were not the case, then either we can generate a closed set of a strictly lower value by including nodes of weight $-M$ and excluding nodes of weight M or else there is a node j of weight $-M$ that has as its predecessor a node i of weight M. But that means that the given value of α satisfies $u_i < \alpha < \ell_j$, and there is no feasible solution where the value of $x_i \in [\ell_i, u_i]$ is at least as large as the value of $x_j \in [\ell_j, u_j]$.

Indeed, the algorithm we employ to solve CCC can be used to verify feasibility as well—for every value of α the nodes of weight M must be in the source set and the nodes of weight $-M$ must be in the sink set or else the problem is infeasible. Yet, if the feasibility test of section 4 is used, then whenever the threshold theorem is invoked in the algorithm the nodes of weight $-M$ are known a priori to be in the minimum closure and thus in the sink set, and those nodes of weight M are known to be in the source set. This permits the "shrinking" of nodes of weight M with the source and nodes of weight $-M$ with the sink. The size of the graph is thus reduced and the value of M is not explicitly used if the feasibility test is invoked as a preprocessing step.

6. The algorithm. One obvious method of using the threshold theorem for solving CCC is to perform a search by calling for the solution of the minimum closure problem for all integer values of α in the interval (ℓ, u). When done, the output of such a process is a partition of the set of variables V into q sets, and the interval into q disjoint intervals, so that all variables in the same set have their optimal values in the same interval. The goal would be to find for each variable x_j the largest value of α for which it is still in the source set and to find the smallest value of α for which it is no longer in the source set. With this information we narrow down the value of x_j at an optimal solution to an interval defined by these values. We later show that once these intervals are identified, all variables assigned to the same interval assume the same value in that interval, and that value is the lower end of the interval. One drawback of the approach just described is that it makes U calls to a minimum cut procedure and is thus pseudopolynomial.

It is easy to see that a binary search type approach could be used to implement the procedure of identifying the intervals to a polynomial time procedure. Next we show that one can do still better by implementing the process of identifying the set and interval partitioning in strongly polynomial time and in the complexity of solving a single minimum cut problem.

The key to our approach is to utilize *parametric minimum cut* to generate all the breakpoints of the decompositions. This can be done, as is shown here, by adapting the method of Gallo, Grigoriadis, and Tarjan [7], which works in the same running time as a single minimum cut procedure.

6.1. The parametric graph G_λ. We create a graph with parametric capacities, $G_\lambda = (V \cup \{s,t\}, A)$. Each node $j \in V$ has an incoming arc from s with capacity $\max\{0, f_j'(\lambda)\}$ and an outgoing arc to the sink t with capacity $-\min\{0, f_j'(\lambda)\}$. The capacities of the arcs adjacent to the source in this graph are monotone nondecreasing as a function of λ, and the arcs adjacent to the sink have capacities that are monotone nonincreasing as a function of λ. Note that each node is connected with a positive capacity arc, either to source, or to sink, but not to both. Denote the source set of a minimum cut in the graph G_λ by S_λ.

Restating the threshold theorem in terms of the corresponding minimum cut for the graph G_λ associated with the closure graph, any optimal solution $\mathbf{x}$ satisfies that $x_j > \lambda$ for $j \in \bar{S}_\lambda$ and $x_j \leq \lambda$ for $j \in S_\lambda$, where S_λ is the maximal source set of a minimum cut.

Let ℓ be the lowest lower bound on any of the variables and u the largest upper bound. Consider varying the value of λ in the interval $[\ell, u]$. As the value of λ increases, the sink set becomes smaller and contained in the previous sink sets corresponding to smaller values of λ, specifically, for some $\lambda \leq \ell$ $S_\lambda = \{s\}$ and some $\lambda \geq u$ $S_\lambda = V \cup \{s\}$. We call each value of λ, where S_λ strictly increases, a *node-shifting breakpoint*. For $\lambda_1 < \cdots < \lambda_\ell$ the set of all node-shifting breakpoints we get a corresponding nested collection of source sets,

$$\{s\} = S_{\lambda_1} \subset S_{\lambda_2} \subset \cdots \subset S_{\lambda_\ell} = \{s\} \cup V.$$

Our goal is to partition the variables into the subsets $S(k) = S_{\lambda_k} - S_{\lambda_{k-1}}$, $k = 2, \ldots, \ell$. The property of each subset $S(k)$ is that all variables in the set have optimal value in the interval $(\lambda_{k-1}, \lambda_k]$. As we prove next, the optimal value of all variables in $S(k)$ is x^*, where

$$x^* = \lambda_{k-1} + 1.$$

LEMMA 6.1. *For $j \in S(k)$, the value of x_j at an optimal solution, x_j^*, is $\lambda_{k-1}+1$.*

Proof. According to the threshold theorem, λ_{k-1} is the largest value so that for $j \in S(k)$, $x_j > \lambda_{k-1}$.

It follows that $x_j = \lambda_{k-1} + 1$. □

6.2. Identifying an integer node-shifting breakpoint. Since we are interested only in integer valued solutions, we can consider the convex functions $f_j(x)$ to be piecewise linear segments connecting the values of $f_j(k)$ on integer points $\ell_j \leq k \leq u_j$. For such functions the derivatives at the integer points are not well defined and indeed could be any subgradient of the function at the respective integer point. We will consider the derivative $f_j'(x)$ to be a step function with the value in the interval $(k-1, k]$ equal to $f_j(k) - f_j(k-1)$.

We denote a maximal minimum cut source set in G_λ by $S_\lambda^{\max}$ and a minimal minimum cut source set by $S_\lambda^{\min}$.

The source set of a minimum cut of G_λ remains invariant for $\lambda \in (k-1, k]$. Thus, in order to verify that λ is a node-shifting breakpoint, it suffices to compare $S_\lambda^{\max}$

with $S^{\min}_{\lambda+\epsilon}$ for $\epsilon > 0$ sufficiently small. In our case we consider only integer values of λ, and $\epsilon = 1$ is a small enough value. So if $S^{\max}_{\lambda} \subset S^{\min}_{\lambda+1}$, then λ is a node-shifting breakpoint.

The existence of a breakpoint in an interval (λ_1, λ_2) is confirmed if and only if $S^{\max}_{\lambda_1} \subset S^{\min}_{\lambda_2}$.

6.2.1. Parametric analysis. Gallo, Grigoriadis, and Tarjan [7] devised a complete parametric analysis algorithm using the push-relabel algorithm that runs in the same time as a single push-relabel algorithm and identifies all node-shifting breakpoints. The algorithm is applicable to graphs with source adjacent arcs having capacities monotone nondecreasing in the parameter λ and sink adjacent arcs having capacities nonincreasing in λ. The running time of the algorithm for linear capacity functions is $O(mn \log \frac{n^2}{m})$. The same result is achieved using the pseudoflow algorithm [13] with a running time of $O(mn \log n)$. We let the generic run time be Qmn, where Q is a constant times $\log \frac{n^2}{m}$ for push-relabel and a constant times $\log n$ for pseudoflow. Whenever we refer in the analysis below to a minimum cut algorithm it can be either the push-relabel algorithm or the pseudoflow algorithm (and its variants). Other minimum cut algorithms do not satisfy the necessary requirements to make them amenable to the analysis of the parametric procedure.

We assume henceforth that the procedure **min-cut**(G_λ) returns both the minimal and maximal source sets of minimum cuts (if different), $S^{\min}_{\lambda}$, $S^{\max}_{\lambda}$, and $S^{\min}_{\lambda+1}$. The procedure also returns the state of the graph at the end of the run, which includes node labels and preflows for the push-relabel algorithm and node labels and pseudoflows for the pseudoflow algorithm.

For a given interval (λ_1, λ_2) we can find all node-shifting breakpoints by using the procedure **parametric**. The input to the procedure includes R_1 and R_2, which are runs of the minimum cut algorithm that are initiated on an s,t-graph G and the reverse graph G^R, respectively.

```
    **Procedure parametric** $(G, \lambda_1, \lambda_2, S^{\max}_{\lambda_1}, S^{\min}_{\lambda_2}, R_1, R_2)$.
Contract in $G$: $s \leftarrow s \cup S^{\max}_{\lambda_1}$, $t \leftarrow t \cup \bar{S}^{\min}_{\lambda_2}$. If $V = \{s,t\}$, or, if $\lambda_2 - \lambda_1 \leq 1$, halt
    "no breakpoints."
Else, let $\lambda^* = \lfloor \frac{\lambda_1+\lambda_2}{2} \rfloor$.
Call **min-cut**$(G_{\lambda^*}, R_1, R_2)$ for the output $S^{\min}_{\lambda^*}$, $S^{\max}_{\lambda^*}$, and $R^*$.
If $\lambda^*$ is a breakpoint, output $\lambda^*$ and $S^{\min}_{\lambda^*}$.
    Call **parametric** $(G, \lambda_1, \lambda^*, S^{\max}_{\lambda_1}, S^{\min}_{\lambda^*}, R_1, R^*)$.
    Call **parametric** $(G, \lambda^*, \lambda_2, S^{\max}_{\lambda^*}, S^{\min}_{\lambda_2}, R^*, R_2)$.
**end**
```

The choice of λ^* as the median in the interval (λ_1, λ_2) leads to an additional run time of $O(n \log(\lambda_1 - \lambda_2))$, where n is the number of adjusted capacity functions. For specific capacity functions λ^* is replaced by the intersection of the two cut capacity functions.

The analysis of the complexity of the procedure follows arguments used in [7].[1] It is essential that the algorithm used in the runs for minimum cut satisfies the following properties:

[1] The source of some of this analysis is from private communication of the first author with R. Tarjan in 1996.

- *Reflectivity*: The complexity of the algorithm remains the same whether run on the graph or reverse graph.
- *Monotonicity*: Running the algorithm on a monotone sequence of parameter values has the same complexity as a single run.

The main recursive procedure is **min-cut**(G_{λ^*}, R_1, R_2), where R_1 is the status of the graph (labels assigned to nodes and flow values) G_{λ_1} after a minimum cut was identified and R_2 is the state of the graph after the minimum cut was found on the reverse graph $G^R_{\lambda_2}$.

The procedure is implemented as follows: Run a maximum flow algorithm on G_{λ^*} as a monotone continuation of the run R_1. Concurrently, run a maximum flow algorithm on $G^R_{\lambda^*}$ as a monotone continuation of the run R_2. Suppose that the algorithm for the forward direction (on G) stops first (the other case is symmetric). If $|S^{\min}_{\lambda^*}| > n/2$, complete the execution of the maximum flow algorithm on G^R and let R^* be the resulting state of the graph.

Consider the execution that follows immediately of the recursive calls to **parametric** (G, λ_1, λ^*, $S^{\max}_{\lambda_1}$, $S^{\min}_{\lambda^*}$, R_1, R^*), and to **parametric** (G, λ^*, λ_2, $S^{\max}_{\lambda^*}$, $S^{\min}_{\lambda_2}$, R^*, R_2). Consider graphs $G(\bar{S}^{\min}_{\lambda^*})$ and $G(S^{\max}_{\lambda^*})$ on which **min-cut** is called recursively. Let R_3 and R_4, respectively, be the forward and backward runs on $G(\bar{S}^{\min}_{\lambda^*})$ when **min-cut** is applied. Let R_5 and R_6, respectively, be the forward and backward runs on $G(S^{\max}_{\lambda^*})$ when **min-cut** is applied. We distinguish two cases.

Case 1. If $n_1 > n/2$, we regard R_4 as a continuation of R_2, and regard R_3 as a restart of R_1, that is, as a continuation of the run of which R_1 was a continuation. We must charge for R_5 and R_6 as starts of new runs. The $2Qm_1n_1$ term in the recurrence for $T(m,n)$ below accounts for the new runs of the push-relabel algorithm that begin with R_5 and R_6.

Case 2. Symmetrically, if $n_1 \leq n/2$, we regard R_5 as a continuation of R_1, and regard R_6 as a restart of R_2. In this case, the $2Qm_1n_1$ term in the recurrence for $T(m,n)$ below accounts for the new runs that begin with R_3 and R_4.

In Case 1, we still must account for the cost of R_1. In Case 2, we still must account for the cost of R_2. Procedure **min-cut** runs R_1 and R_2 concurrently, stopping when the first one stops. Suppose R_1 stops first. Then the cost of R_1 is covered by the cost of R_2, which takes care of Case 1. Note that in this case R_2 is run to completion, even though it takes longer than R_1 (see implementation); R_1 is the abandoned run, but it is cheaper than R_2, which is the good run. Suppose R_1 stops first but we are in Case 2. Although run R_2 is abandoned, we have spent no more time on it than the time spent running R_1, which was a good run. In this case, the run of R_1 covers the time spent on (partially) running R_2. The situation is symmetric if R_2 stops first. In every case the time spent on the completed good run is at least as much as the time spent on partially or completely performing the run that is abandoned.

Throughout the procedure the total complexity of abandoned runs is at most the complexity of one run with monotonically increasing (or decreasing) parameter values. In addition, the total work for good runs on G_{λ^*} and $G^R_{\lambda^*}$ is at most twice the complexity of one run on monotone parameter values. The total complexity charged for these runs is at most that of three runs of the minimum cut algorithm, $3Qmn$.

Let $m_1 + m_2 \leq m$, $n_1 + n_2 \leq n$, and $n_1 \leq \frac{1}{2}n$. The running time $T(m,n)$ is the additional running time required by the algorithm, taking into account the new runs initiated with each recursive call to **min-cut**. Let Q be a constant. Then

$$T(m,n) = T(m_1,n_1) + T(m_2,n_2) + 2Qm_1n_1.$$

The solution to the recursion is $T(m,n) = Qmn$. Thus the overall run time of the parametric procedure with the push-relabel algorithm is $O(mn \log \frac{n^2}{m})$. The run time incurred in adjusting capacities is $O(n \log U)$ throughout the procedure.

6.3. The algorithm. Let ℓ be the lowest lower bound on any of the variables and u be the largest upper bound. Let $U = u - \ell$.

PROCEDURE CONVEX COST CLOSURE $(G, f_j, j = 1, \ldots, n)$.

Step 1: Call **parametric** $(\ell, u, \emptyset, V)$.

Let the output be a set of up to n breakpoints $\lambda_1, \lambda_2, \ldots, \lambda_\ell$ and the corresponding sets of source sets of minimum cuts $S_1 \subset S_2 \cdots \subset S_\ell$.

Step 2: Output the optimal solution $\mathbf{x}^*$ where for $j \in S_k - S_{k-1}, x_j^* = \lambda_{k-1} + 1$.

The complexity of the algorithm is $O(mn \log \frac{n^2}{m} + n \log U)$.

7. Special cases.

7.1. The quadratic CCC problem. Nonlinear and nonquadratic optimization problems with linear constraints were proved impossible to solve in strongly polynomial time in a complexity model of the arithmetic operations, comparisons, and the rounding operation [11]. That negative result, however, is not applicable to the quadratic case, and thus it may be possible to solve constrained quadratic optimization problems in strongly polynomial time. Yet, few quadratic optimization problems are known to be solvable in strongly polynomial time. For instance, it is not known how to solve the minimum quadratic cost network flow problem in strongly polynomial time. For the convex quadratic cost closure problem our result adds to the limited repertoire of quadratic problems solved in strongly polynomial time.

In the quadratic case our algorithm is implemented to run in strongly polynomial time. This is easily achieved since the derivative functions are linear—a case that is shown in [7] to be solved in $O(mn \log \frac{n^2}{m})$. Thus the overall run time of the algorithm is dominated by the complexity of the minimum cut,

$$O\left(mn \log \frac{n^2}{m}\right).$$

7.2. Isotonic regression. The isotonic regression problem is a special case of CCC in which the order is linear and the corresponding graph $G = (V, A)$ is a path from node n to node 1. In other words, the inequalities associated are of the type

$$x_i \leq x_{i+1} \quad \text{for all} \quad i = 1, \ldots, n.$$

There are only n possible cuts in such a graph, each with a source set S_i of the form $S_i = \{1, \ldots, i\}$. Each cut is thus $(\{1, \ldots, i\}, \{i+1, \ldots, n\})$. The minimum cut for such graphs is trivially identified in $O(n)$ time by comparing the capacities of the n possible cuts. The capacity of cut $(S_i, \bar{S}_i)$ is computed in $O(1)$ by subtracting from the capacity of $(S_{i-1}, \bar{S}_{i-1})$ the weight of node i, w_i. Indeed, if the weight w_i is positive, then it contributes w_i to the capacity of the cut $(S_{i-1}, \bar{S}_{i-1})$ but not to the capacity of the cut $(S_i, \bar{S}_i)$. If $w_i < 0$, then node i contributes $-w_i$ to the capacity of $(S_i, \bar{S}_i)$ but 0 to the capacity of $(S_{i-1}, \bar{S}_{i-1})$.

Consider the closure graph in which each node has a weight $f_j'(x)$ associated with it for a given value of x. Minimizing the value of the cut is equivalent to minimizing the sum of weights of the sink set (see section 3). Alternatively, the cut is minimized when the weight of the corresponding source set is maximized, thus seeking an index i to maximize $F_i(x) = \sum_{j=1}^{i} f'_j(x)$. We thus conclude with the following.

LEMMA 7.1. *If $\sum_{j=1}^{i} f'_j(x) = \max_{k=1,\ldots,n} \sum_{j=1}^{k} f'_j(x)$, then the minimum cut in the graph G_x is $(S_i, \bar{S}_i)$.*

Consider the partial sum functions

$$F_1(x), F_2(x), \ldots, F_n(x),$$

where $F_i(x) = \sum_{j=1}^{i} f'_j(x)$. Recall that the functions $f'_j(x)$ are monotone nondecreasing in x. Denote the roots of the partial sum functions by b_i. Thus $F_i(b_i) = 0$. If the function is negative in the interval $[\ell, u]$, then we let $b_i = u + 1$. If the function is positive throughout the interval, then we let $b_i = \ell - 1$. Let $b_{i_1} = \min_i b_i$. Then for $x \leq b_{i_1}$ the optimal minimum cut is $(\emptyset, V)$. For this cut, the maximum weight source set is empty since all the partial sums of weights are nonpositive. The value of $\lambda_1 = b_{i_1}$ is thus a breakpoint beyond which, for $x > b_{i_1}$, the source set of the minimum cut is $\{1, \ldots, i_1\}$.

As the value of x increases sufficiently so that $\sum_{j=i_1+1}^{i_2} f'_j(x) = F_{i_2}(x) - F_{i_1}(x) \geq 0$, the nodes $\{i_1, \ldots, i_2\}$ join the source set of the minimum cut. In other words, the second breakpoint is the smallest value λ_2 so that there is an index $i_2 > i_1$ such that

$$F_{i_2}(\lambda_2) - F_{i_1}(\lambda_2) \geq 0.$$

The general procedure is as follows:

PROCEDURE ISOTONIC REGRESSION BREAKPOINTS.

```
$i_0 = 0$, $\lambda_0 = \ell - 1$, $k = 1$
while $i_{k-1} < n$, do
Find smallest integer value of $\lambda_k$ such that for $i_k > i_{k-1}$, $F_{i_k}(\lambda_k) - F_{i_{k-1}}(\lambda_k) \geq 0$.
$k \leftarrow k + 1$
repeat
Output $\lambda_1, \ldots, \lambda_k$.
end
```

A naive implementation of this algorithm has n iterations with each iteration involving the finding of the roots of $O(n)$ functions. The total complexity is $O(n^2 \log U)$. We can do better with the implementation of the parametric search algorithm that requires the solution of the minimum cut problem for a specific parameter value in $O(n)$. However, each time the procedure calls for the minimum cut, the weights of the nodes must be updated for the new parameter value. This update requires $O(n)$ operation. The additional work of finding the roots of the n functions adds up to a total complexity of $O(n^2 + n \log U)$.

To achieve an even better running time we investigate further the properties of the partial sum functions $F_i(x)$. These functions are obviously monotone nondecreasing as sums of monotone nondecreasing functions. Another important property proved in the next lemma is that each pair of such functions intersects at most once.

LEMMA 7.2. *For $i < j$ and functions F_i and F_j, if for some value of the argument λ, $F_i(\lambda) < F_j(\lambda)$, then $F_i(x) < F_j(x)$ for any $x > \lambda$.*

Proof. $F_j(x) - F_i(x)$ is a sum of monotone nondecreasing functions $\sum_{k=i+1}^{j} f'_k(x)$. Thus the difference $F_j(x) - F_i(x) > F_j(\lambda) - F_i(\lambda) > 0$ and can increase only as the value of x grows. Thus the two functions do not intersect for any value of $x > \lambda$. □

An immediate corollary of the lemma is that any pair of functions F_i, F_j can intersect at most once.

Consider the upper envelope of the functions F_i represented as an array of functions and breakpoints $(\ell, 0, b_{i_1}, F_{i_1}, b_{i_2}, F_{i_2}, \ldots, b_{i_n}, F_{i_n}, u)$. The functions on the envelope have the property that for all j,

$$F_{i_k}(x) \geq F_j(x), \qquad x \in [b_{i_k - 1}, b_{i_k}].$$

From the lemma it follows that the upper envelope of the partial sums functions has at most n breakpoints, where the function on the envelope changes. The first breakpoint is b_{i_1}. The next breakpoint occurs for a value of x when some function $F_{i_2}(x) = F_{i_1}(x)$. It is easy to see from Procedure isotonic regression breakpoints that the list of breakpoints of this envelope is precisely the list of the breakpoints that determine the sequence of cuts.

The following *sweep* algorithm may be used to find the upper or upper envelope of a set of functions: Partition arbitrarily the set of functions into two equally sized sets $\mathcal{F}_1$, $\mathcal{F}_2$. Compute recursively the upper envelopes of $\mathcal{F}_1$, $\mathcal{F}_2$. Let E_1, E_2 denote the two resulting upper envelopes. *Sweep* the two upper envelopes E_1, E_2 from left to right and compute the upper envelope of the two upper envelopes. For a detailed description of the above algorithm the reader is referred to [22, pp. 134–136] and [5].

It remains to show how to implement the sweep algorithm for our particular set of functions. Instead of partitioning arbitrarily the set of functions, we choose the partition of $\mathcal{F}_1 = \{1, \ldots, n\}$ to $\{1, \ldots, \lfloor \frac{n}{2} \rfloor\}$ and $\mathcal{F}_2 = \{\lfloor \frac{n}{2} + 1, \ldots, n \rfloor\}$. That is, one set contains the lower half-set of indices and the other set contains the upper half-set of indices.

Consider the first breakpoint in E_1 and E_2 (recall that at that point the partial sum values are still 0). If the first breakpoint of E_1 is larger than the first breakpoint of E_2, then the first portion of E_1 is below the first portion of E_2. From the lemma we see that no pairs of functions from the two sets intersect, and the entire envelope E_1 lies below the envelope E_2. Thus the merged envelope is E_2.

If, on the other hand, the first breakpoint of E_1 is smaller than the first breakpoint of E_2, then there could be a point where a function from $\mathcal{F}_2$ crosses a function from $\mathcal{F}_1$. We consider the array of breakpoints of the envelope E_1 for the last breakpoint, where it is still above E_2. Similarly, we search the array of breakpoints of the envelope E_2 for the last breakpoint, where it is still below E_1. Since there are $O(n)$ breakpoints per envelope, the search for that breakpoint is done by binary search in $O(\log n)$ steps. The intersection point is then to be determined between this breakpoint and the next one on each envelope. Finding the intersection of $F_i(x)$ and $F_j(x)$ takes at most $O(\log U)$ steps.

Thus the merger of two envelopes of functions is executed in $O(\log n + \log U)$. Since there are at most n mergers in the procedure, the total running time is $O(n \log n + n \log U)$.

Once all the upper envelopes have been identified we have the implied source sets of the associated cuts:

$$\{1, \ldots, i_1\},\ \{1, \ldots, i_2\}, \ldots, \{1, \ldots, i_q\}.$$

If $i \in \{i_{k-1}, \ldots, i_k\}$, then $x_i^* \in (b_{i_{k-1}}, b_{i_k}]$. It remains to apply Lemma 6.1 in order to determine an optimal solution:

$$x_i^* = b_{i_{k-1}} + 1.$$

Thus the total complexity of the algorithm for the isotonic regression problem is $O(n(\log U + \log n))$. In the quadratic case this leads to a complexity of $O(n \log n)$.

8. The continuous CCC problem. When solving the problem in continuous variables, one has to determine how to output the solution. For instance, the minimum of a cubic function can be irrational even if all coefficients are integers. To fully provide the output would then require infinite complexity. To that end we employ the ϵ-accurate complexity model introduced in [10]. According to this model a solution $\mathbf{x}^{(\epsilon)}$ is specified as an integer multiple of ϵ; i.e., it lies on the so-called ϵ-grid. The solution is such that there is an optimal vector $\mathbf{x}^*$ so that

$$||\mathbf{x}^{(\epsilon)} - \mathbf{x}^*||_\infty < \epsilon.$$

The continuous problem can be solved using the same algorithm used for the integer case. The only modification required is in the parametric analysis procedure where the choice of λ^* is such that a median point in the interval (λ_1, λ_2) lies on the ϵ-grid. This is done in additional $O(n \log(U/\epsilon)$ time. The complexity of the algorithm is thus the complexity of finding the roots of the n functions plus the complexity of a minimum cut, $O(mn \log \frac{n^2}{m} + n \log(U/\epsilon))$.

9. Conclusions and extensions. The results here have been extended to a problem that is more general than the CCC problem. The problem is the *convex s-excess problem* which generalizes the s-excess problem discussed in [13]. The problem is formulated as follows:

$$\begin{array}{lll} \text{(Convex s-excess)} & \text{Min} & \sum_{j \in V} f_j(x_j) + \sum e_{ij} z_{ij} \\ \text{subject to} & & x_i - x_j \leq z_{ij} \quad \text{for } (i,j) \in A, \\ & & u_j \geq x_j \geq \ell_j, \quad j = 1, \ldots, n, \\ & & z_{ij} \geq 0, \qquad (i,j) \in A. \end{array}$$

This problem is solved with precisely the same complexity as the minimum closure problem. To that end, we proved a generalization of the threshold theorem reported in [15].

There are applications of the convex s-excess problem in the areas of image segmentation and Markov random fields. The problem is of further interest because of its relationship to the minimum cost network flow problem.

Notice that the terms associated with the variables z_{ij} are linear. This is significant because the dual of the minimum cost network flow (MCNF) problem is

$$\begin{array}{lll} \text{(Dual MCNF)} & \text{Min} & \sum_{j \in V} b_j x_j + \sum e_{ij} z_{ij}, \\ \text{subject to} & & x_i - x_j \leq c_{ij} + z_{ij} \quad \text{for } (i,j) \in A, \\ & & u_j \geq x_j \geq \ell_j, \quad j = 1, \ldots, n, \\ & & z_{ij} \geq 0, \qquad (i,j) \in A. \end{array}$$

That is, the right-hand sides of the constraints have a constant term in addition to the term z_{ij}. Thus, if one can solve the convex s-excess problem with convex function term $\sum e_{ij}(z_{ij})$, then it would have been possible to solve also the dual of the MCNF in the same running time for a single application of maximum flow or minimum cut.

Notes added in proof. (1) The parametric algorithm was shown in [7] to work in strongly polynomial time for linear capacity functions. Hochbaum and Hong [*About strongly polynomial time algorithms for quadratic optimization over submodular constraints,* Math. Programming, 69 (1995), pp. 269–309] showed that the same run time applies when the monotone capacity functions are piecewise linear. The number of pieces in the piecewise linear functions N adds $O(Nn)$ to the complexity of the parametric minimum cut procedure. Since for CCC the capacity functions are derivatives of convex functions it follows that for convex functions that are piecewise linear or piecewise quadratic the run time remains strongly polynomial.

(2) In [13] Hochbaum shows that the pseudoflow algorithm solves the maximum flow minimum cut algorithm for tree graphs in $O(n)$ steps. In tree graphs the set of arcs other than those adjacent to source and sink form an (undirected) acyclic graph. Thus when the partial order graph is a tree the CCC is solved in $O(n \log U)$ using the procedure of [13] with the binary search algorithm described in section 6. This is an alternative algorithm to solve the isotonic regression problem where the linear order graph is a path and thus a tree.

Acknowledgments. The first author thanks Sariel Har-Peled for his input on the state of the art of manipulating lower and upper envelopes of functions and for pointing out references [5], [21], and [22]. The first author also is grateful to Günter Rote for pointing out a simplification in the procedure.

REFERENCES

[1] R. K. Ahuja and J. B. Orlin, *A fast scaling algorithm for minimizing separable convex functions subject to chain constraints*, Oper. Res., 49 (2001), pp. 784–789.

[2] R. K. Ahuja, D. S. Hochbaum, and J. B. Orlin, *A Cut Based Algorithm for the Convex Dual of the Minimum Cost Network Flow Problem*, manuscript.

[3] R. K. Ahuja, D. S. Hochbaum, and J. B. Orlin, *Solving the convex cost integer dual network flow problem*, Management Sci., to appear.

[4] R. E. Barlow, D. J. Bartholomew, J. M. Bremer, and H. D. Brunk, *Statistical Inference Under Order Restrictions*, Wiley, New York, 1972.

[5] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, Springer-Verlag, Berlin, 1997.

[6] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1989.

[7] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan, *A fast parametric maximum flow algorithm and applications*, SIAM J. Comput., 18 (1989), pp. 30–55.

[8] A. V. Goldberg and R. E. Tarjan, *A new approach to the maximum flow problem*, J. Assoc. Comput. Mach., 35 (1988), pp. 921–940.

[9] D. S. Hochbaum and J. Naor, *Simple and fast algorithms for linear and integer programs with two variables per inequality*, SIAM J. Comput., 23 (1994), pp. 1179–1192.

[10] D. S. Hochbaum and J. G. Shanthikumar, *Convex separable optimization is not much harder than linear optimization*, J. Assoc. Comput. Mach., 37 (1990), pp. 843–862.

[11] D. S. Hochbaum, *Lower and upper bounds for allocation problems*, Math. Oper. Res., 19 (1994), pp. 390–409.

[12] D. S. Hochbaum, *Solving integer programs over monotone inequalities in three variables: A framework for half integrality and good approximations,* European J. Oper. Res., 140 (2002), pp. 291–321.

[13] D. S. Hochbaum, *The Pseudoflow Algorithm for the Maximum Flow Problem*, manuscript.

[14] D. S. Hochbaum, *Efficient algorithms for the inverse spanning tree problem*, Oper. Res., to appear.

[15] D. S. Hochbaum, *An efficient algorithm for image segmentation, Markov random fields and related problems*, J. Assoc. Comput. Mach., 48 (2001), pp. 686–701.

[16] W. L. Maxwell and J. A. Muckstadt, *Establishing consistent and realistic reorder intervals in production/distribution systems*, Oper. Res., 33 (1985), pp. 1316–1341.

[17] J. C. Picard, *Maximal closure of a graph and applications to combinatorial problems*, Management Sci., 22 (1976), pp. 1268–1272.

[18] J.-C. Picard and M. Queyranne, *Selected applications of minimum cuts in networks*, INFOR—Canad. J. Oper. Res. Inform. Process., 20 (1982), pp. 394–422.

[19] J. C. Picard and M. Queyranne, *Integer Minimization of a Separable Convex Function Subject to Variable Upper Bound Constraints*, manuscript.

[20] R. Roundy, *A 98%-effective integer-ratio lot-sizing for one-warehouse multi-retailer systems*, Management Sci., 31 (1985), pp. 1416–1430.

[21] M. I. Shamos and D. Hoey, *Geometric intersection problems*, in Proc. 17th Ann. Symp. on Foundations of Computer Science, IEEE, Long Beach, CA, 1976, pp. 208–215.

[22] M. Sharir and P. K. Agarwal, *Davenport-Schinzel Sequences and Their Geometric Applications.* Cambridge University Press, New York, 1995.

[23] P. T. Sokkalingam, R. Ahuja, and J. B. Orlin, *Solving inverse spanning tree problems through network flow techniques*, Oper. Res., 47 (1999), pp. 291–298.

[24] A. F. Veinott, Jr., *Least d-majorized network flows with inventory and statistical applications*, Management Sci., 17 (1971), pp. 547–567.

SIAM J. Discrete Math.
Vol. 16, No. 2, pp. 208–223

ON GENERALIZED DELANNOY PATHS*

JEAN-MICHEL AUTEBERT† AND SYLVIANE R. SCHWER‡

Abstract. A Delannoy path is a minimal path with diagonal steps in $\mathbb{Z}^2$ between two arbitrary points. We extend this notion to the n dimensions space $\mathbb{Z}^n$ and identify such paths with words on a special kind of alphabet: an S-alphabet. We show that the set of all the words corresponding to Delannoy paths going from one point to another is exactly one class in the congruence generated by a Thue system that we exhibit. This Thue system induces a partial order on this set that is isomorphic to the set of ordered partitions of a fixed multiset where the blocks are sets with a natural order relation. Our main result is that this poset is a lattice.

Key words. Delannoy path, Thue system, lattice

AMS subject classifications. 06B05, 68Q42

PII. S0895480101387406

1. Introduction.

A Delannoy path [11] is given as a path that can be drawn on a rectangular grid, starting from the southwest corner, going to the northeast corner, using only three kinds of elementary steps: *north*, *east*, and *northeast*. Hence they are minimal paths with diagonal steps. We generalize the notion of a Delannoy path to the hyperspace $\mathbb{Z}^n$, considering a hyperparallelipedic grid as a set of elementary steps: a step in each direction and the combinations of several of them, the diagonal steps.

We prove that, in a very natural way, an S-alphabet can be associated with the possible elementary steps in a Delannoy path in $\mathbb{Z}^n$, and consequently S-words with Delannoy paths themselves. These notions were introduced by Schwer [8], in a completely different context, for treating simultaneity problems.

We then define a Thue system on the set of S-words that turns out to be noetherian and confluent. This Thue system induces both an ordering on S-words and a congruence. Our main goal is to prove that each equivalence class for this congruence is with this order relation a lattice (Theorem 5.5). (This lattice is a nondistributive lattice as soon as $n > 2$.)

An equivalence class can be viewed as the set of all ordered partitions of a fixed multiset where the blocks are sets (not multisets). There is a transparent bijection between an equivalence class and an element of this set, and the order relation over partitions derived is a very natural one. In [9] are given some links between S-words and others mathematical objects.

Moreover, we exhibit a characterization of the S-words of a class (and so of generalized Delannoy paths going from a point to another) with a family of matrices having its coefficients in $\{-1, 0, 1\}$ (Theorem 4.2), and we prove that the order on S-words can be exactly transposed as the componentwise order on matrices induced by $-1 < 0 < 1$ (Theorem 4.6).

*Received by the editors April 6, 2001; accepted for publication (in revised form) October 16, 2002; published electronically February 20, 2003.
http://www.siam.org/journals/sidma/16-2/38740.html

†Université Paris 7-Denis Diderot, 2 Place Jussieu, 15251 Paris, Cedex 05 France (jean-michel.autebert@liafa.jussieu.fr).

‡Université Paris 13 and LIPN, CNRS UMR 7030, Institut Galilée, Avenue Jean-Baptiste Clément 93430 Villetaneuse, France (schwer@lipn.univ-paris13.fr).

2. Recalls. Concerning lattices, the notations follow [10, 4]. Recall that a lattice is an ordered set such that each pair of elements has a least upper bound and a greatest lower bound. A subset of a lattice is a *sublattice* if for the same order relation it is a lattice. It is a *distributive* lattice if the two operations associating, respectively, with two elements, their least upper bound and a greatest lower bound, are distributive with respect to each other. A lattice ordered by $\leq$ is modular if for all triples of elements (a, b, c) with $a \leq c$ the least upper bound of a and of the greatest lower bound of b and c is equal to the greatest lower bound of c and of the least upper bound of a and b. It is known [10] that every distributive lattice is modular and that the different chains going from one element to another all have the same length in a modular lattice.

Concerning formal languages, we follow [1, 5].

Let X be an alphabet, let X^* be the set of words over X, and let ε be the empty word. If f is a word in X^*, then $|f|$ is the length of f. A word g is a *prefix* of f if some word u exists such that $f = gu$.

Let R be a finite relation over X^*. The Thue system generated by R is the relation over X^*, denoted $\longrightarrow$, that is the smallest relation containing R and compatible with the concatenation product: $(u, v) \in R \Longrightarrow \forall f, g \in X^*, fug \longrightarrow fvg$.

We use freely the usual notions and notations, as can be found, for example, in [1] or [6]. In particular, $\longleftarrow$ denotes the symmetric relation of $\longrightarrow$, $\longleftrightarrow$ the symmetric closure of $\longrightarrow$, and $\longrightarrow^*$ its reflexive and transitive closure. Let set $[f] = \{g \in X^* \mid f \longleftrightarrow^* g\}$ and $\langle f \rangle = \{g \in X^* \mid f \longrightarrow^* g\}$. These notations are extended to languages $[L] = \bigcup_{f \in L} [f]$ and $\langle L \rangle = \bigcup_{f \in L} \langle f \rangle$.

We just recall here the properties [1] of Thue systems that we shall make use of: A *noetherian* system is a system for which no infinite chain exists. A system is *confluent* if $f \longrightarrow^* u$ and $f \longrightarrow^* v$ implies the existence of g such that $u \longrightarrow^* g$ and $v \longrightarrow^* g$. An element f is an *irreducible* element for $\longrightarrow$ if no other element g exists such that $f \longrightarrow g$.

In this paper, we make use of the notions of S-alphabet and S-word introduced by Schwer [8, 9].

Let X be an alphabet. An *S-alphabet* issued from X is a nonempty subset of $\widehat{X} = \{P \in 2^X \mid P \neq \emptyset\}$. $\widehat{X}$ is itself an S-alphabet. The elements of an S-alphabet are called S-letters. Let Y be an S-alphabet subset of $\widehat{X}$; the alphabet $\{x \in X \mid \exists y \in Y \; : \; x \in y\}$ is the *underlying alphabet* of Y. An *S-word* is a word written over an alphabet of S-letters. So we may make use of all the usual notations and definitions of the languages theory for S-words. It is, however, useful to introduce notations that put in relation S-words with the underlying alphabet.

Let $X = \{a_1, a_2, \ldots, a_n\}$, we define the homomorphism $\psi : \widehat{X}^* \longrightarrow \mathbb{N}^n$ by $\psi(P) = (\chi_P(a_1), \ldots, \chi_P(a_n))$, where χ_P is the characteristic function of P. This extends the usual notion of Parikh mapping [5]. The ith component of $\psi(f)$ is denoted $\psi_i(f)$.

We also define the homomorphism $\nu : \widehat{X}^* \longrightarrow \mathbb{N}$ by $\nu(P) = Card(P)$, i.e., $\nu(f) = \Sigma_{1 \leq i \leq n} \psi_i(f)$. So ν is the number of occurrences of letters appearing in all the S-letters of the S-word.

Let $\psi(f) = (p_1, p_2, \ldots, p_n)$; for $m \leq n$, and for $l \leq p_m$, we name *position* of the lth occurrence of the letter a_m the integer $1 + \nu(g)$, where g is the S-word that is the longest prefix of f such that $\psi_m(g) < l$.

To simplify the exposition of the examples, we write the different letters in a S-letter one after the other, without commas to separate them, and we write them in increasing order on the indices.

Example 2.1. *On the alphabet $\widehat{X}$ issued from $X = \{a_1, a_2, a_3\}$, consider the* *word* $f = \{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}\{a_1a_2a_3\}\{a_2\}\{a_2\}$. *It is such that* $\psi(f) = (4, 4, 3)$.

For the letter a_1*, the longest prefixes* g_l *of* f *such that* $\psi_1(g_l) < l$ *when* l *equals* 1*,* 2*,* 3*, and* 4 *are, respectively,* $g_1 = \varepsilon$*,* $g_2 = \{a_1a_2\}\{a_3\}$*,* $g_3 = \{a_1a_2\}\{a_3\}\{a_1\}$*, and* $g_4 = \{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}$*, and we have* $\nu(g_1) = 0$*,* $\nu(g_2) = 3$*,* $\nu(g_3) = 4$*, and* $\nu(g_4) = 6$*. The respective positions of the four occurrences of* a_1 *are then* $1, 4, 5,$ *and* 7*.*

For the letter a_2*, the longest prefixes* g_l *of* f *such that* $\psi_2(g_l) < l$ *when* l *equals* 1*,* 2*,* 3*, and* 4 *are, respectively,* $g_1 = \varepsilon$*,* $g_2 = \{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}$*,* $g_3 = \{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}\{a_1a_2a_3\}$*, and* $g_4 = \{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}\{a_1a_2a_3\}\{a_2\}$*, and we have* $\nu(g_1) = 0$*,* $\nu(g_2) = 6$*,* $\nu(g_3) = 9$*, and* $\nu(g_4) = 10$*. The respective positions of the four occurrences of* a_2 *are then* $1, 7, 10,$ *and* 11*.*

For the letter a_3*, the longest prefixes* g_l *of* f *such that* $\psi_3(g_l) < l$ *when* l *equals* 1*,* 2*, and* 3 *are, respectively,* $g_1 = \{a_1a_2\}$*,* $g_2 = \{a_1a_2\}\{a_3\}\{a_1\}$*, and* $g_3 = \{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}$*, and we have* $\nu(g_1) = 2$*,* $\nu(g_2) = 4$*, and* $\nu(g_4) = 6$*. The respective positions of the three occurrences of* a_3 *are then* $3, 5,$ *and* 7*.*

3. The Thue system. We extend Delannoy paths to the hyperplane $\mathbb{Z}^n$; i.e., we consider minimal paths with diagonal steps between two arbitrary points.

We associate with each dimension a letter of an alphabet $X = \{a_1, a_2, \dots, a_n\}$ and construct the S-alphabet $\widehat{X} = \{P \in 2^X \mid P \neq \emptyset\}$.

The interpretation is the following: the letter $\{a_i\}$ is a step in the dimension i, and more generally the letter $P \in \widehat{X}$ is a simultaneous step in each of the dimensions indicated by the letters of X that belong to P, called *diagonal step* if $Card(P) \geq 2$.

Let us give an arbitrary order over the letters of X by $a_1 < a_2 < \cdots < a_n$. This induces over the S-letters a partial order $P < Q \iff [\forall x \in P, \forall y \in Q :\ x < y]$.

We then define the Thue system, relation denoted $\longrightarrow$ on $\widehat{X}^*$, by the following: $\forall P, Q, R \in \widehat{X}$ such that $P < Q$ and $R = P \cup Q$, set $PQ \longrightarrow R$ and $R \longrightarrow QP$. Note that P and Q are disjoint.

In the case where $n = 2$, with $X = \{a, b\}$, we get $\widehat{X} = \{\{a\}, \{b\}, \{a, b\}\}$, and renaming, respectively, a, b, and c these three letters, the obtained system is precisely the system studied in [2].

Note that doing the bijection of X in itself, which maps a_i on a_{n+1-i}, or reversing the order over the letters of X, which leads exactly to the same relation, one gets $\longleftarrow$, the symmetric relation of $\longrightarrow$. Each property of $\longrightarrow$ is also a property of $\longleftarrow$ (and the converse).

Lemma 3.1. *If* f *and* g *are two words in the same class, their image under* ψ *is the same.*

Proof. By induction, it is sufficient to ensure that each application of a rule preserves the image under ψ. □

Lemma 3.2. *The set of all irreducible words for this Thue system is* $Irr = \{a_n\}^* \dots \{a_2\}^*\{a_1\}^*$*. Symmetrically, the set of all irreducible words for the inverse Thue system is* $\{a_1\}^*\{a_2\}^* \dots \{a_n\}^*$*.*

Proof. Clearly, a word in Irr has no subword being a left factor of a couple in the relation defining the Thue system, and so Irr is a set of irreducible words. Conversely, let f be an S-word not in Irr; then there is either in f an S-letter R containing at least two letters or there are two S-letters $\{a_i\}$ and $\{a_j\}$ with $i < j$ and $\{a_i\}$ is situated before $\{a_j\}$. In the latter case, there exist two such S-letters being consecutive, and the rule $\{a_i\}\{a_j\} \longrightarrow \{a_ia_j\}$ may be applied to f, which is not an irreducible word. In the former case, R can be partitioned between two subsets P and Q so that all the indices of the elements of P are smaller than the indices of the elements of Q, and the rule $R \longrightarrow QP$ may be applied to f, which is not an irreducible word. □

Corollary 3.3. *For each word, there is at most one irreducible word.*

Proof. It is sufficient to check that, among all words having the same image under ψ, there is only one belonging to Irr. □

Lemma 3.4. *The Thue system is noetherian. As a consequence, the relation $\longrightarrow^*$ is an order relation.*

Proof. Let f be an S-word, and let P be an occurrence of one of its S-letters. Let $Post(P,f)$ denote the set of S-letters situated after P in f. To each letter a_m in P is attached the integer $Card(\{i > m \mid a_i \in P\}) + 2. \sum_{Q \in Post(P,f)} Card(\{i > m \mid a_i \in Q\})$, and let $\sigma(f)$ be the sum of these integers for all the occurrences of letters in f. It is easy to check that $f \longrightarrow g \Longrightarrow \sigma(f) > \sigma(g)$. As a consequence, the Thue system is noetherian. The relation $\longrightarrow^*$, which is by definition reflexive and transitive, is antisymmetric as well. It is so an order relation. □

Corollary 3.5. *The Thue system is confluent.*

Proof. Let f and g be two congruent words. As the system is noetherian, they each have an irreducible, and as they are congruent these irreducibles are but one. The two words can be derived on the same word. □

Corollary 3.6. *The following equality holds: $[f] = \{g \in \widehat{X}^* \mid \psi(g) = \psi(f)\}$.*

Proof. The inclusion $[f] \subset \{g \in \widehat{X}^* \mid \psi(g) = \psi(f)\}$ has already been established. Conversely, if two words have the same image under ψ, they have the same irreducible, and so are congruent. □

The n-uple $(p_1, p_2, \ldots, p_n)$ is characteristic of the class of words f satisfying $\psi(f) = (p_1, p_2, \ldots, p_n)$. This class is denoted $\mathfrak{L}(p_1, p_2, \ldots, p_n)$. The quotient $\widehat{X}^*/_{\longleftrightarrow^*}$ is isomorphic to $\mathbb{N}^n$ with componentwise addition.

Altogether, the following holds:

$$\mathfrak{L}(p_1, p_2, \ldots, p_n) = \{g \in \widehat{X}^* \mid \{a_1\}^{p_1}\{a_2\}^{p_2} \ldots \{a_n\}^{p_n} \longrightarrow^* g \longrightarrow^* \{a_n\}^{p_n} \ldots \{a_1\}^{p_1}\}.$$

In other words, $\langle \widehat{X}^*, \longrightarrow^* \rangle$ is a set with a partial order whose set of minimal elements is $\{a_1\}^{p_1}\{a_2\}^{p_2} \ldots \{a_n\}^{p_n}$ and set of maximal elements is $\{a_n\}^{p_n} \ldots \{a_2\}^{p_2}\{a_1\}^{p_1}$.

As noticed before, the set $\mathfrak{L}(p_1, p_2, \ldots, p_n)$ is isomorphic to the set of ordered partitions $(B_1, \ldots, B_k)$ of the multiset $\{1^{p_1} \ldots, n^{p_n}\}$ where the B_i are sets. The covering relation is given by

$$(B_1, \ldots, B_k) \longrightarrow (B_1, \ldots, B_{i-2}, B_{i-1} \cup B_i, B_{i+1}, \ldots, B_k)$$

if $\max B_{i-1} < \min B_i$ and

$$(B_1, \ldots, B_{i-2}, B_{i-1} \cup B_i, B_{i+1}, \ldots, B_k) \longrightarrow (B_1, \ldots, B_k)$$

if $\max B_i < \min B_{i-1}$.

We proved formerly in [2] that $\mathfrak{L}(p_1, p_2)$ with the order relation $\longrightarrow^*$ is a distributive lattice.

The main difference between the case when $n = 2$ and the general case treated here when $n > 2$ is the following: though the order $a < b$ over $X = \{a, b\}$ can easily be extended to a total order over the S-alphabet by setting $\{a\} < \{a, b\} < \{b\}$, the natural generalization of this last: $P < R < Q$ if $\forall x \in P, \forall y \in Q : x < y$ and if $R = P \cup Q$, is not a linear order. This deeply changes the nature of the structure of $\mathfrak{L}(p_1, p_2, \ldots, p_n)$ with the order relation $\longrightarrow^*$.

For instance, the following example shows that $\mathfrak{L}(p_1, p_2, \ldots, p_n)$ is not, in general, a distributive lattice.

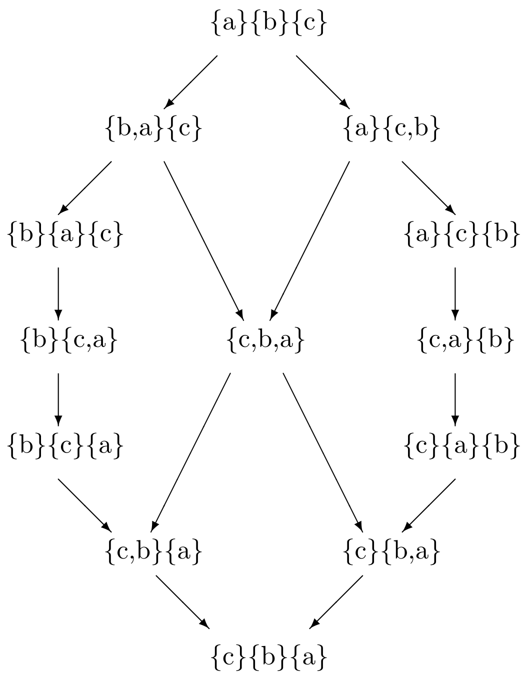


FIG. 3.1. $\mathfrak{L}(1,1,1)$.

EXAMPLE 3.1. *The lattice of* $\mathfrak{L}(1,1,1)$*, represented in Figure* 3.1, *is not modular; hence it is not distributive.*

Nevertheless, it has been announced in [7] that, in the case where all p_i are equal to 1, $\mathfrak{L}(1,1,\dots,1)$ is a lattice. We prove here that it is also true in the general case.

4. The matrix associated to an S-word of $\mathfrak{L}(p_1,p_2,\dots,p_n)$. In what follows, all the S-words are words of $\mathfrak{L}(p_1,p_2,\dots,p_n)$, and we set $s=\Sigma p_i$.

It has already been indicated that the smallest word of $\mathfrak{L}(p_1,p_2,\dots,p_n)$ is the word $f_{\min}=\{a_1\}^{p_1}\{a_2\}^{p_2}\dots\{a_n\}^{p_n}$. For an integer i such that $1\leq i\leq \nu(f)$, we consider the occurrence of the letter in ith position in $f_{\min}$: it is, for some integers l and m, the lth occurrence of a letter a_m. Thus an integer i determines two integers l and m, defined by the relation $i=l+\Sigma_{1\leq s<m}\ p_s$ with $l\leq p_m$. We call *letter of rank* i in a word $f\in\mathfrak{L}(p_1,p_2,\dots,p_n)$ the occurrence of the lth letter a_m where l and m have been so determined. We set $m=r(i)$.

EXAMPLE 4.1. *Let* $X=\{a_1,a_2,a_3\}$. *Considering as in the preceding example the word* $f=\{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}\{a_1a_2a_3\}\{a_2\}\{a_2\}$, *this word is such that* $\psi(f)=(4,4,3)$ *and* $\nu(f)=11$.

The letters of ranks $1,2,3,$ *and* 4 *are occurrences of the letter* a_1, *the letters of ranks* $5,6,7,$ *and* 8 *are occurrences of the letter* a_2, *and the letters of ranks* $9,10,$ *and* 11 *are occurrences of the letter* a_3.

The letter of rank 6 *is thus the second occurrence of the letter* a_2 *belonging to the S-letter* $\{a_1a_2a_3\}$ *that immediately follows the prefix* $\{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}$ *of* f; *its position is* 7.

Table 4.1 *gives explicitly the letters of all ranks and their positions.*

DEFINITION 4.1. *Let* f *be a word of* $\mathfrak{L}(p_1,p_2,\dots,p_n)$. *The* matrix associated

TABLE 4.1

Rank	Letter	S-letter	Former prefix	Position
1	a_1	$\{a_1a_2\}$	ε	1
2	a_1	$\{a_1\}$	$\{a_1a_2\}\{a_3\}$	4
3	a_1	$\{a_1a_3\}$	$\{a_1a_2\}\{a_3\}\{a_1\}$	5
4	a_1	$\{a_1a_2a_3\}$	$\{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}$	7
5	a_2	$\{a_1a_2\}$	ε	1
6	a_2	$\{a_1a_2a_3\}$	$\{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}$	7
7	a_2	$\{a_2\}$	$\{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}\{a_1a_2a_3\}$	10
8	a_2	$\{a_2\}$	$\{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}\{a_1a_2a_3\}\{a_2\}$	11
9	a_3	$\{a_3\}$	$\{a_1a_2\}$	3
10	a_3	$\{a_1a_3\}$	$\{a_1a_2\}\{a_3\}\{a_1\}$	5
11	a_3	$\{a_1a_2a_3\}$	$\{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}$	7

with f, denoted $M(f)$, is the matrix $\nu(f) \times \nu(f)$ whose element $M(f)[i,j]$ of the ith row and of the jth column is

- -1 *if the position in f of the letter of rank i is smaller than the position in f of the letter of rank j;*
- 0 *if the position in f of the letter of rank i is equal to the position in f of the letter of rank j;*
- 1 *if the position in f of the letter of rank i is greater than the position in f of the letter of rank j.*

EXAMPLE 4.2. *Going further with the preceding example, the matrix associated to the word $f = \{a_1a_2\}\{a_3\}\{a_1\}\{a_1a_3\}\{a_1a_2a_3\}\{a_2\}\{a_2\}$ is*

	1	2	3	4	5	6	7	8	9	10	11
1	0	−1	−1	−1	0	−1	−1	−1	−1	−1	−1
2	1	0	−1	−1	1	−1	−1	−1	1	−1	−1
3	1	1	0	−1	1	−1	−1	−1	1	0	−1
4	1	1	1	0	1	0	−1	−1	1	1	0
5	0	−1	−1	−1	0	−1	−1	−1	−1	−1	−1
6	1	1	1	0	1	0	−1	−1	1	1	0
7	1	1	1	1	1	1	0	−1	1	1	1
8	1	1	1	1	1	1	1	0	1	1	1
9	1	−1	−1	−1	1	−1	−1	−1	0	−1	−1
10	1	1	0	−1	1	−1	−1	−1	1	0	−1
11	1	1	1	0	1	0	−1	−1	1	1	0

.

A word f is thus associated with a $\nu(f)\times\nu(f)$ matrix with coefficients in $\{-1,0,1\}$. Conversely, the matrix associated with a word f characterizes this word: it describes which occurrences of letters are situated in the same S-letter and the order of the occurrences of the letters with respect to each other.

The matrix associated with a word owns numerous properties. We list several of them:

- Constructively, a matrix $M(f)$ associated with a word f has only 0's in its diagonal and verifies ${}^tM(f) = -M(f)$.

Denote by $\mathcal{M}$ the set of $s \times s$ matrices M with entries in $\{-1,0,1\}$ verifying ${}^tM = -M$ (and hence $M[i,i] = 0\ \forall i$).

Moreover, the coefficients of the strict upper triangular part share two other properties:

- The first property, called the *commutativity property*, comes out from the commutativity of the occurrences of the same letter between themselves. This property leads us to divide the matrix in submatrices $p_i \times p_j$, just as we did on the example, indicating the orders in the positions of the occurrences of a same letter a_i with those

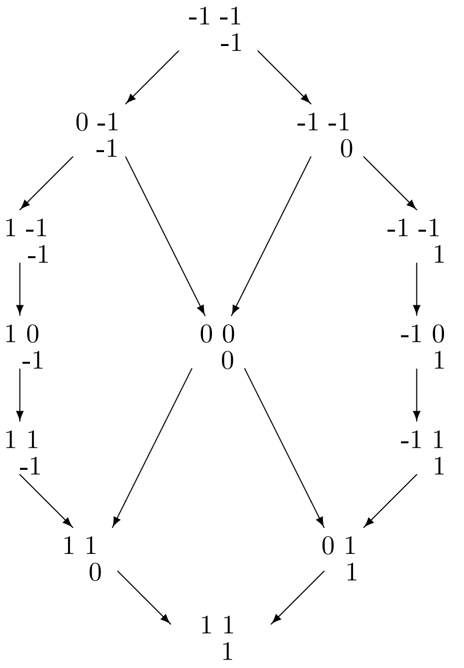


FIG. 4.1. *The lattice of transitivity.*

of another letter a_j. Denote $M_{i,j}$ the submatrix concerning the relationships between letters a_i and a_j.

This commutativity implies that, inside a submatrix $M_{i,j}$, supposing $i < j$,

(i) if i_1 and i_2 are the ranks of two letters a_i, and j_1 the rank of a letter a_j, then $i_1 < i_2$ and $(M[i_1, j_1] = 0 \text{ or } M[i_1, j_1] = 1) \Longrightarrow M[i_2, j_1] = 1$;

(ii) if i_1 is the rank of a letter a_i, and j_1 and j_2 the ranks of two letters a_j, then $j_1 < j_2$ and $(M[i_1, j_1] = 0 \text{ or } M[i_1, j_1] = -1) \Longrightarrow M[i_1, j_2] = -1$.

In the case where $i = j$, i.e., for the submatrix $M_{i,i}$ (square and centered on the diagonal), as we know that the diagonal is made of 0, the upper triangular part is then made of -1.

In what follows, $\mathcal{M}(p_1, \ldots, p_n)$ denotes the set of matrices in $\mathcal{M}$ verifying the commutativity property.

• The second property, called the *transitivity property*, comes from the transitivity of the order relation over the letters of the underlying alphabet: if $a_i < a_j$ and $a_j < a_k$, then $a_i < a_k$ and so the comparisons of the positions of the letters of ranks i and j on one hand, and j and k on the other hand, have an influence upon those of i and k. More precisely, $\forall i, j, k$ such that $i < j < k$, the triple $(M(f)[i,j], M(f)[i,k], M(f)[j,k])$, which we represent under the triangular shape under which it appears in the matrix $\begin{smallmatrix} M(f)[i,j] & M(f)[i,k] \\ & M(f)[j,k] \end{smallmatrix}$ belongs to the following set T_{13} of triples:

$$\left\{ \begin{matrix} -1 & -1 \\ & -1 \end{matrix}, \begin{matrix} 0 & -1 \\ & -1 \end{matrix}, \begin{matrix} 1 & -1 \\ & -1 \end{matrix}, \begin{matrix} 1 & 0 \\ & -1 \end{matrix}, \begin{matrix} 1 & 1 \\ & -1 \end{matrix}, \begin{matrix} 1 & 1 \\ & 0 \end{matrix}, \begin{matrix} 0 & 0 \\ & 0 \end{matrix}, \begin{matrix} -1 & -1 \\ & 0 \end{matrix}, \begin{matrix} -1 & -1 \\ & 1 \end{matrix}, \begin{matrix} -1 & 0 \\ & 1 \end{matrix}, \begin{matrix} -1 & 1 \\ & 1 \end{matrix}, \begin{matrix} 0 & 1 \\ & 1 \end{matrix}, \begin{matrix} 1 & 1 \\ & 1 \end{matrix} \right\},$$

which, ordered by the componentwise order on integers, is a lattice too (cf. Figure 4.1).

One should remark that it is the same lattice as $\mathfrak{L}(1, 1, 1)$.

In what follows, $\mathcal{M}^*(p_1, \ldots, p_n)$ denotes the set of matrices in $\mathcal{M}(p_1, \ldots, p_n)$ verifying the transitivity property.

We shall prove that these conditions do characterize the matrices associated with words f such that $\psi(f) = (p_1, \ldots, p_n)$ (and that, consequently, this association is a bijection between $[f]$ and $\mathcal{M}(p_1, \ldots, p_n)$), establishing the following theorem:

THEOREM 4.2. *Let M be a matrix of $\mathcal{M}$. It is the matrix associated with a word $f \in \mathfrak{L}(p_1, \ldots, p_n)$ if and only if it belongs to $\mathcal{M}^*(p_1, \ldots, p_n)$.*

Let M be a matrix of $\mathcal{M}(p_1, \ldots, p_n)$, and let $s = \Sigma_{j \leq n} p_n$. For all i such that $1 \leq i \leq s$, let pr_i be the number of integers $j > i$ such that $M[i,j] = 1$, and po_i the number of integers $k < i$ such that $M[k,i] = -1$, and we evaluate the integer $pl_i = pr_i + po_i$.

LEMMA 4.3. *For all $i \leq s$, the number of integers j verifying $pl_j < pl_i$ is exactly pl_i.*

Proof. Let i and j be two indices such that $i < j$. These two indices define an integer $x = M[i,j]$ and the following six vectors: V_i is the vector $M[h,i]$ for $1 \leq h < i$; V_j' is the vector $M[h,j]$ for $1 \leq h < i$; V_j'' is the vector $M[h,j]$ for $i < h < j$; H_i' is the vector $M[i,h]$ for $i < h < j$; H_i'' is the vector $M[i,h]$ for $j < h \leq s$; and H_j is the vector $M[j,h]$ for $j < h \leq s$, as indicated in Table 4.2.

TABLE 4.2

		i		j	
		V_i		V_j'	
i		0	H_i'	x	H_i''
				V_j''	
j				0	H_j

Let A be a vector; $|A|_1$ denotes the number of 1's in A and $|A|_{-1}$ denotes the number of -1's in A. In each case, we compare $|H_i''|_1$ and $|H_j|_1$ on one hand, $|V_i|_{-1}$ and $|V_j'|_{-1}$ on the other hand, and finally $|H_i'|_1$ and $|V_j''|_{-1}$, comparisons between vectors of same lengths.

Let i and j be two indices such that $r(i) < r(j)$ (and hence $i < j$).

— In the case where $x = M[i,j] = 1$, one gets $pr_i = |H_i'|_1 + 1 + |H_i''|_1$ and $po_i = |V_i|_{-1}$, and $pr_j = |H_j|_1$ and $po_i = |V_j'|_{-1} + |V_j''|_{-1}$.

The transitivity property implies, $\forall h > j$, $M[j,h] = 1 \Longrightarrow M[i,h] = 1$, and hence $|H_i''|_1 \geq |H_j|_1$, $\forall h < i$, $M[j,h] = -1 \Longrightarrow M[i,h] = -1$, and hence $|V_i|_{-1} \geq |V_j'|_{-1}$, and $\forall i < h < j$, $M[j,h] = -1 \Longrightarrow M[h,i] = 1$, and hence $|H_i'|_1 \geq |V_j''|_{-1}$. So $pl_i > pl_j$.

— In the case where $x = M[i,j] = -1$, one gets $pr_i = |H_i'|_1 + |H_i''|_1$ and $po_i = |V_i|_{-1}$, and $pr_j = |H_j|_1$ and $po_i = |V_j'|_{-1} + 1 + |V_j'|_{-1}$.

In the same way, the transitivity property implies $|H_i''|_1 \leq |H_j|_1$, $|V_i|_{-1} \leq |V_j'|_{-1}$, and $|H_i'|_1 \leq |V_j''|_{-1}$. So $pl_i < pl_j$.

— In the case where $x = M[i,j] = 0$, one gets $pr_i = |H_i'|_1 + |H_i''|_1$ and $po_i = |V_i|_{-1}$, and $pr_j = |H_j|_1$ and $po_i = |V_j'|_{-1} + |V_j'|_{-1}$.

In the same way, the transitivity property implies $|H_i''|_1 = |H_j|_1$, $|V_i|_{-1} = |V_j'|_{-1}$, and $|H'_i|_1 = |V''_j|_{-1}$. So $pl_i = pl_j$.

If i_1 and i_2 are two indices such that $r(i_1) = r(i_2)$ (corresponding to the same i) with $i_1 < i_2$, then, in the same way, following (i) one gets $pr_{i_1} \leq pr_{i_2}$, and following (ii) $po_{i_1} \leq po_{i_2}$, and hence $pl_{i_1} < pl_{i_2}$.

To verify the lemma, it is sufficient now for a fixed i to count down. $\square$

To prove Theorem 4.2, it remains only to prove that the condition is sufficient. Let M be a matrix in $\mathcal{M}^*(p_1, \ldots, p_n)$; we are able to calculate for all i such that $1 \leq i \leq s$ the integer pl_i. A word $f \in \mathfrak{L}(p_1, \ldots, p_n)$ is then constructed by setting its letter of rank i to the position $1 + pl_i$. $\square$

As the matrices associated with congruent words have the same size, they can be ordered by the comparison componentwise of the coefficients of these matrices.

Definition 4.4. *Let f and g be two congruent words of X^*, and let $s = \nu(f) = \nu(g)$. f is* dominated *by g, which is denoted $f \preceq g$, if, for all integers i, j such that $0 < i < j \leq s$, $M(f)[i,j] \leq M(g)[i,j]$ holds.*

In the same way, M and N being two matrices of $\mathcal{M}$, the matrix M is *dominated* by N (or N *dominates* M), which is denoted $M \preceq N$, if, for all integers i, j such that $0 < i < j \leq s$, $M[i,j] \leq N[i,j]$ holds.

We introduce a distance between words in $\mathfrak{L}(p_1, \ldots, p_n)$.

Definition 4.5. *Let d be the application from $\mathfrak{L}(p_1, \ldots, p_n)^2$ to $\mathbb{N}$, with $s = \Sigma p_i$, defined by*

$$d(f,g) = \sum_{0 \leq i < j \leq s} |M(f)[i,j] - M(g)[i,j]|.$$

This application is clearly a distance.

The next theorem is crucial.

Theorem 4.6. $\langle f \rangle = \{g \in [f] \mid f \preceq g\}$.

Proof.

— Let us first prove the inclusion $\langle f \rangle \subseteq \{g \in [f] \mid f \preceq g\}$.

It is sufficient to prove that if $f \longrightarrow g$, then $f \preceq g$, since an easy induction on the number of rewriting rules applied to obtain a word $g \in \langle f \rangle$ from f then gives the result.

• If the applied rule is $PQ \longrightarrow R$ (with $R = P \cup Q$ and $[\forall x \in P, \forall y \in Q : \ x < y]$), then let i be the rank of a letter in P and j the rank of a letter in Q; then $i < j$ and $M(f)[i,j] = -1$, and $M(g)[i,j] = 0$. As these coefficients are the only ones that are changed, $\forall i < j, M(f)[i,j] \leq M(g)[i,j]$ holds.

• If the applied rule is $R \longrightarrow QP$ (with $R = P \cup Q$ and $[\forall x \in P, \forall y \in Q : \ x < y]$), then let i be the rank of a letter in P and j the rank of a letter in Q; then $i < j$ and $M(f)[i,j] = 0$, and $M(g)[i,j] = 1$. As these coefficients are the only ones that are changed, $\forall i < j, M(f)[i,j] \leq M(g)[i,j]$ holds.

— Let us now prove the converse inclusion.

The distance between words will allow us to make an induction on the distance between a word of the set $\{g \in [f] \mid f \preceq g\}$ and f itself.

Let $\mathcal{S}_n$ be the following property: $\{\forall f \in \widehat{X}^*, \forall g \in [f] \mid f \preceq g$ and $d(f,g) \leq n\} \Longrightarrow g \in \langle f \rangle$. We have to prove $\mathcal{S}_n$ for all integer n.

Let $g \in [f]$ be such that $f \preceq g$, and let $n = d(f,g)$.

— If n equals 0, since d is a distance, $g = f$ and $f \longrightarrow^* f$ holds. So $\mathcal{S}_0$ is true.

— Suppose that $n > 0$ and that $\mathcal{S}_{n-1}$ is true. Since $f \preceq g$, there must exist two indices i and j with $1 \leq i < j \leq s$ such that $M(f)[i,j] < M(g)[i,j]$.

Case 1. There are two indices i and j with $1 \leq i < j \leq s$ such that $M(f)[i,j] = 0$

and $M(g)[i,j]=1$.

In this case, let R be the S-letter of f containing the two letters of ranks i and j; among the occurrences of letters in R, there are two verifying the same property as i and j and such that no letter in R has a rank which is an integer between their respective ranks; let P be the set of the letters in R of rank smaller or equal to the smallest of their two ranks, and let Q be the set of the others; $R \longrightarrow QP$ is then a rule of the Thue system. Then let f' be the word obtained from f by substituting to the occurrence of the S-letter R the two S-letters word QP.

Case 2. It is not the case.

Then $\exists i$ and j with $1 \le i < j \le s$ such that $M(f)[i,j] = -1$ and $M(g)[i,j] \ge -1$; we first show that there exist two such indices with, moreover, the condition that the letters of rank i and j are in two consecutive S-letters of f: if not, let k be the rank of a letter inside an intermediate S-letter; if $i < k < j$, then $M(f)[i,k] = M(f)[k,j] = -1$ and either $M(g)[i,k] > -1$, or $M(g)[k,j] > -1$, and so we have the same situation for letters in S-letters that are strictly nearer; if $i < j < k$, then $M(f)[k,j] = 1$, and according to the transitivity property $M(f)[i,k] = -1$, and since $M(g)[k,j] > M(f)[k,j]$, $M(g)[k,j] = 1$, and according to the transitivity property $M(g)[i,k] = 1$, and also in this case we have the same situation for letters in S-letters that are strictly nearer; if $k < i < j$ symmetrically we get the same result.

Supposing now that the letters of rank i and j verifying $1 \le i < j \le s$, $M(f)[i,j] = -1$, and $M(g)[i,j] > -1$ are in two consecutive S-letters in f, say P and Q, and that $j - i$ is the smallest possible, let us show now that i is the largest among the ranks of letters in P: if there is in P a letter of rank $i' > i$, then $M(g)[i,i'] = 0$ because otherwise (if $M(g)[i,i'] = 1$) we would be in Case 1 and if $i' > j$, $M(f)[i',j] = 1$, hence $M(g)[i',j] = 1$, and according to the transitivity property $M(g)[i,i'] = 1$, and again we would be in Case 1, and if $i' \le j$, $M(g)[i',j] \ge -1$ would contradict $j - i$ the smallest possible, and $M(g)[i',j] = -1$ implies according to the transitivity property $M(g)[i,i'] = 1$, and again we would be in Case 1.

Symmetrically, one can prove that j is the smallest among the ranks of letters in Q, and so if $R = P \cup Q$, $PQ \longrightarrow R$ is a rule of the Thue system. Then let f' be a word obtained from the word f replacing the occurrence of the two S-letters word PQ by the S-letter R.

In the two cases, clearly $f \longrightarrow f'$ (and hence $g \in [f']$), and f' is *dominated* by g and $d(f',g) < n$; hence, according to the induction hypothesis, $f' \longrightarrow^* g$. So $f \longrightarrow^* g$ holds, and $\mathcal{S}_n$ is true. □

Noticing that the triples of T_{13} are precisely the upper triangular parts of the matrices attached to the S-words of $\mathfrak{L}(1,1,1)$, we have just proved that the order between S-words of $\mathfrak{L}(1,1,1)$ and the order between the triples of T_{13} are in a complete correspondence, justifying our former remark that it is the same lattice.

5. $\mathfrak{L}(p_1, p_2, \ldots, p_n)$ is a lattice.

Let f and g be two congruent S-words: $f \longleftrightarrow^* g$ with $\nu(f) = \nu(g) = s$. Since the relation $\longrightarrow^*$ is confluent, $\langle f \rangle \cap \langle g \rangle \ne \emptyset$ holds. Let h be an S-word in $\langle f \rangle \cap \langle g \rangle$. The matrix associated with h verifies the following: $\forall i < j, M(f)[i,j] \le M(h)[i,j]$ and $\forall i < j, M(g)[i,j] \le M(h)[i,j]$. Let U be the matrix of $\mathcal{M}$ having in its upper triangular part the following coefficients: $\forall i < j,\ U[i,j] = \mathrm{Max}\{M(f)[i,j], M(g)[i,j]\}$. This matrix has *ipso facto* the commutativity property of matrices in $\mathcal{M}(p_1, \ldots, p_n)$, but it may not have the transitivity property, and so it may not be a matrix in $\mathcal{M}^*(p_1, \ldots, p_n)$.

EXAMPLE 5.1. *Let* $f = \{a_1a_4\}\{a_2a_3a_4\}\{a_3\}$ *and* $g = \{a_1a_3a_4\}\{a_3\}\{a_2a_4\}$. *Their associated matrices are*

$$M(f) = \begin{array}{|c||cc|cc|cc|}
\hline
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & -1 & -1 & -1 & 0 & -1 \\
\hline
2 & 1 & 0 & 0 & -1 & 1 & 0 \\
\hline
3 & 1 & 0 & 0 & -1 & 1 & 0 \\
4 & 1 & 1 & 1 & 0 & 1 & 1 \\
\hline
5 & 0 & -1 & -1 & -1 & 0 & -1 \\
6 & 1 & 0 & 0 & -1 & 1 & 0 \\
\hline
\end{array} \textit{ and } M(g) = \begin{array}{|c||cc|cc|cc|}
\hline
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & -1 & 0 & -1 & 0 & -1 \\
\hline
2 & 1 & 0 & 1 & 1 & 1 & 0 \\
\hline
3 & 0 & -1 & 0 & -1 & 0 & -1 \\
4 & 1 & -1 & 1 & 0 & 1 & -1 \\
\hline
5 & 0 & -1 & 0 & -1 & 0 & -1 \\
6 & 1 & 0 & 1 & 1 & 1 & 0 \\
\hline
\end{array}$$

$$\textit{and the matrix } U \textit{ is } \begin{array}{|c||cc|cc|cc|}
\hline
 & 1 & 2 & 3 & 4 & 5 & 6 \\
\hline
1 & 0 & -1 & 0 & -1 & 0 & -1 \\
\hline
2 & 1 & 0 & 1 & 1 & 1 & 0 \\
\hline
3 & 0 & -1 & 0 & -1 & 1 & 0 \\
4 & 1 & -1 & 1 & 0 & 1 & 1 \\
\hline
5 & 0 & -1 & -1 & -1 & 0 & -1 \\
6 & 1 & 0 & 0 & -1 & 1 & 0 \\
\hline
\end{array}.$$

One can remark that, for example, the triple

$$\begin{array}{cc} U[1,3] & U[1,5] \\ & U[3,5] \end{array} = \begin{array}{cc} 0 & 0 \\ & 1 \end{array}$$

does not belong to the set T_{13}.

However, since U comes from matrices having this transitivity property through the Max operation, among the 14 triples contradicting this property, half of them cannot be in U, namely, the triples

$\begin{array}{cc} 0 & 0 \\ & -1 \end{array}$, $\begin{array}{cc} 0 & 1 \\ & 0 \end{array}$, $\begin{array}{cc} 0 & 1 \\ & -1 \end{array}$, $\begin{array}{cc} -1 & 0 \\ & 0 \end{array}$, $\begin{array}{cc} -1 & 0 \\ & -1 \end{array}$, $\begin{array}{cc} -1 & 1 \\ & 0 \end{array}$, $\begin{array}{cc} -1 & 1 \\ & -1 \end{array}$.

Let us verify for example that $\begin{array}{cc} 0 & 0 \\ & -1 \end{array}$ cannot be in U: this triple comes from two triples of T_{13} $\begin{array}{cc} x & y \\ & -1 \end{array}$ and $\begin{array}{cc} x' & y' \\ & -1 \end{array}$ with $x, x', y, y' \leq 0$. Hence $y = y' = -1$, and so we get a contradiction with $0 = \text{Max}\{y, y'\} = -1$. □

The other triples receive an analogous treatment.

So the only triples not in T_{13} that can be found in U are the following 7:

$\begin{array}{cc} 0 & 0 \\ & 1 \end{array}$, $\begin{array}{cc} 0 & -1 \\ & 0 \end{array}$, $\begin{array}{cc} 0 & -1 \\ & 1 \end{array}$, $\begin{array}{cc} 1 & 0 \\ & 0 \end{array}$, $\begin{array}{cc} 1 & 0 \\ & 1 \end{array}$, $\begin{array}{cc} 1 & -1 \\ & 0 \end{array}$, $\begin{array}{cc} 1 & -1 \\ & 1 \end{array}$.

They are the inverses of the others.

Let T_{20} be the set of triples obtained adding these seven triples to T_{13}.

If T is a subset of the set T_{27} of all the possible triples, let $\mathcal{M}^T(p_1, \ldots, p_n)$ be the set of matrices M in $\mathcal{M}(p_1, \ldots, p_n)$ such that all the triples $(M[i,j], M[i,k], M[j,k])$ belong to T. In particular, $\mathcal{M}^{T_{27}}(p_1, \ldots, p_n) = \mathcal{M}(p_1, \ldots, p_n)$ and $\mathcal{M}^{T_{13}}(p_1, \ldots, p_n) = \mathcal{M}^*(p_1, \ldots, p_n)$.

It is remarkable that, for each of the seven new triples there exists, in the set T_{13} of allowed triples, a unique minimum triple that is bigger than it, respectively:

$\begin{array}{cc} 0 & 1 \\ & 1 \end{array}$, $\begin{array}{cc} 0 & 0 \\ & 0 \end{array}$, $\begin{array}{cc} 0 & 1 \\ & 1 \end{array}$, $\begin{array}{cc} 1 & 1 \\ & 0 \end{array}$, $\begin{array}{cc} 1 & 1 \\ & 1 \end{array}$, $\begin{array}{cc} 1 & 1 \\ & 0 \end{array}$, $\begin{array}{cc} 1 & 1 \\ & 1 \end{array}$.

Let $\odot$ be the operation over $\{-1, 0, 1\}$ defined by the table

$$\begin{array}{r|rrr}
\odot & -1 & 0 & 1 \\
\hline
-1 & -1 & -1 & -1 \\
0 & -1 & 0 & 1 \\
1 & -1 & 1 & 1
\end{array}.$$

We define an operation $\bigodot$ over the matrices in $\mathcal{M}$ by the following: $M \bigodot N$ is the matrix in $\mathcal{M}$ whose coefficients of the upper triangular part are $M \bigodot N[i,k] = \mathrm{Max}_{i \le j \le k} M[i,j] \odot N[j,k]$.

EXAMPLE 5.2. *Going further with the preceding example, we obtain for $U \bigodot U$*

	1	2	3	4	5	6
1	0	−1	0	−1	1	0
2	1	0	1	1	1	1
3	0	−1	0	−1	1	0
4	1	−1	1	0	1	1
5	−1	−1	−1	−1	0	−1
6	0	−1	0	−1	1	0

LEMMA 5.1. *Let M be a matrix of $\mathcal{M}(p_1, \ldots, p_n)$. $M \bigodot M$ is a matrix dominating M belonging to $\mathcal{M}(p_1, \ldots, p_n)$.*

Proof.

— *$M \bigodot M$ dominates M.*

Since $\forall i$ and j such that $i < k$, $M \bigodot M[i,k] = \mathrm{Max}_{i \le j \le k} M[i,j] \odot M[j,k] = \mathrm{Max}\{M[i,i] \odot M[i,k], \mathrm{Max}_{i < j \le k} M[i,j] \odot M[j,k]\}$ holds, and since $M[i,i] = 0$, $M[i,i] \odot M[i,k] = M[i,k]$.

— $M \bigodot M$ has the commutativity property.

First, clearly in a submatrix $M_{i,i}$ the coefficients above the diagonal have value -1; moreover, in a submatrix $M_{i,k}$ with $i < k$, if i_1 and i_2 are the ranks of two letters a_i, and k_1 is the rank of a letter a_k, since $M \bigodot M[i_1,k_1] = 0$ or $M \bigodot M[i_1,k_1] = 1 \Longrightarrow \exists j \mid i_1 \le j \le k_1$ and $M[i_1,j] = 0$ or $M[i_1,j] = 1$ and $M[j,k_1] = 0$ or $M[j,k_1] = 1$; but M having itself the commutativity property, if $i_1 < i_2$, $(M[i_1,j] = 0$ or $M[i_1,j] = 1) \Longrightarrow M[i_2,j] = 1$, and hence $M[i_2,j] \odot M[j,k_1] = 1$, and $M \bigodot M[i_2,k_1] = 1$; in the same way, if i_1 is the rank of a letter a_i, and k_1 and k_2 are the ranks of two letters a_k, $k_1 < k_2$ and $M \bigodot M[i_1,k_1] = 0$ or $M \bigodot M[i_1,k_1] = -1 \Longrightarrow M \bigodot M[i_1,k_2] = -1$. □

Setting $U^{(1)} = U$ and $U^{(i+1)} = U^{(i)} \bigodot U^{(i)}$, starting from U and iterating the operation as long as the obtained matrix does not have the transitivity property, we get a strictly increasing (for the order $\preceq$) sequence of matrices in $\mathcal{M}(p_1, \ldots, p_n)$: $U^{(1)} \prec U^{(2)} \prec \ldots$. The process stops after repeating a finite number of times the operation, and one gets a matrix, denoted U^*, belonging to $\mathcal{M}^*(p_1, \ldots, p_n)$.

According to Theorem 4.2, there exists a word of $\mathfrak{L}(p_1, p_2, \ldots, p_n)$ having this matrix as its associated matrix. Let $f \bigtriangledown g$ be this word. It is a word in the class of f and g.

EXAMPLE 5.3. *Going further with the preceding example, $U \bigodot U$ owns the transitivity property. Hence we get $U^* = U \bigodot U$ which is the matrix associated to the word $f \bigtriangledown g = \{a_4\}\{a_1a_3a_4\}\{a_3\}\{a_2\}$.*

LEMMA 5.2. *Let M be a matrix of $\mathcal{M}^{T_{20}}(p_1, \ldots, p_n)$. $M \bigodot M$ belongs to $\mathcal{M}^{T_{20}}(p_1, \ldots, p_n)$.*

Proof. According to the preceding lemma, $M \bigodot M \in \mathcal{M}^{T_{27}}(p_1, \ldots, p_n)$. Let us review the seven possible cases of triples $\begin{array}{cc} M \odot M[i,j] & M \odot M[i,k] \\ & M \odot M[j,k] \end{array}$ that do not belong to T_{20}.

— Case where $M \bigodot M[i,j] = -1$, $M \bigodot M[i,k] > -1$ and $M \bigodot M[j,k] < 1$.

In this case, $M[i,j] = -1$, and since $M \bigodot M[i,k] > -1$, there exists $j' \neq j$ such that $M[i,j'] > -1$ and $M[j',k] > -1$. Suppose that $j' < j$. Since $M \bigodot M[i,k] > -1$, $M[j',j] = -1$ holds. But $M[j,k] < 1$, and so the triple $\begin{array}{cc} M[j',j] & M[j',k] \\ & M[j,k] \end{array}$ is not in T_{20}, a contradiction. If $j' > j$, since $M[i,j] = -1$ and $M[i,j'] > -1$, $M[j,j'] = 1$ holds because this triple is in T_{20}, or $M[j,j'] = 1$ and $M[j',k] > -1$ implies $M \bigodot M[j,k] = 1$, a contradiction.

— Case where $M \bigodot M[i,j] = 0$, $M \bigodot M[i,k] > -1$, and $M \bigodot M[j,k] = -1$.

In this case, $M[j,k] = -1$, and since $M \bigodot M[i,k] > -1$, there exists $j' \neq j$ such that $M[i,j'] > -1$ and $M[j',k] > -1$. Suppose that $j < j'$. Since $M \bigodot M[j,k] = -1$, $M[j,j'] = -1$ holds. But $M[i,j] < 1$, and so the triple $\begin{smallmatrix} M[i,j] & M[i,j'] \\ & M[j,j'] \end{smallmatrix}$ is not in T_{20}, a contradiction. If $j > j'$, since $M[j,k] = -1$ and $M[j',k] > -1$, $M[j',j] = 1$ holds because this triple is in T_{20}, or $M[j',j] = 1$ and $M[i,j'] > -1$ implies $M \bigodot M[i,j] = 1$, a contradiction.

— Case where $M \bigodot M[i,j] = 0$, $M \bigodot M[i,k] = 1$ and $M \bigodot M[j,k] = 0$.

In this case, $M[i,j] < 1$ and $M[j,k] < 1$, and since $M \bigodot M[i,k] = 1$, there exists $j' \neq j$ such that $M[i,j'] = 1$ and $M[j',k] \geq 0$ or the converse. Suppose that $j' < j$. Since $M \bigodot M[i,j] = 0$, if $M[i,j'] = 1$, $M[j',j] = -1$ holds. But $M[j,k] < 1$, and so the triple $\begin{smallmatrix} M[j',j] & M[j',k] \\ & M[j,k] \end{smallmatrix}$ is not in T_{20}, a contradiction, and if $M[i,j'] = 0$, and hence $M[j',k] = 1$, which with $M[j,k] < 1$ implies $M[j',j] = 1$. Then $M[i,j'] = 0$ and $M[j',j] = 1$ and hence $M \bigodot M[i,j] = 1$, a contradiction with the hypothesis. If $j' > j$, then $M[j',k] > -1$ and $M \bigodot M[j,k] = 0$ implies that $M[j,j'] < 1$, which with $M[i,j] < 1$ implies either $M[i,j'] = -1$, a contradiction with the hypothesis, or $M[i,j] = M[j,j'] = M[i,j'] = 0$; but $M[i,j'] = 0 \Longrightarrow M[j',k] = 1$, which with $M[j,j'] = 0$ implies $M \bigodot M[j,k] = 1$, a contradiction with the hypothesis. □

Lemma 5.3. *If M is a matrix of $\mathcal{M}^*(p_1, \ldots, p_n)$ and N is a matrix of $\mathcal{M}^{T_{20}}(p_1, \ldots, p_n)$ that does not have the transitivity property, then $M \succeq N \Longrightarrow M \succeq N \bigodot N$.*

Proof. Suppose that M does not dominate $N \bigodot N$. Then there exist i and k such that $i < k$ and $N[i,k] \leq M[i,k] < N \bigodot N[i,k]$. Hence, there exists an integer j with $i < j < k$ such that $N \bigodot N[i,k] = N[i,j] \odot N[j,k] > N[i,k]$. So, the triple $\begin{smallmatrix} N[i,j] & N[i,k] \\ & N[j,k] \end{smallmatrix}$ does not belong to T_{13}. Let us review the seven possible cases:

— If $N[j,k] = 1$ and hence $N[i,j] > -1$, then $M[j,k] = 1$ and $M[i,j] > -1$ because M dominates N, and $M[i,k] < N \bigodot N[i,k] = 1$. In all cases, the triple $\begin{smallmatrix} M[i,j] & M[i,k] \\ & M[j,k] \end{smallmatrix}$ does not belong to T_{13}, a contradiction with $M \in \mathcal{M}^*(p_1, \ldots, p_n)$.

— If $N[i,j] = 1$ and $N[j,k] = 0$, then $M[i,j] = 1$ and $M[j,k] > -1$ because M dominates N, and $M[i,k] < N \bigodot N[i,k] = 1$. In all cases, the triple $\begin{smallmatrix} M[i,j] & M[i,k] \\ & M[j,k] \end{smallmatrix}$ does not belong to T_{13}, a contradiction with $M \in \mathcal{M}^*(p_1, \ldots, p_n)$.

— Last, if $N[i,j] = N[j,k] = 0$ and hence $N[i,k] > -1$, then $N \bigodot N[i,k] = 0$ and $M[i,k] < N \bigodot N[i,k] \Longrightarrow M[i,k] = -1$, and M dominates N implies $M[i,j] > -1$ and $M[j,k] > -1$. In all cases, the triple $\begin{smallmatrix} M[i,j] & M[i,k] \\ & M[j,k] \end{smallmatrix}$ does not belong to T_{13}, a contradiction with $M \in \mathcal{M}^*(p_1, \ldots, p_n)$. □

Proposition 5.4. *$\forall h \in \langle f \rangle \cap \langle g \rangle$, $f \bigtriangledown g \preceq h$ holds.*

Proof. Per absurdo, let $h \in \langle f \rangle \cap \langle g \rangle$ be such that $h \neq f \bigtriangledown g$, and let $M(h)$ be its associated matrix. So $M(h)$ dominates U. Hence $M(h) \succeq U^{(1)}$. If $U^{(1)}$ shares the transitivity property, $U^{(1)} = U^*$ holds, and hence $M(h) \succeq U^*$. Otherwise, the preceding lemma shows that $M(h) \succeq U^{(2)}$, and iterating until $U^{(i)} = U^*$, in all cases, $M(h) \succeq U^*$ holds. U^* being the matrix associated with $f \bigtriangledown g$, $f \bigtriangledown g \preceq h$ is true. □

We can now state the following theorem.

Theorem 5.5. *The relation $\longrightarrow^*$ gives to $\mathfrak{L}(p_1, p_2, \ldots, p_n)$ a structure of lattice.*

Proof. Proposition 5.4 means that the word $f \bigtriangledown g$ is a least upper bound of f and g over $[f]$, and $\mathfrak{L}(p_1, p_2, \ldots, p_n)$ has a structure of semilattice.

Symmetrically, $\longrightarrow^*$ confers to $\mathfrak{L}(p_1, p_2, \ldots, p_n)$ a structure of lattice. □

As soon as $n > 2$, the lattice $\mathfrak{L}(p_1, p_2, \ldots, p_n)$ has got $\mathfrak{L}(1, 1, 1)$ as a sublattice. So it is not a modular lattice, hence not a distributive lattice.

Remarks. Since taking the inverse order on the letters of the underlying alphabet leads to the inverse relation of $\longrightarrow^*$, the least upper bound of the mirror images of two congruent S-words is the mirror image of the greatest lower bound of these two words.

Concerning the calculus of matrix U^*, recall that the operation $\bigodot$ replaces a triple in $T_{20} \backslash T_{13}$ by the triple in T_{13} that is the smallest bigger than itself and that this is always done by only increasing the value of the right upper element of the triangle given by the triple. As a consequence, the entries in the matrix that are just above the diagonal are unchanged by the operation, and clearly with each iteration at least one new parallel to the diagonal is definitively set. If s is the dimension of the matrix and if i is the integer for which $U^* = U^{(i)}$, $i \leq s - 2$ holds.

In [3], we present a complete C program, taking advantage of these remarks, computing the least upper bound and the greatest lower bound of two S-words with the method developed in this paper.

6. Conclusion.

We have presented the formalism of S-words that we think is beneficial for treating Delannoy paths. The S-alphabets allow us to describe exactly the set of considered elementary steps. If someone would change the rule allowing only a part of the set of diagonal steps (for instance, only diagonal steps over the faces of a cube), one has only to consider the corresponding S-alphabet, a subalphabet of the S-alphabet we considered, and to proceed to the intersection with the set of words over this subalphabet.

We have associated with S-words, and hence to Delannoy paths, matrices that characterize them. Whatever the rule is, this allows us to order these Delannoy paths by means of the "domination" order, which is nothing more than the componentwise natural order, restricted to the upper triangular part, over these matrices.

The rules could be changed even more drastically to give the possibility of having diagonal steps composed of several elementary steps in a dimension. To describe such paths one has only to make use of multi-S-alphabets, i.e., multisets of letters. In this case, the commutativity property of the associated matrices would be weakened to the following:

In a submatrix $M_{i,j}$, supposing $i < j$,

(i) if i_1 and i_2 are the ranks of two letters a_i, and j_1 the rank of a letter a_j, then $i_1 < i_2 \Longrightarrow M[i_1, j_1] \leq M[i_2, j_1]$;

(ii) if i_1 is the rank of a letter a_i, and j_1 and j_2 the ranks of two letters a_j, then $j_1 < j_2 \Longrightarrow M[i_1, j_1] \geq M[i_1, j_2]$.

An essential part of our work was to exhibit a Thue system that allows us to define the set of Delannoy paths going from one point to another as a class for the congruence generated by the system and to prove that the rewriting process defines an order that coincides with the one of the associated matrices. We think that, if necessary, it would be possible for other rules to exhibit such a Thue system.

Appendix. Table of the sets of triples T_{13}, T_{20}, and T_{27}.

We represent a triple $(M[i,j], M[i,k], M[j,k])$ under the triangular shape it appears in the matrices: $\begin{smallmatrix} M[i,j] & M[i,k] \\ & M[j,k] \end{smallmatrix}$.

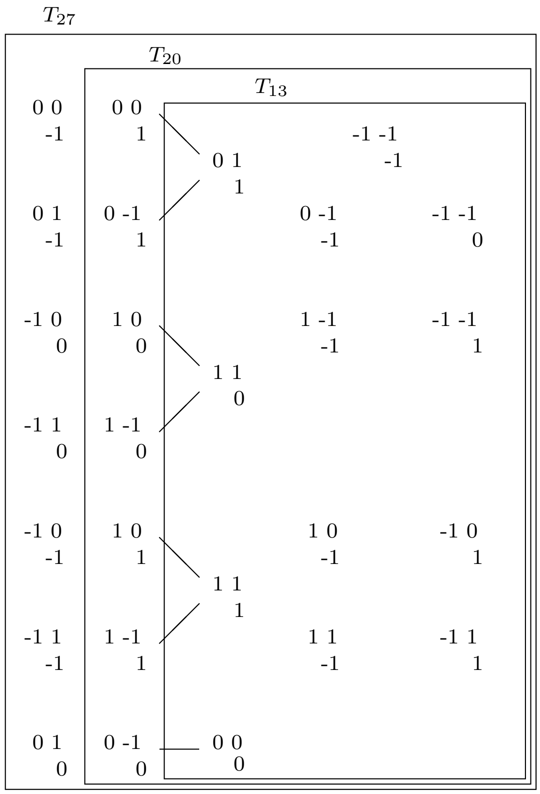


Fig. A.1. *The triples of* T_{13}, T_{20}, *and* T_{27}.

The triples of $T_{20} \backslash T_{13}$ are connected to the triples of T_{13} that cover them. These latter are obtained by replacing the right upper element by the value given by the operation $\odot$ applied to the other two elements of the triple.

REFERENCES

[1] J.-M. Autebert, *Langages Algébriques*, Masson, Paris, 1987.

[2] J.-M. Autebert, M. Latapy, and S. R. Schwer, *Le treillis des chemins de Delannoy*, Discrete Math., 258 (2002) pp. 225–234.

[3] J.-M. Autebert and S. R. Schwer, *Chemins de Delannoy généralisés*, LIPN internal report 2001-04, Villetaneuse, France, 2001.

[4] B. A. Davey and H. A. Priestly, *Introduction to Lattices and Order*, Cambridge University Press, Cambridge, UK, 1990.

[5] S. Ginsburg, *The Mathematical Theory of Context Free Languages*, McGraw-Hill, New York, 1966.

[6] M. Jantzen, *Confluent String Rewriting*, EATCS Monogr. Theoret. Comput. Sci. 14, Springer-Verlag, Berlin, 1988.

[7] D. Krob, M. Latapy, J.-C. Novelli, H. D. Phan, and S. R. Schwer, *Pseudo-permutations* I: *First combinatorial and lattice properties*, in Proceedings of the 13th International Conference on Formal Power Series & Algebraic Combinatorics, Arizona State University, Tempe, AZ, 2001.

[8] S. R. Schwer, *Dépendances temporelles : les mots pour le dire*, LIPN internal report, Villetaneuse, France, 1997.

[9] S. R. Schwer, *S-arrangements avec répétitions*, C. R. Acad. Sci. Paris Ser. I, 334 (2002), pp. 261–266.

[10] G. Szász, *Théorie des treillis*, Dunod, Paris, 1971.

[11] E. W. Weisstein, *CRC Concise Encyclopaedia of Mathematics*, CRC Press, Boca Raton, FL, 2000.

SIAM J. DISCRETE MATH.
Vol. 16, No. 2, pp. 224–261

CONTRAST OPTIMAL THRESHOLD VISUAL CRYPTOGRAPHY SCHEMES[*]

C. BLUNDO[†], P. D'ARCO[‡], A. DE SANTIS[†], AND D. R. STINSON[§]

Abstract. A (k, n)-threshold visual cryptography scheme (VCS) is a method to encode a secret image SI into n shadow images called *shares* such that any k or more shares enable the "visual" recovery of the secret image. However, by inspecting less than k shares one cannot gain any information on the secret image. The "visual" recovery consists of copying the shares onto transparencies and then stacking them. Any k shares will reveal the secret image without any cryptographic computation. In this paper we analyze the contrast of the reconstructed image for a (k, n)-threshold VCS. We define a canonical form for a (k, n)-threshold VCS and provide a characterization of a (k, n)-threshold VCS. We completely characterize a contrast optimal $(n-1, n)$-threshold VCS in canonical form. Moreover, for $n \geq 4$, we provide a contrast optimal $(3, n)$-threshold VCS in canonical form. We first describe a family of $(3, n)$-threshold VCS achieving various values of contrast and pixel expansion. Then we prove an upper bound on the contrast of any $(3, n)$-threshold VCS and show that a scheme in the described family has optimal contrast. Finally, for $k = 4, 5$ we present two schemes with contrast asymptotically equal to $1/64$ and $1/256$, respectively.

Key words. visual cryptography, secret sharing schemes

AMS subject classification. 94A60

PII. S0895480198336683

1. Introduction.

A (k, n)-threshold visual cryptography scheme (VCS) for a set $\mathcal{P}$ of n participants is a method to encode a secret image SI into n shadow images called *shares*, where each participant in $\mathcal{P}$ receives one share. Any (qualified) set of k or more participants can "visually" recover the secret image, but (forbidden) sets of participants of cardinality less than k have no information (in an information-theoretic sense) on SI. A "visual" recovery for a set $X \subseteq \mathcal{P}$ consists of copying the shares given to the participants in X onto transparencies and then stacking them. The participants in a qualified set X will be able to see the secret image without any knowledge of cryptography and without performing any cryptographic computation. VCS are characterized by two parameters: the *pixel expansion*, which is the number of subpixels that each pixel of the original image is encoded into, and the *contrast*, which measures the "difference" between a black pixel and a white pixel in the reconstructed image.

This cryptographic paradigm was introduced by Naor and Shamir [12]. Further results on (k, n)-threshold VCS can be found in [1, 3, 5, 7, 9, 16]. The model by Naor and Shamir has been extended in [1, 3] to general access structures (an access structure is a specification of all qualified and forbidden subsets of participants), where general

[*] Received by the editors April 1, 1998; accepted for publication (in revised form) July 22, 2002; published electronically February 20, 2003.
http://www.siam.org/journals/sidma/16-2/33668.html

[†] Dipartimento di Informatica ed Applicazioni, Università di Salerno, 84081 Baronissi (SA), Italy (carblu@dia.unisa.it, ads@dia.unisa.it). The research of the first and third authors was supported by the Italian Ministry of University and Research (MURST) and by the National Council for Research (CNR).

[‡] Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada (pdarco@cacr.math.uwaterloo.ca).

[§] School of Computer Science, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada (dstinson@cacr.math.uwaterloo.ca). The research of this author was partially supported by NSERC grants IRC#216431-96 and RGPIN#203114-02.

techniques to construct VCS for any access structure have been proposed. Droste [7] gave an algorithm to construct (k, n)-threshold VCS. In [3] the authors provide the first construction for $(2, n)$-threshold VCS having the best possible contrast for any $n \geq 2$. In [5], for any n, is provided a complete characterization of $(2, n)$-threshold VCS having optimal contrast and minimum pixel expansion in terms of certain balanced incomplete block designs. The authors of [9] showed that by solving a suitable linear program one can compute the best contrast achievable in any (k, n)-threshold VCS. In [9], for the cases $k = 2$ with n even and $k = 3$ with n divisible by 4, a (k, n)-threshold VCS achieving the best possible contrast is described.

For a simple and nontechnical introduction to visual cryptography see [15].

In implementing VCS it would be useful to conceal the existence of the secret message; namely, the shares given to participants in the scheme should not appear as random pixels, but recognizable images (a house, a dog, a tree, etc.). Naor and Shamir [12] first considered the problem of concealing the existence of the secret message for the case of 2 out of 2 threshold VCS. In [2] the authors gave a general technique to implement VCS with such an extended capability. Droste [7] also considered the problem of concealing the existence of the secret message and presented a technique to implement such schemes.

Alternative reconstruction methods for VCS based on "opaque" shares [13] and on polarized filters [4] have been recently proposed. Both models make assumptions different from ours on the way the shares combine. VCS to encrypt colored images are given in [10, 14, 16]. Recently, authentication and identification methods for human users based on VCS have been considered [11]. The authors of [6] analyze the amount of randomness needed to visually share a secret image.

In this paper we analyze the contrast for (k, n)-threshold VCS. We are mainly interested in schemes achieving the maximum possible contrast for any fixed values of k and n. We refer to such schemes as *contrast optimal*. We define a canonical form for (k, n)-threshold VCS and characterize (k, n)-threshold VCS (see Lemmas 3.9 and 3.10). We completely characterize contrast optimal $(n-1, n)$-threshold VCS in canonical form. Moreover, for $n \geq 4$, we present a contrast optimal $(3, n)$-threshold VCS in canonical form. We first describe a family of $(3, n)$-threshold VCS achieving various values of contrast and pixel expansion. Then we prove an upper bound on the contrast of any $(3, n)$-threshold VCS and show that a scheme in the described family has optimal contrast. Finally, for $k = 4$ and $k = 5$ we present two schemes with contrast asymptotically equal to 1/64 and 1/256, respectively.

2. The model.

We assume that the secret image consists of a collection of black and white pixels. Each pixel appears in n versions called *shares*, one for each transparency. Each share is a collection of m black and white subpixels. The resulting structure can be described by an $n \times m$ boolean matrix $S = [s_{ij}]$ where $s_{ij} = 1$ if and only if the jth subpixel in the ith transparency is black. Therefore the grey level of the combined shares obtained by stacking the transparencies $i_1, \ldots, i_s$ is proportional to the Hamming weight $w(V)$ of the m-vector $V = OR(r_{i_1}, \ldots, r_{i_s})$, where $r_{i_1}, \ldots, r_{i_s}$ are the rows of S associated with the transparencies we stack. This grey level is interpreted by the visual system of the participants as black or white according to some rule of contrast.

Definition 2.1. *Let k and n be two integers such that $k \leq n$ and let $\mathcal{P}$ be a set of n participants. Two collections (multisets) of $n \times m$ boolean matrices $\mathcal{C}_0$ and $\mathcal{C}_1$ constitute a (k, n)-*threshold VCS *with pixel expansion m if there exist the value α and the set $\{(X, t_X)\}_{X \subseteq \mathcal{P} : |X| = k}$ satisfying the following:*

1. Any (qualified) set $X = \{i_1, i_2, \ldots, i_k\} \subseteq \mathcal{P}$ can recover the shared image by stacking its transparencies. *Formally, for any $M \in \mathcal{C}_0$, the OR V of rows $i_1, i_2, \ldots, i_k$ satisfies $w(V) \leq t_X - \alpha \cdot m$, whereas for any $M \in \mathcal{C}_1$ we have $w(V) \geq t_X$.*

2. Any (forbidden) set $X = \{i_1, i_2, \ldots, i_p\} \subseteq \mathcal{P}$, with $p < k$, has no information on the shared image. *Formally, the two collections of $p \times m$ matrices $\mathcal{D}_t$, with $t \in \{0, 1\}$, obtained by restricting each $n \times m$ matrix in $\mathcal{C}_t$ to rows $i_1, i_2, \ldots, i_p$, are indistinguishable in the sense that they contain the same matrices with the same frequencies.*

Each pixel of the original image will be encoded into n pixels, each of which consists of m subpixels. To share a white (resp., black) pixel, the dealer randomly chooses one of the matrices in $\mathcal{C}_0$ (resp., $\mathcal{C}_1$,) and distributes row i to participant i. Thus, the chosen matrix defines the m subpixels in each of the n transparencies. Notice that, in the previous definition, $\mathcal{C}_0$ is a multiset of $n \times m$ boolean matrices. Therefore we allow a matrix to appear more than once in $\mathcal{C}_0$ (resp., $\mathcal{C}_1$). Finally, observe that the sizes of the collections $\mathcal{C}_0$ and $\mathcal{C}_1$ do not need to be the same.

The first property is related to the contrast of the image. It states that when any k participants stack their transparencies they can correctly recover the image shared by the dealer. The value α is called the *contrast* of the image, and the set $\{(X, t_X)\}_{X \subseteq \mathcal{P}: |X| = k}$ is called the *set of thresholds*. (We use a slightly different terminology from [12], where the contrast is called *relative difference* and the quantity $\alpha \cdot m$ is called the *contrast of the scheme*.) We want the product of the contrast times the pixels expansion to be as large as possible and at least 1, that is, $\alpha \geq 1/m$. The second property is called *security* since it implies that, even by inspecting all their shares, any set of less than k participants cannot gain any information to help in deciding whether the shared pixel was white or black.

Notice that if a set of participants X is a superset of a qualified set X', then the participants can recover the shared image by considering only the shares of the set X'. This does not rule out the possibility that stacking all the transparencies of the participants in X will not reveal any information about the shared image. A *strong* (k, n)-threshold VCS is a (k, n)-threshold VCS in which property 1 of Definition 2.1 is satisfied for any set X of cardinality at least k; that is, the image is visible if and only if k or *more* participants stack their transparencies.

There are few differences between the model of visual cryptography we propose and the one presented by Naor and Shamir [12]. Our model is a generalization of the one proposed in [12] since with each set X of size k we associate a (possibly) different threshold t_X. Nevertheless, all the (k, n)-threshold VCS given in this paper have the property that for any $X, X' \subseteq \mathcal{P}$ with $|X| = |X'| \geq k$, we have $t_X = t_{X'}$.

2.1. Basis matrices. In this paper we consider only (k, n)-threshold VCS in which the collections $\mathcal{C}_0$ and $\mathcal{C}_1$ have the same size, i.e., $|\mathcal{C}_0| = |\mathcal{C}_1| = r$. Actually, this is not a restriction at all. Indeed, in section 2.1 of [1] it has been shown how to obtain, from an arbitrary (k, n)-threshold VCS, a VCS having the same parameters m, α, and $\{(X, t_X)\}_{X \subseteq \mathcal{P}: |X| = k}$, with equally sized $\mathcal{C}_0$ and $\mathcal{C}_1$.

All of the constructions in this paper are realized using two $n \times m$ matrices, S^0 and S^1, called *basis matrices*, satisfying the following definition.

DEFINITION 2.2. *Let k and n be two integers such that $k \leq n$ and let $\mathcal{P}$ be a set of n participants. A (k, n)-threshold VCS with contrast α and set of thresholds $\{(X, t_X)\}_{X \subseteq \mathcal{P}: |X| = k}$ is realized using the two $n \times m$ basis matrices S^0 and S^1 if the following two conditions hold:*

1. *If $X = \{i_1, i_2, \ldots, i_k\} \subseteq \mathcal{P}$, (i.e., if X is a qualified set), then the OR V of*

rows $i_1, i_2, \ldots, i_k$ of S^0 satisfies $w(V) \leq t_X - \alpha \cdot m$, whereas for S^1 we have $w(V) \geq t_X$.

2. *If $X = \{i_1, i_2, \ldots, i_p\} \subseteq \mathcal{P}$ and $p < k$ (i.e., if X is a forbidden set), then the two $p \times m$ matrices obtained by restricting S^0 and S^1 to rows $i_1, i_2, \ldots, i_p$ are equal up to a column permutation.*

The collections $\mathcal{C}_0$ and $\mathcal{C}_1$ are obtained by permuting the columns of the corresponding basis matrix (S^0 for $\mathcal{C}_0$, and S^1 for $\mathcal{C}_1$) in all possible ways. Note that in this case, the size of the collections $\mathcal{C}_0$ and $\mathcal{C}_1$ is the same (equal to $m!$) and is denoted by r. This technique has been introduced in [12]. The algorithm for the VCS based on the previous construction of the collections $\mathcal{C}_0$ and $\mathcal{C}_1$ has small memory requirements (it keeps only the basis matrices S^0 and S^1) and it is efficient (to choose a matrix in $\mathcal{C}_0$ (resp., $\mathcal{C}_1$) it only generates a permutation of the columns of S^0 (resp., S^1)).

3. Canonical (k, n)-threshold VCS. Most of the constructions found in the literature for (k, n)-threshold VCS are realized by using basis matrices. Among these constructions there are a few having the property that all the columns of a given weight appear with the same multiplicity in the basis matrices (see, for instance, [12, 3, 1, 7, 5, 9, 16]). Because of the relevance of this property, we review in (i)–(iv) below some of the constructions for (k, n)-threshold VCS having such a property.

(i) Naor and Shamir [12] proposed a (k, k)-threshold VCS obtained by construction of the basis matrices S^0 and S^1 defined as follows: S^0 is the matrix whose columns are all the boolean k-vectors having an even number of 1's, and S^1 is the matrix whose columns are all the boolean k-vectors having an odd number of 1's. In [12] the basis matrices of $(2, n)$-threshold VCS are realized as follows: S^0 contains $n-1$ columns of weight 0 and one column of weight n, whereas S^1 contains all the columns of weight 1. Naor and Shamir [12] also proposed a $(3, n)$-threshold VCS whose basis matrices are realized as follows: S^0 contains $n-2$ columns of weight 0 and all the columns of weight $n-1$, whereas S^1 contains all the columns of weight 1 and $n-2$ columns of weight n.

(ii) In [3] the authors showed how to construct a $(2, n)$-threshold VCS that is optimal with respect to the contrast. The basis matrix S^1 of such a scheme is realized by considering all the columns of weight $\lfloor n/2 \rfloor$, whereas the basis matrix S^0 contains $\binom{n-1}{\lfloor n/2 \rfloor}$ columns of weight 0 and $\binom{n-1}{\lfloor n/2 \rfloor - 1}$ columns of weight n.

(iii) Droste [7] gave an algorithm to construct basis matrices of any (k, n)-threshold VCS. The basis matrices realized by such an algorithm are constructed by adding or deleting all the columns of particular weights to or from the basis matrices.

(iv) Other (k, n)-threshold VCS in which all the columns of a given weight appear in the basis matrices can be found in [5]. For instance, when $k | n$, setting $\ell = n! / ((n/k)!)^k$, we have that, for $j = 0, \ldots, \lfloor k/2 \rfloor$, the basis matrix S^1 is realized by considering all the columns of weight $(2j+1)n/k$, each appearing with multiplicity ℓ, and the basis matrix S^0 contains all the columns of weight $2jn/k$, each appearing with multiplicity ℓ.

(v) In [9] basis matrices containing all the columns of a given weight, each occuring with the same frequency, are referred to as *totally symmetric* matrices. The authors analyzed (k, n)-threshold VCS having totally symmetric basis matrices. They gave explicit constructions for $k = 2, 3, n$.

(vi) In [16] the authors proposed two constructions for (k, n)-threshold VCS whose parameters are connected to notions in finite geometry and coding theory. The basis matrices derived from such constructions contain all the columns of a given weight.

In this section we consider basis matrices containing all the columns of a given weight, each occurring with the same frequency, with few additional properties (see Definition 3.1). We refer to such matrices as *canonical.* We show how to construct for any (k, n)-threshold VCS a canonical scheme preserving the contrast. Since we are interested in optimizing the contrast, we focus our attention only on the canonical form.

Before we state our results we need to set up our notation. Let M be an $n \times m$ matrix and let $X \subseteq \{1, \ldots, n\}$ and $Z \subseteq \{1, \ldots, m\}$. Let $M[X][Z]$ denote the $|X| \times |Z|$ matrix obtained from M by considering its restriction to rows and columns indexed by X and Z, respectively. Let M be a matrix in the collection $\mathcal{C}_0 \cup \mathcal{C}_1$ of a (k, n)-threshold VCS on a set of participants $\mathcal{P}$. For $X \subseteq \mathcal{P}$, let M_X denote the m-vector obtained by considering the OR of the rows corresponding to participants in X, whereas $M[X] = M[X][\{1, \ldots, m\}]$ denotes the $|X| \times m$ matrix obtained from M by considering only the rows corresponding to participants in X. Let M be a matrix and let D be a submatrix of M having the same number of rows; we denote by $M \backslash D$ the matrix obtained from M by removing all the columns of the matrix D. For sets X and Y and for elements x and y, to avoid overburdening the notation we will often write x for $\{x\}$, xy for $\{x, y\}$, xY for $\{x\} \cup Y$, and XY for $X \cup Y$. Let $\mathbf{c}$ be a boolean vector. We denote by $\overline{\mathbf{c}}$ the vector obtained from $\mathbf{c}$ by complementing all its entries, whereas we denote by $\overline{M}$ the matrix obtained from M by complementing all its entries. For $i = 0, 1$, we denote by $f_{\mathbf{c},i}$ the multiplicity of the column $\mathbf{c}$ in S^i; that is, $f_{\mathbf{c},i}$ is the number of times column $\mathbf{c}$ appears in S^i. By abusing notation, we write $\mathbf{c} \in M$ to denote the fact that $\mathbf{c}$ is a column of the matrix M.

DEFINITION 3.1. *Let (S^0, S^1) be the basis matrices of a (k, n)-threshold VCS. They are in* canonical *form if, for $i = 0, 1$, the following two properties are satisfied:*

1. *For any columns $\mathbf{c}$ and $\mathbf{c}'$ such that $w(\mathbf{c}) = w(\mathbf{c}')$, we have $f_{\mathbf{c},i} = f_{\mathbf{c}',i}$.*
2. *For any column $\mathbf{c}$, we have*

$$f_{\mathbf{c},i} = \begin{cases} f_{\overline{\mathbf{c}},i} & \textit{if } k \textit{ is even,} \\ f_{\overline{\mathbf{c}},1-i} & \textit{if } k \textit{ is odd.} \end{cases}$$

A (k, n)-threshold VCS whose basis matrices are in canonical form is referred to as a *canonical (k, n)-threshold VCS.*

To prove some of our results we need the following theorem.

THEOREM 3.2 (see [5]). *Let S^0 and S^1 be two $n \times m$ boolean matrices. The matrices S^0 and S^1 are basis matrices of a (k, n)-threshold VCS with pixel expansion m and contrast α if and only if for all subsets X consisting of k rows there exist a boolean matrix $D[X]$ and an integer $z_X \geq \alpha \cdot m$ such that $D[X]$ is a submatrix of both $S^0[X]$ and $S^1[X]$, all the even columns appear in $S^0[X] \backslash D[X]$ with multiplicity z_X, and all the odd columns appear in $S^1[X] \backslash D[X]$ with multiplicity z_X.*

Theorem 3.2 follows directly from Theorem 7.1 of [5] and Lemma 3.5 of [1]. More precisely, Theorem 7.1 of [5] establishes that a pair of basis matrices (T^0, T^1), such that the same column does not appear in both, realizes a (k, k)-threshold VCS if and only if there is an integer h for which T^0 contains all the even columns with multiplicity h, and T^1 contains all the odd columns with the same multiplicity h. Since, for any subset X of k rows, the restriction $(S^0[X], S^1[X])$ of (S^0, S^1) defines a pair of basis matrices realizing a (k, k)-threshold VCS, then, from Lemma 3.5 of [1], $S^0[X]$ and $S^1[X]$ have the following structure: There are a matrix $D[X]$ and an integer z_X such that $D[X]$ is a submatrix of both $S^0[X]$ and $S^1[X]$, all the even columns appear in $S^0[X] \backslash D[X]$ with multiplicity z_X, and all the odd columns appear in $S^1[X] \backslash D[X]$ with the same multiplicity z_X.

Example 3.3. Let

$$S^0 = \left[\begin{array}{cccccccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ \hline 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array}\right], \quad S^1 = \left[\begin{array}{cccccccc} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

be two basis matrices realizing a $(3, 5)$-threshold VCS. If we look at the restrictions to the first three rows of these matrices, then it is easy to see that $S^0[X]$ (resp., $S^1[X]$) contains all the even (resp., odd) columns once and that the common matrix, up to a column permutation, is

$$D[X] = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

We will use the next lemma to prove that if there exists a (k, n)-threshold VCS with contrast α, then there exists a canonical (k, n)-threshold VCS having the same contrast α. A weaker version of the result stated by the next lemma was independently proved in [16, Thm. 5.7].

LEMMA 3.4. *Let (S^0, S^1) be the basis matrices of a (k, n)-threshold VCS with pixel expansion m and contrast α. The matrices (B^0, B^1), defined as*

$$(B^0, B^1) = \begin{cases} (\overline{S^1}, \overline{S^0}) & \textit{if } k \textit{ is odd,} \\ (\overline{S^0}, \overline{S^1}) & \textit{if } k \textit{ is even,} \end{cases}$$

are the basis matrices of a (k, n)-threshold VCS with pixel expansion m and contrast α.

Proof. Assume that k is odd and let $B^0 = \overline{S^1}$ and $B^1 = \overline{S^0}$. Since (S^0, S^1) are basis matrices of a (k, n)-threshold VCS, then, from Theorem 3.2, for all subsets X consisting of k rows there exist a boolean matrix D^X and an integer $z_X \geq \alpha \cdot m$ such that D^X is a submatrix of both $S^0[X]$ and $S^1[X]$, all the even columns appear in $S^0[X] \backslash D^X$ with multiplicity z_X, and all the odd columns appear in $S^1[X] \backslash D^X$ with multiplicity z_X. Hence, for all subsets X consisting of k rows there exist a boolean matrix $G^X = \overline{D^X}$ and an integer z_X such that G^X is a submatrix of both $B^0[X]$ and $B^1[X]$, all the even columns appear in $B^0[X] \backslash G^X$ with multiplicity z_X, and all the odd columns appear in $B^1[X] \backslash G^X$ with multiplicity z_X. Therefore, from Theorem 3.2, we get that (B^0, B^1) are basis matrices of a (k, n)-threshold VCS. We immediately see that the contrast of the (k, n)-threshold VCS having basis matrices (B^0, B^1) is the same as the contrast of the scheme with which we started.

The proof for the case k even is analogous to the one for k odd. □

In [5] it was shown that if there exists a (k, n)-threshold VCS Σ realized using collections of $n \times m$ boolean matrices $\mathcal{C}_0$ and $\mathcal{C}_1$ having contrast α, then there exists a (k, n)-threshold VCS realized by using basis matrices having the same contrast as Σ. We state this result as a lemma.

LEMMA 3.5. *Let $\mathcal{C}_0$ and $\mathcal{C}_1$ be the collections of matrices of a (k, n)-threshold VCS with contrast α. Then, there exists a (k, n)-threshold VCS realized by using basis matrices having contrast α.*

Proof. Without loss of generality we can assume that $r = |\mathcal{C}_0| = |\mathcal{C}_1|$ (see section 2.1 of [1]). Suppose that $\mathcal{C}_0 = \{M^{0,1}, \ldots, M^{0,r}\}$ and $\mathcal{C}_1 = \{M^{1,1}, \ldots, M^{1,r}\}$, where

$\circ$ denotes the concatenation of two matrices. We can immediately check that $S^0 = M^{0,1} \circ \cdots \circ M^{0,r}$ and $S^1 = M^{1,1} \circ \cdots \circ M^{1,r}$ constitute the basis matrices of a (k,n)-threshold VCS having the same contrast as Σ. □

The next lemma holds.

LEMMA 3.6. *Let $\mathcal{C}_0$ and $\mathcal{C}_1$ be the collections of matrices of a (k,n)-threshold VCS with contrast α. Then there exists a canonical (k,n)-threshold VCS realized by basis matrices (S^0, S^1) having contrast α.*

Proof. Assume k odd. Let Σ be a (k,n)-threshold VCS with pixel expansion m and contrast α. Suppose that Σ is realized using collections of $n \times m$ boolean matrices $\mathcal{C}_0$ and $\mathcal{C}_1$. By Lemma 3.5 there exists a (k,n)-threshold VCS realized by using basis matrices (S^0, S^1) having the same contrast α as Σ. For $i = 0, 1$, let $\mathcal{D}^i$ be the collection of boolean matrices obtained from S^i by permuting its rows. Now, construct a new pair of matrices D^0 and D^1 by concatenating all the matrices in $\mathcal{D}^0$ and $\mathcal{D}^1$, respectively. We can immediately see that D^0 and D^1 constitute basis matrices of a (k,n)-threshold VCS having the same contrast as Σ. At this point, we have that if a column of weight w appeared in S^0 (S^1), then all the columns of weights w appear in D^0 (resp., D^1). Finally, let $B^0 = \overline{D^1}$ and $B^1 = \overline{D^0}$. By Lemma 3.4, the pair (B^0, B^1) represents the basis matrices of a (k,n)-threshold VCS having contrast α. It is straightforward to check that $A^0 = B^0 \circ D^0$ and $A^1 = B^1 \circ D^1$ are the basis matrices of a canonical (k,n)-threshold VCS having contrast α.

The proof for the case k even is analogous to the one for k odd. □

Notice that the authors of [9] considered totally symmetric matrices which satisfy only property 1 of Definition 3.1 and they proved the analogous result of Lemma 3.6.

In any canonical (k,n)-threshold VCS, by property 1 of Definition 3.1 all the columns of a given weight appear with the same multiplicity. Therefore, we define the multiplicity of a column of weight j in S^i as $h_{j,i}$, i.e., $h_{j,i} = f_{\mathbf{c},i}$ if $w(\mathbf{c}) = j$. Hence, any canonical (k,n)-threshold VCS can be simply described by the pair of vectors $(h_{0,0}, \ldots, h_{n,0})$ and $(h_{0,1}, \ldots, h_{n,1})$. Clearly, the pixel expansion m of a canonical (k,n)-threshold VCS is equal to

$$m = \sum_{j=0}^{n} h_{j,0} \binom{n}{j} = \sum_{j=0}^{n} h_{j,1} \binom{n}{j}.$$

Moreover, it is easy to see that in a canonical (k,n)-threshold VCS, for any $X, X' \subseteq \mathcal{P}$, with $|X| = |X'| = k$, we have that $t_X = t_{X'}$, as in the original definition by Naor and Shamir [12]. This also means that the optimal contrast is the same in our definition as in the Naor–Shamir definition (however, the minimal pixel expansion need not be the same).

The next corollary is a consequence of Definition 3.1.

COROLLARY 3.7. *Let Σ be a (k,n)-threshold VCS in canonical form. If k is odd, then for $j = 0, \ldots, n$ we have that $h_{j,0} = h_{n-j,1}$, whereas if k is even, for $j = 0, \ldots, n$, we have that $h_{j,0} = h_{n-j,0}$ and $h_{j,1} = h_{n-j,1}$.*

There is another equality relating the $h_{i,j}$'s that is based on the security of the (k,n)-threshold VCS. From condition 2 of Definition 2.2 in [1], for $j = 0, \ldots, n$, it has to be that $w(S^0[j]) = w(S^1[j])$, from which one gets that

$$\sum_{i=1}^{n} h_{i,0} \binom{n-1}{i-1} = \sum_{i=1}^{n} h_{i,1} \binom{n-1}{i-1}.$$

Hence, in any canonical (k,n)-threshold VCS all the rows of the basis matrices have the same weight. The next corollary is an immediate consequence of the previous observation and of Lemma 3.6.

COROLLARY 3.8. *The pixel expansion of any canonical (k,n)-threshold VCS is twice the weight of any row of a basis matrix.*

Proof. Suppose n is odd (resp., n is even) and let (S^0, S^1) be the basis matrices of a canonical (k,n)-threshold VCS. From Corollary 3.7 we have that $S^{1-i} = \overline{S^i}$ ($S^i = \overline{S^i}$) for $i = 0, 1$. Hence, as in any canonical (k,n)-threshold VCS where all the rows of the basis matrices have the same weight, we have that the weight of any row of a basis matrix is half of the pixel expansion of the scheme. □

Notice that if (A^0, A^1) and (B^0, B^1) are (k,n)-threshold VCS having contrast α, then $(A^0 \circ B^0, A^1 \circ B^1)$, where $\circ$ denotes the operator "concatenation" of two matrices, is a (k,n)-threshold VCS having contrast α. Hence, if $(h_{0,0}, \ldots, h_{n,0})$ and $(h_{0,1}, \ldots, h_{n,1})$ are a pair of vectors describing a canonical (k,n)-threshold VCS having contrast α, then, for any positive integer ℓ, the vectors $(\ell \cdot h_{0,0}, \ldots, \ell \cdot h_{n,0})$ and $(\ell \cdot h_{0,1}, \ldots, \ell \cdot h_{n,1})$ again describe a canonical (k,n)-threshold VCS having contrast α. Therefore, if we want to minimize the pixel expansion m for a given value of the contrast α, we consider values $h_{0,0}, \ldots, h_{n,0}, h_{0,1}, \ldots, h_{n,1}$ such that $\gcd(h_{0,0}, \ldots, h_{n,0}) = \gcd(h_{0,1}, \ldots, h_{n,1}) = 1$.

Suppose that $n \geq 2$ is an integer and that $2 \leq k \leq n$. For $i = 0, 1$, let $h_i = (h_{0,i}, \ldots, h_{n,i})$ be an $(n+1)$-tuple of nonnegative integers. For $i = 0, 1$, define $S(h_i)$ to be the matrix in which every binary n-tuple of weight j occurs exactly $h_{j,i}$ times as a column $(0 \leq j \leq n)$. In the following we provide a necessary and sufficient condition for the existence of (k,n)-threshold VCS realized by such matrices $S(h_0)$ and $S(h_1)$. The following lemma holds.

LEMMA 3.9. *$S(h_0)$ and $S(h_1)$ are basis matrices of a (k,n)-threshold VCS with pixel expansion m and contrast α if and only if the following properties are satisfied:*

1. $\sum_{j=0}^{n} \binom{n}{j} h_{j,0} = \sum_{j=0}^{n} \binom{n}{j} h_{j,1} = m.$
2. $\sum_{j=p'}^{n-p+p'} \binom{n-p}{j-p'} h_{j,0} = \sum_{j=p'}^{n-p+p'} \binom{n-p}{j-p'} h_{j,1}$ *for* $1 \leq p \leq k-1$ *and* $0 \leq p' \leq p$.
3. $\sum_{j=0}^{n-k} \binom{n-k}{j} (h_{j,0} - h_{j,1}) = \alpha \cdot m.$

Proof. Suppose that $S(h_0)$ and $S(h_1)$ are basis matrices for a VCS with the stated parameters. The number of columns in $S(h_i)$ $(i = 0, 1)$ is

$$\sum_{j=0}^{n} \binom{n}{j} h_{j,i}.$$

Therefore property 1 holds.

Next, let $\mathbf{c}$ be a binary column p-tuple, where $0 \leq p \leq k-1$. Suppose that the weight of $\mathbf{c}$ is p' (note that $p' \leq p$). Fix p rows of $S(h_0)$ and $S(h_1)$, say the first p rows. The number of occurrences of $\mathbf{c}$ as a column of $S(h_i)[\{1, \ldots, p\}]$ is

$$\sum_{j=p'}^{n-p+p'} \binom{n-p}{j-p'} h_{j,i}$$

for $i = 0, 1$. Therefore property 2 holds.

Finally, we look at the weight of the OR of k rows of $S(h_0)$ and $S(h_1)$, say the first k rows. If we let $X = \{1, \ldots, k\}$, then

$$w(S(h_1)_X) - w(S(h_0)_X) \geq \alpha \cdot m.$$

Let ϵ_i denote the number of occurrences of $(0,\dots,0)^T$ as a column of $S(h_i)[X]$ for $i = 0, 1$. It is easy to see that

$$w(S(h_i)_X) = m - \epsilon_i$$

for $i = 0, 1$. Hence,

$$w(S(h_i)_X) = m - \sum_{j=0}^{n-k} \binom{n-k}{j} h_{j,i}$$

for $i = 0, 1$. Therefore property 3 holds.

Conversely, if properties 1–3 hold, it is easy to see that $S(h_0)$ and $S(h_1)$ are basis matrices for a VCS with the stated parameters. □

We can in fact simplify the statement of the above lemma by observing that many of the conditions are redundant. More precisely, property 2 of Lemma 3.9 considers, for each $1 \le p \le k-1$, the restriction of the basis matrices to p rows and requires that the same subcolumns appear with the same frequencies. However, we can simply check if the subcolumns of weight $1 \le p' \le k-1$ appear with the same frequencies in $S(h_0)$ and $S(h_1)$. Indeed, if this property is satisfied, the symmetric structure of the matrices ensures that any restriction of $S(h_0)$ and $S(h_1)$ to $1 \le p \le k-1$ rows contains the same subcolumns with the same frequencies.

From a mathematical point of view, by repeated application of Pascal's identity for binomial coefficients to property 2 of Lemma 3.9, we obtain the following equivalent formulation.

LEMMA 3.10. *$S(h_0)$ and $S(h_1)$ are basis matrices of a (k,n)-threshold VCS with pixel expansion m and contrast α if and only if the following properties are satisfied:*

1. $\sum_{j=0}^{n} \binom{n}{j} h_{j,0} = \sum_{j=0}^{n} \binom{n}{j} h_{j,1} = m$.
2. *For* $1 \le p' \le k-1$, $\sum_{j=0}^{n-p'} \binom{n-p'}{j} h_{j,0} = \sum_{j=0}^{n-p'} \binom{n-p'}{j} h_{j,1}$.
3. $\sum_{j=0}^{n-k} \binom{n-k}{j} (h_{j,0} - h_{j,1}) = \alpha \cdot m$.

Example 3.11. Suppose $k = 2$ and $n = 4$. The following example is from [5]. Let $h_0 = (3,0,0,0,3)$ and let $h_1 = (0,0,1,0,0)$. This defines a $(2,4)$-threshold VCS with $m = 6$ and contrast $\alpha = 1/3$:

$$\sum_{j=0}^{4} \binom{4}{j} h_{j,0} = \binom{4}{0} 3 + \binom{4}{4} 3 = 6,$$

$$\sum_{j=0}^{4} \binom{4}{j} h_{j,1} = \binom{4}{2} 1 = 6,$$

$$\sum_{j=0}^{3} \binom{3}{j} h_{j,0} = \binom{3}{0} 3 = 3,$$

$$\sum_{j=0}^{3} \binom{3}{j} h_{j,1} = \binom{3}{2} 1 = 3,$$

$$\sum_{j=0}^{2} \binom{2}{j} (h_{j,0} - h_{j,1}) = \binom{2}{0} 3 - \binom{2}{2} 1 = 2.$$

Example 3.12. Suppose $k = 3$ and $n = 7$. The following example is an application of a construction we give in section 4.2. Let $h_0 = (9, 0, 0, 0, 0, 1, 0, 0)$ and let $h_1 = (0, 0, 1, 0, 0, 0, 0, 9)$. This defines a $(3, 7)$-threshold VCS with $m = 30$ and contrast $\alpha = 1/10$:

$$\sum_{j=0}^{7} \binom{7}{j} h_{j,0} = \binom{7}{0} 9 + \binom{7}{5} 1 = 30,$$

$$\sum_{j=0}^{7} \binom{7}{j} h_{j,1} = \binom{7}{2} 1 + \binom{7}{7} 9 = 30,$$

$$\sum_{j=0}^{6} \binom{6}{j} h_{j,0} = \binom{6}{0} 9 + \binom{6}{5} 1 = 15,$$

$$\sum_{j=0}^{6} \binom{6}{j} h_{j,1} = \binom{6}{2} 1 = 15,$$

$$\sum_{j=0}^{5} \binom{5}{j} h_{j,0} = \binom{5}{0} 9 + \binom{5}{5} 1 = 10,$$

$$\sum_{j=0}^{5} \binom{5}{j} h_{j,1} = \binom{5}{2} 1 = 10,$$

$$\sum_{j=0}^{4} \binom{4}{j} (h_{j,0} - h_{j,1}) = \binom{4}{0} 9 - \binom{4}{2} 1 = 3.$$

The characterization of (k, n)-threshold VCS provided by Lemma 3.10, because of Lemma 3.6, gives rise to a natural and simple formulation for computing their optimal contrast for any fixed n and k in terms of linear programming. We set $m = 1$ without loss of generality since α is unchanged if all the $h_{j,i}$'s are multiplied by a constant factor. The resulting linear program has only $2n + 2$ variables.

Maximize

$$\alpha = \sum_{j=0}^{n-k} \binom{n-k}{j} (h_{j,0} - h_{j,1})$$

subject to

$$\begin{aligned} &\sum_{j=0}^{n} \binom{n}{j} h_{j,0} = 1, \\ &\sum_{j=0}^{n} \binom{n}{j} h_{j,1} = 1, \\ &\sum_{j=0}^{n-p'} \binom{n-p'}{j} (h_{j,0} - h_{j,1}) = 0 \qquad &&\text{for } p' = 1, \ldots, k-1, \\ &h_{j,0} \geq 0 &&\text{for } j = 0, \ldots, n, \\ &h_{j,1} \geq 0 &&\text{for } j = 0, \ldots, n. \end{aligned}$$

It is worthwhile to notice that our linear program is equivalent to, but simpler than, the one given in [9]. In Appendix B, the entries of Figures B.1–B.3 have been filled in by solving the previous linear programming problem for $2 \leq k \leq n \leq 11$. Also in [9] are tabulated some values of the contrast.

We can further simplify the previous linear program formulation by taking into account Corollary 3.7. For odd values of k the linear program formulation can be written as follows:

Maximize

$$\alpha = \sum_{j=0}^{n-k} \binom{n-k}{j} (h_{j,0} - h_{n-j,0})$$

subject to

$$\begin{array}{ll} \displaystyle\sum_{j=0}^{n} \binom{n}{j} h_{j,0} = 1, & \\ \displaystyle\sum_{j=0}^{n-p'} \binom{n-p'}{j} (h_{j,0} - h_{n-j,0}) = 0 & \text{for } p' = 1, \ldots, k-1, \\ h_{j,0} \geq 0 & \text{for } j = 0, \ldots, n. \end{array}$$

For even values of k the linear program formulation can be obtained similarly. This new linear program formulation is clearly simpler than the previous one, as it uses only half of the variables and reduces the number of constraints.

In view of Lemma 3.6, if we are interested in obtaining schemes with a given contrast or bound on the contrast itself, then we can restrict our attention to canonical (k, n)-threshold VCS. Henceforth, unless otherwise specified, all (k, n)-threshold VCS we consider or analyze are canonical (k, n)-threshold VCS.

4. Contrast optimal (k, n)-threshold VCS.

We recall that, for fixed values of k and n, a contrast optimal scheme is a scheme achieving the maximum possible contrast over all (k, n)-threshold VCS. Contrast optimal (k, n)-threshold VCS for $k = 2$ and $k = n$ already have been extensively studied (see [5, 12]). It is interesting that the basis matrices realizing the (k, k)-threshold VCS described in [12], and the basis matrices of the first construction proposed in [5] for $(2, n)$-threshold VCS, are both in canonical form.

Notice that the same column cannot appear in both basis matrices of a contrast optimal (k, n)-threshold VCS. This property is easy to verify. Indeed, if the same column appears in both basis matrices, then by removing it we obtain a new scheme having a contrast better than the one with which we started. This property implies the following fact.

FACT 4.1. *In any contrast optimal (k, n)-threshold VCS whose basis matrices are in canonical form, for $j = 0, \ldots, n$ and $i = 0, 1$, it holds that*

1. *if $h_{j,1-i} > 0$, then $h_{j,i} = 0$.*
2. *if k is even, then $h_{j,i} = h_{n-j,i}$.*
3. *if k is odd, then $h_{j,i} = h_{n-j,1-i}$.*

As a consequence of this fact and Corollary 3.7, we have that if n is even and k is odd, then $h_{n/2,0} = h_{n/2,1} = 0$.

4.1. Contrast optimal $(n-1,n)$-threshold VCS. In this section we characterize contrast optimal $(n-1,n)$-threshold VCS whose basis matrices are in canonical form.

The next lemma holds.

LEMMA 4.2. *Let $n \geq 3$. In any contrast optimal $(n-1,n)$-threshold VCS whose basis matrices are in canonical form, the $h_{j,i}$'s satisfy the following:*

1. *$h_{j,0} > 0$ if and only if either $j < n/2$ with j even or $j > n/2$ with j odd.*
2. *$h_{j,1} > 0$ if and only if either $j < n/2$ with j odd or $j > n/2$ with j even.*

Proof. Let (S^0, S^1) be the basis matrices of a canonical $(n-1,n)$-threshold VCS which is contrast optimal. It holds that

$$\begin{aligned} &\text{if } j \text{ is odd and } h_{j,1} = 0, \text{ then } h_{j+1,1} > 0, \\ &\text{whereas if } j \text{ is even and } h_{j,0} = 0, \text{ then } h_{j+1,0} > 0; \end{aligned} \tag{1}$$

otherwise we have $h_{j,1} = h_{j+1,1} = 0$ which is impossible as, by Theorem 3.2, all the columns of weight j have to appear among the columns of $S^1[X]$, where X is a subset of $\{1,\ldots,n\}$ of cardinality $n-1$. Similarly, we can prove that if j is even and $h_{j,0} = 0$, then it holds that $h_{j+1,0} > 0$.

We will prove that for any integer $j < n/2$ it holds that

$$\textit{if } j \textit{ is even, then } h_{j,0} > 0, \textit{ whereas if } j \textit{ is odd, then } h_{j,1} > 0. \tag{2}$$

Therefore, applying Corollary 3.7, the lemma holds.

Now assume that n is even and $j < n/2$. Suppose by contradiction that $h_{j,0} = 0$. From (1) and by Fact 4.1 we have $h_{j+1,0} > 0$ and $h_{j+1,1} = 0$. Applying again (1) and Fact 4.1 we get $h_{j+2,1} > 0$ and $h_{j+2,0} = 0$. Iterating the previous argument we get that either $h_{n/2,0} > 0$ or $h_{n/2,1} > 0$, depending on whether $n/2$ is even or odd, which is a contradiction (recall that $h_{n/2,0} = h_{n/2,1} = 0$). If j is odd, then we repeat the proof for the case j even mutatis mutandis.

If n is odd, then by Corollary 3.7 we have that $h_{(n-1)/2,i} = h_{(n+1)/2,i}$, where $i = 0,1$. At this point we repeat the proof for the case n even mutatis mutandis. We get that either $h_{(n-1)/2,0} = 0$ and $h_{(n+1)/2,0} > 0$ or $h_{(n-1)/2,1} = 0$ and $h_{(n+1)/2,1} > 0$, which is a contradiction. Thus, the lemma holds. □

The next lemma states the exact value of the $h_{j,i}$ of any contrast optimal $(n-1,n)$-threshold VCS whose basis matrices are in canonical form.

LEMMA 4.3. *Let $n \geq 3$. In any contrast optimal $(n-1,n)$-threshold VCS whose basis matrices are in canonical form, the $h_{j,i}$'s satisfy the following:*

(i) *If n is even, then for $j = 0,\ldots,\lfloor (n-2)/4 \rfloor$, we have $h_{2j,0} = h_{n-2j,1} = (n/2) - 2j$, whereas for $j = 0,\ldots,\lfloor (n-4)/4 \rfloor$, we have $h_{2j+1,1} = h_{n-(2j+1),0} = (n/2) - (2j+1)$.*

(ii) *If n is odd, then for $j = 0,\ldots,\lfloor n/4 \rfloor$, we have $h_{2j,0} = h_{n-2j,0} = n - 4j$, whereas for $j = 0,\ldots,\lceil (n-5)/4 \rceil$, we have $h_{2j+1,1} = h_{n-(2j+1),1} = n - (4j+2)$.*

Proof. Let Σ be a contrast optimal $(n-1,n)$-threshold VCS. Let (S^0, S^1) be the $n \times m$ basis matrices of Σ and let α be its contrast. Let X be a subset of $\{1,\ldots,n\}$ of cardinality $n-1$ and let $\mathbf{c}$ be a column of weight j, where $n/2 \leq j < n$. Suppose j is even. According to Theorem 3.2, the column $\mathbf{c}$ has to appear at least $\alpha \cdot m$ times more in $S^0[X]$ than in $S^1[X]$. Therefore, since Σ is contrast optimal, by Lemma 4.2 we have that $h_{j+1,0} - h_{j,1} = \alpha \cdot m$. A similar argument applies when j is odd. In this case we obtain $h_{j+1,1} - h_{j,0} = \alpha \cdot m$. For n even, recalling Lemma 4.2 and setting, without loss of generality, $\alpha \cdot m = 1$, we get the following $n/2$ linear equations in n

unknowns:

$$\begin{aligned} h_{n-2j,1} - h_{n-(2j+1),0} &= 1 \ \text{ for } j = 0,\dots,\lfloor (n-2)/4 \rfloor, \\ h_{n-(2j+1),0} - h_{n-(2j+2),1} &= 1 \ \text{ for } j = 0,\dots,\lfloor (n-4)/4 \rfloor. \end{aligned} \tag{3}$$

Summing up (3) and recalling that $h_{n/2,0} = h_{n/2,1} = 0$, we get that $h_n = n/2$ from which we can compute the value of the other unknowns. Therefore, we obtain that if n is even, then for $j = 0,\dots,\lfloor (n-2)/4 \rfloor$, we have $h_{2j,0} = h_{n-2j,1} = (n/2) - 2j$; whereas, for $j = 0,\dots,\lfloor (n-4)/4 \rfloor$, we have $h_{2j+1,1} = h_{n-(2j+1),0} = (n/2) - (2j+1)$.

If n is odd, then we set $\alpha \cdot m = 2$ and repeat the proof for the case n even mutatis mutandis. □

The results of the above lemma can be summarized as follows: If n is even, then, for $j = 0,\dots,n$,

$$h_{j,0} = h_{n-j,1} = \begin{cases} (n/2) - j & \text{if } j \text{ is even and } j < n/2, \\ j - (n/2) & \text{if } j \text{ is odd and } j > n/2, \\ 0 & \text{otherwise.} \end{cases}$$

If n is odd, then, for $j = 0,\dots,\lfloor n/2 \rfloor$,

$$h_{j,0} = h_{n-j,0} = \begin{cases} n - 2j & \text{if } j \text{ is even and } j < n/2, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$h_{j,1} = h_{n-j,1} = \begin{cases} n - 2j & \text{if } j \text{ is odd and } j < n/2, \\ 0 & \text{otherwise.} \end{cases}$$

The next lemma holds.

LEMMA 4.4. *For any $n \geq 3$ and for any contrast optimal canonical $(n-1,n)$-threshold VCS the pixel expansion m is given by*

$$m = \begin{cases} (n/4)\binom{n}{n/2} & \text{if } n \text{ is even,} \\ n\binom{n-1}{(n-1)/2} & \text{if } n \text{ is odd.} \end{cases}$$

Proof. Assume n is even. We have that,

$$\begin{aligned} m &= \sum_{j=0}^{n} h_{j,0}\binom{n}{j} \\ &= \sum_{j=0}^{\lfloor \frac{(n-2)}{4} \rfloor} \left(\frac{n}{2} - 2j\right)\binom{n}{2j} + \sum_{j=0}^{\lfloor \frac{(n-4)}{4} \rfloor} \left(\frac{n}{2} - (2j+1)\right)\binom{n}{2j+1} \\ &= \sum_{j=0}^{\frac{n}{2-1}} \left(\frac{n}{2} - j\right)\binom{n}{j}. \end{aligned}$$

Since for any even integer r and any integer g it holds that (see [8, p. 166])

$$\sum_{j=0}^{g} \left(\frac{r}{2} - j\right)\binom{r}{j} = \frac{g+1}{2}\binom{r}{g+1},$$

then

$$m = \frac{n}{4}\binom{n}{\frac{n}{2}}.$$

On the other hand, if n is odd, then

$$m = \sum_{j=0}^{n} h_{j,0}\binom{n}{j} = 2\sum_{j=0}^{\lfloor n/4 \rfloor}(n-4j)\binom{n}{2j}.$$

We begin by simplifying the sum as follows:

$$\begin{aligned}
\sum_{j=0}^{\lfloor \frac{n}{4} \rfloor}(n-4j)\binom{n}{2j} &= \sum_{j=0}^{\lfloor \frac{n}{4} \rfloor}\left((n-2j)\binom{n}{n-2j} - 2j\binom{n}{2j}\right)\\
&= \sum_{j=0}^{\lfloor \frac{n}{4} \rfloor}\left(n\binom{n-1}{n-2j-1} - n\binom{n-1}{2j-1}\right)\\
&= n\left(\sum_{j=0}^{\lfloor \frac{n}{4} \rfloor}\binom{n-1}{2j} - \sum_{j=1}^{\lfloor \frac{n}{4} \rfloor}\binom{n-1}{2j-1}\right).
\end{aligned}$$

Recall that

$$\sum_{j=0}^{\lfloor \frac{n-1}{2} \rfloor}\binom{n-1}{2j} = \sum_{j=1}^{\lfloor \frac{n}{2} \rfloor}\binom{n-1}{2j-1} = 2^{n-2}$$

for any positive integer n. Suppose $n \equiv 1 \bmod 4$. Then we have the following:

$$\begin{aligned}
\sum_{j=0}^{\lfloor \frac{n-1}{2} \rfloor}\binom{n-1}{2j} &= \sum_{j=0}^{\lfloor \frac{n}{4} \rfloor}\binom{n-1}{2j} + \sum_{j=\lfloor \frac{n}{4} \rfloor}^{\lfloor \frac{n-1}{2} \rfloor}\binom{n-1}{2j} - \binom{n-1}{\frac{n-1}{2}}\\
&= \sum_{j=0}^{\lfloor \frac{n}{4} \rfloor}\binom{n-1}{2j} + \sum_{i=0}^{\lfloor \frac{n}{4} \rfloor}\binom{n-1}{2\left(\lfloor \frac{n}{4} \rfloor + i\right)} - \binom{n-1}{\frac{n-1}{2}}\\
&= \sum_{j=0}^{\lfloor \frac{n}{4} \rfloor}\binom{n-1}{2j} + \sum_{j=0}^{\lfloor \frac{n}{4} \rfloor}\binom{n-1}{n-1-2j} - \binom{n-1}{\frac{n-1}{2}}\\
&= 2\sum_{j=0}^{\lfloor \frac{n}{4} \rfloor}\binom{n-1}{2j} - \binom{n-1}{\frac{n-1}{2}}.
\end{aligned}$$

Suppose $n \equiv 3 \bmod 4$. Then we have the following:

$$\begin{aligned}
\sum_{j=0}^{\lfloor \frac{n-1}{2} \rfloor}\binom{n-1}{2j} &= \sum_{j=0}^{\lfloor \frac{n}{4} \rfloor}\binom{n-1}{2j} + \sum_{j=\lfloor \frac{n}{4} \rfloor+1}^{\lfloor \frac{n-1}{2} \rfloor}\binom{n-1}{2j}\\
&= \sum_{j=0}^{\lfloor \frac{n}{4} \rfloor}\binom{n-1}{2j} + \sum_{i=0}^{\lfloor \frac{n}{4} \rfloor}\binom{n-1}{2\left(\lfloor \frac{n}{4} \rfloor + 1 + i\right)}
\end{aligned}$$

$$= \sum_{j=0}^{\lfloor \frac{n}{4} \rfloor} \binom{n-1}{2j} + \sum_{j=0}^{\lfloor \frac{n}{4} \rfloor} \binom{n-1}{n-2j}$$
$$= 2 \sum_{j=0}^{\lfloor \frac{n}{4} \rfloor} \binom{n-1}{2j}.$$

Therefore, for n odd we have

$$\sum_{j=0}^{\lfloor \frac{n}{4} \rfloor} \binom{n-1}{2j} = \begin{cases} \frac{1}{2}\left(2^{n-2} + \binom{n-1}{\frac{n-1}{2}}\right) & \text{if } n \equiv 1 \bmod 4, \\ \frac{1}{2}\left(2^{n-2}\right) & \text{if } n \equiv 3 \bmod 4. \end{cases}$$

Similarly,

$$\sum_{j=1}^{\lfloor \frac{n}{4} \rfloor} \binom{n-1}{2j-1} = \begin{cases} \frac{1}{2}\left(2^{n-2}\right) & \text{if } n \equiv 1 \bmod 4, \\ \frac{1}{2}\left(2^{n-2} - \binom{n-1}{\frac{n-1}{2}}\right) & \text{if } n \equiv 3 \bmod 4. \end{cases}$$

Hence, for n odd,

$$\sum_{j=0}^{\lfloor \frac{n}{4} \rfloor} (n-4j)\binom{n}{2j} = \frac{n}{2}\binom{n-1}{\frac{n-1}{2}}.$$

Thus, the theorem holds. □

THEOREM 4.5. *For any $n \geq 3$ and for any canonical $(n-1,n)$-threshold VCS the maximum contrast α is given by*

$$\alpha = \begin{cases} \left[\frac{n}{4}\binom{n}{\frac{n}{2}}\right]^{-1} & \textit{if } n \textit{ is even,} \\ \left[\frac{n}{2}\binom{n-1}{\frac{(n-1)}{2}}\right]^{-1} & \textit{if } n \textit{ is odd.} \end{cases}$$

Proof. In the proof of Lemma 4.3, to compute the value of the $h_{j,i}$'s of any contrast optimal $(n-1,n)$-threshold VCS, for n even, we set $\alpha \cdot m = 1$, whereas for n odd, we set $\alpha \cdot m = 2$. Therefore, applying Lemma 4.4 the theorem holds. □

It is worthwhile to notice that according to the previous lemma, in any contrast optimal $(n-1,n)$-threshold VCS, $\alpha = \Theta(2^{-n}n^{-1/2})$. This contrast is lower than an (n,n)-threshold VCS.

4.2. Contrast optimal $(3,n)$-threshold VCS. In this section we provide, for $n \geq 4$, a contrast optimal $(3,n)$-threshold VCS which is also strong and has its basis matrices in canonical form. We first describe a family of $(3,n)$-threshold VCS achieving various values of contrast and pixel expansion. Then, for any fixed $n \geq 4$, we determine the scheme in this family having the best contrast. Finally, we prove that the scheme has optimal contrast among all $(3,n)$-threshold VCS by proving an upper bound on the contrast of any $(3,n)$-threshold VCS.

For any $n \geq 4$ and any integer $1 \leq g < n/2$, consider the VCS whose basis matrices are in canonical form, denoted by $\mathcal{S}(3,n,g)$, described by the following $h_{j,i}$'s:

$$h_{0,0} = h_{n,1} = \binom{n-1}{g} - \binom{n-1}{g-1} \quad \text{and} \quad h_{n-g,0} = h_{g,1} = 1, \tag{4}$$

whereas all the remaining $h_{j,i}$'s are equal to zero. This is a strong $(3, n)$-threshold VCS, as shown by Theorem 4.7.

Example 4.6. If $n = 5$, then g can be either 1 or 2. Let $g = 1$. Then, $h_{0,0} = h_{5,1} = \binom{5-1}{1} - \binom{5-1}{1-1} = 3$, and $h_{4,0} = h_{1,1} = 1$. The corresponding basis matrices are

$$S^0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}, \qquad S^1 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Let $g = 2$. Then, $h_{0,0} = h_{5,1} = \binom{5-1}{2} - \binom{5-1}{2-1} = 2$, and $h_{3,0} = h_{2,1} = 1$. The corresponding basis matrices are

$$S^0 = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix},$$

$$S^1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

THEOREM 4.7. *For any $n \geq 4$ and any integer $1 \leq g < n/2$, the scheme $\mathcal{S}(3, n, g)$ described by* (4) *is a strong $(3, n)$-threshold VCS having pixel expansion and contrast equal to*

$$m = 2\binom{n-1}{g} \qquad \textit{and} \qquad \alpha = \frac{g(n-2g)}{2(n-1)(n-2)},$$

respectively.

Proof. Let $h_i = (h_{0,i}, \ldots, h_{n,i})$ for $i = 0, 1$, where the $h_{j,i}$'s are given by (4), and let $S_g(h_0)$ and $S_g(h_1)$ be binary matrices in which, for $i = 0, 1$, every binary n-tuple of weight j occurs exactly $h_{j,i}$ times as a column of $S_g(h_i)$. Then, $S_g(h_0)$ and $S_g(h_1)$ satisfy the conditions of Lemma 3.10, where

$$\alpha = \frac{\binom{n-3}{g-1} - \binom{n-3}{g-2}}{2\binom{n-1}{g}} = \frac{g(n-2g)}{2(n-1)(n-2)} \quad \text{and} \quad m = 2\binom{n-1}{g}. \tag{5}$$

Indeed, we can immediately verify that

$$\sum_{j=0}^{n} \binom{n}{j} h_{j,0} = \binom{n-1}{g} - \binom{n-1}{g-1} + \binom{n}{n-g} = 2\binom{n-1}{g}$$

and

$$\sum_{j=0}^{n} \binom{n}{j} h_{j,1} = \binom{n}{g} + \binom{n-1}{g} - \binom{n-1}{g-1} = 2\binom{n-1}{g}.$$

Hence, condition 1 of Lemma 3.10 is satisfied. Condition 2 is also satisfied because of

$$\sum_{j=0}^{n-1}\binom{n-1}{j}h_{j,0}=\binom{n-1}{g}-\binom{n-1}{g-1}+\binom{n-1}{n-g}=\binom{n-1}{g}=\sum_{j=0}^{n-1}\binom{n-1}{j}h_{j,1}$$

and

$$\begin{aligned}\sum_{j=0}^{n-2}\binom{n-2}{j}h_{j,0}&=\binom{n-1}{g}-\binom{n-1}{g-1}+\binom{n-2}{n-g}=\binom{n-1}{g}-\binom{n-2}{g-1}\\&=\binom{n-2}{g}=\sum_{j=0}^{n-2}\binom{n-2}{j}h_{j,1}.\end{aligned}$$

Now, we prove that condition 3 of Lemma 3.10 is satisfied, where α and m are as given by (5). We have that

$$\begin{aligned}\sum_{j=0}^{n-3}\binom{n-3}{j}(h_{j,0}-h_{j,1})&=\binom{n-1}{g}-\binom{n-1}{g-1}-\binom{n-3}{g}+\binom{n-3}{n-g}\\&=\binom{n-2}{g}+\binom{n-2}{g-1}-\binom{n-1}{g-1}-\binom{n-3}{g}+\binom{n-3}{g-3}\\&=\binom{n-3}{g-1}-\binom{n-2}{g-2}+\binom{n-3}{g-3}\\&=\binom{n-3}{g-1}-\binom{n-3}{g-2}=\alpha\cdot m.\end{aligned}$$

This proves that condition 3 of Lemma 3.10 holds.

Finally, we prove that the scheme $\mathcal{S}(3,n,g)$ is strong. For any $3\le\ell\le n$ and for any $Y\subseteq\{1,\dots,n\}$ such that $|Y|=\ell$, the number of zero columns in $S_g(h_0)[Y]$ ($S_g(h_1)[Y]$) does not depend on the particular set Y but only on its size ℓ since the basis matrices are in canonical form. Hence, we refer to such a quantity as χ_ℓ^0 (resp., χ_ℓ^1). We have that

$$\chi_\ell^0=\binom{n-1}{g}-\binom{n-1}{g-1}+\binom{n-\ell}{n-g}\quad\text{and}\quad\chi_\ell^1=\binom{n-\ell}{g}.$$

Notice that when $\ell>g$, then $\binom{n-\ell}{n-g}=0$, whereas $\binom{n-\ell}{g}=0$ when $g>n-\ell$. We define the function $\beta(\ell)$, for $3\le\ell\le n$, as $\beta(\ell)\overset{\triangle}{=}\chi_\ell^0-\chi_\ell^1$; that is,

$$\beta(\ell)=\binom{n-1}{g}-\binom{n-1}{g-1}+\binom{n-\ell}{n-g}-\binom{n-\ell}{g}.$$

To prove that the scheme is strong, it is enough to show that $\beta(\ell)\ge\alpha\cdot m$ for $3\le\ell\le n$. We next show that the function $\beta(\ell)$ is nondecreasing by proving that $\beta(\ell+1)-\beta(\ell)\ge 0$. Indeed, this difference can be written as

$$\begin{aligned}\beta(\ell+1)-\beta(\ell)&=\binom{n-\ell-1}{n-g}-\binom{n-\ell}{n-g}+\binom{n-\ell}{g}-\binom{n-\ell-1}{g}\\&=\binom{n-\ell-1}{g-1}-\binom{n-\ell-1}{n-g-1}\end{aligned}$$

$$= \binom{n-\ell-1}{g-1} - \binom{n-\ell-1}{g-\ell}.$$

Notice that if $\ell > g$, then $\binom{n-\ell-1}{g-\ell} = 0$ and $\beta(\ell+1) - \beta(\ell) \geq 0$. Assume $\ell = g$. Then $\beta(\ell+1) - \beta(\ell) = \binom{n-\ell-1}{g-1} - 1$. Since $g < n/2$ and $\ell = g$, then $g - 1 \leq n - \ell - 1$. Thus, $\beta(\ell+1) - \beta(\ell) \geq 0$. Finally, assume $\ell < g$. Then

$$\begin{aligned}\beta(\ell+1) - \beta(\ell) &= \frac{(n-\ell-1)!}{(g-1)! \cdot (n-\ell-g)!} - \frac{(n-\ell-1)!}{(g-\ell)! \cdot (n-g-1)!} \\ &= \frac{(n-\ell-1)!}{(g-\ell)! \cdot (n-\ell-g)!} \cdot \frac{\Pi_{j=1}^{\ell-1}(n-g-j) - \Pi_{j=1}^{\ell-1}(g-j)}{\Pi_{j=1}^{\ell-1}(n-g-j) \cdot (g-j)}.\end{aligned}$$

The above quantity is nonnegative, as $n - g - j \geq g - j$ for $g \leq n/2$. Therefore, the function $\beta(\ell)$ is a nondecreasing function. Hence, since $\beta(3) \geq \alpha \cdot m$, the scheme $\mathcal{S}(3, n, g)$ is strong. □

From the arguments used in the proof of the above theorem one can see that, by stacking together more than three transparencies from the scheme $\mathcal{S}(3, n, g)$, the image we recover becomes more visible (i.e., the difference between a white and a black pixel is larger when we stack together more than three transparencies). When we stack $n - g < \ell \leq n$ transparencies we have that $\beta(\ell) = \binom{n-1}{g} - \binom{n-1}{g-1}$. Since $m = 2\binom{n-1}{g}$, we get that the "contrast" in this case is equal to

$$\frac{\beta(\ell)}{m} = \frac{\binom{n-1}{g} - \binom{n-1}{g-1}}{2\binom{n-1}{g}} = \frac{n-2g}{2(n-g)}.$$

Notice that for fixed n, the contrast of the scheme given by Theorem 4.7 depends only on the parameter g. Hence, the scheme achieving the best contrast among the schemes $\mathcal{S}(3, n, g)$ is obtained by choosing the integer g in the interval $[1, n/2[$ in such a way that the quantity $(n-2g)g$ is maximized. For real g the function $(n-2g)g$ is convex $\cap$ and reaches its maximum at $g = n/4$. Since g has to be an integer, simple algebra shows that the quantity $(n-2g)g$ reaches its maximum at $g = \lfloor (n+1)/4 \rfloor$. Thus, for any $n \geq 4$, the following $h_{j,i}$'s describe a strong $(3, n)$-threshold VCS achieving the best contrast among the family of schemes $\mathcal{S}(3, n, g)$:

$$(6) \qquad h_{0,0} = h_{n,1} = \binom{n-1}{\lfloor \frac{n+1}{4} \rfloor} - \binom{n-1}{\lfloor \frac{n+1}{4} \rfloor - 1} \quad \text{and} \quad h_{n-\lfloor \frac{n+1}{4} \rfloor, 0} = h_{\lfloor \frac{n+1}{4} \rfloor, 1} = 1,$$

whereas all the remaining $h_{j,i}$'s are equal to zero. The contrast of the scheme described by the above $h_{j,i}$'s is equal to

$$(7) \qquad \frac{\left(n - 2\lfloor \frac{n+1}{4} \rfloor\right) \lfloor \frac{n+1}{4} \rfloor}{2(n-1)(n-2)}.$$

We now show that the schemes described by (6) are indeed a contrast optimal $(3, n)$-threshold VCS.

THEOREM 4.8. *Let $n \geq 4$. In any $(3, n)$-threshold VCS it holds that*

$$\alpha \leq \frac{\left(n - 2\lfloor \frac{n+1}{4} \rfloor\right) \lfloor \frac{n+1}{4} \rfloor}{2(n-1)(n-2)}.$$

Proof. Let S^0 and S^1 be the $n \times m$ basis matrices in canonical form of a $(3, n)$-threshold VCS with contrast α. Since our aim is to prove an upper bound on the contrast, we do not lose generality by considering basis matrices in such a form (see Lemma 3.6). Let $T = \{2, \dots, n\}$ and $Z_i = \{j : S^i[1][j] = 0\}$; that is, Z_i denotes the set of indices of columns of S^i having a zero as first entry. Finally, let $A^0 = S^0[T][Z_0]$ and $A^1 = S^1[T][Z_1]$. In other words, the pair of matrices $A = (A^0, A^1)$ is constituted by the submatrices of S^0 and S^1 obtained by removing all the columns having a 1 as a first entry and removing the first row. Hence, up to a column permutation, the basis matrices S^0 and S^1 are of the following form:

$$S^0 = \left[\begin{array}{c|c} 0\cdots 0 & 1\cdots 1 \\ \hline A^0 & B^0 \end{array}\right], \qquad S^1 = \left[\begin{array}{c|c} 0\cdots 0 & 1\cdots 1 \\ \hline A^1 & B^1 \end{array}\right],$$

where B^0 and B^1 are boolean matrices. It is known (see Theorem 6.1 and Corollary 6.2 of [5]) that A^0 and A^1 are basis matrices of a $(2, n-1)$-threshold VCS. Now, denote by $\alpha(A)$ the contrast of the $(2, n-1)$-threshold VCS with basis matrices (A^0, A^1). Since by Corollary 3.8 $m = 2w(S^0[1])$, it is easy to see that the contrast α of the scheme represented by (S^0, S^1) is equal to

$$\alpha = \frac{\alpha(A)}{2}, \tag{8}$$

while the pixel expansion is equal to $m = 2m'$, where m' is the pixel expansion of the scheme having basis matrices (A^0, A^1), that is, $m' = \sum_{j \in J} h_{j,1} \binom{n-1}{j}$, where J is the set of indices j for which $h_{j,1} > 0$ and $j < n$ in A^1. Let X be a set of two rows. We have that

$$\alpha(A) \le \frac{w(A^1_X) - w(A^0_X)}{m'}.$$

Since $w(A^1[i]) = w(A^0[i])$, for $i = 1, \dots, n-1$, we have that $w(A^1_X) - w(A^0_X)$ is equal to the number of columns $[\begin{smallmatrix}1\\0\end{smallmatrix}]$ in $A^1[X]$ minus the number of columns $[\begin{smallmatrix}1\\0\end{smallmatrix}]$ in $A^0[X]$. Therefore, we get that

$$\begin{aligned} w(A^1_X) - w(A^0_X) &= \sum_{j \in J} \left[h_{j,1} \binom{n-3}{j-1} - h_{n-j,0} \binom{n-3}{n-j-1} \right] \\ &= \sum_{j \in J} h_{j,1} \left[\binom{n-3}{j-1} - \binom{n-3}{j-2} \right]. \end{aligned}$$

Hence,

$$\alpha(A) \le \frac{\displaystyle\sum_{j \in J} h_{j,1} \left[\binom{n-3}{j-1} - \binom{n-3}{j-2} \right]}{\displaystyle\sum_{j \in J} h_{j,1} \binom{n-1}{j}}. \tag{9}$$

Notice that for any function $g(x)$, for any positive function $f(x)$, and for any nonempty set D which is a subset of both functions' domain, it holds that

$$\frac{\displaystyle\sum_{x \in D} g(x)}{\displaystyle\sum_{x \in D} f(x)} \le \max_{x \in D} \frac{g(x)}{f(x)}.$$

Therefore, since $J \subseteq \{0, \ldots, n-1\}$, we have that

$$\frac{\sum_{j\in J} h_{j,1}\left[\binom{n-3}{j-1} - \binom{n-3}{j-2}\right]}{\sum_{j\in J} h_{j,1}\binom{n-1}{j}} \leq \max_{j\in J}\frac{\binom{n-3}{j-1} - \binom{n-3}{j-2}}{\binom{n-1}{j}} = \max_{j\in J}\frac{(n-2j)j}{(n-1)(n-2)}.$$

We have already seen earlier in this section that the function $(n-2j)j$ reaches its maximum over the integers $j \in \{0, \ldots, n-1\}$ at $j = \lfloor (n+1)/4 \rfloor$. Therefore,

$$\alpha(A) \leq \frac{\left(n - 2\lfloor\frac{n+1}{4}\rfloor\right)\lfloor\frac{n+1}{4}\rfloor}{(n-1)(n-2)}.$$

The theorem then follows by (8). □

Let $\alpha_3(n)$ be the expression (7). It is easy to see that $\lim_{n\to\infty}\alpha_3(n) = 1/16$. Therefore, the construction for $(3,n)$-threshold VCS given at the end of section 5 in [5] has nearly optimal contrast asymptotically, as well as a small pixel expansion.

5. A canonical $(4, n)$-threshold VCS.

In this section we provide, for $n \geq 4$, a class of strong $(4,n)$-threshold VCS whose basis matrices are in canonical form. We first describe a family of $(4,n)$-threshold VCS achieving various values of contrast and pixel expansion. Then, for any fixed $n \geq 4$, we determine the scheme in this family having the best contrast.

For any even $n \geq 4$ and any integer $1 \leq g < n/2$, consider the VCS whose basis matrices are in canonical form, denoted by $\mathcal{S}(4,n,g)$, described by the following $h_{j,i}$'s:

$$\begin{aligned} & h_{0,0} = h_{n,0} = \binom{n-3}{n/2-1}\frac{t_{n,g}(n-1)(n-2g)^2}{ng(n-g)}, \\ (10) \qquad & h_{n/2,0} = t_{n,g} \quad \text{and} \quad h_{g,1} = h_{n-g,1} = \frac{\binom{n-3}{n/2-1}}{\binom{n-2}{g-1}} \cdot t_{n,g}, \end{aligned}$$

where $t_{n,g} = \binom{n-2}{g-1} / \gcd\{\binom{n-2}{g-1}, \binom{n-3}{n/2-1}\}$ and all the remaining $h_{j,i}$'s are equal to zero. This is a strong $(4,n)$-threshold VCS as shown by the following theorem.

THEOREM 5.1. *For any even integer $n \geq 4$ and any integer $1 \leq g < n/2$, the scheme $\mathcal{S}(4,n,g)$ is a strong $(4,n)$-threshold VCS having pixel expansion and contrast equal to*

$$m = \frac{2nt_{n,g}(n-1)}{g(n-g)}\binom{n-3}{n/2-1} \quad \text{and} \quad \alpha = \frac{g(n-g)(n-2g)^2}{4n(n-1)(n-2)(n-3)},$$

respectively.

Proof. Let $h_i = (h_{0,i}, \ldots, h_{n,i})$, for $i = 0, 1$, where the $h_{j,i}$'s are given by (10), and let $S_g(h_0)$ and $S_g(h_1)$ be binary matrices in which, for $i = 0, 1$, every binary n-tuple of weight j occurs exactly $h_{j,i}$ times as a column of $S_g(h_i)$. Then, $S_g(h_0)$ and $S_g(h_1)$ satisfy the conditions of Lemma 3.10. Indeed, we immediately verify that

$$\begin{aligned} \sum_{j=0}^{n}\binom{n}{j}h_{j,0} &= \left[2\binom{n-3}{n/2-1}\frac{(n-1)(n-2g)^2}{ng(n-g)} + \binom{n}{n/2}\right]t_{n,g} \\ &= \binom{n-3}{n/2-1}\left[\frac{2(n-1)(n-2g)^2}{ng(n-g)} + \frac{8n(n-1)(n-2)}{n^2(n-2)}\right]t_{n,g} \end{aligned}$$

$$= \frac{2nt_{n,g}(n-1)}{g(n-g)}\binom{n-3}{n/2-1}$$

and

$$\sum_{j=0}^{n}\binom{n}{j}h_{j,1} = \frac{\binom{n-3}{n/2-1}}{\binom{n-2}{g-1}}\binom{n}{g}t_{n,g} + \frac{\binom{n-3}{n/2-1}}{\binom{n-2}{g-1}}\binom{n}{n-g}t_{n,g} = \frac{2nt_{n,g}(n-1)}{g(n-g)}\binom{n-3}{n/2-1}.$$

Hence, condition 1 of Lemma 3.10 is satisfied. To prove that condition 2 of Lemma 3.10 is satisfied we have to show that, for $\ell = 1, 2, 3$, the following identity holds:

$$\sum_{j=0}^{n-\ell}\binom{n-\ell}{j}h_{j,0} = \sum_{j=0}^{n-\ell}\binom{n-\ell}{j}h_{j,1}. \tag{11}$$

Notice that, for $\ell = 1$, we have

$$\begin{aligned}\sum_{j=0}^{n-\ell}\binom{n-\ell}{j}h_{j,0} &= t_{n,g}\binom{n-3}{n/2-1}\frac{(n-1)(n-2g)^2}{ng(n-g)} + t_{n,g}\binom{n-1}{n/2}\\ &= t_{n,g}\binom{n-3}{n/2-1}\left[\frac{(n-1)(n-2g)^2}{ng(n-g)} + \frac{4(n-1)}{n}\right]\\ &= t_{n,g}\binom{n-3}{n/2-1}\frac{n(n-1)}{g(n-g)}\end{aligned}$$

and

$$\begin{aligned}\sum_{j=0}^{n-\ell}\binom{n-\ell}{j}h_{j,1} &= \frac{t_{n,g}\binom{n-3}{n/2-1}}{\binom{n-2}{g-1}}\left[\binom{n-1}{g} + \binom{n-1}{g-1}\right]\\ &= \frac{t_{n,g}\binom{n-3}{n/2-1}\binom{n}{g}}{\binom{n-2}{g-1}}\\ &= t_{n,g}\binom{n-3}{n/2-1}\frac{n(n-1)}{g(n-g)}.\end{aligned}$$

Therefore, for $\ell = 1$, we have that identity (11) holds. (The cases $\ell = 2$ and $\ell = 3$ are considered in Appendix A.) Now we prove that Condition 3 of Lemma 3.10, where α and m are as given by (5), that is,

$$\sum_{j=0}^{n-4}\binom{n-4}{j}(h_{j,0} - h_{j,1}) = \frac{t_{n,g}(n-2g)^2}{2(n-2)(n-3)}\binom{n-3}{n/2-1}, \tag{12}$$

is satisfied. We have that

$$\begin{aligned}&\sum_{j=0}^{n-4}\binom{n-4}{j}(h_{j,0} - h_{j,1})\\ &\quad = t_{n,g}\binom{n-3}{n/2-1}\left[\frac{(n-1)(n-2g)^2}{ng(n-g)} + \frac{\binom{n-4}{n/2}}{\binom{n-3}{n/2-1}} - \frac{\binom{n-4}{g} + \binom{n-4}{n-g}}{\binom{n-2}{g-1}}\right]\end{aligned}$$

$$\begin{aligned}&= t_{n,g}\binom{n-3}{n/2-1}\left[\frac{(n-1)(n-2)^2}{ng(n-g)} + \frac{(n-4)(n-6)}{2n(n-3)}\right.\\&\qquad\left. - \frac{(n-g-1)(n-g-2)(n-g-3)}{g(n-2)(n-3)} - \frac{(g-1)(g-2)(g-3)}{(n-g)(n-2)(n-3)}\right]\\&= \frac{t_{n,g}(n-2g)^2}{2(n-2)(n-3)}\binom{n-3}{n/2-1}.\end{aligned}$$

This proves that condition 3 of Lemma 3.10 holds.

Finally, we prove that the scheme $\mathcal{S}(4,n,g)$ is strong. For any $4 \le \ell \le n$ and for any $Y \subseteq \{1,\ldots,n\}$ such that $|Y| = \ell$, the number of zero columns in $S_g(h_0)[Y]$ ($S_g(h_1)[Y]$) does not depend on the particular set Y but only on its size ℓ since the basis matrices are in canonical form. Hence, we refer to such a quantity as χ_ℓ^0 (χ_ℓ^1). We have that

$$\chi_\ell^0 = \frac{t_{n,g}(n-1)(n-2g)^2}{ng(n-g)}\binom{n-3}{n/2-1} + t_{n,g}\binom{n-\ell}{n/2}$$

and

$$\chi_\ell^1 = t_{n,g}\frac{\binom{n-3}{n/2-1}}{\binom{n-2}{g-1}}\left[\binom{n-\ell}{g} + \binom{n-\ell}{n-g}\right].$$

Notice that when $\ell > g$, then $\binom{n-\ell}{n-g} = 0$, whereas $\binom{n-\ell}{g} = 0$ when $g > n-\ell$. We define the function $\beta(\ell)$, for $4 \le \ell \le n$, as $\beta(\ell) \triangleq \chi_\ell^0 - \chi_\ell^1$; that is,

$$\beta(\ell) = \frac{t_{n,g}(n-1)(n-2g)^2}{ng(n-g)}\binom{n-3}{n/2-1} + t_{n,g}\binom{n-\ell}{n/2} - t_{n,g}\frac{\binom{n-3}{n/2-1}}{\binom{n-2}{g-1}}\left[\binom{n-\ell}{g} + \binom{n-\ell}{n-g}\right].$$

To prove that the scheme is strong it is enough to show that $\beta(\ell) \ge \alpha\cdot m$ for $4 \le \ell \le n$. Next we show that the function $\beta(\ell)$ is nondecreasing by proving that $\beta(\ell+1)-\beta(\ell) \ge 0$. Indeed, this difference can be written as

$$\beta(\ell+1)-\beta(\ell) = \left[\binom{n-\ell-1}{g-1} + \binom{n-\ell-1}{n-g-1}\right]\frac{\binom{n-3}{n/2-1}}{\binom{n-2}{g-1}}t_{n,g} - \binom{n-\ell-1}{n/2-1}t_{n,g}.$$

Assume $\ell \le g$. Then, after some algebra, to prove $\beta(\ell+1)-\beta(\ell) \ge 0$ is equivalent to proving that

$$\frac{\displaystyle\prod_{j=1}^{\ell-1}(n-g-j) + \prod_{j=1}^{\ell-1}(g-j) - 2\prod_{j=1}^{\ell-1}\left(\frac{n}{2}-j\right)}{\displaystyle 2\prod_{j=1}^{\ell-1}\left(\frac{n}{2}-j\right)} \ge 0.$$

Since $j < \ell \le g < n/2$, we have that the denominator is positive. Therefore, we have to show that the numerator is nonnegative. To this aim we need some definitions and properties of combinatorial quantities (see [8, pp. 47–48]). For any integer $s \ge 0$ and real x, the *rising factorial power* $x^{\overline{s}}$ is defined as $x^{\overline{s}} = x(x+1)\cdots(x+s-1)$. The rising factorial power is strictly related to the *Stirling numbers of first kind*. For any integers n and k such that $n \geq k \geq 0$ and $n > 0$, the Stirling numbers of first kind, denoted by $\left[{n \atop k}\right]$, count the number of ways to arrange n objects into k cycles and are defined as

$$\left[\begin{array}{c} n \\ k \end{array}\right] = (n-1)\left[\begin{array}{c} n-1 \\ k \end{array}\right] + \left[\begin{array}{c} n-1 \\ k-1 \end{array}\right] \quad \text{with} \quad \left[\begin{array}{c} 0 \\ 0 \end{array}\right] = 1 \quad \text{and} \quad \left[\begin{array}{c} n \\ 0 \end{array}\right] = 0.$$

The Stirling numbers of first kind and the rising factorial powers are related by

$$x^{\overline{n}} = \sum_{k=0}^{n} \left[\begin{array}{c} n \\ k \end{array}\right] x^k.$$

Using the rising factorial powers and the above identity, we have that

$$\begin{aligned}
&\prod_{j=1}^{\ell-1}(n-g-j) + \prod_{j=1}^{\ell-1}(g-j) - 2\prod_{j=1}^{\ell-1}\left(\frac{n}{2}-j\right) \\
&\qquad = (n-g-\ell+1)^{\overline{\ell-1}} - 2\left(\frac{n}{2}-\ell+1\right)^{\overline{\ell-1}} + (g-\ell+1)^{\overline{\ell-1}} \\
&\qquad = \sum_{p=1}^{\ell-1}\left[\begin{array}{c} \ell-1 \\ p \end{array}\right](n-g-\ell+1)^p - 2\sum_{p=1}^{\ell-1}\left[\begin{array}{c} \ell-1 \\ p \end{array}\right]\left(\frac{n}{2}-\ell+1\right)^p \\
&\qquad\qquad + \sum_{p=1}^{\ell-1}\left[\begin{array}{c} \ell-1 \\ p \end{array}\right](g-\ell+1)^p \\
&\qquad = \sum_{p=1}^{\ell-1}\left[\begin{array}{c} \ell-1 \\ p \end{array}\right]\left((n-g-\ell+1)^p - 2\left(\frac{n}{2}-\ell+1\right)^p + (g-\ell+1)^p\right).
\end{aligned}$$

By induction on p, we immediately see that $(n-g-\ell+1)^p - 2(n/2-\ell+1)^p + (g-\ell+1)^p \geq 0$. Indeed, setting $a = n/2-\ell+1$ and $d = n/2-g$, we have to prove that $(a+d)^p - 2a^p + (a-d)^p \geq 0$. (Notice that $a > d > 0$.) For $p = 1$, the basis of the induction is true. By an inductive hypothesis, assume that $(a+d)^p - 2a^p + (a-d)^p \geq 0$ for some $p \geq 1$. We have that

$$\begin{aligned}
(a+d)^{p+1} + (a-d)^{p+1} &= a[(a+d)^p + (a-d)^p] + d[(a+d)^p - (a-d)^p] \\
&\geq a[(a+d)^p + (a-d)^p] \\
&\geq 2a^{p+1} \quad \text{(by the inductive hypothesis).}
\end{aligned}$$

Hence, for $\ell \leq g$, we have that $\beta(\ell+1) - \beta(\ell) \geq 0$.

Assume now $g < \ell \leq n/2$. Then, to prove that $\beta(\ell+1) - \beta(\ell) \geq 0$ is equivalent to proving that

$$\frac{\displaystyle\prod_{j=1}^{\ell-1}(n-g-j) - 2\prod_{j=1}^{\ell-1}\left(\frac{n}{2}-j\right)}{\displaystyle 2\prod_{j=1}^{\ell-1}\left(\frac{n}{2}-j\right)} \geq 0.$$

Since $j < \ell \leq n/2$, we have that the denominator of the above expression is a positive quantity, while the numerator can be written as

$$
\begin{aligned}
&(n-g-\ell+1)^{\overline{\ell-1}} - 2(\tfrac{n}{2}-\ell+1)^{\overline{\ell-1}}\\
&\qquad = \sum_{p=1}^{\ell-1} \begin{bmatrix} \ell-1 \\ p \end{bmatrix} (n-g-\ell+1)^p - 2\sum_{p=1}^{\ell-1} \begin{bmatrix} \ell-1 \\ p \end{bmatrix} \left(\frac{n}{2}-\ell+1\right)^p\\
&\qquad = \sum_{p=1}^{\ell-1} \begin{bmatrix} \ell-1 \\ p \end{bmatrix} \left((n-g-\ell+1)^p - 2\left(\frac{n}{2}-\ell+1\right)^p\right).
\end{aligned}
$$

By induction on p, one can see that $(n-g-\ell+1)^p - 2(n/2-\ell+1)^p \geq 0$. Therefore, for $g < \ell \leq n/2$ we have that $\beta(\ell+1) - \beta(\ell) \geq 0$.

Finally, assume that $\ell > n/2$. Then,

$$\beta(\ell+1) - \beta(\ell) = \frac{t_{_{n,g}} \cdot \binom{n-\ell-1}{g-1}\binom{n-3}{n/2-1}}{\binom{n-2}{g-1}} \geq 0.$$

Therefore, the function $\beta(\ell)$ is nondecreasing. Hence, since $\beta(4) \geq \alpha \cdot m$, the scheme $\mathcal{S}(4,n,g)$ is strong. □

Notice that, for fixed n, the contrast of the scheme given by Theorem 5.1 depends only on the parameter g. Hence, for fixed n, the scheme achieving the best contrast among the schemes $\mathcal{S}(4,n,g)$ is obtained by choosing the integer g in the interval $[1, n/2[$ in such a way that the quantity

$$\alpha_4(g,n) = \frac{g(n-g)(n-2g)^2}{4n(n-1)(n-2)(n-3)}$$

is maximized. For real g and for fixed n, simple algebra shows that the function $g(n-g)(n-2g)^2$, with $g \in [1, n/2[$, is convex $\cap$ and reaches its maximum at $g = (2-\sqrt{2})n/4$. Since g has to be an integer, we have that g can be either equal to $\lfloor (2-\sqrt{2})n/4 \rfloor$ or equal to $\lceil (2-\sqrt{2})n/4 \rceil$. For any fixed $n \geq 4$, let $g_n \in \left\{ \lfloor (2-\sqrt{2})n/4 \rfloor, \lceil (2-\sqrt{2})n/4 \rceil \right\}$ be the integer which maximizes $\alpha_4(g,n)$. One can easily see that $\lim_{n\to\infty} \alpha_4(g_n,n) = 1/64$.

Remark 5.2. Theorem 5.1 holds only when n is even. If n is odd, then, by applying the technique given in Theorem 5.1, we construct a $(4, n+1)$-threshold VCS, and then we consider only the first n rows of the basis matrices of such a scheme. Therefore, for any odd $n \geq 4$ and any integer $1 \leq g < n/2$, there exists a strong $(4,n)$-threshold VCS having pixel expansion and contrast equal to

$$m = \frac{2nt_{_{n,g}}(n+1)}{g(n+1-g)} \binom{n+1-3}{(n+1)/2-1} \quad \text{and} \quad \alpha = \frac{g(n+1-g)(n+1-2g)^2}{4n(n+1)(n-1)(n-2)},$$

respectively.

6. A canonical $(5,n)$-threshold VCS. In this section we provide, for $n \geq 5$, a class of $(5,n)$-threshold VCS whose basis matrices are in canonical form. Similarly to the previous cases, we first describe a family of $(5,n)$-threshold VCS achieving various values of contrast and pixel expansion. Then, for any fixed $n \geq 5$ we determine which scheme in this family has the best contrast.

For any two integers ℓ and g such that $1 \leq \ell < g < n/2$, the $(5,n)$-threshold VCS whose basis matrices are in canonical form, denoted by $\mathcal{S}(5,n,\ell,g)$, is described by the following $h_{j,i}$'s:

$$h_{g,0} = h_{n-g,1} = t_{_{(n,\ell,g)}}, \quad h_{n-\ell,0} = h_{\ell,1} = s_{_{(n,\ell,g)}}, \quad \text{and} \quad h_{0,0} = h_{n,1} = r_{_{(n,\ell,g)}}, \tag{13}$$

where

$$t_{(n,\ell,g)} = \frac{\binom{n-4}{\ell-1} - \binom{n-4}{\ell-3}}{\gcd\left\{\binom{n-4}{\ell-1} - \binom{n-4}{\ell-3}, \binom{n-4}{g-1} - \binom{n-4}{g-3}\right\}}, \quad s_{(n,\ell,g)} = t_{(n,\ell,g)} \frac{\left[\binom{n-4}{g-1} - \binom{n-4}{g-3}\right]}{\left[\binom{n-4}{\ell-1} - \binom{n-4}{\ell-3}\right]},$$

$$r_{(n,\ell,g)} = s_{(n,\ell,g)} \left[\binom{n-4}{\ell} - \binom{n-4}{\ell-4}\right] - t_{(n,\ell,g)} \left[\binom{n-4}{g} - \binom{n-4}{g-4}\right],$$

and all the remaining $h_{j,i}$'s are equal to zero.

THEOREM 6.1. *For any two integers ℓ and g such that $1 \leq \ell < g < n/2$, the scheme $\mathcal{S}(5, n, \ell, g)$ is a canonical $(5, n)$-threshold VCS having pixel expansion and contrast equal to*

$$m = s_{(n,\ell,g)} \left[\binom{n}{\ell} + \binom{n-4}{\ell} - \binom{n-4}{\ell-4}\right] + t_{(n,\ell,g)} \left[\binom{n}{g} + \binom{n-4}{g-4} - \binom{n-4}{g}\right]$$

and

$$\alpha = \frac{\ell(g-\ell)(n-g)(n-2g)(n-2\ell)}{2(n+2\ell-2g)(n-1)(n-2)(n-3)(n-4)},$$

respectively.

Proof. It is easy to see that condition 1 of Lemma 3.10 is satisfied, as the basis matrices of the scheme $\mathcal{S}(5, n, \ell, g)$ are complements of each other. To prove that condition 2 of Lemma 3.10 is satisfied we have to show that, for $1 \leq q \leq 4$, the following equality holds:

$$\sum_{j=0}^{n-q} \binom{n-q}{j} h_{j,1} = \sum_{j=0}^{n-q} \binom{n-q}{j} h_{j,0}. \tag{14}$$

We have that

$$\sum_{j=0}^{n-q} \binom{n-q}{j} h_{j,1} = t_{(n,\ell,g)} \binom{n-q}{\ell} \frac{\binom{n-4}{g-1} - \binom{n-4}{g-3}}{\binom{n-4}{\ell-1} - \binom{n-4}{\ell-3}} + t_{(n,\ell,g)} \binom{n-q}{n-g}$$

and

$$\begin{aligned}\sum_{j=0}^{n-q} \binom{n-q}{j} h_{j,0} = t_{(n,\ell,g)} &\frac{\binom{n-4}{g-1} - \binom{n-4}{g-3}}{\binom{n-4}{\ell-1} - \binom{n-4}{\ell-3}} \left[\binom{n-4}{\ell} - \binom{n-4}{\ell-4} + \binom{n-q}{n-\ell}\right] \\ &- t_{(n,\ell,g)} \left[\binom{n-4}{g} - \binom{n-4}{g-4}\right] + t_{(n,\ell,g)} \binom{n-q}{g}.\end{aligned}$$

Therefore, equality (14) is satisfied if and only if the quantity

$$A(n,\ell,g) \triangleq \frac{\binom{n-4}{g-1} - \binom{n-4}{g-3}}{\binom{n-4}{\ell-1} - \binom{n-4}{\ell-3}} \left[\binom{n-4}{\ell} - \binom{n-4}{\ell-4}\right] - \left[\binom{n-4}{g} - \binom{n-4}{g-4}\right] \tag{15}$$

is equal to

$$B(n,\ell,g,q) \triangleq \frac{\binom{n-4}{g-1} - \binom{n-4}{g-3}}{\binom{n-4}{\ell-1} - \binom{n-4}{\ell-3}} \left[\binom{n-q}{\ell} - \binom{n-q}{n-\ell}\right] - \left[\binom{n-q}{g} - \binom{n-q}{n-g}\right]. \tag{16}$$

If we substitute q for 4 in (16) we get expression (15). Therefore, (14) is satisfied for $q = 4$. We will prove that equality (14) holds when $q = 1$ and $4 \leq \ell < g$. (The remaining cases are analyzed in Appendix A.)

Note that $A(n, \ell, g)$ can be written as

$$\frac{\binom{n-4}{g-3}\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1\right]}{\binom{n-4}{\ell-3}\left[\frac{(n-\ell-1)(n-\ell-2)}{(\ell-1)(\ell-2)} - 1\right]}\binom{n-4}{\ell-3}\left[\frac{(n-\ell-1)(n-\ell-2)(n-\ell-3)}{\ell(\ell-1)(\ell-2)} - \frac{\ell-3}{n-\ell}\right]$$
$$-\binom{n-4}{g-3}\left[\frac{(n-g-1)(n-g-2)(n-g-3)}{g(g-1)(g-2)} - \frac{g-3}{n-g}\right]$$

which is equal to

$$\binom{n-4}{g-3}\left\{\frac{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1\right]\left[\frac{(n-\ell-1)(n-\ell-2)(n-\ell-3)}{\ell(\ell-1)(\ell-2)} - \frac{\ell-3}{n-\ell}\right]}{\left[\frac{(n-\ell-1)(n-\ell-2)}{(\ell-1)(\ell-2)} - 1\right]}\right.$$
$$\left. - \frac{(n-g-1)(n-g-2)(n-g-3)}{g(g-1)(g-2)} + \frac{g-3}{n-g}\right\}.$$

After some algebra, we get that the above expression is reduced to

$$\binom{n-4}{g-3}\frac{(n-1)(n-2)(n-3)(n-2g)(g-\ell)(n-\ell-g)}{g\ell(g-1)(g-2)(n-\ell)(n-g)}. \tag{17}$$

We can rewrite $B(n, \ell, g, 1)$ as

$$\frac{\binom{n-4}{g-3}\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1\right]}{\binom{n-4}{\ell-3}\left[\frac{(n-\ell-1)(n-\ell-2)}{(\ell-1)(\ell-2)} - 1\right]}\binom{n-4}{\ell-3}\left[\frac{(n-1)(n-2)(n-3)}{\ell(\ell-1)(\ell-2)} - \frac{(n-1)(n-2)(n-3)}{(\ell-1)(\ell-2)(n-\ell)}\right]$$
$$-\binom{n-4}{g-3}\left[\frac{(n-1)(n-2)(n-3)}{g(g-1)(g-2)} - \frac{(n-1)(n-2)(n-3)}{(g-1)(g-2)(n-g)}\right]$$

which is equal to

$$\binom{n-4}{g-3}\left\{\frac{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1\right]\left[\frac{(n-1)(n-2)(n-3)}{\ell(\ell-1)(\ell-2)} - \frac{(n-1)(n-2)(n-3)}{(\ell-1)(\ell-2)(n-\ell)}\right]}{\left[\frac{(n-\ell-1)(n-\ell-2)}{(\ell-1)(\ell-2)} - 1\right]}\right.$$
$$\left. - \left[\frac{(n-1)(n-2)(n-3)}{g(g-1)(g-2)} - \frac{(n-1)(n-2)(n-3)}{(g-1)(g-2)(n-g)}\right]\right\}.$$

Simple algebra shows that the above expression reduces to (17). Therefore, when $q = 1$ and $4 \leq \ell < g$ equality (14) is satisfied.

To prove that condition 3 of Lemma 3.10 is satisfied we have to show that

$$\sum_{j=0}^{n-5}\binom{n-5}{j}(h_{j,0} - h_{j,1}) = \alpha \cdot m. \tag{18}$$

We show that (18) holds for $4 \leq \ell < g$. (The cases $1 \leq \ell < g \leq 3$ and $1 \leq \ell \leq 3$ with $g \geq 4$ are considered in Appendix A.) We see immediately that (18) is satisfied if and only if

$$\frac{\displaystyle\sum_{j=0}^{n-5}\binom{n-5}{j}(h_{j,0}-h_{j,1})}{m} = \frac{\ell(g-\ell)(n-g)(n-2g)(n-2\ell)}{2(n+2\ell-2g)(n-1)(n-2)(n-3)(n-4)}. \tag{19}$$

We have that

$$\frac{\displaystyle\sum_{j=0}^{n-5}\binom{n-5}{j}(h_{j,0}-h_{j,1})}{m}$$

$$= \frac{s_{(n,\ell,g)}\left[\binom{n-4}{\ell}-\binom{n-4}{n-\ell}+\binom{n-5}{n-\ell}-\binom{n-5}{\ell}\right]-t_{(n,\ell,g)}\left[\binom{n-4}{g}-\binom{n-4}{n-g}+\binom{n-5}{g}-\binom{n-5}{n-g}\right]}{s_{(n,\ell,g)}\left[\binom{n}{\ell}+\binom{n-4}{\ell}-\binom{n-4}{\ell-4}\right]+t_{(n,\ell,g)}\left[\binom{n}{g}+\binom{n-4}{g-4}-\binom{n-4}{g}\right]}$$

$$= \frac{s_{(n,\ell,g)}\left[\binom{n-5}{\ell-1}-\binom{n-5}{\ell-4}\right]-t_{(n,\ell,g)}\left[\binom{n-5}{g-1}-\binom{n-5}{g-4}\right]}{s_{(n,\ell,g)}\left[\binom{n}{\ell}+\binom{n-4}{\ell}-\binom{n-4}{\ell-4}\right]+t_{(n,\ell,g)}\left[\binom{n}{g}+\binom{n-4}{g-4}-\binom{n-4}{g}\right]}.$$

Let

$$a \triangleq s_{(n,\ell,g)}\left[\binom{n-5}{\ell-1}-\binom{n-5}{\ell-4}\right], \qquad b \triangleq t_{(n,\ell,g)}\left[\binom{n-5}{g-1}-\binom{n-5}{g-4}\right],$$

$$c \triangleq s_{(n,\ell,g)}\left[\binom{n}{\ell}+\binom{n-4}{\ell}-\binom{n-4}{\ell-4}\right], \qquad d \triangleq t_{(n,\ell,g)}\left[\binom{n}{g}+\binom{n-4}{g-4}-\binom{n-4}{g}\right].$$

It is easy to check that the following three equalities hold:

$$\frac{a}{c} = \frac{\ell(n-2\ell)(n^2-n\ell-6n+\ell^2+11)}{2(n-3)(n-4)(n^2-n\ell-3n+2\ell^2+2)},$$

$$\frac{b}{a} = \frac{(n^2-ng-6n+g^2+11)}{(n^2-n\ell-6n+\ell^2+11)},$$

$$\frac{d}{c} = \frac{\ell(n-2\ell)(2n^2-3ng-3n+2g^2+2)}{(n-2g)(n-g)(n^2-n\ell-3n+2\ell^2+2)}.$$

Since $(a-b)/(c+d) = \frac{a(1-b/a)}{c(1+d/c)}$ we have that

$$\frac{s_{(n,\ell,g)}\left[\binom{n-5}{\ell-1}-\binom{n-5}{\ell-4}\right]-t_{(n,\ell,g)}\left[\binom{n-5}{g-1}-\binom{n-5}{g-4}\right]}{s_{(n,\ell,g)}\left[\binom{n}{\ell}+\binom{n-4}{\ell}-\binom{n-4}{\ell-4}\right]+t_{(n,\ell,g)}\left[\binom{n}{g}+\binom{n-4}{g-4}-\binom{n-4}{g}\right]}$$

can be rewritten as

$$\frac{\ell(n-2\ell)(n^2-n\ell-6n+\ell^2+11)}{2(n-3)(n-4)(n^2-n\ell-3n+2\ell^2+2)} \cdot \left[\frac{1 \quad - \quad \frac{(n^2-ng-6n+g^2+11)}{(n^2-n\ell-6n+\ell^2+11)}}{1 \quad + \quad \frac{\ell(n-2\ell)(2n^2-3ng-3n+2g^2+2)}{(n-2g)(n-g)(n^2-n\ell-3n+2\ell^2+2)}}\right].$$

It is simple, but tedious, to check that the previous expression reduces to

$$\frac{\ell(g-\ell)(n-g)(n-2g)(n-2\ell)}{2(n+2\ell-2g)(n-1)(n-2)(n-3)(n-4)}.$$

Therefore, (18) is satisfied and the theorem holds. □

Notice that for fixed n, the contrast of the scheme given in the above theorem depends only on the parameters ℓ and g. Therefore, if we want to obtain from the construction given by Theorem 6.1 the scheme achieving the best contrast, we have to choose, for a fixed n, the integers ℓ and g, where $1 \leq \ell < g < n/2$, in such a way that the quantity

$$\alpha_5(\ell, g, n) = \frac{\ell(g-\ell)(n-g)(n-2g)(n-2\ell)}{2(n+2\ell-2g)(n-1)(n-2)(n-3)(n-4)}$$

is maximized. Choosing ℓ and g proportional to n, setting $\ell = \gamma \cdot n$ and $g = \delta \cdot n$, where γ and δ are constants to be determined later such that $0 < \gamma < \delta < 1$, we have that

$$\alpha_5(\gamma \cdot n, \delta \cdot n, n) = \frac{\gamma(\delta-\gamma)(1-\delta)(1-2\delta)(1-2\gamma)n^5}{2(1+2\gamma-2\delta)n(n-1)(n-2)(n-3)(n-4)}.$$

One can easily see that

$$\lim_{n\to\infty} \alpha_5(\gamma \cdot n, \delta \cdot n, n) = \frac{\gamma(\delta-\gamma)(1-\delta)(1-2\delta)(1-2\gamma)}{2(1+2\gamma-2\delta)}.$$

For real γ and δ, with $0 < \gamma < \delta < 1$, by using the system Mathematica, we see that, for fixed n, the function $\gamma(\delta-\gamma)(1-\delta)(1-2\delta)(1-2\gamma)/2(1+2\gamma-2\delta)$ reaches its maximum at $(\overline{\gamma}, \overline{\delta}) = (0.0954913, 0.345492)$ and the above limit is equal to

$$\lim_{n\to\infty} \alpha_5(\overline{\gamma} \cdot n, \overline{\delta} \cdot n, n) = \frac{1}{256}.$$

Therefore, there are $(5, n)$-threshold VCS that, for large n, have contrast of almost 1/256.

7. Conclusion.

In this paper we have analyzed the contrast of the reconstructed image for (k, n)-threshold VCS. We have defined a canonical form for such VCS and we have also provided a characterization of (k, n)-threshold VCS. Several open problems arise. For instance, we conjecture that the (k, n)-threshold VCS, for $k = 4$ and $k = 5$, has an optimal contrast. Moreover, further research could be done in finding a closed formula for the optimal contrast for general (k, n)-threshold VCS.

Appendix A.

In the following we show the computation omitted from the proof of Theorem 5.1.

(i) *Proof that equality* (11) *in Theorem* 5.1 *is satisfied for the case* $\ell = 2$.

We have that

$$\begin{aligned}\sum_{j=0}^{n-\ell} \binom{n-\ell}{j} h_{j,0} &= t_{n,g} \left[\binom{n-3}{n/2-1} \frac{(n-1)(n-2g)^2}{ng(n-g)} + \binom{n-2}{n/2} \right] \\ &= t_{n,g} \binom{n-3}{n/2-1} \left[\frac{(n-1)(n-2g)^2}{ng(n-g)} + \frac{2(n-2)}{n} \right] \\ &= t_{n,g} \binom{n-3}{n/2-1} \frac{n^2-2ng-n+2g^2}{g(n-g)}\end{aligned}$$

and

$$\sum_{j=0}^{n-\ell} \binom{n-\ell}{j} h_{j,1} = t_{n,g} \binom{n-3}{n/2-1} \frac{\binom{n-2}{g} + \binom{n-2}{n-g}}{\binom{n-2}{g-1}}$$

$$= t_{n,g}\binom{n-3}{n/2-1}\left[\frac{(n-g-1)}{g}+\frac{g-1}{n-g}.\right]$$
$$= t_{n,g}\binom{n-3}{n/2-1}\frac{n^2-2ng-n+2g^2}{g(n-g)}.$$

Therefore, for $\ell = 2$, we have that identity (11) in Theorem 5.1 holds.

(ii) *Proof that equality* (11) *in Theorem* 5.1 *is satisfied for the case* $\ell = 3$.

We have that

$$\sum_{j=0}^{n-\ell}\binom{n-\ell}{j}h_{j,0} = t_{n,g}\left[\binom{n-3}{n/2-1}\frac{(n-1)(n-2g)^2}{ng(n-g)}+\binom{n-3}{n/2}\right]$$
$$= t_{n,g}\binom{n-3}{n/2-1}\left[\frac{(n-1)(n-2g)^2}{ng(n-g)}+\frac{n-4}{n}\right]$$

and

$$\sum_{j=0}^{n-\ell}\binom{n-\ell}{j}h_{j,1}$$

$$= \binom{n-3}{n/2-1}\frac{(n-1)(n-2g)^2}{ng(n-g)}+\binom{n-3}{n/2} = \binom{n-3}{g}\frac{\binom{n-3}{n/2-1}}{\binom{n-2}{g-1}}+\binom{n-3}{n-g}\frac{\binom{n-3}{n/2-1}}{\binom{n-2}{g-1}},$$

that is,

$$\frac{(n-1)(n-2g)^2}{ng(n-g)}+\frac{n/2-2}{n/2} = \frac{\binom{n-3}{g}+\binom{n-3}{n-g}}{\binom{n-2}{g-1}},$$

which turns out to be equivalent to

$$\frac{(n-1)(n-2g)^2}{ng(n-g)}+\frac{n-4}{n} = \frac{(n-g-1)(n-g-2)}{g(n-2)}+\frac{(g-1)(g-2)}{(n-g)(n-2)}.$$

Simple algebra shows that the above equality holds.

In the following we show the computations omitted from the proof of Theorem 6.1. Recall that equality (14) holds if and only if (15) is equal to (16). Now, we show that equality (14) is always satisfied.

(i) For $q = 2$ and $4 \le \ell < g$, we must show that

$$A(n,\ell,g) = B(n,\ell,g,2).$$

In Theorem 6.1 we proved that $A(n,\ell,g)$ is equal to (17). Notice that $B(n,\ell,g,2)$ can be written as

$$\frac{\binom{n-4}{g-3}\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)}-1\right]}{\binom{n-4}{\ell-3}\left[\frac{(n-\ell-1)(n-\ell-2)}{(\ell-1)(\ell-2)}-1\right]}\binom{n-4}{\ell-3}\left[\frac{(n-\ell-1)(n-2)(n-3)}{\ell(\ell-1)(\ell-2)}-\frac{(n-2)(n-3)}{(\ell-2)(n-\ell)}\right]$$
$$-\binom{n-4}{g-3}\left[\frac{(n-g-1)(n-2)(n-3)}{g(g-1)(g-2)}-\frac{(n-2)(n-3)}{(g-2)(n-g)}\right]$$

which is equal to

$$\binom{n-4}{g-3}\left\{\frac{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)}-1\right]\left[\frac{(n-\ell-1)(n-2)(n-3)}{\ell(\ell-1)(\ell-2)}-\frac{(n-2)(n-3)}{(\ell-2)(n-\ell)}\right]}{\left[\frac{(n-\ell-1)(n-\ell-2)}{(\ell-1)(\ell-2)}-1\right]}\right.$$
$$\left.-\left[\frac{(n-g-1)(n-2)(n-3)}{g(g-1)(g-2)}-\frac{(n-2)(n-3)}{(g-2)(n-g)}\right]\right\}.$$

Simple algebra shows that the above expression can be reduced to (17). Therefore, equality (14) is satisfied when $q = 2$ and $4 \leq \ell < g$.

(ii) For $q = 3$ and $4 \leq \ell < g$, we must show that

$$A(n, \ell, g) = B(n, \ell, g, 3).$$

In Theorem 6.1 we proved that $A(n, \ell, g)$ is equal to (17). Note that $B(n, \ell, g, 3)$ can be rewritten as

$$\frac{\binom{n-4}{g-3}\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)}-1\right]}{\binom{n-4}{\ell-3}\left[\frac{(n-\ell-1)(n-\ell-2)}{(\ell-1)(\ell-2)}-1\right]}\binom{n-4}{\ell-3}\left[\frac{(n-\ell-1)(n-\ell-2)(n-3)}{\ell(\ell-1)(\ell-2)}-\frac{(n-3)}{(n-\ell)}\right]$$
$$-\binom{n-4}{g-3}\left[\frac{(n-g-1)(n-g-2)(n-3)}{g(g-1)(g-2)}-\frac{(n-3)}{(n-g)}\right]$$

which is equal to

$$\binom{n-4}{g-3}\left\{\frac{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)}-1\right]\left[\frac{(n-\ell-1)(n-\ell-2)(n-3)}{\ell(\ell-1)(\ell-2)}-\frac{(n-3)}{(n-\ell)}\right]}{\left[\frac{(n-\ell-1)(n-\ell-2)}{(\ell-1)(\ell-2)}-1\right]}\right.$$
$$\left.-\left[\frac{(n-g-1)(n-g-2)(n-3)}{g(g-1)(g-2)}-\frac{(n-3)}{(n-g)}\right]\right\}.$$

Simple algebra shows that the above expression can be reduced to (17). Therefore, equality (14) holds for $q = 3$ when $4 \leq \ell < g$.

(iii) For $q = 1$, $\ell = 1$, and $g = 2$, we have that

$$A(n, 1, 2) = \frac{(n-3)(n-4)}{2} \quad \text{and} \quad B(n, 1, 2, 1) = \frac{(n-3)(n-4)}{2}.$$

Therefore, $A(n, 1, 2) = B(n, 1, 2, 1)$ and equality (14) in Theorem 6.1 is satisfied.

(iv) For $q = 1$, $\ell = 1$, and $g = 3$, we have that

$$A(n, 1, 3) = \frac{(n-2)(n-4)(n-6)}{3} \quad \text{and} \quad B(n, 1, 3, 1) = \frac{(n-2)(n-4)(n-6)}{3}.$$

Therefore, $A(n, 1, 3) = B(n, 1, 3, 1)$ and equality (14) in Theorem 6.1 is satisfied.

(v) For $q = 1$, $\ell = 2$, and $g = 3$, we have that

$$A(n, 2, 3) = \frac{(n-1)(n-5)(n-6)}{12} \quad \text{and} \quad B(n, 2, 3, 1) = \frac{(n-1)(n-5)(n-6)}{12}.$$

Therefore, $A(n, 2, 3) = B(n, 2, 3, 1)$ and equality (14) in Theorem 6.1 is satisfied.

(vi) For $q = 2$, $\ell = 1$, and $g = 2$, we have that

$$A(n,1,2) = \frac{(n-3)(n-4)}{2} \quad \text{and} \quad B(n,1,2,2) = \frac{(n-3)(n-4)}{2}.$$

Therefore, $A(n,1,2) = B(n,1,2,2)$ and equality (14) in Theorem 6.1 is satisfied.

(vii) For $q = 2$, $\ell = 1$, and $g = 3$, we have that

$$A(n,1,3) = \frac{(n-2)(n-4)(n-6)}{3} \quad \text{and} \quad B(n,1,3,2) = \frac{(n-2)(n-4)(n-6)}{3}.$$

Therefore, $A(n,1,3) = B(n,1,3,2)$ and equality (14) in Theorem 6.1 is satisfied.

(viii) For $q = 2$, $\ell = 2$, and $g = 3$, we have that

$$A(n,2,3) = \frac{(n-1)(n-5)(n-6)}{12} \quad \text{and} \quad B(n,2,3,2) = \frac{(n-1)(n-5)(n-6)}{12}.$$

Therefore, $A(n,2,3) = B(n,2,3,2)$ and equality (14) in Theorem 6.1 is satisfied.

(ix) For $q = 3$, $\ell = 1$, and $g = 2$, we have that

$$A(n,1,2) = \frac{(n-3)(n-4)}{2} \quad \text{and} \quad B(n,1,2,3) = \frac{(n-3)(n-4)}{2}.$$

Therefore, $A(n,1,2) = B(n,1,2,3)$ and equality (14) in Theorem 6.1 is satisfied.

(x) For $q = 3$, $\ell = 1$, and $g = 3$, we have that

$$A(n,1,3) = \frac{(n-2)(n-4)(n-6)}{3} \quad \text{and} \quad B(n,1,3,3) = \frac{(n-2)(n-4)(n-6)}{3}.$$

Therefore, $A(n,1,3) = B(n,1,3,3)$ and equality (14) in Theorem 6.1 is satisfied.

(xi) For $q = 3$, $\ell = 2$, and $g = 3$, we have that

$$A(n,2,3) = \frac{(n-1)(n-5)(n-6)}{12} \quad \text{and} \quad B(n,2,3,3) = \frac{(n-1)(n-5)(n-6)}{12}.$$

Therefore, $A(n,2,3) = B(n,2,3,3)$ and equality (14) in Theorem 6.1 is satisfied.

(xii) For $q = 1$, $\ell = 1$, and $g \geq 4$, we must show that $A(n,1,g) = B(n,1,g,1)$. Notice that

$$\begin{aligned} A(n,1,g) = \binom{n-4}{g-3} \Bigg\{ & \left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1 \right] (n-4) \\ & - \left[\frac{(n-g-1)(n-g-2)(n-g-3)}{g(g-1)(g-2)} - \frac{(g-3)}{(n-g)} \right] \Bigg\} \end{aligned}$$

and

$$\begin{aligned} B(n,1,g,1) = \binom{n-4}{g-3} \Bigg\{ & \left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1 \right] (n-2) \\ & - \left[\frac{(n-1)(n-2)(n-3)}{g(g-1)(g-2)} - \frac{(n-1)(n-2)(n-3)}{(g-1)(g-2)(n-g)} \right] \Bigg\}. \end{aligned}$$

After some algebra, $A(n,1,g)$ and $B(n,1,g,1)$ can be reduced to

$$\binom{n-4}{g-3} \frac{(n-2)(n-3)(n-2g)(n-g-1)}{g(g-2)(n-g)}.$$

Therefore, equality (14) in Theorem 6.1 is satisfied when $q = 1$, $\ell = 1$, and $g \geq 4$.

(xiii) For $q = 1$, $\ell = 2$, and $g \geq 4$, we must show that $A(n, 2, g) = B(n, 2, g, 1)$. Notice that

$$A(n,2,g) = \binom{n-4}{g-3}\left\{\frac{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1\right]}{(n-4)}\frac{(n-4)(n-5)}{2} - \left[\frac{(n-g-1)(n-g-2)(n-g-3)}{g(g-1)(g-2)} - \frac{(g-3)}{(n-g)}\right]\right\}$$

and

$$B(n,2,g,1) = \binom{n-4}{g-3}\left\{\frac{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1\right]\left[\frac{(n-1)(n-2)}{2} - (n-1)\right]}{(n-4)} - \left[\frac{(n-1)(n-2)(n-3)}{g(g-1)(g-2)} - \frac{(n-1)(n-2)(n-3)}{(g-1)(g-2)(n-g)}\right]\right\}.$$

After some algebra, $A(n, 2, g)$ and $B(n, 2, g, 1)$ can be reduced to

$$\binom{n-4}{g-3}\frac{(n-1)(n-3)(n-2g)(n-g-2)}{2g(g-1)(n-g)}.$$

Therefore, equality (14) in Theorem 6.1 is satisfied when $q = 1$, $\ell = 2$, and $g \geq 4$.

(xiv) For $q = 1$, $\ell = 3$, and $g \geq 4$, we must show that $A(n, 3, g) = B(n, 3, g, 1)$. Notice that

$$A(n,3,g) = \binom{n-4}{g-3}\left\{\frac{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1\right]\frac{(n-4)(n-5)(n-6)}{6}}{\frac{(n-4)(n-5)}{2} - 1} - \frac{(n-g-1)(n-g-2)(n-g-3)}{g(g-1)(g-2)} + \frac{(g-3)}{(n-g)}\right\}$$

and

$$B(n,3,g,1) = \binom{n-4}{g-3}\left\{\frac{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1\right]\left[\frac{(n-1)(n-2)(n-3)}{6} - \frac{(n-1)(n-2)}{2}\right]}{\frac{(n-4)(n-5)}{2} - 1} - \frac{(n-1)(n-2)(n-3)}{g(g-1)(g-2)} + \frac{(n-1)(n-2)(n-3)}{(g-1)(g-2)(n-g)}\right\}.$$

After some algebra, $A(n, 3, g)$ and $B(n, 3, g, 1)$ can be reduced to

$$\binom{n-4}{g-3}\frac{(n-1)(n-2)(g-3)(n-2g)(n-g-3)}{3g(g-1)(g-2)(n-g)}.$$

Therefore, equality (14) in Theorem 6.1 is satisfied for $q = 1$, $\ell = 3$, and $g \geq 4$.

(xv) For $q = 2$, $\ell = 1$, and $g \geq 4$, we must show that $A(n, 1, g) = B(n, 1, g, 2)$. Notice that

$$B(n,1,g,2) = \binom{n-4}{g-3}\left\{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1\right](n-2) - \left[\frac{(n-g-1)(n-2)(n-3)}{g(g-1)(g-2)} - \frac{(n-2)(n-3)}{(g-2)(n-g)}\right]\right\}.$$

After some algebra, $B(n,1,g,2)$ can be reduced to

$$\binom{n-4}{g-3}\frac{(n-2)(n-3)(n-2g)(n-g-1)}{g(g-2)(n-g)}.$$

Therefore, equality (14) in Theorem 6.1 is satisfied when $q=2$, $\ell=1$, and $g\geq 4$.

(xvi) For $q=2$, $\ell=2$, and $g\geq 4$, we must show that $A(n,2,g)=B(n,2,g,2)$. Notice that

$$B(n,2,g,2)=\binom{n-4}{g-3}\left\{\frac{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)}-1\right]\left[\frac{(n-2)(n-3)}{2}-1\right]}{(n-4)}\right.$$
$$\left.-\left[\frac{(n-g-1)(n-2)(n-3)}{g(g-1)(g-2)}-\frac{(n-2)(n-3)}{(g-2)(n-g)}\right]\right\}.$$

After some algebra, $B(n,2,g,2)$ can be reduced to

$$\binom{n-4}{g-3}\frac{(n-1)(n-3)(n-2g)(n-g-2)}{2g(g-1)(n-g)}.$$

Therefore, equality (14) in Theorem 6.1 is satisfied when $q=2$, $\ell=2$, and $g\geq 4$.

(xvii) For $q=2$, $\ell=3$, and $g\geq 4$, we must show that $A(n,3,g)=B(n,3,g,2)$. Notice that

$$B(n,3,g,2)=\binom{n-4}{g-3}\left\{\frac{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)}-1\right]\left[\frac{(n-2)(n-3)(n-4)}{6}-(n-2)\right]}{\frac{(n-4)(n-5)}{2}-1}\right.$$
$$\left.-\left[\frac{(n-g-1)(n-2)(n-3)}{g(g-1)(g-2)}-\frac{(n-2)(n-3)}{(g-2)(n-g)}\right]\right\}.$$

After some algebra, $B(n,3,g,2)$ can be reduced to

$$\binom{n-4}{g-3}\frac{(n-1)(n-2)(g-3)(n-2g)(n-g-3)}{3g(g-1)(g-2)(n-g)}.$$

Therefore, equality (14) in Theorem 6.1 is satisfied for $q=2$, $\ell=3$, and $g\geq 4$.

(xviii) For $q=3$, $\ell=1$, and $g\geq 4$, we must show that $A(n,1,g)=B(n,1,g,3)$. Notice that

$$B(n,1,g,3)=\binom{n-4}{g-3}\left\{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)}-1\right](n-3)\right.$$
$$\left.-\left[\frac{(n-g-1)(n-g-2)(n-3)}{g(g-1)(g-2)}-\frac{(n-3)}{(n-g)}\right]\right\}.$$

After some algebra, $B(n,1,g,3)$ can be reduced to

$$\binom{n-4}{g-3}\frac{(n-2)(n-3)(n-2g)(n-g-1)}{g(g-2)(n-g)}.$$

Therefore, equality (14) in Theorem 6.1 is satisfied when $q=3$, $\ell=1$, and $g\geq 4$.

(xix) For $q = 3$, $\ell = 2$, and $g \geq 4$, we must show that $A(n, 2, g) = B(n, 3, g, 3)$. Notice that

$$B(n,2,g,3) = \binom{n-4}{g-3}\left\{\frac{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1\right]\left[\frac{(n-3)(n-4)}{2}\right]}{(n-4)} - \left[\frac{(n-g-1)(n-g-2)(n-3)}{g(g-1)(g-2)} - \frac{(n-3)}{(n-g)}\right]\right\}.$$

After some algebra, $B(n, 2, g, 3)$ can be reduced to

$$\binom{n-4}{g-3}\frac{(n-1)(n-3)(n-2g)(n-g-2)}{2g(g-1)(n-g)}.$$

Therefore, equality (14) in Theorem 6.1 is satisfied when $q = 3$, $\ell = 2$, and $g \geq 4$.

(xx) For $q = 3$, $\ell = 3$, and $g \geq 4$, we must show that $A(n, 3, g) = B(n, 3, g, 3)$. Notice that

$$B(n,3,g,3) = \binom{n-4}{g-3}\left\{\frac{\left[\frac{(n-g-1)(n-g-2)}{(g-1)(g-2)} - 1\right]\left[\frac{(n-3)(n-4)(n-5)}{6} - 1\right]}{\frac{(n-4)(n-5)}{2} - 1} - \left[\frac{(n-g-1)(n-g-2)(n-3)}{g(g-1)(g-2)} - \frac{(n-3)}{(n-g)}\right]\right\}.$$

After some algebra, $B(n, 3, g, 3)$ can be reduced to

$$\binom{n-4}{g-3}\frac{(n-1)(n-2)(g-3)(n-2g)(n-g-3)}{3g(g-1)(g-2)(n-g)}.$$

Therefore, equality (14) in Theorem 6.1 is satisfied for $q = 3$, $\ell = 3$, and $g \geq 4$.

In the following we prove that equality (18) in the proof of Theorem 6.1 holds for the cases $1 \leq \ell \leq 3$ with $g \geq 4$ and $1 \leq \ell < g \leq 3$. In order to prove (18), we must show that

$$F(n,\ell,g) \triangleq \frac{s_{(n,\ell,g)}\left[\binom{n-5}{\ell-1} - \binom{n-5}{\ell-4}\right] - t_{(n,\ell,g)}\left[\binom{n-5}{g-1} - \binom{n-5}{g-4}\right]}{s_{(n,\ell,g)}\left[\binom{n}{\ell} + \binom{n-4}{\ell} - \binom{n-4}{\ell-4}\right] + t_{(n,\ell,g)}\left[\binom{n}{g} + \binom{n-4}{g-4} - \binom{n-4}{g}\right]}$$

is equal to

$$D(n,\ell,g) \triangleq \frac{\ell(g-\ell)(n-g)(n-2g)(n-2\ell)}{2(n+2\ell-2g)(n-1)(n-2)(n-3)(n-4)}.$$

Recall that in the proof of Theorem 6.1 we have defined

$$a \triangleq s_{(n,\ell,g)}\left[\binom{n-5}{\ell-1} - \binom{n-5}{\ell-4}\right], \qquad b \triangleq t_{(n,\ell,g)}\left[\binom{n-5}{g-1} - \binom{n-5}{g-4}\right],$$

$$c \triangleq s_{(n,\ell,g)}\left[\binom{n}{\ell} + \binom{n-4}{\ell} - \binom{n-4}{\ell-4}\right], \qquad d \triangleq t_{(n,\ell,g)}\left[\binom{n}{g} + \binom{n-4}{g-4} - \binom{n-4}{g}\right].$$

(i) If $\ell = 1$ and $g \geq 4$, it holds that

$$\frac{a}{c} = \frac{1}{2(n-2)}, \qquad \frac{b}{a} = \frac{(n^2 - ng - 6n + 11 + g^2)}{(n-3)(n-4)},$$

and

$$\frac{d}{c} = \frac{(2n^2 - 3gn - 3n + 2g^2 + 2)}{(n-g)(n-2g)(n-2)}.$$

Since

$$\frac{(a-b)}{(c+d)} = \frac{a(1-\frac{b}{a})}{c(1+\frac{d}{c})} \tag{20}$$

we have that

$$F(n,1,g) = \frac{(n-g)(n-2g)(g-1)}{2(n-2g+2)(n-1)(n-3)(n-4)}$$

and

$$D(n,1,g) = \frac{(n-g)(n-2g)(g-1)}{2(n-2g+2)(n-1)(n-3)(n-4)}.$$

Therefore, equality (18) is satisfied.

(ii) If $\ell = 2$ and $g \geq 4$, it holds that

$$\frac{a}{c} = \frac{(n-5)}{(n^2 - 5n + 10)}, \qquad \frac{b}{a} = \frac{(n^2 - ng - 6n + 11 + g^2)}{(n-3)(n-5)},$$

and

$$\frac{d}{c} = \frac{2(2n^2 - 3gn - 3n + 2g^2 + 2)(n-4)}{(n-g)(n-2g)(n^2 - 5n + 10)}.$$

From (20), we have that

$$F(n,2,g) = \frac{(n-g)(n-2g)(g-2)}{(n-2g+4)(n-1)(n-2)(n-3)}$$

and

$$D(n,2,g) = \frac{(n-g)(n-2g)(g-2)}{(n-2g+4)(n-1)(n-2)(n-3)}.$$

Therefore, equality (18) is satisfied.

(iii) If $\ell = 3$ and $g \geq 4$, we have

$$\frac{a}{c} = \frac{3(n-5)(n-6)}{(n(n-1)(n-2) + (n-4)(n-5)(n-6))}, \quad \frac{b}{a} = \frac{(n^2 - ng - 6n + 11 + g^2)}{(n-4)(n-5)},$$

and

$$\frac{d}{c} = \frac{3(n-6)(2n^2 - 3gn - 3n + 2g^2 + 2)}{(n-g)(n-2g)(n^2 - 6n + 20)}.$$

From (20), we have that

$$F(n,3,g) = \frac{3(n-g)(n-2g)(g-3)(n-6)}{2(n-2g+6)(n-1)(n-2)(n-3)(n-4)}$$

and

$$D(n,3,g) = \frac{3(n-g)(n-2g)(g-3)(n-6)}{2(n-2g+6)(n-1)(n-2)(n-3)(n-4)}.$$

Therefore, equality (18) is satisfied.

(iv) If $\ell = 1$ and $g = 2$, it is easy to see that

$$F(n,1,2) = D(n,1,2) = \frac{1}{2(n-1)(n-3)}.$$

(v) If $\ell = 1$ and $g = 3$, it is easy to see that

$$F(n,1,3) = D(n,1,3) = \frac{(n-6)}{2(n-1)(n-4)}.$$

(vi) If $\ell = 2$ and $g = 3$, it is easy to see that

$$F(n,2,3) = D(n,2,3) = \frac{(n-6)}{2(n-1)(n-2)}.$$

Therefore, equality (18) is satisfied.

Appendix B.

$n \setminus k$	2			3			4			5			6		
	$j, h_{j,0}$	$j, h_{j,1}$	m, α	$j, h_{j,0}$	$j, h_{j,1}$	m, α	$j, h_{j,0}$	$j, h_{j,1}$	m, α	$j, h_{j,0}$	$j, h_{j,1}$	m, α	$j, h_{j,0}$	$j, h_{j,1}$	m, α
2	$0,1$ $2,1$	$1,1$	2 1/2	–	–	–	–	–	–	–	–	–	–	–	–
3	$0,2$ $3,1$	$1,1$	3 1/3	$0,1$ $2,1$	$1,1$ $3,1$	4 1/4	–	–	–	–	–	–	–	–	–
4	$0,3$ $4,3$	$2,1$	6 1/3	$0,2$ $3,1$	$1,1$ $4,2$	6 1/6	$0,1$ $2,1$ $4,1$	$1,1$ $3,1$	8 1/8	–	–	–	–	–	–
5	$0,6$ $5,4$	$1,1$	10 3/10	$0,3$ $4,1$	$1,1$ $5,3$	8 1/8	$0,3$ $2,1$ $5,2$	$1,2$ $4,1$	15 1/15	$0,1$ $2,1$ $4,1$	$1,1$ $3,1$ $5,1$	16 1/16	–	–	–
6	$0,10$ $6,10$	$3,1$	20 3/20	$0,4$ $5,1$	$1,1$ $6,4$	10 1/10	$0,8$ $3,1$ $6,8$	$1,3$ $5,3$	36 1/18	$0,3$ $2,1$ $5,2$	$1,2$ $4,1$ $6,3$	30 1/30	$0,1$ $2,1$ $4,1$ $6,1$	$1,1$ $3,1$ $5,1$	32 1/32

FIG. B.1. *Contrast optimal VCS for* $2 \le k \le n \le 6$.

$n \setminus k$	2			3			4			5			6		
	$j,h_{j,0}$	$j,h_{j,1}$	m,α	$j,h_{j,0}$	$j,h_{j,1}$	m,α	$j,h_{j,0}$	$j,h_{j,1}$	m,α	$j,h_{j,0}$	$j,h_{j,1}$	m,α	$j,h_{j,0}$	$j,h_{j,1}$	m,α
7	0, 20 7, 15	3, 1	35 2/7	0, 9 5, 1	2, 1 7, 9	30 1/10	0, 20 3, 1 7, 15	1, 6 6, 4	70 3/70	0, 6 2, 1 6, 3	1, 3 5, 1 7, 6	48 1/48	0, 4 2, 2 5, 1 7, 3	1, 3 3, 1 6, 2	70 1/70
8	0, 35 8, 35	4, 1	70 2/7	0, 14 6, 1	2, 1 8, 14	42 2/21	0, 45 4, 1 8, 45	1, 10 7, 10	160 3/80	0, 16 3, 1 7, 5	1, 5 5, 1 8, 16	112 1/56	0, 15 2, 3 6, 3 8, 15	1, 8 4, 1 7, 8	198 1/99
9	0, 70 9, 56	4, 1	126 5/18	0, 20 7, 1	2, 1 9, 20	56 5/56	0, 105 4, 1 9, 84	1, 20 8, 15	315 2/63	0, 35 3, 1 8, 9	1, 9 6, 1 9, 35	200 3/200	0, 45 2, 6 7, 4 9, 36	1, 20 4, 1 8, 15	441 1/147
10	0, 126 10, 126	5, 1	252 5/18	0, 180 7, 2	2, 7 10, 105	420 1/12	0, 224 5, 1 10, 224	1, 35 9, 35	700 1/35	0, 64 3, 1 9, 14	1, 14 7, 1 1, 164	324 1/81	0, 128 3, 5 7, 5 10, 128	1, 35 5, 3 9, 35	1456 1/182
11	0, 210 11, 252	6, 1	462 3/11	0, 75 8, 1	3, 1 11, 75	240 1/12	0, 140 6, 1 11, 168	2, 6 9, 8	770 3/110	0, 162 4, 1 10, 28	1, 28 7, 1 1, 1162	800 9/800	0, 105 3, 2 8, 3 11, 126	1, 24 6, 1 10, 30	1056 5/1056

FIG. B.2. *Contrast optimal VCS for* $2 \leq k \leq 6$ *and* $7 \leq n \leq 11$.

$n \setminus k$	7			8			9			10			11		
	$j,h_{j,0}$	$j,h_{j,1}$	m,α	$j,h_{j,0}$	$j,h_{j,1}$	m,α	$j,h_{j,0}$	$j,h_{j,1}$	m,α	$j,h_{j,0}$	$j,h_{j,1}$	m,α	$j,h_{j,0}$	$j,h_{j,1}$	m,α
7	0, 1 2, 1 4, 1 6, 1	1, 1 3, 1 5, 1 7, 1	64 1/64	–	–	–	–	–	–	–	–	–	–	–	–
8	0, 4 2, 2 5, 1 7, 3	1, 3 3, 1 6, 2 8, 4	140 1/140	0, 1 2, 1 4, 1 6, 1 8, 1	1, 1 3, 1 5, 1 7, 1	128 1/128	–	–	–	–	–	–	–	–	–
9	0, 9 2, 2 5, 1 8, 5	1, 5 4, 1 7, 2 9, 9	252 1/252	0, 5 2, 3 4, 1 7, 2 9, 4	1, 4 3, 2 6, 1 8, 3	315 1/315	0, 1 2, 1 4, 1 6, 1 8, 1	1, 1 3, 1 5, 1 7, 1 9, 1	256 1/256	–	–	–	–	–	–
10	0, 35 2, 5 6, 1 9, 16	1, 16 4, 1 8, 5 10, 35	630 1/315	0, 24 2, 8 5, 1 8, 8 10, 24	1, 15 3, 3 7, 3 9, 15	1020 1/510	0, 5 2, 3 4, 1 7, 2 9, 4	1, 4 3, 2 6, 1 8, 3 10, 5	630 1/630	0, 1 2, 1 4, 1 6, 1 8, 1 10, 1	1, 1 3, 1 5, 1 7, 1 9, 1	512 1/512	–	–	–
11	0, 90 2, 9 7, 1 10, 35	1, 35 4, 1 9, 9 11, 90	1300 3/1300	0, 84 2, 20 5, 1 9, 15 11, 70	1, 45 3, 6 8, 4 10, 36	2541 1/847	0, 14 2, 5 5, 1 8, 2 10, 9	1, 9 3, 2 6, 1 9, 5 11, 14	1180 1/1180	0, 6 2, 4 4, 2 7, 1 9, 3 11, 5	1, 5 3, 3 5, 1 8, 2 10, 4	1386 1/1386	0, 1 2, 1 4, 1 6, 1 8, 1 10, 1	1, 1 31 5, 1 7, 1 9, 1 11, 1	1024 1/1024

FIG. B.3. *Contrast optimal VCS for* $7 \leq k \leq n \leq 11$.

REFERENCES

[1] G. Ateniese, C. Blundo, A. De Santis, and D. R. Stinson, *Visual cryptography for general access structures*, Inform. and Comput., 129 (1996), pp. 86–106.

[2] G. Ateniese, C. Blundo, A. De Santis, and D. R. Stinson, *Extended capabilities for visual cryptography*, Theoret. Comput. Sci., 250 (2001), pp. 143–161.

[3] G. Ateniese, C. Blundo, A. De Santis, and D. R. Stinson, *Constructions and bounds for visual cryptography*, in Proc. of the 23rd International Colloquium on Automata, Languages and Programming (ICALP '96), F. M. auf der Heide and B. Monien, eds., Lecture Notes in Comput. Sci. 1099, Springer-Verlag, Berlin, 1996, pp. 416–428.

[4] E. Biham, *Visual cryptography with polarization*, talk presented at the Weizmann Workshop on Cryptography, Weizmann Institute, Rehovot, Israel, June 8–9, 1997.

[5] C. Blundo, A. De Santis, and D. R. Stinson, *On the contrast in visual cryptography schemes*, J. Cryptology, 12 (1999), pp. 261–289.

[6] A. De Bonis and A. De Santis, *Randomness in visual cryptography*, in Proc. of the 17th International Symposium on Theoretical Aspects of Computer Science (STACS 2000), Lille, France, H. Reichel and S. Tison, eds., Lecture Notes in Comput. Sci. 1770, Springer-Verlag, Berlin, 2000, pp. 626–638.

[7] S. Droste, *New results on visual cryptography*, in Advances in Cryptology — CRYPTO '96, N. Koblitz, ed., Lecture Notes in Comput. Sci. 1109, Springer-Verlag, Berlin, 1996, pp. 401–415.

[8] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics. A Foundation for Computer Science*, Addison–Wesley, Reading, MA, 1988.

[9] T. Hofmeister, M. Krause, and H. U. Simon, *Contrast-optimal k out of n secret sharing schemes in visual cryptography*, in COCOON '97, T. Jiang and D. T. Lee, eds., Lecture Notes in Comput. Sci. 1276, Springer-Verlag, Berlin, 1997, pp. 176–185.

[10] D. Naccache, *Colorful cryptography—a purely physical secret-sharing scheme based on chromatic filters,* talk presented at Coding and Information Integrity, French-Israeli Workshop, December 1994.

[11] M. Naor and B. Pinkas, *Visual authentication and identification*, in Advances in Cryptology—CRYPTO '97, B. S. Kaliski Jr., ed., Lecture Notes in Comput. Sci. 1294, Springer-Verlag, Berlin, 1997, pp. 322–336.

[12] M. Naor and A. Shamir, *Visual cryptography*, in Advances in Cryptology—Eurocrypt '94, A. De Santis, ed., Lecture Notes in Comput. Sci. 950, Springer-Verlag, Berlin, 1995, pp. 1–12.

[13] M. Naor and A. Shamir, *Visual cryptography* II*: Improving the contrast via the cover base*, in Security Protocols, M. Lomas, ed., Lecture Notes in Comput. Sci. 1189, Springer-Verlag, Berlin, 1997, pp. 197–202. Available online from the Theory of Cryptography Library at ftp://theory.lcs.mit.edu/pub/tcryptol/96-07.ps.

[14] V. Rijmen, *Efficient colour visual encryption or "shared colors of Benetton,"* talk presented at EUROCRYPT '96 Rump Session. Text available online at http://www.iacr.org/conferences/ec96/rump/index.html.

[15] D. R. Stinson, *An introduction to visual cryptography*, talk presented at Public Key Solutions '97, Toronto, April 28–30, 1997. Text available online at http://www.cacr.math.uwaterloo.ca/∼ dstinson/.

[16] E. R. Verheul and H. C. A. van Tilborg, *Constructions and properties of k out of n visual secret sharing schemes,* in Des. Codes Cryptogr., 11 (1997), pp. 179–196.

SIAM J. DISCRETE MATH.
Vol. 16, No. 2, pp. 262–275

FINITE SUBSETS OF THE PLANE ARE 18-RECONSTRUCTIBLE*

L. PEBODY†, A. J. RADCLIFFE‡, AND A. D. SCOTT§

Abstract. We prove that every finite subset of the plane is reconstructible from the multiset of its subsets of at most 18 points, each given up to rigid motion. We also give some results concerning the reconstructibility of infinite subsets of the plane.

Key words. reconstruction problem, group action

AMS subject classifications. 05C60, 05E20

PII. S0895480101391648

1. Introduction.

Combinatorial reconstruction problems arise when we are given the multiset of subobjects of a certain size of some combinatorial object, up to isomorphism, and are asked whether this is sufficient information to reconstruct the original object. For instance, the reconstruction conjecture, made sixty years ago by Ulam [37] and Kelly [13], asserts that all finite graphs on at least three vertices can be reconstructed from the collection of all their (nontrivial) induced subgraphs. Similarly, the edge reconstruction conjecture (Harary [10]) asserts that every graph with at least four edges can be reconstructed from the collection of all its (nontrivial) subgraphs. There is substantial literature on graph reconstruction (see, for instance, [3, 2, 4, 15, 27]). Reconstruction problems have been considered for a variety of other combinatorial objects, including directed graphs [35, 36], hypergraphs [16], infinite graphs [28], codes [20], sets of real numbers [31], sequences [34, 18], and combinatorial geometries [6, 5].

The necessary ingredients for a combinatorial reconstruction problem are a notion of isomorphism and a notion of subobject. Some progress has been made in recent years in the general case, where we have a group action $G \rightarrowtail X$ providing the notion of isomorphism, and we wish to reconstruct a subset S of X from the multiset of isomorphism classes of its k-element subsets, known as the *k-deck* (see Alon et al. [1], Babai [2], Cameron [7, 9, 8], Krasikov and Roditty [17], Maynard and Siemons [21], Mnukhin [22, 23, 25], and Radcliffe and Scott [32]). Several authors [1, 7, 23] have noted that we can reconstruct S provided $k > \log_2 |G| + 1$; the $n \log_2 n$ bound for edge reconstruction (Müller [26]; Lovász [19]) also follows from this. In general, however, much smaller decks may suffice (see [29, 32]).

In this paper we focus on the case of the plane, $\mathbb{R}^2$, with the group R of rigid motions acting on it. Thus the k-deck of a set S of points in the plane is the multiset of its k-subsets given up to rigid motion. (For instance, the 2-deck is essentially the multiset of distances between pairs of points in S.) We want to know how large k must be so that S is determined up to rigid motion by its k-deck. Alon et al. [1] proved that

*Received by the editors June 29, 2001; accepted for publication (in revised form) October 16, 2002; published electronically February 20, 2003.

http://www.siam.org/journals/sidma/16-2/39164.html

†Department of Mathematical Sciences, University of Memphis, Memphis, TN 38152 (pebodyl@msci.memphis.edu).

‡Department of Mathematics and Statistics, University of Nebraska-Lincoln, Lincoln, NE 68588-0323 (jradclif@math.unl.edu).

§Department of Mathematics, University College, Gower Street, London WC1E 6BT, UK (A.D.Scott@ucl.ac.uk).

subsets of n points in the plane can be reconstructed from their $(\log_2 n + 1)$-decks. Our first aim in this paper is to prove that every finite subset of the plane can be reconstructed from its 18-deck.

We begin by considering sets of points in the plane together with an "orientation," which leads naturally to the problem of reconstructing finite subsets of the circle $\mathbb{T} = \mathbb{R}/\mathbb{Z}$. It is crucial to our approach that finite subsets of $\mathbb{T}$ are reconstructible from bounded decks, under the action of $\mathbb{T}$ on itself by translation. This in turn is proved by considering the circle as a limit (in an appropriate sense) of the groups $\mathbb{Z}_n$ for n large. Alon et al. [1] proved that if $\mathbb{Z}_n$ acts on itself, then arbitrary subsets S are reconstructible from their $(\log_2 n+1)$-decks (see also Mnukhin [23, 24]). Radcliffe and Scott [30] improved their bound substantially in the case of $\mathbb{Z}_n$ acting on itself. Using a Fourier analytic approach, they showed (among other results) that if S is a finite multiset in Z_p and p is prime, then S is reconstructible from its 3-deck. Using more refined Fourier analytic arguments, Pebody [29] proved the following result.

Theorem 1.1. *If S is a finite multiset of elements of $\mathbb{Z}_n$, then S can be reconstructed from its* 6*-deck.*

In fact Pebody proved rather more, computing for every abelian group A the minimum k (as a function of A) for which all multisets in A are k-reconstructible.

In this paper we prove first that finite subsets of $\mathbb{T}$ are reconstructible from their 6-decks and then that finite subsets of the plane $\mathbb{R}^2$, under the action of the group R of rigid motions, are reconstructible from their 18-decks. Our proof for the plane works by reducing the problem of reconstructing a set up to the action of the group of rigid motions to that of reconstructing it up to the action of the group of translations. This requires us to reconstruct the orientations of the sets in an appropriately sized deck. The technique that allows us to do this is the method of "features" and we present it in section 2, in a quite general form, before proving our results on finite subsets of $\mathbb{T}$ and $\mathbb{R}^2$ in section 3. It turns out that we can use this approach in another, slightly different situation, and in section 4 we prove some results concerning the reconstructibility of infinite subsets of the plane.

1.1. Definitions. In the following we suppose that a group action $G \rightarrowtail X$ has been specified. We write the group action generically as $(g, x) \mapsto g.x$. We shall most often be dealing with the group R of rigid motions of the plane acting on $\mathbb{R}^2$, in which case we shall usually think of the elements of R as functions mapping the plane to itself, and write the action as a function application. A rigid motion of the plane is an affine isometry preserving orientation. For notation and terminology, see [11]. We will always assume that $G \rightarrowtail X$ is transitive.

An essential part of our approach to reconstructing *subsets* of the plane is to consider the more general problem of reconstructing *multisets* of points in the plane, where each point is allowed to have finite multiplicity. This should not be too surprising since [30] and [29] both proceed by proving results concerning the action of $\mathbb{Z}_n$ on the group ring $\mathbb{Q}\,\mathbb{Z}_n$.

Definition 1.2. *Formally, a* multiset S in X with finite multiplicities *is a function $m_S : X \to \{0, 1, 2, \ldots\}$. We say that $m_S(x)$ is the* multiplicity *of x in S and define the* support *of S to be the set* $\mathrm{supp}(S) = \{x \in X : m_S(x) > 0\}$. *The* size *of S is $|S| = \sum_{x\in X} m_S(x)$. We shall often refer to a multiset in X of finite size as a* configuration. *We write $\mathcal{M}(X)$ for the collection of all finite multisets in X.*

A multiset K is contained *in a multiset S if $m_K(x) \leq m_S(x)$ for all $x \in X$. The* power set *$\mathcal{P}(S)$ of S is the multiset in which each $K \subset S$ has multiplicity $\prod_{x\in\mathrm{supp}(K)} \binom{m_S(x)}{m_K(x)}$; we write $\mathcal{P}_r(S) = \{A \in \mathcal{P}(S) : |A| = r\}$. With this convention* *the size of $\mathcal{P}(S)$ is $2^{|S|}$, and $|\mathcal{P}_r(S)| = \binom{|S|}{r}$.*

We shall have to consider two different notions of union. The multiset union *of a collection $\mathcal{S}$ of multisets (or sets) is the multiset $\bigoplus_{S\in\mathcal{S}} S$ in which each $x \in X$ has multiplicity $\sum_{S\in\mathcal{S}} m_S(x)$. The* set union *$\bigcup_{S\in\mathcal{S}} S$ gives to each $x \in X$ the multiplicity $\max_{S\in\mathcal{S}} m_S(x)$.*

Definition 1.3. *Given two multisets S, T in X we say that they are* isomorphic, *and write $S \simeq T$, if there exists $g \in G$ such that $g.S = T$. The collection of all multisets in X isomorphic to S is the* isomorphism class *of S, written $[S]_G$ (or simply $[S]$ if the group action is sufficiently clear).*

Definition 1.4. *If S is a multiset in X, then the k-*deck *of S is the multiset*

$$D_k(S) = \{[K]_G \,:\, K \in \mathcal{P}(S), |K| \leq k\}.$$

Note that $K \subset S$ might well arise multiple times as a subset of S: to be precise, K arises $\prod_{x\in \operatorname{supp}(K)} \binom{m_S(x)}{m_K(x)}$ times. Thus, for $|K| \leq k$, the multiset $D_k(S)$ gives the cardinality of the collection of multisets in $\mathcal{P}(S)$ belonging to a fixed isomorphism class $[K]$. We write $m_S([K])$ for the multiplicity $m_{D_k(S)}([K])$. In some cases we will want to emphasize the particular group action, in which case we will write $D_k(G \rightarrowtail S)$. The entire collection of isomorphism classes of finite subsets of S we will call the ($< \omega$)-deck of S, *written $D(S) = \{[K] \,:\, K \in \mathcal{P}(S), |K| < \infty\}$.*

We remark that the k-deck is often defined in terms of the subsets of S of size *exactly k*. However, the two definitions are easily seen to be equivalent here for $\infty \geq |S| \geq k$, by a variant of Kelly's lemma [14]. (Further discussion can be found in [33].)

Definition 1.5. *We say that a multiset S in X is* reconstructible from its k-deck *(or* k-reconstructible*) if every T in X with the same k-deck as S is, in fact, isomorphic to S. Similarly, if $f : \mathcal{M}(X) \to Y$ is an arbitrary function, then we say $f(S)$ is* k-reconstructible *if $D_k(T) = D_k(S) \Rightarrow f(T) = f(S)$. More generally we say that $f : \mathcal{M}(X) \to Y$ is* k-reconstructible *if $f(S)$ is k-reconstructible for all finite multisets S in X. This is equivalent to saying that f is reconstructible if and only if it factors through the map $S \mapsto D_k(S)$. Note that if f is k-reconstructible, it must depend only on $[S]$, since $D_k(S)$ does. We will say that* (finite) multisets in X are reconstructible from their k-decks *if the function $S \mapsto [S]_G$ on $\mathcal{M}(X)$ is k-reconstructible (in other words, finite multisets can be identified up to isomorphism from their k-decks).*

2. The method of features.

In this section we present a method central to our results in this paper, that is, the method of features. We show that from an appropriately sized deck of $G \rightarrowtail S$ we can reconstruct the k-deck of any collection of features naturally associated with configurations lying in S. To make this clearer let us give an example that we will use later.

Example 2.1. We would like to associate with a configuration C in $\mathbb{R}^2$ a direction. This requires us to distinguish two points of C to define a reference line, whose direction we will call the *direction of C*. Thus we are led naturally to the notion of an oriented configuration: an *oriented configuration* is a triple $\langle C, x, y\rangle$ consisting of a finite multiset C in $\mathbb{R}^2$ together with points $x, y \in \operatorname{supp}(C)$ with $x \neq y$.

With the example of oriented configurations in mind we describe the general formalism we will use.

Definition 2.1. *A* configuration style *is a finite sequence $a = (a_1, a_2, \ldots, a_r)$ of positive integers. A* colored configuration in style a *is a pair $\langle C, c\rangle$ consisting of a finite* *multiset C in X and a coloring $c : \operatorname{supp}(C) \to \{0, 1, \ldots, r\}$ such that $|c^{-1}(i)| = a_i$ for $i = 1, 2, \ldots, r$. There is a natural action of G on colored configurations, where $g.\langle C, c\rangle = \langle g.C, c \circ g^{-1}\rangle$. Two colored configurations $\langle C, c\rangle$ and $\langle C', c'\rangle$ are therefore* isomorphic *if there exists $g \in G$ such that $g.C = C'$ and $c'(g.x) = c(x)$ for all $x \in C$. As usual we write $[\langle C, c\rangle]_G$ for the isomorphism class of $\langle C, c\rangle$ under the action of G. The* size *of a colored configuration $\langle C, c\rangle$ is simply the size of C. We write $\mathcal{C}_a$ for the collection of all colored configurations in style a. We say that $\langle C, c\rangle$ is an a-*colored configuration in S *if c is an a-coloring of C and $C \subset S$.*

Example 2.2. We define a *pointed configuration* $\langle C, x\rangle$ to be a colored configuration in style (1), that is, a finite multiset C together with one distinguished element $x \in \operatorname{supp}(C)$, which has color 1. An oriented configuration is, similarly, a colored configuration in style $(1, 1)$. The coloring picks out two distinguished elements of $\operatorname{supp}(C)$, the first, x, having color 1 and the second, y, having color 2.

Now we turn to the central reason for discussing colored configurations. We want to talk about a "feature" of a colored configuration, and, eventually, to be able to reconstruct the set of all such features associated with particular classes of configurations. (Recall the example of the direction of an oriented configuration.) Since these features are also the object of a reconstruction problem, we insist that there be a group H acting on the features and that isomorphic colored configurations have isomorphic features.

Definition 2.2. *Given group actions $G \rightarrowtail X$ and $H \rightarrowtail Y$ we define an H-*feature *of a-colored configurations in X to be a function $f : \mathcal{C}_a \to Y$ on colored configurations together with a homomorphism $\phi : G \to H$ such that $f(g.\langle C, c\rangle) = \phi(g).f(\langle C, c\rangle)$ for all $\langle C, c\rangle$ and g. In other words isomorphic configurations have isomorphic features and, moreover, the isomorphism is chosen in a uniform way.*

Definition 2.3. *Let $\mathcal{C}$ be a set of isomorphism classes of a-colored configurations. The* $\mathcal{C}$-list *of S is*

$$L_{\mathcal{C}}(S) = \{\langle C, c\rangle \,:\, C \in \mathcal{P}(S),\ c \text{ an } a\text{-coloring of } C,\ [\langle C, c\rangle]_G \in \mathcal{C}\}.$$

If f is an H-feature of such configurations, then the $\mathcal{C}$-feature set of S *is the multiset*

$$F_{f,\mathcal{C}}(S) = \{f(\langle C, c\rangle) \,:\, \langle C, c\rangle \in L_{\mathcal{C}}(S)\}.$$

Example 2.3. Given an oriented configuration $\langle C, x, y\rangle$ we can associate with it the direction of the directed line segment from x to y. We consider this direction as an element of the circle group $\mathbb{T} = \mathbb{R}/\mathbb{Z}$. This is a $\mathbb{T}$-feature. Its associated homomorphism ϕ maps $g \in R$ (the group of rigid motions) to $\phi(g) = \theta + \mathbb{Z}$, where $2\pi\theta$ is the common angle through which all line segments rotate under the action of g. So if we let $\mathcal{C}$ consist only of the equivalence class of oriented configurations containing two points at distance 1 apart, then the $\mathcal{C}$-list of S is the collection of all ordered pairs of points in S at distance 1 apart, and the feature set of S is the multiset of all directions of these line segments.

Remark 2.1. Note that the $\mathcal{C}$-list of S and the $\mathcal{C}$-feature set F of S are *not* isomorphism invariants, so there is no hope that we will literally be able to reconstruct them. What we hope is that the isomorphism class $[F]_H$ of the feature set will be reconstructible.

Now we are ready for the first theorem of the section. Where unambiguous we shall suppress the qualifiers in H-feature, a-colored configuration, and $\mathcal{C}$-feature set.

Theorem 2.4 (feature theorem). *Let f be a feature of colored configurations (with associated homomorphism ϕ), $\mathcal{C}$ a set of isomorphism classes of colored configurations, each of size at most m, and S a multiset in X. Set $F = F_{f,\mathcal{C}}(S)$, the feature* *set of S. Then the k-deck of $H \rightarrowtail F$ is reconstructible from the mk-deck of $G \rightarrowtail S$. In particular, if multisets in Y are reconstructible from their k-decks, then $[F]_H$ is reconstructible from the mk-deck of S.*

Proof. Note first that there is a natural bijection between the feature set F and the $\mathcal{C}$-list $L = L_{\mathcal{C}}(S)$. Thus there is also a natural bijection between $\mathcal{P}_r(F)$ and the collections $\{\langle C_i, c_i\rangle : i = 1, 2, \ldots, r\} \in \mathcal{P}_r(L)$. We will partition $\mathcal{P}_r(L)$ according to the set union (of multisets) $C = \bigcup_1^r C_i$: note that a given C may arise in many different ways. For a configuration C in X we say that a *$\mathcal{C}$-splitting* of C is a representation of C as a set union $C = \bigcup_1^r C_i$ together with a-colorings c_i for the C_i such that $[\langle C_i, c_i\rangle]_G \in \mathcal{C}$ for $i = 1, 2, \ldots, r$. We can then write

$$f(C) = \{\{f(\langle C_i, c_i\rangle)\}_1^r : \{\langle C_i, c_i\rangle\}_1^r \text{ is a } \mathcal{C}\text{-splitting of } C\}.$$

We obtain the multiset identity

$$\bigoplus_{i \le k} \mathcal{P}_i(F) = \bigoplus_{\substack{C \in \mathcal{P}(S) \\ |C| \le mk}} f(C),$$

and hence

$$D_k(H \rightarrowtail F) = \left\{ [K]_H : K \in \bigoplus_{i \le k} \mathcal{P}_i(F) \right\} = \bigoplus_{\substack{C \in \mathcal{P}(S) \\ |C| \le mk}} \{[L]_H : L \in f(C)\}. \tag{1}$$

The last, crucial, observation is that the multiset of isomorphism classes

$$\{[L]_H : L \in f(C)\}$$

is reconstructible from $[C]_G$. To see this note that if $D \simeq C$, with say $g.C = D$, then the $\mathcal{C}$-splittings of C are isomorphic to the $\mathcal{C}$-splittings of D: if $C = \bigcup_1^k C_i$ and c_i are appropriate colorings, then we set $D_i = g.C_i$ with colorings $d_i(x) = c_i(g^{-1}.x)$ for all $x \in D_i$. The set of features arising from $\{\langle D_i, d_i\rangle\}_1^k$ is isomorphic to that arising from $\{\langle C_i, c_i\rangle\}_1^k$ because we have

$$\begin{aligned} \{f(\langle D_i, d_i\rangle)\} &= \{f(g.\langle C_i, c_i\rangle)\} \\ &= \{\phi(g).f(\langle C_i, c_i\rangle)\} \\ &= \phi(g).\{f(\langle C_i, c_i\rangle)\}. \end{aligned}$$

Thus, by (1), $D_k(F)$ depends only on the collection of all isomorphism classes of elements of $\mathcal{P}(S)$ of size at most mk, which is the mk-deck of $G \rightarrowtail S$. □

We will use the method of features both directly and via the "certification lemma" below. The certification lemma applies to the situation in which S might be infinite and shows that if some subset P of S can be picked out by a property which can be determined from examining small configurations, then we can reconstruct the decks of P from (larger) decks of S.

DEFINITION 2.5. *Recall that if C is a finite multiset of points in X and $x \in \operatorname{supp}(C)$, then we call the pair $\langle C, x\rangle$ a pointed configuration. Let S be a multiset in X and let P be a subset of S. We say that P has a* certificate of size m *if there exists a set $\mathcal{C}$ of isomorphism classes of pointed configurations, each of size at most m, such that P is exactly the set of points in S "pointed at" by elements of $\mathcal{C}$. To be precise, we require*

$$P = \{y \in S : \exists C \subset S, y \in \operatorname{supp}(C) \text{ such that } [\langle C, y\rangle] \in \mathcal{C}\}.$$

Definition 2.6. *If S is a multiset in X and $\mathcal{C}$ is a collection of pointed configurations, then we write*

$$\mathcal{C}(x) = \{\langle C, y\rangle : C \in \mathcal{P}(S), y \in \operatorname{supp}(C) \text{ such that } [\langle C, y\rangle] \in \mathcal{C}\}.$$

We also define $\lambda_{\mathcal{C}}(x) = |\mathcal{C}(x)|$.

Lemma 2.7 (certification lemma). *Let S be a subset of X and P be a subset of S having a certificate of size, say, m, $\mathcal{C}$. We can reconstruct the k-deck of the multiset P^{λ}, consisting of $\lambda_{\mathcal{C}}(x)$ copies of x for each $x \in P$, from the mk-deck of S. In particular, if $[P^{\lambda}]_G$ is reconstructible from its k-deck, then it is reconstructible from the mk-deck of S, as is $[P]$.*

Proof. The map taking $p : \langle C, x\rangle \mapsto x$ is trivially a G-feature of pointed multisets (with associated homomorphism the identity map $G \to G$) and, moreover, P^{λ} is exactly the feature set $F_{p,\mathcal{C}}(S)$. Thus by Theorem 2.4 the claims of the lemma hold. □

3. The circle and the plane. In this section we prove that finite multisets in the circle are 6-reconstructible and that finite multisets of $\mathbb{R}^2$ are 18-reconstructible.

We deal first with the reconstructibility of multisets of the circle group $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ acting on itself by translation. It turns out that we are able to relate this problem to that of reconstructing multisets in the cyclic group $\mathbb{Z}_n$. Because of this it is helpful to identify $\mathbb{Z}_n$ with the specific subgroup $\{i/n + \mathbb{Z} : i = 0, 1, \ldots, n-1\} < \mathbb{T}$. We will also make use of the fact that $\mathbb{T}$ is a topological group with metric

$$d(r + \mathbb{Z}, s + \mathbb{Z}) = \min\{|r' - s'| : r' \in r + \mathbb{Z}, s' \in s + \mathbb{Z}\}.$$

We shall often identify elements of $\mathbb{T}$ with elements of $[0, 1) \subset \mathbb{R}$.

Theorem 3.1. *All finite multisets in $\mathbb{T}$ are 6-reconstructible.*

We give two proofs of this result. The first proof considers the subgroup of $\mathbb{T}$ generated by the multiset S that we wish to reconstruct; the second proof works by approximating S by a "nearby" copy of $\mathbb{Z}_n$ (standard results on Diophantine approximation imply that such a copy exists).

Proof. [First proof.] Given finite multisets S_1, S_2 in $\mathbb{T}$ with the same 6-deck, we will show that S_1 is a translate of S_2. Consider the subgroup G of $\mathbb{T}$ generated by $S_1 \bigcup S_2$. It is a finitely generated subgroup of $\mathbb{T}$, and therefore there exist integers k, n such that $G \simeq \mathbb{Z}_k \oplus \mathbb{Z}^n$. Let $\theta : G \to \mathbb{Z}_k \oplus \mathbb{Z}^n$ be an isomorphism, and let $T_i = \theta(S_i)$. Then T_1, T_2 are multisets of $\mathbb{Z}_k \oplus \mathbb{Z}^n$ with the same 6-deck.

Represent the elements of $\mathbb{Z}_k \oplus \mathbb{Z}^n$ by sequences of $n+1$ integers. The sequences $(a_1, a_2, \ldots, a_{n+1})$ and $(b_1, b_2, \ldots, b_{n+1})$ represent the same element if $k|(a_1 - b_1)$ and $a_i = b_i$ for all $i > 1$. For $2 \le i \le n+1$, say that a_i is *the ith coordinate* of $(a_1, a_2, \ldots, a_{n+1})$. Let x_i be the smallest ith coordinate of elements of $T_1 \bigcup T_2$, and let y_i be the largest. Finally, let $(p_2, \ldots, p_{n+1})$ be a sequence of distinct primes such that p_i is not a factor of k, and $p_i > 2(y_i - x_i)$.

Let H be the subgroup of $\mathbb{Z}_k \oplus \mathbb{Z}^n$ generated by the elements $(0, p_2, 0, \ldots, 0)$, $(0, 0, p_3, 0, \ldots, 0)$, …, $(0, 0, \ldots, p_{n+1})$, and let

$$\theta' : \mathbb{Z}_k \oplus \mathbb{Z}^n \to (\mathbb{Z}_k \oplus \mathbb{Z}^n)/H \simeq \mathbb{Z}_{kp_2p_3,\ldots,p_{n+1}}$$

be the quotient map. If $T_i' = \theta'(T_i)$, then T_1' and T_2' have the same 6-deck. Since these multisets are multisets of a cyclic group, Theorem 1.1 implies that they are translates.

Therefore there exists a translate T of T_1 and a bijection $\gamma : T \to T_2$ such that for all $t \in T$, $t - \gamma(t) \in H$. Furthermore, by picking T wisely, we may assume that there exists t such that $t = \gamma(t)$ for some t. Then the ith coordinate of t is between x_i and y_i. Therefore for any $u \in T$, the ith coordinate of u is between $x_i-(y_i-x_i) = 2x_i-y_i$ and $y_i + (y_i - x_i) = 2y_i - x_i$. Furthermore, the ith coordinate of $\gamma(u)$ is between x_i and y_i. Therefore the ith coordinate of $u - \gamma(u)$ is between $2(x_i - y_i)$ and $2(y_i - x_i)$ and is definitely less in magnitude than p_i. Since $u - \gamma(u) \in H$, $u = \gamma(u)$. Since this holds for all u, $T = T_2$ (as multisets), and hence T_1 and T_2 are translates. Since θ was an isomorphism, it follows that S_1 and S_2 are translates, and hence multisets in $\mathbb{T}$ are 6-reconstructible. □

Proof. [Second proof.] Given a finite multiset S in $\mathbb{T}$, we will show that it is reconstructible from its 6-deck. First note that we may assume, by translating S if necessary, that $0 \in S$. For $T \in \mathcal{M}(\mathbb{T})$ we will write $\Delta(T) = \{t - t' : t, t' \in T\}$ for the multiset of differences of elements of T. Let $\Delta_1 = \Delta(S)$ and $\Delta_2 = \Delta(\Delta(S))$. It is clear that Δ_1, and hence Δ_2, can be reconstructed from the 2-deck of S; note that $S \subset \Delta_1 \subset \Delta_2$.

By standard results concerning Diophantine approximation (see, for instance, [12, Chap. 1, Prop. 2]) there exists $\rho > 0$ and a sequence $n_i \to \infty$ such that

$$\epsilon_i := \max\{d(\delta, \mathbb{Z}_{n_i}) : \delta \in \Delta_2\} < 1/n^{1+\rho}.$$

(This approximation is used in a similar context in [1].) In particular we may assume

$$\epsilon_i < \frac{1}{4n_i} < \frac{1}{4}\min\{d(\delta, 0) : \delta \in \Delta_2\}. \tag{2}$$

We shall say that n_i is *good for S* if it satisfies (2). Notice that for any particular n, the property that n is good for S is reconstructible from the 2-deck of S. For each of the n_i we define a "projection" $\pi : \Delta_2 \to \mathbb{Z}_{n_i}$ by letting $\pi(\delta)$ be the point in $\mathbb{Z}_{n_i}$ closest to δ. There is no possible ambiguity since by (2) the nearest element of $\mathbb{Z}_{n_i}$ is within distance $\epsilon_i < 1/4n_i$. Moreover, π is injective on Δ_1: if $\delta, \delta' \in \Delta_1$ have $\pi(\delta) = \pi(\delta')$, then $d(\delta, \delta') \le 2\epsilon_i < 1/n_i$ while $\delta - \delta' \in \Delta_2$. By (2) this implies that $\delta = \delta'$.

Now we define $S_{n_i} = \pi(S) = \{\pi(x) : x \in S\}$. It is easily checked that the 6-deck of S_{n_i} is reconstructible from the 6-deck of S, and hence that $[S_{n_i}]$ is reconstructible. Now take an arbitrary orientation of each S_{n_i}: dropping to a convergent subsequence yields an orientation of S. □

We turn now to the proof of our central result, which states that finite multisets in the plane are reconstructible from their 18-decks.

THEOREM 3.2. *Any finite multiset S in $\mathbb{R}^2$ is reconstructible, up to the action of the group R of rigid motion acting on the plane, from its* 18*-deck.*

Proof. We begin by defining a $\mathbb{T}$-feature of configurations contained in S. We identify, in the natural way, the collection of unit vectors in $\mathbb{R}^2$ with the group $\mathbb{T}$. To be precise let $\psi : \{u \in \mathbb{R}^2 : |u| = 1\} \to \mathbb{T}$ be defined by $\psi((x_1, x_2)) = \alpha/(2\pi)$ if $(x_1, x_2) = (\sin\alpha, \cos\alpha)$. As in the discussion in section 2, recall that an oriented configuration $\langle C, x, y\rangle$ is a finite multiset C in $\mathbb{R}^2$ together with points $x, y \in \operatorname{supp}(C)$ with $x \ne y$. The *direction* of $\langle C, x, y\rangle$ is the element $u(\langle C, x, y\rangle) = \psi((x - y)/|x - y|)$ of $\mathbb{T}$.

We claim that u is a $\mathbb{T}$-feature of oriented configurations. To see this, note that there is a homomorphism ρ from R to $\mathbb{T}$ which takes g to $\alpha/2\pi$ if g rotates all line segments through α radians. Moreover, $u(g.\langle C, x, y\rangle) = \rho(g).u(\langle C, c\rangle)$. If $\mathcal{C}$ is any collection of isomorphism classes of oriented configurations, we define the *orientation* *set* of $\mathcal{C}$ (in S) to be the multiset in $\mathbb{T}$ given by

$$\begin{aligned} O(\mathcal{C}) &= F_{u,\mathcal{C}}(S) \\ &= \{u(\langle C,x,y\rangle) \,:\, C \in \mathcal{P}(S), x,y \in \operatorname{supp}(C), x \neq y, [\langle C,x,y\rangle] \in \mathcal{C}\}. \end{aligned}$$

By Theorem 2.4, if all the configurations in $\mathcal{C}$ have size at most m, then we can reconstruct $[O(\mathcal{C})]_{\mathbb{T}}$ from the $6m$-deck of S.

Similarly, if $\epsilon : \mathcal{C} \to \mathbb{T}$ is an arbitrary function, then the map $\langle C,x,y\rangle \mapsto u(\langle C,x,y\rangle) + \epsilon([\langle C,x,y\rangle])$ is also a $\mathbb{T}$-feature, with the same associated homomorphism. Thus, by the same result, we can also reconstruct $[O(\mathcal{C},\epsilon)]_{\mathbb{T}}$ from the $6m$-deck of S, where

$$\begin{aligned} O(\mathcal{C},\epsilon) = \{u(\langle C,x,y\rangle)+&\epsilon([\langle C,x,y\rangle]) \,: \\ &C \in \mathcal{P}(S), x,y \in \operatorname{supp}(C), x \neq y, [\langle C,x,y\rangle] \in \mathcal{C}\}. \end{aligned}$$

Suppose now that $\mathcal{C} = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_t\}$. We will show that we can reconstruct $[(O(\Gamma_i))_{i=1}^t]_{\mathbb{T}}$ from the $6m$-deck of S. (Note that the relevant $\mathbb{T}$ action is that on $\mathcal{M}(\mathbb{T})^t$ given by $s.(A_i)_{i=1}^t = (s.A_i)_{i=1}^t$.) To see this let $\Delta = \{t - t' \,:\, t,t' \in O(\mathcal{C})\}$ and let $W \subset \mathbb{R}$ be the subspace of $\mathbb{R}$ (considered as a vector space over $\mathbb{Q}$) generated by $\Delta \cup \{1\}$. This is clearly independent of the choice of representatives for elements of Δ. Let $\epsilon_1, \epsilon_2, \ldots, \epsilon_t$ be elements of $\mathbb{R}$ linearly independent of each other and W, and define $\epsilon : \mathcal{C} \to \mathbb{T}$ by $\epsilon(\Gamma_i) = \epsilon_i$. As above we can reconstruct $[O(\mathcal{C},\epsilon)]_{\mathbb{T}}$ from the $6m$-deck of S. Now pick $O \in [O(\mathcal{C},\epsilon)]$ and consider $x,y \in O$. We have $O = O(\mathcal{C},\epsilon) + s$ for some unknown s. If $x \in O(\Gamma_i)+\epsilon_i+s$ and $y \in O(\Gamma_j)+\epsilon_j+s$, then $x-y \in W+\epsilon_i-\epsilon_j$. Thus we can recognize, from the difference $x-y$, that $x \in O(\mathcal{C}_i)+\epsilon_i+s$ and that $y \in \mathcal{C}_j+\epsilon_j+s$, and we are therefore able to label every element of O with the Γ_i from which it came. From this we deduce $(O(\Gamma_i)+s)_{i=1}^t$ for some fixed unknown $s \in \mathbb{T}$ by subtracting ϵ_i from every direction labeled with Γ_i. Hence we can reconstruct $[(O(\Gamma_i))_{i=1}^t]_{\mathbb{T}}$ from the $6m$-deck of S.

We are now ready to finish the proof. The group R of rigid motions contains a normal subgroup $\ker(\rho)$ isomorphic to $\mathbb{R}^2$ and consisting of the translations. We refer to this subgroup as $\mathbb{R}^2$ in what follows. The quotient $R/\mathbb{R}^2$ is isomorphic to $\mathbb{T}$.

Let $(\Gamma_i)_{i=1}^t$ be a list of all equivalence classes of oriented configurations of size 3 in S (deducible from the 3-deck of S), and let $(O_i)_{i=1}^t$ be a representative of $[(O(\Gamma_i))_{i=1}^t]_{\mathbb{T}}$. Note that we can determine a suitable $(O_i)_{i=1}^t$ from the 18-deck of S; we will show that from this information we can reconstruct $[D_3(\mathbb{R}^2 \rightarrowtail S)]_{\mathbb{R}^2}$. Here it is crucial to understand what we are reconstructing. $\mathbb{R}^2$ acts on itself by translation. In turn there is an action of $\mathbb{T}$ on $\mathbb{R}^2$-isomorphism classes by $s.[C]_{\mathbb{R}^2} = [g.C]_{\mathbb{R}^2}$, where $g \in R$ is any rigid motion with $\rho(g) = s$, since if $\rho(g_1) = \rho(g_2)$, then $g_1 g_2^{-1}$ is a translation. Hence $\mathbb{T}$ acts on multisets of $\mathbb{R}^2$-isomorphism classes, and in particular on the deck $D_3(\mathbb{R}^2 \rightarrowtail S)$.

Starting from $(O_i)_{i=1}^t \in [(O(\Gamma_i))_{i=1}^t]_{\mathbb{T}}$ we build an element D of $[D_3(\mathbb{R}^2 \rightarrowtail S)]_{\mathbb{R}^2}$; i.e., we reconstruct $D_3(\mathbb{R}^2 \rightarrowtail S)$ up to a global rotation. For any $[C]_R \in D_3(R \rightarrowtail S)$ one can work out which Γ_i arise from orientations of C, and for each one the sequence $(O_i)_{i=1}^t$ tells us which directions to pick for the corresponding elements of D. Clearly we have $D = D_3(\mathbb{R}^2 \rightarrowtail r.S)$ for some unknown $r \in R$. Now pick some unit vector $u \in \mathbb{R}^2$ such that no two points $x,y \in r.S$ have $\langle u,x\rangle = \langle u,y\rangle$; this property can be easily checked from D (indeed, from the 2-deck of $\mathbb{R}^2 \rightarrowtail r.S$) since it is invariant under translations of S. Then let $\lambda = \max\{\langle u,x\rangle - \langle u,y\rangle \,:\, x,y \in r.S\}$. Again, λ can be computed from the 2-deck of $\mathbb{R}^2 \rightarrowtail r.S$. Now $r.S$ can be recovered up to translation: it is a translate of

$$T = \{x : \{0, x, \lambda u\} \in D_3(\mathbb{R}^2 \rightarrowtail r.S)\}.$$

Thus, from some unknown $r \in R, x \in \mathbb{R}^2$ we have $r.S = x + T$. In particular $[S]_R$ is determined by the 18-deck of S. □

4. Infinite subsets of $\mathbb{R}^2$. In this section we discuss the reconstructibility of some infinite subsets of the plane. We shall no longer be concerned with multisets. We immediately run into several examples of nonreconstructible sets.

Example 4.1. Let $S = (0, 1)$ and let $S' = (0, 1)\backslash\{\frac{1}{2}\}$. Clearly these sets are not isomorphic. On the other hand, their decks both consist of an (uncountably) infinite number of copies of every finite configuration which is linear and has diameter strictly less than 1. Moreover, these examples have the same k-deck (for every k) as any set of the form $(0, 1) \setminus C$, where C is any countable subset of $(0, 1)$. Since these are all mutually nonisomorphic this gives quite a large range of examples of nonreconstructible sets. (These examples are all reconstructible from their $\aleph_0$-decks.)

Example 4.2. Similarly, if we take the disc $\{x \in \mathbb{R}^2 : |x| \leq 1\}$, it has the same k-deck, for every k, as the disc with a countable number of points (none of which is the origin) removed. Every configuration for which a copy appears in the disc can be rotated (in uncountably many ways) to avoid the missing points. In fact even the $\aleph_0$-deck does not distinguish these examples from one another. Thus the disc is not even $\aleph_0$-reconstructible.

Example 4.3. Let $\mathbb{P}$ be the standard symmetric probability distribution on the power set $\mathcal{P}(\mathbb{N})$ of $\mathbb{N} \subset \mathbb{R}^2$. Pick two subsets $S, S' \subset \mathbb{N}$ at random according to $\mathbb{P}$. With probability 1 they will each contain infinitely many copies of every finite subset of $\mathbb{N}$ (and of course no copies of any other configuration) and will not be isomorphic. Thus we can find countable subsets of the plane that are not reconstructible.

We have given examples showing that if S is not compact, or has an infinite automorphism group, then S may not be reconstructible. The next result proves that otherwise there exists N_S depending only on S such that given an arbitrary set $S' \subset \mathbb{R}^2$ either $S \simeq S'$ or the N_S-decks of S and S' are different. We call this property of S *finitely reconstructible.*

Theorem 4.1. *Every compact subset of the plane with a finite automorphism group is finitely reconstructible.*

Our proof of this theorem will use the certification lemma, Lemma 2.7, to show that the existence of even one configuration C which appears in S but does not appear infinitely often in S is enough to ensure that S is finitely reconstructible.

Definition 4.2. *If $S \subset \mathbb{R}^2$ and $C \subset S$ is a finite subset with the property that the deck of S contains only finitely many copies of $[C]_R$ (or, equivalently, that S contains only finitely many copies of C), then C is called a* characteristic configuration *in S.*

Lemma 4.3. *If $S \subset \mathbb{R}^2$ contains a characteristic configuration C of size k, then S is $18(2k + 1)$-reconstructible.*

Proof. Let S_0 be the subset of S consisting of points belonging to at least one copy of C. For each $D \subset \mathbb{R}_+$ let S_D be the subset of S containing all points whose distances to at least two points of S_0 belong to D. Note that S_0 is finite and thus S_D is finite for all finite D. Also S_D is an increasing function of D, and $S = \bigcup_{|D|<\infty} S_D$.

We claim that for any D we can reconstruct S_D from the $18(2k + 1)$-deck of S. Certainly S_D has a certificate of size $2k + 1$ since $y \in \mathcal{S}_D$ if and only if it belongs to a pointed configuration $\langle C_1 \cup C_2 \cup \{y\}, y\rangle$, where $C_1, C_2 \simeq C$ and at least two of the distances from y to points in $C_1 \cup C_2$ belong to D. We therefore let $\mathcal{C}$ be the set of isomorphism classes of pointed configurations of this sort. By Lemma 2.7 and Theorem 3.2, S_D is reconstructible from the $18(2k+1)$-deck of S.

Now let H be the automorphism group of S. Clearly, since S has a characteristic configuration, H must be finite. For finite subsets D of $\mathbb{R}_+$ let H_D be the automorphism group of S_D. Clearly, $H \leq H_D$ for all finite D and, if $E \supset D$, then $H_E \leq H_D$. We claim that there is some finite $D_0 \subset \mathbb{R}_+$ such that $H = H_D$. To see this pick D_0 with $|H_{D_0}|$ minimal. Now suppose that $H < H_{D_0}$. Pick $h \in H_{D_0} \backslash H$. There must be some $x \in S$ with $hx \notin S$. Now pick $E \supset D_0$ with $x \in S_E$. Then $H_E \leq H_{D_0}$ and $h \in H_{D_0} \backslash H_E$, contradicting the minimality of $|H_{D_0}|$.

Now since we can reconstruct S_D for all finite D we build

$$\{[S_D]_R \,:\, |D| < \infty, D_0 \subset D\}.$$

We fix a copy T_0 of S_{D_0} and choose $T_D \in [S_D]_R$ such that the copy of S_{D_0} in T_D is equal to T_0. We claim that $\bigcup_{|D|<\infty, D_0 \subset D} T_D \simeq S$. If $D, E \supset D_0$ and we have chosen T_D and T_E to agree on T_0, then $T_D = g_D S_D$ and $T_E = g_E S_E$ for some $g_D, g_E \in R$ such that $g_D^{-1} g_E T_0 = T_0$. Thus, by the minimality of H_{D_0}, we have $g_D^{-1} g_E T_D = T_D$ and $g_D g_E^{-1} T_E = T_E$. Thus T_D and T_E are consistent, and a similar argument shows that both agree with $T_{D \cup E}$. The union of $\{T_D \,:\, D_0 \subset D, |D| < \infty\}$ is therefore a set isomorphic to S. □

Before completing the proof of Theorem 4.1 we note some simple facts concerning subgroups of R. We write $\mathbb{T}_x$ for the subgroup of R consisting of all rotations about x, and $\mathbb{Z}_{n,x}$ for the subgroup of all rotations about x through an integer multiple of $2\pi/n$ radians. We will need some elementary topological properties of R. We note that any element $g \in R$ rotates all line segments through some fixed angle $\alpha(g)$ and we define a metric on R by $d(g, g') = |g((0,0)) - g'((0,0))| + d_{\mathbb{T}}(\alpha(g), \alpha(g'))$. This metric makes R into a topological group.

Proposition 4.4. *If K is any compact subgroup of R, then there exists x in $\mathbb{R}^2$ such that K is either $\mathbb{T}_x$ or $\mathbb{Z}_{n,x}$ for some n.*

Proof. Clearly, the set of iterates of a (nontrivial) translation form an infinite discrete set, and thus K cannot contain a translation. Since the commutator of two rotations about different centers is a translation, K cannot contain such a pair. Therefore K consists purely of rotations about some fixed center x. The set of allowed rotations is either discrete, in which case K is easily seen to be $\mathbb{Z}_{n,x}$ for some n, or dense in $\mathbb{T}_x$, in which case (since K is closed) $K = \mathbb{T}_x$. □

Lemma 4.5. *If S is a compact subset of $\mathbb{R}^2$ with $\operatorname{Aut}(S)$ finite and $C \subset S$ finite, then for every $\epsilon > 0$ there exists a finite superset $E_\epsilon \supset C$ such that whenever $E_1, E_2 \subset S$ have $E_1, E_2 \simeq E_\epsilon$ and $g \in R$ maps D_1 to E_2, then g is within ϵ of some automorphism of R.*

Proof. For any finite subset $E \subset S$ we set

$$K_E = \{g \in R \,:\, g(E) \subset S\} \setminus \{g \in R \,:\, d(g, \operatorname{Aut}(S)) < \epsilon\}.$$

This is clearly a compact subset of R. Suppose that no finite subset E as described in the lemma exists. Then the collection $\{K_E \,:\, E \text{ finite}, C \subset E\}$ has the finite intersection property and thus $\bigcap_{|E|<\infty, E \supset C} K_E$ is nonempty. This intersection consists, however, of only rigid motions which map S to S and are at least ϵ away from any automorphism of S, which is a contradiction. □

We will use Lemma 4.5 to restrict our search for a characteristic configuration in S to subsets which have only "nearby" copies. To be precise, if E_1, $E_2 \subset S$ are both copies of one another, we will write $d(E_1, E_2)$ for $\min\{d(g, \mathrm{id}) \,:\, g(E_1) = E_2\}$.

Proof of Theorem 4.1. Note first that $\mathrm{Aut}(S)$, being finite, must be $\mathbb{Z}_{n,x}$ for some n, x, by Proposition 4.4. Put $\epsilon = 1/2n$. Let M be the diameter of S and let C consist of two points in S at distance M apart. By Lemma 4.5 there exists E containing C such that any two copies of E in S are related by a rigid motion which is within ϵ of an automorphism of S. Pick a copy E' of E with $E' \subset S$ and distinguish a copy C' of C in E'. From all images $g(E')$ in S with $d(g, \mathrm{id}) \leq \epsilon$ pick a pair E_1, E_2 with the angle between their distinguished copies of C' being maximal. This is possible by compactness. Note that it is an elementary geometric fact that, since $M = \mathrm{diam}(S)$, there is at most one copy of C with any given orientation. Now it is clear that $E'' = E_1 \cup E_2$ is a characteristic configuration for S; indeed $[E'']$ occurs with multiplicity at most $|\,\mathrm{Aut}(S)|$ in the k-deck of S. If $F'' \subset S$ is a copy of E'', then by hypothesis $F'' = g(E'')$ for some $g \in R$ with $d(g, \mathrm{Aut}(S)) \leq \epsilon$. Suppose that $h \in \mathrm{Aut}(S)$ has $d(h, g) < \epsilon$. Thus $h^{-1}(F'')$ is the image of E'' under a rigid motion at most ϵ from the identity. This, however, by the construction of E'' ensures that $h^{-1}(F'') = E$, and so $F'' = h(E'')$. In summary, the only copies of E'' in S are the images of E'' under $\mathrm{Aut}(S)$. Now by Lemma 4.3 we are done. □

Example 4.4. Consider the “notched disc”

$$N_\epsilon = \{x : |x| \leq 1, |x - (1, 0)| \geq \epsilon\}.$$

Any finite configuration C for which the multiplicity of $[C]$ in $D(N_\epsilon)$ is different than in the deck of the unnotched disc must have $|C| \geq \pi / \sin^{-1} \epsilon$ (since otherwise either C would not turn up in the disc or uncountably many rotations of C would fit in the notched disc). Thus there is no uniform bound N such that all compact subsets of $\mathbb{R}^2$ with a finite automorphism group are reconstructible from their N-decks.

Remark 4.1. It is worth remarking that if S, T are compact subsets with $\mathrm{Aut}(S) = \mathbb{T}_x$ and $\mathrm{Aut}(T) = \mathbb{T}_y$ and $D_3(S) = D_3(T)$, then $S \simeq T$. To see this note that, for such S with diameter $2R$, if we pick an arbitrary unit vector u we have $S \simeq \mathbb{T}_{(0,0)} . \{\lambda u : \{-Ru, \lambda u, Ru\} \in D_3(S)\}$.

We have seen that if S is bounded but not closed, then it may not be reconstructible even from its $\aleph_0$-deck. However, we *can* reconstruct the closure of S.

THEOREM 4.6. *If $S \subset \mathbb{R}^2$ is bounded, then $[\bar{S}]_R$ can be reconstructed from the $(< \omega)$-deck of S.*

Proof. Let $K = \bar{S}$. Given two finite subsets $C, C' \subset \mathbb{R}^2$ we say that they are ϵ-copies of one another if there exists a map $\phi : C \to C'$ and a rigid motion $g \in R$ such that $|\phi(x) - g(x)| \leq \epsilon$ for all $x \in C$. By compactness, for any finite subset $C \subset \mathbb{R}^2$, the deck of K contains $[C]$ if and only if for all $\epsilon > 0$ there exists an ϵ-copy C_ϵ of C such that $[C_\epsilon] \in D(S)$. However, it may be hard to compute the multiplicity of $[C]$ in $D(K)$. It turns out that we can get away with using only the “reduced deck” of K: the set of isomorphism classes of finite subsets of K. Let $\tilde{D} = \tilde{D}(K)$ be this set. By the observation above, $\tilde{D}$ is reconstructible from $D(S)$.

We now show that the automorphism group of K is reconstructible (up to isomorphism) from $\tilde{D}$. Note first that by Proposition 4.4 the automorphism group of K, which is certainly compact, is either $\mathbb{Z}_{n,x}$ or $\mathbb{T}_x$ for some $x \in \mathbb{R}^2$. If H is a group of rigid motions, we say that K is *H-full* if every finite subset $C \subset K$ can be extended to a configuration $C_G \subset K$ with $H \leq \mathrm{Aut}(C_H)$. Clearly, if $H \leq \mathrm{Aut}(K)$ is finite, then K is H-full, since for $C \subset K$ we can take C_H to be the union $\bigcup_{h \in H} h(C)$. In particular, if $\mathrm{Aut}(K)$ is infinite, then K is $\mathbb{Z}_n$-full for all n. On the other hand, if $\mathrm{Aut}(K)$ is finite, then we know from the proof of Lemma 4.3 that there is a (finite) subset $C \subset K$ such that $\mathrm{Aut}(C) = \mathrm{Aut}(K)$ and every extension D with $C \subset D \subset K$

has $\mathrm{Aut}(D) \leq \mathrm{Aut}(K)$. Thus if $\mathrm{Aut}(K)$ is finite, then K is H'-full if and only if $H' \leq \mathrm{Aut}(K)$. By this observation we see that the isomorphism type of $\mathrm{Aut}(K)$, that is, $\mathbb{Z}_n$ or $\mathbb{T}$, can be reconstructed from $\tilde{D}$.

Now that we know $\mathrm{Aut}(K)$ we can reconstruct as follows. If $\mathrm{Aut}(K)$ is finite, then

$$K = \bigcup_{D \supset C, [D] \in \tilde{D}} D,$$

where C is as above; moreover, the right-hand side can be reconstructed up to rigid motion from $\tilde{D}$. On the other hand if $\mathrm{Aut}(K)$ is infinite, then we can reconstruct K straightforwardly from the reduced 3-deck of K, which can be determined from $\tilde{D}$. $\square$

We can also attempt to weaken the boundedness hypothesis. However, as the following example shows, we cannot remove it altogether.

Example 4.5. There are closed subsets of the plane that cannot be reconstructed even from the set of isomorphism classes of *all* their subsets. For instance, $S = \{(x,y) : x,y \geq 0\}$ and $T = \{(x,y) : x,y \geq 0, x+y \geq 1\}$ each contain a copy of the other and both sets contain any configuration (of arbitrary cardinality) either uncountably often or not at all.

In Theorem 4.1 the compactness of S serves to limit the complexity of S. However, some unbounded sets are finitely reconstructible. We impose a different condition to ensure that the complexity is not too high, namely that S can be covered by a finite number of lines. This is clearly not enough to prove even finite reconstructibility, as Example 4.3 shows. However, the counterexamples are all contained in finite collections of parallel lines. This last property is of course equivalent to that of $P_u(S)$ being finite for some unit vector u, where P_u is the orthogonal projection from $\mathbb{R}^2$ onto the line through the origin perpendicular to u.

THEOREM 4.7. *If $S \subset \mathbb{R}^2$ is contained in the union of the finite set of lines $\mathcal{L}$ and the projection $P_u(S)$ is infinite for all unit vectors u, then S is* 162*-reconstructible.*

We first prove a lemma showing that certain configurations appear only finitely many times on a given collection of lines.

LEMMA 4.8. *If L_1, L_2, L_3 are three pairwise nonparallel lines in the plane and C is a configuration consisting of three points x_1, x_2, x_3 in a straight line with $|x_1 - x_2| = d_1$ and $|x_2 - x_3| = d_2$, then there are only finitely many images $g(C)$ of C with $g(x_i) \in L_i$, $i = 1,2,3$.*

Proof. Parameterize the lines L_1 and L_2 using parameters s and t, respectively: $z_1(s) = a_1 + sv_1$ and $z_2(t) = a_2 + tv_2$. Pick $w_3 \in \mathbb{R}^2 \backslash \{0\}$, $\lambda \in \mathbb{R}$, such that $L_3 = \{z : \langle z, w_3 \rangle = \lambda\}$. The condition $|z_1(s) - z_2(t)|^2 = d_1^2$ is a quadratic equation for s,t. Let $P(s,t) = z_2(t) + \frac{d_2}{d_1}(z_2(t) - z_1(s))$. This is the third point of the copy of C having $g(x_1) = z_1(s)$ and $g(x_2) = z_2(t)$. Values of the parameters s,t describe a copy of C if and only if (s,t) lies on the conic $|z_1(s) - z_2(t)|^2 - d_1^2 = 0$ and the straight line $\langle P(s,t), w_3 \rangle - \lambda = 0$, so there are at most two solutions. $\square$

Proof of Theorem 4.7. Let $\mathcal{L}$ be partitioned into parallel classes of lines $\mathcal{L}_1$, $\mathcal{L}_2$,...,$\mathcal{L}_k$, parallel to directions $u_1, u_2, \ldots, u_k$. Let the *ratios appearing in the ith parallel class* be the set of ratios $|x_2 - x_1|/|x_3 - x_1|$, where $x_1, x_2, x_3 \in \bigcup \mathcal{L}_i$ are collinear points belonging to distinct lines in $\mathcal{L}_i$. Note that this set is finite and is the same as if one required that the line on which x_1, x_2, x_3 lie were perpendicular to those in $\mathcal{L}_i$. Let us write R_i for this set of ratios and let $R = \bigcup_1^k R_i$. Pick a line $L \in \mathcal{L}$ containing infinitely many points; we may assume that $L \in \mathcal{L}_1$. Pick 3 points $x_1, x_2, x_3 \in L \in \mathcal{L}_1$ such that the ratio $|x_2 - x_1|/|x_3 - x_1|$ does not belong to R. This is possible simply by picking x_1 and x_2 arbitrarily on L and then avoiding a finite number of possibilities for x_3. Now consider $P_{u_1}(S)$. It is, by hypothesis, infinite, and therefore there exists $y \in S$ such that $P_{u_1}(y) \notin P_{u_1}(\bigcup \mathcal{L}_1)$. We claim that $\{x_1, x_2, x_3, y\}$ is a characteristic configuration in S. Note first that by Lemma 4.8 the configuration $\{x_1, x_2, x_3\}$ occurs only a finite number of times with the images of x_1, x_2, x_3 not all on one line from L_i. On the other hand, given a line $L \in \mathcal{L}_i$ there exist only finitely many copies of $\{x_1, x_2, x_3, y\}$ with the images of x_1, x_2, x_3 on L since there are at most two such copies with the image of y on L' for each $L' \in \mathcal{L} \setminus \{L\}$. By Lemma 4.3, it follows that S is (18×9)-reconstructible. □

5. Further questions. There are several extremely interesting questions still open. In this paper we have shown that finite subsets of the plane can be reconstructed from their 18-decks. However, we know very little in higher dimensions.

CONJECTURE 5.1. *For all $n \geq 1$ there exists $k = k(n)$ such that every finite multiset in $\mathbb{R}^n$ can be reconstructed from its k-deck.*

The main difficulty here seems to be reconstructing finite subsets of S^{n-1} under the action of $SO(n)$. In section 3 we showed that finite subsets of S^1 are 6-reconstructible under the action of $SO(1)$. In [33] we show that a similar result for S^{n-1} would prove Conjecture 5.1. Note that, for $n \geq 3$, $SO(n)$ presents some difficulties absent in the planar case: $SO(n)$ is nonabelian, and there is no "approximating sequence" of finite subgroups analogous to $\mathbb{Z}_n < \mathbb{T}$.

A seemingly more general question is that of reconstructing finite multisets in $\mathbb{R}^n$ up to isometry from the k-deck (given up to isometry). In fact, it is shown in [33] that if finite multisets in $\mathbb{R}^n$ are reconstructible up to rigid motion from their k-decks, then they can be reconstructed up to isometry from their $2k$-decks (given up to isometry).

Returning to two dimensions, we can ask about the reconstructibility of the hyperbolic plane under the action of its isometry group. Very much along this line also is the problem of reconstructing subsets of the extended complex plane $\mathcal{C}_\infty$ under the action of the group of Möbius transformations. We conjecture that in both cases there is a constant k such that all finite multisets are k-reconstructible (under the appropriate group action).

REFERENCES

[1] N. Alon, Y. Caro, I. Krasikov, and Y. Roditty, *Combinatorial reconstruction problems*, J. Combin. Theory Ser. B, 47 (1989), pp. 153–161.

[2] L. Babai, *Automorphism groups, isomorphism, reconstruction*, in Handbook of Combinatorics, Vol. 1, 2, Elsevier, Amsterdam, 1995, pp. 1447–1540.

[3] J. A. Bondy, *A graph reconstructor's manual*, in Surveys in Combinatorics, (Guildford, 1991), Cambridge University Press, Cambridge, UK, 1991, pp. 221–252.

[4] J. A. Bondy and R. L. Hemminger, *Graph reconstruction—a survey*, J. Graph Theory, 1 (1977), pp. 227–268.

[5] T. Brylawski, *On the nonreconstructibility of combinatorial geometries*, J. Combin. Theory Ser. B, 19 (1975), pp. 72–76.

[6] T. H. Brylawski, *Reconstructing combinatorial geometries*, in Graphs and Combinatorics (Proc. Capital Conf., George Washington Univ., Washington, D.C., 1973), Lecture Notes in Math. 406, Springer, Berlin, 1974, pp. 226–235.

[7] P. J. Cameron, *Some open problems on permutation groups*, in Groups, Combinatorics & Geometry (Durham, 1990), Cambridge University Press, Cambridge, UK, 1992, pp. 340–350.

[8] P. J. Cameron, *Stories about groups and sequences*, Des. Codes Cryptogr., 8 (1996), pp. 109–133.

[9] P. J. Cameron, *Stories from the age of reconstruction*, Congr. Numer., 113 (1996), pp. 31–41.

[10] F. Harary, *On the reconstruction of a graph from a collection of subgraphs*, in Theory of Graphs and Its Applications (Proc. Sympos. Smolenice, 1963), Publ. House Czechoslovak Acad. Sci., Prague, 1964, pp. 47–52.

[11] R. Hartshorne, *Geometry: Euclid and Beyond,* Springer-Verlag, New York, 2000.

[12] E. Hlawka, J. Schoissengeier, and R. Taschner, *Geometric and Analytic Number Theory*, Springer-Verlag, Berlin, 1991.

[13] P. J. Kelly, *On Isometric Transformations*, Ph.D. thesis, University of Waterloo, Waterloo, Ontario, Canada, 1942.

[14] P. J. Kelly, *A congruence theorem for trees*, Pacific J. Math., 7 (1957), pp. 961–968.

[15] W. L. Kocay, *Some new methods in reconstruction theory*, in Combinatorial Mathematics IX (Brisbane, 1981), Springer, Berlin, 1982, pp. 89–114.

[16] W. L. Kocay, *A family of nonreconstructible hypergraphs*, J. Combin. Theory Ser. B, 42 (1987), pp. 46–63.

[17] I. Krasikov and Y. Roditty, *Geometrical reconstructions*, Ars Combin., 25 (1988), pp. 211–219.

[18] I. Krasikov and Y. Roditty, *On a reconstruction problem for sequences*, J. Combin. Theory Ser. A, 77 (1997), pp. 344–348.

[19] L. Lovász, *A note on the line reconstruction problem*, J. Combin. Theory Ser. B, 13 (1972), pp. 309–310.

[20] P. Maynard and J. Siemons, *On the reconstruction of linear codes*, J. Combin. Des., 6 (1998), pp. 285–291.

[21] P. Maynard and J. Siemons, *On the Reconstruction Index of Permutation Groups: Semiregular Groups*, Preprint, University of East Anglia, Norwich, UK, 2000.

[22] V. B. Mnukhin, *Reconstruction of the k-orbits of a permutation group*, Mat. Zametki, 42 (1987), pp. 863–872, 911.

[23] V. B. Mnukhin, *The k-orbit reconstruction and the orbit algebra*, Acta Appl. Math., 29 (1992), pp. 83–117.

[24] V. B. Mnukhin, *The reconstruction of oriented necklaces*, J. Combin. Inform. System Sci., 20 (1995), pp. 261–272.

[25] V. B. Mnukhin, *The k-orbit reconstruction for Abelian and Hamiltonian groups*, Acta Appl. Math., 52 (1998), pp. 149–162.

[26] V. Müller, *The edge reconstruction hypothesis is true for graphs with more than $n \cdot \log_2 n$ edges*, J. Combin. Theory Ser. B, 22 (1977), pp. 281–283.

[27] C. S. J. A. Nash-Williams, *The reconstruction problem*, in Selected Topics in Graph Theory, L. W. Beineke and R. J. Wilson, eds., Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London, New York, 1978, pp. 205–236.

[28] C. S. J. A. Nash-Williams, *Reconstruction of infinite graphs*, Discrete Math., 95 (1991), pp. 221–229.

[29] L. Pebody, *The reconstructibility of finite Abelian groups*, Combin. Probab. Comput., to appear.

[30] A. J. Radcliffe and A. D. Scott, *Reconstructing subsets of Z_n*, J. Combin. Theory Ser. A, 83 (1998), pp. 169–187.

[31] A. J. Radcliffe and A. D. Scott, *Reconstructing subsets of reals*, Electron. J. Combin., 6 (1999), Research Paper 20 (electronic).

[32] A. J. Radcliffe and A. D. Scott, *Reconstructing Subsets of Nonabelian Groups*, Preprint, University of Nebraska, Lincoln, NE, 2000.

[33] A. J. Radcliffe and A. D. Scott, *Reconstruction Under Group Action*, Preprint, University of Nebraska, Lincoln, NE, 2000.

[34] A. D. Scott, *Reconstructing sequences*, Discrete Math., 175 (1997), pp. 231–238.

[35] P. K. Stockmeyer, *The falsity of the reconstruction conjecture for tournaments*, J. Graph Theory, 1 (1977), pp. 19–25.

[36] P. K. Stockmeyer, *A census of nonreconstructible digraphs.* I. *Six related families*, J. Combin. Theory Ser. B, 31 (1981), pp. 232–239.

[37] S. M. Ulam, *A Collection of Mathematical Problems*, Interscience Tracts in Pure and Applied Mathematics 8, Interscience Publishers, New York, London, 1960.

SIAM J. Discrete Math.
Vol. 16, No. 2, pp. 276–287

RANDOM KRYLOV SPACES OVER FINITE FIELDS[*]

RICHARD P. BRENT[†], SHUHONG GAO[‡], AND ALAN G. B. LAUDER[†]

Abstract. Motivated by a connection with block iterative methods for solving linear systems over finite fields, we consider the probability that the Krylov space generated by a fixed linear mapping and a random set of elements in a vector space over a finite field equals the space itself. We obtain an exact formula for this probability and from it we derive good lower bounds that approach 1 exponentially fast as the size of the set increases.

Key words. finite field, vector space, linear transformation, Krylov subspace

AMS subject classifications. 11T99, 15A04, 15A33

PII. S089548010139388X

1. Introduction.

Let $\mathbb{F}_q$ denote the finite field with q elements and $\mathbb{F}_q[X]$ the ring of polynomials in one variable over $\mathbb{F}_q$. Let V be a vector space of dimension n over $\mathbb{F}_q$. Given a linear mapping T on V and a subset of vectors $S \subseteq V$ of size m, the Krylov subspace generated by S under T is defined as

$$\mathrm{Kry}(T,S) := \left\{ \sum_{i=1}^{m} f_i(T)v_i : \ f_i(X) \in \mathbb{F}_q[X] \text{ and } v_i \in S \text{ for } 1 \le i \le m \right\}.$$

This is just the space spanned by all vectors of the form $T^i v$ over all nonnegative powers of T and vectors $v \in S$. Define

$$\kappa_m(T) = \frac{1}{q^{mn}} \cdot \#\{(v_1,\ldots,v_m) \in V^m : \mathrm{Kry}(T,\{v_1,\ldots,v_m\}) = V\};$$

that is, $\kappa_m(T)$ is the density of m-tuples of vectors in V that generate the whole space V under T. In other words, if one selects m vectors $v_1,\ldots,v_m$ uniformly at random and independently from V, then $\kappa_m(T)$ is the probability that $\mathrm{Kry}(T,\{v_1,\ldots,v_m\}) = V$. Our main goal in this paper is to find good lower bounds on $\kappa_m(T)$.

To state our result, we need to define some parameter depending on T. Let ℓ be the minimal number of vectors required to generate V under T. This number ℓ is just the number of invariants in the Frobenius decomposition of V under T. We call ℓ the *Frobenius index* of T. Our main result is the following theorem.

Theorem. *Let T be a linear mapping on a vector space V of dimension n over $\mathbb{F}_q$. Suppose T has Frobenius index ℓ. Then for $m \ge \ell$*

$$\kappa_m(T) \ge \begin{cases} \frac{0.04}{1+\log_q(n-\ell+1)} & \text{if } m = \ell, \\ \frac{1}{8} & \text{if } m = \ell+1 \text{ and } q = 2, \\ 1 - \frac{3}{2^{m-\ell}} \ge \frac{1}{4} & \text{if } m \ge \ell+2 \text{ and } q = 2, \\ 1 - \frac{2}{q^{m-\ell}} \ge \frac{1}{3} & \text{if } m \ge \ell+1 \text{ and } q > 2. \end{cases}$$

[*] Received by the editors August 17, 2001; accepted for publication (in revised form) November 18, 2002; published electronically February 20, 2003.
http://www.siam.org/journals/sidma/16-2/39388.html

[†] Computing Laboratory, Oxford University, Oxford OX1 3QD, UK (richard.brent@comlab.ox.ac.uk, alan.lauder@comlab.ox.ac.uk). The research of the third author was supported by EPSRC grant GR/N35366/01 and by St. John's College, Oxford.

[‡] Department of Mathematical Sciences, Clemson University, Clemson, SC 29634-0975 (sgao@math.clemson.edu). The research of this author was supported in part by NSF grant DMS9970637, NSA grant MDA904-00-1-0048, ONR grant N00014-00-1-0565, and by the South Carolina Commission on Higher Education under a research initiative grant.

When $m = \ell$ the lower bound is almost tight in the sense that there are values of n such that the probability is arbitrarily close to zero; see the remark following Corollary 10. Hence it is impossible to bound the probability away from zero in this case. For fixed ℓ the probability converges exponentially fast to 1 as m increases.

There are two important special cases. One is when T is the identity map, so $\ell = n$. In this case, $\kappa_m(T)$ is equal to the probability that m random vectors in a vector space of dimension n over $\mathbb{F}_q$ span the whole space, and a much better lower bound can be proved (see Lemma 7). The other is when $\ell = 1$, which means that the minimal polynomial of T equals its characteristic polynomial, and better lower bounds are given in Theorem 9.

Our work was motivated by a connection with block iterative methods for solving large sparse linear systems over finite fields; see [3, 4, 8, 12, 14]. It improves upon the result in the report [15] used in an analysis of the block Wiedemann algorithm. We note that the relation between $\kappa_m(T)$ and the Frobenius index ℓ is studied in [15] (see also [16, section 6]), although the formulae obtained are less explicit and a somewhat different approach is taken. A more difficult and important question in the analysis of such algorithms is to bound the probability that certain "truncated" Krylov subspaces generate the whole space. More precisely, let

$$\mathrm{Kry}(T, S; t) = \left\{ \sum_{i=1}^{m} f(T)v_i : \, f_i(X) \in \mathbb{F}_q[X], \deg f_i \le t, \text{ and } v_i \in S \right\}.$$

For t approximately $n/|S|$, one requires a lower bound on the probability that the above space is the whole space. For large finite fields, relative to the dimension n, Kaltofen [8] and Villard [15, section 6] obtain such a bound using the Schwartz–Zippel lemma. For some practical applications, such as integer and polynomial factorization [5, 6, 9, 11], it is desirable to have a good bound for small fields. Using a counting argument Coppersmith obtains a weak bound in [4, 15]; it would be of great interest to strengthen this bound.

We use a module theoretic approach via a sequence of reductions using standard decomposition theorems and an argument from the theory of abelian groups communicated to us by Simon Blackburn. Using existing results on random elements in vector spaces over finite fields, we then obtain an exact formula (Theorem 5) for the probability depending only on certain properties of the mapping. Finally, good lower bounds for this expression are derived.

2. Reductions.

In this section we consider various reductions which allow us to characterize those sets of vectors which generate the whole space under T.

2.1. Module-theoretic interpretation.

Let T be a linear mapping on a vector space V of dimension n over $\mathbb{F}_q$. Denote by V_T the $\mathbb{F}_q[X]$-module with underlying abelian group V and action of $\mathbb{F}_q[X]$ on V defined as

$$f(X) \cdot v := f(T)v$$

for any polynomial $f \in \mathbb{F}_q[X]$. (Any element $v \in V$ may be thought of as lying in V_T, and vice versa. When necessary to distinguish them we shall call elements in V "vectors" and those in V_T "module elements.")

LEMMA 1. *For any set $S \subseteq V$ the Krylov space* $\mathrm{Kry}(T, S)$ *equals V if and only if S generates V_T as an $\mathbb{F}_q[X]$-module.*

Proof. Let S be such that the Krylov space generated by S under T is V. Let $w \in V$. Thus the vector w equals a linear combination over $\mathbb{F}_q$ of vectors of the form

$T^i v$, where $v \in V$. Hence the module element w is a linear combination over $\mathbb{F}_q$ of module elements of the form $X^i.v$ for $v \in S$. Thus S generates V_T as an $\mathbb{F}_q[X]$-module. The converse is similar. □

Thus our main question is equivalent to the following: Given a set of elements S chosen uniformly at random from the module V_T, what is the probability that they generate V_T?

2.2. Reduction to primary modules. Let the principal ideal (m_T) in $\mathbb{F}_q[X]$ be the annihilator of the module V_T, that is,

$$(m_T) = \{g \in \mathbb{F}_q[x] : g(T)v = 0 \text{ for all } v \in V\}.$$

(Thus m_T, which we take to be monic, is just the minimal polynomial of the linear mapping T.) Factorize m_T as

$$m_T = \prod_{i=1}^{a} g_i^{r_i},$$

where g_i are monic irreducible polynomials and each $r_i \geq 1$. Via the primary decomposition theorem [1, Theorem 3.7.12] the module V_T decomposes as

$$V_T = V_1 \oplus V_2 \oplus \cdots \oplus V_a, \tag{1}$$

where the annihilator of V_i is $(g_i^{r_i})$.

For each $1 \leq i \leq a$, let π_i denote the projection of V_T onto its ith factor. For a subset S of elements in V_T write $\pi_i(S)$ for the image of the set S under this projection.

LEMMA 2. *Let S be a set of elements in V_T. Then S generates V_T if and only if $\pi_i(S)$ generates V_i for $1 \leq i \leq a$.*

Proof. The forward implication is straightforward. For the reverse, assume that $\pi_i(S)$ generates V_i for $1 \leq i \leq a$. Let $v \in V_T$, so $\pi_i(v) \in V_i$. We can write $\pi_i(v) = \sum_{j=1}^{m} h_{ij}(X).v_j$, where $S = \{v_1, \ldots, v_m\}$. For each j, $1 \leq j \leq m$, using the Chinese remainder theorem we can find a polynomial $h_j(X)$ such that $h_j(X) \equiv h_{ij}(X) \bmod g_i(X)^{r_i}$ for each i, $1 \leq i \leq a$. Here we use the coprimality of the $g_i(X)$. Defining $w := \sum_{j=1}^{m} h_j(X).v_j$ we see that $\pi_i(w) = \pi_i(v)$ for all $0 \leq i \leq a$, and hence $v = w$. Thus S generates V_T as we wished to show. □

Thus it suffices to understand the number of generating sets of the primary modules V_i.

2.3. Reduction to irreducible exponent case. We now examine the primary parts V_i in the decomposition of the module V_T given in (1). To this end, let W denote any $\mathbb{F}_q[X]$-module with an annihilator the ideal generated by a power g^r of an irreducible polynomial g. We need to determine the probability that a set of randomly chosen elements in W generates the whole module.

Let $\mathrm{Rad}(W)$ denote the Radical of W. This is defined to be the intersection of all maximal submodules. The following result is a special case of a module-theoretic analogue of a result in the theory of abelian groups, namely, "a set of elements generates an abelian group if and only if its image in the quotient by the Frattini subgroup generates the quotient" (see [13, page 135, 5.2.12]).

LEMMA 3. *Let W be a primary $\mathbb{F}_q[X]$-module with annihilator (g^r), where g is irreducible in $\mathbb{F}_q[X]$. A set $S \subseteq W$ is a generating set if and only if $\bar{S} := \{s + \mathrm{Rad}(W) \mid s \in S\}$ is a generating set in the quotient module $W/\mathrm{Rad}(W)$.*

Proof. The forward implication is easy. For the reverse, by the cyclic decomposition theorem [1, Theorem 3.7.1] we can write

$$W = W_1 \oplus W_2 \oplus \cdots \oplus W_b,$$

where each module W_i is cyclic with annihilator the ideal generated by the polynomial g^{r_i} for some power of g. We may take $r_i \geq r_{i+1}$ for $1 \leq i \leq b-1$, and so $r_1 = r$. Since each module in the decomposition is cyclic we have the $\mathbb{F}_q[X]$-module isomorphism

$$W_i \cong \mathbb{F}_q[X]/(g^{r_i}),$$

and so

$$W \cong \oplus_{i=1}^{b} \mathbb{F}_q[X]/(g^{r_i}).$$

The intersection of all maximal submodules is just

$$\mathrm{Rad}(W) \cong \oplus_{i=1}^{b} g \cdot (\mathbb{F}_q[X]/(g^{r_i})),$$

which is just $g(X)W$. Hence

$$W/\mathrm{Rad}(W) \cong \mathbb{F}_q[X]/(g) \oplus \cdots \oplus \mathbb{F}_q[X]/(g),$$

where we have b terms in the sum. Now assume that the images of the elements of $S = \{v_i\}$ in the quotient generate $W/\mathrm{Rad}(W)$. Let $w \in W$. Via the isomorphisms described above we have $w = (w_1, \ldots, w_b)$, where each $w_i \in \mathbb{F}_q[X]/(g^{r_i})$. The image of w in the quotient $W/\mathrm{Rad}(W)$ is then $\bar{w} := (w_1 \bmod g, \ldots, w_b \bmod g)$. By assumption we can write $\bar{w} = \sum_{i=1}^{m} h_i(X).\bar{v}_i$. Then $w - \sum_{i=1}^{m} h_i(X).v_i = (gw'_1, \ldots, gw'_b)$. Defining $w' = (w'_1, \ldots, w'_b) \in W$ and repeating the process, we can express w as a combination of the elements v_i plus an "error vector" each coefficient of which is divisible by g^2. Continuing in this way the error vector eventually reduces to zero, since our module is annihilated by some power of g, and we have the desired combination. □

As in the proof of the above lemma, for W a primary module with annihilator (g^r) the required quotient is just

$$W/\mathrm{Rad}(W) \cong \mathbb{F}_q[X]/(g) \oplus \cdots \oplus \mathbb{F}_q[X]/(g),$$

where we have b terms in the sum. Letting $d = \deg(g)$ we see that this is just the direct sum of b finite fields of order q^d, each viewed as an $\mathbb{F}_q[X]$-module. The action of $\mathbb{F}_q[X]$ on each finite field is just defined for α in the finite field by $X.\alpha = \beta\alpha$, where β is some element such that $g(\beta) = 0$ in the finite field. We have

$$W/\mathrm{Rad}(W) \cong (\mathbb{F}_{q^d})^b$$

as an $\mathbb{F}_q[X]$-module. The right-hand side also has the structure of a vector space over $\mathbb{F}_{q^d}$. A set of elements in $W/\mathrm{Rad}(W)$ is a generating set if and only if the corresponding elements on the right-hand side of the above isomorphism generates the set $(\mathbb{F}_{q^d})^b$ as a $\mathbb{F}_{q^d}$-vector space. This follows from the description of the action of $\mathbb{F}_q[X]$ on each vector space in the summand, since $1, \beta, \ldots, \beta^{d-1}$ generates each finite field as a vector space over $\mathbb{F}_q$. Thus we have reduced our problem to the study of generating sets for vector spaces over finite fields.

2.4. Generating sets for vector spaces. For each nonnegative integer n, define the real function $\pi(n, x)$ by

$$\pi(n, x) := (1 - x)(1 - x^2) \dots (1 - x^n).$$

The following lemma is "classical."

LEMMA 4. *Let U be a vector space of dimension b over $\mathbb{F}_q$. Then the probability that $m \geq b$ elements of U chosen uniformly at random span U is*

$$\frac{\pi(m, 1/q)}{\pi(m - b, 1/q)}.$$

Proof. We follow the proof for the prime field case in [10], making appropriate modifications. (See also Theorem 1.1 in [2].) Let $\Phi_b(m, r)$ denote the number of m-tuples of vectors in $\mathbb{F}_q^b$ which span a subspace of rank r (equivalently, the number of rank r matrices of size $b \times m$ over $\mathbb{F}_q$). Dividing such sequences into those whose last vector is linearly dependent/independent on the previous $m - 1$ we derive the recurrence for $m \geq 1$ and $r \geq 1$

$$\Phi_b(m, r) = q^r \Phi_b(m - 1, r) + (q^b - q^{r-1})\Phi_b(m - 1, r - 1).$$

We also have the initial conditions $\Phi_b(s, 0) = 1$ for all $s \geq 1$ (the zero sequence), $\Phi_b(0, 0) = 1$ (the empty sequence), and $\Phi_b(0, s) = 0$ for all $s \geq 1$. One can now verify that the following formula holds for $r \geq 1$:

$$\Phi_b(m, r) = \prod_{i=0}^{r-1} (q^b - q^i) \frac{q^{m-i} - 1}{q^{i+1} - 1}.$$

Putting $r = b$ and cancelling in a suitable way one finds that

$$\Phi_b(m, b) = (q^m - 1)(q^m - q) \dots (q^m - q^{b-1}).$$

Dividing by the number of sequences, q^m, gives the required probability. □

3. An exact formula. We now piece together the results proved in section 2 to obtain an exact formula for the required probability. Let the minimal polynomial of the linear mapping T be denoted m_T and the characteristic polynomial c_T. Let ℓ be the Frobenius index of T. We consider a cyclic decomposition [1, Theorem 3.7.1] of the module V_T as

$$V_T = U_1 \oplus U_2 \oplus \cdots \oplus U_\ell,$$

where each U_i is a cyclic module with annihilator the ideal generated by a monic polynomial h_i satisfying $h_{i+1} | h_i$ for $1 \leq i \leq \ell - 1$. Thus $m_T = h_1$ and $c_T = h_1 h_2 \dots h_\ell$. As before, let g_j, $1 \leq j \leq a$, be the irreducible factors of m_T. Let d_j be the degree of g_j and ℓ_j the number of polynomials $h_1, \dots, h_l$ divisible by g_j, $1 \leq j \leq a$. Thus $1 \leq \ell_j \leq \ell$ and the cyclic decomposition of the module V_i in the primary decomposition of V_T (see (1)) has exactly ℓ_i factors.

THEOREM 5. *Let T be a linear mapping on a vector space V of dimension n over $\mathbb{F}_q$. Suppose T has Frobenius index ℓ and $m \geq \ell$. With the notation defined above, we have*

$$\kappa_m(T) = \prod_{j=1}^{a} \frac{\pi(m, q^{-d_j})}{\pi(m - \ell_j, q^{-d_j})},$$

where $\pi(m, x) = (1 - x)(1 - x^2) \dots (1 - x^m)$.

Proof. By Lemma 1 one may equivalently find the probability that a uniform at random sequence of elements S in V_T generates V_T as an $\mathbb{F}_q[X]$-module. By Lemma 2 such a set will generate V_T if and only if the set $\pi_j(S)$ generates each primary summand V_j for $1 \le j \le a$. Now for any choice of subsets $S_j \subseteq V_j$ of size m, $1 \le j \le a$, there exists exactly one set S in V_T such that $\pi_j(S) = S_j$ for each $1 \le j \le a$. Conversely, all sets S arise in this way. Thus it suffices to compute the probabilities of generating each primary module V_i by m elements separately and to take the product.

We claim that the jth term in the product in the statement of the theorem is the probability that a sequence of m elements chosen uniformly at random in V_j will generate V_j. Once this claim is proved the result follows. By Lemma 3 a set of elements S_j in V_j is a generating set if and only if its image in the quotient by the Radical of V_j generates this quotient. If S_j is chosen uniformly at random in V_j, the corresponding set of elements $\bar{S}_j$ in the quotient will be uniform at random. (Exactly $|\mathrm{Rad}(V_j)|$ elements of V_j map onto each element in the quotient.) Thus we need to find the probability that m elements chosen uniformly at random in the quotient generate it. But the quotient has the structure of a vector space of dimension ℓ_j over $\mathbb{F}_{q^{d_j}}$. From the comments at the end of section 2.3 this probability is equal to the probability that m elements chosen uniformly at random from a vector space of dimension ℓ_j over $\mathbb{F}_{q^{d_j}}$ span the space. The result now follows from Lemma 4. □

4. Lower bounds. The formula in Theorem 5 is elegant, but it is hard to see the magnitude of the probability $\kappa_m(T)$. In this section we shall derive various simple explicit lower bounds for $\kappa_m(T)$.

We shall repeatedly use the following equality and inequality:

$$\frac{1}{q^k} + \frac{1}{q^{k+1}} + \cdots + \frac{1}{q^m} + \cdots = \frac{1}{q^{k-1}(q-1)},$$
$$(1-x_1)^{a_1}(1-x_2)^{a_2}\cdots(1-x_m)^{a_m} \ge 1 - (a_1x_1 + a_2x_2 + \cdots + a_mx_m)$$

for any real $a_i \ge 1$, $1 \ge x_i \ge 0$, $q > 1$, and any integer $k \ge 0$. The inequality can be seen as follows. First of all it holds if $x_i \ge 1/a_i$ for some i. So we may assume that $0 \le x_i < 1/a_i$ for all i. Then one sees that the inequality follows by induction from the following two inequalities:

$$(1-x_1)(1-x_2) \ge 1 - (x_1 + x_2) \quad \text{for } x_1x_2 \ge 0,$$
$$(1-x)^a \ge 1 - ax \quad \text{for } 0 \le x < \frac{1}{a},\ a \ge 1.$$

The latter inequality here holds since the function $a\ln(1-x) - \ln(1-ax)$ strictly increases for $0 \le x < 1/a$ (for any fixed $a > 1$) and evaluates to 0 when $x = 0$.

The next lemma is an extremely crude estimation, but it is already useful for large q.

Lemma 6. *Let T be any linear map on a vector space of dimension n over $\mathbb{F}_q$. Let ℓ be the Frobenius index of T. Then, for $m \ge \ell$,*

$$\kappa_m(T) \ge 1 - \frac{n}{q-1}.$$

Proof. With the notation in Theorem 5, as $n \geq a$, $m \geq \ell_j$, and $d_j \geq 1$, we have

$$\begin{aligned}
\kappa_m(T) &= \prod_{j=1}^{a}\prod_{i=1}^{\ell_j}\left(1-\left(\frac{1}{q^{d_j}}\right)^{m-\ell_j+i}\right) \\
&\geq \prod_{j=1}^{n}\prod_{i=1}^{\infty}\left(1-\left(\frac{1}{q}\right)^{i}\right) \\
&\geq \left(1-\sum_{i=1}^{\infty}\frac{1}{q^i}\right)^n \geq \left(1-\frac{1}{q-1}\right)^n \geq 1-\frac{n}{q-1}. \qquad \square
\end{aligned}$$

The bound in Lemma 6 is good if q is large, but it says nothing if $q \leq n+1$. To get a good lower bound of $\kappa_m(T)$ for small q, we need a more careful estimation. We start with a simple case when T is the identity map on V.

LEMMA 7. *Let V be a vector space of dimension n over $\mathbb{F}_q$ and let $m \geq n$. Then the probability that m random vectors in V span the whole space V is*

$$\prod_{i=1}^{n}\left(1-\frac{1}{q^{m-n+i}}\right) \geq \begin{cases} 0.288, & \textit{if } m=n \textit{ and } q=2, \\ 1-\frac{1}{q^{m-n}(q-1)} & \textit{otherwise.}\end{cases}$$

Equivalently, this also bounds the probability that a random $m \times n$ matrix over $\mathbb{F}_q$ has rank n.

Proof. By Lemma 4, the probability is

$$\begin{aligned}
\frac{\pi(m,1/q)}{\pi(m-n,1/q)} &= \left(1-\frac{1}{q^{m-n+1}}\right)\left(1-\frac{1}{q^{m-n+2}}\right)\cdots\left(1-\frac{1}{q^m}\right) \\
&\geq 1-\left(\frac{1}{q^{m-n+1}}+\frac{1}{q^{m-n+2}}+\cdots+\frac{1}{q^m}\right) \\
&\geq 1-\frac{1}{q^{m-n+1}}\left(1+\frac{1}{q}+\cdots+\frac{1}{q^{n-1}}+\cdots\right) \\
&\geq 1-\frac{1}{q^{m-n+1}}\frac{1}{1-1/q} \geq 1-\frac{1}{q^{m-n}(q-1)}.
\end{aligned}$$

For $m=n$ and $q=2$, the above bound is zero, so we need a more careful analysis:

$$\begin{aligned}
&\left(1-\frac{1}{2}\right)\left(1-\frac{1}{2^2}\right)\cdots\left(1-\frac{1}{2^m}\right) \\
&> \left(1-\frac{1}{2}\right)\left(1-\frac{1}{2^2}\right)\left(1-\frac{1}{2^3}\right)\left(1-\frac{1}{2^4}\right)\left(1-\frac{1}{2^5}\right)\cdots\left(1-\frac{1}{2^m}\right)\cdots \\
&> \left(1-\frac{1}{2}\right)\left(1-\frac{1}{2^2}\right)\left(1-\frac{1}{2^3}\right)\left(1-\frac{1}{2^4}\right)\left(1-\left(\frac{1}{2^5}+\cdots+\frac{1}{2^m}+\cdots\right)\right) \\
&= \left(1-\frac{1}{2}\right)\left(1-\frac{1}{2^2}\right)\left(1-\frac{1}{2^3}\right)\left(1-\frac{1}{2^4}\right)\left(1-\frac{1}{2^4}\right) \\
&> 0.288.
\end{aligned}$$

This completes the proof. $\square$

To deal with the general case we need the following result, which reduces the problem for a general polynomial to that of a polynomial with irreducible factors of small degrees only.

LEMMA 8. *For $k \geq 1$, let I_k be the number of irreducible polynomials in $\mathbb{F}_q[X]$ of degree k. Let $f \in \mathbb{F}_q[X]$ of degree n and let $u = \lfloor \log_q n \rfloor$. Then for any integer $q_1 > 1$*

$$\prod_{g|f,\, g\ irred} \left(1 - \frac{1}{q_1^{\deg g}}\right) \geq \prod_{k=1}^{u+1} \left(1 - \frac{1}{q_1^k}\right)^{I_k}.$$

Proof. This result is proved in [7] (i.e., the formula (6) on page 144, with q replaced by q_1). □

We consider the important case when V is cyclic as an $\mathbb{F}_q[X]$-module under T; hence $\ell = 1$ and $\ell_j = 1$ in Theorem 5. In this case, the minimal polynomial of T is equal to its characteristic polynomial, and T is called *nonderogatory.*

THEOREM 9. *Let T be a nonderogatory linear map on a vector space V of dimension n over $\mathbb{F}_q$. Then*

$$\kappa_m(T) \geq \begin{cases} \frac{0.218}{1+\log_q n} & \text{if } m = 1, \\ 0.42 & \text{if } m = 2 \text{ and } q = 2, \\ 1 - \frac{1.5}{q^{m-1}} \geq \frac{1}{2} & \text{otherwise.} \end{cases}$$

Proof. Let f be the minimal polynomial of T. Then f has degree n and all $\ell_i = 1$ in Theorem 5. Hence

$$\kappa_m(T) = \prod_{g|f,\, g \text{ irred}} \left(1 - \frac{1}{q^{m \deg g}}\right).$$

First assume $m = 1$. Then $\kappa_1(T)$ is the density of polynomials in $\mathbb{F}_q[X]$ of degrees $< n$ that are relatively prime to f. In this case, by Theorem 2.1 in [7], we have

$$\kappa_1(T) \geq \left(1 - \frac{1}{q}\right) \cdot \frac{1}{e^{0.83}(1 + \log_q n)} > \frac{0.218}{1 + \log_q n},$$

where the factor $1 - 1/q$ accounts for the irreducible factor X that is excluded in [7].

Now assume $m > 1$. Let $u = \lfloor \log_q n \rfloor$ and I_k as in Lemma 8. Note that $I_1 = q$ and

$$I_k \leq \frac{q^k - 1}{k} \leq \frac{q^k}{2}, \quad k \geq 2.$$

By Lemma 8, we have

$$\begin{aligned} \kappa_m(T) &\geq \prod_{k=1}^{u+1} \left(1 - \frac{1}{q^{mk}}\right)^{I_k} \\ &\geq \left(1 - \frac{1}{q^m}\right)^q \prod_{k=2}^{\infty} \left(1 - \frac{1}{q^{mk}}\right)^{\frac{q^k-1}{k}} \\ &\geq \left(1 - \frac{1}{q^m}\right)^q \left(1 - \sum_{k=2}^{\infty} \frac{q^k - 1}{kq^{mk}}\right) \\ &\geq \left(1 - \frac{1}{q^m}\right)^q \left(1 - \sum_{k=2}^{\infty} \frac{1}{2q^{(m-1)k}}\right) \\ &\geq \left(1 - \frac{1}{q^m}\right)^q \left(1 - \frac{1}{2q^{m-1}(q^{m-1} - 1)}\right), \end{aligned}$$

which is at least 0.42 when $m = 2$ and $q = 2$ and generally at least

$$\begin{aligned}\left(1-\frac{1}{q^{m-1}}\right)\left(1-\frac{1}{2q^{m-1}(q^{m-1}-1)}\right) &> 1-\frac{1}{q^{m-1}}-\frac{1}{2q^{m-1}(q^{m-1}-1)}\\ &\geq 1-\frac{1.5}{q^{m-1}}\end{aligned}$$

for all q and m. □

Theorem 9 can be interpreted for the following situation. Let $f \in \mathbb{F}_q[X]$ be any polynomial of degree n. Define $\kappa_m(f)$ to be the probability that

$$\gcd(f, g_1, \ldots, , g_m) = 1$$

for m random polynomials $g_1, \ldots, g_m \in \mathbb{F}_q[x]$ of degrees $< n$. Note that $\kappa_1(f)$ is the Euler function for the polynomial f. Then for any nonderogatory linear map T on a vector space of dimension n over $\mathbb{F}_q$ that has f as its minimal polynomial, we have

$$\kappa_m(f) = \kappa_m(T) = \prod_{g|f,\, g \text{ irred}} \left(1 - \frac{1}{q^{m \deg g}}\right).$$

Hence the lower bounds in Theorem 9 apply to $\kappa_m(f)$ automatically.

Corollary 10. *Let $f \in \mathbb{F}_q[x]$ of degree n. Then*

$$\kappa_m(f) \geq \begin{cases} \frac{0.218}{1+\log_q n} & \text{if } m = 1, \\ 0.42 & \text{if } m = 2 \text{ and } q = 2, \\ 1 - \frac{1.5}{q^{m-1}} \geq \frac{1}{2} & \text{otherwise.} \end{cases}$$

Remark. By Theorem 3.4 in [7], there are infinitely many values of n such that

$$\kappa_1(x^n - 1) \leq \frac{c}{\sqrt{1 + \log_q n}}$$

for some constant $c > 0$ depending only on q. This means that the probability may be arbitrarily close to zero and our lower bound is quite close to the upper bound. This also applies to the lower bound in Theorem 11 below for $m = \ell$.

Now we turn to the general case where we obtain slightly weaker bounds. The next result is the main theorem stated in the introduction.

Theorem 11. *Let T be any linear map on a vector space of dimension n over $\mathbb{F}_q$. Let ℓ be the Frobenius index of T and let $m \geq \ell$. Then*

$$\kappa_m(T) \geq \begin{cases} \frac{0.04}{1+\log_q(n-\ell+1)} & \text{if } m = \ell, \\ \frac{1}{8} & \text{if } m = \ell + 1 \text{ and } q = 2, \\ 1 - \frac{3}{2^{m-\ell}} \geq \frac{1}{4} & \text{if } m \geq \ell + 2 \text{ and } q = 2, \\ 1 - \frac{2}{q^{m-\ell}} \geq \frac{1}{3} & \text{if } m \geq \ell + 1 \text{ and } q > 2. \end{cases}$$

Proof. Let f be the minimal polynomial of T. Then $\deg f \leq n - \ell + 1$ as at least one irreducible factor of f appears ℓ times in the characteristic polynomial of T, which has degree n and is divisible by f. Let $u = \lfloor \log_q(n - \ell + 1) \rfloor$. By Theorem 5 and Lemma 8, we have

$$\begin{aligned}
\kappa_m(T) &= \prod_{j=1}^{a}\prod_{i=1}^{\ell_i}\left(1-\left(\frac{1}{q^{d_j}}\right)^{m-\ell_i+i}\right) \\
&\geq \prod_{j=1}^{a}\prod_{i=1}^{\ell}\left(1-\left(\frac{1}{q^{d_j}}\right)^{m-\ell+i}\right) \\
&= \prod_{i=1}^{\ell}\prod_{g|f,\, g\,\mathrm{irred}}\left(1-\left(\frac{1}{q^{\deg g}}\right)^{m-\ell+i}\right) \\
&\geq \prod_{i=1}^{\ell}\prod_{k=1}^{u+1}\left(1-\left(\frac{1}{q^{k}}\right)^{m-\ell+i}\right)^{I_k}.
\end{aligned}\tag{2}$$

Assume first that $m=\ell$. Then

$$\begin{aligned}
\kappa_m(T) &\geq \prod_{i=1}^{\ell}\prod_{k=1}^{u+1}\left(1-\left(\frac{1}{q^{k}}\right)^{i}\right)^{I_k} \\
&\geq \prod_{i=1}^{\ell}\left(1-\frac{1}{q^i}\right)\prod_{i=1}^{\ell}\prod_{k=1}^{u+1}\left(1-\frac{1}{q^{ki}}\right)^{\frac{q^k-1}{k}} \\
&\geq \prod_{i=1}^{\ell}\left(1-\frac{1}{q^i}\right)\prod_{k=1}^{u+1}\left(1-\frac{1}{q^{k}}\right)^{\frac{q^k-1}{k}}\prod_{k=1}^{\infty}\prod_{i=2}^{\infty}\left(1-\frac{1}{q^{ki}}\right)^{\frac{q^k-1}{k}}.
\end{aligned}$$

By Lemma 7, we know the first product is at least 0.288. For the second product, the proof of Theorem 2.1 in [7] implies

$$\prod_{k=1}^{u+1}\left(1-\frac{1}{q^k}\right)^{\frac{q^k-1}{k}} \geq \frac{1}{e^{0.83}(1+u)} \geq \frac{1}{e^{0.83}(1+\log_q(n-\ell+1))}.$$

To estimate the third product, we recall the fact that

$$\ln(1-x) \geq -(x+x^2), \quad 0 \leq x \leq 0.6.$$

Then

$$\begin{aligned}
\prod_{k=1}^{\infty}\prod_{i=2}^{\infty}\left(1-\frac{1}{q^{ki}}\right)^{\frac{q^k-1}{k}} &= \exp\left(\sum_{k=1}^{\infty}\sum_{i=2}^{\infty}\frac{q^k-1}{k}\ln\left(1-\frac{1}{q^{ki}}\right)\right) \\
&\geq \exp\left(-\sum_{k=1}^{\infty}\sum_{i=2}^{\infty}\frac{q^k-1}{k}\left(\frac{1}{q^{ki}}+\frac{1}{q^{2ki}}\right)\right) \\
&\geq \exp\left(-\sum_{k=1}^{\infty}\frac{q^k-1}{k}\left(\frac{1}{q^k(q^k-1)}+\frac{1}{q^{2k}(q^{2k}-1)}\right)\right) \\
&\geq \exp\left(-\sum_{k=1}^{\infty}\left(\frac{1}{q^k}+\frac{1}{q^{3k}}\right)\right) \\
&\geq \exp\left(-\left(\frac{1}{q-1}+\frac{1}{q^3-1}\right)\right) \\
&\geq \exp\left(-\left(1+\frac{1}{7}\right)\right) > 0.3189.
\end{aligned}$$

Therefore, when $m = \ell$,

$$\kappa_m(T) > \frac{0.288 \cdot 0.3189}{e^{0.83}} \cdot \frac{1}{1+\log_q(n-\ell+1)} > \frac{0.04}{1+\log_q(n-\ell+1)}.$$

Finally assume $m > \ell$. Then from (2)

$$\kappa_m(T) \geq \prod_{i=1}^{\infty}\left(1-\frac{1}{q^{m-\ell+i}}\right)^{q} \prod_{k=2}^{\infty}\prod_{i=1}^{\infty}\left(1-\frac{1}{q^{k(m-\ell+i)}}\right)^{\frac{q^k-1}{k}}.$$

For the first product, we have

$$\begin{aligned}\prod_{i=1}^{\infty}\left(1-\frac{1}{q^{m-\ell+i}}\right)^{q} &\geq \left(1-\frac{q}{q^{m-\ell+1}}\right)\left(1-\sum_{i=2}^{\infty}\frac{q}{q^{m-\ell+i}}\right)\\ &\geq \left(1-\frac{1}{q^{m-\ell}}\right)\left(1-\frac{1}{q^{m-\ell}(q-1)}\right),\end{aligned}$$

which is 1/4 for $m = \ell+1$ and $q = 2$. For the second product, we have

$$\begin{aligned}\prod_{k=2}^{\infty}\prod_{i=1}^{\infty}\left(1-\frac{1}{q^{k(m-\ell+i)}}\right)^{\frac{q^k-1}{k}} &\geq 1-\sum_{k=2}^{\infty}\sum_{i=1}^{\infty}\frac{q^k-1}{kq^{k(m-\ell+i)}}\\ &\geq 1-\sum_{k=2}^{\infty}\sum_{i=1}^{\infty}\frac{1}{kq^{k(m-\ell+i-1)}}\\ &\geq 1-\sum_{k=2}^{\infty}\frac{1}{kq^{k(m-\ell-1)}(q^k-1)}\\ &\geq 1-\sum_{k=2}^{\infty}\frac{1}{q^{k(m-\ell)}}\\ &\geq 1-\frac{1}{q^{m-\ell}(q^{m-\ell}-1)},\end{aligned}$$

which is 1/2 for $m = \ell+1$ and $q = 2$. Therefore $\kappa_m(T)$ is at least $\frac{1}{4}\cdot\frac{1}{2} = \frac{1}{8}$ for $m = \ell+1$ and $q = 2$. In general, when $m > \ell$, it is at least

$$\begin{aligned}&\left(1-\frac{1}{q^{m-\ell}}\right)\left(1-\frac{1}{q^{m-\ell}(q-1)}\right)\left(1-\frac{1}{q^{m-\ell}(q^{m-\ell}-1)}\right)\\ &\geq 1-\frac{1}{q^{m-\ell}}-\frac{1}{q^{m-\ell}(q-1)}-\frac{1}{q^{m-\ell}(q^{m-\ell}-1)}\\ &\geq 1-\frac{q+1}{q-1}\frac{1}{q^{m-\ell}} \geq 1-\frac{3}{q^{m-\ell}}.\end{aligned}$$

For $q = 2$ and $m \geq \ell+2$ this is $1-\frac{3}{2^{m-\ell}} \geq \frac{1}{4}$, and for $q \geq 3$ and $m \geq \ell+1$ it is at least $1-\frac{2}{q^{m-1}} \geq \frac{1}{3}$. □

Acknowledgments. The second author is grateful to Oxford University Computing Laboratory for their warm hospitality during his visit in July 2001. The third author thanks Simon Blackburn for answering a question on group theory.

REFERENCES

[1] W. A. Adkins and S. H. Weintraub, *Algebra: An approach via module theory*, Grad. Texts in Math. 136, Springer-Verlag, New York, 1992.

[2] R. P. Brent and B. D. McKay, *Determinants and ranks of random matrices over $\mathbb{Z}_m$,* Discrete Math., 66 (1987), pp. 35–49.

[3] D. Coppersmith, *Solving linear equations over $GF(2)$: Block Lanczos algorithm,* Linear Algebra Appl., 192 (1993), pp. 33–60.

[4] D. Coppersmith, *Solving homogeneous linear equations over $GF(2)$ via block Wiedemann algorithm,* Math. Comp., 62 (1994), pp. 333–350.

[5] S. Gao, *Factoring multivariate polynomials via partial differential equations,* Math. Comp., to appear.

[6] S. Gao and J. von zur Gathen, *Berlekamp's and Niederreiter's polynomial factorization algorithms,* in Proceedings of the 2nd International Conference on Finite Fields: Theory, Applications, and Algorithms, Las Vegas, 1993, Contemp. Math. 168, AMS, Providence, RI, 1994, pp. 101–116.

[7] S. Gao and D. Panario, *Density of normal elements in finite fields,* Finite Fields Appl., 3 (1997), pp. 141–150.

[8] E. Kaltofen, *Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems,* Math. Comp., 64 (1995), pp. 777–806.

[9] E. Kaltofen and A. Lobo, *Factoring high-degree polynomials by the black box Berlekamp algorithm,* in Proceedings of the International Symposium on Symbolic and Algebraic Computation, Oxford, UK, 1994, pp. 90–98.

[10] G. Landsberg, *Über eine Anzahlbestimmung und eine damit zusammenhängende Reihe,* J. Reine Angew. Math., 111 (1893), pp. 87–88.

[11] A. K. Lenstra and H. W. Lenstra, Jr., eds., *The Development of the Number Field Sieve,* Lecture Notes in Math. 1554, Springer-Verlag, New York, 1993.

[12] P. L. Montgomery, *A block Lanczos algorithm for finding dependencies over $GF(2)$,* in Advances in Cryptology—EUROCRYPT '95, Saint-Malo, France, 1995, Lecture Notes in Comput. Sci. 921, Springer-Verlag, Berlin, 1995, pp. 106–120.

[13] D. J. S. Robinson, *A Course in the Theory of Groups,* 2nd ed., Graduate Texts in Math. 80, Springer-Verlag, New York, 1996.

[14] G. Villard, *Further analysis of Coppersmith's block Wiedemann algorithm for the solution of sparse linear systems,* in Proceedings of the International Symposium on Symbolic and Algebraic Computation, Maui, HI, 1997, ACM, New York, 1997, pp. 32–39.

[15] G. Villard, *A Study of Coppersmith's Block Wiedemann Algorithm Using Matrix Polynomials,* Technical report 975 IM, LMC-IMAG, Grenoble, France, 1997.

[16] D. H. Wiedemann, *Solving sparse linear equations over finite fields,* IEEE Trans. Inform. Theory, 32 (1986), pp. 54–62.

SIAM J. DISCRETE MATH.
Vol. 16, No. 2, pp. 288–300

MAKESPAN MINIMIZATION IN JOB SHOPS: A LINEAR TIME APPROXIMATION SCHEME[*]

KLAUS JANSEN[†], ROBERTO SOLIS-OBA[‡], AND MAXIM SVIRIDENKO[§]

Abstract. In this paper we present a linear time approximation scheme for the job shop scheduling problem with a fixed number of machines and fixed number of operations per job. This improves on the previously best $2+\epsilon$, $\epsilon > 0$, approximation algorithm for the problem by Shmoys, Stein, and Wein [*SIAM J. Comput.*, 23 (1994), pp. 617–632]. Our approximation scheme is very general and it can be extended to the case of job shop scheduling problems with release and delivery times, multistage job shops, dag job shops, and preemptive variants of most of these problems.

Key words. approximation algorithm, approximation scheme, job shop, scheduling

AMS subject classifications. 68W25, 68W40, 68R05

PII. S0895480199363908

1. Introduction.

In the job shop scheduling problem there is a set $J = \{J_1, \ldots, J_n\}$ of n jobs that must be processed on a given set $M = \{M_1, \ldots, M_m\}$ of m machines. Each job J_j consists of a sequence of μ_j operations $O_{1j}, \ldots, O_{\mu_j j}$ that need to be processed in this order. Operation O_{ij} must be processed without interruption on machine $M_{\pi_{ij}}$ during p_{ij} time units. A machine can process at most one operation at a time, and each job may be processed by at most one machine at any time. For a given schedule of J, let C_{ij} be the completion time of operation O_{ij}. The objective is to find a schedule for J that minimizes the maximum completion time, $C_{max} = \max_{ij} C_{ij}$. The value of C_{max} is also called the *makespan* or the *length* of the schedule. Let $\mu = \max_j \mu_j$ be the maximum number of operations in any job.

The job shop scheduling problem is considered to be one of the most difficult problems in combinatorial optimization, from both the theoretical and the practical points of view. Even very constrained versions of the problem are strongly NP-hard. (See, e.g., the survey paper by Lawler et al. [6].) Two other widely studied shop scheduling problems are the flow shop and the open shop problems. In the flow shop problem every job has exactly one operation per machine, and the order of execution for the operations is the same for all jobs. In the open shop problem every job has also one operation per machine, but there is no specified order for the execution of the operations of a job. Williamson et al. [16] proved that for any $\rho < 5/4$ the existence of a ρ-approximation algorithm for any of the above shop scheduling problems when the number of machines is part of the input would imply that P = NP. This result

[*]Received by the editors October 29, 1999; accepted for publication (in revised form) July 22, 2002; published electronically February 20, 2003. Preliminary versions of this paper appeared as *Makespan minimization in job shops: A polynomial time approximation scheme*, in Proceedings of the 31st Annual ACM Symposium on Theory of Computing, Atlanta, GA, 1999, pp. 394–399 and *A linear time approximation scheme for the job shop scheduling problem*, in Proceedings of the Second Workshop on Approximation Algorithms, Berkeley, CA, 1999, Lecture Notes in Comput. Sci. 1671, Springer, Berlin, 1999, pp. 177–188.

http://www.siam.org/journals/sidma/16-2/36390.html

[†]Institut für Informatik und Praktische Mathematik, Universität zu Kiel, Kiel, Germany (kj@informatik.uni-kiel.de). This author's research was supported in part by the Swiss Office Fédéral de l'éducation et de la Science project 97.0315, Platform.

[‡]Department of Computer Science, The University of Western Ontario, London, ON, Canada N6A 5B7 (solis@csd.uwo.ca). This author's research was supported in part by EU ESPRIT LTR project 20244 (ALCOM-IT) and NSERC grant R3050A01.

[§]IBM T.J. Watson, Yorktown Heights, NY (sviri@us.ibm.com).

holds even if every operation has integer processing time, and for the case of the job shop problem, even if each job has at most three operations.

Many papers about shop scheduling problems have recently been written. Several of them are based on the seminal work by Leighton, Maggs, and Rao [7] on the acyclic job shop problem with unit length operations. In this problem every job has exactly one operation per machine. Their main result was to show that this problem always has a solution of length $O(P_{max} + l_{max})$, where l_{max} is the maximum job length and P_{max} is the maximum machine load. This is not an algorithmic result, since it relies on a nonconstructive probabilistic argument. (For a constructive version, see [8].)

Shmoys, Stein, and Wein [15] described an approximation algorithm for the job shop scheduling problem with $O(\log^2(m\mu))$ performance guarantee. This algorithm was later improved by Goldberg et al. [1], who designed an approximation algorithm with a performance guarantee of $O(\log^2(m\mu)/(\log\log(m\mu))^2)$. When the number of machines m and the maximum number μ of operations per job are constant, Shmoys, Stein, and Wein [15] designed an approximation algorithm with performance guarantee $(2+\varepsilon)$ for any fixed value $\varepsilon > 0$. Following the three-field notation scheme [6], we denote this problem as $Jm|op \leq \mu|C_{max}$.

There are few theoretical results known for the preemptive version of the job shop scheduling problem. It is known that this problem is strongly NP-hard even in the case when there are three machines and every job has at most three operations. (See survey paper [6].) On the positive side, Sevastianov and Woeginger [12] designed a 3/2-approximation algorithm for the problem when the number of machines is 2.

A polynomial time approximation scheme (PTAS) for a (minimization) optimization problem is an algorithm that given any constant value $\varepsilon > 0$ finds in polynomial time a solution of value no larger than $1 + \varepsilon$ times the value of an optimum solution. A fully PTAS is an approximation scheme that runs in time polynomial in the size of the input and $1/\varepsilon$.

When the number m of machines is fixed, there exist PTASs for the flow shop [3] and the open shop [13] problems. However, the $2 + \varepsilon$ approximation algorithm of Shmoys, Stein, and Wein [15] was the previously best known algorithm for the job shop problem with m and μ fixed.

In this work we describe a linear time approximation scheme for the job shop scheduling problem when m and μ are fixed. Our work is strongly based on ideas contained in some of the aforementioned papers. We use the idea by Sevastianov and Woeginger [13] of partitioning the set of jobs into three sets: big, small, and tiny jobs. The sets of big and small jobs have constant size. We construct all possible schedules for the big jobs, and, since the number of big jobs is constant, the total number of their schedules is also constant. In any schedule for the big jobs, the starting and completion times of the jobs define a set of time intervals into which we have to schedule the small and tiny jobs. We use linear programming to find a "compact" assignment of small and tiny jobs to these time intervals. Then we show that it is possible to reduce the number of jobs that receive fractional assignments to a constant. Since only small and tiny jobs receive fractional assignments we can use a very simple rounding procedure for them to get a nonpreemptive schedule without increasing the length of the solution by too much. This solution is not feasible, though, since in each interval there might be conflicts among the small and tiny jobs.

We find a feasible schedule for the small and tiny jobs in each time interval by using an algorithm by Sevastianov [10]. (For a detailed presentation, in English, of the algorithm, see [14].) Sevastianov's algorithm runs in $O((\mu mn)^2)$ time and for any instance of the job shop scheduling problem it finds a schedule of length at most $P_{max} + \varphi(m,\mu)p_{max}$, where $\varphi(m,\mu) = (m\mu^2 + 2\mu - 1)(\mu - 1) = O(m\mu^3)$. (See also [11] for survey and historical overview of geometric methods used in the design and analysis of approximation algorithms with absolute performance guarantee for scheduling problems.) By selecting properly the sets of big, small, and tiny jobs we can prove that the total length of the schedule computed by the algorithm is at most $1 + \epsilon$ times the length of an optimum solution.

All steps of the algorithm can be performed in linear time, except two of them: solving the linear program and running Sevastianov's algorithm. Since we do not solve exactly the job shop scheduling problem, we do not need to solve exactly the linear program; an approximate solution would suffice. We use an algorithm of Grigoriadis and Khachiyan [2] to find in linear time a $1 + \varepsilon$ approximation to the solution of the linear program. Then we merge certain subsets of jobs together to form larger jobs to decrease the running time of Sevastianov's algorithm to $O(n)$. The overall complexity of our algorithm is linear in the number of jobs, but it is not polynomial in $1/\varepsilon$. This is not surprising, since the problem is strongly NP-hard [6] and therefore no fully polynomial time approximation scheme for the problem can exist unless P = NP.

Our approach can be used to design linear time approximation schemes for more general problems like the so-called dag shop problem [10, 11, 15], in which only a partial order is specified for the ordering of execution of the operations of a job. This problem includes as a special case the open shop problem. Since the flow shop problem is a special case of the job shop problem, our result generalizes the results of Hall [3] and Sevastianov and Woeginger [13] in the sense that we prove the existence of a PTAS for these two latter problems.

Our approximation scheme can be generalized also to the following problems when m and μ are fixed: multistage job shop, dag job shop, and job shop problems with release and delivery times. It is also possible to modify the schemes to design approximation algorithms for the preemptive versions of these problems, except for the preemptive dag shop problem.

The rest of the paper is organized in the following way. In section 2 we describe a polynomial time approximation scheme for the nonpreemptive job shop scheduling problem. Then in section 3 we show how to reduce the time complexity of the algorithm to $O(n)$. In section 4 we design a linear time approximation scheme for the preemptive version of the problem. Finally, in section 5 we show how to handle other shop scheduling problems.

2. PTAS for the job shop scheduling problem.

For a given instance of the job shop scheduling problem, the value of the optimum makespan is denoted as C^*_{max}. Let $P_t = \sum_{\pi_{ij}=t} p_{ij}$ be the total processing time of operations assigned to machine M_t. We call P_t the *load* of machine M_t. Let $P_{max} = \max\{P_1, \ldots, P_m\}$ be the maximum machine load. Clearly, $P_{max} \le C^*_{max}$. Let $l_j = \sum_{i=1}^{\mu_j} p_{ij}$ be the length of job J_j. We define $p_{max} = \max_{ij} p_{ij}$ to be the maximum operation length.

2.1. Restricted job shop problem.

Let $\varepsilon > 0$ be a constant value. Let $m \ge 2$ and $\mu \ge 1$ be the number of machines and maximum number of operations per job, respectively. We assume that the values of ε, μ, and m are fixed and not part of the input. We partition the set of jobs into three subsets as follows. Let α be a real number such that

$$\varepsilon^{\lceil m/\varepsilon \rceil} \le \alpha \le \varepsilon.$$

We define three sets of jobs:

$$B = \{J_j \mid l_j \geq \alpha P_{max}\},$$

$$S = \{J_j \mid \alpha\varepsilon P_{max} < l_j < \alpha P_{max}\}, \text{ and}$$

$$T = \{J_j \mid l_j \leq \alpha\varepsilon P_{max}\}.$$

The jobs in B are called *big jobs*, the jobs in S are called *small jobs*, and the jobs in T are called *tiny jobs*. For the operations of big, small, and tiny jobs we use a similar notation: the operations of big jobs are called *big operations*, while operations of small and tiny jobs are called *small* and *tiny operations*, respectively, independently of their actual sizes. The number of big jobs is at most $mP_{max}/(\alpha P_{max})$, and thus the size of B is bounded by a constant depending only on ε and m:

$$|B| \leq m/\alpha \leq m\varepsilon^{-\lceil m/\varepsilon \rceil}.$$

Sevastianov and Woeginger [13] show that the number α can be chosen so that

$$\sum_{J_j \in S} l_j \leq \varepsilon P_{max}. \tag{1}$$

This is done as follows. Define a sequence of numbers α_i, where i is a nonnegative integer, by $\alpha_i = \varepsilon^i$ and consider the sets S_i of small jobs with respect to α_i. Note that two sets S_i and S_j are disjoint for $i \neq j$. The total length of all jobs is at most mP_{max}, and so there exists a value $k \leq m/\varepsilon$ for which $S = S_k$ satisfies inequality (1). We set $\alpha = \alpha_k$.

Since the total length of the small jobs is at most εC^*_{max}, we can remove these jobs from the input J and find a schedule for the remaining jobs. Then we add the small jobs to the end of the schedule by using, say, the list scheduling algorithm. This last step increases the length of the schedule by at most εC^*_{max}.

It is not difficult to see that mP_{max} is an upper bound on the length of an optimum schedule. We partition the time interval from 0 to $(1 + \mu m\varepsilon)mP_{max}$ into $\lceil m(1+\mu m\varepsilon)/(\alpha\varepsilon)\rceil$ equal intervals of length at most $\alpha\varepsilon P_{max}$. (The need for the term $\mu m\varepsilon P_{max}$ in the upper bound of the length of an optimum schedule is clarified below.) These intervals are called *intervals of the first type*. We consider only schedules in which every big operation starts processing at the beginning of some interval of the first type. This restriction does not increase the length of the optimum schedule considerably. Indeed, let us consider the first big operation in an optimum schedule that does not start at the beginning of some interval of the first type. We can simply shift this big operation to the right, so that it starts at the beginning of the next interval. All operations starting after this big operation are also shifted to the right by the same length. Then we do the same thing with the remaining big operations. The overall increase in the length of the optimum schedule is bounded by $\mu|B|\alpha\varepsilon P_{max} \leq \mu m\varepsilon P_{max} \leq \mu m\varepsilon C^*_{max}$. Let $\widetilde{C}^*_{max}$ be the length of an optimum schedule in which the big operations start at the beginning of some interval of the first type. As was noted above, $\widetilde{C}^*_{max} \leq C^*_{max} + \mu m\varepsilon P_{max} \leq (1 + \mu m\varepsilon)C^*_{max}$.

In the rest of this section we consider this restricted job shop scheduling problem in which every big operation must start at the beginning of some interval of the first type. Since the number of big operations is constant and since the number of intervals of the first type is constant, we conclude that the number of different schedules with fixed starting times for the big operations is constant too. For each schedule for the big operations we assign the starting times for the tiny operations within some interval of the second type by solving a linear program as described below.

2.2. Scheduling the tiny operations. Fix some feasible restricted schedule for the big operations within the time interval $[0,(1+\mu m\varepsilon)mP_{max}]$. Let S_{ij} and C_{ij} be the starting and completion times of big operation O_{ij}, respectively. Let $A=\{a_k \mid a_k=S_{ij}$ or $a_k=C_{ij}$ for some big operation $O_{ij}\}$ be the set of starting and completion times of big operations. Notice that

$$|A| \le 2\mu|B| \le 2\mu m/\alpha. \tag{2}$$

Assume that the elements in A are indexed so that $a_1 \le a_2 \le \cdots \le a_{|A|}$. We define two new elements $a_0=0$ and $a_{|A|+1}=C \in [a_{|A|},(1+\mu m\varepsilon)mP_{max}]$ (the exact value of C will be specified later), and partition the time interval from 0 to C into $|A|+1$ intervals $[a_k,a_{k+1})$, $k=0,\ldots,|A|$. We call these intervals *intervals of the second type.* Sometimes we refer to interval $[a_k,a_{k+1})$ simply as interval k. Let Δ_k be the length of the interval k, i.e., $\Delta_k=a_{k+1}-a_k$. Define $\Delta_{tk}=0$ if some big operation is processed in interval k on machine M_t, and $\Delta_{tk}=\Delta_k$ otherwise. So Δ_{tk} is the amount of time that machine M_t can be used during interval k to process tiny operations.

For every job $J_j \in T$ let

$$K_j=\{K=(k_1,k_2,\ldots,k_{\mu_j}) \in Z^{\mu_j} \mid 0 \le k_1 \le k_2 \le \cdots \le k_{\mu_j} \le |A|\}$$

be the set of all feasible assignments of operations of job J_j to intervals of the second type. A tuple $(k_1,k_2,\ldots,k_{\mu_j}) \in K_j$ means that the ith operation of job J_j, $1 \le i \le \mu_j$, starts its processing in interval (of the second type) k_i.

Now we use a linear program to schedule the tiny operations. We define variables x_{jK}, $K=(k_1,k_2,\ldots,k_{\mu_j}) \in K_j$, $J_j \in T$, with the following meaning: variable x_{jK} takes value f, $0 \le f \le 1$, to indicate that a fraction f of the first operation of job J_j is processed in interval k_1 on machine $M_{\pi_{1j}}$, a fraction f of the second operation is processed in interval k_2 on $M_{\pi_{2j}}$, and so on. The linear program is the following. (We assume that we have already chosen the value of the length C of the schedule, so we are interested only in knowing whether the jobs can be scheduled within the time interval $[0,C]$; we show below how to choose C.)

$$\sum_{K \in K_j} x_{jK}=1, \quad J_j \in T, \tag{3}$$

$$\sum_{J_j \in T}\sum_{K \in K_j}\sum_{k_i=k,\, \pi_{ij}=t} p_{ij}\, x_{jK} \le \Delta_{tk}, \quad t=1,\ldots,m,\ k=0,\ldots,|A|, \tag{4}$$

$$x_{jK} \ge 0, \quad K \in K_j,\ J_j \in T. \tag{5}$$

Constraint (3) ensures that job J_j is completely scheduled, while constraint (4) ensures that the total length of operations assigned to interval k on machine M_t does not exceed the length of the interval.

LEMMA 1. *For $C=\widetilde{C}^*_{max}$, the linear program* (3)–(5) *has a feasible solution for some restricted schedule for the big operations.*

Proof. Consider an optimum schedule S^* of the restricted job shop problem. Assume that some tiny operation O_{ij} is processed in consecutive time intervals $b_{ij}, b_{ij}+1, \ldots, e_{ij}$ on machine $M_{\pi_{ij}}$, where b_{ij} might be equal to e_{ij} (corresponding to the case when the operation is completely scheduled in a single interval). Let $f_{ij}(k)$ be the fraction of operation O_{ij} that is scheduled in interval k.

Consider the linear program (3)–(5) corresponding to the schedule for the big operations induced by S^*. Assign values to the variables x_{jK}, $K = (b_{1j}, b_{2j}, \ldots, b_{\mu_j,j})$, as follows. Set $x_{jK} = f$, where $f = \min\{f_{ij}(b_{ij}) \mid 1 \leq i \leq \mu_j\}$ is the smallest fraction of an operation of job J_j that is scheduled in the first interval assigned to it in S^*. Next we assign values to the other variables x_{jK} to cover the remaining $1-f$ fraction of each operation. To do this, for every operation O_{ij}, we make $f_{ij}(b_{ij}) = f_{ij}(b_{ij}) - f$. Clearly, for at least one operation O_{ij} the new value of $f_{ij}(b_{ij})$ will be set to zero. For those operations with $f_{ij}(b_{ij}) = 0$ we set $b_{ij} = b_{ij} + 1$, since the first interval for the rest of the operation O_{ij} is interval $b_{ij} + 1$. Then we assign value to the new variable x_{jK}, $K = (b_{1j}, b_{2j}, \ldots, b_{\mu_j,j})$ as above, and repeat the process until $f = 0$. Note that each iteration of this process assigns a value to a different variable x_{jK}, since from one iteration to the next the value of at least one of the indices b_{ij} is increased.

By construction, the above assignment of values to the variables x_{jK} satisfies constraint (3). Since the linear program was defined with respect to the schedule S^*, the values assigned to the variables also satisfy constraint (4). Therefore, this assignment of values to variables x_{jK} is a feasible solution for the linear program. □

Let C_{min} be the smallest value C such that linear program (3)–(5) has a feasible solution for some schedule of the big operations. By Lemma 1, if $C = \widetilde{C}^*_{max}$ the linear program has a feasible solution, so $C_{min} \leq \widetilde{C}^*_{max}$. For any fixed value $\delta \geq 0$, we can find a value C satisfying $C_{min} \leq C \leq C_{min} + \delta P_{max}$ by using binary search. Thus we must solve linear program (3)–(5) at most $\lceil \log_2((1+\mu m\varepsilon)m/\delta)\rceil$ times. Since C_{min} is a lower bound for $\widetilde{C}^*_{max}$, then $C \leq \widetilde{C}^*_{max} + \delta P_{max} \leq C^*_{max} + 2\mu m\varepsilon P_{max}$, for $\delta = \mu m\varepsilon$, because $\widetilde{C}^*_{max} \leq C^*_{max} + \mu m\varepsilon P_{max}$ as noted at the end of section 2.1.

Note that we could easily rewrite the linear program so that its solution gives the value of C_{min}. We decide to perform the binary search to find the value of C tough, since this will help us to design a linear time PTAS for the job shop scheduling problem as we show in section 3.

The linear program has $|T|+m(|A|+1)$ constraints and at most $|T|(|A|+1)^{\mu}$ variables. Therefore, a basic feasible solution is guaranteed to have at most $|T|+m(|A|+1)$ nonzero variables. This solution can have at most $m(|A|+1)$ jobs with fractional variables, since by constraint (3) every job must have at least one positive variable associated with it. (This kind of argument was first made and exploited by Potts [9] in the context of parallel machine scheduling.)

We now describe a simple rounding procedure to obtain an integral (and possibly infeasible) solution for the linear program. If job J_j has more than one nonzero variable associated with it, we set one of them to 1 and the others to 0 in an arbitrary manner. In this solution the tiny operations have a unique assignment to intervals of the second type.

Note that we might have to increase the lengths of some intervals of the second type to accommodate those operations that previously had fractional assignments. Let $D(k)$ be the total processing time of tiny operations assigned to interval k such that the jobs corresponding to these operations receive fractional assignments from the linear program. Notice that by (1) and (2), $\sum_{k=0}^{|A|} D(k) \leq m(|A|+1)\alpha\varepsilon P_{max} = O(m^2\mu)\varepsilon P_{max}$. Thus this rounding procedure produces a nonpreemptive but possibly infeasible schedule for J of length at most $C + O(m^2\mu)\varepsilon C^*_{max}$.

2.3. Finding a feasible schedule. Consider some interval $[a_k, a_{k+1})$ of the second type. Let $p_{max}(k)$ be the length of the longest tiny operation assigned to this interval. By construction, in the above rounded solution the total length of operations assigned to this interval on each machine is at most $a_{k+1} - a_k + D(k)$. We consider now the problem of scheduling the tiny operations within the interval. This is simply a smaller instance of the job shop problem, and by using Sevastianov's algorithm [10] it is possible to find a feasible schedule of length at most $a_{k+1} - a_k + D(k) + O(m\mu^3)p_{max}(k)$.

Note that we can schedule the tiny operations in each interval $[a_k, a_{k+1})$ independently from the operations in any other interval $[a_i, a_{i+1})$. Moreover, if we add $D(k) + O(m\mu^3)p_{max}(k)$ to the length of each interval, the union of the schedules for these intervals yields a feasible solution for the original constrained job shop problem (where all big operations start at the beginning of some interval of the first type). The makespan of this schedule is at most

$$C + \sum_{k=0}^{|A|} \big(D(k) + O(m\mu^3)p_{max}(k)\big) \leq C + O(m^2\mu)\varepsilon P_{max} + O(m\mu^3)(|A| + 1)\alpha\varepsilon P_{max}$$

$$\leq C + O(m^2\mu^4)\varepsilon P_{max} \leq C^*_{max} + O(m^2\mu^4)\varepsilon P_{max}.$$

Since m and μ are both constants and ε is an arbitrary rational number, then our algorithm can find in polynomial time a solution of length at most $1 + \epsilon$ times the optimum for any value $\epsilon > 0$.

THEOREM 1. *The above algorithm is a PTAS for the job shop scheduling problem when m and μ are fixed.*

3. Speed up to linear time. In the PTAS that we have just described there are two steps that seem to require more than linear time: finding a basic feasible solution for the linear program and running Sevastianov's algorithm. In the next three sections we show how to perform these steps in linear time.

3.1. Approximate solution of the linear program. Since we do not solve exactly the job shop scheduling problem, we do not need to solve the linear program exactly either. An approximate solution would suffice. We can find an approximate solution of the linear program using an algorithm by Grigoriadis and Khachiyan [2].

A *convex block-angular resource sharing problem* has the form

$$\min\left\{\lambda \;\middle|\; \sum_{k=1}^{K} f_t^k(x^k) \leq \lambda, \text{ for all } t = 1, \dots, L, \text{ and } x^k \in \mathcal{B}^k,\ k = 1, \dots, R\right\},$$

where $f_t^k : \mathcal{B}^k \to \Re^+$ are nonnegative continuous convex functions and $\mathcal{B}^k$ are disjoint convex compact sets called *blocks*. The potential price directive decomposition method of Grigoriadis and Khachiyan [2] can be used to find a $(1 + \rho)$-approximate solution to this problem for any value $\rho > 0$. This algorithm needs $O(L(\rho^{-2} \ln \rho^{-1} + \ln L)(L \ln\ln(L/\rho) + RF))$ time, where F is the time needed to find a ρ-approximate solution to the following problem on any block $\mathcal{B}^k$:

$$\min\left\{\sum_{i=1}^{L} p_i f_i^k(x^k) \;\middle|\; x^k \in \mathcal{B}^k\right\} \tag{6}$$

for some vector $(p_1, \ldots, p_L) \in \Re^L$.

We can rewrite the linear program (3)–(5) as a convex block-angular resource sharing problem as follows. We replace condition (4) of the linear program by the following:

$$(7)\quad \frac{1}{\Delta_{tk}} \sum_{J_j \in T} \sum_{K \in K_j} \sum_{k_i = k,\, \pi_{ij} = t} p_{ij}\, x_{jK} \le \lambda, \quad t = 1, \ldots, m,\ k = 0, \ldots, |A|,\ \Delta_{tk} \neq 0,$$

where λ is a nonnegative value. If for some pair t, k, the value of Δ_{tk} is zero, we remove the corresponding condition (4) from the linear program and set the corresponding variables x_{jK} to zero. We call this new linear program LP$'(\lambda)$.

This linear program has the above block angular structure. For each tiny job J_j let x_{jK_j} be the at most $(|A| + 1)^{\mu_j}$-dimensional vector whose components are the different variables x_{jK} of job J_j. For job J_j we define the set $\mathcal{B}_j = \{x_{jK_j} \mid$ conditions (3) and (5) are satisfied$\}$. This set is a block of constant dimension and so the block optimization problem (6) can be solved in constant time. Let $f_t^k(x_{jK}) = \frac{1}{\Delta_{tk}} \sum_{J_j \in T} \sum_{K \in K_j} \sum_{k_i = k,\, \pi_{ij} = t} p_{ij}\, x_{jK}$. Note that these functions f_t^k are nonnegative.

The logarithmic potential price directive decomposition method developed by Grigoriadis and Khachiyan [2] can be used either to determine that the linear program LP$'(\lambda)$ is infeasible or to find a $(1 + \varepsilon)$-approximation to the smallest value λ for which LP$'(\lambda)$ has a feasible solution. This procedure runs in linear time [2, Theorem 3]. Since, by choosing $C = \widetilde{C}^*_{max}$, LP$'(\lambda)$ has a feasible solution for $\lambda = 1$, then we can find in linear time a solution of the linear program in which the length of each interval Δ_{tk} is enlarged to $\Delta_{tk}(1 + \varepsilon)$. The length of this solution is no more than $(1 + \varepsilon)$ times larger than the length of a solution for the original linear program.

The algorithm of [2] finds a feasible solution for the linear program but not necessarily a basic feasible solution. So we need a linear time rounding procedure which given a feasible solution of the linear program LP$'(\lambda)$ finds a solution with at most $O(|A|)$ fractional variables where the hidden constants depend only on m and μ.

3.2. Rounding procedure. In this subsection we show how to round any feasible solution for the linear program LP$'(1+\varepsilon)$ to get a new feasible solution in which all but a constant number of variables have value 0 or 1. Moreover, we show that we can do this rounding procedure in linear time.

First we write the linear program in matrix form as $Bx = b$, $x \ge 0$, where B is the constraint matrix. Let $\bar{x}$ be a feasible solution of the linear program. The components of $\bar{x}$ are the values of the variables x_{jK}. Without loss of generality, let us assume that the columns of B are indexed so that the columns corresponding to variables x_{jK} of the same job J_j appear in adjacent positions. We might also assume that at all times during the rounding procedure each job J_j is associated with at least two columns in B. This assumption can be made, since if job J_j has only one associated column, then by constraint (3) the corresponding variable x_{jK} must have value either zero or one. Let $\mathcal{C}$ be the set formed by the first $2m(|A| + 1) + 2$ columns of B. At most $2m(|A| + 1) + 1$ rows of $\mathcal{C}$ have nonzero entries. To see this observe that at most $m(|A| + 1) + 1$ of these entries come from constraint (3) because of the above assumption on the number of columns for each job, while at most $m(|A| + 1)$ nonzero entries come from constraint (7).

Let $\mathcal{M}$ be the matrix formed by the nonzero rows of $\mathcal{C}$. Since $\mathcal{M}$ has at most $2m(|A| + 1) + 1$ rows and exactly $2m(|A| + 1) + 2$ columns, then $\mathcal{M}$ is singular, and hence there exists at least one nonzero vector y such that $\mathcal{M}y = 0$. Let $\delta \geq 0$ be the smallest value such that some component of the vector $\bar{x} + \delta y$ is either zero or one. (If the dimension of y is smaller than the dimension of $\bar{x}$ we pad it with an appropriate number of zero entries.) Note that the vector $\bar{x} + \delta y$ is a feasible solution of the linear program LP$'(1 + \varepsilon)$. Let x^0 and x^1 be lists containing, respectively, the zero and one components of vector $\bar{x} + \delta y$. We update the solution of the linear program by making $\bar{x} \leftarrow \bar{x} + \delta y$ and then removing from $\bar{x}$ all variables in x^0 and x^1. We also discard all columns of B corresponding to the variables in x^0 and x^1. If x^1 is not empty, then vector b is set to $b - \sum_{i \in x^1} B[*, i]$, where $B[*, i]$ is the column of B corresponding to variable i.

This process rounds the value of at least one variable x_{jK} to either 0 or 1. The procedure is repeated until there are at most $2m(|A|+1)+1$ columns in B, and hence there are at most $m(|A|+1)+1$ jobs with fractional variables.

LEMMA 2. *The above algorithm transforms in $O(n)$ time any feasible solution of* LP$'(\lambda)$ *into another feasible solution in which at most a constant number of variables have fractional values.*

Proof. In every iteration of the algorithm, vector y can be found in constant time since the size of matrix $\mathcal{M}$ is constant. Also, the value of δ can be found in constant time, since vector y has a constant number of nonzero entries. Therefore, every iteration of the algorithm requires $O(1)$ time.

Since in each iteration at least one variable is rounded to zero or one, then the algorithm performs only $O(n)$ iterations. □

Let $\mathcal{F}$ be the set of jobs that receive fractional assignments in the rounded solution. For each job in $\mathcal{F}$ we arbitrarily choose one of its nonzero variables and set it to 1 while we set all other variables to 0. Using arguments similar to those of section 2.2 we can show that this rounding procedure produces a nonpreemptive, but possibly infeasible, schedule for the jobs, and the total length of the schedule is at most $(1 + O(m^2\mu)\varepsilon)C^*_{max}$.

3.3. Merging trick. Consider the instance of the job shop scheduling problem defined by the tiny jobs that are assigned to the kth interval of the second type. Sevastianov's algorithm finds in $O(n^2\mu^2m^2)$ time a schedule of length at most $(1 + \varepsilon)(a_{k+1} - a_k) + D(k) + O(m\mu^3)p_{max}(k)$ for these jobs, where $p_{max}(k)$ is the length of the largest operation in interval k. For a job J_j let $(m_{1j}, m_{2j}, \ldots, m_{\mu j})$ be a vector that describes the machines on which its operations must be performed. Let us partition the set of jobs J into m^μ groups $\mathcal{J}_1, \mathcal{J}_2, \ldots, \mathcal{J}_{m^\mu}$ such that all jobs in group $\mathcal{J}_i$ have the same machine vector and jobs from different groups have different machine vectors.

Consider the jobs in one of the groups $\mathcal{J}_i$. Let J_j and J_h be two jobs from $\mathcal{J}_i$ such that each one of them has length smaller than $\alpha\varepsilon P_{max}/2$. We "glue" together these two jobs to form a composed job in which the processing time of the ith operation is equal to the sum of the processing times of the ith operations of J_j and J_h. We repeat this process until at most one job from $\mathcal{J}_i$ has processing time smaller than $\alpha\varepsilon P_{max}/2$. The same procedure is performed in all other groups $\mathcal{J}_j$. At the end of this process, each one of the composed jobs has at most μ operations. The total number of composed jobs is at most $m^\mu + \lceil \frac{2m}{\alpha\varepsilon} \rceil$, and all operations in interval k have processing times smaller than $\max\{p_{max}(k), \alpha\varepsilon P_{max}\}$. Note that this merging procedure runs in linear time and that a feasible schedule for the original jobs can be easily obtained from a feasible schedule for the composed jobs.

We run Sevastianov's algorithm on this set of composed jobs to get a schedule of length $(1+\varepsilon)(a_{k+1}-a_k)+D(k)+O(m\mu^3)\max\{p_{max}(k),\alpha\varepsilon P_{max}\}$. The time needed to get this schedule is $O((m^\mu+\frac{2m}{\alpha\varepsilon})^2\mu^2m^2)$. So Sevastianov's algorithm needs only constant time plus linear preprocessing time. Notice also that the analysis in subsection 2.3 with minor changes holds also for this case.

THEOREM 2. *The algorithm described above is a linear time approximation scheme for the job shop scheduling problem when m and μ are fixed.*

4. Preemptive job shop scheduling problem.

In this section we describe a PTAS for the preemptive version of the job shop scheduling problem when m and μ are fixed. As in the nonpreemptive case we divide the set of jobs J into big jobs B, small jobs S, and tiny jobs T. The sets are chosen as in the nonpreemptive version. The small jobs are removed and reintroduced later at the end of the schedule as in the nonpreemptive case.

We consider only restricted preemptive schedules in which the earliest time when a big operation starts processing and the time when a big operation is completed lie at the boundaries of intervals of the first type. (So if a big operation is preempted, the times at which the operation is suspended and resumed do not need to coincide with boundaries of intervals of the first type.) By using arguments similar to those of section 2.1 we can show that an optimum restricted preemptive schedule has length $\tilde{C}^*_{max}\le(1+O(\mu m\varepsilon))C^*_{max}$, where C^*_{max} is the length of an optimum preemptive schedule.

An *allotment* for the big jobs specifies for each big operation a set of consecutive intervals of the first type where the operation can be scheduled. Since there is a constant number of big operations, there is also a constant number of allotments. Fix one allotment for the big operations such that operations from the same big job are assigned disjoint intervals and there is a feasible preemptive schedule for the big jobs that respects the allotment.

Let S_{ij} be the starting time of the first interval of the first type assigned to big operation O_{ij}, and let C_{ij} be the ending time of the last interval of the first type assigned to it. We define intervals of the second type in a similar way as we did in section 2.2.

An operation O_{ij} of a big job is scheduled in consecutive intervals of the second type $[a_k,a_{k+1}),\ldots,[a_{k+t-1},a_{k+t})$, where a_k is the starting time and a_{k+t} is the completion time of O_{ij}. Any fraction (possible equal to zero) of the operation might be scheduled in any one of these intervals. Because of the way in which the allotment was chosen, in each interval of the second type there is at most one operation from any given big job.

As for the nonpreemptive case, for every tiny job J_j we define

$$K_j=\{K=(k_1,k_2,\ldots,k_{\mu_j})\in Z^{\mu_j}\mid 0\le k_1\le k_2\le\cdots\le k_{\mu_j}\le|A|\}.$$

For each big job J_j we define a similar set K_j, but the tuples in K_j allow only placement of the (pieces of) operations of job J_j in the intervals defined by the allotment.

For each job J_j we define variables x_{jK}, $K\in K_j$. We assign operations to intervals of the second type by solving the following linear program:

$$\sum_{K\in K_j}x_{jK}=1,\quad J_j\in J\setminus S,\tag{8}$$

$$\sum_{J_j\in J\setminus S}\sum_{K\in K_j}\sum_{k_i=k,\,\pi_{ij}=t}p_{ij}\,x_{jK}\le\Delta_k,\quad t=1,\ldots,m,\ k=0,\ldots,|A|,\tag{9}$$

$$x_{jK}\ge0,\quad K\in K_j,\ J_j\in J\setminus S.\tag{10}$$

Note that in any solution of this linear program the schedule for the long jobs is always feasible, since there is at most one operation of a given job in any interval of the second type. Let $C_{min} = a_{|A|+1}$ be the smallest value such that linear program (8)–(10) has a feasible solution for some allotment. Using an argument similar to that of the proof of Lemma 1 we can prove that C_{min} is a lower bound on the makespan of an optimum preemptive schedule for the given set of jobs J.

Using binary search we can find a value C satisfying $C_{min} \leq C \leq C_{min} + \delta P_{max}$, for any fixed $\delta \geq 0$, by approximately solving the above linear program a constant number of times. Since linear programs (3)–(5) and (8)–(10) have the same structure we can use our rounding procedure to find in linear time a solution for the new linear program in which at most $m(|A| + 1) + 1$ jobs receive fractional assignments. (See section 3.2.)

After rounding the solution of the linear program we find a feasible schedule for every interval of the second type as follows. Consider an interval $[a_k, a_{k+1})$. Remove from the interval the operations belonging to big jobs. These operations will be reintroduced to the schedule later. Then use Sevastianov's algorithm as described in section 3.3 to find a feasible schedule for the small and tiny jobs assigned to that interval. Finally place back the operations from the big jobs, scheduling them in the empty gaps left by the small and tiny jobs. Note that it might be necessary to further split an operation of a big job in order to make it fit in the empty gaps. At the end we have a feasible schedule because there is at most one operation of each big job in the interval.

In this schedule the number of preemptions is at most $n\mu$ (since after introducing the operations from the big jobs, we might have this many preemptions for them). So there are in total $O(n)$ preemptions and only big operations are preempted.

THEOREM 3. *The above algorithm is a linear time approximation scheme for the preemptive version of the job shop scheduling problem when m and μ are fixed. The solution that the algorithm finds has $O(n)$ preemptions.*

5. Extensions.

Multistage job shop problem. In the s-stage job shop problem each machine of the classical job shop problem is replaced by a set of m_i parallel identical machines, $1 \leq m_i \leq m$. Our polynomial time approximation scheme works also in this case if the number of machines on each stage and the number of stages are fixed. Let the machines on stage i be numbered $s_1, s_2, \ldots, s_{m_i}$. In the linear program we use variables $x_{jK(r_1,\ldots,r_{\mu_j})}$, where r_i indicates the machine where the ith operation O_{ij} of job J_j is scheduled. The same techniques used for the job shop scheduling problem can be used to design a polynomial time approximation scheme for this more general problem.

Dag shop problem. Another generalization of the job shop problem is the dag shop problem [15] (also called the G-problem by Sevastianov [10, 11]). Here each job consists of a set of operations $\{O_{1j}, \ldots, O_{\mu j}\}$, and each job $J_j \in J$ is associated with an acyclic directed graph $R_j = (O_j, E_j)$. In this graph an arc $(O_{i'j}, O_{ij})$ indicates that operation O_{ij} has to be executed after operation $O_{i'j}$. The problem is to find a schedule of minimum length that respects these ordering constraints.

We define restricted schedules for the big jobs that respect the ordering constraints R_j. An acyclic graph R_j can be translated directly into a set of tuples $K_j = \{(k_1, \ldots, k_\mu) \mid 0 \leq k_j \leq |A|$ for all $1 \leq j \leq \mu$ and $k_{i'} \leq k_i$ for every edge $(O_{i'j}, O_{ij}) \in E_j\}$ for each job $J_j \in T \cup S$. Again, the size of each set of tuples is constant, $|K_j| \leq (|A| + 1)^\mu$, so we can use our algorithm with some small changes. Let us consider a single interval $[a_k, a_{k+1})$. Let $O(k)$ be the set of operations assigned to this interval. For each job J_j corresponding to the operations in $O(k)$ we use, instead of the acyclic graph $R_j(k)$ induced by the operations $O_{ij} \in O(k)$, a linear order that extends $R_j(k)$ and apply Sevastianov's algorithm [10] to a smaller instance of the job shop problem in each interval k. The rest of the algorithm is as before. We note that our algorithm does not seem to extend to the preemptive version of this problem.

Job shop problem with release and delivery times. Our techniques can also handle the case in which each job J_j has a *release time* r_j (when it becomes available for processing) and a *delivery time* q_j. If in some schedule job J_j completes processing at time C_j, then its *delivery completion time* is equal to $C_j + q_j$. The goal is to minimize the maximum delivery completion time of any job. Let r_{max} and q_{max} be the maximum release and delivery times, respectively. Then, $\max\{r_{max}, P_{max}, q_{max}\} \leq C^*_{max} \leq r_{max} + mP_{max} + q_{max}$. The idea is to round each release and delivery time up to the nearest multiple of $\varepsilon \cdot \max\{r_{max}, P_{max}, q_{max}\}$ for some value $\varepsilon > 0$. This increases the length of an optimum schedule by at most $2\varepsilon C^*_{max}$. Next we apply a $(1+\varepsilon)$-approximation scheme (described below) that can handle $O(1/\varepsilon)$ different release times and delivery times. This gives an algorithm that finds a solution of length at most $(1+\varepsilon)(1+2\varepsilon) \leq 1+5\varepsilon$ times larger than the optimum.

We can easily modify our linear program to allow a constant number, $O(1/\varepsilon)$, of release dates and delivery times. We also need to make another change to the linear program, since now we cannot remove the small jobs and simply add them to the end of the schedule. Instead, we must define variables x_{jK} for the small jobs and use the linear program for assigning them to intervals of the second type as we do for the tiny jobs. Now the number of intervals of the second type is larger, since we add to A each release time r_j and each completion time $C - q_j$, where C is the length of the schedule as described in section 2.2. Note that the total number of intervals is still constant: $O(m\mu/\alpha + 1/\varepsilon)$. We can solve the linear program as before in linear time. The rest of the approximation scheme is similar to that for the job shop scheduling problem. The analysis has to be only slightly changed due to the presence of small jobs in the intervals of the second type.

Note that this approach works even if every operation has its own release and delivery time.

Acknowledgments. We are grateful to Alexander Kononov, Lorant Porkolab, and Sergey Sevastianov for many helpful discussions and comments. The authors would like to thank David Shmoys for sending us his manuscript [14] about Sevastianov's algorithm. We thank the referees for their careful comments that allowed us to greatly improve this manuscript.

REFERENCES

[1] L. A. Goldberg, M. Paterson, A Srinivasan, and E. Sweedyk, *Better approximation guarantees for job-shop scheduling*, SIAM J. Discrete Math., 14 (2001), pp. 67–92.

[2] M. D. Grigoriadis and L. G. Khachiyan, *Coordination complexity of parallel price-directive decomposition*, Math. Oper. Res., 21 (1996), pp. 321–340.

[3] L. A. Hall, *Approximability of flow shop scheduling*, Math. Programming, 82 (1998), pp. 175–190.

[4] K. Jansen, R. Solis-Oba, and M. I. Sviridenko, *Makespan minimization in job shops: A polynomial time approximation scheme*, in Proceedings of the 31st Annual ACM Symposium on Theory of Computing, Atlanta, GA, 1999, pp. 394–399.

[5] K. Jansen, R. Solis-Oba, and M. I. Sviridenko, *A linear time approximation scheme for the job shop scheduling problem*, in Proceedings of the Second Workshop on Approximation

Algorithms, Berkeley, CA, 1999, Lecture Notes in Comput. Sci. 1671, Springer, Berlin, 1999, pp. 177–188.

[6] E. L. Lawler, J. K. Lenstra, A. H. G. Rinooy Kan, and D. B. Shmoys, *Sequencing and scheduling: Algorithms and complexity*, in Handbooks in Operations Research and Management Science, Vol. 4, Logistics of Production and Inventory, S. C. Graves, A. H. G. Rinooy Kan, and P. H. Zipkin, eds., North-Holland, Amsterdam, 1993, pp. 445–522.

[7] T. Leighton, B. Maggs, and S. Rao, *Packet routing and job-shop scheduling in* $O(\text{congestion} + \text{dilation})$ *steps*, Combinatorica, 14 (1994), pp. 167–186.

[8] T. Leighton, B. Maggs, and A. Richa, *Fast algorithms for finding* $O(\text{congestion} + \text{dilation})$ *packet routing schedules*, Combinatorica, 19 (1999), pp. 375–401.

[9] C. N. Potts, *Analysis of a linear programming heuristic for scheduling unrelated parallel machines*, Discrete Appl. Math., 10 (1985), pp. 155–164.

[10] S. V. Sevastianov, *Bounding algorithm for the routing problem with arbitrary paths and alternative servers*, Cybernetics, 22 (1986), pp. 773–780.

[11] S. V. Sevastianov, *On some geometric methods in scheduling theory: A survey*, Discrete Appl. Math., 55 (1994), pp. 59–82.

[12] S. V. Sevastianov and G. J. Woeginger, *Makespan minimization in preemptive two machine job shops*, Computing, 60 (1998), pp. 73–79.

[13] S. V. Sevastianov and G. J. Woeginger, *Makespan minimization in open shops: A polynomial time approximation scheme*, Math. Programming, 82 (1998), pp. 191–198.

[14] D. B. Shmoys, unpublished manuscript.

[15] D. B. Shmoys, C. Stein, and J. Wein, *Improved approximation algorithms for shop scheduling problems*, SIAM J. Comput., 23 (1994), pp. 617–632.

[16] D. P. Williamson, L. A. Hall, J. A. Hoogeveen, C. A. J. Hurkens, J. K. Lenstra, S. V. Sevastianov, and D. B. Shmoys, *Short shop schedules*, Oper. Res., 45 (1997), pp. 288–294.

SIAM J. DISCRETE MATH.
Vol. 16, No. 2, pp. 301–312

TESTING BANDWIDTH k FOR k-CONNECTED GRAPHS*

KONRAD ENGEL[†] AND SVEN GUTTMANN[‡]

Abstract. We present a linear-time algorithm to decide whether a given k-connected graph has bandwidth k, where k is a fixed positive integer. This improves the general $O(n^k)$-time-algorithm of Gurari and Sudborough, based on a dynamic programming approach of Saxe, for the recognition of bandwidth-k graphs on n vertices in the special case of connectivity k.

Key words. bandwidth, bandwidth-k graph, k-connected graph, linear-time algorithm, linear layout, start sequence

AMS subject classifications. 05C78, 68R10, 90C35

PII. S0895480199351148

1. Introduction.

In this paper, we deal with simple graphs $G = (V, E)$ on n vertices without loops and multiple edges. A *(linear) layout* of G is a bijective mapping $f : V \to \{1, \ldots, n\}$. The *bandwidth of a layout* f, denoted by $bw(f)$, is defined to be

$$bw(f) := \max\{|f(u) - f(v)| : \{u, v\} \in E\}.$$

The *bandwidth of the graph* G is

$$bw(G) := \min\{bw(f) : f \text{ is a layout of } G\}.$$

A graph G is *k-connected* if $|V| > k$ and, for all subsets V' of V with $|V'| < k$, the graph induced by $V \setminus V'$ is connected.

If $bw(f) = k$ (or $bw(G) = k$), we briefly speak of a *bw–k layout* (or a *bw–k graph*, respectively). For the recognition of bw–2 graphs, a linear-time algorithm was given by Garey et al. [5]. Studying the structure of bw–2 graphs, we presented a new algorithm for this problem in [4]. Saxe [8] developed an algorithm that recognizes bw–k graphs with time and space complexity $O(n^{k+1})$, and Gurari and Sudborough [6] improved it to complexity $O(n^k)$. In [7] Makedon, Sheinwald, and Wolfsthal designed a simple linear-time algorithm for testing bandwidth 2 for 2-connected graphs. Bodlaender, Fellows, and Hallett [1] have shown that testing $bw(G) \leq k$ is not likely to be fixed parameter tractable (cf. [3]) and, in particular, is not likely to be linear-time solvable without additional special assumptions.

In this paper, we generalize [7] in order to test bandwidth k for k-connected graphs in linear time (k fixed). We generally assume that the graph is represented by a linked adjacency list. In the first part, we present some general results for bw–k k-connected graphs. It will turn out that we have to distinguish the cases k odd and even. The problem is much easier to handle for odd k than for even k. In this more difficult case, we have to work with a detailed structural characterization of G. Each bw–1 1-connected graph obviously is a chain. Thus, throughout we let $k \geq 2$.

[*] Received by the editors January 27, 1999; accepted for publication (in revised form) September 19, 2002; published electronically February 20, 2003.
http://www.siam.org/journals/sidma/16-2/35114.html

[†] Fachbereich Mathematik, Universität Rostock, 18051 Rostock, Germany (konrad.engel@mathematik.uni-rostock.de).

[‡] Arvato Systems GmbH, An der Autobahn, P.O. Box 180, 33311 Gütersloh, Germany (sven.guttmann@bertelsmann.de).

2. On the structure of *bw-k k*-connected graphs. First we repeat some definitions from graph theory. A vertex v is a *neighbor* of some given vertex u if v is *adjacent* to u; i.e., $\{u, v\}$ is an edge. We briefly denote edges by uv. The *(first) neighborhood* $N^1(u)$ of a vertex $u \in V$ is the set of all neighbors of u. The *degree* $\deg(u)$ of a given vertex u is the number of all neighbors of u, i.e., $\deg(u) = |N^1(u)|$. The *second neighborhood* $N^2(u)$ of a vertex $u \in V$ is the set of all vertices $w \in V \setminus (N^1(u) \cup \{u\})$ such that there is a vertex $v \in N^1(u)$ with $vw \in E$; i.e., the distance between u and w in G is two.

Let $A \subseteq V$. The *boundary of* A is defined to be the set

$$B(A) := \{v \in V \setminus A : \ v \text{ is adjacent to some } w \in A\}.$$

LEMMA 1. *Let G be k-connected and $A \subseteq V$. Then*

$$|B(A)| \geq \min\{k, n - |A|\}.$$

Proof. Since $B(A) \cap A = \emptyset$, clearly $|B(A)| \leq n - |A|$. If $|B(A)| < n - |A|$, the deletion of $B(A)$ disconnects the graph, and hence $|B(A)| \geq k$. If $|B(A)| = n - |A|$, the assertion is trivially true. □

In particular we have, for a k-connected graph G,

$$\deg(v) \geq k \text{ for all } v \in V. \tag{1}$$

From (1) it follows that $bw(G) \geq k$ if G is a k-connected graph. Thus we have $bw(G) = k$ for such a graph if there is a *bw–k* layout of G. Let $\Delta(G)$ be the maximum vertex degree of G. If G is a *bw–k* graph, then obviously

$$\Delta(G) \leq 2k; \tag{2}$$

cf. [2]. Under the assumption that the graph is represented by a linked adjacency list, inequality (2) can be tested in linear time (while the neighborhood of a vertex does not contain more than $2k$ vertices, go to the next vertex). Thus, we assume this inequality throughout the paper.

For a given layout f of G and for $j = 1, \dots, n$, let

$$A_j := \{v \in V : f(v) \leq j\}, \tag{3}$$

$$E_j := \{v \in V : f(v) \geq j\}. \tag{4}$$

LEMMA 2. *Let G be k-connected and f be a bw–k layout of G. Then, for $j = 1, \dots, n$,*

$$B(A_j) = \{v \in V : f(v) \in \{j + 1, \dots, \min\{n, j + k\}\}\}, \tag{5}$$

$$B(E_j) = \{v \in V : f(v) \in \{\max\{1, j - k\}, \dots, j - 1\}\}. \tag{6}$$

Proof. Let $v \in B(A_j)$. By definition of $B(A_j)$, $f(v) \geq j + 1$. Moreover, there exists some $w \in A_j$ with $vw \in E$. Thus $k \geq f(v) - f(w) \geq f(v) - j$, i.e., $f(v) \leq j + k$. Consequently,

$$B(A_j) \subseteq \{v \in V : f(v) \in \{j + 1, \dots, \min\{n, j + k\}\}\}.$$

By Lemma 1, $|B(A_j)| \geq \min\{k, n - j\}$, and hence the inclusion is an equality. The sets E_j can be treated analogously. □

Note that, under the suppositions of Lemma 2, for $1 < j \leq n-k$ we have

$$|B(A_j) \setminus B(A_{j-1})| = 1$$

and that for $v \in B(A_j) \setminus B(A_{j-1})$, $f(v) = j + k$. This vertex v is adjacent to only one vertex in A_j, namely the vertex w with $f(w) = j$.

Often we present layouts f as n-tuples $(u_1, \ldots, u_n)$. Here u_i is the vertex of G with $f(u_i) = i$, $i = 1, \ldots, n$. For a *bw–k* layout $(u_1, \ldots, u_n)$ the vertex u_1 (resp., u_n) is called *start vertex* (resp., *end vertex*). Note that we have $A_j = \{u_1, \ldots, u_j\}$ and $E_j = \{u_j, \ldots, u_n\}$. We say that u_i is *left* of u_j (and u_j is *right* of u_i) if $i < j$. A *partial layout* of G is a layout of an induced subgraph of G. We also present partial layouts as certain l-tuples $(u_1, \ldots, u_l)$, where $1 \leq l \leq n$. A partial layout $(u_1, \ldots, u_l)$ is called a *start sequence* if $l = k+1$ and there exists a *bw–k* layout $(u_1, \ldots, u_l, u_{l+1}, \ldots, u_n)$.

The following lemma is a generalization of Lemma 1 in [7], where the case $k = 2$ was considered. It follows immediately from Lemma 2 and the subsequent remark.

LEMMA 3. *Let G be a k-connected graph with $n \geq k+1$.*

(a) *Let $f = (u_1, \ldots, u_n)$ be a bw–k layout of G. Then $u_1u_2, \ldots, u_1u_k$, $u_iu_{i+k}, i = 1, \ldots, n-k$, and $u_{n-k+1}u_n, \ldots, u_{n-1}u_n$ are edges in G.*

(b) *A partial layout $(u_1, \ldots, u_{k+1})$ can be extended in at most one way to a bw–k layout $(u_1, \ldots, u_n)$; i.e., start sequences and bw–k layouts are in one-to-one correspondence.*

Given a partial layout $f = (u_1, \ldots, u_l)$ we call the vertices $u_1, \ldots, u_l$ *labeled* and all other vertices *unlabeled*. Let *kToLast*$(f) := u_{l-k+1}$. For $u \notin f$ let $f||u := (u_1, \ldots, u_l, u)$. As mentioned in the preceding lemma, for a k-connected graph with $\Delta(G) \leq 2k$, the extension of a partial layout can be easily carried out in the following way:

```
Procedure Start-sequence
    Input: The partial layout $f = (u_1, \ldots, u_{k+1})$;
    While $|f| < |V|$ do
        $x := kToLast(f)$;
        Let $U$ be the set of unlabeled neighbors of $x$;
        If $|U| > 1$ then STOP — $(u_1, \ldots, u_{k+1})$ is not a start sequence;
        Let $U = \{u\}$;
        $f := f||u$;
    Output: The bw–k layout $f$.
```

Note that we cannot have $U = \emptyset$ in the while loop since otherwise $|B(A)| < k$ for $A := \{u_1, \ldots, x\}$, a contradiction to Lemma 1. The correctness of the procedure immediately follows from Lemma 3. The linear-time complexity is obvious. Thus the *main problem consists of the determination of a start sequence.* The following two lemmas contain characterizations of vertices of a start sequence that will be used later.

LEMMA 4. *Let G be a k-connected graph with $n \geq 2k+1$ and let $f = (u_1, \ldots, u_n)$ be a bw–k layout of G. Then*

$$|N^1(u_1)| = |N^2(u_1)| = k, \tag{7}$$

$$|N^1(u_n)| = |N^2(u_n)| = k. \tag{8}$$

Proof. In (5) of Lemma 2, take $j := 1$ and $j := k+1$ (resp., in (6) of Lemma 2, $j := n$ and $j := n-k$). ☐

LEMMA 5. *Let G be a k-connected graph with $n \geq 2k+1$ and let $f = (u_1, \dots, u_n)$ be a bw–k layout of G. Then, for $i = 1, \dots, k$,*

$$\deg(u_i) \leq k + i - 1, \tag{9}$$

$$|N^1(u_i) \cap N^1(u_1)| \geq k - i, \tag{10}$$

$$\deg(u_{n+1-i}) \leq k + i - 1, \tag{11}$$

$$|N^1(u_{n+1-i}) \cap N^1(u_n)| \geq k - i. \tag{12}$$

Proof. We prove only (9) and (10) since (11) and (12) can be treated analogously. In (5) of Lemma 2, take $j := i$. This yields

$$\deg(u_i) \leq |A_{i-1}| + |B(A_i)| = i - 1 + k.$$

Obviously,

$$|N^1(u_i) \setminus N^1(u_1)| \leq |\{u_1, u_{k+2}, \dots, u_{i+k}\}| = i.$$

Consequently,

$$|N^1(u_i) \cap N^1(u_1)| = |N^1(u_i)| - |N^1(u_i) \setminus N^1(u_1)| \geq k - i. \qquad \square$$

The main purpose of the following theorem is a characterization of vertices which have the property of start vertices given in Lemma 4 but which themselves are not start vertices.

THEOREM 1. *Let $G = (V, E)$ be a k-connected graph with $n \geq 4k+1$ and let $f = (u_1, \dots, u_n)$ be a bw–k layout of G. Let $2k+1 \leq i \leq n-2k$ and $\deg(u_i) = k$. Let $u_{l_1}, \dots, u_{l_\lambda}$ with $l_1 > \cdots > l_\lambda$ and $u_{r_1}, \dots, u_{r_\rho}$ with $r_1 < \cdots < r_\rho$ be the left, resp., right, neighbors of u_i ($\lambda + \rho = k$). Let $t := k/2$.*

(a) *We have $|N^2(u_i)| \geq k$.*

(b) *If $|N^2(u_i)| = k$, then $\lambda = \rho = t$ and, for $j = 1, \dots, t$,*

$$N^1(u_{l_j}) = \{u_{l_j-k}, \dots, u_{l_1-k}, u_{l_t}, \dots, u_{l_1}, u_i, u_{r_1}, \dots, u_{r_{t-j}}\} \setminus \{u_{l_j}\},$$
$$N^1(u_{r_j}) = \{u_{r_j+k}, \dots, u_{r_1+k}, u_{r_t}, \dots, u_{r_1}, u_i, u_{l_1}, \dots, u_{l_{t-j}}\} \setminus \{u_{r_j}\}.$$

Proof. Let $M := \{u_{l_\lambda-k}, \dots, u_{l_1-k}, u_{r_1+k}, \dots, u_{r_\rho+k}\}$. By Lemma 3(a), $M \subseteq N^2(u_i)$, $l_\lambda = i - k$, and $r_\rho = i + k$. Consequently, $|N^2(u_i)| \geq |M| = k$ and (a) is proved.

Thus let $|N^2(u_i)| = k$, i.e., $N^2(u_i) = M$. By Lemma 3(a), $\{u_{l_{\lambda-1}+k}, \dots, u_{l_1+k}\} \subseteq N^1(u_i) \cup N^2(u_i)$. Hence $N^2(u_i) = M$ implies

$$\{u_{l_{\lambda-1}+k}, \dots, u_{l_1+k}\} \subseteq \{u_{r_1}, \dots, u_{r_{\rho-1}}\}.$$

Analogously,

$$\{u_{r_1-k}, \dots, u_{r_{\rho-1}-k}\} \subseteq \{u_{l_{\lambda-1}}, \dots, u_{l_1}\}.$$

From both set inclusions it follows that $\lambda - 1 \leq \rho - 1$ and $\rho - 1 \leq \lambda - 1$, i.e., $\lambda = \rho = k/2 = t$, and that the set inclusions are in fact equalities. In particular, $l_j + k = r_{t-j}$, $j = 1, \dots, t-1$. It remains to study the neighborhood of a left neighbor u_{l_j} of u_i, $1 \leq j \leq t$ (right neighbors can be treated analogously). Obviously,

$$N^1(u_{l_j}) \cap M \subseteq \{u_{l_j-k}, \dots, u_{l_1-k}\}, \tag{13}$$

$$N^1(u_{l_j}) \cap N^1(u_i) \subseteq \{u_{l_t}, \dots, u_{l_1}, u_{r_1}, \dots, u_{r_{t-j}}\} \setminus \{u_{l_j}\}, \tag{14}$$

$$N^1(u_{l_j}) \cap \{u_i\} \subseteq \{u_i\}. \tag{15}$$

Since $N^1(u_{l_j}) \subseteq M \cup N^1(u_i) \cup \{u_i\}$ (recall $M = N^2(u_i)$) we have

$$|N^1(u_{l_j})| \leq j + (t + t - j - 1) + 1 = k.$$

Since also $|N^1(u_{l_j})| \geq k$ (recall (1)) we obtain that $|N^1(u_{l_j})| = k$ and that all set inclusions in (13)–(15) are equalities, i.e.,

$$N^1(u_{l_j}) = \{u_{l_j - k}, \dots, u_{l_1 - k}, u_{l_t}, \dots, u_{l_1}, u_i, u_{r_1}, \dots, u_{r_{t-j}}\} \setminus \{u_{l_j}\},$$

and (b) is proved. □

From now on we assume that $n \geq 4k + 1$. Note that "smaller" graphs can be analyzed by complete enumeration.

Let

$$S := \{v \in V : |N^1(v)| = |N^2(v)| = k\}.$$

From Lemma 4 we know that the start and end vertices of a *bw–k* layout of a k-connected graph belong to S. The set S plays an essential role. If $|S| \leq 1$, we may stop our investigations since the given graph is not a k-connected *bw–k* graph. Given a vertex $s \in S$, it is easy to test whether it is a start vertex as follows:

```
Procedure Start-vertex
    Input: The vertex $s \in S$;
    Determine the neighbors $u_1, \dots, u_k$ of $s$;
    For each permutation $\pi$ of $\{1, \dots, k\}$ do
        If $\deg(u_{\pi(i)}) \leq k+i$ and $|N^1(u_{\pi(i)}) \cap N^1(s)| \geq k-(i+1)$, $i = 1, \dots, k-1$,
        then Start-sequence $(s, u_{\pi(1)}, \dots, u_{\pi(k)})$ and STOP if a bw–k layout
        is found;
    Output: A bw–k layout of $G$ with start vertex $s$ iff there exists one.
```

The correctness of the procedure follows from Lemma 5. Since there are only $k!$ (i.e., a constant number of) permutations of the neighbors, this procedure also has linear-time complexity.

3. The case *k* is odd. For this case, the main observation is that $|S| \leq 4k$ holds if the given k-connected graph has a *bw–k* layout $(u_1, \dots, u_n)$. Indeed, if $|S| \geq 4k+1$, then $n \geq 4k + 1$ and there exists some $u_i \in S$ with $2k + 1 \leq i \leq n - 2k$. By Theorem 1(b), u_i has an even number of neighbors, a contradiction.

This leads us to the following easy algorithm.

```
Algorithm bw–k(odd)
    Input: A positive odd integer $k \geq 3$, a $k$-connected graph $G = (V, E)$ with
    $|V| \geq 4k + 1$ and $\Delta(G) \leq 2k$;
    Determine $S := \{v \in V : |N^1(v)| = |N^2(v)| = k\}$;
    If $|S| \leq 1$ or $|S| \geq 4k + 1$ then STOP — $G$ is not a bw–k graph;
    For all $s \in S$ do
        Start-vertex($s$) and STOP if a bw–k layout is found;
    Output: A bw–k layout of $G$ iff $G$ is a bw–k graph.
```

The linear-time complexity follows from the linear-time complexity of the procedure **Start-vertex**.

4. The case *k* is even. Throughout we let $t := k/2$. In contrast to the preceding case, cycles (for $k = 2$) and their obvious generalizations show that the set S may contain vertices that are "far away" from a start vertex. But, by Theorem 1, these vertices $s \in S$ must have the following property: The elements of $N^1(s) \cup N^2(s)$ can be ordered in the form $(v_{-k}, \ldots, v_{-1}, v_1, \ldots, v_k)$ such that, with $v_0 := s$,

$$N^1(v_i) = \{v_{i-t}, \ldots, v_{i+t}\} \setminus \{v_i\}, \qquad -t \leq i \leq t. \tag{16}$$

We call vertices $s \in S$ with this property *suspicious vertices* and call a corresponding sequence $(v_{-k}, \ldots, v_{-1}, v_0, v_1, \ldots, v_k)$ the *bw–k sequence of s*. Note that the inverse sequence $(v_k, \ldots, v_0, \ldots, v_{-k})$ is also a *bw–k* sequence. Suspicious vertices s can be recognized in constant time: After testing $|N^1(s)| = |N^2(s)| = k$ we may either check all $(2k)!$ permutations of $N^1(s) \cup N^2(s)$ or proceed more elegantly by first ordering the neighbors of s using the degrees and then adding step by step the elements of the second neighborhood of s as follows:

```
**Procedure Suspicious-vertex**
    **Input:** The vertex $v_0 := s$;
    Let $G'$ be the graph induced by $N^1(s) \cup \{s\}$ and consider degrees only in $G'$;
    Determine the set $Q$ of vertices of $G'$ of degree $t$;
    If $|Q| \neq 2$ then STOP else denote one element of $Q$ by $v_{-t}$ and the other by $v_t$;
    For $j = t$ to 2 do
        Determine the set $Q$ of vertices of $G'$ of degree $k - (j-1)$;
        Let $Q_1$ (resp., $Q_2$) be the set of vertices from $Q$ which are adjacent to $v_{-t}, \ldots, v_{-j}$ (resp., to $v_j, \ldots, v_t$);
        If $|Q_1| = |Q_2| = 1$ and $Q_1 \cup Q_2 = Q$ then denote the element of $Q_1$ by $v_{-(j-1)}$ and the element of $Q_2$ by $v_{j-1}$ else STOP;
    For $j = t - 1$ to 1 do
        If not $v_1, \ldots, v_{t-j}$ are adjacent to $v_{-j}$ then STOP;
    Consider neighborhoods again in $G$ (and not in $G'$);
    For $j = 1$ to $t$ do
        If there is some $h$ with $-t - j + 1 \leq h \leq -t - 1$ such that $v_{-j}$ is not adjacent to $v_h$ then STOP;
        If there is some $h$ with $t + 1 \leq h \leq t + j - 1$ such that $v_j$ is not adjacent to $v_h$ then STOP;
        Let $Q_1 := N^1(v_{-j}) \setminus \{v_{-t-j+1}, \ldots, v_{t-j}\}$ and $Q_2 := N^1(v_j) \setminus \{v_{t+j-1}, \ldots, v_{j-t}\}$;
        If $|Q_1| = |Q_2| = 1$ and $|Q_1 \cup Q_2| = 2$ then denote the element of $Q_1$ by $v_{-t-j}$ and the element of $Q_2$ by $v_{t+j}$ else STOP;
    **Output:** The *bw–k* sequence $(v_{-k}, \ldots, v_{-1}, v_0, v_1, \ldots, v_k)$ iff $s$ is suspicious.
```

It is easy to verify that, in each STOP situation, the vertex s is not a suspicious vertex and that (16) holds if there is no STOP situation. From the procedure it follows that the *bw–k* sequence of a suspicious vertex s is unique up to inversion.

By Theorem 1, each vertex $s \in S$ that is not suspicious must be a start vertex or belong to the first or second neighborhood of a start vertex. Hence, if we find a vertex $s \in S$ that is not suspicious, we may proceed as in the case where k is odd: We test all vertices from $\{s\} \cup N^1(s) \cup N^2(s)$ for start vertices.

So it remains to study the case where we do not have a vertex from S which is not suspicious. We cannot test whether all vertices from S are start vertices because linear-time complexity cannot be reached. However, with the characterization of suspicious vertices we will be able to find in S in linear time a start vertex if one exists.

Let $P = (v_l, v_{l+1}, \ldots, v_r)$ be a path in G. It is said to be a *bw–k path* if $r - l \geq 2k$

and

$$N^1(v_i) = \{v_{i-t}, \ldots , v_{i+t}\} \setminus \{v_i\}, \qquad l + t \leq i \leq r - t.$$

Note that the *bw–k* sequence of a suspicious vertex is a *bw–k* path. For a *bw–k* path $P = (v_l, \ldots , v_r)$ let the first and second left and right parts be defined by

$$L^1(P) := (v_{l+t}, \ldots , v_{l+2t-1}), \quad L^2(P) := (v_l, \ldots , v_{l+t-1}),$$
$$R^1(P) := (v_{r-2t+1}, \ldots , v_{r-t}), \quad R^2(P) := (v_{r-t+1}, \ldots , v_r).$$

Further, let

$$L(P) := L^1(P) \cup L^2(P), \quad R(P) := R^1(P) \cup R^2(P), \quad M(P) := L(P) \cup R(P).$$

Clearly, the vertices $v_{l+2t}, \ldots , v_{r-2t}$ of P are suspicious vertices. Let $tToLeft(P) := v_{l+t-1}$ and $tToRight(P) := v_{r-t+1}$. For $u \notin P$ let $u||P := (u, v_l, \ldots , v_r)$ and $P||u := (v_l, \ldots , v_r, u)$. Given a *bw–k* path P it is easy to test whether it can be extended, as the following shows:

```
Procedure Path-extension
    Input: The bw–k path $P$;
    $StopLeft := false$;
    Repeat
        $x := tToLeft(P)$;
        If $x$ is not adjacent to all vertices of $L^2(P) \setminus \{x\}$ then $StopLeft := true$
        else
            $LeftU := N^1(x) \setminus L(P)$;
            If $LeftU \cap R^2(P) = \emptyset$ and $|LeftU| = 1$ with $LeftU = \{u\}$ then
            $P := u||P$ else $StopLeft := true$
        until $StopLeft$
    $StopRight := false$;
    Repeat
        $x := tToRight(P)$;
        If $x$ is not adjacent to all vertices of $R^2(P) \setminus \{x\}$ then $StopRight := true$
        else
            $RightU := N^1(x) \setminus R(P)$;
            If $RightU \cap L^2(P) = \emptyset$ and $|RightU| = 1$ with $RightU = \{u\}$ then
            $P := P||u$ else $StopRight := true$
        until $StopRight$;
    Output: The nonextendable bw–k path $P$.
```

Note that, under the general supposition that G is k-connected, e.g., for $LeftU \cap R^2(P) = \emptyset$ and $|LeftU| = 1$, the new path $u||P$ is indeed a *bw–k* path: Since $\deg(tToLeft(P)) \geq k$ this vertex $tToLeft(P)$ must be adjacent to all other vertices of $L^2(P)$. Obviously, the procedure **Path-extension** can be carried out in linear time. As in the procedure, for a *bw–k* path $P = (v_l, \ldots , v_r)$, let

$$LeftU := N^1(v_{l+t-1}) \setminus L(P),$$
$$RightU := N^1(v_{r-t+1}) \setminus R(P).$$

The next lemma contains the case where the *bw–k* path cannot be further extended because a "circle will be closed." Then the whole graph is included and is, by Lemma 7, a *bw–k* graph.

Lemma 6. *Let $G = (V, E)$ be k-connected and $P = (v_l, \dots, v_r)$ be a bw–k path. Then $LeftU \neq \emptyset$ and $RightU \neq \emptyset$. If $LeftU \subseteq R^2(P)$ or $RightU \subseteq L^2(P)$, then $V = \{v_l, \dots, v_r\}$.*

Proof. We consider only $LeftU$. We have $LeftU \neq \emptyset$ because otherwise $\deg(v_{l+t-1}) \leq (t-1) + t = k-1$, in contrast to (1). Let $LeftU \subseteq R^2(P)$. Let

$$A := \{v_{l+t-1}, \dots, v_{r-t}\}.$$

Obviously,

$$B(A) = \{v_l, \dots, v_{l+t-2}, v_{r-t+1}, \dots, v_r\}.$$

Since $|B(A)| = k-1$ we have, by Lemma 1, $|A| + |B(A)| = n$, i.e., $A \cup B(A) = V$. □

Lemma 7. *Let $P = (v_l, \dots, v_r)$ be a bw–k path in G. If $V = \{v_l, \dots, v_r\}$, then $f = (v_l, v_r, v_{l+1}, v_{r-1}, v_{l+2}, \dots)$ is a bw–k layout.*

Proof. Let $v_i v_j$ be an edge $(l \leq i < j \leq r)$. If $\{i, j\} \subseteq \{l, \dots, l+t-1, r-t+1, \dots, r\}$, clearly $|f(v_j) - f(v_i)| \leq 2t - 1 < k$. Otherwise, obviously $|j - i| \leq t$. Since $|f(v_{h+1}) - f(v_h)| \leq 2$ for $h = l, \dots, r-1$, we have

$$|f(v_j) - f(v_i)| \leq \sum_{h=i}^{j-1} |f(v_{h+1}) - f(v_h)| \leq 2t = k. \qquad \square$$

Of course, it is also possible that the *bw–k* path cannot be extended and that no circle will be closed.

Lemma 8. *Let $P = (v_l, \dots, v_r)$ be a nonextendable bw–k path in G such that $LeftU \not\subseteq R^2(P)$ and $RightU \not\subseteq L^2(P)$. Then the vertices $v_l, \dots, v_{l+2t-1}$ and $v_{r-2t+1}, \dots, v_r$ are not suspicious.*

Proof. Assume the contrary and let v_i be suspicious where w.l.o.g. $l \leq i \leq l+2t-1$. The neighborhood structure implies that $(v_l, \dots, v_i, \dots, v_{i+2t})$ is a subsequence (a right part) of the *bw–k* sequence of v_i (up to inversion). In particular,

$$N^1(v_{l+t-1}) = \{u, v_l, \dots, v_{l+2t-1}\} \setminus \{v_{l+t-1}\}$$

with some $u \notin \{v_l, \dots, v_{2l+t-1}\}$. By the supposition, $u \notin R^2(P)$; hence $u \notin P$. Consequently, $u||P$ is also a *bw–k* path, a contradiction to the supposition that P cannot be extended. □

Recall the sets A_j from (3). Some of them are uniquely determined by P and the start vertex.

Lemma 9. *Let G be k-connected and $P = (v_l, \dots, v_r)$ be a bw–k path. Let f be a bw–k layout of G and let $f(v_i) = 1$ for some i with $l + k \leq i \leq r - k$. Then, for each positive integer p with $l \leq i - pt$ and $i + pt \leq r$,*

$$A_{1+pk} = \{v_{i-pt}, v_{i-pt+1}, \dots, v_i, \dots, v_{i+pt-1}, v_{i+pt}\}, \tag{17}$$

$$\begin{aligned} &\{f(v_{i-pt}), \dots, f(v_{i-(p-1)t-1}), f(v_{i+(p-1)t+1}), \dots, f(v_{i+pt})\} \\ &\qquad = \{2 + (p-1)k, \dots, 1 + pk\}, \end{aligned} \tag{18}$$

$$f(v_{i-pt}) > \cdots > f(v_{i-(p-1)t-1}) \text{ and } f(v_{i+(p-1)t+1}) < \cdots < f(v_{i+pt}). \tag{19}$$

Proof. The statement can be formulated in the following analogous way: If v_i has label 1, then t vertices left of v_i and t vertices right of v_i on P get the labels $2, \ldots, 1+p$. This need not be done in an alternating way but in an increasing way going, on P, to the left of v_i and to the right of v_i. This is then similarly true for labels $2+p, \ldots, 1+2p$ and so on.

For the proof, we proceed by induction on p. Let $p = 1$. Then (17) and (18) are satisfied since $v_{i-t}, \ldots, v_{i-1}, v_{i+1}, \ldots, v_{i+t}$ are the neighbors of v_i. Assume, e.g., that for some g with $i-t \le g < g+1 \le i-1$, $f(v_g) < f(v_{g+1})$. Let $j := f(v_g)$. Then $B(A_j)$ contains t elements v_h with $h < g$, t elements v_h with $h > i$, and the element v_{g+1}, i.e., at least $k+1$ elements. This is a contradiction to Lemma 2. Thus (19) also is proved for $p = 1$. Now look at the step $p-1 \to p$. Since by supposition $2pt \le r-l \le n-1$, we have $2(p-1)t+1 \le n-k$ and hence $f(v) \le n-k$ for all $v \in A_{1+(p-1)k}$. The structure of P and Lemma 3 imply that

$$f(v_h) = \begin{cases} f(v_{h+t}) + k & \text{if } i-pt \le h \le i-(p-1)t-1, \\ f(v_{h-t}) + k & \text{if } i+(p-1)t+1 \le h \le i+pt, \end{cases}$$

and (17)–(19) follow immediately from the induction hypothesis. □

LEMMA 10. *Let G be k-connected and $P = (v_l, \ldots, v_r)$ be a bw–k path. If $|LeftU| \ge 2$ (resp., $|RightU| \ge 2$), then the vertices v_i with $l+k \le i \le \frac{l+r}{2} - t$ (resp., $\frac{l+r}{2} + t \le i \le r-k$) are not start vertices.*

Proof. We consider only $LeftU$. Assume the contrary and let, for some i with $l+k \le i \le \frac{l+r}{2} - t$, the vertex v_i be a start vertex with a corresponding *bw–k* layout f. Let p be the largest integer such that $l+t-1 < i-(p-1)t$. Then $l \le i-pt$, which implies $i+pt \le 2i-l \le r-2t$. Let $j := f(v_{l+t-1})$. From Lemma 9 (note the remark in the beginning of the proof) it follows that

$$A_j = \{v_{l+t-1}, v_{l+t}, \ldots, v_{l+t+j-2}\}$$

and that

$$l+t+j-2 \le i+pt \le r-2t.$$

Consequently,

$$B(A_j) = \{v_l, \ldots, v_{l+t-2}\} \cup LeftU \cup \{v_{l+t+j-1}, \ldots, v_{l+2t+j-2}\}.$$

Since $LeftU \cap \{v_l, \ldots, v_{r-t}\} = \emptyset$, it follows that

$$|B(A_j)| \ge (t-1) + 2 + t = k+1,$$

a contradiction to Lemma 2. □

Recall that, for a *bw–k* path $(v_l, \ldots, v_r)$, $M(P) = \{v_l, \ldots, v_{l+k-1}, v_{r-k+1}, \ldots, v_r\}$.

LEMMA 11. *Let G be a k-connected bw–k graph and let $P = (v_l, \ldots, v_r)$ be a nonextendable bw–k path. Suppose that no vertex of $M(P)$ is a start vertex. Let $LeftU \not\subseteq R^2(P)$ and $RightU \not\subseteq L^2(P)$. Further, let $LeftU \cap R^2(P) \neq \emptyset$ or $RightU \cap L^2(P) \neq \emptyset$. Then one of the vertices v_i with $\frac{l+r}{2} - t < i < \frac{l+r}{2} + t$ is a start vertex.*

Proof. Assume the contrary. By the supposition and Lemma 10, no vertex of P is a start vertex (note that, e.g., $LeftU \cap R^2(P) \neq \emptyset$ implies $|LeftU| \ge 2$ and that, e.g., for $RightU \cap L^2(P) = \emptyset$, $|RightU| \ge 2$ since P cannot be extended). Let f be a *bw–k* layout of G and let, w.l.o.g., $v_h \in LeftU \cap R^2(P)$, $r-t+1 \le h \le r$. Let $j := f(v_{l+k})$.

Since all neighbors of v_{l+k} belong to P, v_{l+k} is not adjacent to the vertex with f-value 1. By Lemma 3, $k+1 \leq j \leq n-k$. Moreover, by Lemma 3 v_{l+k} is adjacent to vertices with f-values $j-k$ and $j+k$. It is easy to see that by the structure of P these vertices are v_{l+t} and v_{l+k+t} (otherwise one could find two adjacent vertices in $\{v_{l+t}, \dots, v_{l+k+t}\}$ with absolute f-difference greater than k). Let, w.l.o.g., $f(v_{l+t}) = j-k$ and $f(v_{l+k+t}) = j+k$. Again the structure of P implies that

$$f(v_i) \begin{cases} < j & \text{if } i \in \{l+t, \dots, l+k-1\}, \\ > j & \text{if } i \in \{l+t+1, \dots, l+k+t\}. \end{cases}$$

By Lemma 3, v_{l+k-1} must have a neighbor such that their f-difference is k. This can only be v_{l+t-1}, i.e.,

$$f(v_{l+t-1}) = f(v_{l+k-1}) - k < j-k.$$

Let ρ be that integer from $\{1, \dots, t\}$ for which $l+t+\rho \equiv h \mod t$. Obviously, $h = l+t+\rho+pt$, where $p \geq 1$. By Lemma 3 (and the fact that $f(v_i) \neq 1$ for $i \in \{l, \dots, r\}$),

$$f(v_h) = f(v_{l+t+\rho}) + pk > j+k.$$

Since v_{l+t-1} and v_h are adjacent we have a contradiction: $f(v_h) - f(v_{l+t-1}) > (j+k) - (j-k) = 2k$. □

LEMMA 12. *Let G be k-connected and $P = (v_l, \dots, v_r)$ be a bw–k path. If $|LeftU \backslash R^2(P)| \geq 2$ and $|RightU \backslash L^2(P)| \geq 2$, then the vertices v_i with $l+k \leq i \leq r-k$ are not start vertices.*

Proof. The proof is similar to the proof of Lemma 10. Assume that for some i with $l+k \leq i \leq r-k$ the vertex v_i is a start vertex with a corresponding *bw–k* layout f. From Lemma 9 it follows that there is some j such that

$$A_j = \{v_{l+t-1}, \dots, v_{l+t+j-2}\} \text{ and } l+t+j-2 \leq r-t+1$$

or

$$A_j = \{v_{r-t-j+2}, \dots, v_{r-t+1}\} \text{ and } r-t-j+2 \geq l+t-1.$$

It is easy to see that $|B(A_j)| \geq k+1$, a contradiction to Lemma 2. □

Now we may present the algorithm. For a *bw–k* path $P = (v_l, \dots, v_r)$ let

$$\nu(P) := v_{\lfloor (l+r)/2 \rfloor}.$$

Algorithm *bw–k*(even)

```
    Input: A positive even integer $k \geq 2$, a $k$-connected graph $G = (V, E)$ with
    $|V| \geq 4k+1$ and $\Delta(G) \leq 2k$;
    Determine $S := \{v \in V : |N^1(v)| = |N^2(v)| = k\}$; Let $T := S$;
    If $|S| \leq 1$ then STOP — $G$ is not a bw–k graph;
    repeat
        Pick a vertex $s_0 \in S$;
        Test with Suspicious-vertex whether $s_0$ is suspicious;
        If $s_0$ is not suspicious then
            for all $s \in (\{s_0\} \cup N^1(s_0) \cup N^2(s_0)) \cap S$ do Start-vertex($s$) and
            STOP
```

```
else
    Let $P$ be the $bw$–$k$ sequence of $s_0$;
    Extend $P$ with **Path-extension** to a nonextendable $bw$–$k$ path $P$;
    If $P$ contains all vertices of $G$ then Output of $f$ according to Lemma
    7 and STOP
    else
        If $LeftU \cap R^2(P) \neq \emptyset$ or $RightU \cap L^2(P) \neq \emptyset$ then
            If $T \cap M(P) \neq \emptyset$ then
                Pick some element $s_1$ from $T \cap M(P)$;
                for all $s \in (\{s_1\} \cup N^1(s_1) \cup N^2(s_1)) \cap S$ do **Start-vertex**($s$)
                and STOP
            else
                for all $s \in \{\nu(P)\} \cup N^1(\nu(P))$ do **Start-vertex**($s$)
                and STOP
        else delete the vertices of $P$ not belonging to $M(P)$ from $S$
until $S = \emptyset$.
**Output:** A $bw$–$k$ layout of $G$ iff $G$ is a $bw$–$k$ graph.
```

THEOREM 2. *For even k, Algorithm* ***bw–k*(even)** *decides in linear time whether the k-connected graph G with $\Delta(G) \leq 2k$ is a bw–k graph, and in the positive case, it provides a bw–k layout.*

Proof. The first STOP follows from Lemma 4. After Procedure **Suspicious-vertex** we already mentioned that, for a nonsuspicious vertex s_0, by Theorem 1, a vertex s from $\{s_0\} \cup N^1(s_0) \cup N^2(s_0)$ must be a start vertex if G is a bw–k graph. This explains the second STOP. The third STOP follows from Lemma 7. Thus suppose that P does not contain all vertices of G. If $LeftU \cap R^2(P) = \emptyset$ and $RightU \cap L^2(P) = \emptyset$, we have $|LeftU| \geq 2$ and $|RightU| \geq 2$ since P cannot be extended, and we have $LeftU \neq \emptyset$, $RightU \neq \emptyset$ by Lemma 6. By Lemma 12, the vertices from P not belonging to $M(P)$ cannot be start vertices and we may delete them from the set of all possible start vertices. Thus let $LeftU \cap R^2(P) \neq \emptyset$ or $RightU \cap L^2(P) \neq \emptyset$. According to Lemma 6, we have $LeftU \not\subseteq R^2(P)$ and $RightU \not\subseteq L^2(P)$. In view of Lemma 8 the vertices of $M(P)$ are not suspicious. Hence, if one of them, say s_1, belongs to the original set S (here T), then again by Theorem 1 a vertex from $\{s_1\} \cup N^1(s_1) \cup N^2(s_1)$ must be a start vertex if G is a bw–k graph and the fourth STOP follows. Otherwise Lemma 11 can be applied to verify the fifth STOP.

The deletion of m $(= r - l + 1 - 2k)$ suspicious vertices obviously needs only $O(m)$ steps. Thus, if there is not an earlier STOP, the deletion of all suspicious vertices is accomplished no later than after $O(|S|) = O(n)$ steps. Then the algorithm stops after $O(n)$ steps since **Start-vertex** has linear-time complexity. Consequently, the whole algorithm has linear-time complexity. ☐

Acknowledgment. The authors are grateful to an anonymous referee whose comments helped to improve the presentation of the paper.

REFERENCES

[1] H. L. BODLAENDER, M. R. FELLOWS, AND M. T. HALLETT, *Beyond NP-completeness for problems of bounded width: Hardness for the W hierarchy*, in Proceedings of the Twenty-sixth Annual ACM Symposium on the Theory of Computing, ACM, New York, 1994, pp. 449–458.

[2] F. R. K. CHUNG, *Labelings of graphs*, in Selected Topics in Graph Theory, Vol. 3, L. W. Beineke and R. J. Wilson, eds., Academic Press, London, 1988, pp. 151–168.

[3] R. G. Downey and M. R. Fellows, *Parameterized Complexity*, Monographs in Computer Science, Springer-Verlag, New York, 1998.

[4] K. Engel and S. Guttmann, *A Structural Characterization and a New Linear-Time Algorithm for the Recognition of Bandwidth-*2 *Graphs*, Preprint 98/2, Universität Rostock, Fachbereich Mathematik, Rostock, Germany, 1998.

[5] M. R. Garey, R. L. Graham, D. S. Johnson, and D. E. Knuth, *Complexity results for bandwidth minimization*, SIAM J. Appl. Math., 34 (1978), pp. 477–495.

[6] E. M. Gurari and I. H. Sudborough, *Improved dynamic programming algorithms for bandwidth minimization and the mincut linear arrangement problem*, J. Algorithms, 5 (1984), pp. 531–546.

[7] F. Makedon, D. Sheinwald, and Y. Wolfsthal, *A simple linear-time algorithm for the recognition of bandwidth-*2 *biconnected graphs*, Inform. Process. Lett., 46 (1993), pp. 103–107.

[8] J. B. Saxe, *Dynamic-programming algorithms for recognizing small-bandwidth graphs in polynomial time*, SIAM J. Alg. Disc. Meth., 1 (1980), pp. 363–369.

SIAM J. DISCRETE MATH.
Vol. 16, No. 2, pp. 313–322

MAKESPAN MINIMIZATION IN NO-WAIT FLOW SHOPS: A POLYNOMIAL TIME APPROXIMATION SCHEME[*]

M. SVIRIDENKO[†]

Abstract. We investigate the approximability of a no-wait permutation flow shop scheduling problem under the makespan criterion. We present a polynomial time approximation scheme (PTAS) for the problem on any *fixed* number of machines.

Key words. approximation algorithms, shop scheduling

AMS subject classifications. 90B35, 68W25

PII. S0895480100370803

1. Introduction. Problem statement. A *job shop* is a multistage production process with the property that all jobs have to pass through several stages. There are n jobs J_j, with $j = 1, \dots, n$, where each job J_j is a chain of m_j operations $O_{1j}, \dots, O_{m_j j}$. Every operation O_{ij} is preassigned to one of m *stages* $M_1, \dots, M_m$ of the production process. The operation O_{ij} has to be processed for p_{ij} time units at its stage; the value p_{ij} is called its *processing time* or its *length*. We will consider a basic model where there is exactly one machine available for each stage; to simplify the presentation we will identify the stage with the corresponding machine. In a feasible schedule for the n jobs, at any moment in time every job is processed by at most one machine and every machine executes at most one job. For each job J_j, operation $O_{i-1,j}$ always is processed before operation O_{ij}, and each operation is processed without interruption on the machine to which it was assigned. A *flow shop* is a special case of the job shop where each job has exactly one operation in each stage, and where all jobs pass through the stages in the same order $M_1 \to M_2 \to \cdots \to M_m$. In an *open shop* the ordering of the operations in a job is not fixed and may be chosen by the scheduler.

In this paper, we are mainly interested in shop problems under the *no-wait constraint*. In such a no-wait shop, there is no waiting time allowed between the execution of consecutive operations of the same job. Once a job has been started, it has to be processed without interruption, operation by operation, until it is completed. In a no-wait flow shop instance without operations of length zero, any feasible schedule is a *permutation schedule*, i.e., a schedule in which each machine processes the jobs in the same order. In the *no-wait permutation flow shop* problem, only permutation schedules are feasible schedules. Our goal is to find a feasible schedule that minimizes the *makespan* (or *length*) $C_{\max}$ of the schedule, i.e., the maximum completion time among all jobs. The minimum makespan among all feasible schedules is denoted by $C^*_{\max}$.

[*] Received by the editors April 10, 2000; accepted for publication (in revised form) September 29, 2002; published electronically February 20, 2003. Portions of the joint preliminary version of this paper are reprinted here, with permission, from M. Sviridenko and G.J. Woeginger, *Approximability and in-approximability results for no-wait shop scheduling*, in Proceedings of the 41st Annual Symposium on the Foundation of Computer Science, pp. 116–125. © 2000 IEEE. This joint preliminary version contains some additional nonapproximability results for no-wait shop scheduling problems.

http://www.siam.org/journals/sidma/16-2/37080.html

[†] IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598 (sviri@us.ibm.com).

Complexity of shop problems. The computational complexity of the classical shop problems (without the no-wait constraint) is easy to summarize: They are polynomially solvable on two machines, and they are $\mathcal{NP}$-hard on three or more machines; see, e.g., Lawler et al. [11]. For no-wait shops, the situation is more interesting. Sahni and Cho [19] proved that the no-wait job shop and the no-wait open shop problems are strongly $\mathcal{NP}$-hard even if there are only two stages and if each job consists of only two operations. The no-wait permutation flow shop problem can be formulated as an asymmetric traveling salesman problem (ATSP); see, e.g., Piehler [15] and Wismer [25]. For two machines, the distance matrix of this ATSP has a very special combinatorial structure, and the famous subtour patching technique of Gilmore and Gomory [5] yields an $O(n \log n)$ time algorithm for the two-machine no-wait flow shop. Röck [17] proves that the three-machine no-wait flow shop is strongly $\mathcal{NP}$-hard, refining the previous complexity result by Papadimitriou and Kanellakis [12] for four machines. Hall and Sriskandarajah [8] provide a thorough survey of complexity and algorithms for no-wait scheduling.

Approximability of shop problems. We say that an approximation algorithm has *performance ratio* or *worst case ratio* ρ for some real $\rho > 1$ if it always delivers a solution with makespan at most $\rho C^*_{\max}$. Such an approximation algorithm is then called a *ρ-approximation* algorithm. A family of polynomial time $(1 + \varepsilon)$-approximation algorithms over all $\varepsilon > 0$ is called a *polynomial time approximation scheme* (PTAS).

The approximability of the classical shop problems (without the no-wait constraint) is fairly well understood: If the number of machines is a fixed value that is not part of the input, then the flow shop [7], the open shop [20], and the job shop [9] possess a PTAS. On the other hand, if the number of machines is part of the input, then none of the three shop problems has a PTAS unless $\mathcal{P} = \mathcal{NP}$ [24].

Prior to our work, only a few results were known on the approximability of the no-wait shop problems: For all shop problems on m machines, sequencing the jobs in arbitrary order yields a (trivial) polynomial time m-approximation algorithm. Röck and Schmidt [18] improve on this for the no-wait flow shop and give an $\lceil m/2 \rceil$-approximation algorithm. Papadimitriou and Kanellakis [12], Glass, Gupta, and Potts [6], and Sidney, Potts, and Sriskandarajah [21] study various generalizations and modifications of the no-wait flow shop problem on two machines. For all these generalizations the authors manage to design approximation algorithms, with performance guarantees strictly better than two, by building on the algorithm of Gilmore and Gomory [5]. Agnetis [1] introduces an approximation algorithm for the no-wait flow shop with only a small number of *distinct* job-types; as the number of jobs in every job-type grows, the performance guarantee of this algorithm tends to one. Sidney and Sriskandarajah [22] obtain a 3/2-approximation algorithm for the two-machine no-wait open shop problem. The joint preliminary version of this paper [23] contains several nonapproximability results due to Woeginger. He proved that the no-wait job shop problem on three machines with at most three operations per job, and the no-wait job shop problem on two machines with at most five operations per job, do not have a PTAS unless $\mathcal{P} = \mathcal{NP}$.

Results and organization of this paper. We design a PTAS for the no-wait permutation flow shop problem when the number m of machines is fixed. This result first uses several job partition and rounding steps and then exploits the connection of the no-wait flow shop to the ATSP. In section 2 we recall and discuss this connection between a no-wait flow shop and the ATSP. In section 3 we derive the PTAS. Some of our rounding and job partition steps seem to be very close to the rounding and job partition steps in the PTASs for the classical shop problems [7, 9, 20], but our technique cannot be generalized to the no-wait job shop problem because of the negative result due to Woeginger [23]. The paper concludes with the statement of several open problems in section 4.

2. The no-wait permutation flow shop and the ATSP. It is well known (see, e.g., Piehler [15] or Wismer [25]) that the no-wait permutation flow shop problem can be modeled as a special case of the ATSP with the triangle inequality: We add a *dummy* job J_0 with zero processing times on all machines to a given flow shop instance. By G we denote the complete, arc-weighted, directed graph with vertex set $\{J_0, J_1, \ldots, J_n\}$ and with the following weights (or *distances* or *lengths*) d_{qj} on the arc from job J_q to J_j. We stress the fact that, in general, $d_{qj} \neq d_{jq}$:

(2.1)
$$d_{qj} = \max_{i=1,\ldots,m} \left\{ \sum_{k=1}^{i} p_{kq} + \sum_{k=i}^{m} p_{kj} \right\} - \sum_{k=1}^{m} p_{iq} = \max_{i=1,\ldots,m} \left\{ \sum_{k=i}^{m} p_{kj} - \sum_{k=i+1}^{m} p_{kq} \right\}.$$

The intuition behind the definition of the distances in (2.1) is the following. Assume that in some schedule job J_q completes at time t, and that job J_q is followed by job J_j, without unnecessary idle time between the two jobs. Then J_j completes at time $t + d_{qj}$. With this it is easy to see that every feasible permutation schedule of the no-wait flow shop problem corresponds to a directed Hamiltonian cycle $\mathcal{C}$ in the digraph G such that the makespan of the schedule equals the length of $\mathcal{C}$. Conversely, if we delete the ingoing arc of vertex J_0 from some Hamiltonian cycle $\mathcal{C}$ in G, then the resulting Hamiltonian path corresponds to a feasible permutation schedule with the same length.

The following observations on the distances d_{qj} are straightforward to verify.

OBSERVATION 2.1. *For every job J_j, denote by $\ell_j = \sum_{k=1}^{m} p_{kj}$ its overall length.*

(i) *For all $0 \le j, q \le n$, $\ell_j - \ell_q \le d_{qj} \le \ell_j$ holds.*

(ii) *For all $0 \le j, k, q \le n$, $d_{qj} \le d_{qk} + d_{kj}$; i.e., the distances d_{qj} fulfill the triangle inequality.*

(iii) *If one of the values p_{ij} changes to $p_{ij} + \Delta$, then the length of any ATSP tour (and the makespan of the corresponding feasible schedule) changes by at most $\pm\Delta$.*

Because of this correspondence between permutation schedules and ATSP tours, the result of Frieze, Galbiati, and Maffioli [4] on the ATSP with triangle inequality yields an $O(\log n)$-approximation algorithm for the no-wait permutation flow shop problem. Recently, Carr and Vempala [2] gave some theoretical evidence for the existence of a 4/3-approximation algorithm for the ATSP with triangle inequality. Of course, such a result would immediately yield a 4/3-approximation algorithm for the no-wait permutation flow shop on an arbitrary number of machines. We remark that the strongest known negative result for the general ATSP with the triangle inequality is due to Papadimitriou and Vempala [14]. They prove that unless $\mathcal{P} = \mathcal{NP}$, the ATSP with triangle inequality cannot have a polynomial time approximation algorithm with performance guarantee better than 41/40. However, this negative result does not seem to carry over to the no-wait flow shop.

3. Approximability of the no-wait flow shop. Throughout this section we consider an instance I of the no-wait permutation flow shop problem, where the number m of machines is a fixed constant and not part of the input. By $\ell_j = \sum_{i=1}^{m} p_{ij}$ we denote the total length of job J_j. Let $L_i = \sum_{j=1}^{n} p_{ij}$ be the load of machine M_i, and let $L_{\max} = \max_i L_i$ be the maximum machine load. Clearly, $L_{\max} \le C^*_{\max}$.

Let $\varepsilon > 0$ be a fixed precision parameter. Our goal is to find a near optimal schedule, for instance, I whose makespan is at most $(1 + \text{const} \cdot \varepsilon) C^*_{\max}$ for some fixed positive constant const that depends only on m. Clearly, this will yield the PTAS. We will use $\log x$ to denote the logarithm base $1 + \varepsilon$ of x. By α we denote a rational number with $\varepsilon^{m/\varepsilon} \le \alpha \le \varepsilon$ whose exact value will be fixed below. From now on we will assume that the number n of jobs is sufficiently large to satisfy

$$(2^{1/m} - 1) \log n \ \ge \ 1 + \log(m/\varepsilon) \tag{3.1}$$

and

$$\alpha n \ \ge \ \log^m n. \tag{3.2}$$

If n does not fulfill (3.1) and (3.2), then it is bounded by a constant in m and ε; such an instance of constant size can be solved in constant time by global enumeration. We partition the set of jobs into three subsets as follows:

$$\begin{aligned} B &= \{J_j \ \mid \ \alpha L_{\max}/\log^m n \ \le \ \ell_j\}, \\ M &= \{J_j \ \mid \ \varepsilon \cdot \alpha L_{\max}/\log^m n \ < \ \ell_j \ < \ \alpha L_{\max}/\log^m n\}, \\ S &= \{J_j \ \mid \ \ell_j \ \le \ \varepsilon \cdot \alpha L_{\max}/\log^m n\}. \end{aligned}$$

The jobs in B are called *big jobs*, the jobs in M are called *medium jobs*, and the jobs in S are called *small jobs*. For the operations of big, medium, and small jobs we use a similar notation: the operations of big jobs are called *big operations*, while operations of medium and small jobs are called *medium* and *small operations*, respectively, independently of their actual sizes. Since $\sum_{j=1}^{n} \ell_j \le m L_{\max}$, the number of big jobs is upper bounded as

$$|B| \ \le \ \frac{m}{\alpha} \log^m n \ \le \ \varepsilon^{-m/\varepsilon} m \log^m n. \tag{3.3}$$

Sevastianov and Woeginger [20] show that the value α can be chosen so that

$$\sum_{J_j \in M} \ell_j \ \le \ \varepsilon L_{\max}. \tag{3.4}$$

This is done as follows. Consider the sets $M(k)$ of medium jobs with respect to $\alpha = \varepsilon^k$, where k is some positive integer. Note that for $k \ne k'$ the two sets $M(k)$ and $M(k')$ are disjoint. Since the total length of all jobs is at most $m L_{\max}$, there exists a value $k_* \le m/\varepsilon$ for which $M = M(k_*)$ satisfies inequality (3.4). We set $\alpha = \varepsilon^{k_*}$. Finally, we define

$$\beta \ \doteq \ \left(m^2 \log^m n\right) \ / \ (\varepsilon\alpha)\,. \tag{3.5}$$

Starting from the flow shop instance $I = I^{(0)}$, we will now define a sequence of instances $I^{(1)}$, $I^{(2)}$, $I^{(3)}$, $I^{(4)}$. The instance $I^{(x+1)}$ always is a rounded and slightly simplified version of instance $I^{(x)}$. In instance $I^{(x)}$, the processing times are $p_{ij}^{(x)}$, the optimal makespan is $C_{\max}^{(x)}$, the digraph for the underlying ATSP is $G^{(x)}$, and so on. In order to get a near optimal schedule, for instance, $I^{(x)}$, it will always be sufficient to get a near optimal schedule for the simplified instance $I^{(x+1)}$. Hence, by constructing a PTAS for $I^{(4)}$ we will establish the existence of the desired PTAS for the no-wait permutation flow shop on a fixed number of machines.

3.1. How to round the instance. In the first rounding step, we remove all medium jobs from I and thus produce instance $I^{(1)}$. If we have a near optimal schedule with makespan $C_{\max}^{APX}$ for the big and small jobs in instance $I^{(1)}$, then we may append the medium jobs in arbitrary order at the end of this schedule. By (3.4), this yields a schedule with makespan at most $C_{\max}^{APX} + \varepsilon L_{\max}$ for $\mathcal{J}$. Hence, to build a PTAS for the original problem, it is sufficient to get a near optimal schedule for $I^{(1)}$.

In the second rounding step, we round the processing times of all big operations with processing time smaller than $L_{\max}/\beta$ up to $L_{\max}/\beta$. This yields instance $I^{(2)}$. The number of rounded operations is at most $m|B| \leq \varepsilon\beta$, and by Observation 2.1(iii) each rounded operation can increase the length of an optimal Hamiltonian cycle in the underlying digraph by at most $L_{\max}/\beta$. Therefore the length $C_{\max}^{(2)}$ of an optimal Hamiltonian cycle (and the length of an optimal no-wait schedule) in $G^{(2)}$ fulfills $C_{\max}^{(1)} - \varepsilon L_{\max} \leq C_{\max}^{(2)} \leq C_{\max}^{(1)} + \varepsilon L_{\max}$. Hence, in order to get a near optimal schedule for $I^{(1)}$, it is sufficient to get one for $I^{(2)}$. Note that in $I^{(2)}$ the longest and the shortest big operation are at most a factor of β away from each other.

In the third rounding step, we produce instance $I^{(3)}$ by rounding up to $\varepsilon L_{\max}/mn$ all the processing times of small operations that are smaller than $\varepsilon L_{\max}/(mn)$ (except the processing times of operations of the dummy job J_0). The number of rounded operations is at most mn, and by Observation 2.1(iii) each rounded operation can increase the length of an optimal Hamiltonian cycle in the underlying digraph by at most $\varepsilon L_{\max}/(mn)$. Hence, the optimal Hamiltonian cycle satisfies $C_{\max}^{(2)} - \varepsilon L_{\max} \leq C_{\max}^{(3)} \leq C_{\max}^{(2)} + \varepsilon L_{\max}$, and to get the PTAS it is sufficient to find a near optimal schedule for $I^{(3)}$. Note that in the new instance $I^{(3)}$ all processing times of small operations are rational numbers between $\varepsilon L_{\max}/mn$ and $L_{\max}$, and hence their minimum and maximum are at most a factor of mn/ε away from each other. We denote by $L_{\max}^{(3)}$ the maximum machine load in $I^{(3)}$.

In the fourth and last rounding step, we round all processing times up to the next integer power of $1+\varepsilon$. This results in instance $I^{(4)}$. The rounding adds at most $\varepsilon L_{\max}^{(3)}$ to the load of any machine. By Observation 2.1(iii) this changes the optimal makespan by at most $m\varepsilon L_{\max}^{(3)}$, and we have $C_{\max}^{(3)} - m\varepsilon L_{\max}^{(3)} \leq C_{\max}^{(4)} \leq C_{\max}^{(3)} + m\varepsilon L_{\max}^{(3)}$. Once again we conclude that in order to get a near optimal schedule for $I^{(3)}$, it is sufficient to get a near optimal schedule for $I^{(4)}$.

We say that two jobs J_j and J_k are *of the same job-type* if $p_{ij} = p_{ik}$ holds for $1 \leq i \leq m$. Such a job-type is represented by an m-dimensional vector $(p_{1j}, \ldots, p_{mj})$ of processing times. Analogously to the *big* and *small* jobs, we distinguish between *big* and *small* job-types. The following lemma will be useful in subsequent sections.

Lemma 3.1. *The rounded no-wait flow shop instance $I^{(4)}$ satisfies the following properties:*

(I1) *The number g of different big job-types is at most $c_1 \log^m \log n$, where the constant c_1 depends only on m and ε.*

(I2) *The number f of different small job-types is at most $2\log^m n$.*

(I3) *Small jobs have length $\leq 2\alpha(1+\varepsilon)\varepsilon L_{\max}/\log^m n$.*

Proof. (I1) Since in $I^{(3)}$ the longest and the shortest big operation are at most a factor of β away from each other, instance $I^{(4)}$ has at most $1 + \log\beta$ different processing times of big operations. Hence, there are at most $(1+\log\beta)^m$ different big job-types, and this number is $O(\log^m \log n)$.

(I2) Since in $I^{(3)}$ the longest and the shortest small operation are at most a factor of mn/ε away from each other, instance $I^{(4)}$ has at most $1+\log(mn/\varepsilon)$ different processing times of small operations. Including the dummy job J_0, this yields at most $1 + (1 + \log(mn/\varepsilon))^m$ small job-types. Because of inequality (3.1), this number is bounded by $2\log^m n$.

(I3) A small job J_j in the original instance I has length $\ell_j \le \varepsilon\alpha L_{\max}/\log^m n$. The first and the second rounding step do not touch this job; the third rounding step adds at most $\varepsilon L_{\max}/(mn)$ to each of the m operations; the fourth rounding step multiplies the length by a factor of at most $1+\varepsilon$. To summarize, $\ell_j^{(4)} \le (1+\varepsilon)(\varepsilon\alpha L_{\max}/\log^m n + \varepsilon L_{\max}/n)$. Because of inequality (3.2), this value is bounded by $2\alpha(1+\varepsilon)\varepsilon L_{\max}/\log^m n$. □

3.2. How to use the ATSP formulation. The instance $I^{(4)}$ contains only a small number of different job-types, and this structure carries over to the underlying digraph $G^{(4)}$ for the ATSP formulation. Let $1,\dots,f$ and $1,\dots,g$ be enumerations of the small and the big job-types that have at least one job in $I^{(4)}$; in this enumeration the dummy job J_0 forms its own small job-type. Let $s_1,\dots,s_f$ and $b_1,\dots,b_g$ denote the numbers of jobs in the corresponding small and big job-types. Note that $|S| = \sum_{i=1}^f s_i$ and that $|B| = \sum_{i=1}^g b_i$.

The corresponding traveling salesman problem becomes a special case of the so-called *many-visits-to-few-cities* ATSP (cf. Cosmadakis and Papadimitriou [3]). An instance to this ATSP is specified as follows: There are f small cities $S' = \{vs_1,\dots,vs_f\}$ that correspond to small job-types and g big cities $B' = \{vb_1,\dots,vb_g\}$ that correspond to big job-types. The distances d_{ij} between two cities are defined as in (2.1) and thus yield an $(f+g)\times(f+g)$ distance matrix D. The matrix D is not necessarily symmetric, nor must it have zeros on the diagonal elements. Finally, there are $f+g$ positive integers $s_1,\dots,s_f,b_1,\dots,b_g$. The goal is to find the shortest closed walk that visits every small city vs_i $(i=1,\dots,f)$ exactly s_i times and every big city vb_j $(j=1,\dots,g)$ exactly b_j times. Note that the same city may be visited several times in a row. The running time of the algorithm in [3] for the many-visits-to-few-cities ATSP grows exponentially in the number of cities. Therefore we get only a superpolynomial running time if we apply this algorithm directly to our situation.

In the following, we will show that the above defined special case of the many-visits-to-few-cities ATSP possesses a PTAS. It is convenient to formulate this ATSP via *Eulerian* subgraphs: For a given $(f+g)\times(f+g)$ distance matrix D for the vertex set $S'\cup B'$, and for $f+g$ positive integers $s_1,\dots,s_f,b_1,\dots,b_g$, find the minimum length Eulerian multigraph on $S'\cup B'$ such that the corresponding vertices have in-degrees $s_1,\dots,s_f,b_1,\dots,b_g$. We recall that a multigraph is Eulerian if and only if it is *strongly connected* and *balanced* (i.e., each vertex has equal in-degree and out-degree). The following lemmas show that it is easy to find a multigraph that is *almost* Eulerian and that is *almost* of minimum length.

LEMMA 3.2. *There exists a balanced multigraph G^* on the vertex set $S'\cup B'$ with in-degrees $(s_1,\dots,s_k,b_1,\dots,b_t)$ that satisfies the following three properties:*

(G1) *S' and B' are connected by exactly two arcs, one from B' to S' and one from S' to B'; such arcs will be called* crossing *arcs.*

(G2) *All vertices of the set B' are in the same strongly connected component of G^*.*

(G3) *The total length of the multigraph G^* is $T \le C_{\max}^{(4)} + 4m(1+\varepsilon)\varepsilon L_{\max}$.*

Proof. Fix an optimal tour and the corresponding optimal Eulerian multigraph of length $C_{\max}^{(4)}$ for the problem. Note that the number of crossing arcs that go from B' to S' equals the number of crossing arcs that go from S' to B'. Moreover, these two types of crossing arcs are alternating along the optimal tour. We define for every arc from B' to S' its *partner* arc to be the next crossing arc when moving along the tour.

As long as the multigraph contains more than two crossing arcs, we repeat the following swapping step: Consider a crossing arc (vb_i, vs_j) from B' to S' in the tour, and let (vs_k, vb_l) be its partner arc. Delete these two crossing arcs from the tour and add two new arcs (vb_i, vb_l) and (vs_l, vs_j). By the construction, the resulting multigraph is again balanced and has the right in-degrees. By using the inequalities from Observation 2.1(i), we estimate the length of the first new arc (vb_i, vb_h) by $d_{vb_i,vb_h} \leq \ell_{vb_h} \leq d_{vs_l,vb_h} + \ell_{vs_l}$. The length of the second new arc (vs_l, vs_j) fulfills $d_{vs_l,vs_j} \leq \ell_{vs_j}$. Hence, the length of the two new arcs is at most $\ell_{vs_l} + \ell_{vs_j}$ more than the length of the two crossing arcs that were removed, and the swapping step increases the length of the multigraph by at most $2\max_{1 \leq i \leq f} \ell_{vs_i}$. By statement (I3) in Lemma 3.1, this amount is at most $4\alpha(1+\varepsilon)\varepsilon L_{\max}/\log^m n$.

Eventually, we will end up with a multigraph G^* with exactly two crossing arcs as required in (G1). Clearly, this multigraph G^* is balanced and obeys the in-degrees. Since in the original multigraph the vertex set B' was strongly connected and since the swapping steps only introduce shortcuts into the connecting paths between big cities, the graph G^* also satisfies property (G2). Finally, the number of swapping steps is bounded by the total in-degree of the vertex set B'. This number amounts to $|B|$ and can be bounded as in (3.3). Hence, the total increase caused by all swaps is at most $4m(1+\varepsilon)\varepsilon L_{\max}$ as required in property (G3). □

LEMMA 3.3. *For a given degree sequence $b = (b_1, \ldots, b_g)$ for the big cities, a given starting city vb_k and a given final city vb_ℓ, we can compute in polynomial time the shortest directed path from vb_k to vb_ℓ that visits city vb_j exactly b_j times, $j = 1, \ldots, g$.*

For technical reasons, we count the starting point of the directed path also as a visit to vb_k.

Proof. We follow the dynamic programming approach of Psaraftis [16] and Cosmadakis and Papadimitriou [3]. For each degree sequence $a = (a_1, \ldots, a_g)$ with $0 \leq a_i \leq b_i$ and each city vb_q $(1 \leq q \leq g)$, let $C(a; q)$ be the length of the shortest path from city vb_k to city vb_q that visits every city vb_i exactly a_i times. Then $C(a; q)$ satisfies the recurrence

$$C(a_1, \ldots, a_g; q) = \min_{1 \leq i \leq g} \left\{ C(a_1, \ldots, a_{i-1}, a_i - 1, a_{i+1}, \ldots, a_g; i) + d_{vb_i, vb_q} \right\}.$$

The initial conditions are $C(e(k); k) = 0$, where $e(k)$ is the 0-1 unit-vector with a single 1-entry in the kth position. The length of the optimal directed path from vb_k to vb_l with degree sequence b is given by $C(b; \ell)$. It is straightforward to evaluate this recurrence in time proportional to $g^2 \prod_{i=1}^{g}(b_i + 1)$. Since each b_i is bounded by $|B|$ which in turn is bounded as in (3.3), and since g is $O(\log^m \log n)$ by property (I1) in Lemma 3.1, this running time is an extremely slowly growing function in n: It is bounded by $2^{O(\log^{m+1} \log n)}$, which is sublinear and grows more slowly than any polynomial in n. □

LEMMA 3.4. *A multigraph G^* as described in Lemma* 3.2 *can be found in polynomial time.*

Proof. We will show how to find in polynomial time (in fact, in sublinear time in n) the *shortest* balanced multigraph that obeys the in-degrees and satisfies (G1) and (G2). Clearly, this multigraph will also satisfy (G3).

We check all $O(g^2 f^2)$ possibilities for the two crossing arcs (vb_i, vs_j) and (vs_k, vb_l) in (G1). The remaining graph decomposes into a subgraph for the small cities and into a subgraph for the big cities. For the small cities, we want to find a multigraph of minimum length in which city vs_j has in-degree $s_j - 1$ and out-degree s_j, city vs_k has in-degree s_k and out-degree $s_k - 1$, and all other cities vs_h ($1 \le h \le f$ and $j \ne h \ne k$) have in-degree and out-degree both equal to s_h. Such a multigraph can be found in polynomial time by solving a transportation problem (see, e.g., Papadimitriou and Steiglitz [13]). Note that the degree constraints enforce a directed path from vs_j to vs_k in the resulting graph. For the big cities, the only way to satisfy property (G2) is to connect them via a directed path from vb_l to vb_i that obeys all the in-degrees. By Lemma 3.3 a shortest such path can be found in polynomial time. Together with the directed path from the small city vs_j to the small city vs_k, this indeed makes B' strongly connected. ☐

3.3. How to get the approximation scheme. By Lemmas 3.3 and 3.4, we can find in polynomial time a multigraph G^* as described in Lemma 3.2. In general this multigraph will not be connected, since some groups of small cities can form separate connected components.

To repair the situation, we add to G^* a directed Hamiltonian cycle on the vertex set S'. In terms of the no-wait flow shop problem, this means that we add one new job to each small job-type. Clearly the resulting graph is balanced and strongly connected, and hence, Eulerian. We denote the length of the corresponding ATSP tour (and the length of the corresponding permutation schedule) by $C^{(5)}_{\max}$. By (I2) and (I3) in Lemma 3.1, adding the Hamiltonian cycle increases the length of G^* by at most $2\log^m n$ times $2\alpha(1+\varepsilon)\varepsilon L_{\max}/\log^m n$. From (G3) in Lemma 3.2 we now get that

$$C^{(5)}_{\max} \le C^{(4)}_{\max} + 4m(1+\varepsilon)\varepsilon L_{\max} + 4\alpha(1+\varepsilon)\varepsilon L_{\max} \le C^{(4)}_{\max} + 5m(1+\varepsilon)\varepsilon L_{\max}.$$

Since the permutation schedule with makespan $C^{(5)}_{\max}$ can be computed in polynomial time, we finally have reached the desired PTAS.

Let us summarize the running time of this PTAS. We assume the unit cost model of computation, where we can perform all standard arithmetic operations in constant time, such as rounding rationals to integers, adding up two values, multiplying two values, or computing the logarithm of a value. In the PTAS, determining the value of α and computing the partition into big, medium, and small jobs clearly can be done in linear time. The first three rounding steps compare only the length of every operation against certain thresholds; since there are $O(n)$ operations, these comparisons only take $O(n)$ time. In the fourth rounding step, we round each operation to the next integer power of $1+\varepsilon$ which also takes $O(n)$ time in the unit cost model. Finally, the computation of the “almost” Eulerian graph in Lemma 3.4 can be done in time that is even sublinear in n: The number of big and small cities is only polylogarithmic in n, and thus guessing the two crossing arcs, solving the dynamic program for the big cities, and solving the transportation problem for the small cities altogether cost only polylogarithmic time. To summarize, up to a constant factor that depends exponentially on the (fixed) number m of machines and the (fixed) precision ε, the constructed PTAS has a running time linear in n.

THEOREM 3.5. *The no-wait permutation flow shop problem on a fixed number of machines possesses a PTAS whose running time is linear in the number of jobs.*

We conclude this section with a remark on the combinatorial structure of the constructed schedules. The Eulerian cycle in the final graph visits the small and the big cities in two separate blocks, since there are only two crossing arcs (vb_i, vs_j) and (vs_k, vb_l). If we transform this cycle into a schedule by deleting the arc (vs_k, vb_l)—instead of deleting the arc that enters the dummy job J_0—then we increase the length by at most ℓ_{vs_k}. By appending all the medium jobs in the end, this yields a near optimal schedule with a surprisingly primitive structure: The big jobs, the small jobs, and the medium jobs are processed in three separate blocks. This behavior is very different from the classical shop problems without no-wait constraint, where the PTASs heavily rely on mixing big and small jobs [7, 9, 20].

4. Conclusion and open problems. We have shown that the no-wait permutation flow shop problem with a fixed number of machines allows a PTAS. There remain quite a few interesting open questions on no-wait shop scheduling. The most challenging problem is to decide whether the no-wait permutation flow shop with an arbitrary number of machines has a PTAS. In fact, it even would be interesting to get a polynomial time approximation algorithm with constant worst case guarantee for this problem. Note that here the gap-technique as used by Williamson et al. [24] cannot be used to get in-approximability results: For any constant k, the problem of finding a tour of length at most k or deciding that there is no such tour is solvable in polynomial time for the ATSP with triangle inequality. Another question concerns the approximability behavior of the job shop where each job consists of at most two operations. We feel that this problem should have a PTAS.

Acknowledgment. The author thanks G. J. Woeginger for many improvements in this presentation of results made during his work on our joint conference paper [23].

REFERENCES

[1] A. Agnetis, *No-wait flow shop scheduling with large lot sizes*, Ann. Oper. Res., 70 (1997), pp. 415–438.

[2] B. Carr and S. Vempala, *Towards a 4/3 approximation for the asymmetric traveling salesman problem*, in Proceedings of the 11th ACM-SIAM Symposium on Discrete Algorithms (SODA), SIAM, Philadelphia, 2000, pp. 116–125.

[3] S.S. Cosmadakis and C.H. Papadimitriou, *The traveling salesman problem with many visits to few cities*, SIAM J. Comput., 13 (1984), pp. 99–108.

[4] A. Frieze, G. Galbiati, and F. Maffioli, *On the worst case performance of some algorithms for the asymmetric traveling salesman problem*, Networks, 12 (1982), pp. 23–39.

[5] P. Gilmore and R. Gomory, *Sequencing a one state-variable machine: A solvable case of the traveling salesman problem*, Oper. Res., 12 (1964), pp. 655–679.

[6] C. Glass, J.N.D. Gupta, and C.N. Potts, *Two-machine no-wait flow shop scheduling with missing operations*, Math. Oper. Res., 24 (1999), pp. 911–924.

[7] L.A. Hall, *Approximability of flow shop scheduling*, Math. Programming, 82 (1998), pp. 175–190.

[8] N. Hall and C. Sriskandarajah, *A survey of machine scheduling problems with blocking and no-wait in process*, Oper. Res., 44 (1996), pp. 510–525.

[9] K. Jansen, R. Solis-Oba, and M. Sviridenko, *Makespan minimization in job shops: A polynomial time approximation scheme*, in Proceedings of the 31st Annual ACM Symposium on Theory of Computing (STOC), ACM, New York, 1999, pp. 394–399.

[10] V. Kann, *Maximum bounded 3-dimensional matching is MAX SNP-complete*, Inform. Process. Lett., 37 (1991), pp. 27–35.

[11] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, and D.B. Shmoys, *Sequencing and scheduling: Algorithms and complexity*, in Logistics of Production and Inventory, Handbooks in Operations Research and Management Science 4, S. Graves, A.H.G. Rinnooy Kan, and P. Zipkin, eds., North-Holland, Amsterdam, 1993, pp. 445–522.

[12] C.H. Papadimitriou and P. Kanellakis, *Flow-shop scheduling with limited temporary storage*, J. ACM, 27 (1980), pp. 533–549.

[13] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[14] C.H. Papadimitriou and S. Vempala, *On the approximability of the traveling salesman problem*, in Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC), ACM, New York, 2000, pp. 126–133.

[15] J. Piehler, *Ein beitrag zum reihenfolgeproblem*, Unternehmensforschung, 4 (1960), pp. 138–142.

[16] H. Psaraftis, *A dynamic programming approach for sequencing groups of identical jobs*, Oper. Res., 28 (1980), pp. 1347–1359.

[17] H. Röck, *The three-machine no-wait flow shop is NP-complete*, J. ACM, 31 (1984), pp. 336–345.

[18] H. Röck and G. Schmidt, *Machine aggregation heuristics in shop-scheduling*, Methods Oper. Res., 45 (1983), pp. 303–314.

[19] S. Sahni and Y. Cho, *Complexity of scheduling shops with no wait in process*, Math. Oper. Res., 4 (1979), pp. 448–457.

[20] S.V. Sevastianov and G.J. Woeginger, *Makespan minimization in open shops: A polynomial time approximation scheme*, Math. Programming, 82 (1998), pp 191–198.

[21] J.B. Sidney, C.N. Potts, and C. Sriskandarajah, *A heuristic for scheduling two-machine no-wait flow shops with anticipatory setups*, Oper. Res. Lett., 26 (2000), pp. 165–173.

[22] J.B. Sidney and C. Sriskandarajah, *A heuristic for the two-machine no-wait open shop scheduling problem*, Naval Res. Logist., 46 (1999), pp. 129–145.

[23] M. Sviridenko and G.J. Woeginger, *Approximability and in-approximability results for no-wait shop scheduling*, in Proceedings of the 41st Annual Symposium on the Foundations of Computer Science, IEEE, Los Alamitos, CA, 2000, pp. 116–125.

[24] D.P. Williamson, L.A. Hall, J.A. Hoogeveen, C.A.J. Hurkens, J.K. Lenstra, S.V. Sevastianov, and D.B. Shmoys, *Short shop schedules*, Oper. Res., 45 (1997), pp. 288–294.

[25] D.A. Wismer, *Solution of the flow shop scheduling problem with no intermediate queues*, Oper. Res., 20 (1972), pp. 689–697.

SIAM J. Discrete Math.
Vol. 16, No. 2, pp. 323–334

1-HYPERBOLIC GRAPHS[*]

HANS-JÜRGEN BANDELT[†] AND VICTOR CHEPOI[‡]

Abstract. The shortest-path metric d of a graph $G = (V, E)$ is called *δ-hyperbolic* if for any four vertices $u, v, w, x \in X$ the two larger of the three sums $d(u, v) + d(w, x), d(u, w) + d(v, x), d(u, x) + d(v, w)$ differ by at most δ. In this paper, we characterize the graphs with 1-hyperbolic metrics in terms of a convexity condition and forbidden isometric subgraphs.

Key words. trees, graphs, 4-point condition, hyperbolicity

AMS subject classifications. 05C12, 05C75, 52CXX

PII. S0895480100380902

Introduction. It is well known that a metric space (X, d) embeds into a tree network (with positive real edge lengths), that is, d is a *tree metric,* if and only if the "classical" 4-point condition holds, that is, for any 4 points u, v, w, x the two larger of the sums

$$d(u, v) + d(w, x), \quad d(u, w) + d(v, x), \quad d(u, x) + d(v, w) \tag{1}$$

are equal; cf. [1, 6] for a comprehensive bibliography. In the case that d is the shortest-path metric of a graph G, this condition is satisfied exactly when G is a block graph; see [4, 16].

Every metric d on a 4-point set $\{u, v, w, x\}$ (tree-realizable or not) has a canonical representation in the rectilinear plane; cf. [2, 10]. For Figure 1 it is stipulated that the three distance sums in the list (1) are ordered from small to large, thus implying $\xi \le \eta$. Then η is half the difference of the largest and the smallest sum, while ξ is half the largest minus the medium sum. In data analysis, especially the phylogenetic analysis of molecular sequences, the ratio ξ/η would thus (locally) measure the deviation from tree-likeness; cf. [11].

For graph metrics the deviation from a tree metric may be measured directly by ξ (rather than ξ/η) since ξ can only take values from $0, \frac{1}{2}, 1, \frac{3}{2}, \ldots$. More specifically, following a general notion of δ-hyperbolic metric spaces due to Gromov, we say that the shortest-path metric d of a graph G (that associates to each vertex pair the length of a shortest path connecting the pair) is *δ-hyperbolic* (or *tree-like with defect at most* δ) if and only if the difference 2ξ between the largest and medium distance sums for any 4 vertices u, v, w, x does not exceed δ, that is,

$$\begin{aligned}(\beta_\delta) \qquad & d(u, v) + d(w, x) \le d(u, w) + d(v, x) \le d(u, x) + d(v, w) \\ & \text{implies } d(u, x) + d(v, w) - d(u, w) - d(v, x) \le \delta.\end{aligned}$$

δ-hyperbolic metric spaces play an important role in geometric group theory; see for example [12, 13, 14].

[*] Received by the editors November 10, 2000; accepted for publication (in revised form) October 14, 2002; published electronically February 20, 2003.
http://www.siam.org/journals/sidma/16-2/38090.html

[†] Fachbereich Mathematik, Universität Hamburg, Bundesstr. 55, D-20146 Hamburg, Germany (bandelt@math.uni-hamburg.de).

[‡] Laboratoire d'Informatique Fondamentale, Université de la Méditerranée, Faculté des Sciences de Luminy, F-13288 Marseille Cedex 9, France (chepoi@lim.univ-mrs.fr).

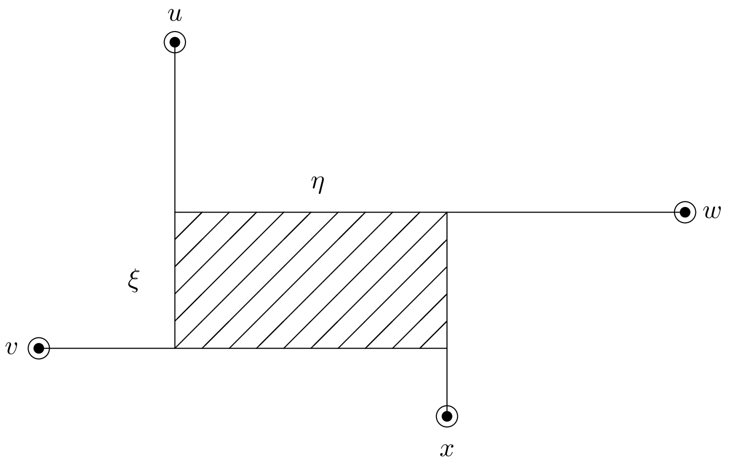


FIG. 1. *Realization of a metric on* $\{u, v, w, x\}$ *in the rectilinear plane.*

In this paper, we will characterize the graphs with 1-hyperbolic metrics, i.e., the graphs satisfying condition (β_1), by structural properties involving cycles and certain forbidden subgraphs. Trivially, 1-hyperbolic graphs can be recognized in polynomial time, but the defining condition alone gives us little immediate insight about why particular graph classes would comprise only 1-hyperbolic graphs. A natural approach is then to seek minimal configurations, each of which violates (β_1) but is a graph in its own right: we say that a subgraph H of G is isometric if H together with its own shortest-path metric constitutes a metric subspace of G endowed with d. Certainly, 1-hyperbolicity in arbitrary graphs cannot be characterized solely in terms of small forbidden isometric subgraphs; however, with an additional requirement for cycles we can achieve a full characterization.

All graphs $G = (V, E)$ occurring here are simple, connected, without loops or multiple edges, but not necessarily finite. We will write $u \sim v$ if the vertices u and v are adjacent (and thus mutual neighbors) in G. A shortest path between two vertices x, y of a cycle C of G is called a *bridge* of C if its length is smaller than the distance between x and y measured along C. A cycle C is called *well-bridged* (in G) if for any vertex $x \in C$ there exists a bridge from x to some vertex of C or if the two neighbors of x from C are adjacent (thus forming a chord). A cycle $C = C_n$ of length $n = 4, 5$ is well-bridged exactly when it is not induced in G, that is, it has some chord, but a noninduced 6-cycle, for instance, is not necessarily well-bridged. The well-known Petersen graph constitutes an example for which all cycles are well-bridged (although it has many induced 6-cycles). The property that all cycles of G except 5-cycles are well-bridged can be translated into a convexity property of balls. A subset A of vertices of G is *convex* if the *interval*

$$I(u, v) = \{x \in X : d(u, x) + d(x, v) = d(u, v)\}$$

between any two vertices u and v of A lies entirely in A. The *ball* $B_k(x)$ of center x and radius $k \geq 0$ consists of all vertices of G at distance $\leq k$ from x.

FACT 1 (see [15, 19]). *For a graph* G, *the following conditions are equivalent:*

(i) *All balls of* G *are convex;*

(ii) *all cycles* C_n, $n \neq 5$, *of* G *are well-bridged;*

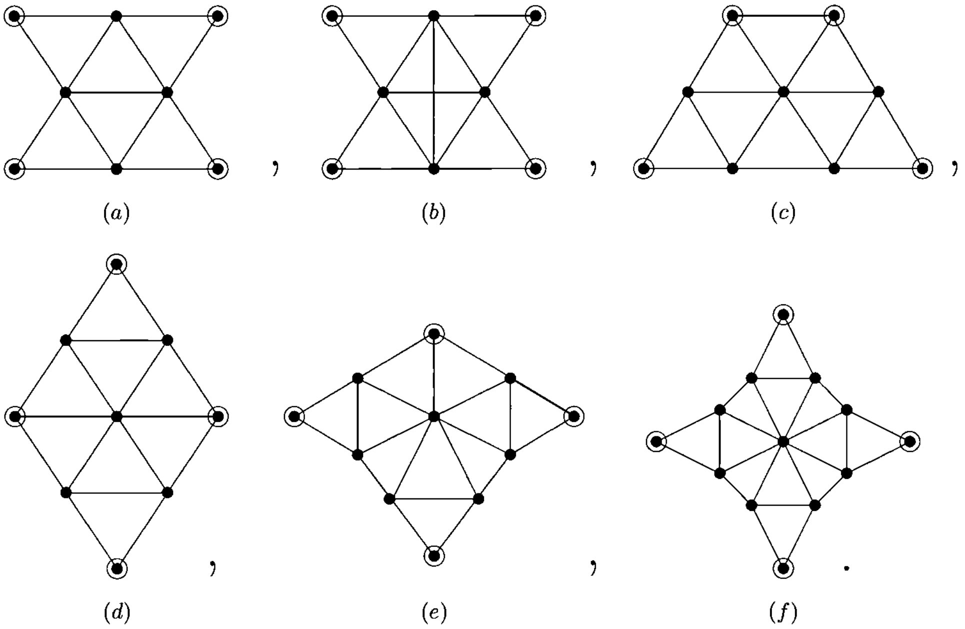


FIG. 2. *Forbidden isometric subgraphs.*

(iii) *G does not contain isometric cycles of length $n > 5$, and for any two vertices x, y the neighbors of x from the interval $I(x, y)$ are pairwise adjacent.*

Any of the preceding equivalent conditions can then be used in our main result characterizing 1-hyperbolicity.

THEOREM. *A graph G is 1-hyperbolic if and only if all cycles $C_n, n \neq 5$, of G are well-bridged and none of the graphs in Figure 2 occur as isometric subgraphs of G.*

A bridged graph G, by definition, has no isometric cycles C_n for $n > 3$; that is, all cycles are well-bridged. Hence our result covers the characterization of 1-hyperbolicity for bridged graphs by forbidden isometric subgraphs as given by Koolen and Moulton [17]. For bridged graphs our proof would simplify in that "Case 2" below cannot occur so that we obtain a rather short proof in this situation.

Preliminaries. In this section, we recall some results used in the proof of the theorem. First, we consider yet another condition involving four points that is related to (β_i) for $i \leq 1$:

(α_i) If $v \in I(u, w)$ and $w \in I(v, x)$ such that v, w are adjacent,
then $d(u, x) \geq d(u, v) + d(w, x) + 1 - i$.

Conditions (α_i) and (β_j) are relevant to near-isometric embedding φ of a graph G (with shortest-path metric d) into a tree network T (with metric d'), where the absolute error is bounded by some $\epsilon > 0$, that is,

$$|d(u, v) - d'(\varphi(u), \varphi(v))| \leq \epsilon,$$

for all vertices u, v of G. If G admits such a representation, then it satisfies (α_i) and (β_j) with $i = \lfloor 3\epsilon \rfloor$ and $j = \lfloor 4\epsilon \rfloor$. Near-isometric tree embeddings were investigated in [7, 9], where it is shown that the error bound ϵ can be chosen to be 2 for chordal graphs and otherwise bounded by a polynomial linear in the maximum length of induced cycles. A variant of this embedding concept additionally requires that $d' \leq d$; then (α_i) and (β_i) both hold with $i = \lfloor 2\epsilon \rfloor$.

Condition (α_0) was first introduced in [18] for metric spaces; in the case of graphs it characterizes the Ptolemaic graphs [8]. All chordal graphs satisfy (α_1) [8], but the latter class is in fact larger.

FACT 2 (see [20]). *A graph G satisfies (α_1) if and only if all cycles C_n, $n \neq 5$, are well-bridged and G does not contain the graph of Figure* 2(c) *as an isometric subgraph.*

It is clear by Facts 1 and 2 that all balls in graphs are convex whenever (α_1) holds. On the other hand, condition (β_1) implies (α_1). Indeed, let v, w be adjacent vertices such that $v \in I(u, w)$ and $w \in I(v, x)$. The distance sums $d(u, w) + d(v, x)$ and $d(u, v) + d(w, x)$ differ by 2. If (β_1) is fulfilled, then necessarily $d(u, x) + d(v, w) \geq d(u, w) + d(v, x) - 1$. Therefore $d(u, x) \geq d(u, v) + d(w, x)$, and hence (α_1) is satisfied. Consequently, in view of Fact 2, we can reformulate the characteristic condition in the theorem in terms of (α_1) and five forbidden isometric subgraphs.

COROLLARY. *A graph G is 1-hyperbolic if and only if it satisfies (α_1) and none of the graphs of Figure* 2(a),(b),(d),(e),(f) *occur as isometric subgraphs.*

Three not necessarily distinct vertices u, v, w of a graph G are said to form a *metric triangle uvw* of type (k_1, k_2, k_3), where $d(u, v) = k_1, d(v, w) = k_2$, and $d(w, u) = k_3$, if the intervals $I(u, v), I(v, w)$, and $I(w, u)$ intersect pairwise only in the common end vertices. If $k_1 = k_2 = k_3 = k$, then this metric triangle is called *equilateral* of *size* k; when $k = 0$ this metric triangle is *degenerate,* consisting of a single vertex. A metric triangle uvw is a *quasi-median* of the triplet x, y, z if the following equations are satisfied:

$$\begin{aligned} d(x, y) &= d(x, u) + d(u, v) + d(v, y), \\ d(y, z) &= d(y, v) + d(v, w) + d(w, z), \\ d(z, x) &= d(z, w) + d(w, u) + d(u, x). \end{aligned}$$

If this quasi-median is an equilateral triangle, then its size will be denoted by $q(x, y, z)$. Note that in [5] quasi-medians were defined more restrictively by admitting only equilateral triangles. If the size $q(x, y, z)$ equals zero, then the (degenerate) quasi-median (triangle) is also referred to as a *median* (vertex).

We conclude this section with a somewhat technical observation, which is quite useful in the proof of the theorem.

LEMMA 1. *Given $1 \leq k \leq l$, let u, v, w, x be vertices of a graph G such that*

$$d(u, v) + d(w, x) + j = d(u, w) + d(v, x) + i = d(u, x) + d(v, w),$$

for which the sum of all distances between u, v, w, x is minimal relative to the constraints $i \geq k$ and $j \geq l$. If all quasi-medians of these vertices are equilateral triangles, then

$$\begin{aligned} d(u, v) + d(u, w) &= d(v, w) + q(u, v, w), \\ d(u, w) + d(w, x) &= d(u, x) + q(u, w, x), \\ d(u, v) + d(v, x) &= d(u, x) + q(u, v, x), \\ d(v, x) + d(w, x) &= d(v, w) + q(v, w, x), \end{aligned}$$

whence

$$\begin{aligned} q(u,v,w)+q(u,w,x) &= 2d(u,w)-j,\\ q(u,v,w)+q(u,v,x) &= 2d(u,v)-i,\\ q(v,w,x)+q(u,w,x) &= 2d(w,x)-i,\\ q(v,w,x)+q(u,v,x) &= 2d(v,x)-j, \end{aligned}$$

and finally

$$q(u,v,w)+q(u,v,x)+q(u,w,x)+q(v,w,x) = 2[d(u,x)+d(v,w)-i-j].$$

Proof. Suppose that the first of the asserted equations fails. Then we can find a neighbor u' of u in $I(u,v)\cap I(u,w)$. Substituting u by u' reduces the smaller two distance sums for u,v,w,x by 1 and the largest one by at most 1, while the sum of all distances between u,v,w,x decreases, contrary to the minimality assumption. Since we may interchange u with x, or v with w, or $\{u,x\}$ with $\{v,w\}$, we obtain the subsequent three equations by symmetry. Adding up the first two equations yields the asserted equality for $q(u,v,w)+q(u,w,x)$, and so forth. The final equation is obtained from the sum of the second set of equations. □

Properties of graphs satisfying (α_1). Throughout this section, G is a graph in which (α_1) holds.

LEMMA 2. *If uvw is a metric triangle of G with $d(u,v)=1$, then $d(w,u)=d(w,v)\leq 2$.*

Proof. Let $u'\in I(w,u)$ and $v'\in I(w,v)$ be neighbors of w. From the initial hypothesis we conclude that $u\in I(u',v)$ and $v\in I(v',u)$. Applying (α_1) we obtain

$$d(u',v')\geq d(u',u)+d(v,v')=d(u,w)+d(v,w)-2=2d(u,w)-2.$$

Since $d(u',v')\leq 2$, it follows immediately that $d(u,w)\leq 2$ holds as well. □

LEMMA 3. *If $x,y\in I(u,v)$ are neighbors of u, then $x\sim y$. Furthermore, x and y have a common neighbor v' satisfying $d(v',v)=d(u,v)-2$.*

Proof. Let $d(u,v)=k+1$. Since $x,y\in B_k(v), u\notin B_k(v)$, and the ball $B_k(v)$ is convex, necessarily x and y are adjacent. Let $v'\in I(x,v)\cap I(y,v)$ be closest to x and y, thus giving a metric triangle xyv'. From Lemma 2 we know that $d(x,v')=d(y,v')\leq 2$. If v' is adjacent to x and y, we are done. Now suppose $d(x,v')=d(y,v')=2$. Denote by x' and y' some common neighbors of v',x and v',y, respectively. Then $x'\sim y'$ because $x',y'\in I(v',v)$. We obtain a 4-cycle (x,y,y',x'), which must have at least one diagonal, say $x'\sim y$. Consequently, $x'\in I(x,v)\cap I(y,v)$, contrary to the choice of v'. □

LEMMA 4. *Every metric triangle uvw of G is of type* $(1,1,1),(1,2,2),(2,1,2),(2,2,1)$, *or* $(2,2,2)$.

Proof. Assume $d(u,v)\leq d(v,w)\leq d(w,u)$. We proceed by induction on $k=d(u,v)+d(v,w)+d(w,u)$. The case $d(u,v)=1$ is covered by Lemma 2. We distinguish two further cases.

Case 1. $d(u,v)=2$.

Suppose by way of contradiction that $d(u,w)\geq 3$. Choose a common neighbor x of u and v. If $d(x,w)>d(u,w)$, then (α_1) yields $d(v,w)\geq d(w,u)+1$, contrary to the maximality of $d(u,w)$. Therefore $d(x,w)=d(u,w)\geq 3$. By Lemma 2, we can find a neighbor w' of w such that $w'\in I(w,u)\cap I(w,x)$. If $d(x,w)=d(v,w)$, then there exists another neighbor $w''\in I(w,v)\cap I(w,x)$ of w. On the other hand, if

$d(v,w) < d(x,w)$, then $v \in I(x,w)$, whence every neighbor $w'' \in I(w,v)$ of w belongs to $I(w,x)$. In both cases, we have found a vertex $w'' \in I(w,v) \cap I(w,x)$ adjacent to w and distinct from w'. Since $w', w'' \in I(w,x)$, Lemma 3 yields $w' \sim w''$. Necessarily, $w' \in I(w'',u)$ and $w'' \in I(w',v)$. By condition (α_1), we have

$$2 = d(u,v) \geq d(u,w') + d(w'',x) = d(u,w) + d(v,w) - 2,$$

whence $d(u,w) = 2$, contrary to our assumption.

Case 2. $d(u,v) \geq 3$.

Choose a neighbor x of w in the interval $I(u,w)$. If $d(x,v) > d(w,v)$, then (α_1) yields $d(u,v) \geq d(u,w) + d(v,w) - 1 > d(u,w)$, contrary to the maximality of $d(u,w)$. Thus $d(x,v) = d(w,v)$. Since this distance exceeds 2, we can find a neighbor y of v in $I(v,w) \cap I(v,x)$ by virtue of Lemma 2. For a quasi-median $uv'x'$ of the triplet u, v, x, we obtain

$$\begin{aligned} &d(u,w) - d(u,x') - 1 + d(x',v') + d(u,v) - d(u,v') \\ &= d(x,x') + d(x',v') + d(v',v) = d(x,v) = d(w,v) \leq d(u,w), \end{aligned} \tag{2}$$

and therefore

$$d(u,v) \leq d(u,x') + d(u,v') - d(x',v') + 1. \tag{3}$$

We can then apply the induction hypothesis to the metric triangle $uv'x'$ because

$$3 \leq d(u,v') + d(v',x') + d(x',u) \leq d(u,v) + d(v,x) + d(x,u) < k.$$

Using this hypothesis and the inequality $d(u,v) \geq 3$, we deduce from (3) that $d(x',u) = d(u,v') = 2$ and

$$4 \leq d(u,v) + d(v',x') \leq 5.$$

If the upper bound 5 is attained, then we infer from (2) that uvw is equilateral (of size ≥ 3). Choose a neighbor y' of v in $I(v,v')$, which is necessarily distinct from y. Since $y, y' \in I(x,v)$, Lemma 3 implies $y \sim y'$. Since uvw is a metric triangle, $d(y',w) > d(y,w)$ and $d(y,u) > d(y',u)$ hold. By condition (α_1), we thus infer

$$d(u,w) \geq d(u,y') + d(y,w) = d(u,v) + d(v,w) - 2 > d(v,w),$$

a contradiction.

Finally, if the lower bound 4 is attained above, then $x' \sim v' \sim v$, and hence $v' \sim y$ by Lemma 3 applied to $I(v,x)$. Moreover,

$$d(v',w) = d(v,w) = d(w,u) - 1$$

by (2), and therefore

$$d(v',w) - 1 = d(y,w) = d(w,x').$$

Consequently, $x' \sim y$ by Lemma 3. Since $uv'x'$ is a metric triangle, any neighbors $v'' \in I(u,v')$ and $x'' \in I(u,x')$ of u are different. From $v'', x'' \in I(u,y)$ and Lemma 3 it follows that v', x', x'', v'' constitutes a 4-cycle. By condition (α_1) either $v' \sim x''$ or $x' \sim v''$, which, however, is impossible. □

Lemma 5. *Let u, v, w, x be vertices of G such that $u \in I(v, w), v \in I(u, x)$, and $u \sim v$. Then the equality $d(w, x) = d(w, u) + d(x, v)$ holds if and only if there exist vertices $u' \in I(u, w)$ and $v' \in I(v, x)$ at distance 2 such that $u' \sim u$ and $v' \sim v$; in particular, u' and v' lie on a common shortest path between w and x.*

Proof. If the condition is fulfilled, then the equality holds trivially. Now assume the converse and, without loss of generality, $I(w, x) \cap I(w, u) = \{w\}$ and $I(x, w) \cap I(x, v) = \{x\}$. We wish to show that $d(u, w) = d(v, x) = 1$, which settles the proof. Observe that $I(u, x) \cap I(u, w) = \{u\}$ and $I(v, w) \cap I(v, x) = \{v\}$ since

$$d(w, u) + d(u, x) = d(w, v) + d(v, x) = d(w, x) + 1.$$

Therefore the quasi-medians of the triplets u, w, x and v, w, x have the forms uwx' and $vw'x$, respectively. From Lemma 4 we know that $d(u, w) \leq 2$ and $d(v, x) \leq 2$. Suppose by way of contradiction that $d(u, w) + d(v, x) \geq 3$.

First, let $d(u, w) = 1$ and $d(v, x) = 2$ so that $d(w, x) = 3$. Then any quasi-median $vw'x$ of v, w, x is a metric triangle of type $(1, 2, 2)$, whence $v \sim w' \sim w$. Since $u, w' \in I(w, v)$, Lemma 3 yields $u \sim w'$. Therefore $v, w' \in I(u, x)$ have a common neighbor x' with x by Lemma 3. As a consequence x' belongs to $I(x, w) \cap I(x, v)$, contrary to our initial assumption.

Finally, let $d(u, w) = d(v, x) = 2$ so that $d(x, w) = 4$. Then any quasi-median $vw'x$ of v, w, x is of type $(1, 2, 2)$. Therefore $u, w' \in I(v, w)$ have a common neighbor w'' with w, thus conflicting with the initial assumption on $I(w, x) \cap I(w, u)$. □

Proof of the theorem. Each of the six forbidden graphs violates condition (β_1), as indicated in Figure 2. Since (β_1) implies (α_1), we deduce from Fact 2 that in a 1-hyperbolic graph all cycles $C_n, n \neq 5$, are well-bridged.

Conversely, assume that all cycles of length $\neq 5$ are well-bridged and none of the graphs from Figure 2 occurs as an isometric subgraph of G. By Fact 1, all balls of G are convex, and by Fact 2, G obeys condition (α_1). Suppose by way of contradiction that condition (β_1) is violated. Select a quartet u, v, w, x minimizing the total distance sum

$$\Sigma(u, v, w, x) = d(u, x) + d(v, w) + d(u, v) + d(u, w) + d(v, x) + d(w, x).$$

Then we have, say,

$$d(u, v) + d(w, x) + j = d(u, w) + d(v, x) + i = d(u, x) + d(v, w), \tag{4}$$

with $i \geq 2$ and $j \geq 2$. As in the proof of Lemma 1, the minimality of $\Sigma(u, v, w, x)$ guarantees that

$$\begin{aligned} I(u, v) \cap I(u, w) = \{u\}, \quad & I(v, u) \cap I(v, x) = \{v\}, \\ I(w, u) \cap I(w, x) = \{w\}, \quad & I(x, v) \cap I(x, w) = \{x\}. \end{aligned} \tag{5}$$

We distinguish two main cases.

Case 1. All quasi-medians of triplets from u, v, w, x are equilateral metric triangles.

If all triplets actually have medians, then u, v, w, x would induce a 4-cycle by Lemma 1, which, however, is impossible in view of (α_1). Therefore, assume without loss of generality that some quasi-median of the triplet u, v, w is an equilateral metric triangle $uv'w'$ of size $k \in \{1, 2\}$. We assert that $i = j = 2$. Suppose by way of contradiction that $d(u, w) + d(v, x) > d(u, v) + d(w, x)$. Choose a neighbor u' of u in the interval $I(u, w')$. Then the ball $B_k(v')$ includes u, w' and thus u' by convexity, whence $d(v, u') = d(v, u)$ by (5). The quartet u', v, w, x also violates condition (β_1); therefore $\Sigma(u', v, w, x) \geq \Sigma(u, v, w, x)$, which implies $d(u', x) > d(u, x)$. By (α_1), we have $d(x, w) \geq d(u, x) + d(u, w) - 1$. Since $d(v, w) = d(v, u) + d(u, w) - k$, we obtain $d(x, w) + d(u, v) - k \geq d(u, x) + d(v, w) - 1$, which contradicts (4) because $k \leq 2$. This proves the assertion $i = j = 2$. From Lemma 1 we immediately infer that the distances $d(u, v), d(u, w), d(x, v)$, and $d(x, w)$ are no larger than 3.

If $q(u, v, w) = 1$, then Lemma 1 implies that $q(u, v, x), q(u, w, x)$, and $q(v, w, x)$ must be odd and thus equal to 1, whence $d(u, w) = d(u, v) = d(w, x) = d(v, x) = 2$. Let $uv'w'$ be a quasi-median of u, v, w and let $xv''w''$ be a quasi-median of x, v, w. Since $d(u, x) = d(v, w) = 3$, the vertices v', w', v'', w'' are pairwise different and belong to the interval $I(v, w)$. Then, by Lemma 3 and (α_1), these four vertices induce a 4-cycle with at least one diagonal so that together with u, v, w, x they induce the graph of Figure 2(a) or (b) as an isometric subgraph, contrary to the initial assumption.

Therefore the quasi-median $uv'w'$ of u, v, w has size 2. Choose vertices t and y with $u \sim t \sim v'$ and $u \sim y \sim w'$. Then $d(v', y) = d(w', t) = 2$ because $B_2(v')$ and $B_2(w')$ are convex. If $d(t, x) > d(u, x)$, then (α_1) implies

$$\begin{aligned} d(v,x) &\geq d(v,t) + d(u,x) \\ &= d(u,v) - 1 + d(u,w) + d(v,x) + 2 - d(v,w) \quad \text{by } (4) \\ &= d(v,x) + 3, \end{aligned}$$

a contradiction. Therefore $\Sigma(t, v, w, x) < \Sigma(u, v, w, x)$, whence the former quartet satisfies (β_1), yielding $t \in I(u, x)$. Analogously, $y \in I(u, x)$ holds so that, by Lemma 3, t and y are adjacent and have a common neighbor z such that $d(x, z) = d(u, z) - 2$. If $d(v, t) < d(v, z)$, then (α_1) implies (as above)

$$d(v,x) \geq d(v,t) + d(z,x) = d(v,t) + d(u,x) - 2 = d(v,x) + 1,$$

a contradiction. We conclude that both t and z belong to $I(v, y)$ and have a common neighbor r such that $d(r, v) = d(u, v) - 2$. Similarly, there exists a vertex s with $y \sim s \sim z$ and $d(s, w) = d(u, w) - 2$. Then urs is a quasi-median of u, v, w, and we may thus assume $v' = r$ and $w' = s$. The six vertices t, u, v', w', y, z induce what is called a 3-*sun* [7]. Without loss of generality, we may assume

$$d(z,x) \leq d(w',x) \leq d(v',x) \leq d(z,x) + 1$$

so that three cases have to be distinguished.

Subcase 1.1. $d(z, x) = d(v', x)$.

Since $v', z \in I(t, x)$ and $w', z \in I(y, x)$, we find vertices $x', x'' \in I(z, x)$ by Lemma 3 such that $z \sim x' \sim v'$ and $z \sim x'' \sim w'$; moreover, x' and x'' are different (because v' and w' are not adjacent) and thus adjacent, having a common neighbor x''' in $I(z, x)$ at distance 2 from z. Then $u, t, y, v', z, w', x', x'', x'''$ induce the graph of Figure 2(d) as an isometric subgraph.

Subcase 1.2. $d(w', x) < d(v', x)$.

Then, as above, we derive from (α_1) that

$$d(v,x) \geq d(v,v') + d(z,x) = d(v,t) + d(z,x) - 1 = d(v,x),$$

whence equality holds throughout. In particular, v', v, x have a quasi-median $v'vp$ of size 1 as $v' \in I(u, v)$ and $d(u, v) \leq 3$. Then p and z, as members of $I(v', x)$, are adjacent and have a common neighbor $x' \in I(v',x)$ with $d(v',x') = 2$. Since $x' \in I(z,x)$, we have $d(x',u) = 3$. If x' is adjacent to w', then v, x', w', u would violate (α_1). Therefore, by Lemma 3, z and w' have a common neighbor $x'' \in I(y,x)$ such that $x'' \neq x'$ and $d(u,x'') = 3$. Further, x' and x'' are adjacent and have a common neighbor $x''' \in I(z,x)$, where $d(u,x''') = 4$. Necessarily, $d(v,x''') = 3$ because $v'vp$ is a quasi-median of v', v, x. Then the vertices $u, t, y, v', z, w', v, p, x', x'', x'''$ induce the graph of Figure 2(e). This subgraph is in fact isometric because $d(v,x'') < 3$ would conflict with (α_1) for the quartet v, x'', w', u.

Subcase 1.3. $d(z,x) < d(w',x)$.

Then, as in the preceding subcase, v', v, x and w', w, x have quasi-medians $v'vp$ and $w'wq$, respectively, of size 1 such that $p \sim z \sim q$. Note that $d(u,x) \leq 4 = d(v,w)$ in view of Lemma 1. By Lemma 3, z has neighbors $x', x'' \in I(z,x)$ in common with p and q, respectively. If x' and w have some common neighbor q', then $q' \sim w'$ because $q', w' \in I(w,v)$. Necessarily, z and q' are adjacent because otherwise z, x', q', w' would induce a 4-cycle. Consequently, $u, t, y, v', z, w', v, p, x', q', w$ induce the graph of Figure 2(e) as an isometric subgraph. Therefore we can now assume $d(x',w) = 3$ and, analogously, $d(x'',v) = 3$. In particular, x' and x'' are different, whence by Lemma 1 the three vertices x', x'', and x are adjacent. Finally, $u, t, y, v', z, w', v, p, x', x'', q, w, x$ induce the graph of Figure 2(f) as an isometric subgraph.

Case 2. Some quasi-median of a triplet from u, v, w, x is not equilateral.

Then, by Lemma 4, we may assume without loss of generality that u, v, w has a quasi-median $uv'w'$ of type $(2,1,2)$ or $(1,2,2)$; see the two principal subcases below. When a common neighbor is selected for each of the two pairs at distance 2 in this quasi-median, then a 5-cycle arises, which must be induced because induced 4-cycles are forbidden.

Subcase 2.1. $d(v',w') = 1$.

Since $\Sigma(v',v,w,x) < \Sigma(u,v,w,x)$, the quartet v', v, w, x must satisfy (β_1), which implies $v' \in I(u,x)$ in view of (4). Analogously, we infer that $w' \in I(u,x)$. Hence, by Lemma 3, any two neighbors of u in $I(u,v') \cup I(u,w') \subseteq I(u,x)$ are adjacent, whence u, v', w' cannot lie on an induced 5-cycle, yielding a contradiction.

Subcase 2.2. $d(u,v') = 1$.

If $d(u,x) < d(v',x)$, then by (α_1) we have

$$\begin{aligned} d(v,x) &\geq d(v,v') + d(u,x) = d(u,v) + d(u,x) - 1 \\ &= d(u,v) + d(u,w) + d(v,x) - d(v,w) + i - 1 \quad \text{by (4)} \\ &= d(v,x) + i, \end{aligned}$$

a contradiction. Therefore $\Sigma(v',v,w,x) < \Sigma(u,v,w,x)$ so that the quartet v', v, w, x must satisfy (β_1). Consequently, we obtain $i = 2$ and $v' \in I(u,x)$. Choose some vertex y with $u \sim y \sim w'$ and a common neighbor z of v' and w' at minimum distance to x. As these five vertices induce a 5-cycle and $v' \in I(u,v) \cap I(u,x)$, neither $I(u,v)$ nor $I(u,x)$ can contain y by Lemma 3, that is, $d(u,v) \leq d(y,v)$ and $d(u,x) \leq d(y,x)$. We also have $d(z,x) \leq d(v',x)$ and hence $d(w',x) \leq d(u,x)$; otherwise by (α_1) we would get

$$\begin{aligned} d(w,x) &\geq d(w,z) + d(v',x) = d(w,u) + d(u,x) - 2 \\ &= d(w,u) + d(u,v) + d(w,x) - d(v,w) + j - 2 \quad \text{by (4)} \\ &= d(w,x) + j - 1, \end{aligned}$$

a contradiction. Hence, if $d(u,x) < d(y,x)$, then $u, w' \in I(y,x)$, which, however, is in conflict with Lemma 3. We conclude that $d(u,x) = d(y,x)$. Now, if $d(u,v) = d(y,v)$, then $\Sigma(y,v,w,x) < \Sigma(u,v,w,x)$, but (β_1) is violated for y,v,w,x. Therefore $d(y,v) = d(u,v)+1$. Finally, as $d(z,x) \leq d(u,x)-1$, the parameter

$$k = d(z,x) + d(w',x) - 2d(u,x) + 4$$

can take only the value 1, 2, or 3, which leads to the following distinction of Subcase 2.2.k, for $k = 1,2,3$.

Subcase 2.2.1. $d(z,x) = d(u,x) - 2$.

Then $d(w',x) = d(u,x) - 1$. Further, by (α_1) and (4),

$$d(v,x) \geq d(v,v') + d(z,x) = d(v,v') + d(u,x) - 2 = d(v,x),$$

whence equality holds throughout. Lemma 5 thus provides us with neighbors $v'' \in I(v',v)$ of v' and $x' \in I(z,x)$ of z such that $d(v'',x') = 2$. Note that $d(v'',w') = 3 = d(u,x')$. Since the quartet u,v'',w',x' violates (β_1) and satisfies $\Sigma(u,v'',w',x') = 14 \leq \Sigma(u,v,w,x)$, we conclude that $v'' = v, w' = w$, and $x' = x$. Choose a common neighbor p of v and x, at minimum distance to y, and then a neighbor $q \in I(p,y)$ of p. First assume that $p \sim v'$. Then $p \sim z$ because induced 4-cycles are forbidden. Since $\{u,w\} \nsubseteq I(p,y)$ in view of Lemma 3, we have $d(p,y) = 2$. Then as q belongs to $I(y,v) \cap I(y,x)$ it must be adjacent to u,w (by Lemma 3) and hence to v',z as well. Consequently, u,v,v',q,w,p,z,x induce the graph of Figure 2(b) as an isometric subgraph.

Therefore we can assume $d(p,v') = 2 = d(p,z)$ so that, as $\{p,z\} \nsubseteq I(u,x)$ and $\{p,v'\} \nsubseteq I(v,w)$ by Lemma 3, we obtain $d(p,u) = d(p,w) = 3$. Thus, $y \in I(u,w) \subseteq B_3(p)$ by convexity. If $d(p,y) = 2$, then $p,v' \in I(v,y)$ would have to be adjacent by Lemma 3, contrary to our assumption. Therefore $d(p,y) = 3$. Then $v,x,y \in B_2(q)$ and thus $u,w \in B_2(q)$ by convexity. Hence as $q,v \in I(p,u)$ and $q,x \in I(p,w)$ we obtain q as a common neighbor of v and x such that $d(q,y) < d(p,y)$, contrary to the choice of p.

Subcase 2.2.2. $d(z,x) = d(w',x) = d(u,x) - 1$.

Since $\Sigma(z,v,w,x) < \Sigma(u,v,w,x)$, the quartet z,v,w,x must satisfy (β_1), which implies $j = 2$. By virtue of the triangle inequality and employing (4) as above, we obtain

$$\begin{aligned} d(w,x) &\leq d(w,w') + d(w',x) = d(w,z) + d(v',x) - 1 \\ &= d(w,x) + j - 2 = d(w,x), \end{aligned}$$

whence equality holds throughout. In particular, $w' \in I(w,u) \cap I(w,x)$ so that $w = w'$ by (5). Since z was chosen as a common neighbor of v' and $w' = w$ closest to x, the triplet v',x,w has a quasi-median $v'x'w$ of type $(2,2,2)$ by Lemma 4. Choose vertices p and q with $v' \sim p \sim x'$ and $w \sim q \sim x'$. By the choice of z, neither is p adjacent to w nor is q adjacent to v'. By convexity of balls, $p \in I(v',x') \subseteq B_2(w)$. Hence, if $d(p,y) = 3$, then $u,w \in I(p,y)$ would have to be adjacent, which is impossible. Therefore $d(p,y) = 2$, and by Lemma 3 applied to $I(x',y)$, we infer that $p \sim q$ and that there exists a common neighbor t of p,q,w,y.

If $d(v,t) = d(v,w)$, then t is not adjacent to u because otherwise $t \sim v'$, and hence $t \in I(v,w)$ would follow. Therefore $\Sigma(u,v,t,x) = \Sigma(u,v,w,x)$ holds and u,v,t,x violate (β_1). Since $uv't$ is a quasi-median of u,v,t such that p is a common neighbor of v' and t with $d(p,x) = d(u,x) - 2$, we can substitute w by t and z by p. Thus, we are back in Subcase 2.2.1.

Therefore we can assume $d(v,t) = d(v,w) - 1 = d(v,y) - 1$. Then $t, u \in I(v,y)$ and, consequently, $t \sim u$ by Lemma 3 and hence $t \sim v'$. Now, however, $uv'w$ is no longer a quasi-median of u, v, w, yielding a contradiction.

Subcase 2.2.3. $d(z,x) = d(w',x) - 1 = d(u,x) - 1$.

We can apply (α_1) to the quartet w, w', z, x and infer, as above,

$$\begin{aligned} d(w,x) &\geq d(w,w') + d(z,x) = d(w,z) + d(v',x) - 1 \\ &= d(w,x) + j - 2, \end{aligned}$$

whence $j = 2$ and equality holds throughout. By Lemma 5 we thus find neighbors $x' \in I(x,z)$ of z and $w'' \in I(w,w')$ of w such that x' and w'' have some common neighbor t. Then as $z \in I(v,w'')$, either $d(v,x') = d(v,z)$ or $d(v,x') = d(v,z) + 1$ holds. If the latter is true, (α_1) applied to the quartet v, z, x', x yields, as above,

$$\begin{aligned} d(v,x) &\geq d(v,z) + d(x',x) = d(v,v') + d(u,x) - 1 \\ &= d(v,x) + i - 1, \end{aligned}$$

a contradiction. Thus the former alternative is true. Therefore $t, w' \in I(v,w'')$, and we infer by Lemma 3 that $t \sim w'$ and hence $t \sim z$. Then $B_2(t)$ contains u by convexity because $v', y \in B_2(t)$. This, however, implies $t \in I(w,u) \cap I(w,x)$, contrary to (5).

This final contradiction shows that G is indeed 1-hyperbolic, which completes the proof. $\square$

REFERENCES

[1] H.-J. Bandelt, *Recognition of tree metrics,* SIAM J. Discrete Math., 3 (1990), pp. 1–6.

[2] H.-J. Bandelt and A.W.M. Dress, *A canonical decomposition theory for metrics on a finite set,* Adv. Math., 92 (1992), pp. 47–105.

[3] H.-J. Bandelt, A. Henkmann, and F. Nicolai, *Powers of distance-hereditary graphs,* Discrete Math., 145 (1995), pp. 37–60.

[4] H.-J. Bandelt and H.M. Mulder, *Distance-hereditary graphs,* J. Combin. Theory Ser. B, 41 (1986), pp. 182–208.

[5] H.-J. Bandelt, H.M. Mulder, and E. Wilkeit, *Quasi-median graphs and algebras,* J. Graph Theory, 18 (1994), pp. 681–703.

[6] J.-P. Barthélemy and A. Guénoche, *Trees and Proximity Representations,* Wiley, New York, 1991.

[7] A. Brandstädt, V. Chepoi, and F. Dragan, *Distance approximating trees for chordal and dually chordal graphs,* J. Algorithms, 30 (1999), pp. 166–184.

[8] V. Chepoi, *Some properties of d-convexity in triangular graphs,* Mat. Issled., 184 (1986), pp. 164–177 (in Russian).

[9] V. Chepoi and F. Dragan, *A note on distance approximating trees in graphs,* European J. Combin., 21 (2000), pp. 761–766.

[10] A.W.M. Dress, *Trees, tight extensions of metric spaces, and the cohomological dimension of certain groups: A note on combinatorial properties of metric spaces,* Adv. Math., 53 (1984), pp. 321–402.

[11] M. Eigen, R. Winkler-Oswatitsch, and A.W.M. Dress, *Statistical geometry in sequence space: A method of comparative sequence analysis,* Proc. Natl. Acad. Sci. USA, 85 (1988), pp. 5913–5917.

[12] E. Ghys, A.H. Haefliger, and A. Verjovsky, eds., *Group Theory from a Geometric Viewpoint,* World Scientific, River Edge, NJ, 1991.

[13] E. Ghys and P. de la Harpe, eds., *Les groupes hyperboliques d'après M. Gromov,* Progress in Mathematics 83, Birkhäuser, Basel, 1990.

[14] M. Gromov, *Hyperbolic groups*, in Essays in Group Theory, S.M. Gersten, ed., Math. Sci. Res. Inst. Pub. 8, Springer, New York, 1987, pp. 75–263.

[15] M. Farber and R.E. Jamison, *On local convexities in graphs,* Discrete Math., 66 (1987), pp. 231–247.

[16] E. Howorka, *On metric properties of certain clique graphs,* J. Combin. Theory Ser. B, 27 (1979), pp. 67–74.

[17] J. Koolen and V. Moulton, *Hyperbolic bridged graphs,* European J. Combin., 23 (2002), pp. 683–699.

[18] C. Reda, *Straight lines in metric spaces,* Demonstr. Math., 6 (1973), pp. 809–819.

[19] V.P. Soltan and V.D. Chepoi, *Conditions for invariance of set diameters under d-convexification in a graph,* Cybernetics, 19 (1983), pp. 750–756.

[20] S.V. Yushmanov and V. Chepoi, *A general method of investigation of metric graph properties related to the eccentricity,* in Mathematical Problems in Cybernetics 3, Nauka, Moscow, 1991, pp. 217–232 (in Russian).

SIAM J. DISCRETE MATH.
Vol. 16, No. 2, pp. 335–343

STRONGLY CONNECTED SPANNING SUBDIGRAPHS WITH THE MINIMUM NUMBER OF ARCS IN QUASI-TRANSITIVE DIGRAPHS[*]

JØRGEN BANG-JENSEN[†], JING HUANG[‡], AND ANDERS YEO[§]

Abstract. We consider the problem of finding a strongly connected spanning subdigraph with the minimum number of arcs in a strongly connected digraph. This problem is NP-hard for general digraphs since it generalizes the Hamiltonian cycle problem. We show that the problem is polynomially solvable for quasi-transitive digraphs. We describe the minimum number of arcs in such a spanning subdigraph of a quasi-transitive digraph in terms of the path covering number. Our proofs are based on a number of results (some of which are new and interesting in their own right) on the structure of cycles and paths in quasi-transitive digraphs and in extended semicomplete digraphs. In particular, we give a new characterization of the longest cycle in an extended semicomplete digraph. Finally, we point out that our proofs imply that the MSSS problem is solvable in polynomial time for all digraphs that can be obtained from strong semicomplete digraphs on at least two vertices by replacing each vertex with a digraph belonging to a family of digraphs whose path covering number can be decided in polynomial time.

Key words. minimum equivalent digraph, strong subdigraph, Hamiltonian cycle, polynomial algorithm, quasi-transitive digraph, extended semicomplete digraph, path factor, cycle factor, path cover, longest cycle

AMS subject classifications. 05C20, 05C38, 05C40

PII. S0895480199354220

1. Introduction.

We consider the following problem, which we denote by MSSS (minimum spanning strong subdigraph): Given a strongly connected digraph D, find a strongly connected spanning subdigraph D' of D such that D' has as few arcs as possible. This problem, which generalizes the Hamiltonian cycle problem, and hence is NP-hard, is of practical interest and has been considered several times in the literature; see, e.g., [1, 9, 13, 15, 16, 17]. The MSSS problem is an essential subproblem of the so-called *minimum equivalent digraph problem* (in fact, these two problems can be reduced to each other in polynomial time). Here one is seeking a spanning subdigraph with the minimum number of arcs in which the reachability relation is the same as in the original graph (i.e., there is a path from x to y if and only if the original digraph has such a path). Since the MSSS problem is NP-hard, it is natural to study the problem under certain extra assumptions. In order to find classes of digraphs for which we can solve the MSSS problem in polynomial time, we must consider classes of digraphs for which we can solve the Hamiltonian cycle problem in polynomial time. This follows from the fact that the Hamiltonian cycle problem can be solved if we can solve the MSSS problem.

[*] Received by the editors March 24, 1999; accepted for publication (in revised form) September 23, 2002; published electronically February 20, 2003. The first and second authors wish to acknowledge financial support from NSERC (under grant 203191-98).

http://www.siam.org/journals/sidma/16-2/35422.html

[†] Department of Mathematics and Computer Science, University of Southern Denmark, DK-5230 Denmark (jbj@imada.sdu.dk).

[‡] Department of Mathematics and Statistics, University of Victoria, Victoria, BC, V8W 3P4 Canada (jing@math.uvic.ca).

[§] Department of Computer Science, Royal Holloway, University of London, Egham Surrey TW20 0EX, UK (anders.cs.rhul.ac.uk). This author wishes to acknowledge financial support from the Danish Research Council (under grant 9800435).

In [16] the MSSS problem was considered for digraphs whose longest cycle has length r for some r. It was shown that if $r \leq 3$, then the problem is polynomial and that it is NP-hard already when $r = 5$. In [7], the MSSS problem was solved for various generalizations of tournaments. In particular, polynomial algorithms were given for the classes of extended semicomplete digraphs and semicomplete bipartite digraphs. Furthermore, it was conjectured in [7] that the MSSS problem is also polynomially solvable for general semicomplete multipartite digraphs. It was shown in [5] that the Hamiltonian cycle problem can be solved in polynomial time for semicomplete multipartite digraphs. The algorithm is very complicated and requires several nontrivial steps. Recently, the third author found another polynomial algorithm for the more general problem of finding a cycle covering a prescribed set of vertices in a semicomplete multipartite digraph [19], but that algorithm is also very complex.

In this paper we study the MSSS problem for another class of digraphs, where the Hamiltonian cycle problem is solvable in polynomial time, but so far no elementary algorithm is known for this problem. This is the class of quasi-transitive digraphs. These digraphs have a nice, recursive structure [6]; see Theorem 3.4. They can be decomposed into disjoint induced subdigraphs each of which is either a transitive digraph or a semicomplete digraph. Using the structure theorem for quasi-transitive digraphs, Gutin [12] proved that the Hamiltonian cycle problem is polynomially time solvable for quasi-transitive digraphs. His approach involves solving the problem of finding a minimum path cover of a quasi-transitive digraph (via a recursive algorithm; see, e.g., [2, section 5.9]).

The structure of quasi-transitive digraphs is closely related to that of extended semicomplete digraphs—in particular, in the case of Hamiltonian cycles and strong spanning subdigraphs (see, e.g., Theorem 3.5 and section 5.). Due to their recursive structure, quasi-transitive digraphs also have nice algorithmic properties, as we illustrate in this paper (see also [2, section 5.9]). However, quasi-transitive digraphs are also of interest because their underlying undirected graphs are exactly the comparability graphs [8]. Comparability graphs (also known as those graphs that allow a transitive orientation) have been widely studied in the literature; see, e.g., [10].

We give a lower bound for the number of arcs in any MSSS of an arbitrarily given strong quasi-transitive digraph. This bound can be calculated in polynomial time using Gutin's algorithm for finding a Hamiltonian cycle in a quasi-transitive digraph. We prove that this lower bound is also attainable for quasi-transitive digraphs [12]. The proof of this uses a new characterization of a longest cycle in an extended semicomplete digraph.

In the last section we point out that our methods imply that the MSSS problem can be solved efficiently for a much larger superclass of semicomplete digraphs than just quasi-transitive digraphs. The interested reader is encouraged to consult [2] for much more information on the interrelationship between various classes of generalizations of tournaments.

2. Terminology.

We refer to [2, Chap. 1] for a general account of the terminology on digraphs. We shall always use the number n to denote the number of vertices in the digraph currently under consideration. Digraphs are finite and have no loops or multiple arcs. We use $V(D)$ and $A(D)$ to denote the vertex set and the arc set of a digraph D. We use $|D|$ (instead of $|V(D)|$) to denote the number of vertices in D. The arc from a vertex x to a vertex y will be denoted by xy. If xy is an arc, then we say that x *dominates* y and y is *dominated* by x. For disjoint subsets $H, K \subset V(D)$ we use the notation $H \Rightarrow K$ to denote that there are no arcs from K to H.

By a *cycle (path,* respectively) we mean a directed (simple) cycle (path, respectively). If R is a cycle or a path with two vertices u, v such that u can reach v on R, then $R[u, v]$ denotes the subpath of R from u to v. A cycle (path) of a digraph D is *Hamiltonian* if it contains all the vertices of D. A digraph is *Hamiltonian* if it has a Hamiltonian cycle.

An (x, y)-*path* is a path from x to y. A digraph D is *strongly connected* (or just *strong*) if there exists an (x, y)-path and a (y, x)-path for every choice of distinct vertices x, y of D. Let U, W be disjoint subsets of $V(D)$. A (U, W)-path is a path $x_1x_2 \dots x_k$ such that $x_1 \in U, x_k \in W$, and no other x_i belongs to $U \cup W$.

A digraph T is *semicomplete* if it has no pair of nonadjacent vertices. A *tournament* is a semicomplete digraph with no cycles of length 2. It is well known and easy to prove that every semicomplete digraph has a Hamiltonian path and that every strong semicomplete digraph has a Hamiltonian cycle. A digraph $D = (V, A)$ is *quasi-transitive* if, for any distinct $x, y, z \in V$, the arcs $xy, yz \in A$ imply that there exists an arc between x and z, i.e., $xz \in A$ or $zx \in A$ or both.

Let $D = (V, A)$ be a digraph. Let $U \subseteq V$ and let $W = (V', A')$ be a subdigraph of D. We say that W *covers* U if $U \subseteq V'$.

A collection $\mathcal{F}$ of pairwise vertex disjoint paths and cycles of a digraph D is called a k-*path-cycle factor* of D if $\mathcal{F}$ covers $V(D)$ and has exactly $k \geq 0$ paths. If $\mathcal{F}$ has $k = 0$ paths, we call it a *cycle factor* and, similarly, if $\mathcal{F}$ has no cycles, it is called a k-*path factor.* A *cycle subdigraph* is a collection of vertex disjoint cycles. The *path covering number* of a digraph D, denoted $pc(D)$, is the smallest k for which D has a k-path factor.

Let D be a digraph on p vertices $v_1, \dots, v_p$ and let $L_1, \dots, L_p$ be a disjoint collection of digraphs. Then $D[L_1, \dots, L_p]$ is the new digraph obtained from D by replacing each vertex v_i of D by L_i and adding an arc from every vertex of L_i to every vertex of L_j if and only if v_iv_j is an arc of D $(1 \leq i \neq j \leq p)$. Let D and R be digraphs. Then D is an *extension of* R if there is a decomposition $D = R[I_{a_1}, \dots, I_{a_r}]$, $r = |V(R)|$, such that each I_{a_i} induces an independent set in D. An *extended semicomplete digraph* is a digraph which is an extension of a semicomplete digraph. Two vertices x and y in an extended semicomplete digraph $D = R[I_{a_1}, \dots, I_{a_r}]$ are said to be *similar* if $x, y \in I_{a_j}$ for some j.

Note that in the rest of the paper, whenever we consider a digraph with a decomposition $D = R[L_1, \dots, L_{|R|}]$, we shall think of each L_i as both a subset of $V(D)$ and a subdigraph of D. Furthermore, we also think of R as a subdigraph of D.

3. Results from other papers. In this section we list a number of results which we will use in the next sections.

LEMMA 3.1 (see [18]). *Let $D = (V, A)$ be a digraph which has no cycle factor. Then the vertices of D can be partitioned into disjoint sets Y, Z, R_1, R_2 such that the following hold:*

1. *$D\langle Y\rangle$ has no arcs.*
2. *$R_1 \Rightarrow Y \cup R_2$ and $Y \Rightarrow R_2$.*
3. *$|Z| < |Y|$.*

THEOREM 3.2 (see [11]). *A strong extended semicomplete digraph D is Hamiltonian if and only if it has a cycle factor. Furthermore, the length of a longest cycle in D is equal to the maximum number of vertices in a cycle subdigraph of D.*

THEOREM 3.3 (see [11]). *A longest cycle of an extended semicomplete digraph can be found in time $O(n^{\frac{5}{2}})$.*

Theorem 3.4 (see [6]). *Let D be a quasi-transitive digraph on at least two vertices. Then the following hold:*

1. *If D is not strong, then D can be decomposed as $D = T[W_1, W_2, \dots, W_{|T|}]$, where T is a transitive digraph with $|T| \geq 2$ and each W_i is a strong quasi-transitive digraph.*
2. *If D is strong, then D can be decomposed as $D = S[W_1, W_2, \dots, W_{|S|}]$, where S is semicomplete with $|S| \geq 2$ and each W_i is either a single vertex or a nonstrong quasi-transitive digraph. Furthermore, if $s_i s_j s_i$ is a cycle of S, then the corresponding W_i, W_j both have just one vertex.*

The following characterization of Hamiltonian quasi-transitive digraphs is given implicitly in [12].

Theorem 3.5 (see [12]). *Let D be a strongly connected quasi-transitive digraph with decomposition $D = S[W_1, W_2, \dots, W_s]$, where $s = |S|$. Let $pc(W_i)$ be the path covering number of the quasi-transitive digraph W_i, $i = 1, 2, \dots, s$. Let $D_0 = S[H_1, H_2, \dots, H_s]$ be the extended semicomplete digraph obtained by deleting all arcs inside each W_i (that is, $|H_i| = |W_i|$). Then D is Hamiltonian if and only if D_0 has a cycle subdigraph which covers at least $pc(W_i)$ vertices of H_i, $i = 1, 2, \dots, s$.*

Theorem 3.6 (see [12]). *The path covering number $pc(D)$ of a quasi-transitive digraph D can be calculated and a path cover with $pc(D)$ paths constructed in time $O(n^4)$.*

Theorem 3.7 (see [12]). *There is an $O(n^4)$ algorithm which, given a quasi-transitive digraph D, either returns a Hamiltonian cycle in D or a proof that no such cycle exists in D.*

Theorem 3.8 (see [6]). *A quasi-transitive digraph $D = S[W_1, W_2, \dots, W_{|S|}]$ is Hamiltonian if and only if it has a cycle factor $\mathcal{C}$ such that no cycle of $\mathcal{C}$ is a cycle of some $D\langle W_i \rangle$.*

4. Longest cycles in extended semicomplete digraphs.

In this section we prove a new characterization of a longest cycle in an extended semicomplete digraph. Besides being a very useful tool in our proof of the main result in the next section, this characterization is also of independent interest. In particular, it implies that, up to switching similar vertices, there is only one longest cycle in an extended semicomplete digraph.

Lemma 4.1. *Let D be an extended semicomplete digraph with an independent set I. If $\mathcal{C}$ is a cycle subdigraph covering I, then D contains one cycle C which covers I. Furthermore, given $\mathcal{C}$ and I, we can find one cycle covering I in time $O(n)$.*

Proof. By discarding some cycles if necessary, we may assume that every cycle in $\mathcal{C}$ contains a vertex from I. If $\mathcal{C}$ contains at least two cycles, then let C, C' be distinct cycles from $\mathcal{C}$. Let $x \in V(C), y \in V(C')$ be chosen such that $x, y \in I$. Let x^+, y^+ be the successors of x, y on C, C', respectively. Then xy^+ and yx^+ are arcs of D, since x and y are similar, and hence $C[x^+, x]C'[y^+, y]x^+$ is a cycle containing precisely the vertices of $V(C) \cup V(C')$. Now the first claim follows easily by induction on the number of cycles in $\mathcal{C}$. The complexity claim follows from the fact that we can merge the two cycles C, C' in constant time. □

Lemma 4.2. *If D is an acyclic extended semicomplete digraph, then we have $pc(D) = \max\{|I| : I$ is an independent set in $D\}$. Furthermore, starting from D, one can obtain a path cover with $pc(D)$ paths by removing the vertices of a longest path $pc(D)$ times.*

Proof. Let $D = S[H_1, H_2, \dots, H_s]$ be the (unique) decomposition of D such that $H_1, H_2, \dots, H_s$ are independent sets and let k denote the size of a largest independent set in D. Since S is semicomplete, it has a Hamiltonian path P, and since D is acyclic, P is also a longest path in D. Note that since D is acyclic, P contains precisely one vertex from each H_i. Now the claim follows by induction on k. $\square$

The following lemma is a special case of a more general result for semicomplete multipartite graphs due to Ayel (see [14]). Note that it also follows from Theorems 3.2 and 4.4.

LEMMA 4.3. *Let D be a strong extended semicomplete digraph and let C be a longest cycle in D. Then $D - C$ is acyclic.*

The following characterization of a longest cycle in a strong extended semicomplete digraph is a generalization of Theorem 3.2.

THEOREM 4.4. *Let D be a strong extended semicomplete digraph with decomposition $D = S[H_1, H_2, \ldots, H_t]$, $t = |S|$. Let m_i, $i = 1, 2, \ldots, t$, denote the maximum number of vertices from H_i which are contained in a cycle subdigraph of D. Then every longest cycle of D contains precisely m_i vertices from each H_i, $i = 1, 2, \ldots, t$.*

Proof. Let C be a longest cycle and suppose without loss of generality that C does not use m_1 vertices from H_1. Let m'_1 be the number of vertices from H_1 which are contained in C. First observe that C contains at least one vertex from each H_i. Indeed, if this is not the case, then choose i so that C has no vertex from H_i. Let x be an arbitrary vertex of H_i. If x has arcs to and from C in D, then it is easy to see that x can be inserted between two vertices of C, contradicting the maximality of C. Suppose without loss of generality that $V(C) \Rightarrow x$. Since D is strong, there is an $(x, V(C))$-path $xq_1q_2 \ldots q_t$ in D. Let q_t^- be the predecessor of q_t on C. Then $C[q_t, q_t^-]xq_1q_2 \ldots q_t$ is a cycle in D, contradicting the maximality of C. It follows that $1 \leq m'_1 < m_1$.

By the definition of m_1 and Lemma 4.1, there is some cycle Q which uses m_1 vertices from H_1. Since all vertices in H_1 have the same adjacencies and $m'_1 < m_1$, we can choose Q so that it contains all vertices from H_1 that are on C and at least one extra vertex $x \in H_1 - V(C)$. We will also choose Q so that under the assumption above, $|V(Q) \cap V(C)|$ is maximized.

We claim that for every i such that $H_i \cap V(Q) \not\subset V(C)$ we have $H_i \cap V(C) \subset V(Q)$. If this is not the case, then let u be a vertex of H_i which is on Q but not on C and v a vertex of H_i which is on C but not on Q. Since u and v are similar, we can replace u by v and obtain a new cycle Q' containing m_1 vertices of H_1 which has a larger intersection with C, contradicting the choice of Q above.

Now consider the digraph $D' = D\langle V(C) \cup V(Q)\rangle$. It follows from the fact that C has a vertex from each H_i and that all vertices in H_i are similar that the digraph D' is strong. We claim that D' has a cycle factor. If this is not the case, then we can apply Lemma 3.1 to get a partition Y', Z', R'_1, R'_2 of $V(D')$ satisfying the conditions of the lemma. It follows from the structure of the arcs determined in Lemma 3.1 that every cycle through a vertex in Y' must use a vertex of Z'. Hence there can be no cycle factor which covers all the vertices in Y'. Since Y' is an independent set in the extended semicomplete digraph D' and hence in D, we have $Y' \subset H_i$ for some i.

For every i such that $H_i \cap V(Q) \not\subset V(C)$ we argued above that all vertices in $H_i \cap V(D')$ are on the cycle Q. Hence we cannot have $Y' \subset H_i$ for any of these sets. On the other hand, for every j such that $H_j \cap V(Q) \subset V(C)$, we have all vertices of $H_j \cap V(D')$ on the cycle C. Hence Y' cannot be a subset of H_j either, implying that a partition Y', Z', R'_1, R'_2 of $V(D')$ satisfying the conditions of Lemma 3.1 does not exist.

Thus we have shown that the strong extended semicomplete subdigraph D' of D

has a cycle factor. By Theorem 3.2, D' has a Hamiltonian cycle C'. Now we obtain a contradiction to the assumption that C was a longest cycle in D. □

5. Smallest spanning strong subdigraphs of quasi-transitive digraphs. For an arbitrary quasi-transitive digraph D and a natural number k, we define the quasi-transitive digraph $H_k(D)$ obtained from D as follows: Add two sets of k new vertices $x_1, x_2, \ldots, x_k, y_1, y_2, \ldots, y_k$. Add all possible arcs from $V(D)$ to x_i along with all possible arcs from y_i to $V(D)$, $i = 1, 2, \ldots, k$. Finally, add all arcs of the kind $x_i y_j$, $i, j = 1, 2, \ldots, k$. Note that $H_0(D) = D$.

DEFINITION 5.1. *Let D be a strong quasi-transitive digraph and let $\epsilon(D)$ be the smallest $k \geq 0$ such that $H_k(D)$ is Hamiltonian.*

Observe that if $\epsilon(D) \geq 1$, then $\epsilon(D)$ is precisely the path covering number of D. Hence we can calculate $\epsilon(D)$ in time $O(n^4)$ using the algorithms of Theorems 3.6 and 3.7. We show below that $n + \epsilon(D)$ is a lower bound for the number of arcs in every spanning strong subdigraph of D.

LEMMA 5.2. *For every strongly connected quasi-transitive digraph D, every spanning strong subdigraph of D has at least $n + \epsilon(D)$ arcs.*

Proof. Let D be a strong quasi-transitive digraph with decomposition $D = S[W_1, W_2, \ldots, W_s]$, $s = |S| \geq 2$ (compare with Theorem 3.4). Suppose D has a spanning strong subdigraph D' with $n + k$ arcs. We may assume (by deleting some arcs if necessary) that no proper subdigraph of D' is spanning and strong. It is easy to prove by induction on k that D' can be decomposed into a cycle $P_0 = C$ and k arc-disjoint paths or cycles $P_1, P_2, \ldots, P_k$ with the following properties (where D_i denotes the digraph with vertices $\bigcup_{j=0}^{i} V(P_j)$ and arcs $\bigcup_{j=0}^{i} A(P_j)$ for $i = 0, 1, \ldots, t$):

1. For each $i = 1, \ldots, t$: If P_i is a cycle, then it has precisely one vertex in common with $V(D_{i-1})$. Otherwise the end-vertices of P_i are distinct vertices of $V(D_{i-1})$ and no other vertex of P_i belongs to $V(D_{i-1})$.
2. $\bigcup_{j=0}^{t} A(P_j) = A(D')$.

It is easy to see that this decomposition can be started with P_0 as any cycle in D'. It follows that we may choose $C = P_0$ so that

$$V(C) \not\subseteq W_i \text{ for } i = 1, 2, \ldots, s. \tag{5.1}$$

Now consider D' as a subdigraph of $H_k(D)$. By the minimality assumption on D', each P_i has length at least two. It follows that $H_k(D)$ has a cycle factor consisting of C and k cycles of the form $y_i P'_i x_i y_i$, $i = 1, 2, \ldots, k$, where P'_i is the path one obtains from P_i by removing the vertices it has in common with $V(D_{i-1})$ (defined above). By (5.1) and Theorem 3.8, $H_k(D)$ has a Hamiltonian cycle and hence $\epsilon(D) \leq k$. □

In fact, it is easy to see that the lower bound in Lemma 5.2 is valid for any digraph D (but may not be very useful, as calculating $\epsilon(D)$ is NP-hard for general digraphs).

Below we characterize the optimal solution to the MSSS problem for quasi-transitive digraphs and show that the problem is polynomially solvable.

THEOREM 5.3. *The MSSS of a quasi-transitive digraph has precisely $n + \epsilon(D)$ arcs. Furthermore, we can find such a subdigraph in time $O(n^4)$.*

Proof. Let $D = S[W_1, W_2, \ldots, W_s]$, $s = |S| \geq 2$, be a strong quasi-transitive digraph. Using the algorithm of Theorem 3.7 we can check whether D is Hamiltonian and find a Hamiltonian cycle if one exists. If D is Hamiltonian, then any Hamiltonian cycle is the optimal spanning strong subdigraph. Suppose below that D is not Hamiltonian.

Let $D_0 = S[H_1, H_2, \ldots, H_s]$ be the extended semicomplete digraph one obtains by deleting all arcs inside each W_i (that is, $|H_i| = |W_i|$ and H_i is obtained from W_i

by deleting all arcs). By Theorem 3.5, D_0 has no cycle subdigraph which covers at least $pc(W_i)$ vertices of each H_i, $i = 1, 2, \dots, s$.

For each $i = 1, 2, \dots, s$, let m_i denote the maximum number of vertices which can be covered in H_i by any cycle subdigraph of D_0. According to Theorem 4.4 every longest cycle C in D_0 contains exactly m_i vertices from H_i, $i = 1, 2, \dots, s$. By Theorem 3.3 we can find C in time $O(n^{\frac{5}{2}})$. Let

$$k = \max\{pc(W_i) - m_i : i = 1, 2, \dots, s\}. \tag{5.2}$$

Define the extended semicomplete subdigraph D^* of D as $D^* = S[H_1^*, H_2^*, \dots, H_s^*]$, where H_i^* is an independent set containing $m_i^* = \max\{pc(W_i), m_i\}$ vertices, $i = 1, 2, \dots, s$. Since vertices inside an independent set are similar, we may think of C as a longest cycle in D^* (i.e., C contains precisely m_i vertices from H_i^*, $i = 1, 2, \dots, s$). By Lemmas 4.2 and 4.3, $D^* - C$ can be covered by k paths $P_1^*, P_2^*, \dots, P_k^*$. Since $D^* - C$ is acyclic, we may assume (by Lemma 4.2) that P_1^* starts at a vertex x and ends at a vertex y such that x has in-degree zero and y has out-degree zero in $D^* - C$. It follows that there is an arc cx from C to x and an arc yc' from y to C in D^*, and hence we can glue P_1^* onto C by adding the arcs cx, yc'. Remove P_1^* and its vertices and consider the remaining paths. It follows by induction on k that adding $P_2^*, P_3^*, \dots, P_k^*$ one by one, using two new arcs each time, we can obtain a spanning strong subdigraph D^{**} of D^* with $|V^*| + k$ arcs.

Now we obtain a spanning strong subdigraph of the quasi-transitive digraph D as follows: Since $m_i^* \geq pc(W_i)$ for $i = 1, 2, \dots, s$, each W_i contains a collection of $t_i = m_i^*$ paths $P_{i1}, P_{i2}, \dots, P_{it_i}$ such that these paths cover all vertices of W_i. Such a collection of paths can easily be constructed from a given collection of $pc(W_i)$ paths which cover $V(W_i)$. Let $x_{i1}, x_{i2}, \dots, x_{it_i}$ be the vertex set of H_i^*. Replace x_{ij} in D^{**} by the path P_{ij} for each $i = 1, 2, \dots, s$, $j = 1, 2, \dots, t_i$. We obtain a spanning strong subdigraph D' of D. The number of arcs in D' is

$$\begin{aligned} A(D') &= \sum_{i=1}^{s}(|W_i| - m_i^*) + (|V^*| + k) \\ &= (n - |V^*|) + (|V^*| + k) \\ &= n + k. \end{aligned} \tag{5.3}$$

It remains to argue that D' is smallest possible. By Lemma 5.2, it suffices to prove that $\epsilon(D) \geq k$.

Suppose $\epsilon(D) = r < k$. By Definition 5.1, the quasi-transitive digraph $H_r(D)$ has a Hamiltonian cycle C. It follows from the definition of $H_r(D)$ that we can decompose $H_r(D)$ as $H_r(D) = S'[W_1, W_2, \dots, W_s, I_r, I_r]$, where I_r is an independent set of r vertices and S' is obtained from S by adding two new vertices x, y such that xy is an arc, x is dominated by all vertices of S, and y dominates all vertices of S. Let C' be obtained by contracting each subpath of C which lies entirely inside some W_i. Now delete all remaining arcs inside each W_i. The resulting digraph T is extended semicomplete and has a decomposition $T = S'[I_{a_1}, I_{a_2}, \dots, I_{a_s}, I_r, I_r]$, where each I_{a_j} denotes an independent set on $a_j \geq 1$ vertices. Since inside every W_i we contracted only subpaths of C, it follows that $a_i \geq pc(W_i)$ for $i = 1, 2, \dots, s$. Furthermore, C' is a Hamiltonian cycle in T.

Remove the vertices $x_1, x_2, \dots, x_r, y_1, y_2, \dots, y_r$ from C'. As the only arcs leaving each x_i go to $\{y_1, y_2, \dots, y_r\}$, this gives us a collection of r paths $P_1, P_2, \dots, P_r$ that covers all vertices in $T^* = S[I_{a_1}, I_{a_2}, \dots, I_{a_s}]$. Since all vertices inside the same independent set are similar, we can assume that $P_1, P_2, \dots, P_r$ are paths in D_0 (D_0 was defined in the beginning of the proof). Let i be chosen such that

$$pc(W_i) - m_i = k. \tag{5.4}$$

Since $a_i \geq pc(W_i)$ and $r < k$, it follows that some P_j contains two vertices of H_i. Note that if $P_j = z_1 z_2 \dots z_p$ and $a < b$ are indices so that z_a and z_b are similar, then $z_{a+1} \dots z_{b-1} z_b z_{a+1}$ is a cycle and $z_a z_{b+1}$ is an arc if $b < p$. Thus we can replace P_j by a cycle and a path $P'_j = P_j[z_1, z_a]P_j[z_{b+1}, z_p]$. Clearly, we can continue this way (replacing paths in the current collection by a cycle and a path) until every path in the current collection contains at most one vertex from H_i. This shows that D_0 has a cycle subdigraph which covers at least $a_i - r \geq pc(W_i) - r > pc(W_i) - k = m_i$ vertices from H_i. However, this contradicts the definition of m_i. This contradiction shows that $\epsilon(D) \geq k$ and that the optimality of D' follows from Lemma 5.2.

The proof above can be easily turned into an algorithm which finds an MSSS of a given quasi-transitive digraph D. The complexity of the algorithm is dominated by the time it takes to find an optimal path cover in each W_i. By Theorem 3.6 this can be done in $O(n^4)$ time. □

6. Remarks and open problems. In order to speed up the algorithm implied by the proof of Theorem 5.3, one would need to find a faster algorithm for finding a Hamiltonian cycle in a quasi-transitive digraph. One approach (following Gutin's idea in [12]) would be to find a faster algorithm for the path cover number of quasi-transitive digraphs. This, as well as finding a completely different method for solving the Hamiltonian cycle problem in quasi-transitive digraphs, seems to be a challenging open problem.

Another paper that makes good use of the nice recursive structure of quasi-transitive digraphs is [4] in which the problem of finding a heaviest cycle (with respect to weights on the vertices) was solved for quasi-transitive digraphs. See also [3].

Below we point out that the proofs of our theorems imply a polynomial time algorithm for a much larger class of digraphs than just quasi-transitive digraphs.

By using the approach used in this paper, it is not difficult to prove the following extension of Theorem 3.5.

THEOREM 6.1. *Let D be a strong digraph with decomposition $D = S[W_1, W_2, \dots, W_s]$, where $s = |S|$, W_i is an arbitrary digraph, $i = 1, 2, \dots, s$, and S is a strong semicomplete digraph on $s \geq 2$ vertices. Let $pc(W_i)$ be the path covering number of the digraph W_i, $i = 1, 2, \dots, s$. Let $D_0 = S[H_1, H_2, \dots, H_s]$ be the extended semicomplete digraph obtained by deleting all arcs inside each W_i (that is, $|H_i| = |W_i|$). Then D is Hamiltonian if and only if D_0 has a cycle subdigraph which covers at least $pc(W_i)$ vertices of H_i, $i = 1, 2, \dots, s$.*

For every natural number t, let ψ_t be the class of all digraphs for which an optimal path cover can be found in polynomial time $O(n^t)$. For every natural number t, let ϕ_t be the class of all digraphs of the form $D = S[H_1, H_2, \dots, H_s]$, $s = |S| \geq 2$, where S is a strong semicomplete digraph and $H_i \in \psi_t$, $i = 1, 2, \dots, s$. By Theorem 3.6 the class ϕ_4 contains all quasi-transitive digraphs.

Gutin's approach to solving the Hamiltonian cycle problem for quasi-transitive digraphs easily extends to a proof of the following.

THEOREM 6.2. *For every natural number t, the Hamiltonian cycle problem is solvable in time $O(n^{\max\{3,t\}})$ for digraphs that belong to ϕ_t.*

Let $D = S[H_1, H_2, \dots, H_s]$ be a digraph in ϕ_t. To find the MSSS in D, let D' be the extended semicomplete digraph obtained from D by deleting all arcs within each H_i for $i = 1, 2, \ldots, s$. By Theorem 3.3, we can find a longest cycle C in D'. Let $m_i = |V(H_i) \cap V(C)|$ for $i = 1, 2, \ldots, s$ and let

$$k = \max\{pc(H_i) - m_i : \ i = 1, 2, \ldots, s\}.$$

Using a proof analogous to that of Theorem 5.3, we can show that the MSSS of D contains $n + k$ arcs when $k \geq 1$ and is a Hamiltonian cycle when $k \leq 0$. Combining this with Theorems 6.1 and 6.2 we get the following.

THEOREM 6.3. *For every natural number t, the MSSS problem is solvable in time $O(n^{\max\{3,t\}})$ for all digraphs in ϕ_t.*

REFERENCES

[1] A.V. AHO, M.R. GAREY, AND J.D. ULLMAN, *The transitive reduction of a directed graph*, SIAM J. Comput., 1 (1972), pp. 131–137.

[2] J. BANG-JENSEN AND G. GUTIN, *Digraphs: Theory, Algorithms and Applications*, Springer-Verlag London, Ltd., London, 2001.

[3] J. BANG-JENSEN AND G. GUTIN, *On the complexity of Hamiltonian path and cycle problems in certain classes of digraphs*, Discrete Appl. Math., 95 (1999), pp. 41–60.

[4] J. BANG-JENSEN AND G. GUTIN, *Vertex heaviest paths and cycles in quasi-transitive digraphs*, Discrete Math., 163 (1996), pp. 217–223.

[5] J. BANG-JENSEN, G. GUTIN, AND A. YEO, *A polynomial algorithm for the Hamiltonian cycle problem in semicomplete multipartite digraphs*, J. Graph Theory, 29 (1998), pp. 111–132.

[6] J. BANG-JENSEN AND J. HUANG, *Quasi-transitive digraphs*, J. Graph Theory, 20 (1995), pp. 141–161.

[7] J. BANG-JENSEN AND A. YEO, *The minimum spanning strong subdigraph problem for extended semicomplete digraphs and semicomplete bipartite digraphs*, J. Algorithms, 41 (2001), pp. 1–19.

[8] A. GHOUILÀ-HOURI, *Caractérisation des graphes non orientés dont on peut orienter les arrêtes de manière à obtenir le graphe d'un relation d'ordre*, C.R. Acad. Sci. Paris, 254 (1962), pp. 1370–1371.

[9] P. GIBBONS, R. KARP, V. RAMACHANDRAN, D. SOROKER, AND R. TARJAN, *Transitive compaction in parallel via branchings*, J. Algorithms, 12 (1991), pp. 110–125.

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

[11] G. GUTIN, *Cycles and paths in complete multipartite digraphs, theorems and algorithms: A survey*, J. Graph Theory, 19 (1995), pp. 481–505.

[12] G. GUTIN, *Polynomial algorithms for finding Hamiltonian paths and cycles in quasi-transitive digraphs*, Australas. J. Combin., 10 (1994), pp. 231–236.

[13] H.T. HSU, *An algorithm for finding a minimal equivalent graph of a digraph*, J. Assoc. Comput. Mach., 22 (1975), pp. 11–16.

[14] B. JACKSON, *Long paths and cycles in oriented graphs*, J. Graph Theory, 5 (1981), pp. 145–157.

[15] S. KHULLER, B. RAGHAVACHARI, AND N. YOUNG, *Approximating the minimum equivalent digraph*, SIAM J. Comput., 24 (1995), pp. 859–872.

[16] S. KHULLER, B. RAGHAVACHARI, AND N. YOUNG, *On strongly connected digraphs with bounded cycle length*, Discrete Appl. Math., 69 (1996), pp. 281–289.

[17] K. SIMON, *Finding a minimal transitive reduction in a strongly connected digraph within linear time*, in Graph Theoretic Concepts in Computer Science (Kerkrade, 1989), Springer-Verlag, Berlin, 1990, pp. 244–259.

[18] A. YEO, *How close to regular must a semicomplete multipartite digraph be to secure Hamiltonicity?* Graphs Combin., 15 (1999), pp. 481–493.

[19] A. YEO, *A polynomial time algorithm for finding a cycle covering a given set of vertices in a semicomplete multipartite digraph*, J. Algorithms, 33 (1999), pp. 124–139.

SIAM J. Discrete Math.
Vol. 16, No. 3, pp. 345–353

A NEW LOWER BOUND FOR TREE-WIDTH USING MAXIMUM CARDINALITY SEARCH[*]

BRIAN LUCENA[†]

Abstract. The tree-width of a graph is of great importance in applied problems in graphical models. The complexity of inference problems on Markov random fields is exponential in the tree-width of the graph. However, computing tree-width is NP-hard in general. Easily computable upper bounds exist, but there are few lower bounds. We give a novel technique to compute a lower bound for the tree-width of a graph using maximum cardinality search. This bound is efficiently computable and is guaranteed to do at least as well as finding the largest clique in the graph.

Key words. tree-width, lower bound, triangulation, maximum cardinality search

AMS subject classifications. 68R10, 62-09, 05C85

PII. S0895480101397773

1. Introduction.

The *tree-width* of a graph is a well-known quantity and one that is quite important in several applied areas. When computing on Markov random fields, the tree-width of the associated graph determines the computational complexity of inference problems such as finding the most likely configuration of the variables or the marginal distributions of the variables. The complexity of such tasks is exponential in the tree-width of the graph. For this reason and others, people are interested in calculating tree-width.

It is well known that finding the tree-width of an arbitrary graph is an NP-hard problem [3]. So the best we can hope for in practice is to find useful bounds. Upper bounds have been relatively easy to find. Choose an ordering of the vertices of G, and compute the size of the largest border incurred while progressing through the ordering [5], [8]. This gives an upper bound for the tree-width of G. Another method is to choose a triangulation of G, find the size of the largest clique, and subtract one.

Lower bounds have been more difficult to find. One bound, given in [9], is the minimum over all nonadjacent pairs of vertices of the maximum degree of the pair. This bound stems from the fact that a k-tree has at least two nonadjacent vertices of degree k, and so any subgraph of a k-tree must have a pair of nonadjacent vertices of degree $\leq k$. There are other commonplace lower bounds, such as the one less than the size of the largest clique in G.

In this paper we show that a procedure called *maximum cardinality search* can be used to determine a lower bound for the tree-width of a graph. This bound, although still weak in many situations, always does at least as well as finding the biggest clique in the graph and typically beats the bound in [9].

Maximum cardinality search (MCS) is described concisely in the following manner.

Give number 1 to an arbitrary node. Number the nodes consecutively, choosing as the next to number an unnumbered node with a maximum number of previously numbered neighbors. Break ties arbitrarily [7].

[*] Received by the editors November 8, 2001; accepted for publication (in revised form) January 8, 2003; published electronically April 30, 2003. This work was supported by a National Science Foundation Graduate Research Fellowship and NSF grant DMS-0074276.

http://www.siam.org/journals/sidma/16-3/39777.html

[†] Division of Applied Mathematics, Box F, Brown University, Providence, RI 02912. Current address: Department of Statistics, University of Washington, Box 35422, Seattle, WA 98195-4322 (lucena@stat.washington.edu).

The somewhat surprising result is that during this process the "number of previously numbered neighbors" of an unnumbered node gives a lower bound to the tree-width of the entire graph.

2. Tree width. Throughout this paper, all graphs will be undirected and contain no self-loops or multiple edges. We denote the *neighbors* of a vertex v in graph G by $\Gamma_G(v)$ or just $\Gamma(v)$ if there is no ambiguity. So $\Gamma(v) = \{w \in V : (v,w) \in E\}$. The *family* of v is denoted $\bar{\Gamma}(v) = \Gamma(v) \cup \{v\}$. The degree of a vertex v, $deg(v) = |\Gamma(v)|$. By an ordering of the vertices of G, we mean a bijection $\pi : V_G \longrightarrow \{1, 2, \ldots, n\}$, where $n = |V_G|$. We sometimes represent the ordering π by the ordered sequence $(v_1, v_2, \ldots, v_n)$, which means $\pi^{-1}(i) = v_i$. We say $v <_\pi w$ to denote that $\pi(v) < \pi(w)$.

We begin by defining *k-trees* and the *tree-width* of a graph and summarizing some of their known properties.

DEFINITION 2.1. *A* k-tree *can be best defined recursively in the following way. First of all, the complete graph on $k+1$ vertices*[1] *is a k-tree. Second, given a k-tree on n vertices (for $n \geq k+1$), we can form a k-tree on $n+1$ vertices by connecting our new vertex to k existing vertices which form a complete subgraph in our n-vertex subgraph* [11].

We also have this alternate definition of a k-tree.

THEOREM 2.2 (see [11]). *The following are necessary and sufficient conditions for a graph G to be a k-tree:*

1. *G is connected.*
2. *G contains a $k+1$-clique*[1] *but no $k+2$ clique.*
3. *Every minimal x-y separator of G is a k-clique.*

Note that an *x-y separator* refers to a set of vertices S $(x, y \notin S)$ such that any path from vertex x to vertex y must pass through a vertex in S. It is defined only when x and y are nonadjacent.

DEFINITION 2.3. *Let G be a k-tree on n vertices, and let α be an ordering of the vertices of G. Let $v_i = \alpha^{-1}(i)$. We define $\mu_{\alpha,i}(G) = \Gamma(v_i) \cap \{v_1, v_2, \ldots, v_{i-1}\}$. We say that α is a* construction order *of G if $\{v_1, v_2, \ldots, v_k\}$ is a clique and $\mu_{\alpha,i}(G)$ is a clique of size k for all $k+1 \leq i \leq n$.*

It can be easily verified from the definitions that G is a k-tree if and only if there exists some ordering α such that α is a construction order of G.

DEFINITION 2.4. *A* partial k-tree *is a graph which is a subgraph of some k-tree.*

LEMMA 2.5 (see [11]). *Let H be a k-tree, and let $C = \{w_1, w_2, \ldots, w_k\}$ be a k-clique in H. Then there exists a construction order α on H such that if $v_i = \alpha^{-1}(i)$, then $\{v_1, \ldots, v_k\} = \{w_1, \ldots, w_k\}$. In other words, any clique can be the starting clique for the recursive process of building a k-tree given in the definition.*

DEFINITION 2.6. *The* tree width *of a graph G, denoted $TW(G)$, is defined as the smallest positive integer k for which G is a partial k-tree.*

The algorithm called *maximum cardinality search* is best known as a method to test whether a graph is triangulated [12]. To repeat, maximum cardinality search (MCS) works as follows. We form a *numbering* by first assigning the number 1 to an arbitrary vertex. Then given that we have numbered i vertices already, we give the number $i+1$ to the unnumbered vertex with the most neighbors in the set of already numbered vertices, breaking ties arbitrarily. Now define an ordering π which maps

[1] The cited definitions actually use k here instead of $k+1$. The only difference is whether a k-clique should be considered as both a k-tree and a $k-1$-tree, or just a $k-1$ tree. Here, following [4], we use the latter interpretation.

each vertex to its number. We say that π is an ordering generated by MCS, or more concisely, an MCS ordering. We will now give a more formal definition.

Definition 2.7. *Let $G = (V, E)$, and let $T \subset V$. For $v \in V$, let $d_T(v) = |\{w \in T : (v, w) \in E\}|$.*

Definition 2.8. *Let $G = (V, E)$ be a graph, and let π be an ordering of the vertices. Let $v_i = \pi^{-1}(i)$, and let $T_i = \{v_1, v_2, \dots, v_i\}$. If for all $i = 2, 3, \dots, n$ we have that $d_{T_{i-1}}(v_i) \geq d_{T_{i-1}}(v_j)$ for all $j = i+1, i+2, \dots, n$, then we say π is an* MCS ordering *on G.*

Definition 2.9. *An ordering π of the vertices of a graph G is said to be a* perfect elimination ordering *if for all $i = 1, 2, \dots, n$ we have that the set $A_i = \Gamma(v_i) \cap \{v_{i+1}, v_{i+2}, \dots, n\}$ is completely connected.*

MCS can be used to test whether a graph is triangulated in the following manner. Let G be a graph, and let π_1 be an MCS ordering on G. Now let π_2 be the *reverse* of π_1. That is, let $\pi_2(v) = n + 1 - \pi_1(v)$. Then G is triangulated if and only if π_2 is a perfect elimination ordering on G [12].

As we mentioned earlier, upper bounds to tree-width are relatively easy to find. Any ordering π of the vertices of a graph G determines an upper bound to $TW(G)$. This is done by computing the so-called fill-in F_π to form the elimination graph G^π. (See [2] or [12] for details.) One less than the size of the largest clique in G^π is an upper bound to $TW(G)$. In fact, an equivalent definition for $TW(G)$ is to take the minimum of that quantity over all orderings π [2].

However, since computing tree-width is NP-hard, there is no way to determine the optimal ordering π of the vertices, so in practice various heuristics are used. MCS has been proposed as one such heuristic (using the *reverse* of MCS orderings as they are defined in this paper) [10].

3. Main result. The main result in this paper is that MCS also gives a lower bound on the tree-width of a graph in the following manner.

Theorem 3.1. *Let $G = (V_G, E_G)$ be a graph on n vertices. Let $\pi = (v_1, v_2, \dots, v_n)$ be an MCS ordering on G. Then $TW(G) \geq deg(v_n)$.*

This is our main theorem, and the proof will follow shortly. First we show that the following corollary is an immediate consequence of Theorem 3.1.

Corollary 3.2. *Let G be a graph, and let $\pi = (v_1, v_2, \dots, v_n)$ be an MCS ordering on G. Let $T_i = \{v_1, v_2, \dots, v_i\}$. Then $TW(G) \geq \max_i d_{T_{i-1}}(v_i)$.*

Proof. Let $k = \max_i d_{T_{i-1}}(v_i)$ and $j = \arg\max_i d_{T_{i-1}}(v_i)$. Let H be the subgraph of G generated by the set of vertices T_j. The ordering $v_1, v_2, \dots, v_j$ is an MCS ordering for H, and in the graph H the vertex v_i has degree k. So by Theorem 3.1, $TW(H) \geq k$, which implies $TW(G) \geq k$. □

Before proceeding to the proof of Theorem 3.1 we prove a necessary lemma.

Lemma 3.3. *Let $G = (V, E)$ be a graph with $|V| = n$. Suppose we have a partition of the set V into three disjoint sets, $X \cup Y \cup S = V$, such that for any $x \in X$ and $y \in Y$, S is an x, y-separator. Let π be an ordering of the vertices generated by MCS, and let $w_i = \pi^{-1}(i)$. Let $T_i = \{w_1, w_2, \dots, w_i\}$ for $i = 1, 2, \dots, n$. Then*

$$|T_i \cap S| \geq \min\{\max_{v \in X - T_i} d_{T_i}(v), \max_{v \in Y - T_i} d_{T_i}(v)\}. \tag{3.1}$$

Proof. We will prove the lemma by induction on i. Consider first the base case $i = 1$. In order for the right-hand side of our inequality to be 1, w_1 must be adjacent to both a vertex in X and a vertex in Y. Clearly, such a vertex must be in S, which means the left-hand side is also 1. Otherwise, the right-hand side is 0 and the inequality holds trivially.

We will now assume that the inequality holds for i and prove that it must be true for $i+1$. So assume that (3.1) holds for i and recall that $T_{i+1} = T_i \cup \{w_{i+1}\}$. We will examine how the two sides of the inequality change as we go from i to $i+1$ under two cases.

Case 1. $w_{i+1} \in S$. Then the left-hand side increases by 1, and the right-hand side increases by at most 1. So the inequality still holds.

Case 2. $w_{i+1} \notin S$. So w_{i+1} cannot border both a vertex in X and a vertex in Y. Without loss of generality, assume that $w_{i+1} \in X$. This means that $d_{T_i}(w_{i+1}) \geq d_{T_i}(v)$ for all $v \in V - T_i$. So we have

$$\max_{v \in X - T_i} d_{T_i}(v) \geq \max_{v \in Y - T_i} d_{T_i}(v). \tag{3.2}$$

Since $w_{i+1} \in X$, we know that

$$\max_{v \in Y - T_i} d_{T_i}(v) = \max_{v \in Y - T_{i+1}} d_{T_{i+1}}(v). \tag{3.3}$$

Since the right-hand side of (3.1) is the min of two items, the smaller of which is not increasing, we can conclude that the right-hand side does not increase as we go from i to $i+1$. □

We are now ready to prove our main result.

Proof of Theorem 3.1. Let $k = deg(v_n)$. This will be a proof by contradiction. We assume that $TW(G) \leq k-1$ and go on to show that this is inconsistent with an ordering π that ends in v_n. Specifically, we will work for a long time to isolate a particular vertex v^* and a set of vertices D which separates v^* from the previously numbered vertex in our MCS. We use Lemma 3.3 to indicate that certain vertices in D must have already been numbered. We then derive a contradiction by showing that when another vertex, z, is numbered according to the ordering π, it in fact has fewer numbered neighbors than v_n, contradicting our assumption that π is an MCS ordering.

Let $w_1, w_2, \ldots, w_k$ be the k neighbors of v_n labeled such that $i < j \Rightarrow w_i <_\pi w_j$. If $TW(G) \leq k-1$, then there exists a $(k-1)$-tree $H = (V_G, E_H)$ such that $E_G \subseteq E_H$. Let i be the lowest index such that $\{w_{i+1}, w_{i+2}, \ldots, w_k, v_n\}$ form a clique in H. A $(k-1)$-tree cannot contain a $(k+1)$-clique, so that we know that $i \geq 1$ and $\{w_k, v_n\}$ form a clique of size 2 so we know that $i \leq k-1$. Therefore i exists and $1 \leq i \leq k-1$. By the definition of i we know that w_i is not adjacent to all of $\{w_{i+1}, w_{i+2}, \ldots, w_k\}$ (it is adjacent to v_n, of course). So let j be the smallest index in $i+1, \ldots, k$ such that $(w_i, w_j) \notin E_H$.

We will now define the following sets:

1. $C_1 = \{w_{i+1}, w_{i+2}, \ldots, w_{j-1}\}$.
2. $C_2 = \{w_l : j \leq l \leq k, (w_i, w_l) \notin E_H\}$.
3. $C_3 = \{w_l : j \leq l \leq k, (w_i, w_l) \in E_H\}$.

Note the following straightforward properties of these sets:

1. $C_1 \cap C_2 = C_1 \cap C_3 = C_2 \cap C_3 = \emptyset$.
2. $|C_1 \cup C_2 \cup C_3| = k - i$.
3. $v \in C_1 \cup C_2 \cup C_3 \Rightarrow v >_\pi w_i$.
4. $v \in C_1 \cup C_3 \Rightarrow (w_i, v) \in E_H$.
5. $v \in C_2 \Rightarrow (w_i, v) \notin E_H$.
6. $w_j \in C_2$.
7. $C_1 \cup C_2 \cup C_3 \cup \{v_n\}$ form a clique in H.

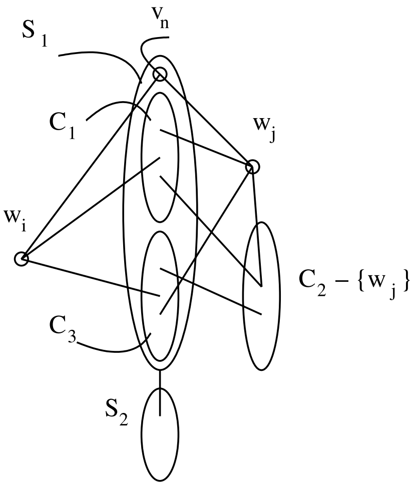


FIG. 1. *Lines demonstrate edges that must exist in H.*

Since $(w_i, w_j) \notin E_H$, by the separation property, we know that there exists a set of vertices S which form a $(k-1)$-clique in H such that any path from w_i to w_j in H must pass through a vertex in S. Choose such a set S. It is easy to see that $\{v_n\} \cup C_1 \cup C_3 \subseteq S$ since all of those vertices are adjacent to both w_i and w_j in H. Let $S_1 = C_1 \cup C_3 \cup \{v_n\}$, let $S_2 = S - S_1$, and let $c_2 = |C_2|$. So $|S_1| = k - i + 1 - c_2$ (see Figure 1).

Now let α be a construction order on H which starts with the clique S as its basis. By Lemma 2.5, such an ordering exists. Let v^* be the last element of C_2 with respect to the ordering α. In other words, for all $v \in C_2, v \neq v^*$, we have that $v <_\alpha v^*$. Since $w_j \in C_2$ and w_j is not in the basis for the construction order α, we know that v^* is not in the basis of α. So let $D =$ the $(k-1)$-clique that v^* is adjoined to when H is constructed using the construction order α.

CLAIM 3.4. *D is a w_i, v^*-separator.*

Proof. We know S is a w_i, v^*-separator. So consider any path from w_i to v^*. It uses some vertex in S. If that vertex is also in D, then clearly the path goes through D. Otherwise, since S is our basis in the construction order α, any path from a vertex in $S - D$ to v^* must go through D. So any path from w_i to v^* must go through D. □

Let $D_1 = S_1 \cup (C_2 - \{v^*\})$. Clearly, $D_1 \subset D$ since $v \in D_1 \Rightarrow (v, v^*) \in E_H$ and $v <_\alpha v^*$. Note that $|D_1| = k - i$. Let $D_2 = D - D_1$, $|D_2| = i - 1$ (see Figure 2).

Let T_1 be the set of vertices numbered before w_i. Since $\{w_1, \ldots, w_{i-1}\} \subset T_1$, we know that $d_{T_1}(v_n) = i - 1$. Since w_i is numbered next, it must have at least as many "numbered neighbors" as v_n. Therefore $d_{T_1}(w_i) \geq i - 1$. If the set D were removed from H, the resulting graph would be disconnected. Let Z be the connected component containing v^* in this disconnected graph. Let $Z_1 = Z - T_1$ so that Z_1 is the set of vertices in Z numbered after w_i. We know $v^*, w_j \in Z_1$ since they are both in C_2 and are therefore numbered after w_i (note that it is possible that $v^* = w_j$). Clearly, for any vertex $v \in Z_1$, $d_{T_1}(v) \leq d_{T_1}(w_i)$. Let $m = \max_{v \in Z_1} d_{T_1}(v)$. By Lemma 3.3 (with D separating Z from $V - \{Z \cup D\}$), we know at least m vertices of D must already be numbered. Let $D_2 = D - D_1$. Since $|D| = k-1$, $|D_1| = k - i$, and $D_1 \subseteq D$, we know that $|D_2| = i - 1$. Let $N = T_1 \cap D$; this is the set of vertices in D which are

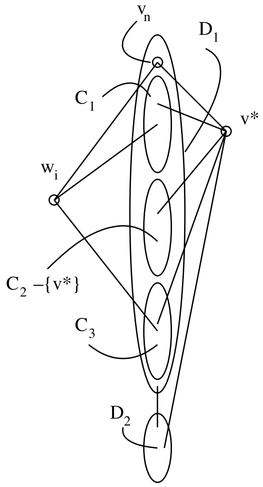


FIG. 2. *Lines demonstrate edges that must exist in H.*

numbered before w_i. So $|N| \geq m$. Furthermore, $N \subseteq D_2$ since $v \in D_1 \Rightarrow w_i <_\pi v$. In other words, $T_1 \cap D_1 = \emptyset$. So we can conclude that $m \leq |N| \leq |D_1| = i-1$.

Let $T_2 = T_1 \cup \{w_i\}$. This corresponds to the set of numbered vertices immediately after w_i is numbered. For all $v \in Z_1$ we know that $(v, w_i) \notin E_H$, which of course implies that $(v, w_i) \notin E_G$. So

$$\max_{v \in Z_1} d_{T_2}(v) = \max_{v \in Z_1} d_{T_1}(v) = m \leq i-1. \tag{3.4}$$

Meanwhile,

$$d_{T_2}(v_n) = i. \tag{3.5}$$

Now let z be the first vertex in $Z_1 - T_2$ to be numbered. So for all $v \in Z_1, v \neq z$, we have $z <_\pi v$. Let T_3 be the set of vertices numbered before z, i.e., $T_3 = \{v \in V : v <_\pi z\}$. By the assumption of the theorem, v_n comes last in the ordering, so we know that $v_n \notin T_3$. Since $z \in Z_1$, by (3.4) we know that

$$d_{T_2}(z) \leq m. \tag{3.6}$$

We have nearly obtained our contradiction. We know that

$$d_{T_3}(v_n) = d_{T_2}(v_n) + d_{T_3 - T_2}(v_n) = i + d_{T_3 - T_2}(v_n). \tag{3.7}$$

So we must have

$$d_{T_3}(z) \geq i + d_{T_3 - T_2}(v_n). \tag{3.8}$$

Clearly, $d_{T_3}(z) = d_{T_2}(z) + d_{T_3 - T_2}(z)$, and we know that $d_{T_2}(z) \leq m$. So we know that

$$d_{T_3 - T_2}(z) \geq i - m + d_{T_3 - T_2}(v_n). \tag{3.9}$$

We will obtain a contradiction of (3.9) by showing that

$$d_{T_3 - T_2}(z) \leq i - 1 - m + d_{T_3 - T_2}(v_n). \tag{3.10}$$

Since we chose z to be the first element of Z_1 to be numbered after w_i, it is clear that $(T_3 - T_2) \cap Z_1 = \emptyset$. Also we know that any vertex which borders z is either in Z or D. Therefore,

$$d_{T_3-T_2}(z) = d_{(T_3-T_2)\cap D}(z) = d_{(T_3-T_2)\cap D_1}(z) + d_{(T_3-T_2)\cap D_2}(z). \tag{3.11}$$

We know that $|D_2| = i - 1$ and $|T_2 \cap D_2| = m$. So we can assert that

$$d_{(T_3-T_2)\cap D_2}(z) \leq i - 1 - m. \tag{3.12}$$

Furthermore, since $(v_n, v) \in E_G$ for all $v \in D_1$ we know that

$$d_{(T_3-T_2)\cap D_1}(z) \leq d_{(T_3-T_2)\cap D_1}(v_n). \tag{3.13}$$

So we have that $d_{T_3-T_2}(z) \leq i - 1 - m + d_{T_3-T_2}(v_n)$, which contradicts (3.9) and proves the theorem. □

3.1. The MCS lower bound. To summarize, Corollary 3.2 states that in the process of the MCS, if an unnumbered vertex is numbered with m numbered neighbors, then the tree-width of G must be at least m. If we go through one iteration of MCS and keep track of the best bound acquired through that process, we get the MCS lower bound for that ordering.

DEFINITION 3.5. *For an MCS ordering, π, let $MCSLB_\pi(G)$ be the best lower bound to the tree-width of G given by π. Precisely, if $\pi = (v_1, \ldots v_n)$ and $T_i = \{v_1, \ldots v_i\}$, then*

$$MCSLB_\pi(G) = \max_{i=2,\ldots n} d_{T_{i-1}}(v_i). \tag{3.14}$$

4. Conclusions. This bound is of both practical and theoretical interest. On the practical side, it will provide a lower bound which may be of use to those interested in calculating or approximating the tree-width of particular graphs. As an example of this, consider the graph G in Figure 3. Suppose we were interested in finding a lower

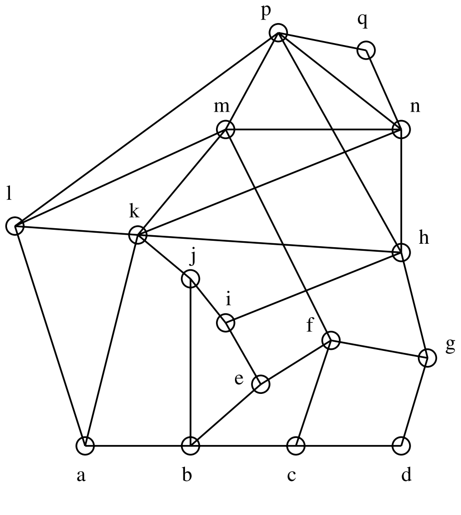


FIG. 3. *An example graph G.*

bound for the tree-width of G. First we look for cliques and find many 3-cliques but no 4-cliques. This yields a lower bound of 2. Next we apply the bound given in [9]. Since nodes q and d are nonadjacent vertices of degree 2, that method also yields a lower bound of 2. So we try to use the MCS lower bound. We find an MCS ordering such as

$$\pi_1 = (a, b, c, d, f, e, g, i, j, h, k, l, m, n, p, q). \tag{4.1}$$

In this case $MCSLB_{\pi_1}(G) = 4$, since node p has four neighbors which come before it in the ordering. It turns out that in this case, the bound is tight and $TW(G) = 4$.

Notice that the MCS bound can be different for different MCS orderings. For example, if we had arbitrarily chosen the MCS ordering

$$\pi_2 = (l, a, k, m, p, n, h, q, f, g, b, j, i, e, c, d), \tag{4.2}$$

we would get $MCSLB_{\pi_2}(G) = 3$, since no unnumbered vertex ever has more than three numbered neighbors, while several (e.g., node h) have exactly three. Since the number of MCS orderings can be large, it is in general not feasible to examine *all* of the MCS orderings to see which gives the best lower bound. However, it is simple enough to find a few different MCS orderings and examine the bounds that arise.

It also may be possible to assign simple heuristics to the arbitrary choices which increase the probability of choosing the MCS orderings which yield the best bounds. One such heuristic would be to always choose a vertex of lowest degree, so as to increase the probability that the higher degree vertices accumulate more numbered neighbors before they are numbered themselves.

One nice property of using MCS as a bounding technique is that if G contains a k-clique, then *any* MCS ordering will give a lower bound of at least $k - 1$. In other words, regardless of the "arbitrary" choices made during the MCS, the bound given will be at least as good as finding the largest clique in the graph. This is easy to see when you consider that the last vertex in a clique of size k to be numbered will have at least $k - 1$ numbered neighbors before it gets numbered. Furthermore, as demonstrated previously, our bound can do better than just finding the largest clique in the graph.

It is not difficult to find examples where the MCS lower bound is actually quite weak, regardless of which MCS ordering(s) are examined. Foremost, the lower bound that could be yielded by this method is bounded above by the vertex of highest degree. So while an $n \times n$ lattice has tree-width n, the MCS lower bound could never be greater than 4 (and in fact it will be 2). So this bound can be arbitrarily weak.

On a theoretical level, this result shows an unexpected link between the MCS algorithm and the tree-width of a graph. It provides a convenient method of identifying or creating graphs of high tree-width. Furthermore, it immediately yields an entire class of forbidden minors for graphs of low tree-width. It also opens new questions for further research. For example, what are the class of graphs for which the (best) MCS lower bound is tight? Which obstructions to low tree-width can this procedure detect? Could the bound be improved by adaptive strategies which selectively contract edges? It would also be interesting to see how this method performs on graphs used in practice (arising from expert systems, for example). Such computational experiments for upper bounds can be found in [6] and [1].

Acknowledgments. I would like to thank Matthew Harrison, who first conjectured the main theorem in this paper; Luis Ortiz for useful discussions and pointing me to references; and Stuart Geman for help in revisions and organization of this paper.

REFERENCES

[1] E. Amir, *Efficient approximation for triangulation of minimum tree-width*, in Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, 2001.

[2] S. Arnborg, *Efficient algorithms for combinatorial graphs with bounded decomposability—A survey*, BIT, 25 (1985), pp. 2–23.

[3] S. Arnborg, D. G. Corneil, and A. Proskurowski, *Complexity of finding embeddings in a k-tree*, SIAM J. Alg. Disc. Meth., 8 (1987), pp. 277–284.

[4] H. Bodlaender and D. Thilikos, *Graphs with Branchwidth at Most Three*, Technical report UU-CS-1997-37, Utrecht University, Department of Computer Science, Utrecht, The Netherlands, 1997.

[5] S. Geman and K. Kochanek, *Dynamic programming and the representation of soft-decodable codes*, IEEE Trans. Inform. Theory, 47 (2001), pp. 549–568.

[6] A. Koster, H. Bodlaender, and S. van Hoesel, *Treewidth: Computational Experiments*, Technical report 01-38, Konrad-Zuse-Zentrum fur Informationstechnik Berlin, Berlin, Germany, 2001.

[7] S. Lauritzen and D. Speigelhalter, *Local computations with probabilities on graphical structures and their application to expert systems*, J. R. Stat. Soc. Ser. B Stat. Methodol., 50 (1988), pp. 157–224.

[8] B. Lucena, *Dynamic Programming, Tree-Width, and Computation on Graphical Models*, Ph.D. thesis, Brown University, Providence, RI, 2002.

[9] S. Ramachandramurthi, *The structure and number of obstructions to treewidth*, SIAM J. Discrete Math., 10 (1997), pp. 146–157.

[10] B. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, UK, 1996.

[11] D. J. Rose, *On simple characterizations of k-trees*, Discrete Math., 7 (1974), pp. 317–322.

[12] R. E. Tarjan and M. Yannakakis, *Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*, SIAM J. Comput., 13 (1984), pp. 566–579.

SIAM J. Discrete Math.
Vol. 16, No. 3, pp. 354–365

NONLINEARITY OF BOOLEAN FUNCTIONS AND HYPERELLIPTIC CURVES[*]

JUNG HEE CHEON[†] AND SEONGTAEK CHEE[‡]

Abstract. We give a novel relationship between the nonlinearity of rational functions over $\mathbb{F}_{2^n}$ and the number of points of the associated hyperelliptic curve. Using this, we obtain a lower bound on the nonlinearity for rational functions over $\mathbb{F}_{2^n}$. Compared to previous work that provides a lower bound on the nonlinearity only for monomials of special types, our result gives a general bound applicable to all rational functions defined over $\mathbb{F}_{2^n}$. By applying this result, we get a lower bound on the nonlinearity for various $n \times kn$ S-boxes.

Key words. S-box, nonlinearity, Boolean functions, hyperelliptic curves

AMS subject classifications. 94A60, 14G05, 11T71, 14H52, 11G20

PII. S0895480199366251

1. Introduction.

One of the most powerful attacks for block ciphers is the linear cryptanalysis developed by Matsui in 1993 [10]. The basic idea of the linear cryptanalysis is to find a linear relation among the plaintexts, ciphertexts, and key bits. Such a relation, in general, can be found by a low nonlinearity of a substitution (called S-box) in block ciphers.

This substitution can be expressed as a Boolean function. The nonlinearity of a Boolean function with a single bit output was well established [15]. However, it is very difficult to analyze the nonlinearity of a Boolean function with a multibit output (called a vector Boolean function) in general. Most known results on the nonlinearity of vector Boolean functions are aimed at the special types of monomials over $\mathbb{F}_{2^n}$ such as x^{-1} or x^r, where the Hamming weight of r is two or three [3, 12, 13]. It has been regarded as an especially hard problem to find a (nontrivial) bound on the nonlinearity for more complicated vector Boolean functions such as polynomials or rational functions. (Of course, for Boolean functions with a small number of input bits, we can get simulation results.)

In this paper, we derive a novel relationship between the nonlinearity of a rational function over $\mathbb{F}_{2^n}$ and the number of points of the associated hyperelliptic curve over that field. Using this relationship we obtain a lower bound on the nonlinearity for a rational function over $\mathbb{F}_{2^n}$. Our result can be applied to much more complicated vector Boolean functions regardless of their size. Note that direct computation of the nonlinearity for an $n \times n$ S-box takes about $n \times 2^{3n}$ operations and is therefore not feasible for $n \geq 16$. Further, we give a lower bound on the nonlinearity for S-boxes constructed by concatenating two or more S-boxes over $\mathbb{F}_{2^n}$. A similar method has been used in the CAST block cipher [1], in which 8×32 S-boxes were constructed by selecting 32 bent Boolean functions over $\mathbb{F}_{2^8}$. S-boxes used in the CAST block cipher

[*] Received by the editors December 6, 1999; accepted for publication (in revised form) December 20, 2002; published electronically April 30, 2003. A preliminary version of this paper was presented in Advances in Cryptology—EUROCRYPT '99, Lecture Notes in Comput. Sci. 1592, Springer-Verlag, Berlin, 1999, pp. 286–294. In this paper, we extend our result to more complicated Boolean functions.

http://www.siam.org/journals/sidma/16-3/36625.html

[†] School of Mathematical Sciences, Seoul National University, Seoul 151-747, Korea (jhcheon@math.snu.ac.kr).

[‡] National Security Research Institute, Daejon 305-350, Korea (chee@etri.re.kr).

have been believed to be highly nonlinear, but the proof of the lower bound on their nonlinearity estimated by a probabilistic method is considered to be hard [14]. However, our method gives a construction of an $n \times kn$ S-box with a provable nonlinearity bound.

In section 2, we recall the notions of nonlinearity and hyperelliptic curves, and some useful results required to prove the main theorem. In section 3, we present the main theorem relating the nonlinearity of a vector Boolean function to the number of rational points of the associated hyperelliptic curve. Also we present several highly nonlinear $n \times n$ S-boxes whose nonlinearity is bounded below by the main theorem. In section 4, we extend this method to highly nonlinear $n \times kn$ S-boxes. Section 5 is the conclusion of this paper.

2. Preliminaries.

We consider a vector Boolean function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$. Let $b = (b_1, b_2, \ldots, b_m)$ be a nonzero element in $\mathbb{F}_2^m$. We denote by $b \cdot F$ the Boolean function which is the linear combination $b_1 f_1 + b_2 f_2 + \cdots + b_m f_m$ of the coordinate Boolean functions $f_1, f_2, \ldots, f_m$ of F over $\mathbb{F}_2^n$.

DEFINITION 2.1. *The nonlinearity of F, $\mathcal{N}(F)$, is defined as*

$$\mathcal{N}(F) = \min_{b \neq 0} \min_{A \in \Gamma} \#\{x | A(x) \neq b \cdot F(x)\},$$

where Γ is the set of all affine functions over $\mathbb{F}_2^n$.

If we define $\mathcal{L}(F, a, b) = \#\{x | a \cdot x = b \cdot F(x)\}$, then we have

$$\mathcal{N}(F) = 2^{n-1} - \max_{b \neq 0} \max_{a} |2^{n-1} - \mathcal{L}(F, a, b)|. \tag{2.1}$$

Observe that nonlinearity of arbitrary vector Boolean functions is bounded above by

$$\mathcal{N}(F) \leq 2^{n-1} - 2^{\frac{n}{2}-1},$$

and the equality holds only for bent functions, which exist if and only if $n \geq 2m$.

Note that a function from $\mathbb{F}_{2^n}$ to $\mathbb{F}_{2^m}$ can be identified as a Boolean function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$ if we specify a basis for each finite field. Since nonlinearity is invariant under basis changes, we can define the nonlinearity of a map between finite fields without specifying their bases. Conversely, any (vector) Boolean function over a vector space can be converted as a map between two finite fields. Throughout this paper, unless specified otherwise, every Boolean function is a map of $\mathbb{F}_{2^n}$ to itself.

The simplest map on a finite field is a monomial. The nonlinearity of monomials is investigated by Nyberg [13].

THEOREM 2.2.

1. *Let $F(x) = x^{2^k+1}$.*
   (a) *If n/s is odd for $s = \gcd(n, k)$, then*

   $$\mathcal{N}(F) = 2^{n-1} - 2^{(n+s)/2-1}. \tag{2.2}$$

   (b) *If n is odd and $\gcd(n, k) = 1$, then*

   $$\mathcal{N}(F) = 2^{n-1} - 2^{(n-1)/2}. \tag{2.3}$$

2. *For $F(x) = x^{-1}$,*

$$\mathcal{N}(F) \geq 2^{n-1} - 2^{n/2}. \tag{2.4}$$

2.1. Hyperelliptic curves. We recall the notion of a hyperelliptic curve and Weil's theorem, which play important roles in proving our main theorem. Consider a curve C given by the equation

$$C : y^2 + h(x)y = f(x), \tag{2.5}$$

where $f(x), h(x) \in \mathbb{F}_{2^n}[x]$ with $\deg h(x) \leq g$ and $\deg f(x) = 2g + 1$ for a positive integer g. A point (x, y) on the curve is said to be *singular* if both partial derivatives of $y^2 + h(x)y - f(x)$ vanish there so that there is no well-defined tangent line. When a curve has no singular point, we say that it is nonsingular. Otherwise, we say that it is singular. A nonsingular curve C of the above form is called a hyperelliptic curve of genus g.

We define the set of $\mathbb{F}_{2^n}$-rational points on C, denoted by $C(\mathbb{F}_{2^n})$, to be the set of all points $(x, y) \in \mathbb{F}_{2^n} \times \mathbb{F}_{2^n}$ that satisfy (2.5) of the curve C, together with a special point at infinity, denoted by O.

For the number $\#C(\mathbb{F}_{2^n})$ of the $\mathbb{F}_{2^n}$-rational points on C, we have the following nontrivial bound [6].

THEOREM 2.3 (Weil). *For any nonsingular projective C of genus g over $\mathbb{F}_{2^n}$, we have*

$$|\#C(\mathbb{F}_{2^n}) - 2^n - 1| \leq 2g\sqrt{2^n}. \tag{2.6}$$

Moreover, a hyperelliptic curve of genus g satisfies (2.6). *(One easily checks using the standard device of taking the projective closure of affine curves that in all cases the projective curve corresponding to* (2.5) *contains only one point at infinity.)*

When a plane curve is singular, the theorem cannot be applied. When a singular curve C is absolutely irreducible (i.e., irreducible over the algebraic closure of the ground field), however, we have the following result by desingularizing the singular algebraic curve [2, 6]:

$$|\#C(\mathbb{F}_{2^n}) - 2^n - 1| \leq 2g\sqrt{2^n} - g + \frac{(d-1)(d-2)}{2}, \tag{2.7}$$

where g and d are the genus and degree of C, respectively. This can be combined with Theorem 2.3 to give the following corollary.

COROLLARY 2.4. *Let C be a curve given by an equation $y^2 + h(x)y = f(x)$, where the degree d of f is an odd integer greater than or equal to* $\max\{2 \deg h + 1, 3\}$. *Assume that C is nonsingular or $d \leq 2^{n/4+1} + 2$. Then we have*

$$|\#C(\mathbb{F}_{2^n}) - 2^n - 1| \leq (d-1)\sqrt{2^n}. \tag{2.8}$$

Proof. If the affine part of C is nonsingular, the genus g of C is $g = (d-1)/2$. Hence the corollary follows. Otherwise, g becomes strictly smaller than $(d-1)/2$ so that $g \leq (d-1)/2 - 1$. In this case, (2.7) gives

$$|\#C(\mathbb{F}_{2^n}) - 2^n - 1| \leq (2\sqrt{2^n} - 1)\left(\frac{d-1}{2} - 1\right) + \frac{(d-1)(d-2)}{2}. \tag{2.9}$$

The right-hand side of (2.9) is less than or equal to $(d-1)\sqrt{2^n}$ if $d^2 - 4d + 5 \leq 4\sqrt{2^n}$. Hence the corollary holds for $3 \leq d \leq 2^{n/4+1} + 2$. □

3. Main theorem.

In this section, we obtain a lower bound on the nonlinearity of rational functions over a finite field, using the bound on the number of points of hyperelliptic curves over that field. Throughout this paper, for any rational function $P(x)/Q(x)$ for $P(x), Q(x) \in \mathbb{F}_{2^n}[x]$ we assume it is defined for all elements in $\mathbb{F}_{2^n}$ by assigning some (arbitrary) value at the zeros of $Q(x)$.

First, we introduce a lemma. We denote by $Tr(\cdot)$ an absolute trace map.

LEMMA 3.1. *The following polynomial equation of one variable x,*

$$x^2 + ax + b = 0, \quad a \neq 0, \quad b \in \mathbb{F}_{2^n}, \tag{3.1}$$

is reducible over $\mathbb{F}_{2^n}$ if and only if $Tr(\frac{b}{a^2}) = 0$.

Proof. If we replace by ax, x of (3.1) and divide the equation by a^2, we obtain $x^2+x+b/a^2 = 0$. Hence $x^2+ax+b = 0$ is reducible over $\mathbb{F}_{2^n}$ if and only if $x^2-x = b/a^2$ has a root in $\mathbb{F}_{2^n}$. By Hilbert theorem 90 [8], it is equivalent to $Tr(b/a^2) = 0$. □

By using the above lemma, we can derive the following theorem.

THEOREM 3.2. *Let $P(x)$, $Q(x)$, and $G(x)$ be polynomials over $\mathbb{F}_{2^n}$, where $G(x)$ is injective. Assume that $C_{a,b}$ is a plane curve defined by $y^2+Q(x)y = aQ(x)^2G(x)+bP(x)$ and $\#C_{a,b}(\mathbb{F}_{2^n})$ is the number of $\mathbb{F}_{2^n}$-rational points on $C_{a,b}$. Then any function $F(x) = P(x)/Q(x)^2$ on $\mathbb{F}_{2^n}$ satisfies*

$$|2\mathcal{L}(F \circ G^{-1}, a, b) - \#C_{a,b}(\mathbb{F}_{2^n}) + 1| \leq r, \tag{3.2}$$

where r is the number of distinct roots of $Q(x)$ in $\mathbb{F}_{2^n}$.

Proof. Choose a basis B of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$ and take its dual basis $\hat{B}$. Represent binary vectors in $\mathbb{F}_{2^n}$, a and b by the basis B, and $G(x)$ and $F(x)$ by its dual basis $\hat{B}$. Then we have

$$a \cdot G(x) = Tr(aG(x)), \quad b \cdot F(x) = Tr(bF(x)).$$

Hence

$$\begin{aligned}\mathcal{L}(F \circ G^{-1}, a, b) &= \#\{x|a \cdot x = b \cdot F(G^{-1}(x))\}\\ &= \#\{x|Tr(aG(x)) = Tr(bF(x))\}\\ &= \#\{x|Tr(aG(x) + bF(x)) = 0\}.\end{aligned}$$

Let $\alpha_1, \alpha_2, \dots, \alpha_r$ be r distinct roots of $Q(x)$. If $\alpha \neq \alpha_i$ for all i, $C_{a,b}$ has two distinct points whose x-coordinate is α whenever $y^2+Q(\alpha)y-(aQ(\alpha)^2G(\alpha)+bP(\alpha))$ is reducible. Also, $C_{a,b}$ has one point whose x-coordinate is α_i since $y^2 - bP(\alpha_i)$ is always reducible. Hence we have

$$\begin{aligned}\#C_{a,b}(\mathbb{F}_{2^n}) &= 2 \cdot \# \left\{x|Tr\left(\frac{aQ(x)^2G(x)+bP(x)}{Q(x)^2}\right) = 0, Q(x) \neq 0\right\} + r + \#\{O\}\\ &= 2 \cdot \#\{x|Tr(aG(x) + bF(x)) = 0, Q(x) \neq 0\} + r + 1 \qquad (3.3)\\ &= 2\mathcal{L}(F \circ G^{-1}, a, b) - 2 \cdot \#\{i|Tr(aG(\alpha_i) + bF(\alpha_i)) = 0\} + r + 1.\end{aligned}$$

The first equality follows from Lemma 3.1. Hence we have

$$\begin{aligned}&|2\mathcal{L}(F \circ G^{-1}, a, b) - \#C_{a,b}(\mathbb{F}_{2^n}) + 1|\\ &\leq |2 \cdot \#\{i|Tr(aG(\alpha_i)) = Tr(bF(\alpha_i))\} - r|\\ &\leq r. \qquad □\end{aligned}$$

4. Nonlinearity of rational functions over $\mathbb{F}_{2^n}$. In this section, we present a lower bound on the nonlinearity of some rational functions using Theorem 3.2. For the convenience of proof, we divide this section into three subsections. Consider a rational function $F(x) = P(x)/Q(x)$ such that $P(x)$ and $Q(x)$ are polynomials over $\mathbb{F}_{2^n}$. In the first subsection, we treat the case of $Q(x) = 1$. In the second subsection, we treat the case of $\deg P > \deg Q$. In the last subsection, we treat the case of $\deg P < \deg Q$.

4.1. Polynomials.

THEOREM 4.1. *Let $d \geq 3$. Consider two polynomials $F(x)$ and $G(x)$ over $\mathbb{F}_{2^n}$, where $\deg F = d$, $\deg G < d$, and $G(x)$ is bijective.*

1. *If d is odd,*

$$\mathcal{N}(F \circ G^{-1}) \geq 2^{n-1} - (d-1)2^{n/2-1}. \tag{4.1}$$

2. *If d is even, let d' be the largest integer among the odd divisors of a degree of a term of $F(x)$. If $d' \geq 3$,*

$$\mathcal{N}(F \circ G^{-1}) \geq 2^{n-1} - (d'-1)2^{n/2-1}. \tag{4.2}$$

Proof.

1. Take $Q(x) = 1$ and $P(x) = F(x)$ in Theorem 3.2. We have

$$2\mathcal{L}(F \circ G^{-1}, a, b) = \#C_{a,b}(\mathbb{F}_{2^n}) - 1 \tag{4.3}$$

for a curve $C_{a,b} : y^2 + y = aG(x) + bF(x)$.

Since each curve $C_{a,b}$ is nonsingular and has the odd degree d at x for each $a, b \neq 0$, by Theorem 2.3

$$|\#C_{a,b}(\mathbb{F}_{2^n}) - 2^n - 1| \leq (d-1)\sqrt{2^n}. \tag{4.4}$$

Combining (4.3) with (4.4), we obtain the first assertion.

2. Assume that d is even and $F(x) = cx^d + F_1(x)$, where $F_1(x)$ is a polynomial over $\mathbb{F}_{2^n}$ of degree less than d. Take $Q(x) = 1$ and $P(x) = F(x)$ as in Theorem 3.2. Then the associated curve $C_{a,b} : y^2 + y = aG(x) + bF(x)$ can be transformed into $C'_{a,b} : y^2 + y = aG(x) + bF_1(x) + b'x^{d/2}$ by $y \to y + b'x^{d/2}$, where b' is a root of $x^2 - bc$ (which always exists because x^2 is a permutation of $\mathbb{F}_{2^n}$). Note that $C_{a,b}$ and $C'_{a,b}$ have the same number of $\mathbb{F}_{2^n}$-rational points. By repeating this process, we can obtain a curve of degree d' at x, which is nonsingular and has the same number of $\mathbb{F}_{2^n}$-rational points with $C_{a,b}$. Hence

$$|\#C_{a,b} - 2^n - 1| \leq (d'-1)\sqrt{2^n}.$$

By applying Theorem 3.2, we complete the proof. □

By applying Theorem 4.1, we can derive easily a lower bound on the nonlinearity of polynomials. Observe that a lower degree polynomial is inclined to have higher nonlinearity.

COROLLARY 4.2. *Let $F(x) \in \mathbb{F}_{2^n}[x]$, $d \geq 3$, and $k \geq 2$.*

1. *For any integer s, d with $2 \nmid d$, $F(x) = x^{2^s d}$ satisfies*

$$\mathcal{N}(x^{2^s d}) \geq 2^{n-1} - (d-1)2^{n/2-1}. \tag{4.5}$$

2. *For any* $a_i \in \mathbb{F}_{2^n}$ *with* $a_{2k-1} \neq 0$*, consider* $F(x) = a_{2k}x^{2k} + a_{2k-1}x^{2k-1} + \cdots + a_0$ *and an injective polynomial* $G(x)$ *of degree less than* $2k-1$*. Then*

$$\mathcal{N}(F \circ G^{-1}) \geq 2^{n-1} - (k-1)2^{n/2}. \tag{4.6}$$

Proof. The assertions follow from Theorem 4.1. □

We present an example using the composition of F and G^{-1} in order to obtain higher degree monomials with high nonlinearity.

Example 1. Consider $F(x) = x^3$ and $G(x) = x^5$ over $\mathbb{F}_{2^7}$. Since $G^{-1}(x) = x^{51}$, we have $F \circ G^{-1}(x) = x^{153} = x^{26}$. By Corollary 4.2, we have

$$\mathcal{N}(x^{26}) \geq 2^{n-1} - 2^{n/2+1}.$$

Since nonlinearity is preserved under composition with linear functions like x^2, x^{13} has the same nonlinearity as $(x^{13})^2$. Hence we have

$$\mathcal{N}(x^{13}) \geq 2^{n-1} - 2^{n/2+1}.$$

4.2. The case of $P(x)/Q(x)^2$ with $\deg P > 2\deg Q + 1$. Theorem 4.1 gives us a lower bound on the nonlinearity of polynomial functions. But if the function contains a term of negative degree, the theorem cannot be applied. For this case, we need the following theorem.

THEOREM 4.3. *Let* $P(x)$*,* $Q(x)$*, and* $G(x) \in \mathbb{F}_{2^n}[x]$*, where* $2\deg Q + \deg G < \deg P \leq 2^{n/4+1} + 2$ *and* $G(x)$ *is injective. Consider a rational function* $F(x) = P(x)/Q(x)^2$ *over* $\mathbb{F}_{2^n}$*. Then if* $d = \deg P \geq 3$ *and* d *is odd,*

$$\mathcal{N}(F \circ G^{-1}) \geq 2^{n-1} - (d-1)2^{n/2-1} - \frac{r}{2},$$

where r *is the number of the distinct roots of* $Q(x)$ *in* $\mathbb{F}_{2^n}$*.*

Proof. By Corollary 2.4, the associated curve $C_{a,b} : y^2 + Q(x)y = aQ(x)^2G(x) + bP(x)$ for each $a, b \neq 0$ satisfies

$$|\#C_{a,b}(\mathbb{F}_{2^n}) - 2^n - 1| \leq (d-1)\sqrt{2^n}. \tag{4.7}$$

Combining (4.7) with (3.2), we obtain the theorem. □

By applying Theorem 4.3, we can derive a lower bound on the nonlinearity of some rational functions.

COROLLARY 4.4. *Let* $F(x) \in \mathbb{F}_{2^n}[x]$*.*

1. *For any* $a_i \in \mathbb{F}_{2^n}$ *with* $a_3 \neq 0$ *and* $1 \leq k \leq 2^{n/4} - 1/2$*,* $F(x) = a_3x^3 + a_2x^2 + \cdots + a_{-2k}x^{-2k}$ *satisfies*

$$\mathcal{N}(F) \geq 2^{n-1} - (k+1)2^{n/2} - \frac{1}{2}. \tag{4.8}$$

2. *For any* $a_i \in \mathbb{F}_{2^n}$ *with* $a_5 \neq 0$ *and* $1 \leq k \leq 2^{n/4} - 3/2$*, consider* $F(x) = a_5x^5 + a_3x^3 + \cdots + a_{-2k}x^{-2k}$ *and an injective polynomial* $G(x)$ *of degree less than* 4*. Then*

$$\mathcal{N}(F \circ G^{-1}) \geq 2^{n-1} - (k+2)2^{n/2} - \frac{1}{2}. \tag{4.9}$$

Proof. Take $G(x) = x$, $Q(x) = x^k$, and $P(x) = x^{2k}F(x) = a_3x^{2k+3} + \cdots + a_{-2k}$ in Theorem 4.3. Then $2\deg Q + \deg G = 2k+1 < 2k+3 = \deg P$ and $3 \leq \deg P \leq 2^{n/4+1}$. Since $d = 2k+3$ and $r = 1$, we obtain the first assertion.

Take $Q(x) = x^k$ and $P(x) = x^{2k}F(x) = a_5x^{2k+5} + \cdots + a_{-2k}$ in Theorem 4.3. Then $2\deg Q + \deg G < 2k+5 = \deg P$ and $3 \leq \deg P \leq 2^{n/4+1}$. Since $d = 2k+5$ and $r = 1$, we obtain the first assertion. □

4.3. The case of $P(x)/Q(x)^2$ with $\deg P < \deg Q$.

THEOREM 4.5. *Let $Q(x), G(x)$ and $P(x)$ be polynomials over $\mathbb{F}_{2^n}$, where $\deg P - 1 \leq \deg Q \leq 3$, $d = 2\deg Q + \deg G \leq 2^{n/4+1} + 2$, and $G(x)$ is injective. Consider a rational function $F(x) = P(x)/Q(x)^2$ over $\mathbb{F}_{2^n}$. If $y^2 + Q(x)y = bP(x)$ is irreducible for every $b \neq 0$, then*

$$\mathcal{N}(F \circ G^{-1}) \geq 2^{n-1} - (d-1)2^{n/2-1} - \frac{r}{2}, \tag{4.10}$$

where r is the number of the distinct roots of $Q(x)$ in $\mathbb{F}_{2^n}$.

Proof. By Corollary 2.4, for each $a \neq 0, b \neq 0$ the associated curve $C_{a,b} : y^2 + Q(x)y = aQ(x)^2G(x) + bP(x)$ satisfies

$$|\#C_{a,b}(\mathbb{F}_{2^n}) - 2^n - 1| \leq (d-1)\sqrt{2^n}. \tag{4.11}$$

Consider the case of $a = 0$. If $\deg P - 1 \leq \deg Q \leq 3$, then the associated curve $C_{0,b} : y^2 + Q(x)y = bP(x)$ is of degree $\deg Q + 1$. If $C_{0,b}$ is nonsingular, $C_{0,b}$ satisfies (4.11) since it has genus $\deg Q(\deg Q - 1)/2 \leq (d-1)/2$. If $C_{0,b}$ is singular, $C_{0,b}$ has genus g less than $\deg Q(\deg Q - 1)/2$. Since $C_{0,b}$ is irreducible by assumption, we have by (2.7)

$$|\#C_{0,b}(\mathbb{F}_{2^n}) - 2^n - 1| \leq 2g\sqrt{2^n} - g + \frac{\deg Q(\deg Q - 1)}{2},$$

where the right-hand side is less than $(d-1)\sqrt{2^n}$.

In all cases, (4.11) holds for each $a, b \neq 0$. By Theorem 3.2, we complete the proof. □

COROLLARY 4.6. *Let $F(x) \in \mathbb{F}_{2^n}[x]$ with $n \geq 6$.*

1. *Let $\alpha \in \mathbb{F}_{2^n}$. For any odd integer k and $a_i \in \mathbb{F}_{2^n}$ with $a_k \neq 0$, $F(x) = a_1(x+\alpha)^{-1} + a_2(x+\alpha)^{-2} + \cdots + a_k(x+\alpha)^{-k}$ satisfies*

$$\mathcal{N}(F) \geq 2^{n-1} - (k+1)2^{n/2-1} - \frac{1}{2}. \tag{4.12}$$

2. *For any polynomial $H(x) \in \mathbb{F}_{2^n}$ of degree $k = 2$ or 3, $F(x) = x/H(x)^2$ satisfies*

$$\mathcal{N}(F) \geq 2^{n-1} - k \cdot 2^{n/2} - \frac{r}{2}, \tag{4.13}$$

where r is the number of the distinct roots of $H(x)$.

Proof. If we take $G(x) = x + \alpha$, $Q(x) = x^{(k+1)/2}$, and $P(x) = a_1x^k + a_2x^{k-1} + \cdots + a_kx$, then $C_{0,b} : y^2 + x^{(k+1)/2}y = b(a_1x^k + a_2x^{k-1} + \cdots + a_kx)$ is irreducible for every nonzero b and nonzero a_k. Since $d = k + 2 \leq 6 \leq 2^{n/4+1} + 2$, the first assertion holds.

If we take $G(x) = x$, $Q(x) = H(x)$, and $P(x) = x$, then $C_{0,b} : y^2 + H(x)y = bx$ is irreducible for every nonzero b. Since $d = 2k + 1 \leq 6 \leq 2^{n/4+1} + 2$, the second assertion holds. □

THEOREM 4.7. *Let A, B be distinct nonzero elements in $\mathbb{F}_{2^n}$, $\alpha \in \mathbb{F}_{2^n}^*$, and $n \geq 2$. If $F(x)$ is a function on $\mathbb{F}_{2^n}$ satisfying*

$$F(x) = \frac{A}{x} + \frac{B}{(x+\alpha)} \text{ for each } x \neq 0,$$

TABLE 1
Lower bound on the nonlinearity for $n \times n$ vector Boolean functions.

Function	Lower bound	Constraint
$x^{2k-1} + \cdots + a_0$	$2^{n-1} - (k-1)2^{n/2}$	$k \geq 2$
$x^3 + \cdots + a_{-2k}x^{-2k}$	$2^{n-1} - (k+1)2^{n/2} - 1/2$	$1 \leq k \leq 2^{n/4} - 1/2$
$x^5 + \cdots + a_{-2k}x^{-2k}$	$2^{n-1} - (k+2)2^{n/2} - 1/2$	$1 \leq k \leq 2^{n/4} - 3/2$
$a_1x^{-1} + \cdots + a_kx^{-k}$	$2^{n-1} - (k+1)2^{n/2-1} - 1/2$	$k \geq 1$, $a_k \neq 0$
$x/H(x)^2$	$2^{n-1} - k \cdot 2^{n/2} - r/2$	$k = \deg H = 2$ or 3 $r = \#$ of roots of $H(x)$
$\frac{A}{x} + \frac{B}{x+\alpha}$	$2^{n-1} - 2^{n/2+1} - 1$	$AB\alpha \neq 0$

TABLE 2
Comparison of the lower bound and the exact value of the nonlinearity.

Function	n	Our lower bound	Exact value
$x^3 + x^5 + x^6$	7	48	48
	8	96	96
$x^{-1} + x^3$	7	41	46
	8	96	100
$x^{-3} + x^{-1}$	7	41	46
	8	96	97

then we have

$$\mathcal{N}(F) \geq 2^{n-1} - 2^{n/2+1} - 1. \tag{4.14}$$

Proof. Take $Q(x) = x(x+\alpha)$, $G(x) = x$, and $P(x) = ((A+B)x + A\alpha)^2$ as in Theorem 3.2. By Corollary 2.4, the associated curve $C_{a,b} : y^2 + Q(x)y = aQ(x)^2G(x) + bP(x)$ for each $a \neq 0, b \neq 0$ satisfies

$$|\#C_{a,b}(\mathbb{F}_{2^n}) - 2^n - 1| \leq 2\sqrt{2^n}. \tag{4.15}$$

Consider that case of $a = 0$. The associated curve $C_{0,b} : y^2 + Q(x)y = bP(x)$ is of degree 3. If $C_{0,b}$ is nonsingular, $C_{0,b}$ satisfies (4.15) since it has genus 1. If $C_{0,b}$ is singular, $C_{0,b}$ has genus 0. Since $C_{0,b}$ is irreducible for each nonzero b, we have by (2.7)

$$|\#C_{a,b}(\mathbb{F}_{2^n}) - 2^n - 1| \leq 1.$$

In all cases, (4.15) holds for each $a, b \neq 0$. By Theorem 3.2, $P(x)/Q(x)^2$ satisfies

$$\mathcal{N}(P(x)/Q(x)^2) \geq 2^{n-1} - 2^{n/2+1} - 1.$$

Since $F(x)^2 = P(x)/Q(x)^2$, $F(x)$ has the same nonlinearity with $P(x)/Q(x)^2$ which completes the proof. □

4.4. Experimental results. In Table 1, we present the main results of this chapter in short. Every function in the table is a vector Boolean function from $\mathbb{F}_{2^n}$ to $\mathbb{F}_{2^n}$. Also, A, B, α, and a_i denote an element of $\mathbb{F}_{2^n}$.

In Table 2, we compare our bound on the nonlinearity and the exact value for several $n \times n$ vector Boolean functions. Observe that our bound is very tight in case the degree of function has small absolute values.

5. Nonlinearity of x^{-1} and x^3. In this section, we apply the previous results to obtain the exact nonlinearity of x^{-1} and x^3, which are frequently used in designing block ciphers. This result is not new. The nonlinearity of x^{-1} is bounded below using a Kloosterman sum [7, p. 228] and its exact value was determined using elliptic curves [5]. The nonlinearity of x^3 can be derived from the weight distribution of the BCH code $Tr(ax+bx^3)$ [9, pp. 451–452].

THEOREM 5.1. *Let $F(x)$ be the function on $\mathbb{F}_{2^n}$ such that*

$$F(x) = \begin{cases} \frac{1}{x}, & x \neq 0, \\ 0, & x = 0. \end{cases}$$

Then every component of $F(x)$ has the same nonlinearity.

Proof. If we take $G(x) = Q(x) = P(x) = x$ in (3.2) of the proof of Theorem 3.2, we have $\#C_{a,b}(\mathbb{F}_{2^n}) = 2\mathcal{L}(F,a,b)$, where $C_{a,b} : y^2 + xy = ax^3 + bx$. Note that $F(x)$ is injective so that $\mathcal{L}(F,0,b) = 2^{n-1}$. Hence it is enough to show that for given nonzero $b, b' \in \mathbb{F}_{2^n}$ there is a nonzero $a' \in \mathbb{F}_{2^n}$ such that $C_{a,b}$ is isomorphic to $C_{a',b'}$ for any nonzero $a \in \mathbb{F}_{2^n}$. By the transformation $(x,y) \mapsto (ax, ay)$, we know that $C_{a,b}$ is isomorphic to $y^2 + xy = x^3 + abx$. Hence if we take $a' = ab/b'$, $C_{a,b}$ is isomorphic to $C_{a',b'}$, which completes the proof. □

Any polynomial $F(x)$ of degree 3 on $\mathbb{F}_{2^n}$ has lower bound on the nonlinearity not less than $2^{n-1} - 2^{n/2}$. More precisely, we obtain the following.

THEOREM 5.2. *Let $F(x) = x^3$ be on $\mathbb{F}_{2^n}$. Then the exact nonlinearity of $F(x)$ is as follows:*

$$\mathcal{N}(F) = \begin{cases} 2^{n-1} - 2^{\frac{n-1}{2}} & \text{if } n \text{ is odd,} \\ 2^{n-1} - 2^{n/2} & \text{if } n \text{ is even.} \end{cases} \tag{5.1}$$

Moreover, if n is odd, every component function of $F(x)$ has the same nonlinearity, and if n is even, the nonlinearity of any component function of $F(x)$ is either $2^{n-1} - 2^{n/2-1}$ or $2^{n-1} - 2^{n/2}$.

Proof. If we take $G(x) = Q(x) = 1$ and $P(x) = x^3$ in (3.2) of the proof of Theorem 3.2, we have $\#C_{a,b}(\mathbb{F}_{2^n}) = 2\mathcal{L}(F,a,b) + 1$, where $C_{a,b} : y^2 + y = bx^3 + ax$. Since $C_{a,b}$ is a supersingular elliptic curve for each $a, b \neq 0$, if we let $t = \#C_{a,b}(\mathbb{F}_{2^n}) - 2^n - 1$, t^2 is $0, 2^n, 2 \cdot 2^n, 3 \cdot 2^n$ or $4 \cdot 2^n$. Hence we have

$$|2\mathcal{L}(F,a,b) - 2^n| = |\#C(F,a,b) - 2^n - 1| = \begin{cases} 0, \quad 2^{(n+1)/2} & \text{for odd } n, \\ 0, \quad 2^{n/2}, \quad 2^{1+n/2} & \text{for even } n. \end{cases}$$

Therefore,

$$|2^{n-1} - \mathcal{L}(F,a,b)| = \begin{cases} 0, \quad 2^{(n-1)/2} & \text{for odd } n, \\ 0, \quad 2^{n/2-1}, \quad 2^{n/2} & \text{for even } n \end{cases}$$

for any $a, b \neq 0$.

On the other hand, $C_{a,b}$ is isomorphic to $y^2 + by = x^3 + abx$ by the transformation $(x,y) \mapsto (bx, by)$ for any nonzero b. If n is odd, it is isomorphic to $y^2 + y = x^3 + ax/\gamma$ by the transformation $(x,y) \mapsto (x/\gamma^2, y/\gamma^3)$, where $\gamma^3 = b$. Hence for each nonzero b, there exists $a \in \mathbb{F}_{2^n}$ such that $C_{a,b}$ can have order $2^n + 1 + \sqrt{2^n}$ or $2^n + 1 - \sqrt{2^n}$ [11]. In any case, $b \cdot F$ has the nonlinearity $2^{n-1} - 2^{(n-1)/2}$. Similarly, when n is even, $b \cdot F$ has the nonlinearity $2^{n-1} - 2^{n/2}$ if b is a cube of an element in $\mathbb{F}_{2^n}$, or $2^{n-1} - 2^{n/2-1}$ otherwise [11]. □

6. Nonlinearity of $n \times kn$ S-boxes. In this section, we derive the nonlinearity of $n \times kn$ S-box constructed by concatenating k $n \times n$ S-boxes over $\mathbb{F}_{2^n}$. At first, we present a theorem to relate nonlinearity of an $n \times kn$ S-box to that of an $n \times n$ S-box.

THEOREM 6.1. *Let $F : \mathbb{F}_{2^n} \to \mathbb{F}_{2^{kn}}$ be a vector Boolean function with $F = (F_1, F_2, \dots, F_k)$ for $F_i : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$. Then we have*

$$\mathcal{N}(F) = \min_{(c_1,c_2,\dots,c_k)\in\mathbb{F}^*_{2^{kn}}} \mathcal{N}(c_1F_1 + c_2F_2 + \cdots + c_kF_k),$$

where the sum and the product are the field operations in $\mathbb{F}_{2^{kn}}$.

Proof. Choose a basis B of $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$ and take its dual basis $\hat{B}$. Let us represent by the basis B the left sides of all inner products and by its dual basis $\hat{B}$ their right sides. For any nonzero $b = (c_1, c_2, \dots, c_k)$ with $c_i \in \mathbb{F}_{2^n}$, we have

$$\begin{aligned}\mathcal{L}(F, a, b) &= \#\{x | a \cdot x = b \cdot F(x)\}\\ &= \#\{x | Tr(ax + bF(x)) = 0\}\\ &= \#\{x | Tr(ax + c_1F_1(x) + \cdots + c_kF_k(x)) = 0\}\\ &= \mathcal{L}(c_1F_1 + \cdots + c_kF_k, a, 1),\end{aligned}$$

where 1 is the binary vector representing the identity element by the basis B.

Conversely, for any nonzero $(c_1, c_2, \dots, c_k) \in \mathbb{F}_{2^{kn}}, c_i \in \mathbb{F}_{2^n}$, and a nonzero $b_0 \in \mathbb{F}_{2^n}$, there exists a nonzero $b \in \mathbb{F}_{2^{kn}}$ such that $\mathcal{L}(c_1F_1 + \cdots + c_kF_k, a, b_0) = \mathcal{L}(F, a, b)$, which completes the proof. □

By the above theorem, we can apply Theorem 3.2 to get a lower bound on the nonlinearity of an $n \times kn$ S-box. For example, consider an $n \times 2n$ S-box $F = (F_1, F_2)$, where $F_1(x) = x^{-1}$ and $F_2(x) = x^3$ are S-boxes over $\mathbb{F}_{2^n}$. Then

$$\begin{aligned}\mathcal{N}(F) &= \min_{(c_1,c_2)\neq 0} \mathcal{N}(c_1x^{-1} + c_2x^3)\\ &= \min\{\min_{c_i\neq 0} \mathcal{N}(c_1x^{-1} + c_2x^3), \mathcal{N}(x^{-1}), \mathcal{N}(x^3)\}\\ &\geq 2^{n-1} - 2^{n/2+1} + \frac{1}{2}.\end{aligned}$$

The first equality follows from Theorem 6.1 and the last inequality follows from Corollary 4.4.

Similarly, we can obtain a lower bound on the nonlinearity for various $n \times kn$ S-boxes. We present some of them in Table 3. Observe that every rational function such as x^{-1} and x^3 in Table 3 is a vector Boolean function from $\mathbb{F}_{2^n}$ to $\mathbb{F}_{2^n}$. The second column shows a lower bound on the nonlinearity of the S-boxes in the first column, whose value for $n = 8$ appears in the third column. The fourth column shows the exact value on the nonlinearity calculated by a computational experiment.

In Table 3, we can see that our bound is tight for the case of 8×16 S-boxes. However, we could not compute the exact nonlinearity of $8 \times 8k$ S-boxes for $k \geq 3$ since the computation cost is too large. We think that our bound may be tight in case the degree of function has small absolute values.

If we combine our result with Theorem 17 in [14], we can also construct $kn \times kn$ S-boxes. But we could not obtain a good lower bound on the nonlinearity of such $kn \times kn$ S-boxes. We hope to find a method to construct highly nonlinear $kn \times kn$ S-boxes from $n \times n$ S-boxes.

TABLE 3
Lower bound on the nonlinearity for $n \times kn$ S-boxes.

S-box	Lower bound of nonlinearity	for $n = 8$	Exact value
(x^{-1}, x^3)	$2^{n-1} - 2^{n/2+1} - \frac{1}{2}$	96	100
(x^{-1}, x^{-3})	$2^{n-1} - 2^{n/2+1} - \frac{1}{2}$	96	100
(x^3, x^5)	$2^{n-1} - 2^{n/2+1}$	96	96
$(x^{-1}, (x-1)^{-1})$	$2^{n-1} - 2^{n/2+1} - 1$	96	96
(x^{-3}, x^{-5})	$2^{n-1} - 3 \cdot 2^{n/2} - \frac{1}{2}$	80	96
(x^{-3}, x^{-1}, x^3)	$2^{n-1} - 3 \cdot 2^{n/2} - \frac{1}{2}$	80	-
(x^{-1}, x^3, x^5)	$2^{n-1} - 3 \cdot 2^{n/2} - \frac{1}{2}$	80	-
(x^3, x^5, x^7)	$2^{n-1} - 3 \cdot 2^{n/2}$	80	-
$(x^{-3}, x^{-1}, x^3, x^5)$	$2^{n-1} - 4 \cdot 2^{n/2} - \frac{1}{2}$	64	-
(x^{-1}, x^3, x^5, x^7)	$2^{n-1} - 4 \cdot 2^{n/2} - \frac{1}{2}$	64	-
(x^3, x^5, x^7, x^9)	$2^{n-1} - 4 \cdot 2^{n/2}$	64	-

7. Conclusion. In this paper, we derived a novel relationship between the nonlinearity of a rational function and the number of points of the associated hyperelliptic curve. As a result, we can obtain a lower bound on the nonlinearity for various rational functions. Our result can be used to generate highly nonlinear S-boxes with much more complicated algebraic structures. Also we presented a method to construct highly nonlinear $n \times kn$ S-boxes whose nonlinearity bound can be easily computed. This method is useful for designing an asymmetric Feistel network such as Bear and Lion.

Further, we can consider constructing highly nonlinear $kn \times kn$ S-boxes from $n \times n$ S-boxes. Such a construction enables us to design a block cipher with large S-boxes implemented by several small S-boxes. It can make designing block cipher much simpler. However, we have not found such a construction yet. Thus the problem remains open.

Acknowledgements. We would like to thank Sang Geun Hahn, Sangjoon Park, and Yongdae Kim for motivating this work and their helpful comments. We also thank the anonymous referees for providing suitable references on Weil bounds for singular curves and pointing out numerous errors in the early draft.

REFERENCES

[1] C. ADAMS AND S. E. TAVARES, *Designing S-boxes for Ciphers Resistant to Differential Cryptanalysis*, in Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography, Rome, Italy, 1993.

[2] Y. AUBRY AND M. PERRET, *A Weil theorem for singular curves*, in Arithmetic, Geometry and Coding Theory, de Gruyter, Berlin, 1996, pp. 1–7.

[3] T. BETH AND D. DING, *On almost perfect nonlinear permutations*, in Advances in Cryptology—EUROCRYPT '93, Lecture Notes in Comput. Sci. 765, Springer-Verlag, Berlin, 1994, pp. 65–76.

[4] J. CHEON AND S. CHEE, *S-boxes with controllable nonlinearity*, in Advances in Cryptology—EUROCRYPT '99, Lecture Notes in Comput. Sci. 1592, Springer-Verlag, Berlin, 1999, pp. 286–294.

[5] G. Lachaud and J. Wolfman, *The weights of the orthogonals of the extended quadratic binary Goppa codes*, IEEE Trans. Inform. Theory, 36 (1990), pp. 686–692.

[6] D. Leep and C. Yeomans, *The number of points on a singular curve over a finite field*, Arch. Math., 63 (1994), pp. 420–426.

[7] R. Lidl and H. Niederreiter, *Finite Fields*, Addison-Wesley, Reading, MA, 1983.

[8] R. Lidl and H. Niederreiter, *Introduction to Finite Fields and their Applications*, Cambridge University Press, Cambridge, UK, 1986.

[9] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, Amsterdam, 1977.

[10] M. Matsui, *Linear cryptanalysis method for DES cipher*, in Advances in Cryptography—EUROCRYPT '93, Lecture Notes in Comput. Sci. 765, Springer-Verlag, Berlin, 1994, pp. 386–397.

[11] A. Menezes, *Elliptic Curve Public Key Cryptography*, Kluwer Academic Publishers, Boston, 1993.

[12] K. Nyberg, *On the construction of highly nonlinear permutation*, in Advances in Cryptology—EUROCRYPT '92, Lecture Notes in Comput. Sci. 658, Springer-Verlag, Berlin, 1993, pp. 92–98.

[13] K. Nyberg, *Differentially uniform mappings for cryptography*, in Advances in Cryptography—EUROCRYPT '93, Lecture Notes in Comput. Sci. 765, Springer-Verlag, Berlin, 1994, pp. 55–64.

[14] K. Nyberg, *S-boxes and round functions with controllable linearity and differential uniformity*, in Proceedings of the Second Fast Software Encryption, Lecture Notes in Comput. Sci. 1008, Springer-Verlag, Berlin, 1995, pp. 111–130.

[15] J. Seberry, X. -M. Zhang, and Y. Zheng, *Nonlinearly balanced functions and their propagation characteristics*, in Advances in Cryptography—CRYPTO '93, Lecture Notes in Comput. Sci. 773, Springer-Verlag, Berlin, 1994, pp. 49–60.

[16] J. Silverman, *The Arithmetic of Elliptic Curves*, Springer-Verlag, New York, 1992.

SIAM J. DISCRETE MATH.
Vol. 16, No. 3, pp. 366–370

A HIGH GIRTH GRAPH CONSTRUCTION*

L. SUNIL CHANDRAN†

Abstract. We give a deterministic algorithm that constructs a graph of girth $\log_k(n) + O(1)$ and minimum degree $k-1$, taking number of nodes n and number of edges $e = \lfloor nk/2 \rfloor$ (where $k < \frac{n}{3}$) as input. The degree of each node is guaranteed to be $k-1, k$, or $k+1$, where k is the average degree. Although constructions that achieve higher values of girth—up to $\frac{4}{3}\log_{k-1}(n)$—with the same number of edges are known, the proof of our construction uses only very simple counting arguments in comparison. Our method is very simple and perhaps the most intuitive: We start with an initially empty graph and keep introducing edges one by one, connecting vertices which are at large distances in the current graph. In comparison with the Erdös–Sachs proof, ours is slightly simpler while the value it achieves is slightly lower. Also, our algorithm works for all values of n and $k < \frac{n}{3}$, unlike most of the earlier constructions.

Key words. girth, algorithm

AMS subject classifications. 68R10, 05C35

PII. S0895480101387893

1. Introduction.

1.1. The main result.

We give an algorithm that takes a positive integer n and an "expected degree" k (where $k < \frac{n}{3}$) as input and creates a graph $G = (V, E)$ with n nodes, $\lfloor nk/2 \rfloor$ edges, and girth g satisfying the relation

$$\frac{n}{2} \leq 1 + \frac{(k+1)(k^{g-1}-1)}{(k-1)}.$$

It follows that $g > \log_k(n) + O(1)$. We prove that the degree of any node in the graph constructed by our algorithm will be $k-1, k$, or $k+1$. Thus given the values of girth (say g) and minimum degree (say t) our algorithm can be used to construct graphs with at most $2 + \frac{2(t+2)((t+1)^{g-1}-1)}{t}$ nodes. Note that this bound is comparable to the results of Erdös and Sachs [4] and Sauer [17], who showed that the minimum number of vertices $n(g,t)$, required for the girth to be greater than or equal to g and the minimum degree to be greater than or equal to t, satisfies

$$n(g,t) \leq 2\frac{(t-1)^{g-1}-1}{t-2} \quad \text{if } g \text{ is odd},$$

$$n(g,t) \leq 4\frac{(t-1)^{g-2}-1}{t-2} \quad \text{if } g \text{ is even}.$$

(See also [3, p. 107].)

Although we achieve a slightly lower value, our proof looks slightly simpler.

[*] Received by the editors April 11, 2001; accepted for publication (in revised form) November 25, 2002; published electronically April 30, 2003. A preliminary version of this paper appeared as "A high girth graph construction and a lower bound for hitting set size for combinatorial rectangles" in *Proceedings of the* 19*th Conference on Foundations of Software Technology and Theoretical Computer Science* (Chennai, India, December, 1999), Lecture Notes in Comput. Sci. 1738, C. Pandu Rangan, V. Raman, and R. Ramanujam, eds., Springer, Heidelberg, 1999, pp. 283–290. Preliminary version ©1999 Springer. Portions reprinted with permission.

http://www.siam.org/journals/sidma/16-3/38789.html

[†] Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India 560012. Current address: Max–Planck Institute für informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany (sunil@mpi-sb.mpg.de).

1.2. Other known high girth graph constructions. The problem of constructing high girth graphs with high minimum degree was difficult. For many years the only significant results in this direction were the theorems of Erdös and Sachs and their improvements by Sauer [17], Walther [18, 19], and others (see page 107 in [3] for a brief history) and later by Mai, Wang, and Luo [14], who proved the existence of infinite families of k-regular graphs with girth $\log_{k-1} n$, where n is the number of nodes in the graph. The first explicit construction of a family of high girth graphs was given by Margulis [15] in which girth approximately $0.44 \log_{k-1}(n)$ was proved for some infinite families with arbitrary large k, and girth $0.83 \log_{k-1}(n)$ was proved for an infinite family with $k = 4$. Imrich [5] was able to improve the result to girth $0.48 \log_{k-1}(n)$, in the case of arbitrary large k, and to produce a family of cubic graphs ($k = 3$) with girth $= \log_{k-1}(n)$. In [1], a family of geometrically defined cubic graphs, the so-called sextet graphs, was introduced by Biggs and Hoare. They conjectured that these graphs have large girth. Weiss [20] proved the conjecture by showing that for the sextet graphs (or their double cover) girth $= (4/3) \log_{k-1}(n)$. Then, independently, Margulis (see [16] and references therein) and Lubotzky, Phillips, and Sarnak [6] came up with similar examples of graphs with girth $\geq (4/3) \log_{k-1}(n)$, which was proved by Biggs and Boshier to be exact [2]. In [7], Lazebnik and Ustimenko constructed a family of graphs with girth approximately equal to $\log_{k-1}(n)$ for arbitrary large k. In [13] it is proved that the graphs in [7] are disconnected and each component has girth approximately $(4/3) \log_{k-1}(n)$. Some other girth-related references are [8, 9, 10, 11, 12].

In this paper we give a construction which works for any n and $k < \frac{n}{3}$, achieving girth $\log_k(n)$, where k is the average degree. The graphs constructed by our algorithm are only approximately regular—every node is guaranteed to have a degree $k-1, k$, or $k+1$. The advantage of our construction is that it is very elementary and only uses combinatorial arguments, avoiding sophisticated mathematics. (The method of Erdös and Sachs also is combinatorial while others are algebraic.) We start with an initially empty graph of n vertices and keep adding edges one by one, connecting vertices which are at large distances in the current graph. We feel that this is the most intuitive way of achieving high girth. It works for any n and $k < \frac{n}{3}$, unlike most earlier constructions.

2. High girth graph construction.

The following algorithm takes the number of nodes n and the average degree k (where $k < \frac{n}{3}$) as input and constructs a graph of girth at least $\log_k(n) + O(1)$. All the nodes in the graph will have degree $k-1, k$, or $k+1$.

2.1. The algorithm. Let n be an even integer. (This is just for convenience. We will describe the case when n is odd shortly.) Assume that, in the beginning, we have a perfect matching on the n nodes. That is, we start with a graph having $\frac{n}{2}$ edges, the degree of each node being 1. Do the following steps for $i = \frac{n}{2} + 1$ to $\frac{kn}{2}$:

1. Let $S = \{u \in V : \mathit{degree}(u) \leq \mathit{degree}(v)\, \forall v \in V\}$.
2. Let $T = \{(u, v) \in S \times V : \mathit{distance}(u, v) \geq \mathit{distance}(x, y)\, \forall\, (x, y) \in S \times V\}$.
3. If there is a pair $(u, v) \in T$, such that $\mathit{degree}(v) \leq j$, where $j = \lceil \frac{2i}{n} \rceil$, and the edge $\{u, v\}$ is not already in the graph, introduce a new edge $\{u, v\}$ and go to the beginning of the loop. If there are several such pairs, pick one arbitrarily. Else go to 4.
4. Let $\rho = \mathit{distance}(u, v)$, where $(u, v) \in T$. Put $\rho = \rho - 1$. Now assign $T = \{(u, v) \in S \times V : \mathit{distance}(u, v) = \rho\}$. Go to 3.

(The above algorithm does the following. In step 1, it collects in S all the vertices having the least degree in the graph. Next, it collects in T all the pairs of nodes (u, v) such that $distance(u, v)$ is maximum from the set of all pairs (u, v) with u in S. Put an edge (if it is not already there) between one such pair from T, making sure that both u and v have degree less than or equal to $j = \lceil \frac{2i}{n} \rceil$. If no (u, v) satisfies this degree requirement, let T be redefined as the set of pairs (u, v), such that u is in S and $distance(u, v) = \rho - 1$, where $\rho =$ distance between any pair (u, v) which was in T earlier.)

LEMMA 2.1. *The graph created by the above algorithm will be such that for all $v \in V$, degree(v) will be $k-1, k,$ or $k+1$. If V_d denotes the set of vertices with degree d in the graph, $|V_{k-1}| = |V_{k+1}| \le \frac{n}{2}$.*

Proof. Let us use induction as each batch of $\frac{n}{2}$ edges is introduced. (That is, as the average degree of the graph increases by 1.) (Note that the parameter $j = \lceil \frac{2i}{n} \rceil$ remains constant as a batch of $\frac{n}{2}$ edges is introduced and, when the next batch starts, it increases by 1.)

Consider the following induction hypothesis.

When $i = \frac{dn}{2}$ iterations of the loop are over, the degree of any node in the graph will be $d-1, d,$ or $d+1$. Let X be the set of vertices with degree $d-1$, Y be that of vertices with degree d, and Z be that of vertices with degree $d+1$. Then $|X| = |Z|$.

In the beginning of the algorithm, average degree $= 1$ and the statement is true because there are only vertices of degree 1 in the graph. We have $|X| = |Z| = 0$.

Now assuming the induction hypothesis after d batches of $\frac{n}{2}$ edges are introduced, let us prove that it is true even after the next batch is introduced. We note that since $|X| = |Z|$, $|X| \le \frac{n}{2}$. Thus when $\frac{n}{2}$ edges are introduced, each vertex in the set X gets a chance to increase its degree (because minimum degree vertices have preference). So after $\frac{n}{2}$ edges are introduced there will be no vertices of degree $d-1$ left. Further, no vertex will have degree $> (d+2)$ because while these edges are introduced the parameter j will remain equal to $d+1$. Since the sum of degrees during this stage can reach a maximum of $(d+1)n$ only, and $d \le k < \frac{n}{3}$, it can be easily verified that for each minimum degree node (which will be at least $d-1$ at this stage) there will always be a node which is nonadjacent to it and of degree less than $d+2$. Therefore the algorithm will add an edge on each iteration. Thus after the new $\frac{n}{2}$ edges are introduced we retain the induction hypothesis statement that the degrees are $d, d+1,$ *or* $d+2$. Now let X_1, Y_1, Z_1 be the new sets of $d, d+1$, and $d+2$ degree vertices, respectively. It suffices to show that $|X_1| = |Z_1|$. We know that

$$|X_1|d + |Y_1|(d+1) + |Z_1|(d+2) = (d+1)n. \tag{2.1}$$

We also have $|X_1| + |Y_1| + |Z_1| = n$. Therefore, $|Z_1| = |X_1|$. Thus we get back the induction hypotheses, and hence the result follows. □

2.2. Construction for odd n. If n is odd, one may start with an n-length cycle C^n instead of a matching. Then at the beginning of the loop $i = n+1$. Again, when we start the algorithm, every node has a degree of 2, which is equal to the average degree. Now think of introducing batches of $\lfloor \frac{n}{2} \rfloor$ and $\lceil \frac{n}{2} \rceil$ edges alternately. That is, each odd-numbered batch will contain $\lfloor \frac{n}{2} \rfloor$ edges and every even-numbered batch will contain $\lceil \frac{n}{2} \rceil$ edges. We can consider a slightly different induction hypothesis. If k is even, then just after we have introduced $\frac{nk}{2}$ edges, every node in the graph will be of degree $k-1, k,$ or $k+1$. Moreover, $|V_{k+1}| = |V_{k-1}|$, implying that $|V_{k+1}| = |V_{k-1}| \le \frac{n}{2}$. If k is odd, then just after introducing $\lfloor \frac{kn}{2} \rfloor$ edges, every node in the graph will be of degree $k-1, k$, or $k+1$ as before. However, $|V_{k+1}| = |V_{k-1}| - 1$. We still have $|V_{k+1}| \leq \frac{n}{2}$. The induction hypothesis also implies that, just after each odd-numbered batch of edges is introduced, $|V_{k-1}| \leq \lceil \frac{n}{2} \rceil$, and just after every even-numbered batch is introduced, $|V_{k-1}| \leq \lfloor \frac{n}{2} \rfloor$. Thus when a new batch of edges is introduced, each node in the set V_{k-1} will increase its degree as before. Also, no node can attain a degree greater than $(k+2)$. Thus when $k+1$ is even, we get back the same equation as (2.1). When $k+1$ is odd, since we have only introduced $\lfloor \frac{n(k+1)}{2} \rfloor$ edges, the equation becomes

$$|V_k|k + |V_{k+1}|(k+1) + |V_{k+2}|(k+2) = (k+1)n - 1. \tag{2.2}$$

Solving this we get $|V_{k+2}| = |V_k| - 1$. Thus again $|V_{k+2}| \leq \frac{n}{2}$.

THEOREM 2.2. *The above algorithm creates a graph which has a girth*

$$g > \log_k(n) + O(1).$$

Proof. Look at the final graph. Let the girth be g. This cycle (girdle) closed sometime back in the process. Go back to the stage just before closing the smallest cycle. Let $d = \lceil \frac{2i}{n} \rceil$, where i is the loop iteration number at that time. For that iteration, we had selected a current minimum degree vertex, u. (That is, we had selected a pair (u, v), with $u \in S$.) Let $B = \{x \in V : \mathit{distance}(u, x) \geq g\}$. Why did we not select a vertex from B to be connected with u? Because those vertices, if any, had already achieved a degree of $(d+1)$. The algorithm prohibits us from connecting u with them. But we know by Lemma 2.1 that the number of vertices of degree $d+1$ can be at most $\frac{n}{2}$. Thus $V - B$ contains at least $\frac{n}{2}$ nodes. Using the fact that $k+1 \geq d+1$ ($d+1$ being the maximum degree of the graph at that stage), we see that the maximum number of nodes possible in $V - B$ is $1 + (k+1) + (k+1)k + \cdots + (k+1)k^{(g-2)}$. Combining the lower and upper bounds for $|V - B|$, we have

$$\frac{n}{2} \leq 1 + (k+1) + (k+1)k + \cdots + (k+1)k^{(g-2)},$$

$$\frac{n}{2} \leq 1 + \frac{(k+1)(k^{g-1} - 1)}{(k-1)},$$

which gives

$$\log_k(n) + O(1) < g. \qquad \square$$

REFERENCES

[1] N.L. Biggs and M.J. Hoare, *The sextet construction for cubic graphs*, Combinatorica, 3 (1983), pp. 153–165.

[2] N.L. Biggs and A.G. Boshier, *Note on the girth of Ramanujan graphs*, J. Combin. Theory Ser. B, 49 (1990), pp. 190–194.

[3] B. Bollobás, *Extremal Graph Theory*, Academic Press, London, 1978.

[4] P. Erdös and H. Sachs, *Reguläre Graphe gegebener Taillenweite mit minimaler Knotenzahl*, Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe, 12 (1963), pp. 251–257.

[5] W. Imrich, *Explicit construction of graphs without small cycles*, Combinatorica, 2 (1984), pp. 53–59.

[6] A. Lubotzky, R. Phillips, and P. Sarnak, *Ramanujan graphs*, Combinatorica, 8 (1988), pp. 261–271.

[7] F. Lazebnik and V.A. Ustimenko, *Explicit construction of graphs with an arbitrary large girth and of large size*, Discrete Appl. Math., 60 (1995), pp. 275–284.

[8] F. Lazebnik, V.A. Ustimenko, and A.J. Woldar, *A characterization of the components of the graphs $D(k,q)$*, Discrete Math., 157 (1996), pp. 271–283.

[9] F. Lazebnik, V.A. Ustimenko, and A.J. Woldar, *New upper bounds on the order of cages*, Electron. J. Combin., 4 (1997), Research paper 13 (electronic).

[10] F. Lazebnik, V.A. Ustimenko, and A.J. Woldar, *Polarities and $2k$-cycle-free graphs*, Discrete Math., 197/198 (1999), pp. 503–513.

[11] F. Lazebnik, V.A. Ustimenko, and A.J. Woldar, *Properties of certain families of $2k$-cycle free graphs*, J. Combin. Theory Ser. B, 60 (1994), pp. 293–298.

[12] F. Lazebnik, V.A. Ustimenko, and A.J. Woldar, *New constructions of bipartite graphs on m, n vertices, with many edges, and without small cycles*, J. Combin. Theory Ser. B, 61 (1994), pp. 111–117.

[13] F. Lazebnik, V.A. Ustimenko, and A.J. Woldar, *A new series of dense graphs of high girth*, Bull. Amer. Math. Soc. (N.S.), 32 (1995), pp. 73–79.

[14] J. Mai, S. Wang, and H. Luo, *Number of edges of graph with girth $> n+1$*, J. China Univ. Sci. Tech., 14 (1984), pp. 467–474 (in Chinese).

[15] G.A. Margulis, *Explicit constructions of graphs without short cycles and low density codes*, Combinatorica, 2 (1982), pp. 71–78.

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

[17] N. Sauer, *Extremaleigenschaften regulärer Graphen gegebener Taillenweite.* I, II, Österreich Akad. Wiss. Math.-Natur. Kl. S.-B. II, 176 (1967), pp. 9–25; ibid. 176 (1967), pp. 27–43.

[18] H. Walther, *Eigenschaften von regulären Graphen gegebener Taillenweite und minimaler Knotenpunktanzahl*, Wiss. Z. Techn. Hochsch. Ilmenau, 11 (1965), pp. 167–168.

[19] H. Walther, *Über reguläre Graphen gegebener Taillenweite und minimaler Knotenpunktzahl*, Wiss. Z. Techn. Hochsch. Ilmanau, 11 (1965), pp. 93–96.

[20] A.I. Weiss, *Girth of bipartite sextet graphs*, Combinatorica, 4 (1984), pp. 241–245.

SIAM J. DISCRETE MATH.
Vol. 16, No. 3, pp. 371–389

COMPLEXES OF t-COLORABLE GRAPHS*

SVANTE LINUSSON[†] AND JOHN SHARESHIAN[‡]

Abstract. We study the simplicial complex of t-colorable graphs on n vertices. We prove this complex is homotopy equivalent to a wedge of spheres all of dimension $n(t-1)-\binom{t}{2}-1$ when $t=2$ and when $t \geq n-3$. We show that such a homotopy equivalence does not hold for general t and n.

Key words. monotone graph property, chromatic number, discrete Morse theory

AMS subject classifications. 5E25, 6A11

PII. S0895480100366968

1. Introduction and statement of results.

Recall that a monotone graph property is a collection Γ of graphs on a fixed labeled vertex set V that satisfies the following two conditions:

(A) If $G \in \Gamma$ and H is obtained from G by removing an edge, then $H \in \Gamma$.

(B) If $G \in \Gamma$ and H is isomorphic to G, then $H \in \Gamma$.

From now on we assume that $V = [n]$ for some $n \in \mathbb{N}$. Any collection Γ of graphs on V that satisfies condition (A) determines a simplicial complex (also called Γ) whose vertex set corresponds to the set of edges in the complete graph K_n. That is, a graph $G \in \Gamma$ with $k+1$ edges corresponds to a k-dimensional face whose proper faces correspond to proper subgraphs of G. If Γ also satisfies condition (B), then the action of the symmetric group S_n on V determines an action of S_n on Γ as a group of simplicial automorphisms.

The theory of group actions on topological spaces was successfully applied to the study of monotone graph properties by Kahn, Saks, and Sturtevant in their examination of computational complexity (see [KSS]). More recently, certain classes of monotone graph properties have appeared in various areas of mathematics. Complexes of graphs of bounded degree arise in group theory, topology, combinatorics, geometry, and commutative algebra (see [Bo, BLVZ, KRW, RR]), and complexes of graphs of bounded connectivity arise in the study of invariants of knots and similar objects (see [BBLSW, Sh1, Sh2, Tu, Va1, Va2, Va3]). The appealing results discovered in the papers mentioned led us to investigate the topology of other monotone graph properties which have simple graph-theoretic definitions. In this paper we discuss our results on complexes of graphs of bounded chromatic number.

Recall that a graph G is t-colorable if its vertices can be colored with t colors so that no two adjacent vertices receive the same color. The chromatic number $\chi(G)$ is the smallest t such that G is t-colorable. For $1 \leq t \leq n$, the complex of t-colorable

*Received by the editors January 31, 2000; accepted for publication (in revised form) October 16, 2002; published electronically April 30, 2003.
http://www.siam.org/journals/sidma/16-3/36696.html

[†] Department of Mathematics, Linköping University, S-581 83 Linköping, Sweden (linusson@mai.liu.se). This author was supported by a Swedish Natural Sciences Research Council (NFR) postdoctoral fellowship.

[‡] Department of Mathematics, University of Miami, Coral Gables, FL 33124-4250 (shareshi@math.miami.edu). Current address: Department of Mathematics, Washington University, St. Louis, MO 63130 (shareshi@math.wustl.edu). This author was supported by National Science Foundation grant DMS-0070757.

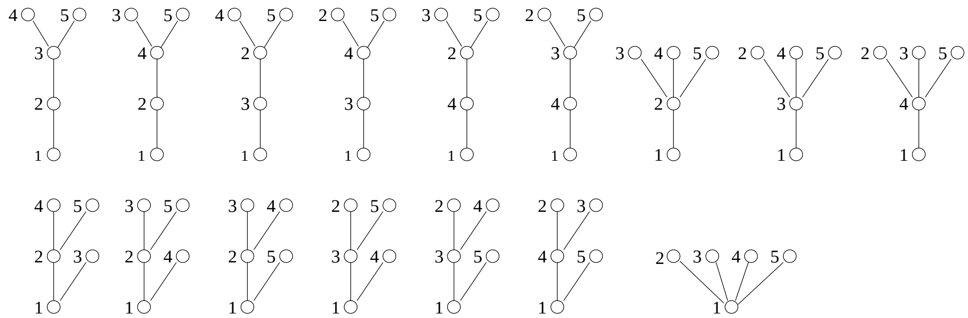


FIG. 1.1. *The* 16 *trees in* $\mathcal{T}(5)$.

graphs on n vertices is

$$\Gamma_n^t := \{G \subseteq K_n : \chi(G) \leq t\}.$$

If $t = 1$, then Γ_n^t consists of only the empty graph, while if $t = n$, then Γ_n^t is the full $(\binom{n}{2} - 1)$-simplex. The only graph on n vertices that is not $(n-1)$-colorable is the complete graph K_n, so Γ_n^{n-1} is the boundary of the $(\binom{n}{2} - 1)$-simplex and is therefore homeomorphic with the $(\binom{n}{2} - 2)$ sphere. We are able to determine the homotopy type of Γ_n^t when t is one of $2, n-2$, or $n-3$.

To state our result in the case $t = 2$ we need some definitions. Recall that for vertices x, y of G the distance $d(x,y)$ from x to y is defined by $d(x,y) = \infty$ if x and y are not in the same connected component of G, and $d(x,y)$ is the length, i.e., the number of edges, of the shortest path from x to y in G otherwise. We now define, for any graph $G = ([n], E)$ and $i \in \mathbb{N} \cup \{\infty\}$,

$$D_i(G) := \{x \in [n] \setminus \{1\} : d(1,x) = i\}$$

and

$$md(G) = \max\{i \in \mathbb{N} : D_i(G) \neq \emptyset\}.$$

For $1 \leq i \leq md(G)$, define

$$a_i(G) := \min D_i(G).$$

Now define $\mathcal{T}(n)$ to be the set of all trees T on vertex set $[n]$ that satisfy the following three conditions:

- For $1 \leq i < md(T)$, the only element of $D_i(T)$ that has neighbors in $D_{i+1}(T)$ is $a_i(T)$.
- For $1 \leq i < md(T)$, there is some element of $\bigcup_{j \geq i} D_j(T)$ that is larger than $a_i(T)$.
- $|D_{md(T)}(T)| > 1$.

See Figure 1.1 for a list of the trees in $\mathcal{T}(5)$.

In section 4, we will prove the following result.

THEOREM 1.1. *For all* $n \geq 2$ *the complex* Γ_n^2 *has the homotopy type of a wedge of* $|\mathcal{T}(n)|$ *spheres of dimension* $n-2$.

The fact that Γ_n^2 has the homotopy type of a wedge of spheres of dimension $n-2$ is a special case of a result of Chari (see [Ch1, Ch2]), who showed that if every edge of some graph G is contained in an odd cycle, then the complex $\Gamma^2(G)$ of all 2-colorable subgraphs of G has the homotopy type of a wedge of spheres of dimension $|V(G)|-2$. It is also the case that if some edge of G is not contained in an odd cycle, then $\Gamma^2(G)$ is collapsible and therefore contractible. Further discussion of the relationship between our result and Chari's appears in section 4.

Using a result that appears as an exercise in [St], we can obtain a generating function for $|\mathcal{T}(n)|$.

Corollary 1.2. *The generating function*

$$\mathcal{B}(z) := \sum_{n\geq 2} |\mathcal{T}(n)| \frac{z^n}{n!}$$

satisfies

$$\mathcal{B}(z) = -\sqrt{2e^{-z}-1} - z + 1.$$

Proof. By Theorem 1.1, $|\mathcal{T}(n)|$ is equal to the absolute value of the reduced Euler characteristic $\tilde{\chi}(\Gamma_n^2)$. In exercise 5.5 of [St], it is shown that if $b(n,m)$ is the number of bipartite graphs on vertex set $[n]$ with m edges, then

$$\sum_{n,m\geq 0} b(n,m) q^m \frac{x^n}{n!} = \left[\sum_{n\geq 0}\left(\sum_{i=0}^{n}(1+q)^{i(n-i)}\binom{n}{i}\right)\frac{x^n}{n!}\right]^{1/2}.$$

Setting $q=-1$ and $x=-z$ completes the proof. □

We now turn to the cases $t=n-2$ and $t=n-3$. In section 5 we will prove the following results.

Theorem 1.3. *For each $n\geq 3$ the complex Γ_n^{n-2} has the homotopy type of a wedge of $\binom{n-1}{2}$ spheres of dimension $\binom{n}{2}-4$*

Theorem 1.4. *For each $n\geq 4$ the complex Γ_n^{n-3} has the homotopy type of a wedge of $\binom{n-1}{3}+12\binom{n}{5}$ spheres of dimension $\binom{n}{2}-7$.*

With Theorems 1.1, 1.3, and 1.4 in hand, it is reasonable to conjecture that Γ_n^t has the homotopy type of a wedge of spheres of dimension $n(t-1)-\binom{t}{2}-1$ for all n,t. However, computer calculations show that this conjecture is false. In particular, $\widetilde{H}_{19}(\Gamma_8^4)$ is free of rank one. On the other hand, computer calculations also show that $\widetilde{H}_i(\Gamma_8^4)=0$ for $i\notin\{17,19\}$ and $\widetilde{H}_{17}(\Gamma_8^4)$ is free of rank $9,396$. So, one can still ask whether Γ_n^t always has nontrivial free homology in dimension $n(t-1)-\binom{t}{2}-1$ and whether this is the smallest dimension in which nontrivial homology appears. In the course of proving Theorems 1.3 and 1.4, we prove a lemma from which the next result follows immediately.

Proposition 1.5. *For all n,t the complex Γ_n^t is $(\lceil\frac{(n-1)(t-1)}{2}\rceil-1)$-connected.*

Although Theorems 1.1, 1.3, and 1.4 can all be obtained using traditional tools from topological combinatorics such as the Quillen fiber lemma, Alexander duality, and the theory of lexicographic shellability (see [Bj]), the proofs we will give all use the discrete Morse theory of Forman, which is described in [Fo1]. These proofs, although somewhat technical and complicated, are considerably simpler and more natural than the proofs which use the traditional tools. One of our main purposes for writing this paper is to further demonstrate the power of Forman's theory in the study of monotone graph properties (other evidence appears in [BBLSW, Fo2, Jo, Sh1, Sh2]) and to explain certain techniques which have proven useful when applying the theory.

After giving explicit definitions of some of the objects mentioned in this introduction and introducing some other necessary concepts in section 2, we discuss discrete Morse theory in section 3. The proof of Theorem 1.1 and some further discussion of that theorem appear in section 4, and the proofs of Theorems 1.3 and 1.4 appear in section 5. Concluding remarks appear in section 6.

2. Preliminaries. By a graph $G = (V(G), E(G))$ we mean a loopless graph without multiple edges (equivalently, a one-dimensional simplicial complex) on the vertex set $V(G)$ with edge set $E(G) \subseteq \binom{V(G)}{2}$. Our standard vertex set will be the set $[n] := \{1, 2, \ldots, n\}$. A graph G is called *t-colorable* if there is a function $f : V(G) \to [t]$ such that if $\{x, y\} \in E(G)$, then $f(x) \neq f(y)$. Thus the preimages $\left\{f^{-1}(i) : i \in [t]\right\}$ partition $V(G)$ into (at most) t independent sets. Graphs that are 2-colorable are also called bipartite.

Of course, if $G = (V(G), E(G))$ is a graph that is t-colorable for some $t \geq 2$, then for any subset $E' \subseteq E(G)$ the graph $G' = (V(G), E')$ on the same vertex set is also t-colorable. Hence if we fix an n-element vertex set V (e.g., $[n]$) and identify a graph with the set of its edges, then we may regard the set of t-colorable graphs on V as a simplicial complex.

DEFINITION 2.1. *Γ_n^t is the complex of t-colorable graphs on n vertices* $(2 \leq t \leq n)$*. Its simplices are the subsets $E \subseteq \binom{[n]}{2}$ such that the graph $([n], E)$ is t-colorable.*

For a graph G and a vertex v, $G - v$ will denote the graph that is obtained from G by deleting the vertex v from its set of vertices and deleting all edges emerging from v from the set of edges. Also, $N_G(v)$ will denote the neighborhood of v in G, that is, the set of all $u \in V(G)$ such that $\{u, v\} \in E(G)$. If v and w are two distinct vertices of G, then vw will denote the two-element set $\{v, w\}$, $G - vw$ will denote the graph $(V(G), E(G) \setminus \{vw\})$, and $G + vw$ will denote the graph $(V(G), E(G) \cup \{vw\})$. Note that (by definition) if $xy \in E(G)$, then $G + xy = G$ and if $xy \notin E(G)$, then $G - xy = G$. For any graph G, $\overline{G}$ will denote the complement graph $(V(G), \binom{V(G)}{2} \setminus E(G))$. Also, for $U \subseteq V(G)$, G_U will denote the subgraph of G induced on U, so $G_U = (U, E_U)$, where E_U is the set of all edges $vw \in E(G)$ such that $\{v, w\} \subseteq U$.

The *face poset* of a simplicial complex is the poset of simplices ordered by inclusion. In this paper we will include the empty face as $\hat{0}$, but not an artificial top element $\hat{1}$. All homology groups discussed in this paper have integer coefficients.

3. Forman's discrete Morse theory. In this section we give a combinatorial description of Forman's discrete Morse theory for simplicial complexes (see [Fo1]). This description is originally due to Chari (see [Ch1]).

For a poset P, we define $D(\mathsf{P}) = (\mathsf{P}, A(\mathsf{P}))$ to be the directed graph obtained by directing each edge in the Hasse diagram of P downwards, so there is an arc (x, y) in $A(\mathsf{P})$ if and only if x covers y in P. For $M \subseteq A(\mathsf{P})$, we define $D_M(\mathsf{P})$ to be the directed graph obtained from $D(\mathsf{P})$ by reversing the direction of all the arcs in M, so $D_M(\mathsf{P}) = (\mathsf{P}, A_M(\mathsf{P}))$, where

$$A_M(\mathsf{P}) = (A(\mathsf{P}) \setminus M) \cup \{(x, y) : (y, x) \in M\}.$$

If P is the face poset (including the empty face) of a simplicial complex Σ, we will write $D(\Sigma)$ for $D(\mathsf{P})$ and $D_M(\Sigma)$ for $D_M(\mathsf{P})$. We call $M \subseteq A(\mathsf{P})$ an *acyclic matching* if M satisfies the following conditions:

(M1) M is a matching, that is, each element of P is an endpoint of at most one arc in M.

(M2) $D_M(\mathsf{P})$ has no directed cycles.

If $M \subseteq A(\mathsf{P})$ is an acyclic matching, the *critical points* of M are those $x \in \mathsf{P}$ such that there is no arc in M having x as an endpoint. M is an *acyclic perfect matching* if M is acyclic and has no critical points. A *Morse matching* on a simplicial complex Σ is an acyclic matching in $D(\Sigma)$.

The next theorem is a special case of the main result of [Fo1].

THEOREM 3.1 (see [Fo1, Corollary 3.5]). *Let M be a Morse matching on a nonempty simplicial complex Σ. Assume that $\emptyset$ is not a critical point of M. Then Σ is homotopy equivalent to a CW-complex which has one 0-cell and, for $k \geq 0$, one additional k-cell for each k-dimensional face of Σ which is a critical point of M.*

Remark. It follows immediately that if M is a Morse matching on Σ and there is some k such that each critical point of M is k-dimensional, then Σ is homotopy equivalent to a wedge of k-dimensional spheres.

The next three results are useful in verifying that a given M is a Morse matching and will be used repeatedly in what follows. The first result, which allows one to produce a Morse matching on a complex Σ by piecing together acyclic matchings on subposets of the face poset of Σ, first appeared in [Jo]. The second result first appeared in [Sh1]. Both have straightforward proofs. The third result gives a general method for defining Morse matchings. It gives a uniform explanation of the ad hoc constructions of Morse matchings found in [BBLSW, Jo] and [Sh1, Sh2].

LEMMA 3.2 (Cluster Lemma [Jo, Lemma 2.2]). *Let $\mathsf{P}_1, \dots, \mathsf{P}_r$ be pairwise disjoint, order convex subposets of P. For each $i \in [r]$, let M_i be an acyclic matching on $D(\mathsf{P}_i)$. Define a relation on the P_i by $\mathsf{P}_i \leq_c \mathsf{P}_j$ if there exist $x \in \mathsf{P}_i$ and $y \in \mathsf{P}_j$ such that $x \leq y$. Assume that the P_i satisfy the following condition:*

($\mathcal{P}$) The relation $\leq_c$ defines a partial order on the P_i's.
Then

$$M := \bigcup_{i=1}^{r} M_i$$

is an acyclic matching on $D(\mathsf{P})$.

LEMMA 3.3 (Cycle Lemma [Sh1, Proposition 3.1]). *Let P be an order convex subposet of the face poset of a simplicial complex Σ and assume that $M \subseteq A(\mathsf{P})$ satisfies condition* (M1). *Then every directed cycle in $D_M(\mathsf{P})$ is of the form $\sigma_1, \tau_1, \sigma_2, \tau_2, \dots, \sigma_{r-1}, \tau_{r-1}, \sigma_r = \sigma_1$, where*

1. *$r \geq 3$;*
2. *for each $i \in [r-1]$, there is some $x_i \in \tau_i$ such that $\tau_i = \sigma_i \cup \{x_i\}$ and $(\tau_i, \sigma_i) \in M$;*
3. *for each $i \in [r-1]$, there is some $y_i \in \tau_i$ such that $\sigma_{i+1} = \tau_i \setminus \{y_i\}$;*
4. *the multisets $\{x_i : i \in [r]\}$ and $\{y_i : i \in [r]\}$ are equal.*

LEMMA 3.4 (Pairing Lemma). *Let Σ be a simplicial complex on a partially ordered vertex set $(V, \preceq)$. Let P be an order convex subposet of the face poset of Σ. For a function $f : \mathsf{P} \to V$, set*

$$\mathsf{P}_f := \{\sigma \in \mathsf{P} : f(\sigma) \notin \sigma\}.$$

For $\sigma \in \mathsf{P}_f$, set

$$\sigma^+ := \sigma \cup \{f(\sigma)\},$$

and for $\tau \in \mathsf{P} \setminus \mathsf{P}_f$ set

$$\tau^- = \tau \setminus \{f(\tau)\}.$$

Assume that f satisfies the following conditions:

(A) *If $\sigma \in \mathsf{P}_f$, then $\sigma^+ \in \mathsf{P}$.*

(B) *If $\sigma \in \mathsf{P}_f$, then $f(\sigma^+) = f(\sigma)$.*

(C) *If $\tau \in \mathsf{P} \setminus \mathsf{P}_f$ and $\tau^- \in \mathsf{P}$, then $f(\tau^-) = f(\tau)$.*

(D) *If $x \in \sigma \in \mathsf{P}_f$ and $\sigma^+ \setminus \{x\} \in \mathsf{P}$, then $f(\sigma) \preceq f(\sigma^+ \setminus \{x\})$.*

Then

$$M_f := \left\{(\sigma^+, \sigma) : \sigma \in \mathsf{P}_f\right\}$$

is an acyclic matching on $D(\mathsf{P})$, and the critical points of M_f are those $\rho \in \mathsf{P} \setminus \mathsf{P}_f$ such that $\rho \setminus \{f(\rho)\} \notin \mathsf{P}$.

Proof. Since f satisfies condition (A), the endpoints of each arc in M_f lie in P. For any arc (ρ, τ) in M_f we have $\rho \in \mathsf{P} \setminus \mathsf{P}_f$ and $\tau \in \mathsf{P}_f$. It follows that for $\sigma \in \mathsf{P}_f$ there is exactly one arc in M_f with σ as an endpoint, namely, (σ^+, σ). Say $\tau \in \mathsf{P} \setminus \mathsf{P}_f$. If $\tau^- \in \mathsf{P}$ then, since f satisfies condition (C), we have $\tau^- \in \mathsf{P}_f$ and $(\tau, \tau^-) \in M_f$. Since $\tau \in \mathsf{P} \setminus \mathsf{P}_f$, we see that there is no arc $(\rho, \tau) \in M$, and any arc $(\tau, \sigma) \in M_f$ satisfies $\sigma \in \mathsf{P}_f$ and $\tau = \sigma \cup \{f(\sigma)\}$. Since f satisfies condition (B), we have $\sigma = \tau^-$. Therefore, if $\tau^- \in \mathsf{P}$, then (τ, τ^-) is the unique arc in M_f having τ as an endpoint, while if $\tau^- \notin \mathsf{P}$, there is no arc in M having τ as an endpoint. Thus M_f satisfies condition (M1), and the elements of P that are not covered by an arc in M_f are the elements that are claimed to be critical points of M_f. It remains to show that M_f satisfies condition (M2). Assume for contradiction that

$$\sigma_1, \tau_1, \ldots, \sigma_{r-1}, \tau_{r-1}, \sigma_r = \sigma_1$$

is a cycle in $D_{M_f}(\mathsf{P})$, which satisfies the conditions of Cycle Lemma 3.3. Then for each $i \in [r-1]$ we have $(\tau_i = \sigma_i^+, \sigma_i) \in M_f$ and $\sigma_{i+1} = \tau_i \setminus y_i$, where $y_i \neq x_i = f(\sigma_i)$. Since f satisfies condition (D), we have

$$f(\sigma_1) = x_1 \preceq f(\sigma_2) = x_2 \preceq \cdots \preceq f(\sigma_r) = x_r = x_1,$$

so $x_i = x_1$ for all $i \in [r-1]$. Now, since condition 4 of Cycle Lemma 3.3 is satisfied, we have $y_i = x_1$ for all i, contradicting our assumption that $y_i \neq x_i$. □

Convention. When applying Lemma 3.4 to a complex Σ of graphs on vertex set $[n]$ so that the vertices of Σ are the elements of $\binom{[n]}{2}$, we will assume, unless otherwise stated, that the order $\preceq$ on $\binom{[n]}{2}$ is the lexicographic order, so $\{a < b\} \preceq \{c < d\}$ if either $a < c$ or $a = c$ and $b \leq d$.

4. The complex of bipartite graphs.

In this section we will prove Theorem 1.1 by producing a Morse matching on $D(\Gamma_n^2)$ whose critical points are the trees in the set $\mathcal{T}(n)$ defined in section 1. As we will discuss in more detail below, a more general result has been obtained by Chari (see [Ch2]), who produced a Morse matching whose critical points are trees for the complex of bipartite subgraphs of any graph G such that each edge of G is contained in an odd cycle.

For the reader's convenience, we repeat the definition of $\mathcal{T}(n)$, recalling that $d(x, y)$ is the length of the shortest path between two vertices x, y of G (with $d(x, y) = \infty$ if no such path exists). For any graph $G = ([n], E)$ and $i \in \mathbb{N} \cup \{\infty\}$, we define

$$D_i(G) := \{x \in [n] \setminus \{1\} : d(1, x) = i\}$$

and

$$md(G) = \max\{i \in \mathbb{N} : D_i(G) \neq \emptyset\}.$$

For $1 \leq i \leq md(G)$, define

$$a_i(G) := \min D_i(G).$$

Now $\mathcal{T}(n)$ is defined to be the set of all trees on vertex set $[n]$ satisfying the following:

1. For $1 \leq i < md(T)$, the only element of $D_i(T)$ that has neighbors in $D_{i+1}(T)$ is $a_i(T)$.
2. For $1 \leq i < md(T)$, there is some element of $\bigcup_{j \geq i} D_j(T)$ that is larger than $a_i(T)$.
3. $|D_{md(T)}(T)| > 1$.

As stated above, we will prove the following result, from which Theorem 1.1 follows immediately upon application of Theorem 3.1 and the remark following it.

THEOREM 4.1. *For $n \geq 2$, there is a Morse matching on $D(\Gamma_n^2)$ whose critical points are all trees $T \in \mathcal{T}(n)$.*

Before proving Theorem 4.1 we will discuss the general result of Chari mentioned above. Fix a graph $G = ([n], E(G))$ and a linear order $\preceq$ on $E(G)$. Let T be a spanning tree of G. For an edge $e \in E(T)$, let $A(e), B(e)$ be the connected components of $T - e$. Then e is called internally active (with respect to T) if e is least (with respect to $\preceq$) among all edges of G which have one endpoint in $A(e)$ and one endpoint in $B(e)$. The internal activity $IA(T)$ is the set of all $e \in e(T)$ which are internally active with respect to T. Let $e \in E(G) \setminus E(T)$ be an edge such that the unique cycle $C(e)$ in $T + e$ has odd length. Then e is called oddly externally active (with respect to T) if e is the least element of $C(e)$. The odd external activity $OEA(T)$ is the set of all $e \in E(G) \setminus E(T)$ which are oddly externally active with respect to T. Using a result of Colbourn and Pulleyblank (see [CP]), Chari proves the following theorem.

THEOREM 4.2 (see [Ch2]). *For any graph G such that each $e \in E(G)$ is contained in an odd cycle in G, let $\Gamma^2(G)$ be the complex of bipartite subgraphs $H = (V(G), E(H))$ of G. Then $D(\Gamma^2(G))$ admits a Morse matching whose critical points are those spanning trees T of G such that $IA(T) = OEA(T) = \emptyset$.*

If some $e \in E(G)$ is not contained in any odd cycle, then

$$\{(H, H - e) : e \in E(H)\}$$

is a Morse matching on $\Gamma^2(G)$ with no critical points, so $\Gamma^2(G)$ is contractible (actually, collapsible). Note that the set of critical points of the Morse matching given in Theorem 4.2 depends on the chosen linear order $\preceq$ on $E(G)$. It is straightforward to show that when $n = 4$ there exists no linear ordering of the edges of K_4 such that the trees in $\mathcal{T}(4)$ are the trees satisfying $IA(T) = OEA(T) = \emptyset$. Indeed, if for each $i \in [4]$ we let T_i be the unique tree on vertex set $[4]$ such that $|N_G(i)| = 3$, then $\mathcal{T}(4) = \{T_1, T_2, T_3\}$. If $IA(T_1) = OEA(T_1) = \emptyset$, then the given linear order on $E(K_4)$ must be either

$$34 \prec 13 \prec 23 \prec 12 \prec 24 \prec 14$$

or

$$12 \prec 23 \prec 13 \prec 34 \prec 14 \prec 24.$$

If the first order is chosen, then 13 is externally active with respect to T_2, while if the second order is chosen, then 12 is internally active with respect to T_2. Therefore we cannot have $IA(T_i) = OEA(T_i) = \emptyset$ for $i = 1$ and $i = 2$ simultaneously. Enumerating the trees with no internal activity and no odd external activity in an arbitrary graph G seems quite difficult, and it would be interesting if one could find another Morse matching on $D(\Gamma^2(G))$ (for arbitrary G) whose critical points are easy to enumerate.

We now prove Theorem 4.1 by induction on n. If $n = 2$, then Γ_n^2 consists of the empty graph and K_2, so there is a Morse matching on $D(\Gamma_2^2)$ with no critical points. Since no tree on two vertices satisfies condition 3 in the definition of $\mathcal{T}(n)$, we see that $\mathcal{T}(2) = \emptyset$ and the theorem holds. Now assume $n > 2$. Define the following:

- $I_n := \left\{G \in \Gamma_n^2 : N_G(1) = \emptyset\right\}$.
- $J_n := \left\{G \in \Gamma_n^2 \setminus I_n : \text{ if } x \in D_2(G), \text{ then } N_G(x) \cap D_1(G) = \{a_1(G)\}\right\}$.
- For $2 \leq i \leq n$, $J_n(i) := \{G \in J_n : a_1(G) = i\}$.

So, $G \in I_n$ if and only if 1 is isolated in G, and for $G \in \Gamma_n^2 \setminus I_n$ we have $G \in J_n$ if and only if every path $1, x, y$ in G satisfies $x = a_1(G)$. Also, $J_n = \bigcup_{i=2}^n J_n(i)$, and $G \in J_n(n)$ if and only if $N_G(1) = \{n\}$. It is straightforward to verify the following facts:

- I_n is a subcomplex of Γ_n^2, as is $I_n \cup J_n$.
- If $G \in J_n(i)$ and $e \in E(G) \setminus \{1a_1(G)\}$, then $G - e \in J_n(i)$.
- If $G \in J_n(i)$, then $G - 1a_1(G) \in I_n \cup \bigcup_{j>i} J_n(j)$.

It follows from these facts that the pairwise disjoint, order convex subsets I_n,$J_n(2), \dots$, $J_n(n)$, $\Gamma_n^2 \setminus (I_n \cup J_n)$ cover Γ_n^2 and satisfy condition $\mathcal{P}$ of Cluster Lemma 3.2.

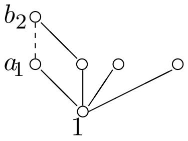
Set $\mathsf{P} = \mathsf{P}_1 := I_n \cup J_n(n)$. Note that P is a subcomplex of Γ_n^2. Define $f = f_1 : \mathsf{P} \to \binom{[n]}{2}$ by $f_1(G) = \{1, n\}$ for all G. Then f satisfies conditions (A)–(D) of Pairing Lemma 3.4, with $\mathsf{P}_f = I_n$. Since $G - \{1, n\} \in I_n$ for each $G \in J_n(n)$, $M_1 := M_f$ is an acyclic perfect matching on P_1.

Now set $\mathsf{P} = \mathsf{P}_n := \Gamma_n^2 \setminus (I_n \cup J_n)$. Note that $G \in \mathsf{P}_n$ if and only if the set

$$R_2(G) := \{x \in D_2(G) : N_G(x) \cap D_1(G) \neq \{a_1(G)\}\}$$

is not empty. For $G \in \mathsf{P}_n$, set $b_2(G) = \min R_2(G)$. Note that in any proper 2-coloring ϕ of G

$$\phi(a_1(G)) \neq \phi(1) = \phi(b_2(G)).$$



Therefore, if $G \in \mathsf{P}_n$ with $\{a_1(G), b_2(G)\} \notin E(G)$, then $G + \{a_1(G), b_2(G)\} \in \mathsf{P}_n$. It follows that if we define $f = f_n : \mathsf{P}_n \to \binom{[n]}{2}$ by

$$f_n(G) := \{a_1(G), b_2(G)\},$$

then f satisfies condition (A) of Pairing Lemma 3.4. Adding or removing $f(G)$ from $E(G)$ changes neither $D_1(G)$ nor $R_2(G)$, so f satisfies conditions (B) and (C), and every element of P is an endpoint of some arc in M_f. As stated in section 3, we order the vertices of Γ_n^2, which are the edges of K_n, lexicographically. Say $G \in \mathsf{P}_f$ and $H = G^+ - e$ for some $e \in E(G) \setminus \{f(G)\}$. If $e \neq \{1, a_1(G)\}$, then $a_1(H) = a_1(G)$. Also, $R_2(H) \subseteq R_2(G)$, so either $R_2(H) = \emptyset$ and $H \notin \mathsf{P}$ or $b_2(H) \geq b_2(G)$. Say $e = \{1, a_1(G)\}$. If $H \in \mathsf{P}$, then we again have $R_2(H) \subseteq R_2(G)$, since the edge $f(G)$ is not in any path $1, x, y$ in H. Thus $b_2(H) \geq b_2(G)$ and $a_1(H) > a_1(G)$. Therefore, f satisfies condition (D) of Lemma 3.4 and $M_n := M_f$ is an acyclic perfect matching on P_n.

We use our inductive hypothesis to define our matching on the remaining parts of Γ_n^2. Fix $i \in \{2, \dots, n-1\}$ and set $\mathsf{P} = \mathsf{P}_i := J_n(i)$. For $X \subseteq [n] \setminus [i]$, set

$$\mathsf{P}(X) := \{G \in \mathsf{P} : N_G(1) = X \cup \{i\}\}.$$

The order convex subsets $\mathsf{P}(X)$ of P satisfy condition $\mathcal{P}$ of Cluster Lemma 3.2, and we will define an acyclic matching M_i on $D(\mathsf{P}_i)$ by defining an acyclic matching $M(X)$ on each $D(\mathsf{P}(X))$ and then setting $M_i := \bigcup_X M(X)$. Set $Y = [n] \setminus (X \cup \{1\})$ and let $m = |Y|$. Note that $G = ([n], E) \in \mathsf{P}(X)$ if and only if G satisfies the following conditions:

- $N_G(1) = X \cup \{i\}$.
- $N_G(x) = \{1\}$ for all $x \in X$.
- G_Y is bipartite.

If $m = 1$, then $i = 2$ and $N_G(1) = [n] \setminus \{1\}$ for all $G \in \mathsf{P}(X)$. It follows that the unique element of $\mathsf{P}(X)$ is the tree T in which vertex 1 is the only vertex which is not a leaf. In this case we define $M(X) = \emptyset$, so T is a critical point of $M(X)$.

Now assume $m > 1$. Define $\psi : Y \to [m]$ by the following:

- $\psi(i) = 1$.
- If $x, y \in Y \setminus \{i\}$ with $x < y$, then $1 < \psi(x) < \psi(y)$.

Let Γ_Y^2 be the complex of bipartite graphs on vertex set Y. Then ψ determines a simplicial isomorphism from Γ_Y^2 to Γ_m^2. For $G \in \mathsf{P}(X)$ let $\pi(G) = G_Y$. By the characterization of $\mathsf{P}(X)$ given just above, $\psi\pi$ determines a digraph isomorphism between $D(\mathsf{P}(X))$ and $D(\Gamma_m^2)$. In this case, by the inductive hypothesis there exists a Morse matching M^m on $D(\Gamma_m^2)$ whose critical points are the elements of $\mathcal{T}(m)$. Set $\tau = (\psi\pi)^{-1}$ and define

$$M(X) := \{(\tau(G), \tau(H)) : (G, H) \in M^m\}.$$

Then $M(X)$ is an acyclic matching on $\mathsf{P}(X)$. The critical points of $M(X)$ are those trees T satisfying the following conditions:

- $a_1(T) = i$.
- $N_T(1) = X \cup \{i\}$.
- $N_T(x) = \{1\}$ for all $x \in X$.
- For $2 \le r \le md(T)$, there is some element of $\cup_{j>r} D_j(T)$ that is larger than $a_r(T)$.
- $|D_{md(T)}(T)| > 1$.

In other words, the critical points of $M(X)$ are all $T \in \mathcal{T}(n)$ such that $a_1(T) = i$ and $D_1(T) = X \cup \{i\}$. Now the critical points of $M_i = \bigcup_X M(X)$ are exactly those trees $T \in \mathcal{T}_n$ such that $a_1(T) = i$. Finally, by Cluster Lemma 3.2,

$$M^n := \bigcup_{i=1}^{n} M_i$$

is a Morse matching on $D(\Gamma_n^2)$ whose critical points are the elements of $\mathcal{T}(n)$. The proof of Theorem 4.1 is complete.

5. Complexes of highly colorable graphs. In this section, we will prove Theorems 1.3 and 1.4 by finding Morse matchings with the appropriate number of critical points in the appropriate dimensions. First we will find a Morse matching on Γ_n^t for arbitrary n, t whose critical points are graphs which are quite large, given that they are t-colorable, thereby proving Proposition 1.5. The notation introduced in the next definition will be used repeatedly.

Definition 5.1. *For a graph $G = ([n], E)$, G' is the subgraph of G induced on $[n] \setminus \{1\}$.*

For $n \in \mathbb{N}$ and $t \in [n]$, define

$$\mathcal{I}(n,t) := \left\{G \in \Gamma_n^t : \chi(G') \leq t-1\right\}.$$

Note that if H is any graph on vertex set $W := \{2, 3, \dots, n\}$ with $\chi(H) \leq t-1$ and G is any graph on vertex set $[n]$ such that the subgraph of G induced on W is H, then $G \in \mathcal{I}(n,t)$.

For $G \in \Gamma_n^t \setminus \mathcal{I}(n,t)$, define

$$\Theta(G) := \{v \in V(G') : |N_{G'}(v)| < t-1\}$$

and

$$\mathcal{J}(n,t) := \left\{G \in \Gamma_n^t \setminus \mathcal{I}(n,t) : \Theta(G) \neq \emptyset\right\}.$$

Also, define

$$\mathcal{K}(n,t) := \mathcal{I}(n,t) \bigcup \mathcal{J}(n,t).$$

Lemma 5.2. *Let $n \in \mathbb{N}$ and $t \in [n]$. Then $\mathcal{K}(n,t)$ is a subcomplex of Γ_n^t and $D(\mathcal{K}(n,t))$ admits an acyclic perfect matching.*

Proof. Removing an edge from any graph $G = ([n], E)$ cannot increase $\chi(G')$ or increase $|N_{G'}(v)|$ for $v \in V(G')$, so both $\mathcal{I}(n,t)$ and $\mathcal{K}(n,t)$ are subcomplexes of Γ_n^t. By Cluster Lemma 3.2, it is now sufficient to produce acyclic perfect matchings on $\mathcal{I}(n,t)$ and $\mathcal{J}(n,t)$.

We begin with $\mathcal{I}(n,t)$. Define

$$\mathcal{I}_0(n,t) := \{G \in \mathcal{I}(n,t) : \{1,2\} \notin E(G)\}.$$

For $G \in \mathcal{I}_0(n,t)$, set

$$G^+ := G + \{1,2\}.$$

Then $G' = (G^+)'$, so $G^+ \in \mathcal{I}(n,t)$. For $G \in \mathcal{I}(n,t) \setminus \mathcal{I}_0(n,t)$, set

$$G^- := G - \{1,2\}.$$

Then $G^- \in \mathcal{I}_0(n,t)$ and $G = (G^-)^+$, so

$$\mathcal{I}(n,t) = \mathcal{I}_0(n,t) \bigcup \left\{G^+ : G \in \mathcal{I}_0(n,t)\right\}.$$

Now define $f(G) := \{1,2\}$ for all $G \in \mathcal{I}(n,t)$. Setting $\mathsf{P} = \mathcal{I}(n,t)$ and using the language of Pairing Lemma 3.4, we have $\mathsf{P}_f = \mathcal{I}_0(n,t)$. By the arguments given just above, f satisfies conditions (A), (B), and (C) of Pairing Lemma 3.4. Since f is a constant function, f satisfies condition (D). It now follows that

$$M_f = \left\{(G, G^-) : G \in \mathcal{I}(n,t) \setminus \mathcal{I}_0(n,t)\right\}$$

is an acyclic perfect matching on $D(\mathcal{I}(n,t))$.

For $G \in \mathcal{J}(n,t)$, define

$$\theta(G) := \max \Theta(G)$$

and

$$\mathcal{J}_0(n,t) := \{G \in \mathcal{J}(n,t) : \{1, \theta(G)\} \notin E(G)\}\,.$$

For $G \in \mathcal{J}_0(n,t)$, set

$$G^+ := G + \{1, \theta(G)\}\,.$$

First note that $|N_{G^+}(\theta(G))| < t$, so one can obtain a proper t-coloring of G^+ from a proper t-coloring ϕ of G by assigning to $\theta(G)$ any color i such that $\phi^{-1}(i) \cap N_{G^+}(\theta(G)) = \emptyset$. Hence $G^+ \in \Gamma_n^t$. Furthermore $G' = (G^+)'$, so $G^+ \in \mathcal{J}(n,t)$ and $\theta(G^+) = \theta(G)$. For $G \in \mathcal{J}(n,t) \setminus \mathcal{J}_0(n,t)$, set

$$G^- = G - \{1, \theta(G)\}\,.$$

Then $G^- \in \mathcal{J}(n,t)$ and $\theta(G^-) = \theta(G)$, so $G^- \in \mathcal{J}_0(n,t)$. It follows that

$$\mathcal{J}(n,t) = \mathcal{J}_0(n,t) \bigcup \left\{G^+ : G \in \mathcal{J}_0(n,t)\right\}.$$

For $G \in \mathcal{J}(n,t)$, define $f(G) := \{1, \theta(G)\}$. Then, again using the language of Pairing Lemma 3.4 and setting $\mathsf{P} = \mathcal{J}(n,t)$, we have $\mathsf{P}_f = \mathcal{J}_0(n,t)$. By the arguments given just above, f satisfies conditions (A), (B), and (C) of the lemma. Say $G \in \mathcal{J}_0(n,t)$ and $e \in E(G) \setminus \{\{1, \theta(G)\}\}$ with $G^+ - e \in \mathcal{J}(n,t)$. For any $H \in \mathcal{J}(n,t)$ and $d \in E(H)$ such that $H - d \in \mathcal{J}(n,t)$, we have $\theta(H) \le \theta(H-d)$. Thus

$$f(G) = f(G^+) \preceq f(G^+ - e),$$

so f satisfies condition (D). Therefore,

$$M_f = \left\{(G, G^-) : G \in \mathcal{J}(n,t) \setminus \mathcal{J}_0(n,t)\right\}$$

is an acyclic perfect matching on $D(\mathcal{J}(n,t))$. □

Note that we now have a Morse matching on Γ_n^t whose critical points are the elements of $\Gamma_n^t \setminus \mathcal{K}(n,t)$, that is, the graphs $G \in \Gamma_n^t$ which satisfy the conditions

- $\chi(G') = t$, and
- $|N_{G'}(v)| \ge t-1$ for all $v \in V(G')$.

Each such graph has at least $\lceil \frac{(n-1)(t-1)}{2} \rceil$ edges, and Proposition 1.5 now follows from Theorem 3.1. When $t = n-k$ and k is small, the graphs satisfying these conditions have highly restricted structure, as we shall see. Recall the notation $\overline{G}$ for the complement of G, that is, the graph on the same vertex set with complementary edge set.

Say $k = 2$, so $t = n-2$. If $\overline{G'}$ contains two nonadjacent edges vw and xy, then we can construct a proper $(t-1)$-coloring of G' by assigning one color to v and w, another color to x and y, and $t-3 = n-5$ additional colors to the remaining $n-5$ vertices of G'. Therefore, if $G \in \Gamma_n^t \setminus \mathcal{K}(n,t)$, then $\overline{G'}$ does not contain two nonadjacent edges. Also, each vertex of G' has degree at least $n-3$ in G', so $\overline{G'}$ has no vertex of degree more than 1. On the other hand, $\overline{G'}$ must contain at least one edge, since otherwise $\chi(G') = n-1$. Therefore,

$$\Gamma_n^t \setminus \mathcal{K}(n,t) = \left\{G \in \Gamma_n^t : |E(\overline{G'})| = 1\right\}.$$

For $\{x,y\} \in \binom{V(G')}{2}$, define

$$\mathsf{P}(x,y) := \left\{G \in \Gamma_n^t : E(\overline{G'}) = \{xy\}\right\}.$$

The pairwise disjoint, order convex subsets $\mathcal{K}(n,t)$ and $\mathsf{P}(x,y)$ (where $\{x,y\}$ runs through all of $\binom{V(G')}{2}$)) cover Γ_n^t and satisfy condition $\mathcal{P}$ of Cluster Lemma 3.2, so we will concentrate on the posets $\mathsf{P}(x,y)$ and then apply the lemma.

Fix $\{x,y\} \in \binom{V(G')}{2}$ with $x<y$. If $G \in \mathsf{P}(x,y)$, then every proper t-coloring of G' assigns one color to both of x,y and all of the $n-3$ remaining colors to the remaining $n-3$ vertices of G'. It follows that either

- $N_G(1) \subseteq V(G') \setminus \{x,y\}$, or
- there is some $z \in V(G') \setminus \{x,y\}$ such that $z \notin N_G(1)$.

Conversely, if $G = ([n], E)$ with $E(\overline{G'}) = \{xy\}$ and $N_G(1)$ satisfies either of the two conditions itemized just above, then $G \in \mathsf{P}(x,y)$. Let $G(x,y)$ be the graph in $\mathsf{P}(x,y)$ defined by

$$N_{G(x,y)}(1) := V(G') \setminus \{x,y\}\,.$$

Set

$$\mathsf{P}_1(x,y) := \mathsf{P}(x,y) \setminus \{G(x,y)\}$$

and

$$\mathsf{P}_0(x,y) := \{G \in \mathsf{P}_1(x,y) : 1x \notin E(G)\}\,.$$

Then $\mathsf{P}_1(x,y)$ is an order ideal in $\mathsf{P}(x,y)$, and it is straightforward to show that

$$M(x,y) := \{(G+1x, G) : G \in \mathsf{P}_0(x,y)\}$$

is an acyclic perfect matching on $D(\mathsf{P}_1(x,y))$. Thus $M(x,y)$ is an acyclic matching on $D(\mathsf{P}(x,y))$ whose unique critical point is $G(x,y)$. Now applying Cluster Lemma 3.2 gives the following result, which is stronger than Theorem 1.3.

THEOREM 5.3. *There is a Morse matching on Γ_n^{n-2}, whose critical points are the graphs $G(x,y)$ defined by*

$$E(\overline{G(x,y)}) := \{1x, 1y, xy\}\,,$$

for all $\{x,y\} \in \binom{[n-1]}{2}$. Therefore, Γ_n^{n-2} has the homotopy type of a wedge of $\binom{n-1}{2}$ spheres of dimension $\binom{n}{2} - 4$.

To handle the case $t = n-3$ we introduce additional notation. Let $j \in \mathbb{N}$. Then Ma_j will denote a graph with $2j$ vertices and j pairwise nonadjacent edges. If these edges are $x_1y_1, \ldots, x_jy_j$, then the graph may be denoted by $\mathsf{Ma}(x_1,y_1;\ldots;x_j,y_j)$. Also, Cy_j will denote a graph with j vertices and j edges which form a cycle of length j. If the edges are $x_1x_2, \ldots, x_{j-1}x_j, x_jx_1$, then the graph may be denoted by $\mathsf{Cy}(x_1,\ldots,x_j)$. Finally, Pa_j will denote a graph with $j+1$ vertices and j edges which form a path of length j. If these edges are $x_1x_2,\ldots,x_jx_{j+1}$, then the graph may be denoted by $\mathsf{Pa}(x_1,\ldots,x_{j+1})$.

For any pair of integers $1 \le t \le n$, define a set of graphs

$$\Lambda(n,t) := \{H : V(H) = [2,n], \chi(H) = t \text{ and } |N_H(v)| \ge t-1 \text{ for all } v \in [2,n]\}\,.$$

For $H \in \Lambda(n,t)$ set

$$\mathsf{K}(H) := \left\{G \in \Gamma_n^t : G' = H\right\}.$$

Note that $\mathsf{K}(H) \subseteq \Gamma_n^t \setminus \mathcal{K}(n,t)$ for all $H \in \Lambda(n,t)$, and that as H ranges over $\Lambda(n,t)$ the pairwise disjoint, order convex subsets $\mathsf{K}(H)$ cover $\Gamma_n^t \setminus \mathcal{K}(n,t)$.

For any graph G, set

$$U(G) := \{v \in V(G) : N_G(v) \neq V(G) \setminus \{v\}\}\,.$$

Note that if G is not a complete graph, then $U(G) \neq \emptyset$. In particular, if $G \in \Gamma_n^t$ and $t < n-1$, then $U(G') \neq \emptyset$.

For $H \in \Lambda(n,t)$, set

$$\mathsf{L}(H) := \{G \in \mathsf{K}(H) : V(G') \setminus U(G') \not\subseteq N_G(1)\}\,,$$

so, in plain English, $\mathsf{L}(H)$ consists of those $G \in \mathsf{K}(H)$ such that there is some $v \in [2,n]$ with $N_G(v) = [2,n] \setminus \{v\}$. Note that $\mathsf{L}(H)$ is an order ideal in $\mathsf{K}(H)$ and is therefore convex. A special case of the following lemma was used in the proof of Theorem 5.3.

Lemma 5.4. *Given integers $n-2 \geq t \geq 1$ and $H \in \Lambda(n,t)$, we define for each $G \in \mathsf{L}(H)$*

$$\alpha(G) = \max U(H).$$

Set also

$$\mathsf{L}_0(H) := \{G \in \mathsf{L}(H) : 1\alpha(G) \notin E(G)\}\,.$$

Then $\{(G + 1\alpha(G), G) : G \in \mathsf{L}_0(H)\}$ is an acyclic perfect matching on $D(\mathsf{L}(H))$.

Proof. We will apply Pairing Lemma 3.4 to the order convex subset $\mathsf{P} := \mathsf{L}(H)$ of Γ_n^t. Define $f : \mathsf{P} \to \binom{[n]}{2}$ by $G \mapsto \{1, \alpha(G)\}$. Then $\mathsf{P}_f = \mathsf{L}_0(H)$. Let $G \in \mathsf{P}_f$ and let ϕ be a proper t-coloring of G'. We can extend ϕ to a proper t-coloring of $G^+ := G + 1\alpha(G)$ by giving vertex 1 the same color as any element of $(V(G') \setminus U(G')) \setminus N_G(1)$. Therefore, $G^+ \in \Gamma_n^t$. Since changing $N_G(1)$ does not change G', we have $G^+ \in \mathsf{L}(H) = \mathsf{P}$. Moreover, $\alpha(G^+) = \alpha(G)$. It follows that f satisfies conditions (A), (B), and (C) of Pairing Lemma 3.4. Finally, if $G \in \mathsf{P}_f$ and $e \in E(G)$ with $H := G^+ - e \in \mathsf{P}$, then $\alpha(H) \geq \alpha(G^+)$, so

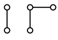
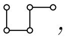
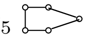
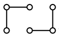
$$f(G) = f(G^+) \preceq f(H).$$

Therefore, f satisfies condition (D). $\square$

Now assume $t = n-3$. If G, H are graphs with disjoint vertex sets, then $G + H$ will denote the graph $(V(G) \cup V(H), E(G) \cup E(H))$. We want to describe $\Lambda(n,t)$. Let $G \in \Gamma_n^t$.

If $G \notin \mathcal{K}(n,t)$, then every vertex of G' has degree at least $n-4$, so every vertex of $\overline{G'}$ has degree $0, 1$, or 2. It follows that the induced subgraph $\overline{G'}_{U(G')}$ is a union of paths and cycles. If $\overline{G'}$ contains a subgraph isomorphic to either Ma_3 or $\mathsf{Cy}_3 + \mathsf{Ma}_1$, then $\chi(G') \leq n-4$ and $G \in \mathcal{K}(n,t)$. Conversely, in order for G' to be $(n-3)$-colorable it is necessary that $\overline{G'}$ contain either Ma_2 or Cy_3. The next proposition follows.

Proposition 5.5. *Let $n \in \mathbb{N}$ with $n \geq 4$. Set $t = n-3$ and let $G \in \Gamma_n^t$. Then $G \in \Gamma_n^t \setminus \mathcal{K}(n,t)$ if and only if one of the following conditions holds:*

- $|U(G')| = 3$ *and* $\overline{G'}_{U(G')}$ *is isomorphic to* Cy_3 .
- $|U(G')| = 4$ *and* $\overline{G'}_{U(G')}$ *is isomorphic to one of* Ma_2 , Pa_3 , *or* Cy_4 .
- $|U(G')| = 5$ *and* $\overline{G'}_{U(G')}$ *is isomorphic to one of* $\mathsf{Ma}_1 + \mathsf{Pa}_2$ , Pa_4 , *or* Cy_5 .
- $|U(G')| = 6$ *and* $\overline{G'}_{U(G')}$ *is isomorphic to* $\mathsf{Pa}_2 + \mathsf{Pa}_2$ .

Therefore, $H = ([2,n], E)$ lies in $\Lambda(n,t)$ if and only if $\overline{H}_{U(H)}$ is isomorphic to one of Cy_3, Ma_2, Pa_3, Cy_4, $\mathsf{Ma}_1 + \mathsf{Pa}_2$, Pa_4, Cy_5, *or* $\mathsf{Pa}_2 + \mathsf{Pa}_2$.

It is natural to try to apply Cluster Lemma 3.2 to the posets $\mathsf{K}(H)$ as H ranges over $\Lambda(n,t)$, as Lemma 5.4 gives us some insight into how to handle each $\mathsf{K}(H)$. However, this approach will not yield the desired result before it is slightly modified, as we now describe. For $\{a < b < c < d\} \subseteq [2,n]$, set

$$\mathsf{J}(a,b,c,d) := \bigcup_{\substack{H \in \Lambda(n,t) \\ U(H) = \{a,b,c,d\}}} \mathsf{K}(H).$$

We consider all subsets $\mathsf{K}(H)$ of $\Gamma_n^t \setminus \mathcal{K}(n,t)$ as H runs through all elements of $\Lambda(n,t)$ such that $|U(H)| \neq 4$ along with all the subsets $\mathsf{J}(a,b,c,d)$ as $\{a<b<c<d\}$ runs through $\binom{[2,n]}{4}$. First note that each $\mathsf{K}(H)$ is convex, as if $F, G, I \in \Gamma_n^t$ with $E(F) \subseteq E(G) \subseteq E(I)$ and $F' = I' = H$, then $G' = H$. It is also the case that each $\mathsf{J}(a,b,c,d)$ is convex. Indeed, if $F, I \in \mathsf{J}(a,b,c,d)$, then $U(F') = U(I') = \{a,b,c,d\}$, and thus if $E(F) \subseteq E(G) \subseteq E(I)$, then $U(G') = \{a,b,c,d\}$ so $G \in \mathsf{J}(a,b,c,d)$.

Now we show that the subsets under consideration satisfy condition $\mathcal{P}$ of Cluster Lemma 3.2. We use $\sim$ to indicate the equivalence relation determined by the partition of $\Gamma_n^t \setminus \mathcal{K}(n,t)$ into these subsets. Thus if $F \sim G$, then $U(F') = U(G')$. Conversely, by Proposition 5.5, we see that if $U(F') = U(G')$ and $F \not\sim G$, then $|U(G')| \in \{5,6\}$ and for all I we have $I \sim G$ if and only if $I' = G'$. We begin with antisymmetry. Note that for any $F, G \in \Gamma_n^t \setminus \mathcal{K}(n,t)$, if $E(F) \subseteq E(G)$, then $U(F') \supseteq U(G')$. If $U(F') \neq U(G')$, then there are no $I \sim G, J \sim F$ such that $E(I') \subseteq E(J')$. Say $U(F') = U(G')$. If $F \not\sim G$, then $E(F_{U(F')}) \subsetneq E(G_{U(G')})$ and, as noted above, if $I \sim G$ and $J \sim F$, then $I' = G'$, $J' = F'$, and $E(I) \not\subseteq E(J)$. Now we prove transitivity. It suffices to show that if $G_1 \sim G_2$ and $E(G_1) \subseteq E(I_1)$ and $E(F_1) \subseteq E(G_2)$, then there exist $I_2 \sim I_1$ and $F_2 \sim F_1$ with $E(F_2) \subseteq E(I_2)$. If $G_1' = G_2'$, we can take $I_2 = I_1$ and let F_2 be the graph obtained from F_1 by removing all edges which contain vertex 1. If $G_1' \neq G_2'$, then $G_1, G_2 \in \mathsf{J}(a,b,c,d)$ for some $a<b<c<d$, and inspection of the cases listed in Proposition 5.5 shows that $I_1 \in \mathsf{J}(a,b,c,d)$, and we are done.

Now we will define useful acyclic matchings for each of the subsets described above, and Theorem 1.4 will follow from Cluster Lemma 3.2. For $H \in \Lambda(n,t)$, set

$$\mathsf{R}(H) := \mathsf{K}(H) \setminus \mathsf{L}(H).$$

Note that, by Lemma 5.4 and the fact that $\mathsf{L}(H)$ is an ideal in $\mathsf{K}(H)$, any Morse matching on $\mathsf{R}(H)$ can be extended to one on $\mathsf{K}(H)$ with the same critical points.

We begin with the posets $\mathsf{J}(a,b,c,d)$. Fix $\{a<b<c<d\} \subseteq \binom{[2,n]}{4}$ and set

$$\mathsf{L}(a,b,c,d) := \bigcup_{\substack{H \in \Lambda(n,t) \\ U(H) = \{a,b,c,d\}}} \mathsf{L}(H).$$

Note that $\mathsf{L}(a,b,c,d)$ is an ideal in $\mathsf{J}(a,b,c,d)$. Indeed, for all $G \in \mathsf{J}(a,b,c,d)$ we have $U(G') = \{a,b,c,d\}$ and $G \in \mathsf{L}(a,b,c,d)$ if and only if $N_G(1) \not\supseteq \{a,b,c,d\}$. Thus $\mathsf{L}(a,b,c,d)$ is order convex. It is also the case that the order convex subsets $\mathsf{L}(H)$, as H runs over all elements of $\Gamma(n,t)$ such that $U(H) = \{a,b,c,d\}$, satisfy condition $\mathcal{P}$ of Cluster Lemma 3.2. Indeed, in this case the relation $\leq_c$ from Cluster Lemma 3.2 is simply the containment relation on $E(H)$. Thus by Cluster Lemma 3.2 and Lemma

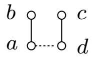
5.4, there is an acyclic perfect matching on $D(\mathsf{L}(a,b,c,d))$. Therefore, if we define

$$\mathsf{R}(a,b,c,d) := \bigcup_{\substack{H\in\Lambda(n,t)\\ U(H)=\{a,b,c,d\}}} \mathsf{R}(H),$$

then any acyclic matching on $\mathsf{R}(a,b,c,d)$ extends to one on $\mathsf{J}(a,b,c,d)$ with the same critical points.

For a graph X on vertex set $W \subseteq [2,n]$ which has no vertex of degree $|W|-1$, $H[X]$ will denote the unique graph $H=([2,n],E)$ such that $\overline{H}_{U(H)} = X$. Set

$$\begin{aligned}
\mathsf{Q}_1 &:= \mathsf{R}(H[\mathsf{Ma}(a,b;c,d)]) \bigcup \mathsf{R}(H[\mathsf{Pa}(b,a,d,c)]),\\
\mathsf{Q}_2 &:= \mathsf{R}(H[\mathsf{Ma}(a,c;b,d)]) \bigcup \mathsf{R}(H[\mathsf{Pa}(c,a,d,b)]),\\
\mathsf{Q}_3 &:= \mathsf{R}(H[\mathsf{Ma}(a,d;b,c)]) \bigcup \mathsf{R}(H[\mathsf{Pa}(d,a,c,b)]).
\end{aligned}$$



Then the Q_i are pairwise disjoint dual order ideals in $\mathsf{R}(a,b,c,d)$ and therefore satisfy condition $\mathcal{P}$ of Cluster Lemma 3.2. We will find an acyclic matching on $D(\mathsf{R}(a,b,c,d))$ whose critical points are those graphs not contained in $\bigcup_{i=1}^{3}\mathsf{Q}_i$ by finding an acyclic perfect matching on each $D(\mathsf{Q}_i)$. The $D(\mathsf{Q}_i)$ are pairwise isomorphic, so it suffices to find an acyclic perfect matching on $D(\mathsf{Q}_1)$. If $G \in \mathsf{R}(H[\mathsf{Ma}(a,b;c,d)])$, then every proper $(n-3)$-coloring of G' is obtained by assigning one color to vertices a,b, another color to vertices c,d, and all of the remaining $n-5$ colors to the remaining $n-5$ vertices of G'. Such a coloring can be extended to a proper coloring of G, so $N_G(1)\cap\{a,b,c,d\}$ must be contained in either $\{a,b\}$ or $\{c,d\}$. Conversely, if $G=([n],E)$ satisfies

1. $E(\overline{G'}) = \{ab, cd\}$,
2. $V(G')\setminus\{a,b,c,d\} \subseteq N_G(1)$, and
3. $N_G(1)\cap\{a,b,c,d\}$ is contained in one of $\{a,b\}$ or $\{c,d\}$,

then $G \in \mathsf{R}(H[\mathsf{Ma}(a,b;c,d)])$, so the three conditions just given characterize the set $\mathsf{R}(H[\mathsf{Ma}(a,b;c,d)])$. Now a similar argument shows that $\mathsf{R}(H[\mathsf{Pa}(b,a,d,c)])$ is characterized by conditions 2, 3 and

$1'$. $E(\overline{G'}) = \{ab, cd, ad\}$.

It now follows that

$$\mathsf{Q}_1 = \mathsf{R}(H[\mathsf{Ma}(a,b;c,d)]) \cup \{G+ad : G\in \mathsf{R}(H[\mathsf{Ma}(a,b;c,d)])\}$$

and that

$$M_1 := \{(G+ad, G) : G \in \mathsf{R}(\mathsf{Ma}(a,b;c,d))\}$$

is an acyclic perfect matching on $D(\mathsf{Q}_1)$.

Now $\mathsf{R}(a,b,c,d)\setminus\bigcup_{i=1}^{3}\mathsf{Q}_i$ is a union of the pairwise disjoint order convex subposets $\mathsf{R}(H)$, where $\overline{H}_{U(H)}$ is one of the three graphs $\mathsf{Cy}(a,b,c,d)$, $\mathsf{Cy}(a,b,d,c)$, or $\mathsf{Cy}(a,c,b,d)$, or one of the nine graphs on vertex set $\{a,b,c,d\}$ which are isomorphic to Pa_3, but not one of $\mathsf{Pa}(c,a,d,b)$, $\mathsf{Pa}(d,a,c,b)$, or $\mathsf{Pa}(b,a,d,c)$. These posets satisfy condition $(\mathcal{P})$ of Cluster Lemma 3.2. So, we will now produce acyclic matchings on the posets $\mathsf{R}(H)$ for all $H\in\Lambda(n,t)$, other than $H[X]$ when $X\approx\mathsf{Ma}_2$. For any $H\in\Lambda(n,t)$, define

$$\Delta(H) := \{N_G(1)\cap V(H) : G\in R(H)\}.$$

Then $\Delta(H)$ is a simplicial complex and $D(\mathsf{R}(H))$ is isomorphic to $D(\Delta(H))$ via the map $G\mapsto N_G(1)\cap V(H)$. Thus a Morse matching on $D(\Delta(H))$ can be lifted to a

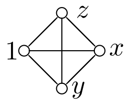
Morse matching on $D(\mathsf{R}(H))$. We will describe Morse matchings for each possible isomorphism type of H. Each of the claims made without proof during these descriptions can be verified by observation, as the complexes $\Delta(H)$ are quite small. Note that if $G \in \mathsf{R}(H[X])$, then any proper $(n-3)$-coloring ϕ of G satisfies the following conditions:

- Exactly $|V(X)| - 2$ colors are used to color $V(X)$, and $\phi(1)$ is one of these colors.
- All of the remaining $n - |V(X)| - 1$ colors are used to color the remaining $n - |V(X)| - 1$ vertices of $G' = H[X]$.

The promised Morse matchings are described below.

- Say $\overline{H}_{U(H)} = \mathsf{Cy}(x, y, z)$ and $G \in \mathsf{R}(H)$. Then every proper $(n-3)$-coloring of G is obtained by assigning one color to $1, x, y$, and z and all of the remaining $n-4$ colors to the remaining vertices of G'. Therefore, $N_G(1) \cap \{x, y, z\} = \emptyset$ and the only face in $\Delta(H)$ is the empty face. The associated acyclic matching on $D(\mathsf{R}(H))$ is the empty matching, with one critical point, namely the graph G, defined by

$$E(\overline{G}) := \{1x, 1y, 1z, xy, xz, yz\}\,.$$



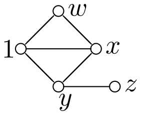
There are $\binom{n-1}{3}$ such graphs G in Γ_n^t.

- Say $\overline{H}_{U(H)} = \mathsf{Pa}(w, x, y, z)$ with $w < z$, and $G \in \mathsf{R}(H)$. Then every proper $(n-3)$-coloring of G is obtained by assigning one color γ to w and x, another color γ' to y and z, the remaining $n-5$ colors to the elements of $V(G) \setminus \{1, w, x, y, z\}$, and giving vertex 1 either color γ or γ'. It follows that the maximal elements of $\Delta(H)$ are $\{w, x\}$ and $\{y, z\}$. Now

$$\{(\{w\}, \emptyset), (\{w, x\}, \{x\}), (\{y, z\}, \{y\})\}$$

is a Morse matching on $\Delta(H)$ whose unique critical point is the 0-simplex $\{z\}$. The unique critical point of the associated acyclic matching on $D(\mathsf{R}(H))$ is the graph G defined by

$$E(\overline{G}) := \{1w, 1x, 1y, wx, xy, yz\}\,.$$



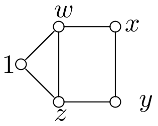
There are $12\binom{n-1}{4}$ posets $\mathsf{R}(\mathsf{Pa}(w, x, y, z))$ to be considered. However, $3\binom{n-1}{4}$ of them are contained in the posets Q_i described above. Therefore, there are $9\binom{n-1}{4}$ such graphs G in Γ_n^t that are critical points of the matchings just described.

- Say $\overline{H}_{U(H)} = \mathsf{Cy}(w, x, y, z)$ with $\min\{w, x, y, z\} = w$, and $G \in \mathsf{R}(H)$. Then every proper $(n-3)$-coloring ϕ of G is obtained by coloring w, x, y, z with two colors γ, γ' so that $\phi(w) \neq \phi(y)$ and $\phi(x) \neq \phi(z)$, coloring the remaining $n-5$ vertices in $V(G')$ with all of the remaining $n-5$ colors, and then giving vertex 1 color γ or γ'. It follows that the maximal faces of $\Delta(H)$ are $\{w, x\}$, $\{w, z\}$, $\{x, y\}$, and $\{y, z\}$ and that

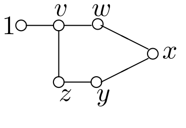
$$\{(\{w\}, \emptyset), (\{w, x\}, \{x\}), (\{w, z\}, \{z\}), (\{y, z\}, \{y\})\}$$

is a Morse matching on $D(\Delta(H))$ whose unique critical point is the 1-simplex $\{x,y\}$. The unique critical cell of the associated acyclic matching on $D(\mathsf{R}(H))$ is the graph G defined by

$$E(\overline{G}) := \{1w, 1z, wx, xy, yz, wz\}\,.$$



There are $3\binom{n-1}{4}$ such graphs G.

- Say $\overline{H}_{U(H)} = \mathsf{Pa}(a,b)+\mathsf{Pa}(x,y,z)$ and $G \in \mathsf{R}(H)$. Then every proper $(n-3)$-coloring of G is obtained by coloring a and b with some color γ, coloring x, y, and z with two additional colors δ, δ' so that x and z do not get the same color, coloring the remaining $n-6$ vertices in $V(G')$ with the remaining $n-6$ colors, and then giving vertex 1 one of the colors γ, δ, δ'. It follows that the maximal faces of $\Delta(H)$ are $\{a,b,x,y\}$, $\{a,b,y,z\}$, and $\{x,y,z\}$. Define
$$\Delta_y(H) := \{\sigma \in \Delta(H) : y \notin \sigma\}\,.$$
Every maximal face in $\Delta(H)$ contains vertex y, so
$$M(H) := \{(\sigma \cup \{y\}, \sigma) : \sigma \in \Delta_y(H)\}$$
is an acyclic perfect matching on $D(\Delta(H))$. Therefore, $D(\mathsf{R}(H))$ admits an acyclic perfect matching.
- Say $\overline{H}_{U(H)} = \mathsf{Pa}(v,w,x,y,z)$ with $v < z$, and $G \in \mathsf{R}(H)$. Then any proper $(n-3)$-coloring ϕ of G is obtained by assigning three distinct colors β, γ, δ to v, x, z, respectively; assigning one of β, γ to w and one of γ, δ to y so that $\phi(w) \neq \phi(y)$, assigning all of the remaining $n-6$ colors to the remaining $n-6$ vertices in $V(G')$, and assigning one of β, γ, δ to vertex 1. It follows that the maximal faces of $\Delta(H)$ are $\{v,w,x,y\}$, $\{v,w,y,z\}$, and $\{w,x,y,z\}$. Now every maximal face contains y, and we can argue, as we did in the case just above to show that $D(\mathsf{R}(H))$ admits an acyclic perfect matching.
- Say $\overline{H}_{U(H)} = \mathsf{Cy}(v,w,x,y,z)$ with $\min\{v,w,x,y,z\} = v$ and $G \in \mathsf{R}(H)$. Then every proper $(n-3)$-coloring of G is obtained by assigning three colors β, γ, δ to $\{v,w,x,y,z\}$ so as to obtain a proper 3-coloring of $C(v,x,z,w,y)$; assigning all of the remaining $n-6$ colors to the remaining $n-6$ vertices in $V(G')$; and assigning one of the colors β, γ, δ to vertex 1. It follows that the maximal faces of $\Delta(H)$ are $\{v,w,x,y\}$, $\{v,w,x,z\}$, $\{v,w,y,z\}$, $\{v,x,y,z\}$, and $\{w,x,y,z\}$, so $\Delta(H)$ is the boundary of the simplex $\{v,w,x,y,z\}$. Set
$$\Delta^v(H) := \{\sigma \in \Delta(H) : v \in \sigma\}\,.$$
Then
$$M(H) := \{(\sigma, \sigma \setminus \{v\}) : \sigma \in \Delta^v(H)\}$$
is a Morse matching on $D(\Delta(H))$ whose unique critical point is the 3-simplex $\{w,x,y,z\}$. The unique critical point of the associated acyclic matching on $D(\mathsf{R}(H))$ is the graph G defined by
$$E(\overline{G}) := \{1v, vw, wx, xy, yz, vz\}\,.$$


There are $12\binom{n-1}{5}$ such graphs G.

Table 1

Table of nonzero Betti numbers for the complex Γ^t_n of all t-colorable graphs on n vertices. All known nonzero Betti numbers occur in dimension $n(t-1) - \binom{t}{2} - 1$ except *. The values † have been calculated by computer. We have also computed $\tilde{\chi}(\Gamma^3_8) = 31,846$.

$n\backslash t$	2	3	4	5	6	7	8
3	β_1=1						
4	β_2=3	β_4=1					
5	β_3=16	β_6=6	β_8=1				
6	β_4=105	β_8=82	β_{11}=10	β_{13}=1			
7	β_5=841	$\beta^\dagger_{10}$=1535	β_{14}=272	β_{17}=15	β_{19}=1		
8	β_6=7938	?	$\beta^\dagger_{17}$=9396 $\beta^{*\dagger}_{19}$=1	β_{21}=707	β_{24}=21	β_{26}=1	
9	β_7=86311	?	?	?	β_{29}=1568	β_{32}=36	β_{34}=1

- Say $\overline{H}_{U(H)} = \mathsf{Pa}(a,b,c) + \mathsf{Pa}(x,y,z)$ and $G \in \mathsf{R}(H)$. Then every proper $(n-3)$-coloring χ of G is obtained by assigning two colors γ, γ' to $\{a,b,c\}$ so that $\chi(a) \neq \chi(c)$; assigning two additional colors δ, δ' to $\{x,y,z\}$ so that $\chi(x) \neq \chi(z)$; assigning all of the remaining $n-7$ colors to the remaining $n-7$ vertices in $V(G')$; and assigning one of the colors $\gamma, \gamma', \delta, \delta'$ to vertex 1. It follows that the maximal faces of $\Delta(H)$ are $\{a,b,c,x,y\}$, $\{a,b,c,y,z\}$, $\{a,b,x,y,z\}$, and $\{b,c,x,y,z\}$. Again, every maximal face of $\Delta(H)$ contains vertex y, and it follows that $D(\mathsf{R}(H))$ admits an acyclic perfect matching.

All possibilities for $\mathsf{R}(H)$ have now been examined, and all the critical points G of all the acyclic matchings that have been obtained satisfy

$$|E(\overline{G})| = 6.$$

There are

$$\binom{n-1}{3} + 9\binom{n-1}{4} + 3\binom{n-1}{4} + 12\binom{n-1}{5} = \binom{n-1}{3} + 12\binom{n}{5}$$

critical points. The next theorem, which is stronger than Theorem 1.4, now follows from Cluster Lemma 3.2.

Theorem 5.6. *Let $n \in \mathbb{N}$ with $n \geq 4$. Then there is a Morse matching on Γ^{n-3}_n whose critical points are $\binom{n-1}{3} + 12\binom{n}{5}$ faces of dimension $\binom{n}{2} - 7$. Therefore, Γ^{n-3}_n has the homotopy type of a wedge of $\binom{n-1}{3} + 12\binom{n}{5}$ spheres of dimension $\binom{n}{2} - 7$.*

6. Remarks. As noted in section 1, we used computer calculations to show that it is not the case that Γ^t_n has the homotopy type of a wedge of spheres of dimension $n(t-1) - \binom{t}{2} - 1$ for all n, t. Our knowledge of the homology of Γ^t_n when $n \leq 9$ is summarized in Table 1. The only pairs (n,t) for which the results are not determined by the theorems in this paper are $(7,3)$ and $(8,4)$.

As noted in the introduction, many questions remain. Is there a nice generating function (or another good description) for the Euler characteristic of Γ^t_n, similar to the result given in Corollary 1.2? Is "most" of the nontrivial homology of Γ^t_n in dimension $n(t-1) - \binom{t}{2} - 1$? In what dimensions does nontrivial homology occur? Is the homology torsion free? Is Γ^t_n homotopy equivalent to a wedge of spheres? The work on complexes of non-Hamiltonian graphs by Jonsson [Jo], along with our computer calculations with Γ^4_8, indicates that reasonable conjectures about graph complexes arising from computations with small vertex sets are often false. We finish by noting that we have not examined the character of the symmetric group S_n determined by its action on the homology of Γ_n^t, and it might be interesting to do so in the case $t = 2$.

Acknowledgments. We thank Anders Björner, Manoj Chari, and Michelle Wachs for encouragement and helpful comments. All our computer computations have been done using a C-program by Frank Heckenbach.

REFERENCES

[BBLSW] E. Babson, A. Björner, S. Linusson, J. Shareshian, and V. Welker, *Complexes of not i-connected graphs*, Topology, 38 (1999), pp. 271–299.

[Bj] A. Björner, *Topological methods*, in Handbook of Combinatorics, R. Graham, M. Grötschel, and L. Lovász, eds., North-Holland, Amsterdam, 1995, pp. 1819–1872.

[BLVZ] A. Björner, L. Lovász, S. T. Vrećica, and R. T. Živaljević, *Chessboard complexes and matching complexes*, J. London Math. Soc. (2), 49 (1994), pp. 25–39.

[Bo] S. Bouc, *Homologie de certains ensembles de 2-sous-groupes des groupes symétriques*, J. Algebra, 150 (1992), pp. 158–186.

[Ch1] M. K. Chari, *On discrete Morse functions and combinatorial decompositions*, Discrete Math., 217 (2000), pp. 101–113.

[Ch2] M. K. Chari, preprint, Louisiana State University, Baton Rouge, LA, 1999.

[CP] C. Colbourn and W. Pulleyblank, *Matroid Steiner problems, the Tutte polynomial and network reliability*, J. Combin. Theory Ser. B, 47 (1989), pp. 20–31.

[Fo1] R. Forman, *Morse theory for cell complexes*, Adv. Math., 134 (1998), pp. 90–145.

[Fo2] R. Forman, *Morse theory and evasiveness*, Combinatorica, 20 (2000), pp. 489–504.

[Jo] J. Jonsson, *On the Homology of Some Complexes of Graphs*, preprint, Stockholm University, Stockholm, Sweden, 1999.

[KSS] J. Kahn, M. Saks, and D. Sturtevant, *A topological approach to evasiveness*, Combinatorica, 4 (1984), pp. 279–306.

[KRW] D. Karagueuzian, V. Reiner, and M. Wachs, *Matching complexes, bounded degree complexes and weight spaces of GL_n-complexes*, J. Algebra, 239 (2001), pp. 77–92.

[RR] V. Reiner and J. Roberts, *Minimal resolutions and the homology of matching and chessboard complexes*, J. Algebraic Combin., 11 (2000), pp. 135–154.

[Sh1] J. Shareshian, *Discrete Morse theory for complexes of 2-connected graphs*, Topology, 40 (2001), pp. 681–701.

[Sh2] J. Shareshian, *Links in complexes of separable graphs*, J. Combin. Theory Ser. A, 88 (1999), pp. 54–65.

[St] R. P. Stanley, *Enumerative Combinatorics*, Vol. 2, Cambridge University Press, Cambridge, UK, 1999.

[Tu] V. Turchin, *Homology of complexes of biconnected graphs*, Russian Math. Surveys, 52 (1997), pp. 426–427.

[Va1] V. Vassiliev, *Complexes of connected graphs*, in The Gelfand Mathematical Seminar, 1990–1992, L. Corwin et al., eds., Birkhäuser, Boston, 1993, pp. 223–235.

[Va2] V. Vassiliev, *Complements of Discriminants of Smooth Maps: Topology and Applications*, revised ed., Transl. Math. Monogr. 98, AMS, Providence, RI, 1994.

[Va3] V. Vassiliev, *Topology of two-connected graphs and homology of spaces of knots*, in Differential and Symplectic Topology of Knots and Curves, Amer. Math. Soc. Transl. Ser. 2 190, S. Tabachnikov, ed., AMS, Providence, RI, 1999, pp. 253–286.

SIAM J. Discrete Math.
Vol. 16, No. 3, pp. 390–392

FULL-RANK TILINGS OF $\mathbb{F}_2^8$ DO NOT EXIST*

ARI TRACHTENBERG[†] AND ALEXANDER VARDY[‡]

Abstract. We show that there are no full-rank tilings of $\mathbb{F}_2^8$, using a carefully designed exhaustive search. This solves an open problem posed in [T. Etzion and A. Vardy, *SIAM J. Discrete Math.*, 11 (1998), pp. 205–233]. This also implies that a full-rank perfect binary code of length 15 with a kernel of dimension 7 does not exist.

Key words. full-rank tilings, perfect binary codes

AMS subject classifications. 05B45, 52C22, 94B99

PII. S0895480102410213

1. Introduction.

Let $\mathbb{F}_2^n$ be a vector space of dimension n over GF(2). A *tiling* of $\mathbb{F}_2^n$ is a pair (V, A) of subsets of $\mathbb{F}_2^n$, such that every $x \in \mathbb{F}_2^n$ has a unique representation of the form $x = v + a$, with $v \in V$ and $a \in A$. A tiling (V, A) of $\mathbb{F}_2^n$ is *trivial* if one of the sets V, A is $\mathbb{F}_2^n$ and the other is $\{\mathbf{0}\}$, where $\mathbf{0}$ denotes the all-zero vector in $\mathbb{F}_2^n$. It is of *full rank* if $\operatorname{rank}(V) = \operatorname{rank}(A) = n$ and $\mathbf{0} \in (V \cap A)$. The work of [3] shows that any tiling of $\mathbb{F}_2^n$ can be uniquely decomposed into (or constructed from) smaller tilings that are either trivial or have full rank. This reduces the classification of tilings of $\mathbb{F}_2^n$ to the study of full-rank tilings. Hence, the following question is of interest: For which values of n does $\mathbb{F}_2^n$ admit a full-rank tiling?

It is established in [3, 4] that full-rank tilings of $\mathbb{F}_2^n$ exist for $n = 14$ and $n \geq 112$ and do not exist for $n \leq 7$. Proposition 5.1 of [5] shows that if $\mathbb{F}_2^{n_0}$ admits a full-rank tiling, then so does $\mathbb{F}_2^n$ for all $n \geq n_0$. Thus full-rank tilings of $\mathbb{F}_2^n$ exist for all $n \geq 14$.

There is also an interesting connection between full-rank tilings and full-rank perfect codes. A binary code $\mathbb{C}$ of length n is a subset of $\mathbb{F}_2^n$. A code $\mathbb{C} \subset \mathbb{F}_2^n$ is *perfect* if for some $r \geq 1$, the Hamming spheres of radius r about the codewords of $\mathbb{C}$ partition $\mathbb{F}_2^n$. A code $\mathbb{C} \subseteq \mathbb{F}_2^n$ is *full rank* if $\mathbf{0} \in \mathbb{C}$ and $\operatorname{rank}(\mathbb{C}) = n$. It is known [4] that full-rank perfect codes exist if and only if $r = 1$ and $n = 2^m - 1$ for $m \geq 4$. The kernel of a code $\mathbb{C} \subseteq \mathbb{F}_2^n$, denoted $\ker \mathbb{C}$, is the set of all $x \in \mathbb{F}_2^n$ such that $x + \mathbb{C} = \mathbb{C}$. It is easy to see that $\ker \mathbb{C}$ is a linear subspace of $\mathbb{F}_2^n$. It is shown in [5] that there exists a full-rank perfect code of length $n = 2^m - 1$ with a kernel of dimension k if and only if there exists a full-rank tiling (V, A) of $\mathbb{F}_2^{n-k}$ with $|V| = 2^m$ and $\ker A = \{\mathbf{0}\}$. LeVan and Phelps [8] found by computer search full-rank perfect codes of length 15 with kernels of dimension 2, 3, 4, and 5. This implies [5] that full-rank tilings of $\mathbb{F}_2^n$ exist for all $n \geq 10$.

[*] Received by the editors June 27, 2002; accepted for publication (in revised form) October 29, 2002; published electronically April 30, 2003. This research was supported in part by the David and Lucile Packard Foundation and by the National Science Foundation. Part of this research was performed while both authors were at the University of Illinois.

http://www.siam.org/journals/sidma/16-3/41021.html

[†] Department of Electrical and Computer Engineering, Boston University, 8 St. Mary's Street, Boston, MA 02215 (trachten@bu.edu).

[‡] Department of Electrical and Computer Engineering, Department of Computer Science and Engineering, University of California, San Diego, 9500 Gilman Drive, MC 0407, La Jolla, CA 92093-0407 (vardy@kilimanjaro.ucsd.edu).

Thus the only unresolved cases where it is not known whether $\mathbb{F}_2^n$ admits a full-rank tiling are $n = 8$ and $n = 9$. We quote the following problem posed in [5, p. 220]:

> Construct full-rank tilings of $\mathbb{F}_2^n$ for $n = 8$ and $n = 9$, or prove that such tilings do not exist. This problem appears to be quite challenging despite the small size of the sets involved.

The main objective of this paper is to provide an answer to this problem for $n = 8$. We describe a carefully designed exhaustive search that proves the following theorem.

THEOREM 1. *A full-rank tiling of $\mathbb{F}_2^8$ does not exist.*

Theorem 1, along with Proposition 5.9 of [5], also implies that there is no full-rank perfect binary code of length 15 with a kernel of dimension 7. For more details on the rank and kernel-dimension of perfect binary codes, we refer the reader to [1, 2, 5, 9].

2. Nonexistence of full-rank tilings in eight dimensions.

Let (V, A) be a full-rank tiling of $\mathbb{F}_2^8$. Since every $x \in \mathbb{F}_2^8$ can be represented uniquely as $x = v + a$ with $v \in V$ and $a \in A$, we have $|V||A| = 2^8$. By definition, $\mathbf{0} \in V$ and $\mathbf{0} \in A$. Since $\operatorname{rank}(V) = \operatorname{rank}(A) = 8$, we must have $|V| \geq 9$ and $|A| \geq 9$, implying that $|V| = |A| = 16$.

LEMMA 2. *Let (V, A) be a full-rank tiling of $\mathbb{F}_2^n$, let M be an invertible $n \times n$ binary matrix, and let $\varphi_M(x) = xM$. Then $(\varphi_M(V), \varphi_M(A))$ is a full-rank tiling of $\mathbb{F}_2^n$.*

Proof. Since M is invertible, we have $\varphi_M(x) = \varphi_M(v) + \varphi_M(a)$ if and only if $x = v + a$. It is clear that the mapping φ_M is one-to-one and preserves the rank. □

Let $\{e_1, e_2, \ldots, e_8\}$ denote the set of vectors of weight one in $\mathbb{F}_2^8$. Using Lemma 2, we can transform a full-rank tiling (V, A) of $\mathbb{F}_2^8$ into a full-rank tiling $(\varphi_M(V), \varphi_M(A))$ with the property that $\{e_1, e_2, \ldots, e_8\} \subset \varphi_M(V)$. Thus we will henceforth assume without loss of generality that $\{e_1, e_2, \ldots, e_8\} \subset V$. Together with $\mathbf{0} \in V$, this determines 9 out of the 16 vectors of V.

LEMMA 3. *Let (V, A) be a full-rank tiling of $\mathbb{F}_2^8$. Then $d(a_1, a_2) \geq 3$ for any distinct vectors $a_1, a_2 \in A$, where $d(\cdot, \cdot)$ denotes the Hamming distance.*

Proof. Assume to the contrary that $\operatorname{wt}(a_1 + a_2) \leq 2$. Since $\{\mathbf{0}, e_1, \ldots, e_8\} \subset V$ by assumption, it follows that there exist distinct $v_1, v_2 \in V$ such that $v_1 + v_2 = a_1 + a_2$. But this implies that $a_1 + v_1 = a_2 + v_2$, which violates the unique representation property of a tiling. □

If (V, A) is a full-rank tiling of $\mathbb{F}_2^8$ and π is any permutation of the eight positions, then $(\pi V, \pi A)$ is also a full-rank tiling of $\mathbb{F}_2^8$. Since the set $\{\mathbf{0}, e_1, \ldots, e_8\}$ is preserved under all permutations $\pi \in S_8$, we have $\{\mathbf{0}, e_1, \ldots, e_8\} \subset \pi V$, and Lemma 3 holds with A replaced by πA. Hence, as potential candidates for A, it suffices to consider the nonisomorphic $(8, 16, 3)$ codes of full rank containing the vector $\mathbf{0}$.

To efficiently reject isomorphisms, we convert the set isomorphism problem to a graph isomorphism problem, as in [7]. Specifically, given a set $\mathcal{S} = \{a_1, \ldots, a_s\} \subset \mathbb{F}_2^8$, we define the bipartite graph $G(\mathcal{S})$ as follows: There are s left-hand vertices $\alpha_1, \ldots, \alpha_s$ and eight right-hand vertices $\beta_1, \ldots, \beta_8$, with (α_i, β_j) in the edge set of $G(\mathcal{S})$ if and only if the jth position of a_i is nonzero. Then two sets $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{F}_2^8$ are isomorphic if and only if the corresponding graphs $G(\mathcal{S}_1)$ and $G(\mathcal{S}_2)$ are isomorphic (cf. [7]). We check for graph isomorphism using the well-tested program NAUTY of [6]. Due to memory constraints, we have limited isomorphism rejection to a subset of A consisting of seven linearly independent vectors. Finally, we have also made use of the following lemma, which implies that any one vector in either A or V can be computed as the sum of the other vectors in this set.

LEMMA 4. *Let (V, A) be a full-rank tiling of $\mathbb{F}_2^8$, let $V = \{\mathbf{0}, v_1, v_2, \ldots, v_{15}\}$, and let $A = \{\mathbf{0}, a_1, a_2, \ldots, a_{15}\}$. Then $v_1 + v_2 + \cdots + v_{15} = a_1 + a_2 + \cdots + a_{15} = \mathbf{0}$.*

Proof. Let $H(V)$ be the 8×15 matrix having $v_1, v_2, \ldots, v_{15}$ as its columns, and consider the code $\mathbb{C} = \{\, x \in \mathbb{F}_2^{15} : H(V)x^t \in A \,\}$. It follows from [5, Theorem 5.3 and Propositions 5.4, 5.5] that $\mathbb{C}$ is a full-rank perfect code with a kernel of dimension $7+\dim(\ker A)$. It is furthermore shown in [3, Proposition 8.3] that $v_1+v_2+\cdots+v_{15} \in \ker A$. Thus if $v_1 + v_2 + \cdots + v_{15} \neq \mathbf{0}$, then $\ker \mathbb{C}$ has dimension at least 8. In view of Proposition 5.6 of [5] this, in turn, implies the existence of a full-rank tiling of $\mathbb{F}_2^n$ for $n \leq 7$. But it was established in [3, Corollary 7.3] that such a tiling does not exist. The fact that $a_1 + a_2 + \cdots + a_{15} = \mathbf{0}$ follows by symmetry. □

An exhaustive search based on the foregoing results did not produce a full-rank tiling of $\mathbb{F}_2^8$, thereby proving Theorem 1. The source code of our search program is available at http://people.bu.edu/trachten/. The search takes about a week on a contemporary PC workstation.

Acknowledgment. We are grateful to the MEDICIS Project at the École Polytechnique, Palaiseau, France, for the generous use of their computational resources.

REFERENCES

[1] S.V. AVGUSTINOVICH, O. HEDEN, AND F.I. SOLOV'EVA, *On Ranks and Kernels of Perfect Codes*, preprint, 2002.

[2] S.V. AVGUSTINOVICH, F.I. SOLOV'EVA, AND O. HEDEN, *On ranks and kernels problem of perfect codes*, in Proceedings of the Eighth International Workshop on Algebraic and Combinatorial Coding Theory (ACCT-VIII), St. Petersburg, Russia, 2002.

[3] G. COHEN, S. LITSYN, A. VARDY, AND G. ZÉMOR, *Tilings of binary spaces*, SIAM J. Discrete Math., 9 (1996), pp. 393–412.

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

[5] T. ETZION AND A. VARDY, *On perfect codes and tilings: Problems and solutions*, SIAM J. Discrete Math., 11 (1998), pp. 205–233.

[6] B.D. MCKAY, *Nauty User Guide*, Technical Report TR–CS–94–10, Computer Science Department, Australian National University, Canberra, Australia, 1994.

[7] P.R.J. ÖSTERGÅRD, T. BAICHEVA, AND E. KOLEV, *Optimal binary one-error-correcting codes of length* 10 *have* 72 *codewords*, IEEE Trans. Inform. Theory, 45 (1999), pp. 1229–1231.

[8] K.T. PHELPS, *Private communication*, Department of Discrete and Statistical Science, Auburn University, Auburn, AL, 1996.

[9] K.T. PHELPS AND M. LEVAN, *Kernels of nonlinear Hamming codes*, Des. Codes Cryptogr., 6 (1995), pp. 247–257.

SIAM J. Discrete Math.
Vol. 16, No. 3, pp. 393–417

TESTING OF CLUSTERING*

NOGA ALON[†], SEANNIE DAR[‡], MICHAL PARNAS[‡], AND DANA RON[§]

Abstract. A set X of points in $\Re^d$ is (k,b)-*clusterable* if X can be partitioned into k subsets (*clusters*) so that the diameter (alternatively, the radius) of each cluster is at most b. We present algorithms that, by sampling from a set X, distinguish between the case that X is (k,b)-clusterable and the case that X is ϵ-far from being (k,b')-clusterable for any given $0<\epsilon\le 1$ and for $b'\ge b$. By ϵ-far from being (k,b')-clusterable we mean that more than $\epsilon\cdot|X|$ points should be removed from X so that it becomes (k,b')-clusterable. We give algorithms for a variety of cost measures that use a sample of size independent of $|X|$ and polynomial in k and $1/\epsilon$.

Our algorithms can also be used to find *approximately good* clusterings. Namely, these are clusterings of all but an ϵ-fraction of the points in X that have optimal (or close to optimal) cost. The benefit of our algorithms is that they construct an *implicit representation* of such clusterings in time independent of $|X|$. That is, without actually having to partition all points in X, the implicit representation can be used to answer queries concerning the cluster to which any given point belongs.

Key words. property testing, clustering, randomized algorithms, approximation algorithms

AMS subject classifications. 68Q25, 68W20, 68W25, 68W40

PII. S0895480102410973

1. Introduction.

Clustering problems arise in many areas and have a variety of applications (cf. [7, 30, 41, 32, 15, 47, 34, 31]). There are many definitions of optimal clustering, and the choice of the appropriate definition depends on the particular application studied. Here we consider one of the standard forms, where the problem is to decide whether a given set X of n points in the d-dimensional Euclidean space can be partitioned into k subsets (*clusters*) so that the *cost* of each cluster is at most b. Two of the most well-studied cost measures are the *radius* cost and the *diameter* cost. In the first case, the cost of a cluster is defined as the minimum radius of a ball containing all the points in the cluster. In the latter case, the cost of a cluster is defined as the maximum distance between pairs of points in the cluster.[1] If such a k-way partition of X exists, then we say that X is (k,b)-*clusterable* (with respect to the radius or the diameter cost). Unfortunately, both decision problems are NP-complete for $d\ge 2$ (and variable k) [19, 36] and remain hard even when only a certain constant approximation of the cluster size is sought [17].

In this work we consider the following relaxation of the above decision problems: For a given approximation parameter $\beta\ge 0$ and distance parameter $0\le\epsilon\le 1$, we

*Received by the editors July 11, 2002; accepted for publication (in revised form) September 23, 2002; published electronically April 30, 2003. An extended abstract of this work appeared in *Proceedings of the 41st Annual Symposium on Foundations of Computer Science* (*FOCS* 2000), IEEE Computer Society, Los Alamitos, CA, 2000, pp. 240–249.

http://www.siam.org/journals/sidma/16-3/41097.html

[†]Department of Mathematics, Tel-Aviv University, Ramat Aviv, Israel (noga@math.tau.ac.il). This author's research was supported in part by a USA Israeli BSF grant, by a grant from the Israel Science Foundation, and by the Hermann Minkowski Minerva Center for Geometry at Tel-Aviv University.

[‡]Department of Computer Science, The Academic College of Tel-Aviv-Yaffo, 4 Antokolsky St., Tel-Aviv, Israel (michalp@mta.ac.il).

[§]Department of EE—Systems, Tel-Aviv University, Ramat Aviv, Israel (danar@eng.tau.ac.il). This author's research was supported by Israel Science Foundation grant 32/00-1.

[1] The first problem is also known as *center* clustering (since all points in a given cluster are at a distance of at most b from the center of the bounding ball) and the second is also known as *pairwise* clustering.

would like to determine whether the set X is (k, b)-clusterable or ϵ-*far* from being $(k,(1+\beta)b)$-clusterable. By ϵ-*far* from $(k,(1+\beta)b)$-clusterable we mean that more than an ϵ-fraction of the points in X should be removed (or moved) so that X becomes $(k,(1+\beta)b)$-clusterable. Given this relaxation of the decision problem, we seek algorithms that will be significantly faster than those required for solving the exact decision problems. In particular, we ask that our algorithms observe as few points as possible from X and run in time sublinear in $n = |X|$ or even independent of n.

We refer to algorithms that perform such relaxed (approximate) decision tasks as *testing* algorithms: they are required to output accept if X is (k, b)-clusterable, and to output reject with probability at least $2/3$, if X is ϵ-far from $(k,(1+\beta)b)$-clusterable. (If neither holds, the testing algorithm may output either accept or reject.) Such testing algorithms can be useful as an alternative to an exact or even approximate decision procedure when the number of points n is very large. Even if n is not too large and there is time to run a clustering algorithm on all the points, testing can be applied as a preliminary step to approximate the quality of the best achievable clustering.

1.1. Our results. We present and analyze testing algorithms both for the radius cost and for the diameter cost. All our algorithms run in time *independent* of $n = |X|$ and use a sample from X that has size polynomial in k and ϵ.

We describe algorithms for the L_2 metric (Euclidean distance) as defined above, which in the case of the radius cost easily extend to other metrics (such as L_∞). We also give algorithms that work under any general metric for $\beta = 1$. With the exception of our algorithms for general metrics, all our algorithms have the following form: They uniformly select a sample of points from X and run an exact decision procedure for verifying whether the sample is (k, b)-clusterable. Specifically, we show the following:

1. For general metrics we give algorithms that work for $\beta = 1$. For both costs, the sample selected is of size $O(k/\epsilon)$, and the running time is $O(k^2/\epsilon)$. We also observe that any algorithm for testing diameter clustering for $\beta < 1$ under a general metric requires a sample of size $\Omega(\sqrt{n/\epsilon})$.
2. For the L_2 metric and the radius cost, the algorithm works for $\beta = 0$ and the sample size is $\tilde{O}(d \cdot k/\epsilon)$. We also show how the analysis of this algorithm can be easily extended to clusters that do not correspond to d-dimensional balls, as is the case with radius clustering under the L_2 metric, but rather are determined by other "simple" geometric regions (that is, where the family of sets defined by these regions has a small Vapnik–Chervonenkis (VC)-dimension). An alternative analysis of the algorithm that allows for $\beta > 0$ uses a sample of size $\tilde{O}(\frac{k^2}{\epsilon \cdot \beta^2})$, which is independent of the dimension d.
3. For the L_2 metric and the diameter cost, the sample is of size $\tilde{O}(\frac{k^2}{\epsilon} \cdot (\frac{2}{\beta})^{2d})$. A dependence on $1/\beta$, as well as an exponential dependence on the dimension, are unavoidable. We prove a lower bound of $\Omega(\beta^{-(d-1)/4})$ on the size of the sample required for testing for $k = 1$ and a constant ϵ.

In items 2 and 3 we stated only the size of the sample selected by the algorithms. The running times depend on the exact decision procedures applied and, given the difficulty of the problems, are exponential in k and d. We also note that in many settings k and b are not predetermined, in which case one wants to find a setting of these parameters that is appropriate for the data. Using the above algorithms it is possible to search for such good settings.

Approximately good clusterings. In addition to the above, our algorithms can be used to obtain approximately good clusterings.

DEFINITION 1 (ϵ-good clustering). *A k-way partition P of X is an* ϵ-good (k, b')-*clustering of X if it is a partition having cost at most b' of all but at most an ϵ-fraction of the points in X.*

If X is (k, b)-clusterable, then using our testing algorithms it is possible to obtain in time independent of n an *implicit representation* of an ϵ-good $(k, (1+\beta)b)$-clustering of X. Namely, given this implicit representation we can determine for any given point $x \in X$ the cluster to which it belongs. This can be done in time $O(k)$ per point, or even $O(\log k)$, depending on the cost measure. For example, in the case of radius clustering, the implicit representation is simply a set of k cluster centers. The benefit of such an implicit representation is that it allows us to answer queries of the form "Do points $x, y \in X$ belong to the same cluster?" without actually having to partition all points. This approach was applied previously in [22] to graph partitioning problems, and a related approach was applied in [20].

Independently from our work, Mishra, Oblinger, and Pitt [37] study the problem of approximately good clustering when the cost measure is the sum of distances (or distances squared) to the cluster centers. Their algorithms use a sample of size independent of n and polynomial in $1/\epsilon$, d, k, and M, where the points belong to $[0, M]^d$.

Possible implications. We can draw several conclusions from the above results. First, suppose that it suffices for one's purposes to distinguish between a set that is (k, b)-clusterable and a set that is ϵ-far from $(k, 2b)$-clusterable (or, in the case of finding a clustering, it suffices to find an ϵ-good $(k, 2b)$-clustering). That is, a factor of 2 in the size of the radius/diameter of the clusters is of no great consequence. Then we have a very simple and efficient algorithm for the task both for the radius cost and for the diameter cost (as well as under different metrics). On the other hand, if we would like to go below a factor of 2 in the cost (i.e., to go from $\beta = 1$ to $\beta < 1$), then the two cost measures will exhibit a very different behavior. While for the radius cost we can easily achieve $\beta = 0$, with a sample having almost linear dependence on the dimension d, testing under the diameter cost can be done only with a sample of size $\Theta(\beta^{-\Theta(d)})$. One possible conclusion is that if there is no special (application-specific) reason to use the diameter cost, then the radius cost is preferable.

Techniques. The following approach is a common thread passing through the analysis of most of our testing algorithms. Recall that our algorithms work by sampling from X. The sample is viewed as being selected in phases, where we show that, with high probability, in each phase certain *progress* is made. In particular, in the case that X is ϵ-far from being $(k, (1+\beta)b)$-clusterable, this progress leads to rejection after a bounded number of phases. For example, in the case of the diameter cost and a single cluster ($k = 1$), progress is measured in terms of reducing the volume of the region in $\Re^d$ which contains all the points having distance of at most b from every sample point.

For the radius cost under the L_2 metric, our analysis uses ϵ-*nets* and their relation to the VC-dimension of families of sets. This relation was previously exploited both in the context of learning and in the context of computational geometry.

1.2. Perspective. In this paper we approach the problem of clustering from within the framework of property testing [45, 22]. In property testing the goal is to decide whether a given object (e.g., graph or function) has a predetermined property (e.g., connectivity or monotonicity) or is *far* from having the property. The notion of being far from having the property depends on the type of object considered. For example, if the object is a graph, then we say that it is far from having a particular property if many edge modifications should be made so that it obtains the property. By "many" we mean at least a certain ϵ-fraction of all the edges in the graph.

Previous work in property testing has mainly dealt with properties of functions [9, 45, 44, 16, 33, 21, 13] and properties of graphs [22, 24, 23, 4, 39, 5, 8]. More recently, property testing has been applied in other domains; cf. [6, 16, 12]. (For surveys see [43, 18].)

Here we further extend the scope of property testing to the domain of clustering problems. Our proof techniques combine geometric analysis with probabilistic analysis that is characteristic of work in property testing. We thus hope to enrich both areas of research.

Other related work. Hochbaum and Shmoys [29] were the first to show that it is hard to approximate the cost of an optimal clustering to within a factor of 2 for a general distance function. They also give a 2-approximation algorithm for the problem [28, 29]. As noted above, Feder and Greene [17] show that constant approximation is also hard for L_2 and L_∞ metrics (where the specific constants depend on the metric and cost measure used). An approximation factor of 2 can be achieved efficiently for all geometric variants we consider [25, 17]. For the radius cost, and under both L_∞ and L_2 metrics, Agrawal and Procopiuc [1] give an algorithm for finding a clustering having a radius of at most $(1 + \beta)$ times larger than the optimal radius in time $O(n \cdot \log k + (k/\beta)^{O(d^2 k^{1-1/d})})$. For more information on clustering, see [14, 2, 40] and references therein. In recent work [11], Czumaj and Sohler improve on our result for testing diameter clustering. They show that for any L_p norm the sufficient sample size for testing is $\tilde{O}(k(2/\beta)^{d-1}/\epsilon)$.

1.3. Organization. In section 2 we introduce notation and definitions used in the paper. In section 3 we discuss testing when the underlying distance function is a general metric. In section 4 we present and briefly discuss our generic testing algorithm, whose different variants will be presented in the subsequent sections. In section 5 we consider the basic 1-dimensional case for both radius and diameter testing. In sections 6 and 7, we present the variants of the generic algorithm for the radius cost and the diameter cost, respectively, when $d \geq 1$ and when the L_2 metric is used. In section 7 we also give our lower bound for the diameter cost. Finally, in section 8 we describe the alternative analysis for radius clustering that works for $\beta > 0$ and uses a sample whose size is independent of d.

2. Preliminaries. We denote by $\mathrm{dist}(x, y)$ the distance between two points x, y. We follow the standard practice of assuming that the distance between a pair of points can be computed in constant time. Since most of this paper deals with the L_2 metric, in what follows we refer to the Euclidean distance. Thus, if $x, y \in \Re^d$, that is, $x = (x_1, \ldots, x_d)$ and $y = (y_1, \ldots, y_d)$, then the Euclidean distance between x and y is $\mathrm{dist}(x, y) \overset{\mathrm{def}}{=} \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$.

Given a subset $S \subseteq X$, denote by $d(S)$ the *diameter* of S, that is, the maximum distance between any two points in S. Denote by $r(S)$ the *radius* of the smallest ball containing S, that is, $r(S) = \min_{y \in \Re^d} \max_{x \in S} \mathrm{dist}(x, y)$. The point $y \in \Re^d$ for which the minimum radius is achieved is the *center* of the *minimum bounding ball* of S.

Let $P = (X^1, \ldots, X^k)$ be a k-way partition of X. The *diameter* of the partition P is defined as $D(P) = \max_j d(X^j)$. The *radius* of P is defined as $R(P) = \max_j r(X^j)$.

For such a k-way partition P of X, we consider the following cost measures:

1. Diameter cost: $Cost(P) = D(P)$.
2. Radius cost: $Cost(P) = R(P)$.

Hence, a set X is (k, b)-clusterable according to one of the above cost measures if there *exists* a k-way partition $P = (X^1, \ldots, X^k)$ of X such that $Cost(P) \leq b$. The set X is *ϵ-far* from being $(k, (1 + \beta)b)$-clusterable for a given $0 \leq \epsilon \leq 1$ and $\beta \geq 0$ if for *every* subset $Y \subseteq X$ of size at most $(1 - \epsilon)|X|$, and for every k-way partition $P_Y = (Y^1, \ldots, Y^k)$ of Y, we have $Cost(P_Y) > (1 + \beta)b$.

Since all our algorithms have a one-sided error, we shall use the following definition of testing algorithms for clustering. We say that an algorithm is a *diameter-clustering* (*radius-clustering*) *tester* if, given access to points in a set $X \subset \Re^d$ and parameters k, b, ϵ, and β, the algorithm accepts X if it is (k, b)-clusterable with respect to the diameter cost (radius cost) and rejects X with probability of at least $2/3$ if it is ϵ-far from being $(k, (1 + \beta)b)$-clusterable.

3. Testing of clustering under general metrics.

We begin by describing a testing algorithm for diameter clustering when the underlying distance function is any metric and $\beta = 1$. That is, the algorithm distinguishes between the case in which X is (k, b)-clusterable and the case in which X is ϵ-far from $(k, 2b)$-clusterable under the assumption that the distances between points in X obey the triangle inequality. A slight variant of the algorithm works for radius clustering under the same conditions. The basic idea of the algorithm is to try and find points in X that are *representatives* of different clusters. That is, their pairwise distances are greater than the allowed diameter b. This algorithm is reminiscent of the factor 2 approximation algorithm of Gonzalez [25] for radius clustering. In the case that X is (k, b)-clusterable, there can be at most k such representatives. On the other hand, as we show in the analysis of the algorithm, if X is ϵ-far from $(k, 2b)$-clusterable, then with probability of at least $2/3$ the algorithm will find $k + 1$ such representatives. A certain refinement of this idea serves as a basis for the analysis of some of our other algorithms.

Algorithm 1 (general metric, diameter cost, $k \geq 1$, $\beta = 1$).

```
1. REPS ← $\mathrm{rep}_1$, where $\mathrm{rep}_1$ is an arbitrary point in $X$.
2. For $i = 2$ to $m = 6k/\epsilon$:
    (a) Uniformly and independently select a point $x \in X$.
    (b) If $\mathrm{dist}(x, \mathrm{rep}_j) > b$ for every $\mathrm{rep}_j \in$ REPS, then REPS ← REPS $\cup \{x\}$.
    (c) If $|\mathrm{REPS}| > k$, then halt and reject.
3. Accept.
```

Since the algorithm computes at most $k \cdot m$ distances, the running time of the algorithm is $O(k^2/\epsilon)$.

Theorem 1. *Algorithm 1 is a diameter-clustering tester for $\beta = 1$ under any metric.*

Proof. We first observe that the algorithm rejects only if it finds at least $k + 1$ points whose pairwise distances are all greater than b. If X is (k, b)-clusterable, then every two points that belong to the same cluster are at a distance of at most b from each other, and hence in this case the algorithm never rejects. From now on assume that X is ϵ-far from $(k, 2b)$-clusterable.

Consider any particular iteration $i \geq 2$ at the start of which $|\mathrm{REPS}| \leq k$. We say that a point $x \in X \setminus \mathrm{REPS}$ is a *candidate representative* with respect to REPS if it has distance greater than b from each of the points in REPS. We claim that if X is ϵ-far from $(k, 2b)$-clusterable, then there must be more than ϵn such candidate representatives. To verify this, assume by contradiction that there are at most ϵn such points, and let $\mathrm{REPS} = \{\mathrm{rep}_1, \ldots, \mathrm{rep}_\ell\}$, $\ell \leq k$. Then we could remove all candidate representatives from $X \setminus$ REPS, and for every other point $y \in X$, assign y to a cluster j such that $\text{dist}(y, \text{rep}_j) \leq b$. By the triangle inequality, every two points that are assigned to the same cluster are at a distance of at most $2b$, contradicting our assumption on X.

Therefore, in each iteration at the start of which $|\text{REPS}| \leq k$, there is a probability of at least ϵ that we obtain a candidate representative (which becomes a new representative and increases the size of REPS). By a multiplicative Chernoff bound, the probability that there are less than $(1/6) \cdot \epsilon m = k$ iterations, in which a new representative is added to REPS, is less than $\exp(-(1/2)(1-1/6)^2 \epsilon m) < 1/3$.[2] Hence, with probability of at least $2/3$, there exists an iteration in which $|\text{REPS}| > k$, and the algorithm rejects as required. □

Finding an approximately good clustering. Suppose that X is (k, b)-clusterable. Then Algorithm 1 always terminates with at most k representatives in REPS. Moreover, by a slight variant of the above analysis, with probability of at least $2/3$, the number of candidate representatives in $X \setminus$ REPS at the time of the termination is at most ϵn. To see why this is true, observe first that if the number of representatives in REPS is exactly k, then there are *no* candidate representatives. Next observe that the number of candidate representatives is monotonically nonincreasing. As long as their number is greater than ϵn, the probability of adding a new representative to REPS is at least ϵ. Hence, if the number of candidate representatives does not go below ϵn, then with probability of at least $2/3$ the final set REPS contains k representatives.

This implies that with probability of at least $2/3$, the final set $\text{REPS} = \{\text{rep}_1, \ldots, \text{rep}_\ell\}$, where $\ell \leq k$, has the following property: It defines an implicit representation of a partition having diameter of at most $2b$ of all but at most an ϵ-fraction of the points in X. That is, excluding the at most ϵn points in $X \setminus$ REPS that are candidate representatives (i.e., that are at a distance greater than b from the points in REPS), every other point $x \in X \setminus$ REPS can be assigned to some cluster j for which $\text{dist}(x, \text{rep}_j) \leq b$. We thus obtain $\ell \leq k$ clusters with diameter of at most $2b$. The time required to find the cluster to which a given point belongs is $O(k)$.

A lower bound for $\beta < 1$. If all that is known about the distance function between points in X is that it obeys the triangle inequality, then the above result is tight in the following sense. It is not possible to test for diameter clustering for $\beta < 1$ using a sample of size independent of n or even of size $o(\sqrt{n})$. To see why this is true consider a metric that is defined by a complete graph on $N = 2n$ vertices with the following weights (distances) on the edges. There exists a perfect matching between the vertices such that each edge in the matching has weight 2 and every other edge has weight 1. If X corresponds to any subset of size n of the vertices such that no two vertices in X are matched, then X is $(1, 1)$-clusterable. On the other hand, if X contains more than ϵn pairs of matched vertices, then it is ϵ-far from $(1, 2 - \delta)$-clusterable for any $\delta > 0$. However, in order to distinguish between the two cases with a nonnegligible probability, the algorithm has to sample $\Omega(\sqrt{n/\epsilon})$ vertices.

Testing radius clustering under general metrics. The algorithm for radius clustering is the same as Algorithm 1, except that a point is selected as a new representative only if it is at a distance greater than $2b$ from each representative selected so far. By the triangle inequality, if X is (k, b)-clusterable, then there can be at most k representatives. On the other hand, if X is ϵ-far from $(k, 2b)$-clusterable, then as long as

[2] The exact form of the Chernoff bound we are using is the following: Let $X_1, \ldots, X_m$ be m independent random variables where $X_i \in [0, 1]$ and the expected value of each X_i is p. Then for every $\gamma \in [0, 1]$ we have $\Pr\left[\sum X_i < (1 - \gamma)pm\right] < \exp(-(1/2)\gamma^2 pm)$.

$|\text{REPS}| \leq k$ there must be more than ϵn candidate representatives (as the representatives in REPS can serve as cluster centers). Hence the analysis of the radius-clustering algorithm follows along the same lines as that of the diameter-clustering algorithm. Furthermore, as in the case of diameter clustering, if X is (k, b)-clusterable, then we can use the representatives found by the algorithm to induce an ϵ-good $(k, 2b)$-clustering of X. In particular, the representatives in REPS serve as the centers of the clusters.

4. A generic testing algorithm for clustering. As noted in the introduction, with the exception of the algorithms for general metric presented in the previous section, all our testing algorithms have the same generic form.

ALGORITHM 2 (generic testing algorithm for clustering).

1. *Uniformly and independently select $m(k, \epsilon, d, \beta)$ points in X.*
2. *If there exists a k-way partition P of the selected sample points such that $Cost(P) \leq b$, then output* accept*, otherwise, output* reject.

The differences between the specific algorithms are

1. the range of parameters for which they work (e.g., $\beta = 0$ or $\beta > 0$);
2. the size of the sample $m(k, \epsilon, d, \beta)$;
3. the choice of the cost measure $Cost(\cdot)$ and the underlying distance metric, and the corresponding implementation of step 2 of the algorithm.

Note that in all cases, if X is (k, b)-clusterable, then the algorithm always accepts. Hence, in analyzing the different variants of the above algorithm we focus on proving that if X is ϵ-far from $(k, (1 + \beta)b)$-clusterable, then the algorithm rejects with probability of at least $2/3$.

Running times. Before we present and prove the different variants of this algorithm, we discuss the running time for both the radius cost and the diameter cost as a function of the sample size m. The running time is dominated, of course, by step 2 of the algorithm.

For the radius cost, step 2 requires finding k spheres with minimum radius that contain all m points in the sample (known as the *Euclidean k-center problem*). For $k = 1$, finding the minimum bounding sphere of a set of m points can be done by linear programming in time polynomial in m and d. In the case that d is constant, there are linear-(in m)-time algorithms for the problem [35, 3, 10]. For $k > 1$, finding k minimum bounding spheres can be done in time $O(m^{kd+2})$; cf. [2, sect. 7.1]. When d is relatively small it is possible to obtain an improvement of this running time by using the algorithm of Agrawal and Procopiuc [1], which has running time $m^{O(f(d) \cdot k^{1-1/d})}$, where $f(d)$ is always bounded by $O(d^{5/2})$.

As for the diameter cost, step 2 requires us to verify whether there exists a k-way partition having diameter at most b of a given set of m points. For $k = 1$ the time is clearly bounded by $O(m^2)$, since we only need to check whether all pairs of points in the sample are at a distance of at most b from each other. If $k = 1$ and the dimension d of the points is at most 3, then this can be done in time $O(m \log m)$ [42]. As for general values of k and d, this can be done in time $(O(m))^{d \cdot k^2}$ [46]. The basic observation is that we may consider only partitions for which the convex hulls of the different clusters are disjoint. This is true since, given a minimum diameter partition for which some point in cluster i belongs to the convex hull of cluster i', we can move this point from cluster i to cluster i' without increasing the diameter. Thus, in step 2 the algorithm enumerates all such partitions of the sample and computes their diameter. This is done by considering all choices of $\binom{k}{2}$ hyperplanes among the $O(m^{d+1})$ hyperplanes that separate the m sample points and then merging subsets of points that fall in the resulting regions into k clusters.

5. Clustering in one dimension. Before addressing the problem of testing clustering in d-dimensional Euclidean space for a general d, we address the simple special case of $d = 1$. In one dimension the radius and diameter problems are the same (under any L_p norm): every cluster corresponds to an interval, where in the case of radius clustering the interval is of length at most $2b$, and in the case of diameter clustering it is of length at most b. Determining whether a set of points in one dimension is contained in a union of at most k intervals of a given bounded length can be done by dynamic programming. As we show below, the sample size sufficient for Algorithm 2 is $\tilde{O}(k/\epsilon)$, and hence the running time of the algorithm in this case is poly(k/ϵ).

THEOREM 2. *Algorithm 2 with sample size $m = \Theta(\frac{k}{\epsilon} \cdot \log \frac{k}{\epsilon})$ and Cost $=$ R(Cost $= D$) is a testing algorithm for radius (diameter) clustering in one dimension for $k \geq 1$ and $\beta = 0$.*

The proof of Theorem 2 follows directly from the following lemma (which is stated for the radius cost) and by a standard "balls and bins" analysis.

LEMMA 1. *Let X be ϵ-far from being (k, b)-clusterable with respect to the radius cost. Then there exist k nonintersecting segments $[left_i, right_i]$, each of length $2b$, such that there are at least $(\epsilon n)/(k+1)$ points from X between every two segments as well as to the left of the leftmost segment and to the right of the rightmost segment.*

Proof. Let us assume for simplicity that X contains distinct points. The first (leftmost) segment is placed such that there are $(\epsilon n)/(k + 1)$ points from X to the left of it. Since X is ϵ-far from being (k, b)-clusterable, there must exist at least $(\epsilon nk)/(k + 1)$ points to the right of this first segment. We thus place the second segment to the right of the first segment so that there are $(\epsilon n)/(k+1)$ points from X between the two segments. The remaining segments are placed in a similar way. □

6. Testing of radius clustering under the L_2 metric. In this section we consider clustering of points in $\Re^d$, $d \geq 1$, when the cost measure is the radius cost and the underlying distance metric is the L_2 metric. Recall that for this cost and metric all points in each cluster must be contained in a ball of radius b. We provide sufficient conditions on the sample size so that the generic testing algorithm is a radius-clustering tester for $\beta = 0$. As we show below, this analysis can be easily generalized to testing clustering when the clusters correspond to other "simple" geometric regions (that is, where the family of sets defined by these regions has a small VC-dimension). For example, this is true for clusters that are contained in axis aligned cubes with bounded side lengths and for clusters that are contained in ellipsoids of a bounded size. (Note that the former are obtained when the cost measure is the radius cost and the underlying metric is L_∞.) As we see below, the size of the sample is almost linear in d. An alternative analysis of the algorithm, which works for $\beta > 0$ and uses a sample of size independent of d, is given in section 8.

THEOREM 3. *Algorithm 2 with sample size $m = \Theta(\frac{d \cdot k}{\epsilon} \cdot \log(\frac{d \cdot k}{\epsilon}))$ and Cost $= R$ is a radius-clustering tester for $k \geq 1$, $d \geq 1$, and $\beta = 0$.*

Remark. Using a result of [22] concerning the relation between learning algorithms and testing algorithms, we could obtain a testing algorithm for radius clustering with the same complexity as stated in Theorem 3 but with a two-sided error. This would be based on the learnability of the concept class defined by unions of k balls. Here we give a direct analysis and obtain a one-sided error. We note that the same idea applied here can be used to obtain testing algorithms having a one-sided error for any property that can be defined by a family of subsets having a bounded VC-dimension.

In order to prove Theorem 3 we shall need the following definitions (which for the sake of presentation are not given in their full generality). Let $\mathcal{S}$ be a family of subsets of $\Re^d$, let R be a finite subset of $\Re^d$, and let $0 < \epsilon < 1$. We say that $N \subset R$ is an *ϵ-net* of R with respect to $\mathcal{S}$ if, for every $S \in \mathcal{S}$ such that $|S \cap R| > \epsilon \cdot |R|$, there exists at least one point $x \in S \cap N$. In other words, N is an ϵ-net if it "hits" every subset in $\mathcal{S}$ that has a relatively large intersection with R. Our interest in ϵ-nets will soon become clear, but first we need one more definition.

We say that a subset $A \subset \Re^d$ is *shattered* by a family of subsets $\mathcal{S}$ if, for every $A' \subseteq A$, there exists a set $S \in \mathcal{S}$ such that $A' = A \cap S$. The VC-*dimension* of $\mathcal{S}$, denoted by $\mathrm{VCD}(\mathcal{S})$, is the maximum size of a subset $A \subset \Re^d$ that is shattered by $\mathcal{S}$. The VC-dimension of a family of subsets is hence a certain measure of richness (or diversity) of the family.

The following theorem is a special case of a theorem that was proved by Haussler and Welzl [27] based on the work of Vapnik and Chervonenkis [48].

THEOREM 4 (see [27]). *Let $\mathcal{S}$ be any family of subsets of $\Re^d$, let R be any finite subset of $\Re^d$, and let $0 < \epsilon < 1$. Consider a sample U of size $m \geq \frac{8\mathrm{VCD}(\mathcal{S})}{\epsilon} \cdot \log \frac{8\mathrm{VCD}(\mathcal{S})}{\epsilon}$ selected uniformly and independently from R. Then with probability of at least $2/3$, U is an ϵ-net for R with respect to $\mathcal{S}$.*

The proof of Theorem 4 actually gives a bound on the sample size m in terms of a slightly different measure from $\mathrm{VCD}(\mathcal{S})$, which we refer to as the *shatter exponent* (where $\mathrm{VCD}(\mathcal{S})$ is an upper bound on this measure). In our case we can get a slightly better bound on m if we use the shatter exponent directly. We next define it and state a corresponding variant of Theorem 4.

For a subset $A \subset \Re^d$, let $\Phi_{\mathcal{S}}(A) \overset{\mathrm{def}}{=} \{A \cap S : S \in \mathcal{S}\}$ be the *projection* of $\mathcal{S}$ on A. For any integer m, let $\phi_{\mathcal{S}}(m) = \max_{A,\,|A|=m} |\Phi_{\mathcal{S}}(A)|$ be the maximum size of the projection of $\mathcal{S}$ on a set of size m. In particular, by the definition of the VC-dimension, for every $m \leq \mathrm{VCD}(\mathcal{S})$, $\phi_{\mathcal{S}}(m) = 2^m$, while for $m > \mathrm{VCD}(\mathcal{S})$, $\phi_{\mathcal{S}}(m) < 2^m$. Let the *shatter exponent*, denoted $\mathrm{SE}(\mathcal{S})$, be the smallest integer such that for every $m \geq 2$, $\phi_{\mathcal{S}}(m) \leq c \cdot m^{\mathrm{SE}(\mathcal{S})}$ for some fixed constant c. It can be shown that for every family of subsets $\mathcal{S}$, $\mathrm{SE}(\mathcal{S}) \leq \mathrm{VCD}(\mathcal{S})$, but as noted above, we can sometimes get a better bound on $\mathrm{SE}(\mathcal{S})$.

THEOREM 4$'$. *Let $\mathcal{S}$ be any family of subsets of $\Re^d$, let R be any finite subset of $\Re^d$, and let $0 < \epsilon < 1$. Consider a sample U of size $m \geq \frac{8\mathrm{SE}(\mathcal{S})}{\epsilon} \cdot \log \frac{8\mathrm{SE}(\mathcal{S})}{\epsilon}$ selected uniformly and independently from R. Then with probability of at least $2/3$, U is an ϵ-net for R with respect to $\mathcal{S}$.*

Proof of Theorem 3. If X is (k, b)-clusterable, then Algorithm 2 clearly always accepts. Hence, assume from now on that X is ϵ-far from being (k, b)-clusterable. We shall show that the algorithm rejects with probability of at least $2/3$.

Let $\mathcal{B}_{k,b}$ be the family of subsets of $\Re^d$ that are defined by unions of k balls each of radius at most b, and let $\overline{\mathcal{B}}_{k,b}$ be the family of complements of subsets in $\mathcal{B}_{k,b}$. By our assumption on X, we have that for every collection of k balls each having a radius of at most b, there are more than ϵn points in X that do not belong to any of the balls. In other words, for every $S \in \overline{\mathcal{B}}_{k,b}$, we have that $|S \cap X| > \epsilon|X|$. This implies that a subset $N \subset X$ is an ϵ-net for X with respect to $\overline{\mathcal{B}}_{k,b}$ if and only if it contains at least one point from every $S \in \overline{\mathcal{B}}_{k,b}$.

Now assume that the sample U selected by Algorithm 2 is an ϵ-net for X. Then, by the definition of ϵ-nets and our assumption on X, there is *no* k-way partition P of U such that $R(P) \leq b$ (and so the algorithm will reject). This is true since such a partition corresponds to k balls having radius b that contain all points in the sample.

However, this would contradict the assumption that U contains at least one point from every $S \in \overline{\mathcal{B}}_{k,b}$.

In order to bound the size of a sample that is sufficient to ensure that it constitutes an ϵ-net for X with respect to $\overline{\mathcal{B}}_{k,b}$, we bound $\mathrm{SE}(\overline{\mathcal{B}}_{k,b})$. It is easy to verify that $\mathrm{SE}(\overline{\mathcal{B}}_{k,b}) = \mathrm{SE}(\mathcal{B}_{k,b})$, and so it remains to bound $\mathrm{SE}(\mathcal{B}_{k,b})$. Given any set A of m points in $\Re^d$, the number of different subsets $A' = A \cap B$, where $B \in \mathcal{B}_{1,b}$ (i.e., sets defined by single balls), is at most m^{d+1}.[3] This follows from the fact that for each subset A' such that there exists balls $B \in \mathcal{B}_{1,b}$ for which $A' = A \cap B$, let $B_{A'}$ be such a ball having minimum radius. It is well known that for any such bounding ball there exists a subset $A'' \subseteq A'$ having size of at most $d+1$ such that $B_{A'} = B_{A''}$. Hence the number of balls enclosing different subsets of A is at most $\binom{m}{d+1} < m^{d+1}$. Since $\mathcal{B}_{k,b}$ includes unions of k balls, we have that $\mathrm{SE}(\mathcal{B}_{k,b}) \le k(d+1)$. Hence, Theorem 3 follows by applying Theorem 4′. □

Testing of clustering for clusters having shapes other than balls. Let $\mathcal{S}$ be a family of subsets of $\Re^d$ that are defined by containment in certain geometric regions (such as d-dimensional cubes or ellipsoids). Let size$(\cdot)$ be a fixed size measure of these regions (such as side lengths in the case of cubes). Let $\mathcal{S}_{k,b}$ be the family of subsets of $\Re^d$ that are defined by unions of k sets (regions) in $\mathcal{S}$, each having a size of at most b. Consider the instantiation of Algorithm 2, where for a given partition $P = (Y^1, \ldots, Y^k)$ of the sample we define

$$Cost(P) = \max_{Y^j \in P} \; \min_{S \in \mathcal{S},\, Y^j \subset S} \mathrm{size}(S).$$

Then the above analysis implies that this instantiation of Algorithm 2 with a sample of size $O(\mathrm{SE}(\mathcal{S}_{k,b}) \log \mathrm{SE}(\mathcal{S}_{k,b}))$ is a testing algorithm for clusters bounded by sets (regions) in $\mathcal{S}$. Depending on the particular choice of $\mathcal{S}$, one can obtain a bound on $\mathrm{SE}(\mathcal{S}_{k,b})$ either directly (as done in the proof of Theorem 3) or through the VC-dimension of $\mathcal{S}_{k,b}$, which by known results is at most k times the VC-dimension of $\mathcal{S}$.

Finding an approximately good clustering. Suppose that X is (k,b)-clusterable. In such a case Algorithm 2 finds a k-way partition P of the sample such that $R(P) \le b$. That is, the algorithm finds k centers $z_1, \ldots, z_k$ of balls of radius b that contain all sample points. An argument similar to the proof of Theorem 3 shows that with probability of at least $2/3$ the centers found by the algorithm actually define an ϵ-good (k,b)-clustering of X. Specifically, as shown in the proof of Theorem 3, with probability of at least $2/3$ the sample selected by the algorithm is an ϵ-net for X with respect to $\overline{\mathcal{B}}_{k,b}$. That is, for every $S \in \overline{\mathcal{B}}_{k,b}$ such that $|X \cap S| > \epsilon|X|$, the sample contains at least one point in S. (Note that here it is not true that for every S, $|X \cap S| > \epsilon|X|$, since X is assumed to be (k,b)-clusterable. However, this is immaterial to the claim.) Assume that the sample is in fact an ϵ-net for X. Then by the definition of $\overline{\mathcal{B}}_{k,b}$, this means that for every k balls of radius b such that more than $\epsilon|X|$ points of X fall *outside* these balls, the sample contains such a point outside the balls. This in turn implies that for the k balls defined by the centers $z_1, \ldots, z_k$ found by the algorithm, there are at most $\epsilon|X|$ points in X that do not belong to these balls. Thus the k centers induce an ϵ-good (k,b)-clustering of X.

Testing and the VC-dimension. The above analysis can be extended to obtain the following relation between the VC-dimension and testing similarly to how such a relation is obtained between the VC-dimension and PAC (Probably Approximately Correct) learning.

[3] In fact, the bound b on the radius of the balls can be used to obtain a bound of m^d. However, the reasoning is slightly more complicated.

Consider any property P of Boolean functions over some domain Z, and let $\mathcal{F}_P$ be the class of functions having property P. A testing algorithm for property P is given query access to the tested function f (and in particular may ask for the value of f on a uniformly selected sample). If f has property P (that is, f belongs to $\mathcal{F}_P$), then the algorithm should accept. If f is ϵ-far from having property P (that is, for every function $g \in \mathcal{F}_P$, $\Pr[g(z) \neq f(z)] > \epsilon$, where the probability is over a uniformly selected z), the algorithm should reject with probability of at least 2/3.

In what follows we shall sometimes view Boolean functions as sets. In particular, the VC-dimension of $\mathcal{F}_P$ is defined as the VC-dimension of the family of subsets: $\{S_f\}_{f\in\mathcal{F}_P}$ where $S_f \stackrel{\text{def}}{=} \{z : f(z) = 1\}$. Suppose that there is an algorithm $\mathcal{A}$ that, given a sample of labeled examples $\{z_i, b_i\}$ where $z_i \in Z$ and $b_i \in \{0, 1\}$, determines whether there exists a function in $\mathcal{F}_P$ that is consistent with the sample. That is, if $\exists g \in \mathcal{F}_P$, such that $g(z_i) = b_i$ for every i, then $\mathcal{A}$ outputs accept, and otherwise it outputs reject. We shall refer to $\mathcal{A}$ as a *consistency checker* for $\mathcal{F}_P$.

THEOREM 5. *For any property P, a consistency checker for $\mathcal{F}_P$ can be used for testing P by applying it to a uniformly selected sample of size $m \geq \frac{8\mathrm{VCD}(\mathcal{F}_P)}{\epsilon} \cdot \log \frac{8\mathrm{VCD}(\mathcal{F}_P)}{\epsilon}$.*

Proof. The proof of Theorem 5 is a generalization of the proof of Theorem 3. By the definition of a consistency checker, if $f \in \mathcal{F}_P$, then it accepts. Let $\overline{\mathcal{F}_P} \stackrel{\text{def}}{=} \{\neg g : g \in \mathcal{F}_P\}$ (so that in particular $\mathrm{VCD}(\overline{\mathcal{F}_P}) = \mathrm{VCD}(\mathcal{F}_P)$). Then by Theorem 4, for any function f with probability of at least 2/3, a sample of size m as stated in the theorem is an ϵ-net for f (i.e., S_f) with respect to $\overline{\mathcal{F}_P}$ (i.e., $\{S_g\}_{g\in\overline{\mathcal{F}_P}}$). As argued in the proof of Theorem 3, this implies that if f is ϵ-far from having property P, then it is rejected with probability of at least 2/3. $\square$

We note that in many cases (e.g., the property of monotonicity), the VC-dimension of the class of functions defined by the property is prohibitively large, and we seek other techniques (that in particular may use adaptive querying).

7. Testing of diameter clustering under the L_2 metric.

7.1. The case $k = 1$.

We start by studying the problem of testing for a single cluster. In the next subsection we extend the analysis to any number of clusters k. In all that follows the underlying distance metric is the L_2 metric.

THEOREM 6. *Algorithm 2 with sample size $m = \Theta(\frac{1}{\epsilon} \cdot d^{3/2} \cdot \log(\frac{1}{\beta})(\frac{2}{\beta})^d)$ and Cost $= D$ is a diameter-clustering tester for $k = 1$, $d \geq 1$, and $0 < \beta \leq 1$.*

We start by proving the theorem for two dimensions and then show how it generalizes to any number of d dimensions.

Proof of Theorem 6 *for* $d = 2$. Clearly if X is $(1, b)$-clusterable, then the algorithm accepts. We thus focus on proving that if X is ϵ-far from being $(1, (1+\beta)b)$-clusterable, then the algorithm rejects with probability of at least 2/3.

We shall view the sample as being selected in $p = 2\pi/\beta^2$ phases, where in each phase $\Theta(\log(p)/\epsilon)$ points are selected uniformly and independently. We shall show that, with high probability over the choice of the sample, in each phase certain *progress* is made. The progress is such that, after at most p phases, the diameter of all sample points is greater than b (causing the algorithm to reject).[4] For each $1 \leq j \leq p$, let

[4] We note that by slightly modifying the analysis to require progress only in a sufficient fraction of the phases, we could save a factor of $\log \frac{1}{\beta}$ in the sample size. However, the current analysis is slightly simpler, and in the general case of $k \geq 1$ (which is based on the current analysis), the $\log \frac{1}{\beta}$ factor becomes negligible.

U_j denote the union of all points selected in the first j phases. We shall need the following definitions.

DEFINITION 2.

- *For $x \in \Re^2$, let C_x denote the disk of radius b centered at x.*
- *For $T \subseteq \Re^2$, let $I(T)$ denote the* intersection *of all disks C_x of points $x \in T$.*
- *For any region R in $\Re^2$, let $A(R)$ denote the size (area) of R.*

By the above definition, for each phase j, every point $y \in I(U_j)$ is at a distance of at most b from *every* point in the sample selected so far. If in phase $j+1$ a new sample point x falls outside $I(U_j)$, then the algorithm rejects, as this means that the new point is at a distance greater than b from some sample point. Otherwise, $x \in I(U_j)$, and we consider the decrease in the area of the intersection caused by the addition of x. That is, $A(I(U_j)) - A(I(U_j \cup \{x\})) = A(I(U_j) \setminus C_x)$.

DEFINITION 3. *We say that a point $x \in X$ is* influential *with respect to $I(U_j)$ if $x \notin I(U_j)$ or if x causes a significant decrease in the area of $I(U_j)$, namely, if the area $A(I(U_j) \setminus C_x)$ that is removed from the intersection is greater than $(\beta b)^2/2$.*

We claim that if X is ϵ-far from being $(1,(1+\beta)b)$-clusterable, then for every $1 \le j \le p-1$, in phase $j+1$ there are at least ϵn points in X that are influential with respect to $I(U_j)$. Subject to this claim, if the sample in each phase is of size of at least $\ln(3p)/\epsilon$, then the probability that an influential point is not selected in a *fixed* phase is at most

$$(1-\epsilon)^{\ln(3p)/\epsilon} < \exp(-\ln(3p)) = 1/(3p).$$

Hence, the probability that for *some* phase no influential point is selected is less than 1/3.

Thus, assume from now on that for every $1 \le j \le p-1$, the sample selected in phase $j+1$ contains an influential point x with respect to $I(U_j)$. As stated above, if $x \notin I(U_j)$, then Algorithm 2 rejects. Otherwise, x decreases the area of $I(U_j)$ by at least $(\beta b)^2/2$. However, since the area of the initial disk (defined by the first sample point) is πb^2, then the number of phases in which such a decrease can occur is at most $p = 2\pi/\beta^2$.

In order to complete the proof of Theorem 6 for $d = 2$, we must show that for every $1 \le j \le p-1$, there are at least ϵn points in X that are influential with respect to $I(U_j)$. Assume, contrary to the claim, that there are at most ϵn influential points with respect to some $I(U_j)$. Then we can remove these (at most) ϵn influential points from X. The points that remain in X all belong to $I(U_j)$ and, as the following lemma shows, they form a cluster of diameter at most $(1+\beta)b$, in contradiction to our assumption on X.

LEMMA 2. *Let T be any finite subset of $\Re^2$. Then for every $x, y \in I(T)$ such that x is noninfluential with respect to T,* $\mathrm{dist}(x,y) \le (1+\beta)b$.

In order to prove Lemma 2, we shall need the following geometrical claim. For an illustration see Figure 7.1.

CLAIM 3. *Let C be a circle of radius at most b. Let s and t be any two points on C, and let o be a point on the segment connecting s and t such that* $\mathrm{dist}(s,o) \ge b$. *Consider the line perpendicular to the line through s and t at o, and let w be its (closer) meeting point with the circle C. Then* $\mathrm{dist}(w,o) \ge \mathrm{dist}(o,t)/2$.

Proof. Denote $\ell = \mathrm{dist}(o,t)$ and $\ell' = \mathrm{dist}(w,o)$. We place the center of the circle C at the origin $(0,0)$ and set the y axis to be parallel to the line connecting s and t. Let $r \le b$ be the radius of C. If we denote the y coordinate of t by η, then its x coordinate is $\alpha = \sqrt{r^2 - \eta^2}$. Given the distances between the points (and the

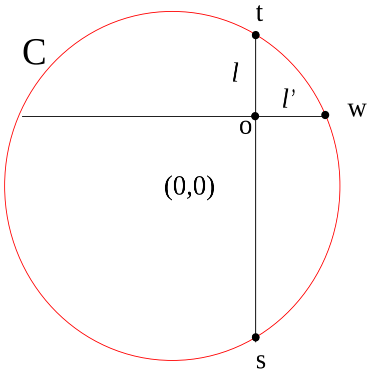


FIG. 7.1. *An illustration for the proof of Claim* 3.

orientation of the coordinate system), the point o is at coordinates $(\alpha, \eta - \ell)$, and the point w is at $(\alpha + \ell', \eta - \ell)$. Since w is on the circle, we must have that

$$(\alpha + \ell')^2 + (\eta - \ell)^2 = r^2, \tag{7.1}$$

which implies that

$$\ell' = \sqrt{r^2 - (\eta - \ell)^2} - \alpha. \tag{7.2}$$

If we now substitute $\alpha = \sqrt{r^2 - \eta^2}$, we get

$$\begin{aligned}\ell' &= \sqrt{r^2 - (\eta - \ell)^2} - \sqrt{r^2 - \eta^2}\\ &= \frac{(r^2 - (\eta - \ell)^2) - (r^2 - \eta^2)}{\sqrt{r^2 - (\eta - \ell)^2} + \sqrt{r^2 - \eta^2}}\\ &= \frac{\eta^2 - (\eta - \ell)^2}{\sqrt{r^2 - (\eta - \ell)^2} + \sqrt{r^2 - \eta^2}}\\ &= \frac{\ell(2\eta - \ell)}{\sqrt{r^2 - (\eta - \ell)^2} + \sqrt{r^2 - \eta^2}}\\ &> \frac{\ell(2\eta - \ell)}{2r}.\end{aligned}$$

Since both s and t are on the circle and $\text{dist}(s,t) = \text{dist}(s,o) + \text{dist}(o,t) \geq b + \ell$, we have that $2\eta \geq b + \ell$. Hence, since $r \leq b$, we obtain that $\ell' \geq \ell/2$ as claimed. □

Proof of Lemma 2. It is clear of course that if $y \in C_x$, then $\text{dist}(x,y) \leq b$. Therefore, let $y \in I_j \setminus C_x$. Consider the line through x and y, and let o be the point where it intersects with C_x. Then,

$$\text{dist}(x,y) = \text{dist}(x,o) + \text{dist}(o,y).$$

Clearly $\text{dist}(x,o) = b$. Thus, we want to show that $\text{dist}(o,y) \leq \beta b$.

Let us draw the tangent to C_x at o and let z and w be the first two points it meets on the boundary of I_j. The points y, w, z define a triangle T, whose height is $h = \text{dist}(o,y)$. Let $\ell_1 = \text{dist}(w,o)$ and $\ell_2 = \text{dist}(z,o)$. Thus, the length of the base of the triangle S is $\ell_1 + \ell_2$. Let $A(T)$ denote the area of T. Since $T \subseteq I_j \setminus C_x$, and x is noninfluential, we have

$$A(T) = \frac{h(\ell_1 + \ell_2)}{2} \leq \frac{(\beta b)^2}{2}. \tag{7.3}$$

We will now show that $h \leq \ell_1 + \ell_2$ and from this conclude that $h \leq \beta b$ as required.

We prove that $\ell_1 \geq h/2$. Let C_1 be the circle on which w sits, and let s and t be the intersection points of the line connecting x and y with the circle C_1 (see Figure 7.2).

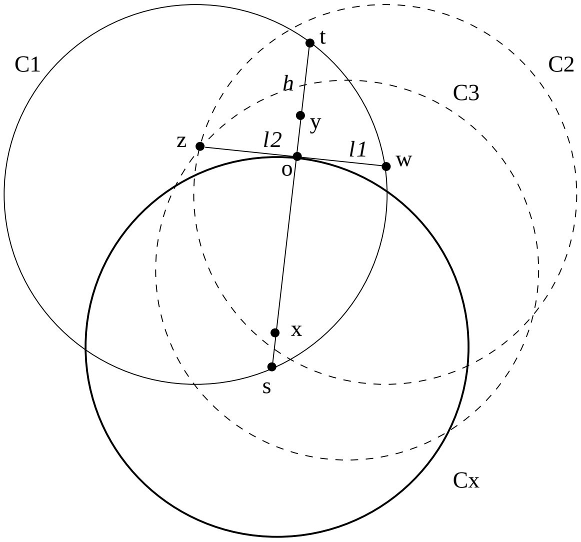


FIG. 7.2. *An illustration for the proof of Lemma 2. The circles C_x and C_1 are as defined in the proof. The circles C_2 and C_3 denote additional circles defined by points in the sample.*

We have that $\text{dist}(o, t) \geq h$ and $\text{dist}(s, o) \geq b$. We can thus apply Claim 3 and get that

$$\ell_1 = \text{dist}(w, o) \geq \text{dist}(o, t)/2 \geq h/2.$$

In an analogous way we can show that $\ell_2 \geq h/2$. This implies that $h \leq 2{\cdot}\min(\ell_1, \ell_2) \leq \ell_1 + \ell_2$. By (7.3) we can conclude that

$$\frac{h^2}{2} \leq \frac{h(\ell_1 + \ell_2)}{2} \leq \frac{(\beta b)^2}{2}. \qquad \square$$

Extending the proof to higher dimensions. For each sample point x let B_x denote the d-dimensional ball of radius b centered at x. Let $I(U_j)$ be the intersection of all balls centered at points selected in phases 1 through j, and let $V(I(U_j))$ denote the volume of the intersection. Let V_d denote the volume of the d-dimensional unit ball. Here we shall say that a point x is *influential* with respect to $I(U_j)$ if $x \notin I(U_j)$, or if the volume removed by x is at least

$$\begin{aligned} V(I(U_j)) - V(I(U_j \cup \{x\})) &= V(I(U_j) \setminus B_x) \\ &> \frac{(\beta b)^d \cdot V_{d-1}}{d \cdot 2^{d-1}}. \end{aligned}$$

Since the volume of the initial ball of radius b (defined by the first sample point) is $V_d \cdot b^d$, then the number p of phases required is at most

$$\frac{d \cdot V_d}{2 \cdot V_{d-1}} \cdot \left(\frac{2}{\beta}\right)^d = O\left(\sqrt{d}\left(\frac{2}{\beta}\right)^d\right).$$

Once again, the following lemma completes the proof of Theorem 6 for any $d > 2$.

Lemma 4. *Let T be any finite subset of $\Re^d$. Then for every $x, y \in I(T)$ such that x is noninfluential with respect to T,* $\mathrm{dist}(x, y) \leq (1 + \beta)b$.

Proof. Let $y \in I_j \setminus B_x$. Consider the line through x and y, and let o be the point where it intersects with B_x. Then,

$$\mathrm{dist}(x, y) = \mathrm{dist}(x, o) + \mathrm{dist}(o, y),$$

where $\mathrm{dist}(x, o) = b$. Again we show that $h = \mathrm{dist}(o, y) \leq \beta b$.

Consider some plane that passes through the line defined by x and y. Draw in this plane the line tangent to B_x at o. Let z and w be the first two points that this line meets on the boundary of I_j. Notice that any such plane intersects each of the d-dimensional balls defining I_j in a disk of radius at most b. Thus, we can again use Claim 3 and prove (as in Lemma 2) that $\mathrm{dist}(z, o) \geq h/2$ and $\mathrm{dist}(w, o) \geq h/2$. This will be true for any plane passing through the line defined by x and y.

Therefore, a $(d-1)$-dimensional ball of radius $h/2$ is contained in the intersection of $I_j \setminus B_x$ with the $(d-1)$-dimensional hyperplane tangent to B_x at o. Thus, the cone of height h, whose base is this $(d-1)$-dimensional ball of radius $h/2$, is contained in $I_j \setminus B_x$. The volume of this cone is

$$\frac{h(h/2)^{d-1}V_{d-1}}{d}$$

and, since x is noninfluential, we have that

$$\frac{h(h/2)^{d-1}V_{d-1}}{d} \leq \frac{(\beta b)^d \cdot V_{d-1}}{d \cdot 2^{d-1}}.$$

Thus, $h \leq \beta b$ as required. □

7.2. General k. We obtain the following theorem for $k \geq 1$.

Theorem 7. *Algorithm 2 with sample size $m = \Theta(\frac{k^2 \log k}{\epsilon} \cdot d \cdot (\frac{2}{\beta})^{2d})$ and Cost $= D$ is a diameter-clustering tester for $k \geq 1$, $d \geq 1$, and $0 < \beta \leq 1$.*

We start by extending the notion of influential points.

Definition 4. *Let $P_S = (S^1, \ldots, S^k)$ be a partition of a subset $S \subset X$. We say that a point x is* influential *with respect to P_S if either $x \notin \bigcup_{i=1}^k I(S^i)$ (that is, x is at a distance greater than b from some point in every S^i) or for every S^i,*

$$V(I(S^i) \setminus B_x) > \frac{(\beta b)^d \cdot V_{d-1}}{d \cdot 2^{d-1}}$$

(that is, the volume of $I(S^i)$ is reduced significantly by x for every S^i). Let $Y(P_S) \subset X$ denote the set of all points that are influential with respect to P_S.

Claim 5. *Suppose that X is ϵ-far from $(k, (1 + \beta)b)$-clusterable, and let $P_S = (S^1, \ldots, S^k)$ be a partition of some $S \subset X$. Then for any given $0 < \delta < 1$, with probability of at least $1 - \delta$, a uniformly and independently selected sample of size $s \geq \frac{\ln(1/\delta)}{\epsilon}$ contains at least one point $y \in Y(P_S)$.*

Proof. By Lemma 4, if X is ϵ-far from $(k, (1 + \beta)b)$-clusterable, then necessarily $|Y(P_S)| > \epsilon n$. Otherwise, we could remove all influential points and assign each of the other points $x \in X$ to a cluster i such that x is noninfluential with respect to S^i. This would result in a k-way partition of all but at most an ϵ-fraction of the points in X such that each cluster has diameter of at most $(1 + \beta)b$.

Therefore, the probability that a sample of size $s \geq \frac{\ln(1/\delta)}{\epsilon}$ will *not* contain any point in $Y(P_S)$ is at most $(1-\epsilon)^s < \exp(-\epsilon \cdot s) = \delta$, as desired. □

Proof of Theorem 7. Once again, if X is (k,b)-clusterable, then the algorithm always accepts. We thus focus on the case in which X is ϵ-far from being $(k,(1+\beta)b)$-clusterable.

As in the proof of Theorem 6, we view the sample as being selected in phases. Let $p = \Theta(\sqrt{d} \cdot (2/\beta)^d)$ be the number of phases sufficient for the $k=1$ case, and let $p(k) = k \cdot (p+1)$ be the number of phases used here in the analysis of $k > 1$. Let m_j be the size of the sample selected in the jth phase, where $\sum_{j=1}^{p(k)} m_j = m$. Let U_j denote the union of all the samples selected in the first j phases. Thus, $U = U_{p(k)}$ is the complete sample.

Our goal is to show that, with probability of at least 2/3 over the choice of the sample, for *every* partition P of $U_{p(k)}$ we have that $D(P) > b$. To this end we define a family of *influential* partitions. For each phase j there is a subfamily of influential partitions that correspond to that phase. These are partitions of subsets of U_j. We show that with probability of at least 2/3, for every phase j and every influential partition $\hat{P}$ corresponding to that phase, the sample selected in the next phase contains an influential point for $\hat{P}$. This will imply that, after at most $p(k) = k(p+1)$ phases, the diameter of each influential partition, and consequently of *every partition of the sample*, is greater than b.

We define the influential partitions in an inductive manner. In the initial phase (phase 0) there is a single influential partition of a sample of size 1 (i.e., $m_0 = 1$). Suppose that for each influential partition $\hat{P} = (S^1, \ldots, S^k)$ in phase $j-1$, the jth sample contains a point from $Y(\hat{P})$, and let us denote this point by $y(\hat{P})$. (If there is more than one such point, then $y(\hat{P})$ is defined as the one having the smallest index.) Then in phase j we shall have the k influential partitions

$$(S^1 \cup \{y(\hat{P})\}, S^2, \ldots, S^k), \ldots, (S^1, \ldots, S^{k-1}, S^k \cup \{y(\hat{P})\}).$$

This implies that the total number of influential partitions in phase j is at most k^j.

We now apply Claim 5 to each one of the k^{j-1} influential partitions in phase $j-1$ (having diameter at most b). If

$$m_j = \frac{(j-1)\ln k + \ln(3p(k))}{\epsilon},$$

then with probability of at least $1 - \frac{1}{3p(k)}$, the jth sample in fact contains a point $y(\hat{P}) \in Y(\hat{P})$ for every influential partition $\hat{P}$ in phase $j-1$. Setting

$$\begin{aligned} m = \sum_{j=1}^{p(k)} m_j &= \Theta\left(\frac{p(k)^2 \cdot \log k + p(k)\log p(k)}{\epsilon}\right) \\ &= \Theta\left(\frac{k^2 \log k}{\epsilon} \cdot d \cdot (2/\beta)^{2d}\right), \end{aligned}$$

we get that, with probability of at least 2/3, the jth sample contains a point $y(\hat{P}) \in Y(\hat{P})$ for every phase j and every influential partition $\hat{P}$ from phase $j-1$.

Assume that the above event holds and so, in particular, the influential partitions are well defined. We now show that this implies that, after at most $p(k)$ phases, the diameter of every partition of the sample must be greater than b.

Consider any partition $P = (U^1_{p(k)}, \dots, U^k_{p(k)})$ of $U_{p(k)}$, and let $P_j = (U^1_j, \dots, U^k_j)$ be its *restriction* to U_j. That is, $U^i_j = U^i_{p(k)} \cap U_j$. We claim that there must exist a sequence of influential partitions $\hat{P}_1, \dots, \hat{P}_{p(k)}$, where $\hat{P}_j = (S^1_j, \dots, S^k_j)$, so that the following holds: For every i, $S^i_j \subseteq U^i_j$, and for some i, $S^i_j = S^i_{j-1} \cup \{y(\hat{P}_{j-1})\}$. This follows immediately by induction on j: The base of the induction, $j = 0$, is clear. We assume that it is true for $j-1$ and prove it for j. Let $1 \leq i \leq k$ be such that $y(\hat{P}_{j-1}) \in U^i_j$. Then we let $\hat{P}_j = (S^1_{j-1}, \dots, S^i_{j-1} \cup \{y(\hat{P}_{j-1})\}, \dots, S_k)$, which by the definition of the influential partitions is an influential partition.

Let us fix the above sequence of influential partitions. Since there are $p(k) = k \cdot (p+1)$ phases, there must be some $1 \leq i \leq k$ such that in at least $p+1$ phases $j_1, \dots, j_{p+1}$, $S^i_{j_t} = S^i_{j_t - 1} \cup \{y(\hat{P}_{j_t - 1})\}$ (the first such phase will cause $S^i_{j_1}$ to be nonempty). But by our analysis of the $k = 1$ case, this implies that $d(S^i_{p(k)}) > b$. Since $S^i_{p(k)} \subseteq U^i_{p(k)}$, we have that $d(U^i_{p(k)}) > b$, and so $D(P) > b$. Since the above holds for every partition P of $U = U_{p(k)}$, the theorem follows. □

7.3. Finding an approximately good clustering. Similarly to what was shown in section 6 for radius clustering, if X is (k,b)-clusterable, then Algorithm 2 can be used to find an implicit representation of an approximately good $(k,(1+\beta)b)$-clustering of X. Here the process is slightly more complex.

Recall that for a set T of points in $\Re^d$, $I(T)$ denotes the intersection of all d-dimensional balls B_x having radius b that are centered at points $x \in T$.

DEFINITION 5. *Let $P_S = (S^1, \dots, S^k)$ be a partition of a subset $S \subseteq X$. A point $x \in X$ is* compatible *with P_S if there exists an index $1 \leq i \leq k$ such that $x \in I(S^i)$ and $\mathrm{dist}(x,y) \leq (1+\beta)b$ for every $y \in I(S^i)$. Otherwise, x is* incompatible *with P_S.*

*A partition P_S is α-*successful *for a given $0 \leq \alpha \leq 1$ if the number of points that are incompatible with P_S is at most αn. Otherwise, P_S is α-*unsuccessful.

Observe that given an ϵ-successful partition P_S of a subset $S \subseteq X$, the partition P_S can be used to induce an ϵ-good $(k,(1+\beta)b)$-clustering of X (as defined in Definition 1). Also note that by Lemma 4 if a point x is incompatible with a partition P_S, then x must be influential with respect to P_S.

ALGORITHM 3 (approximately good clustering, diameter cost).

1. *Call Algorithm 2 with a sample of size $m = O(\frac{d^{5/2} \cdot k^3}{\epsilon} \cdot (\frac{2}{\beta})^d \log(\frac{d \cdot k}{\epsilon \cdot \beta}))$ and $Cost = D$.*
2. *Let P be the k-way partition of the sample that is found by Algorithm 2 (if such a partition is found).*
3. *View the sample as being selected in $p(k) = \Theta(k \cdot \sqrt{d} \cdot (2/\beta)^d)$ phases, where U_j denotes the union of all samples selected in the first j phases, and $|U_j| = \Theta(\frac{j}{\epsilon} \cdot d \cdot k^2 \cdot \log \frac{d \cdot k \cdot j \cdot p(k)}{\epsilon})$. Let P_j be the restriction of P to U_j. That is, if $P = (U^1, \dots, U^k)$, then $P_j = (U^1 \cap U_j, \dots, U^k \cap U_j)$.*
4. *Take an additional sample of size $\Theta(\log(p(k))/\epsilon)$, and count the number of points that are incompatible with each partition P_j in this additional sample.*[5]
5. *Select the restriction P_g that has the smallest number of incompatible points in the sample and use it to induce a partition of X. That is, if $P_g = (U^1_g, \dots, U^k_g)$, then for every $x \in X$, if there exists an index i such that $x \in I(U^i_g)$ and $\mathrm{dist}(x,y) \leq (1+\beta)b$ for every $y \in I(U^i_g)$, then assign x to cluster i.*

[5] Checking whether a point is incompatible with a given partition can be done by linear programming.

Notice that the above algorithm calls Algorithm 2 with a sample of size slightly larger than what was needed in the proof of Theorem 7. We shall return to this issue at the end of this subsection.

THEOREM 8. *With a probability of at least* $2/3$*, the selected partition* P_g *is* ϵ*-successful.*

DEFINITION 6. *Let* $S \subseteq X$ *be a set of points. A partition* $P_S = (S^1, \ldots, S^k)$ *of* S *is called a* convex partition *if the convex hulls of the different* S^i*'s are disjoint.*

LEMMA 6. *Let* S *be a fixed set of points from* X*, and let* c *be a sufficiently large constant. Consider the uniform selection of a sample of* $s = \frac{d\cdot k^2\cdot\ln(c|S|)+\ln(6p(k))}{\epsilon}$ *points from* X*. Then with a probability of at least* $1 - 1/(6p(k))$ *over the choice of the sample, for every* $(\epsilon/2)$*-unsuccessful convex partition* Q *of* S *there exists at least one point in the sample that is incompatible with* Q*.*

Proof. Let Q be any fixed $(\epsilon/2)$-unsuccessful convex partition of S. The probability that a sample of size s, as stated in the lemma, does not contain a point that is incompatible with Q is at most

$$(1-(\epsilon/2))^s < \exp(-(\epsilon/2)s) = \frac{1}{(c|S|)^{dk^2}\cdot 6p(k)}.$$

It remains to verify that the number of convex partitions of S is at most $(c|S|)^{dk^2}$ for some constant c. Each convex partition of S can be defined by first selecting $\binom{k}{2}$ hyperplanes among the $O(|S|^{d+1})$ hyperplanes that separate $|S|$ points in d dimensions, and then by merging subsets of points that fall into the resulting regions into k clusters. The total number of convex partitions is hence $O(|S|)^{dk^2}$. □

LEMMA 7. *With a probability of at least* $5/6$*, if we select a sample of size*

$$m = \Theta\left(\frac{d^{\frac{5}{2}}\cdot k^3}{\epsilon}\cdot\left(\frac{2}{\beta}\right)^d \log\left(\frac{d\cdot k}{\epsilon\cdot\beta}\right)\right),$$

then for every phase j *and for every convex partition* Q *of* U_j *that is* $(\epsilon/2)$*-unsuccessful, the sample selected in phase* $j+1$ *contains at least one point that is incompatible with* Q*.*

Proof. Let m_j be the size of the sample selected in phase j so that $|U_j| = |U_{j-1}| + m_j$, where $|U_0| = 1$. If we apply Lemma 6 with $S = U_{j-1}$ and $m_j = s$, then it is not hard to verify that

$$|U_j| \leq \frac{2j}{\epsilon}\cdot\left(d\cdot k^2\cdot\log\frac{c\cdot d\cdot k^2\cdot j\cdot(6p(k))}{\epsilon}\right).$$

Since $p(k) = \Theta(k\cdot\sqrt{d}\cdot(2/\beta)^d)$, we have that $|U_{p(k)}| = O(\frac{1}{\epsilon}\cdot d^{5/2}\cdot k^3\cdot(2/\beta)^d\cdot\log\frac{d\cdot k}{\epsilon\cdot\beta})$. Hence, if we take a sample of size $m = |U_{p(k)}|$, then the probability that, for some $1 \leq j \leq p(k)$ and for some $(\epsilon/2)$-unsuccessful convex partition of U_j, the $(j+1)$ sample does *not* contain a point that is incompatible with the partition is at most $p(k)\cdot\frac{1}{6(p(k))} = \frac{1}{6}$. □

COROLLARY 8. *With a probability of at least* $5/6$*, there exists an index* $1 \leq a \leq p(k)$ *such that the restriction* P_a *is* $(\epsilon/2)$*-successful.*

Proof. Algorithm 2 with *Cost* = D finds an optimal partition P of the sample by enumerating all convex partitions of the sample. Since the partition P that Algorithm 2 finds is convex, so is each of its restrictions P_j.

By Lemma 7, with a probability of at least 5/6, for every phase j and for every $(\epsilon/2)$-unsuccessful convex partition Q of the sample U_j selected so far, the sample selected in phase $j+1$ contains a point that is incompatible with Q. Suppose that this in fact happens. Then there exists a phase $1 \leq a \leq p(k)$ such that the restriction P_a is $(\epsilon/2)$-successful. Otherwise, since every incompatible point is an influential one then, similarly to what was argued in the proof of Theorem 7, the partition P could not have diameter at most b. □

Proof of Theorem 8. Let P_a be an $(\epsilon/2)$-successful partition guaranteed with a probability of at least 5/6 by Corollary 8. Then, the probability that the partition P_g selected by Algorithm 3 is ϵ-successful is lower bounded by the probability that the following two events occur:

1. For every ϵ-unsuccessful partition P_j, the fraction of points in the sample that are incompatible with P_j is greater than $3\epsilon/4$;
2. the fraction of points in the sample that are incompatible with the $(\epsilon/2)$-successful partition P_a is at most $3\epsilon/4$.

Clearly, if both events occur, then the selected partition P_g cannot be ϵ-unsuccessful. To lower bound the probability that both these events occur, we upper bound the probability that either one of them does not occur.

Consider any fixed ϵ-unsuccessful partition P_j. The probability that a uniformly selected sample point is incompatible with P_j is greater than ϵ. Since the points are selected independently, by a multiplicative Chernoff bound, the probability that the fraction of incompatible points is at most $3\epsilon/4$ (i.e., $(1-1/4)$ of the expected value) is less than $\exp(-(1/2)\epsilon(1/4)^2 m)$, where m is the size of the sample. Similarly, for the $(\epsilon/2)$-successful partition P_a, the probability that any uniformly selected sample point is incompatible with P_a is at most $\epsilon/2$. By a multiplicative Chernoff bound, the probability that the fraction of incompatible points is greater than $3\epsilon/4$ is less than $\exp(-(1/3)\epsilon(1/2)^2 m)$. Since the number of ϵ-unsuccessful partitions is less than $p(k)$, by a probability union bound, a sample of size $m = \Theta(\log(p(k))/\epsilon)$ ensures that the probability that any one of these "bad" events occurs is at most 1/6, as required.

Adding the two sources of failure, that is, (1) the probability that there is no $(\epsilon/2)$-successful partition P_a and (2) the probability that the selected partition P_g is ϵ-unsuccessful (given that an $(\epsilon/2)$-successful partition P_a exists), we get a total of 1/3 failure probability. □

As noted previously, the size of the sample used by Algorithm 3 is larger than that used for proving Theorem 7. The reason is that in the proof of Theorem 7 we used influential partitions, while here we use convex partitions, whose number is larger. We could not see how to use the former in our (constructive) argument here.

7.4. A lower bound for testing diameter clustering.

We obtain the following lower bound for testing diameter clustering.

THEOREM 9. *For any $\beta > 0$, any algorithm that successfully determines, with probability of at least 2/3, whether X is either $(1,b)$-clusterable or whether X is $(1/2)$-far from being $(1,(1+\beta)b)$-clusterable with respect to the diameter cost must sample $\Omega((\frac{1}{\beta})^{(d-1)/4})$ points from X.*

In order to prove the theorem we shall need the following lemma. We note that a related analysis is performed in [26] in the context of constructing codes.

LEMMA 9. *For any dimension d, value $r \in \Re$, and $\delta > 0$, it is possible to choose $\Omega(\sqrt{d} \cdot (\frac{1}{\delta})^{d-1})$ antipodal pairs of points on the surface of the $(d-1)$-dimensional sphere of radius r, where the distance between any two points is larger than $\delta \cdot r$.*

Proof. We choose the pairs one by one in the following way. Choose a pair of antipodal points that are at a distance greater than $\delta \cdot r$ from all the points chosen so far. Continue to choose antipodal pairs in this way as long as possible.

We claim that the $(d-1)$-dimensional caps of radius $\delta \cdot r$ centered at the points we chose cover the surface of the $(d-1)$-dimensional sphere of radius r. Otherwise, if there exists a point that is not covered, then it must also be the case that its antipodal point is not covered, and thus we can add an additional pair of antipodal points.

Let θ be the angular diameter of a cap of radius $\delta \cdot r$, and let $\theta_0 = \pi/2 - \theta/2$. Then, $\delta = \sqrt{2 - 2\sin\theta_0}$.

Hence, the ratio between the surface area of a $(d-1)$-dimensional sphere of radius r and the surface area of a cap of such a sphere of radius $\delta \cdot r$ is

$$\frac{\int_{-\pi/2}^{\pi/2} \cos^{d-2} t \, dt}{\int_{\theta_0}^{\pi/2} \cos^{d-2} t \, dt}.$$

The numerator is $\Theta(1/\sqrt{d})$ and the denominator is equal to

$$\begin{aligned} \int_{\theta_0}^{\pi/2} \cos^{d-2} t \, dt &= \int_{\theta_0}^{\pi/2} (1 - \sin^2 t)^{\frac{d-3}{2}} \cos t \, dt \\ &\le \int_{\theta_0}^{\pi/2} (2(1 - \sin t))^{\frac{d-3}{2}} \cos t \, dt \\ &= -\frac{1}{2} \cdot \frac{2}{d-1} \cdot (2(1-\sin t))^{\frac{d-1}{2}} \Bigg|_{\theta_0}^{\pi/2} \\ &= \frac{1}{d-1} \cdot (2(1 - \sin\theta_0))^{\frac{d-1}{2}} \\ &= \frac{\delta^{d-1}}{d-1}. \end{aligned}$$

Hence the number of points we can choose is $\Omega(\sqrt{d} \cdot (\frac{1}{\delta})^{d-1})$. □

Proof of Theorem 9. Consider the d-dimensional ball of radius r, where r is slightly greater than $(1+\beta)b/2$. By definition, the distance between any two antipodal points on the surface of this ball is greater than $b(1+\beta)$. By Lemma 9 we can choose $\Omega(\sqrt{d} \cdot (\frac{1}{\delta})^{d-1})$ antipodal pairs of points on the surface of this ball such that the distance between any two points is at least $\delta \cdot r$. Thus, by the Pythagorean theorem, if we choose $\delta > \frac{2\sqrt{\beta(2+\beta)}}{1+\beta} = \Omega(\sqrt{\beta})$, then the distance between any two points that are *not* antipodal is at most b.

Let us fix such a selection of $s = \Omega(\sqrt{d} \cdot (\frac{1}{\sqrt{\beta}})^{d-1})$ antipodal pairs of positions on the surface of the ball and suppose that X is such that we have $n/(s/2)$ points in each position.[6] Clearly, X is 1/2-far from being $(1, (1+\beta)b)$-clusterable. However, by the "birthday paradox" (see, for example, [38]), with high probability, a sample of size $c \cdot \sqrt{s}$ will not contain a pair of points in antipodal positions (for some constant $c < 1$). That is, all the points in the sample will be at a distance of at most b from each other. This implies that our "natural" algorithm (and actually any algorithm having a one-sided error) requires $\Omega((\frac{1}{\beta})^{(d-1)/4})$ sample points.

[6] To ensure that X is an actual set and not a multiset, we can place the points at slightly different but very close positions. Note that our algorithms do not rely on the fact that the points in X are different from each other (or at any minimal distance from each other).

To prove the claim for any algorithm, we can apply an argument similar to that used in the lower bound proofs of [24]. Here we sketch the idea. We define two families of sets of n points, where in the first family all the sets are far from being $(1,(1+\beta)b)$-clusterable, and in the second family all the sets are $(1,b)$-clusterable. The first family is defined by all namings of the n points on the surface of a d-dimensional ball as defined above. In the second family, a set X is defined by selecting, for each one of the s pairs of antipodal positions, one of the positions and putting n/s points in that position. Every such X is $(1,b)$-clusterable.

We now define two processes, one for each family. Each process constructs a random set X in the family as it answers the algorithm's queries, and it completes this construction after the algorithm terminates. Without loss of generality we may assume that the algorithm never queries the same point twice. Then, for each query of the algorithm, the process selects a new point on the sphere in a random fashion which depends on the family to which X belongs.

Assume that we are now answering the jth query, and the processes must decide where to position the new point. Each of the two processes first flips a coin with bias τ, where τ is approximately j/s. According to the outcome, the new point will be placed in the same (or antipodal) position of a previously selected point or placed in an unoccupied position (whose antipodal position is also unoccupied). In the latter case, an antipodal pair is selected uniformly among all unoccupied pairs, and the new point is placed with equal probability on each position in the pair. In the former case, both processes randomly select an occupied position, whereas the second process places the new point in the selected position and the first process places the new point either in this position or in its antipodal position.

Note that as long as the former case does not occur, the distributions on the positions of the points are exactly the same for both processes. However, for a number of queries $j < c \cdot \sqrt{s}$, for some constant $c < 1$, the probability that an occupied position is selected (in either process) is less than 1/3. This implies that the statistical difference between the distributions on sequences of queries and answers for the two processes is less than 1/3, and the theorem follows. □

Remark. Essentially the same argument as in the above proof gives an $\Omega(\sqrt{n})$ lower bound for $\beta = 0$.

8. An alternative analysis for radius clustering. Here we present an alternative analysis of Algorithm 2 when *Cost* $= R$. Recall that the analysis presented in section 6 which works for $\beta = 0$ assumes that the size of the sample is roughly linear in the dimension d. Below we show that it is possible to trade the dependence on d (in terms of the sample complexity) for a dependence on $1/\beta$ (and a slightly higher dependence on k).

We start by analyzing the algorithm for $k = 1$.

THEOREM 10. *Algorithm 2 with sample size* $m = \Theta(\frac{\log(1/\beta)}{\epsilon\cdot\beta})$ *and Cost* $= R$ *is a radius-clustering tester for* $k = 1$, $d \geq 1$, *and* $0 < \beta \leq 1$.

We shall prove the theorem by appealing to the following lemma.

LEMMA 10. *Let* $S \subset \Re^d$, *and let* $z \in \Re^d$ *be the center of the minimum sphere bounding* S. *Consider any point* $y \in \Re^d$ *such that* $\mathrm{dist}(y,z) > t$ *for some* $t \geq r(S)$. *Then*

$$r(S \cup \{y\}) > r(S) \cdot \frac{1}{2} \cdot \left(\frac{t}{r(S)} + \frac{r(S)}{t}\right).$$

Proof. Let $r = r(S)$, let $r' = r(S \cup \{y\})$, and let $z' \in \Re^d$ be the center of the minimum sphere bounding $S \cup \{y\}$. Let H be the hyperplane that passes through z and is perpendicular to the line $z\,z'$. Let H^- be the closed half-space defined by H that does not contain z'. It is easy to verify that there must be a point in $S \cap H^-$ that is at a distance of exactly r from z. Indeed, if no such point exists, we could move the center z by a small distance so that a sphere of radius strictly smaller than r centered at the new position of z would contain all points in S.

Hence, let a be a point in $S \cap H^-$ that is at a distance of exactly r from z. Since the angle $z'\,z\,a$ is obtuse, we have

$$(\mathrm{dist}(z',a))^2 \geq (\mathrm{dist}(z,z'))^2 + (\mathrm{dist}(z,a))^2. \tag{8.1}$$

By our choice of a, we have that $(\mathrm{dist}(z,a))^2 = r^2$, and by definition of z', $(\mathrm{dist}(z',a))^2 \leq (r')^2$. By the triangle inequality,

$$\mathrm{dist}(z,z') \geq \mathrm{dist}(z,y) - \mathrm{dist}(z',y) > t - r',$$

where the last inequality follows from the premise of the lemma and the definition of r'. Combining these inequalities with (8.1), we get that $(r')^2 \geq (t-r')^2 + r^2$. It directly follows that $r' \geq t/2 + r^2/2t$, which is equivalent to $r' > (r/2)(t/r + r/t)$. □

Proof of Theorem 10. If X is $(1,b)$-clusterable, then the algorithm clearly always accepts. Hence, assume from now on that X is ϵ-far from $(1,(1+\beta)b)$-clusterable. We shall show that the algorithm rejects with probability of at least $2/3$.

We view the sample of size m as being selected in $p = \Theta(1/\beta)$ *phases*, where in each phase $\Theta(\log(p)/\epsilon)$ points are selected (uniformly and independently). Let U_i be the union of the samples selected in the first i phases, and let $r_i = r(U_i)$.

We show that with probability of at least $2/3$ over the choice of the sample, for every phase i, $r_i \geq r_{i-1}(1+\alpha_i)$ for some sufficiently large α_i. It will follow that after $O(1/\beta)$ phases, we must obtain that $r_i > b$, causing the algorithm to reject.

For each new phase i, let $z_{i-1} \in \Re^d$ be the center of the minimum sphere bounding U_{i-1}. Since X is ϵ-far from $(1,(1+\beta)b)$-clusterable, there are at least ϵn points $y \in X$ such that $\mathrm{dist}(y,z_{i-1}) > (1+\beta)b$. We shall say that each such point is $(1+\beta)b$-*distant* from z_{i-1}. Suppose that in each phase the sample taken is of size at least $\ln(3p)/\epsilon$. Then for any *fixed* phase i, the probability that no point that is $(1+\beta)b$-distant from z_{i-1} is selected is at most $(1-\epsilon)^{\ln(3p)/\epsilon} < \exp(-\ln(3p)) = 1/(3p)$. The probability that for *some* phase no point that is $(1+\beta)b$-distant from z_{i-1} is selected is therefore less than $1/3$. Hence, assume from now on that for every phase i, the sample selected in this phase contains a point y that is $(1+\beta)b$-distant from z_{i-1}.

We now show that $O(1/\beta)$ phases suffice until $r_i > b$. Let y be a point that is $(1+\beta)b$-distant from z_{i-1}. By Lemma 10, $r(U_{i-1} \cup \{y\}) \geq r(U_{i-1}) \cdot \frac{1}{2}(\frac{(1+\beta)b}{r_{i-1}} + \frac{r_{i-1}}{(1+\beta)b})$. Therefore, assuming such a point is selected in the ith sample, we have

$$r_i \geq r_{i-1} \cdot \frac{1}{2}\left(\frac{(1+\beta)b}{r_{i-1}} + \frac{r_{i-1}}{(1+\beta)b}\right) = \frac{1}{2}\left((1+\beta)b + \frac{r_{i-1}^2}{(1+\beta)b}\right). \tag{8.2}$$

We first observe that for every $i > 1$, $r_i \geq \frac{b}{2}$. Thus, we may assume that $r_2 \geq \frac{b}{2}$. On the other hand, since $\frac{(1+\beta)b}{r_{i-1}} + \frac{r_{i-1}}{(1+\beta)b}$ decreases as r_{i-1} increases, then as long as $r_{i-1} \leq b$,

$$r_i \geq r_{i-1} \cdot \frac{1}{2}\left(1+\beta+\frac{1}{1+\beta}\right) = r_{i-1} \cdot \left(1 + \frac{\beta^2}{2(1+\beta)}\right). \tag{8.3}$$

By applying this lower bound on the rate of increase of r_i (for $i > 2$), we get that after $O(1/\beta^2)$ phases, $r_i > b$. However, we can do a slightly more refined analysis and exploit the fact that for smaller radii, the rate of increase is greater. In particular, let γ be such that $r_{i-1} \leq b(1-\gamma)$. Given (8.2), it can be shown (using simple manipulations) that for every $\gamma \leq \beta$,

$$r_i \geq r_{i-1} \cdot \frac{1}{2}\left(\frac{(1+\beta)}{(1-\gamma)} + \frac{(1-\gamma)}{(1+\beta)}\right) \geq r_{i-1} \cdot \left(1 + \frac{\gamma^2}{2}\right). \tag{8.4}$$

For each integer $1 \leq a < \log(1/\beta)$, let $s(a)$ be the first phase such that $b \cdot (1-2^{-a}) \leq r_{s(a)} < b \cdot (1-2^{-(a+1)})$ (if there exists such a phase). We would like to upper bound the number t of phases required so that $r_{s(a)+t} \geq b \cdot (1-2^{-(a+1)})$. By (8.4), as long as $r_i \leq (1-2^{-a+1})$ we have that $r_{i+1} \geq r_i \cdot (1+2^{-(2(a+1)+1)})$. Therefore, we need t to be such that

$$(1-2^{-a}) \cdot (1+2^{-(2(a+1)+1)})^t \geq (1-2^{-(a+1)}).$$

Since for every $\delta \leq 1/2$ we have the bounds $(1-\delta) \geq \exp(-2\delta)$ and $(1+\delta) \geq \exp(\delta/2)$, it suffices that $t = 2^{a+4}$. It follows that the number of phases required to get from $r_2 \geq b/2$ to $r_i \geq b(1-\beta)$ is at most $16 \cdot \sum_{a=1}^{\log(1/\beta)} 2^a = O(1/\beta)$. Finally, to get from $r_i \geq b(1-\beta)$ to $r_{i+t} > b$, we use the bound from (8.3) and conclude that it takes an additional $O(1/\beta)$ phases. □

8.1. *k* > 1. The proof of the following theorem is analogous to the proof of Theorem 7, where in Theorem 11 we use the arguments from the proof of Theorem 10 as a basis.

THEOREM 11. *Algorithm* 2 *with sample size* $m = \Theta(\frac{k^2 \log k}{\epsilon \cdot \beta^2})$ *and Cost* $= R$ *is a radius-clustering tester for* $k \geq 1$, $d \geq 1$, *and* $0 < \beta \leq 1$.

Acknowledgments. We would like to thank Leonard Schulman and Micha Sharir for helpful information on clustering. We would also like to thank the anonymous referees for comments which helped to improve the presentation of the paper.

REFERENCES

[1] P. K. Agrawal and C. M. Procopiuc, *Exact and approximation algorithms for clustering*, Algorithmica, 33 (2002), pp. 201–226.

[2] P. K. Agrawal and M. Sharir, *Efficient algorithms for geometric optimization*, ACM Comput. Surveys, 30 (1998), pp. 412–458.

[3] P. K. Agrawal, M. Sharir, and S. Toledo, *An Efficient Multi-dimensional Searching Technique and Its Applications*, Technical Report CS-1993-20, Department of Computer Science, Duke University, Durham, NC, 1993.

[4] N. Alon, E. Fischer, M. Krivelevich, and M Szegedy, *Efficient testing of large graphs*, Combinatorica, 20 (2000), pp. 451–476.

[5] N. Alon and M. Krivelevich, *Testing k-colorability*, SIAM J. Discrete Math., 15 (2002), pp. 211–227.

[6] N. Alon, M. Krivelevich, I. Newman, and M Szegedy, *Regular languages are testable with a constant number of queries*, SIAM J. Comput., 30 (2001), pp. 1842–1862.

[7] M. R. Anderberg, *Cluster Analysis for Applications*, Academic Press, New York, 1973.

[8] M. Bender and D. Ron, *Testing properties of directed graphs: Acyclicity and connectivity*, Random Structures Algorithms, 20 (2002), pp. 184–205.

[9] M. Blum, M. Luby, and R. Rubinfeld, *Self-testing/correcting with applications to numerical problems*, J. Assoc. Comput. Mach., 47 (1993), pp. 549–595.

[10] B. Chazelle and J. Matoušek, *On linear-time deterministic algorithms for optimization problems in fixed dimension*, J. Algorithms, 21 (1996), pp. 579–597.

[11] A. Czumaj and C. Sohler, *Combinatorial programs and efficient property testers*, in Proceedings of 43rd Annual Symposium on Foundations of Computer Science, IEEE, Los Alamitos, CA, 2002, pp. 83–92.

[12] A. Czumaj, C. Sohler, and M. Ziegler, *Property testing in computational geometry*, in Proceedings of the 8th ESA, Lecture Notes in Comput. Sci. 1879, M. Paterson, ed., Springer-Verlag, Berlin, 2000, pp. 155–166.

[13] Y. Dodis, O. Goldreich, E. Lehman, S. Raskhodnikova, D. Ron, and A. Samorodnitsky, *Improved testing algorithms for monotonicity*, in Proceedings of RANDOM, Lecture Notes in Comput. Sci. 1671, Springer-Verlag, Berlin, 1999, pp. 97–108.

[14] D. Hochbaum, ed., *Approximation Algorithms for NP-Hard Problems*, PWS Publishing Company, Boston, 1997.

[15] Z. Drezner, ed., *Facility Location*, Springer-Verlag, New York, 1995.

[16] F. Ergun, S. Kannan, S. R. Kumar, R. Rubinfeld, and M. Viswanathan, *Spot-checkers*, J. Comput. System Sci., 60 (2000), pp. 717–751.

[17] T. Feder and D. H. Greene, *Optimal algorithms for approximate clustering*, in Proceedings of 20th Annual Symposium on Theory of Computing, ACM, New York, 1988, pp. 434–444.

[18] E. Fischer, *The art of uninformed decisions: A primer to property testing*, Bull. Eur. Assoc. Theor. Comput. Sci. EATCS, 75 (2001), pp. 97–126.

[19] R. J. Fowler, M. S. Paterson, and S. L. Tanimoto, *Optimal packing and covering in the plane are NP-complete*, Inform. Process. Lett., 12 (1981), pp. 133–137.

[20] A. Frieze and R. Kanan, *Quick approximation to matrices and applications*, Combinatorica, 19 (1999), pp. 175–220.

[21] O. Goldreich, S. Goldwasser, E. Lehman, D. Ron, and A. Samordinsky, *Testing monotonicity*, Combinatorica, 20 (2000), pp. 301–337.

[22] O. Goldreich, S. Goldwasser, and D. Ron, *Property testing and its connection to learning and approximation*, J. Assoc. Comput. Mach., 45 (1998), pp. 653–750.

[23] O. Goldreich and D. Ron, *A sublinear bipartiteness tester for bounded degree graphs*, Combinatorica, 19 (1999), pp. 335–373.

[24] O. Goldreich and D. Ron, *Property testing in bounded degree graphs*, Algorithmica, 32 (2002), pp. 302–343.

[25] T. Gonzalez, *Clustering to minimize the maximum intercluster distance*, Theoret. Comput. Sci., 38 (1985), pp. 293–306.

[26] J. Hamkins and K. Zeger, *Asymptotically dense spherical codes—part* i: *Wrapped spherical codes*, IEEE Trans. Inform. Theory, 43 (1997), pp. 1774–1785.

[27] D. Haussler and E. Welzl, *ϵ-nets and simplex range queries*, Discrete Comput. Geom., 2 (1987), pp. 127–151.

[28] D. Hochbaum and D. Shmoys, *A best possible heuristic for the k-center problem*, Math. Oper. Res., 10 (1985), pp. 180–184.

[29] D. Hochbaum and D. Shmoys, *A unified approach to approximation algorithms for bottleneck problems*, J. Assoc. Comput. Mach., 33 (1986), pp. 533–550.

[30] A. K. Jain and R. C. Dubes, *Algorithms for Clustering*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

[31] J. Jolion, P. Meer, and S. Batauche, *Robust clustering with applications in computer vision*, IEEE Trans. Pattern Analysis Mach. Intell., 13 (1991), pp. 791–802.

[32] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley, New York, 1990.

[33] M. Kearns and D. Ron, *Testing problems with sub-learning sample complexity*, J. Comput. System Sci., 61 (2000), pp. 428–456.

[34] R. Lupton, F. M. Maley, and N. Young, *Data collection for the Sloan digital sky survey—a network-flow heuristic*, J. Algorithms, 27 (1998), pp. 339–356.

[35] N. Megiddo, *Linear programming in linear time when the dimension is fixed*, J. Assoc. Comput. Mach., 31 (1984), pp. 114–127.

[36] N. Megiddo and E. Zemel, *An $o(n \log n)$ randomizing algorithm for the weighted Euclidean 1-center problem*, J. Algorithms, 7 (1986), pp. 358–368.

[37] N. Mishra, D. Oblinger, and L. Pitt, *Sublinear time approximate clustering*, in Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2001, pp. 439–447.

[38] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, Cambridge, UK, 1995.

[39] M. Parnas and D. Ron, *Testing the diameter of graphs*, Random Structures Algorithms, 20 (2002), pp. 165–183.

[40] C. M. Procopiuc, *Clustering Problems and Their Applications (A Survey)*, http://www.cs.duke.edu/˜magda (2000).

[41] P. Raghavan, *Information retrieval algorithms: A survey*, in Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, 1997, pp. 11–18.

[42] E. A. Ramos, *Deterministic algorithms for* 3*-d diameter and some* 2*-d lower envelopes*, in Proceedings of the 16th Symposium on Computational Geometry, ACM, New York, 2000, pp. 290–299.

[43] D. Ron, *Property testing*, in Handbook of Randomized Computing, Vol. II, S. Rajasekaran, P. Pardalos, J. Reif, and J. Rolim, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001, pp. 597–649.

[44] R. Rubinfeld, *On the robustness of functional equations*, SIAM J. Comput., 28 (1999), pp. 1972–1997.

[45] R. Rubinfeld and M. Sudan, *Robust characterizations of polynomials with applications to program testing*, SIAM J. Comput., 25 (1996), pp. 252–271.

[46] L. Schulman, *Private communication*, 2000.

[47] J. Shafer, R. Agrawal, and M. Mehta, *Sprint: A scalable parallel classifier for data mining*, in Proceedings of 22nd International Conference on Very Large Databases, Morgan Kaufmann, San Francisco, 1996, pp. 544–555.

[48] V. N. Vapnik and A. Ya. Chervonenkis, *On the uniform convergence of relative frequencies of events to their probabilities*, Theory Probab. Appl., 16 (1971), pp. 264–280.

SIAM J. Discrete Math.
Vol. 16, No. 3, pp. 418–421

THE SPECTRA OF CYCLE PREFIX DIGRAPHS[*]

FRANCESC COMELLAS[†] AND MARGARIDA MITJANA[‡]

Abstract. Cycle prefix digraphs comprise a class of vertex symmetric digraphs with many interesting properties, such as large order for a given degree and diameter, Hamiltonicity, and hierarchical structure. From their known structural properties, we determine the spectra of the digraphs. We also show that, although cycle prefix digraphs are not distance regular according to the usual definition, they have properties that fully characterize, in the undirected case, distance regular graphs.

Key words. directed graphs, cycle prefix digraph, adjacency matrix, eigenvalues

AMS subject classifications. 05C20, 05C50, 05C12, 05C75, 15A18

PII. S0895480100380604

1. Introduction and notation.

Cycle prefix digraphs were first introduced as Cayley coset digraphs by Faber and Moore in 1988 (see [7]) and have been proposed as a model of interconnection networks for their remarkable properties such as high symmetry, large order for a given degree and diameter, simple shortest path routing, Hamiltonicity [9], pancyclicity [4], optimal connectivity [10], and hierarchical structure [3]. Together with new families constructed from them, cycle prefix digraphs constitute most of the entries of the table of largest known vertex-symmetric (Δ, D) digraphs [2]. When the diameter is two, the cycle prefix digraphs are Kautz digraphs. The spectra of the Kautz digraphs, for any value of the diameter, was given by Delorme and Tillich in [6]. The knowledge of the spectrum of a graph is of interest for its connection to important structural properties of the graph (diameter, bisection width, expansion, etc.).

From the known structural properties of cycle prefix digraphs, we obtain in section 2 their distance matrices as polynomials on the adjacency matrix, and we show that the distance matrices constitute a basis of the adjacency algebra of the digraph. This property fully characterizes, in the undirected case, distance regular graphs, but cycle prefix digraphs are not distance regular, according to the usual definition; see [5]. In section 3 we determine the spectra of the digraphs and we see that, although the adjacency matrix of a cycle prefix digraph is nonsymmetric, its spectrum is real and the number of distinct eigenvalues is the minimum possible.

A cycle prefix digraph $\Gamma_\Delta(D)$ may be defined as a digraph on an alphabet of $\Delta + 1$ symbols $\{1, 2, \ldots, \Delta + 1\}$ as follows: Each vertex $x_1 x_2 \cdots x_D$ is a sequence of distinct symbols from the alphabet. The adjacencies are given by

$$x_1 x_2 \cdots x_D \quad \to \quad \begin{cases} x_2 x_3 x_4 \cdots x_D x_{D+1}, & x_{D+1} \neq x_1, x_2, \ldots, x_D, \\ x_2 x_3 x_4 \cdots x_D x_1, & \\ x_1 x_2 \cdots x_{k-1} x_{k+1} \cdots x_D x_k, & 2 \leq k \leq D-1. \end{cases}$$

[*] Received by the editors November 8, 2000; accepted for publication (in revised form) September 18, 2002; published electronically April 30, 2003. This research was supported by the CICYT, Spain, under project TIC-97-0963.

http://www.siam.org/journals/sidma/16-3/38060.html

[†] Departament de Matemàtica Aplicada IV, Universitat Politècnica de Catalunya, Campus Nord, Edifici C3, J. Girona Salgado 1-3, E-08034 Barcelona, Catalonia, Spain (comellas@mat.upc.es).

[‡] Departament de Matemàtica Aplicada I, Universitat Politècnica de Catalunya, Gregorio Marañon 44, E-08028 Barcelona, Catalonia, Spain (margarida.mitjana@upc.es).

The cycle prefix digraph $\Gamma_\Delta(D)$ is a vertex-symmetric digraph that has order $(\Delta+1)_D = \frac{(\Delta+1)!}{(\Delta+1-D)!}$, has diameter D, and is Δ-regular $(\Delta \geq D)$.

Let d_k be the number of vertices at distance k, $k \leq D$, from a given vertex. In [7], it is shown that there are $(\Delta+1)_k$ vertices in $\Gamma_\Delta(D)$ within distance k from the vertex $12\cdots D$. Therefore, the value of d_k may be determined from here and is also computed in a different way in [11]. We use this result in section 2.

PROPOSITION 1.1. *For $\Gamma_\Delta(D)$, $\Delta \geq D$, the number of vertices d_k at distance k from a given vertex is*

$$d_k = (\Delta+1)\Delta(\Delta-1)\cdots(\Delta-k+3)(\Delta-k+1),$$

where $1 < k \leq D$, $d_0 = 1$, and $d_1 = \Delta$.

2. The distance matrices.

Let A be the adjacency matrix of a digraph Γ of order N and diameter D. For every k, $k = 0, \dots, D$, we define the *k-distance matrix* A_k as a matrix with entries

$$(A_k)_{ij} = \begin{cases} 1 & \text{if } d(i,j) = k, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, it follows from the above definition that $A_0 = I$, $A_1 = A$, and $A_0 + A_1 + A_2 + \cdots + A_D = J$, where J is the matrix in which each entry is 1. When $\Gamma = \Gamma_\Delta(D)$, the distance matrices A_k are polynomials of degree k on the adjacency matrix; more precisely, see the following proposition.

PROPOSITION 2.1. *Let $\Gamma_\Delta(D)$ be the cycle prefix digraph of degree Δ and diameter D, A its adjacency matrix, A_k the k-distance matrix, and $v_k(x) \in \mathbb{R}_k[x]$ the polynomial defined by $v_0(x) = 1$, $v_1(x) = x$, $v_k(x) = (x+1)x(x-1)\cdots(x-k+3)(x-k+1)$, $k = 2, \dots, D$. Then $A_k = v_k(A)$, $k = 0, \dots, D$.*

Proof. From Proposition 1.1, we know that the number of vertices at distance k from a fixed vertex is $d_k = v_k(\Delta)$, where Δ is the degree of the digraph.

To prove that $A_k = v_k(A)$, we will proceed by induction on k. We denote by $d_{ij}^{(k)}$ the (i,j) entry of $v_k(A)$ and by a_{ij} the (i,j) element of the adjacency matrix A.

We will first verify that $A_2 = v_2(A)$.

Since $v_2(A) = A^2 - I$, we have to show that $d_{ij}^{(2)}$ equals 1 if $d(i,j) = 2$ and 0 otherwise. From $v_2(A)$,

$$d_{ij}^{(2)} = \sum_{l=1}^{N} a_{il}a_{lj} \text{ if } i \neq j,$$
$$d_{ii}^{(2)} = \sum_{l=1}^{N} a_{il}a_{li} - 1 \text{ if } i = j.$$

When a summand in $d_{ii}^{(2)}$ is not 0 it is $a_{il}a_{lj} = 1$, and vertex **i** is adjacent to vertex **l**, whereas vertex **l** is adjacent to vertex **j**. Since the shortest path between two different vertices in the cycle prefix digraph is unique [7], there is only a single vertex **l** that verifies $a_{il} = 1$ and $a_{lj} = 1$ so that $d_{ij}^{(2)} = 1$ if $d(i,j) = 2$. Note that, from the definition of $\Gamma_\Delta(D)$, vertex i cannot be adjacent to vertex j. When $i = j$, we have the (only) digon of vertex **i** and, in that case, $d_{ii}^{(2)} = a_{il}a_{li} - 1 = 0$.

Let us now assume that matrix $v_k(A)$ is the k-distance matrix, A_k. We wish to prove that $v_{k+1}(A)$ is the $k+1$-distance matrix, A_{k+1} for $\Gamma_\Delta(D)$. From the definition of the distance polynomials,

$$A_0 + A_1 + \cdots + A_{k-1} = (A+I)A \cdots (A-(k-4)I)(A-(k-3)I)$$

and $v_{k+1}(A)$ is

$$\begin{aligned} v_{k+1}(A) &= (A-(k-1)I)A_k - (A_0 + A_1 + \cdots + A_{k-1}) \\ &= AA_k - (A_0 + A_1 + \cdots + A_{k-1} + (k-1)A_k). \end{aligned}$$

Entries in matrix $v_{k+1}(A)$ fulfill

$$d_{ij}^{(k+1)} = \sum_{l=1}^{N} d_{il}^{(k)} a_{lj} - d_{ij}^{(0)} - \cdots - d_{ij}^{(k-1)} - (k-1) d_{ij}^{(k)}.$$

For each vertex $\mathbf{i}$ in $\Gamma_\Delta(D)$, the element $d_{ii}^{(k+1)} = d_{il}^{(k)} a_{li} - d_{ii}^{(0)} = 1 - 1 = 0$ since vertex $\mathbf{i}$ is in a unique $k+1$-cycle.

For each vertex $\mathbf{j}$ at distance q from $\mathbf{i}$, $0 < q < k$, there exists only one vertex $\mathbf{l}$ such that $d(i,l) = k$ and $d(l,j) = 1$. This follows from calculating the distance between vertices given in [7] and [11]. Thus $d_{ij}^{(k+1)} = d_{il}^{(k)} a_{lj} - d_{ij}^{(q)} = 1 - 1 = 0$.

It is also shown in the above references that, if $d(i,j) = k$, when we calculate the neighbors of vertex $\mathbf{j}$, we obtain $k-1$ vertices at distance also k from i. In such a situation, $d_{ij}^{(k+1)} = 0$.

Finally, if $d(i,j) = k+1$, there exists a unique vertex $\mathbf{l}$ such that $d(i,l) = k$ and $d(l,j) = 1$ because the shortest path is unique. Thus $d_{ij}^{(k+1)} = 1$. □

Corollary 2.2. *Given the cycle prefix digraph $\Gamma_\Delta(D)$ and its adjacency matrix A, we have*

$$(A+I)A(A-I)\cdots(A-(D-2)I) = J.$$

Note that the polynomial obtained above is in fact the so-called Hoffman polynomial of $\Gamma_\Delta(D)$.[1]

We define the *adjacency algebra* $\mathcal{A}$ of $\Gamma_\Delta(D)$ as the algebra generated by all the powers of the adjacency matrix A of the cycle prefix digraph of diameter D and degree Δ. From Proposition 2.1 it is not difficult to verify that each power of A is a linear combination of the $D+1$ linearly independent matrices $A_0, A_1, A_2, \ldots, A_D$. Since $I, A, A^2, \ldots, A^D$ are also linearly independent, both $\{I, A, A^2, \ldots, A^D\}$ and $\{A_0, A_1, A_2, \ldots, A_D\}$ are bases for $\mathcal{A}$. This property, in the undirected case, fully characterizes a distance regular graph [1], but $\Gamma_\Delta(D)$ is not a distance regular digraph in the sense of the definition given by Damerell in [5].

From the equations $A_k = p_0^k I + p_1^k A + \cdots + p_D^k A^D$, $0 \le k \le D$, and Proposition 2.1, we obtain the matrix

$$C = \begin{bmatrix} 1 & & & & \\ p_0^1 & 1 & & & \\ \vdots & \vdots & \ddots & & \\ p_0^{D-1} & p_1^{D-1} & \cdots & 1 & \\ p_0^D & p_1^D & \cdots & p_{D-1}^D & 1 \end{bmatrix},$$

[1] We recall the following result (see, for example, [8]).

Theorem. *Let Γ be a strongly connected digraph with order n and adjacency matrix A. There exists a polynomial $P(x)$ such that $P(A) = J$ if and only if Γ is Δ-regular. In this case, the unique polynomial P of least degree is known as the Hoffman polynomial $H(x) = \frac{nS(x)}{S(\Delta)}$, where $(x-\Delta)S(x)$ is the minimal polynomial of A.*

which is the transition matrix from $\{A_0, A_1, A_2, \ldots, A_D\}$ to $\{I, A, A^2, \ldots, A^D\}$, and hence the inverse transition matrix is (q_i^j), where $A^k = \sum_{i=0}^{D} q_i^k A_i$.

The next result follows from this last equality.

PROPOSITION 2.3. *The number of walks of length k between any two vertices at distance j of $\Gamma_\Delta(D)$is q_j^k, $j \le k \le D$.*

3. The spectrum of $\Gamma_\Delta(D)$.

The knowledge of the Hoffman polynomial of a cycle prefix digraph allows for a straightforward determination of its spectrum.

THEOREM 3.1. *Let $\Gamma_\Delta(D)$ be the cycle prefix digraph of degree Δ and diameter D and A its adjacency matrix. Then*

(i) *A diagonalizes.*

(ii) *The eigenvalues of A are*

$$\lambda_0 = \Delta,\ \lambda_1 = D-2,\ \lambda_2 = D-3, \ldots, \lambda_{D-1} = 0,\ \textit{and}\ \lambda_D = -1.$$

Proof. The minimal polynomial of A is obtained from the Hoffman polynomial found in Corollary 2.2 and is

$$(x-\Delta)(x+1)x \cdots (x-(D-3))(x-(D-2)). \qquad \square$$

The number of distinct eigenvalues is therefore $D+1$, the minimum possible. Note that $\Gamma_\Delta(D)$ and $\Gamma_{\Delta'}(D)$ have the same eigenvalues (but distinct multiplicities), except λ_0, which takes values Δ and Δ', respectively. Note also that $\Gamma_\Delta(D+1)$ has the same eigenvalues as $\Gamma_\Delta(D)$, except $D-1$, which is the second largest eigenvalue of $\Gamma_\Delta(D+1)$.

Acknowledgments. We thank Charles Delorme, Miquel Angel Fiol, and Joan Gimbert for helpful discussions and comments on the manuscript.

REFERENCES

[1] N. BIGGS, *Algebraic Graph Theory*, 2nd ed., Cambridge University Press, Cambridge, UK, 1993.

[2] F. COMELLAS AND M.A. FIOL, *Vertex symmetric digraphs with small diameter*, Discrete Appl. Math., 58 (1995), pp. 1–12.

[3] F. COMELLAS AND M. MITJANA, *Broadcasting in cycle prefix digraphs*, Discrete Appl. Math., 83 (1998), pp. 31–39.

[4] F. COMELLAS AND M. MITJANA, *Cycles in the cycle prefix digraph*, Ars Combin., 60 (2001), pp. 171–180.

[5] R.M. DAMERELL, *Distance-transitive and distance-regular digraphs*, J. Combin. Theory Ser. B, 31 (1981), pp. 46–53.

[6] C. DELORME AND J.-P. TILLICH, *The spectrum of de Bruijn and Kautz graphs*, European J. Combin., 19 (1998), pp. 307–319.

[7] V. FABER, J.W. MOORE, AND W.Y.C. CHEN, *Cycle prefix digraphs for symmetric interconnection networks*, Networks, 23 (1993), pp. 641–649.

[8] A.J. HOFFMAN AND M.H. MCANDREW, *The polynomial of a directed graph*, Proc. Amer. Math. Soc., 16 (1965), pp. 303–309.

[9] M. JIANG AND F. RUSKEY, *Determining the Hamilton-connectedness of certain vertex transitive graphs*, Discrete Math., 133 (1994), pp. 159–170.

[10] E. KNILL, *Notes on the Connectivity of Cayley Coset Digraphs*, Technical report LAUR-94-3719, Los Alamos National Laboratory, Los Alamos, NM, 1994.

[11] M. MITJANA, *Propagació d'Informació en Grafs i Digrafs que Modelen Xarxes d'Interconnexió Simètriques*, Ph.D. thesis, Universitat Politècnica de Catalunya, Catalonia, Spain, 1999.

SIAM J. DISCRETE MATH.
Vol. 16, No. 3, pp. 422–425

ON POLYNOMIAL-FACTOR APPROXIMATIONS TO THE SHORTEST LATTICE VECTOR LENGTH*

RAVI KUMAR† AND D. SIVAKUMAR†

Abstract. For every constant $\epsilon > 0$, we obtain a $2^{O(n(1/2+1/\epsilon))}$ time randomized algorithm to approximate the length of the shortest vector in an n-dimensional lattice to within a factor of $n^{3+\epsilon}$.

Key words. lattice algorithms, shortest lattice vector, SVP

AMS subject classifications. 11H06, 68W20, 68W25

PII. S0895480100379981

Given a lattice L in n dimensions, let $\lambda_1(L)$ denote the Euclidean length of any shortest nonzero vector in L. *SVP-Length* is the problem of exactly computing $\lambda_1(L)$ for a given lattice L. An *ρ-approximate solution of SVP-Length* is an algorithm that, given L, produces a number λ such that $\lambda_1(L)/\rho \leq \lambda \leq \rho\lambda_1(L)$. The complexity of approximately solving SVP-Length is an intriguing problem. This problem is

(1) in time $O(n^n)$ when the factor is 1 (i.e., exact) [9];
(2) NP-hard when the approximation factor is less than $\sqrt{2}$ [2, 13];
(3) in NP $\cap$ co-AM when the factor is $\sqrt{n/\log n}$ [8];
(4) in NP $\cap$ co-NP when the factor is n [11];
(5) in polynomial time when the factor is $(1+\epsilon(n))^n$ for $\epsilon(n) = o(1)$ [12, 14].

An interesting question is, What is the complexity of SVP-Length when the approximation factor is poly(n)? This is not only a natural mathematical question but is also important from the viewpoint of lattice-based cryptography. Indeed, the Ajtai–Dwork cryptosystem [3]—the only known cryptosystem whose security depends on the worst-case hardness of the underlying computational problem—depends precisely on the hardness of finding polynomial approximations to the shortest lattice vector (in a special family of lattices). By results (2) and (3), this problem is unlikely to be NP-hard and, therefore, its precise (deterministic or randomized) complexity becomes important for the cryptographic applications.

An obvious candidate for producing polynomial approximations to SVP-Length—Schnorr's improvement of the Lovász basis reduction algorithm [14]—turns out to be uninteresting: Schnorr's algorithm takes $O(n^2(k^{k/2+o(k)} + n^2))$ arithmetic steps (on polynomial-size operands) to produce a $(\sqrt{6}k)^{n/k}$ approximation. To obtain poly(n) approximation factors, $k = \Omega(n)$, so the running time is $2^{\Omega(n \log n)}$, which is pointless in light of result (4). In this paper, we show the following theorem.

THEOREM 1. *There is an absolute constant $\gamma > 1$ such that for any $\epsilon > 0$, SVP-Length can be approximated to within $n^{3+\epsilon}$ in probabilistic time $2^{\gamma n(1/2+1/\epsilon)}$.*

Remark 1. Subsequent to the appearance of this paper in preliminary form [10], Ajtai, Kumar, and Sivakumar [4] obtained a $2^{O(n)}$ time randomized algorithm for finding the shortest vector in the lattice (and thus subsumes the result in this paper).

*Received by the editors October 16, 2000; accepted for publication (in revised form) October 25, 2002; published electronically April 30, 2003. A preliminary version of this paper appeared in *Proceedings of the* 12*th Annual ACM-SIAM Symposium on Discrete Algorithms*, Washington, D.C., 2001, pp. 126–127.

http://www.siam.org/journals/sidma/16-3/37998.html

†IBM Almaden Research Center, Department K-53/B-1, 650 Harry Road, San Jose, CA 95120 (ravi@almaden.ibm.com, siva@almaden.ibm.com).

The algorithm of [4] is inspired by the algorithm in this paper, together with additional ideas.

Before we present the proof of Theorem 1, we will briefly outline our approach. Our algorithm uses Ajtai's [1] reduction of SVP-Length to the problem of finding a short vector in a special class of lattices; we solve the latter problem by adapting an idea of Blum, Kalai, and Wasserman [5]. To obtain the best approximation factors, we use the sharpest form of the reduction, due to Cai [6] and Cai and Nerurkar [7].

For integers n, m, and q, Ajtai [1] defines a family of lattices in $\mathbf{Z}^m$ defined by $\Lambda(n, m, q) = \{L(A)\}$, where A is an $n \times m$ matrix over $\mathbf{Z}_q$, and $L(A) = \{x \in \mathbf{Z}^m \mid Ax \equiv 0 (\text{mod } q)\}$. The main result of [1] is that if there is an algorithm $\mathcal{A}$ that, with certain settings of q and m, computes a nonzero vector of length n in a lattice chosen uniformly at random from $\Lambda(n, m, q)$ (i.e., the lattice $L(A)$ when A is a uniformly chosen $n \times m$ matrix over $\mathbf{Z}_q$) with nonnegligible probability, then there is a randomized algorithm $\mathcal{B}$ that computes poly(n) approximations to SVP-Length for any n-dimensional lattice. The improved version (see Remark 2) of Ajtai's reduction [7, 6] gives the following theorem.

Theorem 2 (see [1, 7, 6]). *Let $c > 2$, and let m be such that there is a probabilistic algorithm $\mathcal{A}$ that computes a nonzero vector of length $n^{c-2}/2$ in a uniform random lattice in $\Lambda(n, m, n^c)$ with nonnegligible probability. Then there is an algorithm $\mathcal{B}$ that, for any $\delta > 0$, and any lattice $L \in \mathbf{R}^n$, computes a number $\widetilde{\lambda}$ such that $\lambda_1(L)/n^{c+1+\delta} \leq \widetilde{\lambda} \leq \lambda_1(L)$, where $\lambda_1(L)$ is the length of the shortest nonzero vector in L. Furthermore, if we assume that $\mathcal{A}$ runs in time $t(n, m)$, then the $\mathcal{B}$ runs in time* poly$(t(n, m)/\delta)$.

Remark 2. In [1], $c \approx 8$, $m = \Theta(n \log n)$; in [7], $c = 3$, $m = \Theta(n)$. Since we will use Theorem 2, the parameter c determines the approximation factor. The parameter m in Theorem 2 has only one role: it should be suitably large to ensure that with nonnegligible probability a random lattice in $\Lambda(n, m, n^c)$ *does* have a nonzero vector of length at most $n^{c-2}/2$. This is shown in [1, 7] (for every lattice in $\Lambda(n, m, n^c)$) by applying Minkowski's theorem. Other than that, m has no bearing on the approximation factor. It does, however, figure in the running time of $\mathcal{B}$, given $\mathcal{A}$.

Remark 3. Note that for $m > n$, for every lattice in $\Lambda(n, m, n^c)$, Gaussian elimination gives only a vector of Euclidean length $\Theta(n^{c+(1/2)})$.

Proof of Theorem 1. Let $c = 2 + \epsilon/2$, $q = n^c$, $a = \epsilon \log n$, and $b = n/a$. Let $m = (a + n + \ln(aq^b))q^b \leq 2^{\left(\frac{1}{2}+\frac{2}{\epsilon}\right)dn}$ for any constant $d > 1$. We now show that with these settings, given a uniformly random lattice $L(A)$ from $\Lambda(n, m, q)$, with high probability, we will be able to find a nonzero vector of length $n^{\epsilon/2}$ in $L(A)$ in poly(m) time. Once we find such a vector, we can apply Theorem 2 with $\delta = \epsilon/2$ to get an algorithm that can find $n^{3+\epsilon}$-approximations to the SVP-Length in $2^{O(n/\epsilon)}$ time.

Since A is a random $n \times m$ matrix over $\mathbf{Z}_q$, the multiset S that consists of the columns of A is a uniform sample (with replacement) of m vectors in $\mathbf{Z}_q^n$. We will show below how to express any vector $u \in \mathbf{Z}_q^n$ as a sum of at most n^ϵ vectors from S; the 0–1 coefficient vector clearly has Euclidean length $n^{\epsilon/2}$. A nonzero vector in $L(A)$ is obtained by considering the coefficient vector for $u = 0$. The arguments below are adaptations of the arguments by Blum, Kalai, and Wasserman [5].

Divide the n coordinates into a groups of b coordinates each. (Recall that $a = \epsilon \log n$ and $b = n/a$.) Number the groups 1 through a. We will create $a + 1$ sample sets $S_0, S_1, \ldots, S_a \subseteq \mathbf{Z}_q^n$ as follows.

Let $S_0 = S$. The inductive step is to create S_i from S_{i-1}, which is done as follows. Partition S_{i-1} into q^b multisets, one for each $\alpha \in \mathbf{Z}_q^b$, defined by $S_{i-1}^\alpha = \{v \in S_{i-1} \mid$

v agrees with α in group $i\}$. Let u_i denote the projection of u to the coordinates in the ith group. For each α, pick (arbitrarily) a representative $r_{i-1}^{\alpha} \in S_{i-1}^{\alpha}$ and define the multiset

$$S_i = \bigcup_{\alpha} \left\{ (r_{i-1}^{u_i - \alpha} + v) \mid v \in S_{i-1}^{\alpha} \backslash \{r_{i-1}^{\alpha}\} \right\}.$$

We claim that the sample sets $S_0, \ldots, S_a$ satisfy the following properties:

(1) For every i, $0 \leq i \leq a$, every $v \in S_i$ agrees with u on every coordinate in the groups $1, \ldots, i$.

(2) For every i, $0 \leq i \leq a$, the projection of S_i to the coordinates in groups $i+1, \ldots, a$ is a collection of $m - iq^b$ independent and uniformly distributed points from $\mathbf{Z}_q^{b(a-i)}$ (with replacement—thus there could be repetitions).

(3) For every i, $1 \leq i \leq a$, every $v \in S_i$ can be written as the sum of two vectors in S_{i-1}.

It is easy to see that if the construction proceeds successfully, then properties (1) and (3) above are satisfied. We prove property (2) by induction; the base case $i = 0$ is trivial. Assume inductively for $i > 0$ that the projection of S_{i-1} to the coordinates in groups i through a gives a collection of $m - (i-1)q^b$ independent and uniformly distributed points in $\mathbf{Z}_q^{b(a-(i-1))}$. Note that $|S_{i-1}| = m - (i-1)q^b \geq m - aq^b = (a + n + \ln(aq^b))q^b - aq^b = (n + \ln(aq^b))q^b$. Since S_{i-1} contains $\geq n + \ln(aq^b)q^b$ samples whose projection to group i gives uniform and independent vectors in $\mathbf{Z}_q^b$, it follows that for any fixed $\alpha \in \mathbf{Z}_q^b$, the probability that S_{i-1}^{α} is empty is at most $\left(1 - 1/q^b\right)^{(n+\ln(aq^b))q^b} \leq e^{-n}/(aq^b)$. Summing this error probability over all α and over all a stages of the construction, the total error probability is at most e^{-n}. Thus with high probability every S_{i-1}^{α} is nonempty; furthermore, since the projection of S_{i-1} to the coordinates in groups i through a is uniform, the value of r_{i-1}^{α} in groups $i+1$ through a is uniformly distributed. Therefore, the projection of every sample in S_i to the coordinates in groups $i+1$ through a is uniform. For independence, let $x, y \in S_{i-1}^{\alpha}$. The projection of x and y to groups $i+1$ through a are independent random variables, and so $x + r$ and $y + r$ are independent, where $r = r_{i-1}^{u_i - \alpha}$. (Note that it is to maintain stochastic independence that the representatives are thrown out in going from S_{i-1} to S_i.) This completes the induction step.

Finally, note that by property (1), $S_a = \{u\}$ (with certain multiplicity). By properties (2) and (3), u can be written as the sum of $2^a = n^{\epsilon}$ vectors in S_0.

The running time is poly(m), which is $2^{O(n/\epsilon)}$. □

Remark 4. Blum, Kalai, and Wasserman [5] recently gave the first $2^{o(n)}$ time algorithm for learning noisy parity functions in the probably approximately correct (PAC) model. The key idea in their work, which we use in the proof of Theorem 1 above, is to express any vector in $\mathbf{Z}_2^n$ as a linear combination of $O(\sqrt{n})$ vectors from a set of $2^{O(n/\log n)}$ uniformly chosen vectors in $\mathbf{Z}_2^n$. They note that if there is an algorithm that can express any vector as the sum of $O(\sqrt{n})$ vectors from a randomly chosen set of $2^{O(\sqrt{n})}$ vectors in $\mathbf{Z}_2^n$ (which is possible, with high probability), then one can solve the noisy parity problem in $2^{O(\sqrt{n})}$ time. We note that such an algorithm—if it generalizes to $\mathbf{Z}_q$ (as the algorithm in [5] does)—would lead to a $2^{o(n)}$ time algorithm for finding poly(n) approximations to SVP-Length.

REFERENCES

[1] M. Ajtai, *Generating hard instances of lattice problems*, in Proceedings of the 28th Annual ACM Symposium on Theory of Computing, Philadelphia, PA, 1996, pp. 99–108.

[2] M. Ajtai, *The shortest vector problem in L_2 is NP-hard for randomized reductions*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, Dallas, TX, 1998, pp. 10–19.

[3] M. Ajtai and C. Dwork, *A public-key cryptosystem with worst-case/average-case equivalence*, in Proceedings of the 29th Annual ACM Symposium on Theory of Computing, El Paso, TX, 1997, pp. 284–293.

[4] M. Ajtai, R. Kumar, and D. Sivakumar, *A sieve algorithm for the shortest lattice vector problem*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, Crete, Greece, 2001, pp. 601–610.

[5] A. Blum, A. Kalai, and H. Wasserman, *Noise-tolerant learning, the parity problem, and the statistical query model*, in Proceedings of the 32nd Annual ACM Symposium on Theory of Computing, Portland, OR, 2000, pp. 435–440.

[6] J.-Y. Cai, *Applications of a new transference theorem to Ajtai's connection factor*, in Proceedings of the 14th Annual IEEE Conference on Computational Complexity, Atlanta, GA, 1999, pp. 205–214.

[7] J.-Y. Cai and A. Nerurkar, *An improved worst-case to average-case connection for lattice problems*, in Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, Miami Beach, FL, 1997, pp. 468–477.

[8] O. Goldreich and S. Goldwasser, *On the limits of non-approximability of lattice problems*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, Dallas, TX, 1998, pp. 1–9.

[9] R. Kannan, *Minkowski's convex body theorem and integer programming*, Math. Oper. Res., 12 (1987), pp. 415–440.

[10] R. Kumar and D. Sivakumar, *On polynomial approximation to the shortest lattice vector length*, in Proceedings of the 12th Annual ACM–SIAM Symposium on Discrete Algorithms, Washington, D.C., 2001, pp. 126–127.

[11] J. Lagarias, H. W. Lenstra, Jr., and C. P. Schnorr, *Korkine-Zolotarev bases and successive minima of a lattice and its reciprocal lattice*, Combinatorica, 10 (1990), pp. 334–348.

[12] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann., 261 (1982), pp. 515–534.

[13] D. Micciancio, *The shortest vector in a lattice is hard to approximate to within some constant*, in Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, Palo Alto, CA, 1998, pp. 92–98.

[14] C. P. Schnorr, *A hierarchy of polynomial time lattice basis reduction algorithms*, Theoret. Comput. Sci., 53 (1987), pp. 201–224.

SIAM J. DISCRETE MATH.
Vol. 16, No. 3, pp. 426–437

A THEOREM ABOUT THE CHANNEL ASSIGNMENT PROBLEM*

DANIEL KRÁL'† AND RISTE ŠKREKOVSKI‡

Abstract. A list channel assignment problem is a triple (G, L, w), where G is a graph, L is a function which assigns to each vertex of G a list of integers (colors), and w is a function which assigns to each edge of G a positive integer (its weight). A coloring c of the vertices of G is proper if $c(v) \in L(v)$ for each vertex v and $|c(u) - c(v)| \geq w(uv)$ for each edge uv. A weighted degree $\deg_w(v)$ of a vertex v is the sum of the weights of the edges incident with v. If G is connected, $|L(v)| > \deg_w(v)$ for at least one v, and $|L(v)| \geq \deg_w(v)$ for all v, then a proper coloring always exists. A list channel assignment problem is balanced if $|L(v)| = \deg_w(v)$ for all v. We characterize all balanced list channel assignment problems (G, L, w) which admit a proper coloring. An application of this result is that each graph with maximum degree $\Delta \geq 2$ has an $L(2,1)$-labeling using integers $0, \ldots, \Delta^2 + \Delta - 1$.

Key words. graph coloring, list-coloring, channel assignment problem

AMS subject classification. 05C15

PII. S0895480101399449

1. Introduction.

We study a common generalization of coloring, list-coloring, and channel assignment problem. We call this generalization a list channel assignment problem. A *list channel assignment problem* is a triple (G, L, w) in which G is a graph, L is a function which assigns to each vertex of G a set of positive integers, i.e., $L : V(G) \to 2^{\mathbb{N}}$, and w is a function which assigns to each edge of G a positive integer, i.e., $w : E(G) \to \mathbb{N}$. An assignment $c : V(G) \to \mathbb{N}$ of colors to the vertices of G is *proper* if $c(v) \in L(v)$ for each $v \in V(G)$ and $|c(u) - c(v)| \geq w(uv)$ for each $uv \in E(G)$. A *weighted degree* $\deg_w(v)$ of a vertex v of G is the sum of the weights of the edges incident with v. The *maximum weighted degree* $\Delta_w(G)$ is the largest $\deg_w(v)$, where $v \in V(G)$. If $w(e) = 1$ for all $e \in E(G)$, then the problem becomes a list-coloring problem [9, 16]. If $L(v) = \mathbb{N}$, then the problem becomes a channel assignment problem [12]. In the latter case, we define $\chi_w(G)$ to be the smallest number for which there is a proper assignment c such that $1 \leq c(v) \leq \chi_w(G)$ for all $v \in V(G)$. If both $w(e) = 1$ for all $e \in E(G)$ and $L(v) = \mathbb{N}$, then the problem becomes just an ordinary coloring problem for a graph G; note that $\chi(G) = \chi_w(G)$ in this case. A list channel assignment problem can be interpreted as follows: The vertices of G are transmitters, $L(v)$ is a set of frequencies which can be assigned to a vertex v, and $w(uv)$ corresponds to interference between transmitters u and v (the minimal distance between frequencies assigned to u and v).

*Received by the editors December 11, 2001; accepted for publication (in revised form) December 20, 2002; published electronically May 22, 2003. The Institute for Theoretical Computer Science is supported by the Ministry of Education of the Czech Republic as project LN00A056.

http://www.siam.org/journals/sidma/16-3/39944.html

†Department of Applied Mathematics, DIMATIA and Institute for Theoretical Computer Science (ITI), Charles University, Malostranské náměstí 25, 118 00, Prague, Czech Republic (kral@kam.mff.cuni.cz).

‡Department of Applied Mathematics, DIMATIA and Institute for Theoretical Computer Science (ITI), Charles University, Malostranské náměstí 25, 118 00, Prague, Czech Republic (skreko@kam.mff.cuni.cz). Current address: Department of Mathematics, University of Ljubljana, Jadranska 19, 1111 Ljubljana, Slovenia (skreko@fmf.uni-lj.si). This author's research was supported in part by the Ministry of Science and Technology of Slovenia, research project Z1-3129, and in part by GAČR 201/1999/0242.

Some theorems for ordinary colorings, list-colorings, or channel assignment problems may be (naturally) extended to list channel assignment problems but others cannot. A (weighted) graph is called *k-degenerated* if each of its induced subgraphs contains a vertex of (weighted) degree at most k. If $|L(v)| \geq k+1$ for each $v \in V(G)$ and $w(e) = 1$ for each $e \in E(G)$, then there exists a proper coloring. This can be reformulated using terminology of [9, 16]: Each k-degenerated graph is $(k+1)$-choosable; i.e., it admits a proper assignment for any lists such that $|L(v)| \geq k+1$ for all $v \in V(G)$. If we remove the condition $w(e) = 1$, the conclusion becomes false, as noted in [14].

In light of the previous paragraph, it might be surprising to know that if $|L(v)| = \deg_w(v)$ for each $v \in V(G)$ and $|L(v)| > \deg_w(v)$ for at least one $v \in V(G)$, then (G, L, w) admits a proper assignment (Theorem 2.3). This is a counterpart of a well-known inequality $\chi(G) \leq \Delta(G) + 1$, where $\chi(G)$ is the chromatic number of G and $\Delta(G)$ is the maximum degree of G. The inequality $\chi_w(G) \leq \Delta_w(G) + 1$ for the channel assignment problem was recently proved by McDiarmid in [11, 13, 14]. In this paper, we state and prove an analogue of Brooks' theorem for list channel assignment problems. Brooks' theorem for ordinary colorings is proved in [3, 10], for choosability in [5, 17], for list-colorings in [1, 2, 5], and for list-colorings with separation in [8]. An extension of Brooks' theorem for channel assignment problems was stated as an open problem in [12].

A list channel assignment problem is *balanced* if $|L(v)| = \deg_w(v)$ for each $v \in V(G)$. We characterize all balanced list channel assignment problems which admit a proper assignment (Theorem 4.1). In particular, we prove that a balanced list channel assignment problem (G, L, w) admits a proper assignment if G is a 2-connected graph and is neither a complete graph nor an odd cycle (for definitions see subsection 1.1).

We first describe in section 1 a greedy algorithm which was previously used by McDiarmid [11, 13, 14] for channel assignment problems. Then in section 3 we prove Brooks' theorem for list channel assignment problems (G, L, w), where G is a 2-connected graph. Distinct theorems for odd cycles (Theorem 3.5), complete graphs (Theorem 3.8), and remaining 2-connected graphs (Theorem 3.3) are stated. We prove the main theorem, Brooks' theorem for list channel assignment problems, in section 4 (Theorem 4.1). Brooks' theorem for channel assignment problems, and previously known Brooks-type theorems for other problems mentioned earlier can be easily derived from Theorem 4.1. Our result suggests a polynomial algorithm (Corollary 4.2) which, given a balanced list channel assignment, either outputs a proper assignment or states its nonexistence.

We devote subsection 4.1 to corollaries of Theorem 4.1 for the channel assignment problem. We state that if $\chi_w(G) = \Delta_w(G) + 1$ and G is connected, then G is a Gallai tree in Theorem 4.3, and we show an example of such a Gallai tree in Proposition 4.4. Theorem 4.3 can be modified to the "if and only if" form as discussed in subsection 4.1, and those pairs of G and w, for which $\chi_w(G) = \Delta_w(G) + 1$, can be recognized in polynomial time. This gives a complete characterization of pairs of a graph G and a weight function w for which $\chi_w(G) = \Delta_w(G) + 1$, but we do not provide a complete characterization of graphs G for which there exists a weight function w such that $\chi_w(G) = \Delta_w(G) + 1$.

Subsection 4.2 is devoted to $L(2,1)$-labelings of graphs. An *$L(2,1)$-labeling* of a graph is an assignment of integers $0, \ldots, k$ to its vertices such that the numbers assigned to every two neighbors differ by at least two and the numbers assigned to every two vertices at distance two differ by at least one (note that, unlike in the channel assignment problem, the number zero can be assigned to vertices in an $L(2,1)$-labeling). An $L(2,1)$-labeling may be viewed as a special type of the channel assignment problem: The weights of the original edges are set to two and edges of weight one are added between each pair of vertices at distance two. It was conjectured in [7] that there always exists an $L(2,1)$-labeling using integers $0,\ldots,\Delta^2$, where Δ is the maximum degree of the graph for a connected graph G with $\Delta \geq 2$. The existence of an $L(2,1)$-labeling using numbers $0,\ldots,\Delta^2+\Delta$ was proved in [4]; this corresponds to the bound of McDiarmid from [14]. Since the underlying graph of the channel assignment problem obtained from the graph through the above described construction is always 2-connected, our Brooks-type theorem yields that there is an $L(2,1)$-labeling using integers $0,\ldots,\Delta^2+\Delta-1$ for every graph with maximum degree Δ, as stated in Theorem 4.5.

1.1. Notation. We often deal with sets of integers in the paper; we write $[a,b]$ for the interval of integers between a and b (inclusively). We write $G-v$ for the graph obtained from G by deleting the vertex v together with the edges incident with v. We use standard graph notation throughout the paper and, when necessary, refer the reader to various books about graph theory. We briefly recall some lesser-known definitions: A graph with at least $k+1$ vertices is *k-connected* if it remains connected after removing any $k-1$ or less vertices. A *block* of a graph is a maximal (in edge-inclusion) subgraph which is 2-connected. A graph whose blocks are complete graphs and odd cycles is a *Gallai forest*. Gallai forests form an important class of graphs related to colorings of graphs, as shown in [6].

2. Greedy algorithm. The following greedy algorithm was used by McDiarmid [11, 13, 14] to prove an upper bound for the span of channel assignment problems.

ALGORITHM 1.

```
Input:  ordering of the vertices v_1,...,v_n
        edge-weight function w
        lists of colors L[1],...,L[n]
Output: assignment c of the colors to the vertices

color  := minimum color in L[i]'s
maxcol := maximum color in L[i]'s
while color <= maxcol do
  for i := 1 to n do
    if v_i is not colored and color is in L[i] then
      if for all neighbors v_j of v_i which are colored holds
        | c[v_j] - color | >= w ( v_i, v_j ) then
          c[v_i] := color
      fi
    fi
  color := color + 1
done
```

We first state two propositions about Algorithm 1 that are straightforward to prove.

PROPOSITION 2.1. *If Algorithm* 1 *assigns colors to all the vertices, then the obtained assignment is proper.*

PROPOSITION 2.2. *A vertex v of G does not get a color $k \in L(v)$ when Algorithm* 1 *is applied if and only if*

(a) *it is assigned a color $k' < k$, $k' \in L(v)$, or*

(b) *the color k is assigned to a neighbor of v preceding v in the ordering, or*

(c) *a color $k' < k$ is assigned to a neighbor v' of v such that $k - k' < w(vv')$.*

We will prove the list channel assignment counterpart of the well-known inequality $\chi(G) \leq \Delta(G) + 1$.

Theorem 2.3. *Let (G, L, w) be a list channel assignment problem. If $|L(v)| \geq \deg_w(v)$ for each $v \in V(G)$, the inequality is strict for at least one vertex, and G is connected, then (G, L, w) admits a proper assignment.*

Proof. Let $v_1, \ldots, v_n$ be an ordering of the vertices of G such that $|L(v_n)| > \deg_w(v_n)$ and each vertex v_i, $i < n$, has a neighbor v_j such that $j > i$. Such an ordering can be obtained as a postordering of the vertices produced by a depth-first search algorithm started in v_n. We prove that each vertex gets a color when we apply Algorithm 1 to this ordering (this is sufficient due to Proposition 2.1). Let v_i be a fixed vertex of G. Each neighbor u preceding v_i can prevent assigning a color to v_i at most $w(v_i u)$ times, and each neighbor u following v_i can prevent v_i from assigning a color at most $w(v_i u) - 1$ times by Proposition 2.2. This, together with the choice of the ordering, implies that each vertex gets a color. □

One can immediately generalize the usage of Proposition 2.2 in the previous proof and state the following.

Proposition 2.4. *Suppose that Algorithm 1 is applied to a list channel assignment problem (G, L, w) with an ordering $v_1, \ldots, v_n$ of its vertices. If v_i has not been assigned a color, then $L(v_i)$ is a subset of the union of intervals $[c(v_j), c(v_j) + w(v_j v_i) - 1]$, where v_j is a colored neighbor of v_i preceding v_i (i.e., $j < i$) and intervals $[c(v_j) + 1, c(v_j) + w(v_j v_i) - 1]$, where v_j is a colored neighbor of v_i following v_i (i.e., $j > i$).*

We also formulate Proposition 2.4 for the special case when (G, L, w) is balanced.

Proposition 2.5. *Suppose that Algorithm 1 is applied to a balanced list channel assignment problem (G, L, w) with an ordering $v_1, \ldots, v_n$ of its vertices such that for each vertex v_i, $1 \leq i \leq n-1$, there is $j > i$ such that v_i and v_j are adjacent. Then the vertices $v_1, \ldots, v_{n-1}$ are assigned colors. If v_n has not been colored, then*

$$L(v_n) = \bigcup_{v_i v_n \in E(G)} [c(v_i), c(v_i) + w(v_i v_n) - 1],$$

where the intervals in the above union are disjoint.

3. 2-connected graphs.

Lemma 3.1. *Let (G, L, w) be a balanced list channel assignment problem. If G is 2-connected and $\min \bigcup_{v \in V(G)} L(v)$ or $\max \bigcup_{v \in V(G)} L(v)$ is not contained in all the lists, then (G, L, w) admits a proper assignment.*

Proof. We first deal with the case where the minimum color is not contained in all the lists. Let $c_m = \min \bigcup_{v \in V(G)} L(v)$. Since G is connected, there exist adjacent vertices v_1 and v_n such that $c_m \in L(v_1)$ and $c_m \notin L(v_n)$. Let $v_1, \ldots, v_n$ be an ordering of the vertices of G such that each vertex v_i, $i < n$, has a neighbor v_j with $j > i$. Such an ordering can be a postordering of the vertices produced by a depth-first search algorithm applied to $G - v_1$ started in v_n. Let us apply Algorithm 1. Each vertex (with the possible exception of v_n) has been assigned a color due to Proposition 2.5. If v_n has not been assigned a color, then the facts that the color of v_1 is c_m and $c_m \notin L(v_n)$ yield a contradiction due to Proposition 2.5.

The case when the maximum color is not contained in all the lists can be dealt with as follows: Let $L'(v) = \{M - k | k \in L(v)\}$ for sufficiently large M. Then (G, L', w)

has a proper assignment c' because the minimum color is not contained in all the lists. The mapping $c(v) = M - c'(v)$ is a proper assignment of (G, L, w). □

The following lemma can be found in [15, Lem. 1.15].

Lemma 3.2. *Every 2-connected graph G which is neither a cycle nor a complete graph contains vertices x, y, and z such that x and y are neighbors of z, the vertices x and y are nonadjacent, and $G - x - y$ is connected.*

Theorem 3.3. *Let (G, L, w) be a balanced list channel assignment problem. If G is 2-connected and is neither an odd cycle nor a complete graph, then (G, L, w) admits a proper assignment.*

Proof. Let $c_1 = \min \bigcup_{v \in V(G)} L(v)$ and $c_2 = \max \bigcup_{v \in V(G)} L(v)$. If c_1 or c_2 is not contained in all the lists, we apply Lemma 3.1.

Suppose that G is an even cycle. Let $v_1, \ldots, v_{2n}$ be the vertices of the cycle and let $c(v_i) = c_1$ for odd i and $c(v_i) = c_2$ for even i. The assignment c is proper due to

$$|c(v_i) - c(v_{i+1})| = c_2 - c_1 \geq |L(v_i)| - 1 = w(v_{i-1}v_i) + w(v_i v_{i+1}) - 1 \geq w(v_i v_{i+1}).$$

Now we deal with the case when G is not a cycle. Let x, y, and z be three vertices with the properties of Lemma 3.2. Let $x, y, v_3, \ldots, v_{n-1}, v_n = z$ be an ordering of the vertices of G such that each vertex v_i, $3 \leq i < n$, has a neighbor v_j with $j > i$. Such an ordering can be obtained as a postordering of the vertices produced by a depth-first search algorithm applied to $G - x - y$ started in $v_n = z$. Let us apply Algorithm 1. This yields a partial assignment c. Each vertex (with the possible exception of v_n) has been assigned a color due to Proposition 2.5. If $v_n = z$ has not been assigned a color, then $L(v_n) = \bigcup_{v_i v_n \in E(G)} [c(v_i), c(v_i) + w(v_i v_n) - 1]$ and the intervals in the union have to be disjoint by Proposition 2.5. Since $c(x) = c(y) = c_1$, this is a contradiction. □

3.1. Coloring odd cycles. We assume throughout this subsection that (G, L, w) is a balanced list channel assignment problem such that G is an odd cycle and $\{c_1, c_2\} \subseteq L(v)$ for all $v \in V(G)$, where $c_1 = \min \bigcup_{v \in V(G)} L(v)$ and $c_2 = \max \bigcup_{v \in V(G)} L(v)$. Note that $c_2 - c_1 \geq w(e)$ for each edge $e \in E(G)$.

Lemma 3.4. *Suppose that (G, L, w), with the properties described in the beginning of the subsection, does not admit a proper list assignment. Then, for each of its vertices v incident with edges e_1 and e_2, one of the following holds:*

$$L(v) = \begin{cases} [c_1, c_1 + w(e_1) - 1] \cup [c_2 - w(e_1) + 1, c_2] & \text{if } w(e_1) = w(e_2), \\ [c_1, c_2] & \text{otherwise.} \end{cases}$$

Proof. Suppose that the claim is false. Let v be a vertex adjacent to the edges e_1 and e_2, which does not satisfy either of the above cases. In particular, $L(v)$ is not an interval. Let us assume $w(e_1) \leq w(e_2)$. If $c_2 - w(e_2) < c_1 + w(e_1)$, then we get that $w(e_1) + w(e_2) = c_2 - c_1 - 1$ and $L(v)$ is an interval. We prove that there is $k \in L(v)$ such that $c_1 + w(e_1) \leq k \leq c_2 - w(e_2)$ or $c_1 + w(e_2) \leq k \leq c_2 - w(e_1)$. If there is no such k, then $L(v) \subseteq [c_1, c_1 + w(e_1) - 1] \cup [c_2 - w(e_2) + 1, c_2]$ and $L(v) \subseteq [c_1, c_1 + w(e_2) - 1] \cup [c_2 - w(e_1) + 1, c_2]$. This implies the following inclusion:

$$L(v) \subseteq [c_1, c_1 + w(e_1) - 1] \cup [c_2 - w(e_2) + 1, c_1 + w(e_2) - 1] \cup [c_2 - w(e_1) + 1, c_2].$$

Note that the middle interval in the above union might be empty. From the above inclusion one gets easily that $|L(v)| \leq w(e_1) + w(e_2) - 1$, which contradicts that (G, L, w) is balanced.

Let k be such that $c_1 + w(e_1) \leq k \leq c_2 - w(e_2)$ or $c_1 + w(e_2) \leq k \leq c_2 - w(e_1)$. Then we can alternately assign to the vertices of G (except for v) colors c_1 and c_2

and to the vertex v the color k. The above inequalities ensure that one of the two possible alternating assignments is proper. □

THEOREM 3.5. *A list channel assignment problem (G, L, w) with the properties described in the beginning of this subsection does not have a proper assignment if and only if there exist integers $1 \leq a \leq b$ and $1 \leq k$ such that one of the following holds for each vertex v:*

(a) *The vertex v is adjacent to two edges with weights a and $L(v) = [k, k + a - 1] \cup [k + b, k + a + b - 1]$.*

(b) *The vertex v is adjacent to two edges of different weights a and b (thus $a < b$ in this case) and $L(v) = [k, k + a + b - 1]$.*

Proof. If (G, L, w) does not admit a proper assignment, then it is of the form described in Lemma 3.4: Let a be the weight of the lightest edge, k the minimum color contained in all the lists (cf. Lemma 3.1), and k' the maximum one. Let $b = k' - k - a + 1$. If a vertex is incident with two edges both of weight a, its list is a disjoint union of two intervals of length a by Lemma 3.4 and, due to the choice of a and b, it is of the form described in (a). On the other hand, if a vertex is incident with an edge of weight a and an edge of another weight, its list has to be an interval $[k, k'] = [k, k + a + b - 1]$. Thus the weight of its other edge is b by Lemma 3.4. The other end-vertex of the edge of weight b cannot be incident with another edge of weight b; thus its list has to be an interval $[k, k']$ and the weight of the other edge (distinct from that with weight b) is a. In this fashion, one can prove that each vertex of the cycle is incident with an edge of weight a and the lists of the vertices are of the form described in the theorem.

Next, we will prove that the list channel assignment problems described in the theorem do not admit proper assignments. We say a vertex has been assigned a *low* color if its color is in $[k, k+a-1]$ and it has been assigned a *high* color if its color is in $[k+b, k+a+b-1]$. Each vertex must be assigned either a low or a high color: A vertex incident with an edge with weight b cannot be assigned color in $[k + a, k + b - 1]$—this would disable coloring the other end of such an edge because of $a \leq b$. No two adjacent vertices can be assigned both low and high colors (the weight of each edge is at least a). This, together with the fact that G is an odd cycle, proves the theorem. □

3.2. Coloring complete graphs. We assume throughout this subsection that (G, L, w) is a balanced list channel assignment problem such that G is a complete graph.

LEMMA 3.6. *Suppose that (G, L, w) with the properties described in the beginning of this subsection does not admit a proper assignment. Let $v_1, \dots, v_n$ be an ordering of the vertices of G and c the assignment obtained by Algorithm 1 applied to the sequence $v_1, \dots, v_n$ (this assigns all vertices except v_n colors due to Proposition 2.4). The following hold:*

(a) $c(v_1) < \cdots < c(v_{n-1})$;

(b) $\bigcup_{j=1}^{i-1}[c(v_j), c(v_j)+w(v_jv_i)-1] = \{k | k \in L(v_i) \wedge k < c(v_i)\}$ *for all* $2 \leq i \leq n-1$;

(c) $L(v_n) = \bigcup_{1 \leq i \leq n-1}[c(v_i), c(v_i) + w(v_iv_n) - 1]$.

Proof. Let k_1 be the least color contained in any of the lists; by Lemma 3.1, k_1 is contained in all the lists. If there is $i > 1$ such that $[k_1, k_1 + w(v_1v_i) - 1] \not\subseteq L(v_i)$, then Algorithm 1 applied to the sequence $v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_n, v_i$ yields a proper assignment due to Proposition 2.5 (the partitioning described in it cannot exist since $[k_1, k_1 + w(v_1v_i) - 1] \not\subseteq L(v_i)$ and $c(v_1) = k_1$).

Let k_2 be the least color contained in the following set:

$$\bigcup_{i=2}^{n} \left(L(v_i) \setminus [k_1, k_1 + w(v_1v_i) - 1]\right).$$

The color k_2 is the second (according to the time) color assigned to a vertex of G by Algorithm 1 applied to the sequence $v_1, \dots, v_n$. We prove that $k_2 \in L(v_2)\backslash[k_1, k_1 + w(v_1v_2) - 1]$ and $[k_2, k_2 + w(v_2v_i) - 1] \subseteq L(v_i)\backslash[k_1, k_1 + w(v_1v_i) - 1]$ for $i \geq 3$: If $k_2 \notin L(v_2)\backslash[k_1, k_1 + w(v_1v_2) - 1]$, we apply Algorithm 1 to the sequence $v_1, v_3, \dots, v_n, v_2$. We get a coloring of (G, L, w)—the partitioning described in Proposition 2.5 cannot exist because there is a vertex with a color k_2 and $k_2 \notin L(v_2)\backslash[k_1, k_1 + w(v_1v_2) - 1]$. Thus Algorithm 1 assigns color k_2 to v_2. If there is $i \geq 3$ such that $[k_2, k_2 + w(v_2v_i) - 1] \nsubseteq L(v_i)\backslash[k_1, k_1 + w(v_1v_i) - 1]$, then we apply Algorithm 1 to the sequence $v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_n, v_i$. We get a coloring of (G, L, w) by Proposition 2.5 (the partitioning described in it cannot exist since $[k_2, k_2 + w(v_2v_i) - 1] \nsubseteq L(v_i)\backslash[k_1, k_1 + w(v_1v_i) - 1]$, $c(v_1) = k_1$, and $c(v_2) = k_2$). Hence, $[k_2, k_2 + w(v_2v_i) - 1] \subseteq L(v_i)$ for all $i \geq 3$.

Let k_3 be the third (according to the time) color assigned to a vertex of G by Algorithm 1 applied to the sequence $v_1, \dots, v_n$. Then the following hold (for $i \geq 4$):

$$k_3 \in L(v_3)\backslash \left([k_1, k_1 + w(v_1v_3) - 1] \cup [k_2, k_2 + w(v_2v_3) - 1]\right)$$

and

$$[k_3, k_3 + w(v_3v_i) - 1] \in L(v_i)\backslash \left([k_1, k_1 + w(v_1v_i) - 1] \cup [k_2, k_2 + w(v_2v_i) - 1]\right).$$

The argument is essentially the same as in the previous paragraph: If the above is not the case, then we apply Algorithm 1 to the sequence v_1, v_2, $v_4, \dots, v_n$, v_3 (if the first is false) or $v_1, \dots, v_{i-1}, v_{i+1}, v_n, v_i$ (if the latter is false), and we get a proper assignment. We conclude that Algorithm 1 assigns color k_3 to v_3. We can continue in this fashion, and we assign the colors $k_4, \dots, k_{n-1}$ to $v_4, \dots, v_{n-1}$, respectively.

Finally, claim (a) follows from $k_1 < k_2 < \cdots < k_{n-1}$. Claim (b) is established by inclusions $[k_1, k_1 + w(v_1v_2)] \subseteq L(v_2)$, $[k_1, k_1 + w(v_1v_3)] \cup [k_2, k_2 + w(v_2v_3)] \subseteq L(v_3)$, etc. Claim (c) also follows from these inclusions as well as from Proposition 2.5. □

Lemma 3.7. *Suppose that (G, L, w) with the properties described in the beginning of this subsection does not admit a proper assignment. If a vertex v is adjacent to at least two edges of different weights, then $L(v)$ is an interval.*

Proof. Let v be a fixed vertex of G and k the smallest number of $L(v)$. Let a be the minimum weight of an edge adjacent to v and b the maximum weight of an edge adjacent to v. Consider an ordering $O = v_1, \dots, v_n$ of the vertices of G such that $v_n = v$, $w(v_{n-2}v_n) = a$, and $w(v_{n-1}v_n) = b$. Let $w_i = w(v_iv)$ for $1 \leq i \leq n-1$. We prove by induction on i that $[k, k + w_1 + \cdots + w_i - 1] \subseteq L(v)$. Since (G, L, w) is balanced, we get easily that $L(v)$ is an interval.

If $i = 1$, it is enough to apply Lemma 3.6 to the above sequence of the vertices of G. Let us suppose $i > 1$. Suppose first that $w_{i-1} < b$. If we apply Lemma 3.6 to the sequence $v_1, \dots, v_{i-2}, v_{n-1}, v_{i-1}, v_i, \dots, v_{n-2}, v_n$, we get that $k + w_1 + \cdots + w_{i-1} \in L(v)$ since $[k, k + w_1 + \cdots + w_{i-1} - 1] \subseteq L(v)$ and $w_{i-1} < b$. This is because $[k, k + w_1 + \cdots + w_{i-1} - 1] \subseteq L(v)$ and $L(v)$ can be covered by intervals of length $w_1, w_2, \dots, w_{i-2}, w_{n-1} = b, w_{i-1}, w_i, \dots, w_{n-2}$, which follow one after another. If we apply Lemma 3.6 to the ordering O, then we get $[k, k + w_1 + \cdots + w_i - 1] \subseteq L(v)$ since Algorithm 1 applied to the ordering O colors v_i by $k + w_1 + \cdots + w_{i-1}$.

We deal with the remaining case $w_{i-1} = b$ in this paragraph. We first prove that $k+w_1+\cdots+w_{i-1} \in L(v)$: Let us apply Lemma 3.6 to the sequence $v_1, \ldots, v_{i-2}, v_{n-2}, v_{i-1}, v_i, \ldots, v_{n-3}, v_{n-1}, v_n$. Observe that $[k, k+w_1+\cdots+w_{i-1}-1] \subseteq L(v)$, $w_{n-2} < b = w_{i-1}$ and $L(v)$ can be covered by intervals of length $w_1, w_2, \ldots, w_{i-2}, w_{n-2} = a, w_{i-1}, w_i, \ldots, w_{n-3}, w_{n-1}$, which follow one after another. Next, we apply Lemma 3.6 to the ordering O, and we conclude that $[k, k+w_1+\cdots+w_i-1] \subseteq L(v)$. The argument is as in the previous case. □

THEOREM 3.8. *Let (G, L, w) be a list channel assignment problem with the properties described in the beginning of this subsection and let $V(G) = \{v_1, \ldots, v_n\}$. Then, (G, L, w) does not admit a proper assignment if and only if one of the following holds:*

(a) *There exist integers $1 \leq a$ and $1 \leq k_1 < \cdots < k_{n-1}$ such that $k_i + a \leq k_{i+1}$ for $1 \leq i \leq n-2$, $w(e) = a$ for all $e \in E(G)$, and $L(v_i) = \bigcup_{1\leq j\leq n-1}[k_j, k_j+a-1]$ for all $1 \leq i \leq n$.*

(b) *There exist integers $1 \leq a < b$ and $1 \leq k$ such that (possibly after an appropriate permutation of the vertices) $w(v_iv_j) = b$ for $1 \leq i, j \leq n-1$, $w(v_iv_n) = a$ for $1 \leq i \leq n-1$, $L(v_i) = [k, k+b(n-2)+a-1]$ for $1 \leq i \leq n-1$, and $L(v_n) = \bigcup_{0\leq j\leq n-2}[k+bj, k+bj+a-1]$.*

Proof. None of the list channel assignment problems described in the statement admit a proper assignment. We prove that the problems described are the only ones which do not admit a proper assignment. We distinguish several cases:

- **The weights of all the edges are the same.** Let a be the common weight of all the edges. By Lemma 3.6, it is enough to prove that $L(v_i) = L(v_j)$ for all $1 \leq i, j \leq n$. Suppose this is false. We may assume that $L(v_{n-1}) \neq L(v_n)$. Let κ be the color assigned to v_{n-1} by Algorithm 1 applied to the sequence $v_1, \ldots, v_n$. Then by Lemma 3.6,

  $$\{i | i \in L(v_{n-1}) \wedge i < \kappa\} = \{i | i \in L(v_n) \wedge i < \kappa\} \text{ and } [\kappa, \kappa+a-1] \subseteq L(v_n).$$

  If we apply Lemma 3.6 to the sequence $v_1, \ldots, v_{n-2}, v_n, v_{n-1}$, we get $[\kappa, \kappa+a-1] \subseteq L(v_{n-1})$. This implies that $L(v_{n-1}) = L(v_n)$.
- **All the lists are intervals.** We claim that the weights of the edges are the same (which was dealt with in the first case). Otherwise, there exists a vertex adjacent to two edges of different weights. We may assume that v_1 is such a vertex, the edge v_1v_2 has the largest weight incident with v_1, and v_1v_3 has the smallest weight incident with v_1. If we apply Algorithm 1 to the sequence $v_1, \ldots, v_n$, we get an assignment such that $c(v_3) < c(v_2)$, which is contradicted by Lemma 3.6.
- **There exist edges of different weights and a vertex whose list is not an interval.** Let k be the smallest color in the lists. By Lemma 3.1, k is contained in all the lists. Let v_n be a vertex such that $L(v_n)$ is not an interval. By Lemma 3.7, the edges incident with v_n have the same weight. Let a be their common weight. We first prove that the weight of any other edge is at least a. Suppose the opposite and assume that $w(v_1v_2) < a$ (note that then $L(v_2)$ is an interval by Lemma 3.7). If we apply Algorithm 1 to the sequence $v_1, v_n, v_2, \ldots, v_{n-1}$, we get an assignment such that $c(v_2) < c(v_n)$, which is impossible due to Lemma 3.6. Thus the weight of each edge in the graph is at least a. Further, let b be the largest weight of an edge. We may assume that $w(v_1v_2) = b$. Note that $a < b$ since there are edges of different weights, and thus both $L(v_1)$ and $L(v_2)$ are intervals.

  Lemma 3.6 applied to the sequence $v_1, v_2, \ldots, v_n$ gives (because the algorithm assigns k to v_1 and $k + w(v_1v_2) = k+b$ to v_2) that $L(v_n) \cap [k, k+b] =$

$[k, k+a-1] \cup \{k+b\}$. If there is an edge $e = xy$ such that $a \le w(e) < b$ where neither x nor y is v_n, then Algorithm 1 applied to the sequence $x, y, \dots, v_n$ assigns x the color k and y the color $k + w(e)$ (note that $L(y)$ has to be an interval by Lemma 3.7) due to Lemma 3.6. However, by the same lemma, $k + w(e) \in L(v_n)$, which is false (note that $k + a \le k + w(e) < k + b$). Thus the weights of all the edges which are not incident with v_n are b. Lemma 3.6 applied to the sequence $v_1, \dots, v_{i-1}, v_{i+1}, \dots, v_{n-1}, v_i, v_n$ gives, together with Lemma 3.7, that $L(v_i) = [k, k + (n-2)b + a - 1]$ and $L(v_n) = \bigcup_{0 \le i \le n-2} [k + bi, k + bi + a - 1]$. □

4. Brooks' theorem for a list channel assignment problem. We prove that only bad list channel assignment problems can be obtained by pasting bad odd cycles and bad complete graphs whose lists are disjoint. Recall that it is enough to consider balanced list channel assignment problems by Theorem 2.3.

THEOREM 4.1. *A balanced list channel assignment problem (G, L, w) does not admit a proper assignment if and only if G is a Gallai forest whose blocks are $G_1, \dots, G_m$ and there exist $L_i : V(G_i) \to 2^{\mathbb{N}}$ for $1 \le i \le m$ with the following properties:*

- *$L(v)$ is a union of $L_i(v)$ for i such that $v \in G_i$ and $L_i(v) \cap L_j(v) = \emptyset$ for $i \ne j$.*
- *Let w_i be the weight function w restricted to the edges of G_i. Then (G_i, L_i, w_i) is balanced and is one of the following three types:*
  - (a) *G_i is an odd cycle and there exist integers $1 \le a \le b$ and k such that each vertex v of G_i is* either *incident with two edges with the weights a in (G_i, L_i, w_i) and $L_i(v) = [k, k+a-1] \cup [k+b, k+a+b-1]$* or *$v$ is incident with an edge with weight a and an edge with weight b and $L_i(v) = [k, k+a+b-1]$.*
  - (b) *G_i is a complete graph with n vertices. There exist integers $1 \le a$ and $1 \le k_1 < \dots < k_{n-1}$ such that $k_j + a \le k_{j+1}$ for $1 \le j \le n-2$, $w_i(e) = a$ for all $e \in E(G_i)$, and $L_i(v) = \bigcup_{1 \le j \le n-1} [k_j, k_j + a - 1]$ for all $v \in V(G)$.*
  - (c) *G_i is a complete graph with n vertices. There exist integers $1 \le a < b$, $1 \le k$, and a vertex $v \in V(G_i)$ such that $w_i(e) = b$ for each edge e of G_i which is not incident with v, $w_i(e) = a$ for each edge e of G_i incident with v, the list $L_i(u) = [k, k + b(n-2) + a - 1]$ for each $u \in V(G)$, $u \ne v$, and $L(v) = \bigcup_{0 \le i \le n-2} [k + bi, k + bi + a - 1]$.*

Proof. It is enough to prove the theorem for connected graphs. The proof proceeds by induction on the number of blocks. If G has just one block, then the proof immediately follows from Theorems 3.3, 3.5, and 3.8.

Suppose G has at least two blocks. Let G_1 be one of its end-blocks, v the cut vertex separating G_1 from the rest of G, and let G' be the rest of G including v. Let L_1 (resp., L') be the function L restricted to G (resp., G') except for v, and let w_1 (resp., w') be the function w restricted to the edges of G (resp., G'). Let U_1 (resp., U') be the largest set of colors such that the list channel assignment problem (G_1, L_1, w_1) (resp., (G', L', w')) with $L_1(v) = U_1$ (resp., $L'(v) = U'$) does not admit a proper assignment. Note that $|U_1| \le \deg_{w_1}(v)$ and $|U'| \le \deg_{w'}(v)$ due to Theorem 2.3. The sets U_1 and U' are uniquely determined (they are simply the sets of those colors such that when assigned to v, there is no proper extension to the rest of the graph). Thus for each $k \in L(v) \setminus U_1$ (resp., $k \in L(v) \setminus U'$) there is a proper assignment of (G_1, L_1, w_1) (resp., (G', L', w')) such that the color of v is k.

If there is k such that $k \in L(v) \setminus (U_1 \cup U')$, we can assign to v the color k and extend this to a proper assignment of G_1 and G', and thus to a proper assignment of G. If

(G, L, w) does not admit a proper assignment, then $|U_1| = \deg_{w_1}(v)$, $|U'| = \deg_{w'}(v)$, and $L(v) = U_1 \cup U'$ (we have equality because (G, L, w) is balanced). In such a case, (G_1, L_1, w_1) has to be either an odd cycle described in (a) due to Theorem 3.5 or a complete graph described in (b) or (c) due to Theorem 3.8. The channel assignment problem (G', L', w') is of the desired form due to the induction hypothesis.

On the other hand, if (G_1, L_1, w_1) is a "bad" cycle or a "bad" complete graph and (G', L', w') is the union of "bad" cycles and complete graphs described in the statement of the theorem, then (G, L, w) does not admit a proper assignment. $\square$

Note that the proof of Theorem 4.1 suggests an algorithm for recognizing balanced channel assignment problems which admit proper assignments: We take an end-block of the given graph and we consider one of its vertices which is not a cut vertex. This vertex, together with the weights of the edges of that block, determines the type of a bad graph (if it is bad) and the corresponding lists of colors at each vertex. We remove this block and continue until either we find an end-block which is not bad or we get an empty graph. If we find a block which is not bad, we use the way suggested by the proofs of Theorem 3.3, Theorem 3.5, and Theorem 3.8 to color it. Hence, we may conclude as follows.

Corollary 4.2. *There exists a polynomial-time algorithm which for a given balanced list channel assignment problem either finds a proper assignment or decides that a proper assignment does not exist.*

4.1. Channel assignment problem. Theorem 4.1 provides results for the channel assignment problem when applied to the lists which all are equal to $[1, \Delta_w(G)]$, where G is a given graph and w is a weight function on the edges of G. Recall that if there is a vertex v in G such that $\deg_w(v) < \Delta_w(G)$ and G is connected, then $\chi_w(G) \le \Delta_w(G)$ as proved in [11, 13, 14] (this also follows from Theorem 2.3). Thus we immediately get from Theorem 4.1 the following.

Theorem 4.3. *Let G be a connected graph and let w be a function which assigns to the edges of G positive weights. If $\chi_w(G) = \Delta_w(G) + 1$, then the weighted degree of each vertex of G is equal to $\Delta_w(G)$ and one of the following holds:*

- *G is an odd cycle and all its edges have the same weights.*
- *G is a complete graph and all its edges have the same weights.*
- *G is a Gallai tree with at least two blocks.*

There really exist Gallai trees such that $\chi_w(G) = \Delta_w(G)+1$ as shown in Proposition 4.4. On the other hand, there are also Gallai trees such that there is no function w for which $\chi_w(G) = \Delta_w(G) + 1$. It is possible to restate Theorem 4.3 in the "if and only if" form by adding a condition that the set of colors $[1, \Delta_w]$ in the third case can be partitioned into lists L_i for each of the blocks in the way described in Theorem 4.1. By Corollary 4.2, pairs of Gallai trees G and weight functions w for which $\chi_w(G) = \Delta_w(G) + 1$ can be recognized in polynomial time.

Proposition 4.4. *There exists a connected graph G and a function w which assigns to the edges of G positive weights such that $\chi_w(G) = \Delta_w(G) + 1$ and G is not 2-connected.*

Proof. Let $1 \le a < b$ and $2 \le n$ be fixed integers. Let G_i be a complete graph on $n + 1$ vertices and let v_i be one of its vertices for $1 \le i \le n$. We assign the weight b to the edges of G_i which are not adjacent to v_i and the weight a to the edges which are adjacent to v_i. We further form a complete graph on the vertices $v_1, \dots, v_n$ and we assign the edges of this graph the weights equal to $b - a$. It is easy to check that the weighted degree of each of the vertices is equal to $a + (n - 1)b$ and the minimal span of this channel assignment problem is $a + (n - 1)b + 1$ by Theorem 4.1 (we use

Theorem 4.1 with the set $[1, a + (n-1)b]$ assigned to all the vertices). □

We remark that the construction of Proposition 4.4 can be extended to Gallai trees in which some blocks are odd cycles and the structure is more complex. But we do not have a characterization of Gallai trees for which there exists a weight function w such that $\chi_w(G) = \Delta_w(G) + 1$ (recall that the input of the algorithm from Corollary 4.2 is a pair G and w), so this leads to the following problem.

PROBLEM 1. *For which Gallai trees G does there exist a weight function w such that $\chi_w(G) = \Delta_w(G) + 1$?*

4.2. $L(2,1)$-labeling of graphs.

The definition of an $L(2,1)$-labeling was provided in section 1. If G is a graph, an $L(2,1)$-labeling of G is a solution of a channel assignment problem for G^2 with a weight function w, where G^2 is the second power of G and w is the weight function which assigns the weight 2 to the edges of G and the weight 1 to the edges of G^2 which are not edges of G. The second power of the graph is the graph with the same vertex set where vertices u and v are joined by an edge if their distance in G is at most two. The only difference between $L(2,1)$-labeling and a channel assignment problem is that here the color 0 can be used in an $L(2,1)$-labeling. We state the following theorem, thus improving a bound of [4].

THEOREM 4.5. *Let G be a graph with maximum degree $\Delta \geq 2$. Then there exists an $L(2,1)$-labeling of G using numbers $0, \ldots, \Delta^2 + \Delta - 1$.*

Proof. Assume that G is connected and let w be the weight function for G^2 introduced in the beginning of this subsection. We want to prove that $\chi_w(G^2) \leq \Delta^2 + \Delta$. It is easy to see that the maximum weighted degree Δ_w of the corresponding channel assignment problem is at most $\Delta^2 + \Delta$. If $\chi_w(G^2) \geq \Delta^2 + \Delta + 1$, then $\Delta_w = \Delta^2 + \Delta$, and G^2 with w is of the form described in Theorem 4.3. Since G^2 is 2-connected (a second power of a connected graph with maximum degree at least 2 is always 2-connected), it cannot be a Gallai tree with 2 or more blocks. So G^2 is either a cycle or a complete graph. Except for P_3 and K_3, there is no graph G whose second power is a cycle. Both P_3 and K_3 have an $L(2,1)$-labeling using integers $0, \ldots, 5$.

The remaining case is that G^2 is a complete graph. By Theorem 4.3, if $\chi_w(G^2) = \Delta_w + 1$, all the weights of the edges are the same, and hence they all are equal to two. Therefore, G is a complete graph. Let n be the number of vertices of G. Then $\chi_w(G^2) = 2n - 1$, $\Delta = n - 1$, and $\Delta^2 + \Delta = n^2 - n$. The fact $\Delta \geq 2$ yields $n \geq 3$. Since $2n - 1 \leq n^2 - n$ for $n \geq 3$, we have proved $\chi_w(G^2) \leq \Delta^2 + \Delta$. □

REFERENCES

[1] O. V. BORODIN, *Criterion of chromaticity of a degree prescription*, in Abstracts of Fourth All-Union Conf. on Theoretical Cybernetics, Novosibirsk, Russia, 1977, pp. 127–128 (in Russian).

[2] O. V. BORODIN, *Problems of Colouring and of Covering the Vertex Set of a Graph by Induced Subgraphs*, Ph.D. thesis, Novosibirsk State University, Novosibirsk, Russia, 1979 (in Russian).

[3] R. L. BROOKS, *On colouring the nodes of a network*, Proc. Cambridge Philos. Soc., 37 (1941), pp. 194–197.

[4] G. J. CHANG AND D. KUO, *The $L(2,1)$-labeling problem on graphs*, SIAM J. Discrete Math., 9 (1996), pp. 309–316.

[5] P. ERDÖS, A. L. RUBIN, AND H. TAYLOR, *Choosability in graphs*, in Proc. West Coast Conf. on Combinatorics, Graph Theory and Computing, Congress. Numer. XXVI, P. Z. Chinn and D. McCarthy, eds., Utilitas Mathematica Publishing, Inc., Winnipeg, Manitoba, Canada, 1980, pp. 125–157.

[6] T. GALLAI, *Kritische Graphen* I, Publ. Math. Inst. Hung. Acad. Sci. Ser. A, 8 (1963), pp. 373–395.

[7] J. R. Griggs and R. K. Yeh, *Labelling graphs with a condition at distance* 2, SIAM J. Discrete Math., 5 (1992), pp. 586–595.

[8] J. Kratochvíl, Z. Tuza, and M. Voigt, *Brooks-type theorems for choosability with separation*, J. Graph Theory, 27 (1998), pp. 43–49.

[9] J. Kratochvíl, Z. Tuza, and M. Voigt, *New trends in the theory of graph colorings: Choosability and list coloring*, in Contemporary Trends in Discrete Mathematics (from DIMACS and DIMATIA to the Future), DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 49, R. L. Graham et al., eds., AMS, Providence, RI, 1999, pp. 183–197.

[10] L. Lovasz, *Three short proofs in graph theory*, J. Combin. Theory Ser. B, 19 (1975), pp. 269–271.

[11] C. McDiarmid, *Bounds for the Span in Channel Assignment Problems*, talk presented at the 18th British Combinatorial Conference, University of Sussex, Falmer, Brighton, UK, July 2001.

[12] C. McDiarmid, *Discrete mathematics and radio channel assignment*, in Recent Advances in Algorithms and Combinatorics, CMS Books Math./Ouvrages Math. SMC 11, C. Linhares-Salas and B. Reed, eds., Springer, New York, 2003, pp. 27–63.

[13] C. McDiarmid, *On the span in channel assignment problems*, in Abstracts of Talks Presented at the Eighth Midsummer Combinatorial Workshop, P. Smolikova, ed., KAM-DIMATIA Series 2002-561, Charles University, Prague, 2001, p. 11.

[14] C. McDiarmid, *On the span in channel assignment problems: Bounds, computing and counting*, Discrete Math., 266 (2003), pp. 387–397.

[15] M. Molloy and B. Reed, *Graph Colouring and the Probabilistic Method*, Algorithms and Combinatorics 23, Springer-Verlag, Berlin, 2002.

[16] Z. Tuza, *Graph colorings with local constraints—a survey*, Discuss. Math. Graph Theory, 17 (1997), pp. 161–228.

[17] V. G. Vizing, *Colouring the vertices of a graph with prescribed colours*, Metody Diskret. Anal. V Teorii Kodov i Shem., 29 (1976), pp. 3–10 (in Russian).

SIAM J. Discrete Math.
Vol. 16, No. 3, pp. 438–448

LINE GRAPHS OF HELLY HYPERGRAPHS[*]

YURY METELSKY[†] AND REGINA TYSHKEVICH[†]

Abstract. A natural generalization of the notion of domino introduced and investigated in [T. Kloks, D. Kratsch, and H. Müller, *Dominoes*, Lecture Notes in Comput. Sci. 903, Springer-Verlag, Berlin, 1995, pp. 106–120] is considered. A graph is called an r-*mino* if each of its vertices belongs to at most r maximal cliques. The class of r-minoes is denoted $\mathcal{M}_r$. Thus $\mathcal{M}_2$ is the class of dominoes. It is shown that $\mathcal{M}_r$ coincides with the class of line graphs of Helly hypergraphs with rank at most r. For an arbitrary r, the existence of a finite list of forbidden induced subgraphs characterizing $\mathcal{M}_r$ is proved. An explicit finite characterization is given for $\mathcal{M}_3$. An r-mino is called *linear* if each of its edges belongs to exactly one maximal clique. We prove that the GRAPH 3-COLORABILITY problem remains NP-complete when restricted to linear dominoes with vertex degrees at most 4.

Key words. graph, hypergraph, r-mino, line graph of hypergraph, Helly hypergraph, rank of hypergraph, clique, r-covering, induced subgraph, finite characterization

AMS subject classifications. 05C62, 05C75, 05C70, 05C65

PII. S089548019936521X

1. Introduction.

This paper is inspired by the work of Kloks, Kratsch, and Müller [6], in which a class of graphs called dominoes is introduced and investigated. A graph is called a *domino* if each of its vertices belongs to at most two maximal cliques. Taking r instead of 2, one can introduce the notion of an r-*mino*. Obviously, each graph is an r-mino for appropriate r. Therefore the set of graphs is the union of the strictly increasing sequence

$$\mathcal{M}_1 \subset \mathcal{M}_2 \subset \cdots \subset \mathcal{M}_n \subset \cdots,$$

where $\mathcal{M}_r$ is the class of r-minoes. These classes are investigated in the paper.

Two characterizations of the class $\mathcal{M}_r$ are given in sections 3 and 4. The first one states that $\mathcal{M}_r$ is the class of line graphs of Helly hypergraphs with rank at most r. In particular, $\mathcal{M}_r$ is a *hereditary class*; i.e., an induced subgraph of a graph in $\mathcal{M}_r$ is also in $\mathcal{M}_r$. It is well known that every hereditary class of graphs $\mathcal{P}$ can be characterized by means of a list of forbidden induced subgraphs. If $\mathcal{F}$ is such a list, then we write $\mathcal{P} = Forb(\mathcal{F})$. If, in addition, $\mathcal{F}$ is finite, then we call it a *finite characterization* of $\mathcal{P}$. In this paper we recursively define a finite characterization $\mathcal{F}_r$ of the class $\mathcal{M}_r$ (for every $r \geq 1$). It follows from the existence of a finite characterization that for every fixed r there is a polynomial time algorithm for determining if a graph is in $\mathcal{M}_r$.

The following circumstances increase our interest in the class $\mathcal{M}_r$.

For a fixed constant r, let $\mathcal{L}_r$ be the class of line graphs of hypergraphs with rank at most r. A nontrivial characterization of the class is known only for $r \leq 2$ (see Bermond and Meyer [4]). Poljak, Rödl, and Turzik [9] proved that the problem of determining if a graph belongs to $\mathcal{L}_r$ is NP-complete for an arbitrary $r \geq 3$. Moreover, they proved that the similar problem remains NP-complete for every fixed $r \geq 4$. Lovász [7] posed the problem of characterizing the class $\mathcal{L}_3$. The question of

[*] Received by the editors December 3, 1999; accepted for publication (in revised form) November 18, 2002; published electronically May 22, 2003. This work was supported by INTAS and the Belarus Government under Project INTAS-BELARUS 97-0093.

http://www.siam.org/journals/sidma/16-3/36521.html

[†] Department of Mechanics and Mathematics, Belarus State University, av. F. Skoriny 4, Minsk 220050 Belarus (metelsky@bsu.by, tyshkevich@bsu.by).

whether or not the class $\mathcal{L}_3$ can be recognized in polynomial time is still open, but recognizing line graphs of simple hypergraphs with rank at most 3 is NP-complete as well [9].

Consider the following two graph-theoretic invariants:

- the *rank-dimension* $\mathrm{rd}(G) = \min\{r : G \in \mathcal{L}_r\}$,
- the *Helly rank-dimension* $\mathrm{hd}(G) = \min\{r : G \in \mathcal{M}_r\}$.

Our characterization of the class $\mathcal{M}_r$ below implies the strict inclusion $\mathcal{M}_r \subset \mathcal{L}_r$, so $\mathrm{rd}(G) \leq \mathrm{hd}(G)$. However, the difference $\mathrm{hd}(G) - \mathrm{rd}(G)$ can be arbitrarily large. The complexity of determining $\mathrm{hd}(G)$ is still unknown.

Associate with an arbitrary graph G the independence system S_G on the vertex set $V(G)$ whose independent sets are exactly the stable sets of G. Like any independence system, S_G can be represented as the intersection of matroids. Tyshkevich and Urbanovich [10] proved that the minimal number of matroids in such a representation is precisely the equivalence covering number $eq(G)$. The class of monominoes $\mathcal{M}_1$ consists of graphs with $eq(G) = 1$ (see Benzaken and Hammer [1]). Section 4 also contains a finite characterization of the class $\mathcal{M}_2$ already obtained both in [10] in a different context (when investigating the class of graphs with $eq(G) \leq 2$, which is the class of line graphs of bipartite multigraphs and is contained in $\mathcal{M}_2$) and in [6].

An r-mino is called *linear* if each of its edges belongs to exactly one maximal clique. For a linear r-mino G, the list of maximal cliques and $\mathrm{hd}(G)$ ($= \mathrm{rd}(G)$ in this situation) can be found in polynomial time. A finite characterization of linear r-minoes is given in section 5. Here we also prove that the GRAPH 3-COLORABILITY problem remains NP-complete when restricted to linear dominoes with vertex degrees at most 4. Thus we answer the question posed in [6].

Section 6 contains a finite characterization of the class $\mathcal{M}_3$.

2. Background.

The vertex set of a graph G is denoted $V(G)$. If $N(v) = N_G(v)$ is the *neighborhood* of a vertex v in G, then $N[v] = N(v) \cup \{v\}$. Let $G(X)$ denote the subgraph of G induced by a set $X \subseteq V(G)$; to simplify the notation we write $G(x_1, x_2, \ldots, x_k)$ instead of $G(\{x_1, x_2, \ldots, x_k\})$.

We consider only finite hypergraphs in which every vertex is contained in some edge. For a hypergraph H with the incidence matrix M, the *dual hypergraph* H^* is the hypergraph with the transposed incidence matrix M^t. The *line graph* $L(H)$ is defined as follows: the vertices of $L(H)$ are in a bijective correspondence with the edges of H, and two vertices are adjacent in $L(H)$ if and only if the corresponding edges intersect in H. If $L(H) \cong G$, then H is a *root* of G.

Berge [2] described all roots for an arbitrary graph in terms of clique coverings. A set C of pairwise adjacent vertices of a graph is called a *clique*. A *maximal* clique is maximal with respect to inclusion. A finite family $Q = \{C_i : i \in I\}$ of cliques of a graph G is called a *clique covering* if every vertex as well as every edge of G is contained in some C_i. The cliques C_i are the *clusters* of Q. For an arbitrary clique covering $Q = \{C_i : i \in I\}$ of G, define the hypergraph $H(Q)$ as follows: the vertices of $H(Q)$ are just the vertices of G, and the edges are the clusters of Q. The edges C_i and C_j are different for $i \neq j$ even if the sets C_i and C_j coincide. The dual hypergraph $H(Q)^* = C(Q)$ is called the *canonical hypergraph*.

THEOREM 2.1 (see [2]). *The roots of a graph G are exactly the canonical hypergraphs $C(Q)$, where Q runs over all clique coverings of G.*

Let $\mathcal{P}$ be a *hypergraph-theoretic property*, i.e., a class of hypergraphs distinguished up to isomorphism. We say that a clique covering Q of a graph G *has the property* $\mathcal{P}$

if $H(Q) \in \mathcal{P}$. Put

$$\mathcal{L}(\mathcal{P}) = \{L(H) : H \in \mathcal{P}\}, \quad \mathcal{P}^* = \{H^* : H \in \mathcal{P}\}.$$

Theorem 2.1 immediately implies the following corollary.

Corollary 2.2. *Let $\mathcal{P}$ be an arbitrary hypergraph-theoretic property, and let G be a graph. Then $G \in \mathcal{L}(\mathcal{P})$ if and only if G has a clique covering with the property $\mathcal{P}^*$.*

A hypergraph is called *linear* if every pair of edges has at most one common vertex. A clique covering Q of a graph is called *linear* if $H(Q)$ is linear. A clique covering is called an *r-covering* if each vertex of the graph belongs to at most r clusters.

The rank of a hypergraph is the maximum size of its edges.

Let $\mathcal{P}_r$ be the class of hypergraphs of rank at most r, and let $\mathcal{P}_r^l$ be the class of linear hypergraphs in $\mathcal{P}_r$. Putting

$$\mathcal{L}_r = \mathcal{L}(\mathcal{P}_r), \quad \mathcal{L}_r^l = \mathcal{L}(\mathcal{P}_r^l),$$

we immediately obtain Corollaries 2.3 and 2.4.

Corollary 2.3 (see [2]). *$G \in \mathcal{L}_r$ if and only if G has an r-covering.*

Corollary 2.4 (Berge [3]). *$G \in \mathcal{L}_r^l$ if and only if G has a linear r-covering.*

A hypergraph whose edge sizes are all equal to r is called *r-uniform*.

Note. Obviously, the class $\mathcal{L}_r$ is exactly the set of line graphs of r-uniform hypergraphs.

A hypergraph H is a *Helly hypergraph* if the family $\mathcal{E}$ of its edges satisfies the following *Helly condition*: for each subfamily $\mathcal{E}' \subseteq \mathcal{E}$ of pairwise intersecting edges, there is a vertex which belongs to all edges in $\mathcal{E}'$.

The 2*-section graph* $(H)_2$ of a hypergraph H is the graph whose vertices are the vertices of H, and two vertices are adjacent if and only if they belong to the same edge of H. A hypergraph H is called *conformal* if each clique of $(H)_2$ is contained in some edge of H.

Lemma 2.5 (see [2]). *A hypergraph H is a Helly hypergraph if and only if H^* is conformal.*

A clique covering Q of a graph G is called *conformal* if the hypergraph $H(Q)$ is conformal.

Lemma 2.6. *For any graph G, the set of maximal cliques is the unique minimal (with respect to inclusion) conformal covering of G.*

Proof. Let Q be a clique covering of G and $H = H(Q)$. Obviously, $(H)_2 = G$. Therefore H is conformal if and only if each maximal clique of G is an edge of H, i.e., a cluster of Q. □

Corollary 2.7 (McKee and McMorris [8]). *Each graph is the line graph of a Helly hypergraph.*

3. r-minoes. We say that a graph G is an *r-mino* if each vertex of G belongs to at most r maximal cliques. If, in addition, every edge of G belongs to exactly one maximal clique, then we have a *linear r-mino*. Denote $\mathcal{M}_r$ and $\mathcal{M}_r^l$ the sets of r-minoes and of linear r-minoes, respectively.

Theorem 3.1. *The following statements hold:*

(i) *$\mathcal{M}_r$ coincides with the set of line graphs of Helly hypergraphs with rank at most r.*

(ii) *$\mathcal{M}_r^l$ coincides with the set of line graphs of linear Helly hypergraphs with rank at most r.*

Proof. If $G \in \mathcal{M}_r$, then the set Q of maximal cliques is an r-covering of G. By Lemma 2.6, Q is conformal. Then the canonical hypergraph $C(Q)$ is a Helly hypergraph by Lemma 2.5, and $\operatorname{rank} C(Q) \leq r$. By Theorem 2.1, $L(C(Q)) \cong G$.

Conversely, let H be a Helly hypergraph, where $\operatorname{rank} H \leq r$ and $L(H) \cong G$. Then, by Corollary 2.2 and Lemma 2.5, there is a conformal r-covering of G. By Lemma 2.6, this covering contains all maximal cliques of G. Therefore the set of maximal cliques is also an r-covering of G. Thus, $G \in \mathcal{M}_r$ and (i) is proved.

Obviously, the linearity condition is self-dual, so (ii) holds. □

COROLLARY 3.2 (see [6]). *The following statements hold:*

(i) $\mathcal{M}_2$ *coincides with the set of line graphs of multigraphs without triangles.*

(ii) $\mathcal{M}_2^l$ *coincides with the set of line graphs of simple graphs without triangles.*

COROLLARY 3.3. $\mathcal{M}_r$ *and* $\mathcal{M}_r^l$ *are hereditary graph classes.*

Proof. The property "to be a Helly hypergraph with rank at most r" is hereditary with respect to deleting edges. □

COROLLARY 3.4.

(i) *For a constant* $r \geq 2$*, the following inclusion holds:*

$$\mathcal{M}_r \subset \mathcal{L}_r. \tag{3.1}$$

(ii) *For a graph* G*,*

$$\operatorname{rd}(G) \leq \operatorname{hd}(G). \tag{3.2}$$

Equality in (3.2) *is possible, but the difference*

$$\operatorname{hd}(G) - \operatorname{rd}(G) \tag{3.3}$$

can be arbitrarily large.

Proof. The inclusion (3.1) follows immediately from Theorem 3.1. For the $(r+1)$-vertex wheel W_r, where $r > 3$ (see the graph W_4 in Figure 4.1), we have

$$\operatorname{hd}(W_r) = r, \quad \operatorname{rd}(W_r) = \lceil r/2 \rceil .$$

Hence, the inclusion (3.1) is strict and the difference (3.3) can be arbitrarily large. The inclusion (3.1) implies (3.2) immediately. Finally, for the star $K_{1,r}$,

$$\operatorname{rd}(K_{1,r}) = r = \operatorname{hd}(K_{1,r}). \qquad □$$

4. Forbidden induced subgraph characterizations.

We write $a \sim b$ ($a \not\sim b$) if the vertices a and b are adjacent (respectively, nonadjacent) in a graph G. If $A, B \subseteq V(G)$, then $A \sim B$ ($A \not\sim B$) means that $a \sim b$ ($a \not\sim b$) for all $a \in A$, $b \in B$. Denote $s(G)$ the number of maximal cliques of G.

THEOREM 4.1. *For any constant* r*, there exists a finite characterization of the class* $\mathcal{M}_r$*.*

Proof. The proof is by induction on r. It is obvious that $\mathcal{M}_1 = Forb(\{P_3\})$; i.e., one can assume that $\mathcal{F}_1 = \{P_3\}$.

Now let $r \geq 2$. By hypothesis, $\mathcal{M}_{r-1} = Forb(\mathcal{F}_{r-1})$ and $\mathcal{F}_{r-1}$ is finite. Without loss of generality, suppose that the characterization $\mathcal{F}_{r-1}$ is minimal. Evidently, if $G \in \mathcal{F}_{r-1}$, then G has a unique dominating vertex $v(G)$ and is the union of $s(G)$ maximal complete subgraphs, $s(G) \geq r$.

We shall construct the list $\mathcal{F}_r$, which is a finite characterization of the class $\mathcal{M}_r$. Any graph G satisfying the conditions

$$G \in \mathcal{F}_{r-1}, \quad s(G) \geq r + 1,$$

is included in the list $\mathcal{F}_r$. Next, let

$$G \in \mathcal{F}_{r-1}, \quad s(G) = r, \quad v(G) = v, \tag{4.1}$$

and let

$$C_1, C_2, \ldots, C_r \tag{4.2}$$

be the list of maximal cliques of G.

Define two supergraphs F and H for the graph G as follows:

(1) $G = F - A$, where A is a clique in F, $1 \leq |A| \leq r$, $A \sim v$, $s(F) \geq r+1$.

(2) $G = H - a - b$, $a \not\sim b$, $a \sim C_i$, $b \sim C_i$ for some index i.

The remaining adjacencies in the graphs F and H can be arbitrary. Thus, conditions (1) and (2) define the sets of graphs of two types, F and H. For every graph G satisfying the conditions in (4.1) we now include all graphs of type F and H into $\mathcal{F}_r$. The list $\mathcal{F}_r$ is constructed.

If m_i is the maximal order of the graphs in $\mathcal{F}_i$, then $m_r \leq m_{r-1} + r$. Hence, the set $\mathcal{F}_r$ is finite.

We shall prove that

$$\mathcal{M}_r = Forb(\mathcal{F}_r). \tag{4.3}$$

Obviously, $\mathcal{F}_r \cap \mathcal{M}_r = \emptyset$ since the vertex $v(G)$ of any graph G in $\mathcal{F}_r$ belongs to at least $r+1$ maximal cliques. The induced subgraphs of the graphs in $\mathcal{M}_r$ do not belong to the list $\mathcal{F}_r$ since they belong to $\mathcal{M}_r$. Hence

$$\mathcal{M}_r \subseteq Forb(\mathcal{F}_r).$$

Now let $B \in Forb(\mathcal{F}_r)$. Consider the subgraph $C = B(N[u])$ for a vertex $u \in V(B)$. Each clique of B containing u is a clique of C. If $C \in Forb(\mathcal{F}_{r-1})$, then u belongs to at most $r-1$ maximal cliques of C. If some graph G in $\mathcal{F}_{r-1}$ is an induced subgraph of C, then G satisfies the conditions in (4.1). Without loss of generality, suppose that $u = v(G) = v$. Let (4.2) be the list of maximal cliques of G. Every clique C_i is contained in one maximal clique D_i of C, $i = 1, \ldots, r$, since B has no induced subgraphs of type H.

It remains to prove that

$$D_1, D_2, \ldots, D_r \tag{4.4}$$

is the complete list of maximal cliques of C. Let D be an arbitrary clique of C, $v \in D$, $|D \backslash v| = k$. Then

$$D \subseteq D_i \tag{4.5}$$

for some index i. In fact, (4.5) holds for $k \leq r$ since C contains no induced subgraphs of type F. Let $k \geq r+1$, and let (4.5) hold if $|D \backslash v| < k$. Consider $(k-1)$-subsets of the clique $D \backslash v$. The number of these subsets is equal to k, and each one is contained in a corresponding clique D_i. Since $k > r$, then there exist two such subsets, for instance X_1 and X_2, both contained in the same clique D_i. But $D \backslash v = X_1 \cup X_2$. Therefore $D \subseteq D_i$. So (4.4) is the complete list of maximal cliques of the graph B that contain the vertex u. Hence $B \in \mathcal{M}_r$, and equality (4.3) is proved. □

Applying the recursive procedure from Theorem 4.1 to the list $\mathcal{F}_1 = \{P_3\}$, one can easily obtain the following theorem.

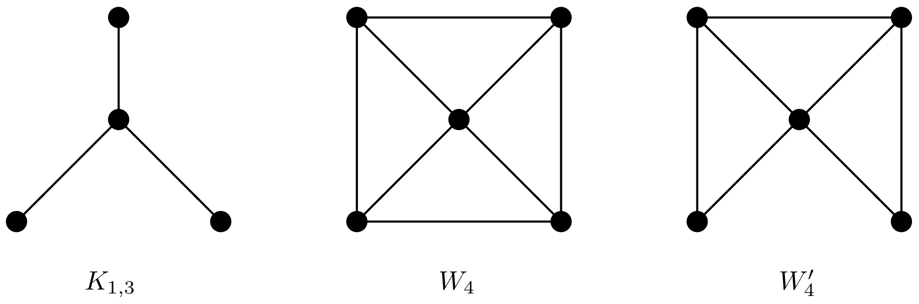


Fig. 4.1. *A finite characterization of the class $\mathcal{M}_2$.*

Theorem 4.2 (see [10], [6]). $\mathcal{M}_2 = Forb(K_{1,3}, W_4, W_4')$ (Figure 4.1).

Note. In this paper a finite characterization of the class $\mathcal{M}_3$ is also presented. In this case applying the recursive procedure from Theorem 4.1 to the list of Theorem 4.2 is rather tedious. In section 6 this characterization is obtained in a different way.

Adding all odd simple cycles of length at least 5 to the list in Figure 4.1, we obtain the following characterization of the class $\mathcal{L}(2)$ of line graphs of bipartite multigraphs.

Corollary 4.3 (see [10]). $\mathcal{L}(2) = Forb(K_{1,3}, W_4, W_4', C_{2n+1} : n \geq 2)$.

It is proved in [10] that $\mathcal{L}(2)$ is exactly the class of graphs with condition $eq(G) \leq 2$.

Adding all simple cycles of length at least 4 to the list in Figure 4.1, we obtain the following characterization of the class $\mathcal{C}h_2$ of chordal dominoes.

Corollary 4.4 (see [6]). *The following statements hold:*

(i) *$\mathcal{C}h_2$ coincides with the class of line graphs of acyclic multigraphs.*

(ii) $\mathcal{C}h_2 = Forb(K_{1,3}, W_4, W_4', C_n : n \geq 4)$.

5. Linear r-minoes.

Theorem 5.1. *The list of maximal cliques and the Helly rank-dimension of a graph can be found in polynomial time for the class of linear r-minoes.*

Proof. Let G be a linear r-mino, $v \in V(G)$, and let A be a connected component of the subgraph $G(N(v))$. Then $A \cup \{v\}$ is a maximal clique of G, and each maximal clique can be obtained analogously.

Furthermore, $\mathrm{hd}(G)$ is the maximal number of connected components of the graphs $G(N(v))$ for all $v \in V(G)$. □

Theorem 5.2. $\mathcal{M}_r^l = Forb(K_{1,r+1}, K_4 - e)$.

Proof. Obviously, every edge of a graph belongs to exactly one maximal clique if and only if this graph is $(K_4 - e)$-free.

Now let G be a $(K_4 - e)$-free graph, and let its vertex v belong to exactly p maximal cliques

$$C_1, C_2, \ldots, C_p.$$

Taking an arbitrary vertex $v_i \neq v$ in C_i, $i = 1, \ldots, p$, we obtain the induced star

$$G(v, v_1, \ldots, v_p) = K_{1,p}. \qquad □$$

Corollary 5.3. *For a linear r-mino G,*

$$\mathrm{hd}(G) = \mathrm{rd}(G) = \max\{p : G \text{ contains an induced } K_{1,p}\}.$$

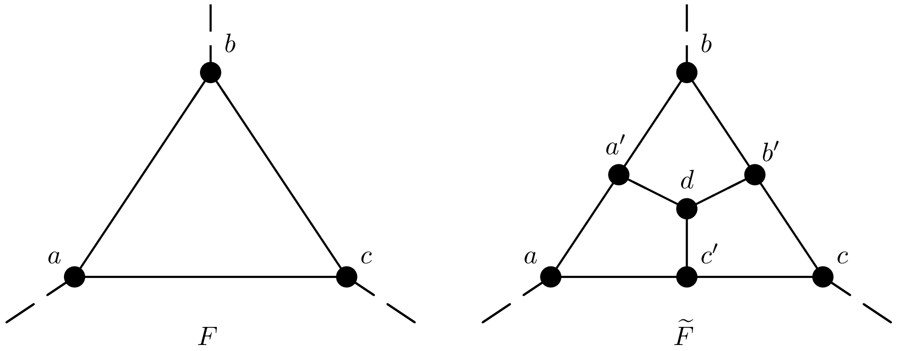


Fig. 5.1. *Graphs F and $\widetilde{F}$.*

Now consider linear dominoes. Denote $\chi(G)$ and $\chi'(G)$ the chromatic number and the chromatic index (the edge chromatic number) of a graph G, respectively.

Theorem 5.4. *The decision problem "$\chi(G) \le 3$" is NP-complete for linear dominoes G with $\Delta(G) \le 4$.*

Proof. First we consider the following two decision problems:

$$\chi'(G) \le 3 \text{ for a graph } G \text{ with } \Delta(G) \le 3. \tag{5.1}$$

$$\chi'(G) \le 3 \text{ for a triangle-free graph } G \text{ with } \Delta(G) \le 3. \tag{5.2}$$

Holyer [5] proved that the problem (5.1) is NP-complete. We shall show that the problem (5.1) can be reduced to the problem (5.2) in polynomial time; i.e., the problem (5.2) is NP-complete.

Let F be a graph with $\Delta(F) \le 3$. Consider a triangle with the vertex set $\{a, b, c\}$ in F. Replace this triangle in F by the 7-vertex graph shown in Figure 5.1. Denote the resulting graph $\widetilde{F}$.

The graph $\widetilde{F}$ has fewer triangles than F. Obviously, the implication

$$\chi'(F) \le 3 \;\Rightarrow\; \chi'(\widetilde{F}) \le 3$$

is true (see Figure 5.2).

Conversely, let $\chi'(\widetilde{F}) \le 3$. Fix a proper 3-coloring φ of the edges of $\widetilde{F}$. Associate with the vertex a the 2-element set $\{a_1, a_2\}$ of colors of the edges aa', ac'. Define the sets $\{b_1, b_2\}$ and $\{c_1, c_2\}$ for the vertices b and c analogously. The correspondence

$$x \;\mapsto\; \{x_1, x_2\}, \quad x = a, b, c,$$

is injective. Indeed, suppose $\{a_1, a_2\} = \{b_1, b_2\}$ and, without loss of generality,

$$\varphi(aa') = \varphi(bb') = 1, \quad \varphi(ac') = \varphi(a'b) = 2.$$

Then $\varphi(a'd) = 3$, $\varphi(b'd) = 2$, $\varphi(c'd) = 1$. We have $\varphi(cc') = \varphi(b'c) = 3$, a contradiction.

Without loss of generality, let

$$\{a_1, a_2\} = \{1, 2\}, \quad \{b_1, b_2\} = \{1, 3\}, \quad \{c_1, c_2\} = \{2, 3\}.$$

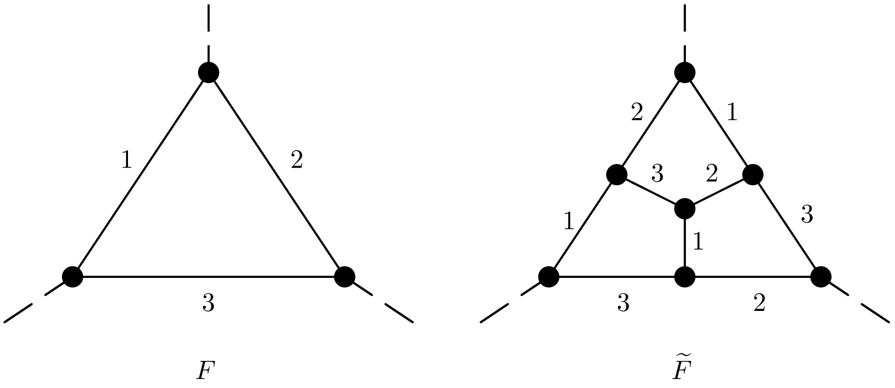


FIG. 5.2. *The proof of an implication in Theorem* 5.4.

Put

$$\varphi(ab) = 1, \quad \varphi(bc) = 3, \quad \varphi(ac) = 2.$$

The colors of all the other edges in F are the same as in $\widetilde{F}$. Thus the implication

$$\chi'(\widetilde{F}) \leq 3 \;\Rightarrow\; \chi'(F) \leq 3$$

is true as well.

Applying the transformation above, we eliminate all triangles in F one after another. Denote the resulting graph H. If the graph F has no triangles, then put $H = F$.

Obviously, the correspondence $F \mapsto H$ is a polynomial reduction of the problem (5.1) to the problem (5.2).

Now we shall give a polynomial reduction of the NP-complete problem (5.2) to the problem in the assertion of the theorem. Let H be a triangle-free graph with $\Delta(H) \leq 3$ and $G = L(H)$. We have

$$\chi'(H) = \chi(G), \quad \Delta(G) \leq 4.$$

Moreover, G is a linear domino by Corollary 3.2. Obviously, the correspondence $H \mapsto G$ is the required polynomial time reduction. $\square$

6. Determining $\mathcal{F}_3$.

THEOREM 6.1. *$\mathcal{M}_3 = Forb(\{G_1, \dots, G_8\})$, where the graphs G_i are shown in Figure* 6.1.

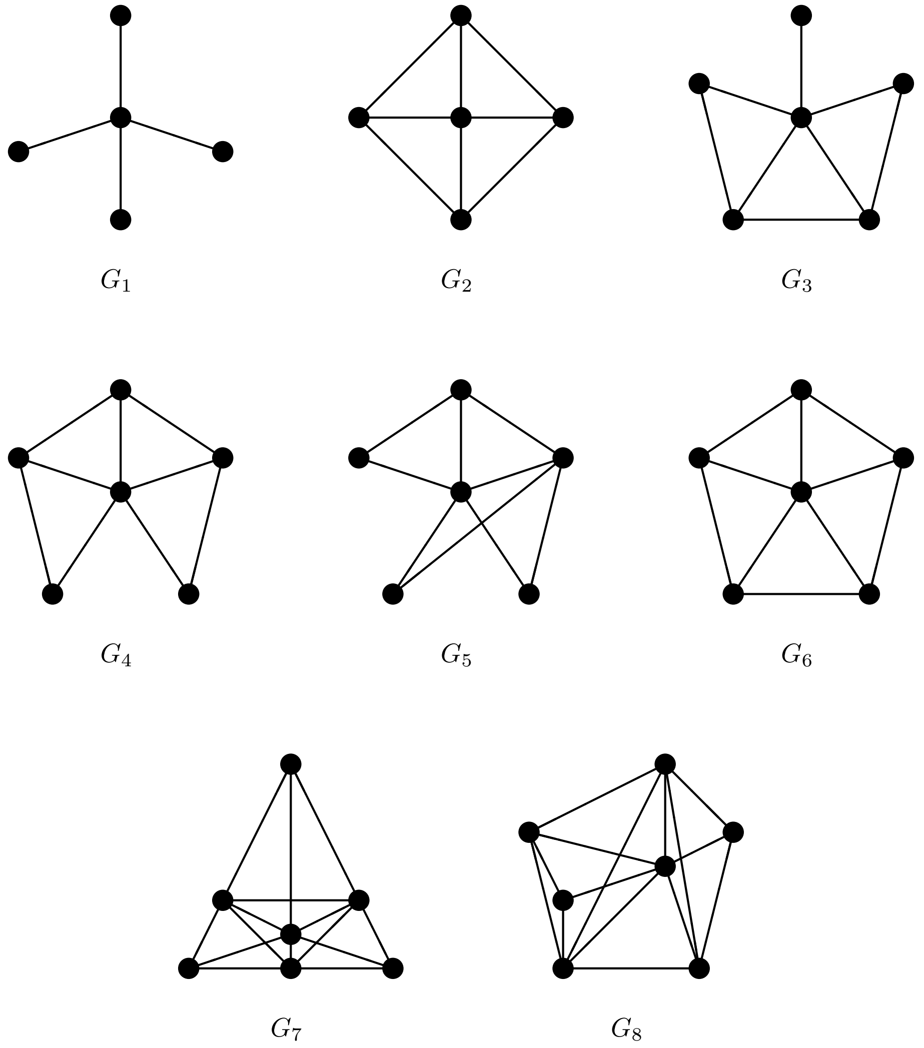
Proof. One can easily see that

$$\mathcal{M}_3 \subseteq Forb(\{G_1, \dots, G_8\}).$$

Now let $G \in Forb(\{G_1, \dots, G_8\})$. Take an arbitrary $u \in V(G)$ such that $H = G(N(u))$ is not a complete graph. Without loss of generality, suppose that

$$H \text{ has no dominating vertices.} \tag{6.1}$$

Put $G'_i = G_i - v(G_i)$, where $v(G_i)$ is the unique dominating vertex of G_i.



G_1 G_2 G_3

G_4 G_5 G_6

G_7 G_8

FIG. 6.1. *A finite characterization of the class $\mathcal{M}_3$.*

Case 1. Suppose H contains three pairwise nonadjacent vertices x, y, z. Then

$$V(H) = N[x] \cup N[y] \cup N[z], \tag{6.2}$$

$$X = N[x],\ Y = N[y],\ Z = N[z] \text{ are cliques in } H. \tag{6.3}$$

The equality (6.2) holds since H does not contain an induced G'_1. Suppose there exist $x_1, x_2 \in N(x)$ such that $x_1 \not\sim x_2$. The graphs $H(x, x_1, x_2, y)$ and $H(x, x_1, x_2, z)$ are not isomorphic to G'_2, and $H(x_1, x_2, y, z)$ is not empty. Therefore $H(x, x_1, x_2, y, z) \cong G'_3, G'_4$, or G'_5, a contradiction.

Thus, (6.2) and (6.3) hold for any pairwise nonadjacent $x, y, z \in V(H)$. Let K

be a maximal clique in H, $K \neq X, Y, Z$. For any $a, b \in K$, we have

$$(6.4) \qquad a, b \in X, \qquad a, b \in Y, \quad \text{or} \quad a, b \in Z.$$

In fact, if $a \in X \backslash Y$, $b \in Y \backslash X$, $a \notin Z$, then a, y, z are pairwise nonadjacent in H. By (6.3), $N[a]$ is a clique. But $x, b \in N[a]$ and $x \not\sim b$, a contradiction.

By (6.1) and (6.3), each vertex of K belongs to exactly two of the cliques X, Y, Z. Let $a \in K$ and $a \in (X \cap Y) \backslash Z$. Since K is maximal, there exist $b, c \in K$ such that $b \in (X \cap Z) \backslash Y$, $c \in (Y \cap Z) \backslash X$. We have $H(a, b, c, x, y, z) \cong G_7'$, a contradiction.

Thus, H has no maximal cliques different from X, Y, Z, i.e., $s(H) = 3$.

Case 2. H does not contain three pairwise nonadjacent vertices. Let x and y be two nonadjacent vertices of H.

Clearly, $N[x] \cup N[y] = V(H)$. Set

$$A = N(x) \backslash N(y), \quad B = N(y) \backslash N(x), \quad C = N(x) \cap N(y).$$

Observe that A, B, C are cliques. In fact, $A \cup \{y\}$ and $B \cup \{x\}$ do not contain three pairwise nonadjacent vertices, and $H(C \cup \{x, y\})$ has no induced G_2'.

Without loss of generality, assume that no two of the sets A, B, C are empty. Otherwise $s(H) \leq 2$.

Subcase 2a. $C = \emptyset$.

Suppose H contains maximal cliques C_1, C_2 different from $A \cup \{x\}$ and $B \cup \{y\}$. Then $C_1, C_2 \subseteq A \cup B$, and there are $a \in A, b \in B, a \not\sim b$ such that $a \in C_1 \backslash C_2, b \in C_2 \backslash C_1$. Obviously,

$$C_i \cap A \neq \emptyset, \quad C_i \cap B \neq \emptyset, \quad i = 1, 2.$$

Hence, there exist $a_1 \in A, b_1 \in B$ such that $a_1 \sim b$, $b_1 \sim a$. Since $H(a, a_1, b, b_1) \ncong G_2'$, then $a_1 \sim b_1$. We have $H(x, y, a, b, a_1, b_1) \cong G_8'$, a contradiction. Thus $s(H) \leq 3$.

Subcase 2b. $C \neq \emptyset$.

PROPOSITION 6.2. *If C has no dominating vertices of the graph $F_A = H(A \cup C \cup \{x\})$ (the graph $F_B = H(B \cup C \cup \{y\})$), then $s(F_A) = 2$ ($s(F_B) = 2$, respectively).*

Let C have no dominating vertices of F_A. Then $A \neq \emptyset$. If $A = \{a\}$, then $a \not\sim C$, and F_A contains exactly two maximal cliques: $A \cup \{x\}$ and $C \cup \{x\}$. If $C = \{c\}$, then the only such cliques are $N[c] \backslash (B \cup \{y\})$ and $A \cup \{x\}$.

Let $|A| \geq 2, |C| \geq 2$. Divide A into subsets $A_1 = \{a \in A : a \sim C\}$ and $A_2 = A \backslash A_1$. By the assumptions, $A_2 \neq \emptyset$. Suppose F_A contains a maximal clique K different from $A \cup \{x\}$ and $C \cup A_1 \cup \{x\}$. Then there exist $a \in A_2$ and $c \in C$ such that $a, c \in K$. By the definition of A_2, there exists $c' \in C \backslash \{c\}, c' \not\sim a$. By the assumptions, there exists $a' \in A_2$, $c \not\sim a'$. Since $H(a, c, a', c') \ncong G_2'$, one gets $a' \not\sim c'$. We have $H(a, c, a', c', x, y) \cong G_8'$, a contradiction. Thus $s(F_A) = 2$. This finishes the proof of the proposition.

By (6.1), it follows from Proposition 6.2 that $s(H) \leq 3$ if $A = \emptyset$ or $B = \emptyset$.

Hence we can assume that $A \neq \emptyset$ and $B \neq \emptyset$. Next, we prove that

$$(6.5) \qquad A \sim C \quad \text{or} \quad B \sim C.$$

Otherwise, if there exist $a \in A, b \in B$, and $c \in C$ such that $c \not\sim a, b$, then $H(a, b, c, x, y) \cong G_4'$ or G_6', a contradiction. If there are $c_1, c_2 \in C, a \in A, b \in B$ such that $c_1 \not\sim a, c_2 \not\sim b, c_1 \sim b, c_2 \sim a$, then $a \not\sim b$ since $H(a, b, c_1, c_2) \ncong G_2'$. Hence $H(a, b, c_1, c_2, x, y) \cong G_8'$, a contradiction.

Thus, (6.5) holds. Taking (6.1) into account, we conclude that $C \sim A \cup B$ is impossible. Without loss of generality, let $A \sim C$. By Proposition 6.2, $s(F_B) = 2$. The inequality $s(H) \leq 3$ will be proved if we show that $A \not\sim B$.

Let $b \in B, c \in C, b \not\sim c$. If there exists $a \in A$, $a \sim b$, then $H(a, b, c, y) \cong G'_2$. Otherwise, if $a \in A, b' \in B\backslash\{b\}, a \sim b', a \not\sim b$, then $b' \sim c$ since $H(a, b', c, y) \ncong G'_2$. We have $H(a, b, b', c, x, y) \cong G'_8$.

So $A \not\sim B$ and $s(H) \leq 3$. □

Acknowledgments. The authors thank A. Brandstädt, V. B. Le, V. I. Sarvanov, and all participants of the computer science seminar of Rostock University for fruitful discussions and thank the referees for numerous comments which helped to improve the text.

REFERENCES

[1] C. Benzaken and P. L. Hammer, *Boolean techniques for matroidal decomposition of independence systems and applications to graphs*, Discrete Math., 56 (1985), pp. 7–34.

[2] C. Berge, *Graphs and Hypergraphs*, North–Holland, Amsterdam, 1973.

[3] C. Berge, *Hypergraphs. Combinatorics of Finite Sets*, North–Holland, Amsterdam, 1989.

[4] J. C. Bermond and J. C. Meyer, *Graphs representatif des arêtes d'un multigraphe*, J. Math. Pures Appl., 52 (1973), pp. 299–308.

[5] I. Holyer, *The NP-completeness of edge-coloring*, SIAM J. Comput., 10 (1981), pp. 718–720.

[6] T. Kloks, D. Kratsch, and H. Müller, *Dominoes*, Lecture Notes in Comput. Sci. 903, Springer-Verlag, Berlin, 1995, pp. 106–120.

[7] L. Lovász, *Problem* 9, in Beiträge zur Graphentheorie and deren Anwendungen. Vorgetragen auf dem internationalen Kolloquium in Oberhof (DDR), Mathematische Gesellschaft der DDR—Technische Hochschule llimenau, 1977, p. 313.

[8] T. A. McKee and F. R. McMorris, *Topics in Intersection Graph Theory*, SIAM Monogr. Discrete Math. Appl. 2, SIAM, Philadelphia, 1999.

[9] S. Poljak, V. Rödl, and D. Turzik, *Complexity of representation of graphs by set systems*, Discrete Appl. Math., 3 (1981), pp. 301–312.

[10] R. I. Tyshkevich and O. P. Urbanovich, *Graphs with matroidal number* 2, Vestsi Akad. Navuk Belarusi. Ser. Fiz.- Mat. Navuk, 3 (1989), pp. 13–17 (in Russian).

SIAM J. DISCRETE MATH.
Vol. 16, No. 3, pp. 449–478

LIST PARTITIONS[*]

TOMAS FEDER[†], PAVOL HELL[‡], SULAMITA KLEIN[§], AND RAJEEV MOTWANI[¶]

Abstract. List partitions generalize list colorings and list homomorphisms. (We argue that they may be called list "semihomomorphisms.") Each symmetric matrix M over $0, 1, *$ defines a list partition problem. Different choices of the matrix M lead to many well-known graph theoretic problems, often related to graph perfection, including the problem of recognizing split graphs, finding homogeneous sets, clique cutsets, stable cutsets, and so on. The recent proof of the strong perfect graph theorem employs three kinds of decompositions that can be viewed as list partitions.

We develop tools which allow us to classify the complexity of many list partition problems and, in particular, yield the complete classification for small matrices M. Along the way, we obtain a variety of specific results, including generalizations of Lovász's communication bound on the number of clique-versus-stable-set separators, polynomial time algorithms to recognize generalized split graphs, a polynomial algorithm for the list version of the clique cutset problem, and the first subexponential algorithm for the skew cutset problem of Chvátal. We also show that the dichotomy (NP-complete versus polynomial time solvable), conjectured for certain graph homomorphism problems, would, if true, imply a slightly weaker dichotomy (NP-complete versus quasi-polynomial) for our list partition problems.

Key words. list homomorphisms, H-colorings, graph partitions, perfect graphs, split graphs, skew cutsets, clique cutsets, 2-joins, communication bound, dichotomy, constraint satisfaction problems, quasi-polynomial algorithms, NP-completeness, polynomial time algorithms

AMS subject classifications. 05C85, 05C15, 05C17, 68Q25

PII. S0895480100384055

1. Introduction.

Many combinatorial problems seek a partition of the vertices of a given graph into subsets satisfying certain constraints *internally* (a set may be required to be stable or complete) and *externally* (two sets may be required to be completely nonadjacent—no vertex of one adjacent to any vertex of the other—or completely adjacent—each vertex of one adjacent to each vertex of the other). We may formulate a common generalization of such problems as follows: partition the vertices of an input graph into k parts $A_1, A_2, \ldots, A_k$ with a fixed "pattern" of requirements as to which A_i's are stable or complete, and which pairs A_i, A_j are completely nonadjacent or completely adjacent. (In some cases, we also deal with a generalization where we replace "stable" and "complete" with more general notions of "sparse" and "dense.") These requirements may be conveniently captured by a symmetric k-by-k matrix M in which the diagonal entries $M_{i,i}$ encode the internal restrictions on the

[*] Received by the editors March 18, 2000; accepted for publication (in revised form) January 9, 2003; published electronically May 22, 2003. A preliminary version of this paper appeared in *Proceedings of the* 31*st Annual ACM Symposium on Theory of Computing*, Atlanta, GA, 1999, pp. 464–472.

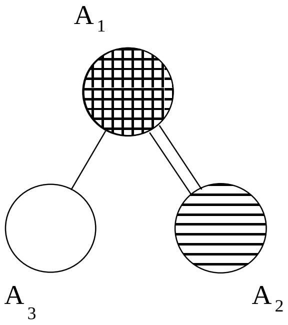
http://www.siam.org/journals/sidma/16-3/38405.html

[†] tomas@theory.stanford.edu.

[‡] School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, V5A1S6 (pavol@cs.sfu.ca). This author's research was supported by a research grant from the National Sciences and Engineering Research Council.

[§] Departamento da Ciência da Computação - I.M., COPPE/Sistemas, Universidade Federal do Rio de Janeiro, RJ, 21945-970, Brazil (sula@cos.ufrj.br). This author's research was supported by CNPq and PRONEX 107/97.

[¶] Department of Computer Science, Stanford University, CA 94305-9045 (rajeev@cs.stanford.edu). This author's research was supported by ARO MURI grant DAAH04–96–1–0007, NSF grant IIS-9811904, and NSF Young Investigator Award CCR–9357849, with matching funds from IBM, Mitsubishi, Schlumberger Foundation, Shell Foundation, and Xerox Corporation.



$$M = \begin{pmatrix} 1 & 1 & * \\ 1 & * & 0 \\ * & 0 & 0 \end{pmatrix} \qquad A = \left(\begin{array}{ccc|c|cc} * & 1 & 1 & 1 & * & * \\ 1 & * & 1 & 1 & * & * \\ 1 & 1 & * & 1 & * & * \\ \hline 1 & 1 & 1 & * & 0 & 0 \\ \hline * & * & * & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 \end{array}\right)$$

FIG. 1. *A partition, its matrix M, and example block structure of an adjacency matrix A corresponding to a graph with an M-partition (in the matrix A, each $*$ represents either* 0 *or* 1*).*

sets A_i and the off-diagonal entries $M(i,j), i \neq j$, encode the restriction on the edges between A_i and A_j.

Specifically, let M be a fixed symmetric k-by-k matrix over $0, 1, *$. An *M-partition* of a graph G is a partition of the vertex set $V(G)$ into k parts $A_1, A_2, \ldots, A_k$ such that A_i is stable (i.e., independent) if $M_{i,i} = 0$, or complete (i.e., a clique) if $M_{i,i} = 1$ (with no restriction if $M_{i,i} = *$), and such that A_i and A_j are completely nonadjacent if $M_{i,j} = 0$, or completely adjacent if $M_{i,j} = 1$ (with no restriction if $M_{i,j} = *$). When k is small, we usually refer to parts $A, B, C, \ldots$ instead of $A_1, A_2, A_3, \ldots$ and write, for example, $A = 0$ to mean $M_{A,A} = 0$ or $AB = 1$ instead of $M_{A,B} = 1$.

A graph G admits an M-partition if and only if its adjacency matrix $A = A(G)$ can be written, after a suitable simultaneous row and column permutation, in a block form corresponding to M, where 0 denotes an all-zero matrix, 1 denotes an all-one matrix (with $*$'s assumed on the main diagonal), and $*$ denotes any matrix. In Figure 1 we give an example matrix M and illustrate what an adjacency matrix A of graph G with an M-partition might look like. In the same figure, we also introduce a symbolic figure showing a general M-partition. The empty circle depicts a stable set (0 on the main diagonal of M), a shaded circle depicts an arbitrary set (a diagonal $*$ in M), and a doubly shaded circle depicts a clique (a diagonal 1); similarly, two parts are joined by two lines if they are completely adjacent (an off-diagonal 1), joined by a single line if there is no restriction on the edges between them (an off-diagonal $*$), and not joined at all if they are completely nonadjacent (an off-diagonal 0).

Many graph theoretic concepts can be modeled by M-partitions. Indeed, in Figure 2, we illustrate three such concepts—from the well-known notions of a graph coloring and a split graph [32] to the more recent notion of a clique-cross partition [21].

All three concepts have natural generalizations which may also be modeled as M-partitions.

A k-coloring of a graph G is an M-partition of G where the matrix M has zeros on the main diagonal and asterisks everywhere else. In other words, M is obtained from the adjacency matrix of the complete k-graph by replacing all ones with asterisks. An M-partition of G, where M is obtained the same way from the adjacency matrix of an *arbitrary* graph H, is called an *H-coloring* or a *homomorphism* [33, 34]. Thus an H-coloring of G, or a homomorphism of G to H, is a partition of $V(G)$ into sets $A_h, h \in V(H)$, such that A_h is stable when h is not a loop of H, and $A_h, A_{h'}$ are completely nonadjacent when hh' is not an edge of H. The k-coloring problem is well known to be polynomial time solvable when $k \leq 2$ and NP-complete otherwise [31].

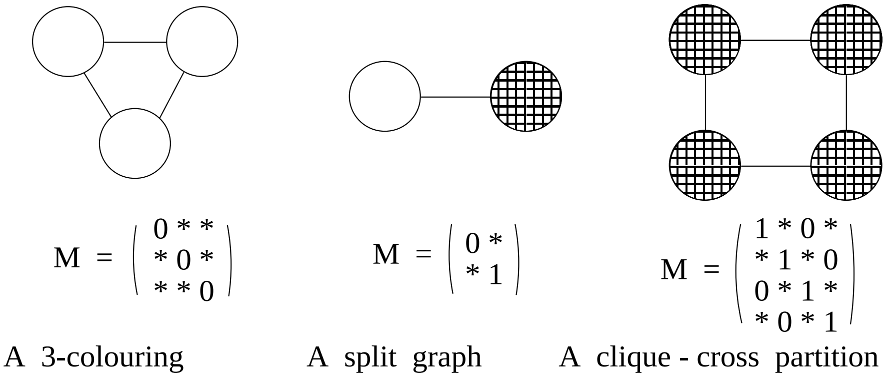

M = (0 * * / * 0 * / * * 0)
M = (0 * / * 1)
M = (1 * 0 * / * 1 * 0 / 0 * 1 * / * 0 * 1)
A 3-colouring
A split graph
A clique - cross partition

FIG. 2. *Three typical partition problems.*

The H-coloring problem is polynomial time solvable when H is bipartite or when H contains a loop, and it is NP-complete otherwise [34].

Consider now a matrix M obtained from the adjacency matrix of a graph H by replacing all zeros with asterisks. Then an M-partition of a graph G is a partition of $V(G)$ into sets $A_h, h \in V(H)$, such that A_h is complete when h is a loop of H, and $A_h, A_{h'}$ are completely adjacent when hh' is an edge of G. In other words, an M-partition of G is a homomorphism of the complement of G to the complement of H. Such a partition has been called a *cohomomorphism* of G to H. Since the general case of an M-partition of G mixes the homomorphism and cohomomorphism partition types of constraints, we propose to call it a *semihomomorphism* partition.

A *split graph* is a graph which admits a partition into a stable set and a clique [32], i.e., an M-partition where M is the matrix given in Figure 2, with asterisks off the main diagonal, and exactly one 0 and one 1 on the main diagonal. An (a, b)-*graph* [5] is a natural generalization—a graph whose vertices can be partitioned into a stable sets and b cliques; the corresponding M is an $(a+b)$-by-$(a+b)$ matrix having all off-diagonal entries equal to $*$ and with a zeros and b ones on the main diagonal. When $a, b \leq 2$ (this includes split graphs, which have $a = b = 1$), the (a, b)-graphs can be recognized in polynomial time. (Brandstädt claimed such algorithms in [5], which were in error [6]; more involved polynomial time algorithms were given in [7], and a new algorithm of complexity $O((n+m)^2)$ was given in [8]; polynomial time algorithms also follow from our more general results in section 3.) On the other hand, it is easy to see that when a or b is at least 3 it is NP-complete to recognize (a, b)-graphs [5, 8]. The split graphs ($a = b = 1$) are a well-known class of perfect graphs (cf. below), and they admit efficient algorithms for many standard combinatorial optimization problems [32]. The class of (a, b)-graphs has also been investigated from the perspective of perfect graphs [36].

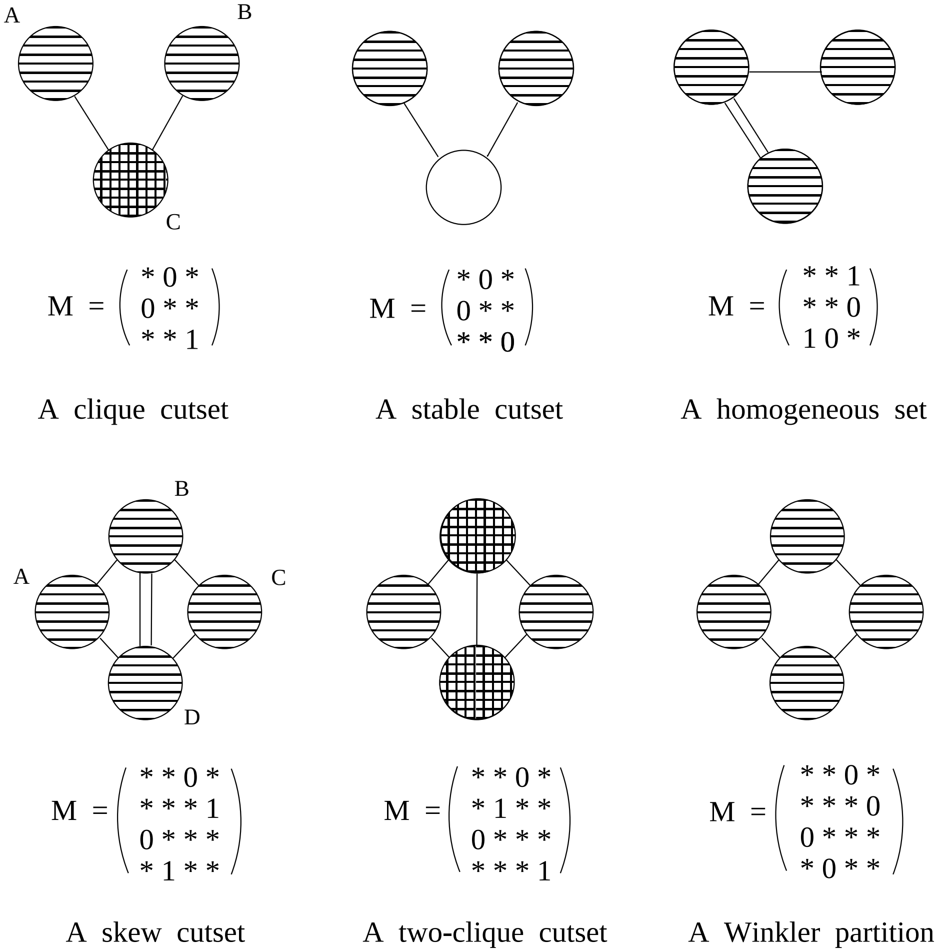
A *clique-cross partition* [21] of a graph G is a partition of the vertices of G into four disjoint cliques A, B, C, D such that A, C as well as B, D are completely nonadjacent. This is an M-partition, where M is given in Figure 2; note that M is obtained from the adjacency matrix of the four-cycle by replacing all ones with asterisks and setting all diagonal entries to 1. The more general concept [39] of an *H-clique partition* is the M-partition problem where M is the matrix obtained in the same way from the adjacency matrix of an *arbitrary* graph H. A clique-cross partition can be found in linear time [21]. The more general H-clique partition problem is polynomial time solvable when H is a triangle-free graph; otherwise it is NP-complete [39].

Several other well-known graph concepts correspond to M-partitions with additional restrictions. Many of these concepts arise in connection with graph perfection. Briefly, a graph is *perfect* if the chromatic number and the maximum clique size are the same for the graph and all its induced subgraphs. In the 1960s, Berge [3] formulated two perfect graph conjectures: (1) the weak conjecture, asserting that a graph is perfect if and only if its complement is perfect; and (2) the strong conjecture, asserting that odd cycles and their complements are the only minimal imperfect graphs (that is, imperfect graphs such that every induced subgraph is perfect). The weak perfect graph conjecture was proved by Lovász in the 1970s [38]; the strong conjecture has just been verified by Chudnovsky et al. [12]. Their solution consists of a detailed structural characterization of graphs which do not contain an induced odd cycle or its complement. These graphs are shown to be constructible from some basic perfect graphs by three operations which preserve perfection. Each of these three operations turns out to be an M-partition (with some additional properties); cf. below. We note that as of this writing there is still no polynomial time algorithm for the recognition of perfect graphs. (One may try to efficiently find a decomposition of a given graph into the basic perfect graphs using these operations. However, even though one application of each of these operations is now possible in polynomial time, it is not known how to perform the entire decomposition in polynomial time.)

We first bring up M-partitions with the additional restriction that the parts be nonempty.

A *clique cutset* [42, 47] of a connected graph G is a complete subgraph C whose removal disconnects G. Clearly, G has a clique cutset if and only if it admits a partition of the vertices into three nonempty subsets A, B, C such that C is a clique and A, B are completely nonadjacent (so that the removal of C disconnects A from B), i.e., if and only if it admits an M-partition, where M is the matrix given in Figure 3, with the additional restriction that all parts are nonempty. Finding clique cutsets is possible in polynomial time [42, 47, 48] (cf. also section 5.2) and is the basis of a decomposition algorithm [42], which allows efficient solution of many optimization problems for the class of decomposable graphs [42]. A *stable cutset* [43] is defined analogously (C is stable) and also corresponds to an M-partition with all parts nonempty. Stable cutsets arose because of an early result of Tucker [43] that a minimal imperfect graph other than an odd cycle cannot contain a stable cutset; finding a stable cutset has been proved NP-complete in [28]. A *two-clique cutset* is defined similarly as a union of two complete subgraphs that disconnects the input graph, and the two-clique cutset problem corresponds to the matrix M in Figure 3, again with the additional restriction that all parts be nonempty. In section 5 we give a subexponential algorithm for the list version of the problem. This has recently been improved to a polynomial time algorithm [10].

A *skew cutset* of a connected graph G is a pair of disjoint nonempty sets B, D in G such that the removal of $B \cup D$ disconnects the graph and such that B, D are completely adjacent (the "skew property"). Once again, this is clearly a partition problem—we wish to partition the vertices of G into four nonempty sets A, B, C, D such that A, C are completely nonadjacent and B, D are completely adjacent. This is an M-partition, where M is given in Figure 3, with all parts nonempty. Skew cutset partitions (of a certain kind) are one of the three operations used by Chudnovsky et al. [12] in the proof of the strong perfect graph conjecture. This was anticipated by Chvátal [13], who conjectured that a minimal imperfect graph cannot contain a skew



$$M = \begin{pmatrix} * & 0 & * \\ 0 & * & * \\ * & * & 1 \end{pmatrix}$$

A clique cutset

$$M = \begin{pmatrix} * & 0 & * \\ 0 & * & * \\ * & * & 0 \end{pmatrix}$$

A stable cutset

$$M = \begin{pmatrix} * & * & 1 \\ * & * & 0 \\ 1 & 0 & * \end{pmatrix}$$

A homogeneous set

$$M = \begin{pmatrix} * & * & 0 & * \\ * & * & * & 1 \\ 0 & * & * & * \\ * & 1 & * & * \end{pmatrix}$$

A skew cutset

$$M = \begin{pmatrix} * & * & 0 & * \\ * & 1 & * & * \\ 0 & * & * & * \\ * & * & * & 1 \end{pmatrix}$$

A two-clique cutset

$$M = \begin{pmatrix} * & * & 0 & * \\ * & * & * & 0 \\ 0 & * & * & * \\ * & 0 & * & * \end{pmatrix}$$

A Winkler partition

FIG. 3. *Other well-known partition problems.*

cutset. He proved this for the special skew cutsets where B consists of a single vertex (the entire conjecture has now been proved in [12]). For this special case he also gave a polynomial time recognition algorithm, and he asked for the complexity of finding a general skew cutset. (When both B and D are required to be stable ($B = D = 0$), we are asking for a complete bipartite cutset, and this recognition problem is NP-complete [28].) In [26], we offered the first subexponential time algorithm, strongly suggesting that the problem is not NP-complete. Most recently, one of us (Klein), together with de Figueiredo, Kohayakawa, and Reed, indeed found a polynomial time algorithm [29].

Winkler formulated a similar problem, seeking a partition into nonempty sets A, B, C, D where there are no edges between A and C nor between B and D, but where there is at least one edge between A and B, between B and C, between C and D, and between D and A. Winkler asked for the complexity of this problem; it has been shown NP-complete in [45]. This is an M-partition problem (M is given in Figure 3), where there are not only additional restrictions on the nonemptiness of the parts but also on the presence of edges between certain pairs of parts.

A *homogeneous set* [17] in a graph G is a set C of vertices of G such that each vertex outside of C is adjacent to either all or to none of the vertices in C. It is again easy to see that this is a partition problem—we want to partition the vertices into three subsets A, B, C such that A, C are completely adjacent and B, C are completely nonadjacent. To avoid the trivial homogeneous sets consisting of a single vertex or the entire vertex set, we also require that C has at least two vertices and that $A \cup B$ be nonempty. Therefore, this is an M-partition (with M given in Figure 3) where there are more complex restrictions on the sizes of the parts. Homogeneous sets also define a decomposition (the "modular decomposition") which facilitates the recognition of comparability graphs (and other similar classes of graphs) [17, 40]. Homogeneous sets (and modular decompositions) can be found efficiently [40]. The fact that minimal imperfect graphs cannot have a homogeneous set was used by Lovász [38] to prove the weak perfect graph conjecture.

Two generalizations of the homogeneous set partition have been used in the proof of the strong perfect graph conjecture in [12]. They are the *homogeneous pair* and the 2-*join* partitions. The matrices M corresponding to these two partition problems are given below. Both problems require certain size restrictions.

$$\begin{pmatrix} \ast & \ast & 1 & 0 & 1 & 0 \\ \ast & \ast & 1 & 0 & 0 & 1 \\ 1 & 1 & \ast & \ast & \ast & \ast \\ 0 & 0 & \ast & \ast & \ast & \ast \\ 1 & 0 & \ast & \ast & \ast & \ast \\ 0 & 1 & \ast & \ast & \ast & \ast \end{pmatrix} \qquad \begin{pmatrix} \ast & \ast & \ast & 1 & 0 & 0 \\ \ast & \ast & \ast & 0 & 1 & 0 \\ \ast & \ast & \ast & 0 & 0 & 0 \\ 1 & 0 & 0 & \ast & \ast & \ast \\ 0 & 1 & 0 & \ast & \ast & \ast \\ 0 & 0 & 0 & \ast & \ast & \ast \end{pmatrix}$$

Homogeneous pair 2-join

Chvátal and Sbihi [14] were the first to show that no minimal imperfect graph contains a homogeneous pair and used this result to prove the perfectness of a class of graphs. A polynomial time algorithm for the recognition of homogeneous pairs has been given by Everett, Klein, and Reed [22]. 2-joins were first introduced in [15] and also used to prove the perfectness of a certain class of graphs.

A number of other generalizations have been studied and used for proving the perfectness of several graph classes [4, 9, 20, 15, 42, 48]; polynomial time algorithms for finding these partitions can be found in [18, 19, 9, 15]. The most general of these appear to be the concepts of universal 2-amalgam and universal 2-join [11]; they include most of the generalizations, and are M-partitions with a matrix M of size 7, and some (fairly involved) size constraints.

To capture all these additional requirements (that certain parts be nonempty, or have at least a certain number of vertices, individually, or in groups, or have at least some edges joining them, etc.), we shall introduce the concept of lists. In the list version of a partition problem, each vertex of the input graph has a list of the parts in which it is allowed to be placed. This gives us a wide variety of options in restricting the contents of the individual parts or of their connections. For instance, in the case of homogeneous sets, we may ensure that C has at least two vertices and $A \cup B$ is nonempty by choosing three vertices x, y, z of the input graph and specifying that the lists of x, y consist only of C and the list of z consists of A, B. Thus the problem of finding a homogeneous set in a graph with n vertices is reduced to n^3 list partition problems. (A homogeneous set exists if and only if at least one of the n^3 choices of x, y, z has a desired list partition.) Analogously, one can ensure that there is at least one edge between parts X and Y by restricting (with the choice of lists) two adjacent vertices x, y to be placed into X, Y, respectively, for all possible choices of an edge xy in the input graph.

Concretely, let M be a fixed k-by-k matrix. Given a graph G, and for each vertex $v \in V(G)$ a set ("list") $L(v) \subseteq \{1, 2, \ldots, k\}$, we define a *list M-partition* of G, with respect to the lists L, to be an M-partition $A_1, A_2, \ldots, A_k$ of G in which each $v \in V(G)$ belongs to a part A_i with $i \in L(v)$. The *list M-partition problem* asks whether or not an input graph G with lists L admits a list M-partition.

Both the basic M-partition problem ("Does the input graph admit an M-partition?") and the problem of the existence of an M-partition with all parts nonempty admit polynomial time reductions to the list M-partition problem, as do all of the above problems with more complex constraints.

List partitions generalize list colorings, which have proved to be very fruitful in the study of graph colorings [1, 30]. They also generalize list homomorphisms (or list H-colorings) which have brought a degree of order to the study of the complexity of graph homomorphisms; cf. below. One reason why lists are useful is that they allow us to solve problems by recursing to subproblems with modified lists. (This was also exploited in the algorithms in [29, 10].)

List homomorphisms (or list H-colorings) [23, 24, 25] are close in spirit to list partitions. A list H-coloring of a graph G is a list M-partition of G, where M is obtained from the adjacency matrix of G by replacing all 1's with $*$'s. A complete classification of the complexity of list H-colorings, i.e., of the complexity of list M-partition when M is a $(0, *)$-matrix, is given in the series of papers [23, 24, 25].

When all the diagonal entries of M are $*$ (i.e., when H has all loops), the problem is polynomial time solvable if H is an interval graph and is NP-complete otherwise [23]. When all diagonal entries of M are 0 (i.e., when H has no loops), the problem is polynomial time solvable if H is bipartite and its complement $\overline{H}$ is a circular arc graph, and is NP-complete otherwise [24].

For general $(0, *)$-matrices M a complete classification is given in [25]. It again relates to a kind of geometric representation of H. The important point for this paper is that this classification implies that all list M-partition problems for $(0, *)$-matrices M (i.e., all list H-coloring problems) are polynomial time solvable or NP-complete. This kind of "dichotomy" is rare in general and is conjectured for the more general context of constraint satisfaction problems in [27].

Similar comments apply to M-partitions where M is a $(*, 1)$-matrix (cf. Proposition 2.7). This problem corresponds to a homomorphism problem among the complementary graphs (still a constraint satisfaction problem). The appealing feature of the general M-partition problem is that it allows these homomorphism-type (constraint–satisfaction-type) constraints on both edges and nonedges of the graph. In particular, general list M-partition problems are *not* constraint satisfaction problems.

As the above examples illustrate, we are often interested in the complexity of finding the desired partitions. This is the recurring theme of all of the above discussion. In this paper, we shall focus on this aspect, although, of course, list partitions offer other interesting questions.

The organization of the paper is as follows.

In section 2, we describe some basic techniques.

In section 3, we introduce sparse-dense partitions. Graphs which admit sparse-dense partitions can be recognized efficiently if sparse and dense graphs can. Many partition problems can be modeled as sparse-dense partitions, including many of our

M-partitions, and we obtain polynomial time algorithms for several such problems.

In section 4, we investigate separator theorems. Motivated by a result of Lovász, we derive several extensions which will be used later. This technique leads to subexponential, but not necessarily polynomial, algorithms for certain M-partitions.

In section 5, we illustrate the use of our tools on some prominent example list M-partition problems: the $(2, 1)$- and $(2, 2)$-graphs of Brandstädt, the clique cutset problem, the skew cutset problem of Chvátal, and the two-clique cutset problem.

In section 6, we apply the techniques to classify the complexity of list M-partition problems when the matrix M is small. All these problems are polynomial time solvable when M is a 2-by-2 matrix. For 3-by-3 matrices we classify the problems as polynomial time solvable or NP-complete. When M is a 4-by-4 matrix, we are able to show that all these problems are NP-complete or "quasi-polynomial."

In section 7, we prove that if it is true (as conjectured in [27]; cf. also [25]) that all constraint satisfaction problems are polynomial or NP-complete, then it also follows that all list M-partition problems are quasi-polynomial or NP-complete.

We use the term *quasi-polynomial* for a function that is bounded by $n^{c \log^t n} = 2^{c \log^{t+1} n}$ for some positive constants c, t. While we, of course, prefer to find polynomial time algorithms, we take the existence of a quasi-polynomial (time) algorithm as evidence that the problem is not likely to be NP-complete. Indeed, no NP-complete problem is known to be solved by a quasi-polynomial algorithm, and, since all NP-complete problems are polynomially equivalent, a quasi-polynomial algorithm for any one NP-complete problem would imply the existence of such algorithms for *all* NP-complete problems.

Since the preliminary version of these results has been presented at STOC [26], new results have appeared or been announced, such as [29, 10], discussed elsewhere in this paper. In addition, further work has been done on list M-partitions for chordal graphs [41, 35] and on list M-partitions when the matrix M is not necessarily symmetric (these are partitions of directed graphs) [44].

2. Basic tools.

We begin by assembling some basic techniques. For some matrices M, these are sufficient to solve the list M-partition problem in polynomial time. We shall also use them in conjunction with other tools to be described in later sections.

The most basic technique is the 2-satisfiability algorithm of [2]. Suppose first that M is a 2-by-2 matrix seeking to partition the input graph into two parts, say A, B. We can solve the list M-partition problem by introducing a boolean variable x_v for each vertex v of the input graph G; we think of the value of x_v as encoding whether or not the vertex v belongs to the part A of the partition ($x_v = 1$ means $v \in A$; $x_v = 0$ means $v \notin A$). It is then easy to see that all the constraints, and lists, of the list M-partition problem can be stated by polynomially many clauses with at most two literals each. For instance, if A is to be a stable set ($A = 0$), we want to express the constraint that adjacent vertices cannot both be in A; in other words, for any edge uv we must have $u \notin A$ or $v \notin A$. Therefore, we impose the constraint $\overline{x_u} \vee \overline{x_v}$ for every edge uv of G. Similarly, if, say, A, B are to be completely adjacent ($AB = 1$), we want to make sure that for any nonedge uv has $u \notin A$ or $v \notin B$. Therefore we impose the constraint $\overline{x_u} \vee x_v$ for each nonedge uv of G. (Note that if uv is a nonedge, then so is vu; thus we obtain a pair of clauses.) Finally, it is easy to encode the lists as clauses of size 1—e.g., if the list of v is, say, B, we impose the constraint $\overline{x_v}$. Hence the problem can now be solved by the 2-satisfiability algorithm [2].

The same technique applies any time we have an instance in which every list has size at most two. We simply view each list $L(v)$ as an ordered set, and we interpret $x_v = 1$ to mean v belongs to the first member of its list and $x_v = 0$ to mean it belongs to the second member of its list.

Proposition 2.1. *There is a polynomial time algorithm which solves any list M-partition problem restricted to instances in which the list of every vertex of the input graph has size at most two.* □

One other basic technique occurs in many places—the placing of a vertex. This is one big advantage of lists—one can recurse to a smaller problem by deleting a vertex and modifying the remaining lists. Suppose the input consists of the graph G with lists L, and let v be a vertex of G. We may decide at some point to place a vertex v into a part X (either because the list of v has only X in it or because we will consider the other options later). This can be accomplished by removing v from the graph and updating the lists of all the other vertices to take it into account. Specifically, for all Y with $XY = 0$, we remove Y from the lists of all neighbors of v (they can no longer be placed in Y), and for all Z with $XZ = 1$, we remove Z from the lists of all nonneighbors of v (for a similar reason). Call the resulting lists L_X^v.

Proposition 2.2. *There is a list M-partition of the graph G with respect to the lists L, with $v \in X$, if and only if there is a list M-partition of the graph $G - v$ with respect to the lists L_x^v.* □

Suppose a row X of M contains both a 0 and a 1, say $XY = 0$ and $XZ = 1$ (either of Y and Z could be X; i.e., we could have $X = 0,\ XZ = 1$ or $XY = 0,\ X = 1$). In this case we can reduce the list M-partition problem for an n-vertex input graph G (with respect to lists L) to the following (at most $n + 1$) subproblems.

First check whether or not the input graph G has a partition in which no vertex lies in the part X, and then check for each vertex v of G which has X in its list whether or not G has a partition with v in X. The former can clearly be accomplished by removing X from all lists (call the resulting lists L'), and the latter can be tested by placing v in X and updating the lists of all other vertices of G as explained above. Note that since $XY = 0,\ XZ = 1$, these updates will result in no list in L_X^v containing both Y and Z.

Proposition 2.3. *Suppose the matrix M has $XY = 0,\ XZ = 1$. Then the input graph G admits a list M-partition with respect to lists L if and only if G admits a list M-partition with respect to the lists L' or if $G - v$ admits a list M-partition with respect to the lists L_X^v for some vertex v of G.* □

Corollary 2.4. *Suppose the matrix M has $XY = 0,\ XZ = 1$. Then the list M-partition problem can be reduced to one instance with no list containing X and at most n instances with no list containing both Y and Z.* □

This is particularly useful when $k = 3$, as in this case all lists become size at most two.

We say that X *dominates* Y in the matrix M if for each Z (possibly equal to X and Y) we have $XZ = YZ$ or $XZ = *$. If X dominates Y, we can eliminate Y from any list containing X, since any vertex that goes to part Y can be placed to X instead. Thus for an input graph G with lists L we may define the modified lists L' obtained from L by removing Y from any list that contains X. (Note that this also may, in some cases, result in all lists having size at most two.)

Proposition 2.5. *If X dominates Y in the matrix M, then an input graph G admits a list M-partition with respect to lists L if and only if it admits a list M-partition with respect to lists L'.* □

Thus if X dominates Y, we may assume that no list contains both X and Y. In particular, when X dominates all other parts, we may assume that each list is either just $\{X\}$ or does not contain X. This allows us to drop X by placing all vertices with lists $\{X\}$, as explained above, and reducing the matrix by eliminating the row and column corresponding to X.

We say that a k-by-k matrix M *contains* a k'-by-k' matrix M', $k' \leq k$ if M' is a principal submatrix of M. In other words, the parts of the M'-partition problem are a subset of the parts of the M-partition problem, with the same constraints on the parts and their connections.

PROPOSITION 2.6. *If M contains M', and the list M'-partition problem is NP-complete, then so is the list M-partition problem.*

Proof. We reduce the list M'-partition problem to the list M-partition problem as follows: Let G with lists $L(v) \subseteq \{1, 2, \dots, k'\}$ be any instance of the list M'-partition problem. We may view the same graph G, with the same lists L, as an instance of the list M-partition problem as well, since each $L(v) \subseteq \{1, 2, \dots, k'\} \subseteq \{1, 2, \dots, k\}$. Clearly, G with lists L admits a list M'-partition if and only if it admits a list M-partition. □

The *complement* $\overline{M}$ of a matrix M is obtained from M by replacing each 0 by 1 and each 1 by 0 (asterisks remain unchanged).

PROPOSITION 2.7. *A graph G admits a list M-partition, with respect to the lists L, if and only if its complement $\overline{G}$ admits a list $\overline{M}$-partition with respect to the same lists L.* □

We close this section by formally stating the observations made in the preceding section, summarizing the relevant results of [23, 24, 25].

PROPOSITION 2.8. *If M is a $(0, *)$-matrix or a $(1, *)$-matrix, then the list M-partition problem is polynomial time solvable or NP-complete.* □

3. Sparse-dense partitions.

We now introduce a class of problems which will be useful for several M-partition problems and which are interesting in their own right.

Let $\mathcal{S}$ and $\mathcal{D}$ be two classes of graphs. We say that $\mathcal{S}$ is the class of *sparse graphs* and $\mathcal{D}$ the class of *dense graphs* if $\mathcal{S}$ and $\mathcal{D}$ satisfy the following constraints:

- Both $\mathcal{S}$ and $\mathcal{D}$ are closed under taking induced subgraphs.
- There exists a constant c such that the intersection $S \cap D$ has at most c vertices for any $S \in \mathcal{S}$ and $D \in \mathcal{D}$.

In a given graph G, we say that a set of vertices is sparse (dense) if the subgraph of G they induce is sparse (respectively dense) with respect to some classes $\mathcal{S}$, $\mathcal{D}$ of sparse and dense graphs.

A *sparse-dense partition* of a graph G, with respect to the classes $\mathcal{S}$ and $\mathcal{D}$ of sparse and dense graphs, is a partition of $V(G)$ into two parts $V(G) = S \cup D$ such that $S \in \mathcal{S}$ (S is sparse) and $D \in \mathcal{D}$ (D is dense).

Sparse-dense partitions are inspired by split graphs. Indeed, we may take $\mathcal{S}$ to consist of all edgeless graphs (stable sets) and $\mathcal{D}$ to consist of all complete graphs (cliques). It is clear that both $\mathcal{D}$ and $\mathcal{S}$ are closed under taking induced subgraphs, and as an $S \in \mathcal{S}$ and a $D \in \mathcal{D}$ have at most one vertex in common, we can take $c = 1$. A graph has a sparse-dense partition with respect to this choice of $\mathcal{S}, \mathcal{D}$ if and only if it can be partitioned into a stable set and a clique, i.e., if and only if it is a split graph.

There are a number of other situations conveniently modeled by sparse-dense partitions. Several are described at the end of this section. Let us just mention the following typical examples:

(a, b)-graphs. Let $\mathcal{S}$ consist of all a-colorable graphs, and let $\mathcal{D}$ consist of all graphs whose complements are b-colorable; we can take $c = ab$.

Partitions into a graph with clique-size at most a and a graph with stable-set-size at most b. Let $\mathcal{S}$ consist of all graphs without cliques of size $a + 1$, and let $\mathcal{D}$ consist of all graphs without stable sets of size $b + 1$. The constant c can be taken to be the Ramsey number $R(a + 1, b + 1)$, as is explained in Proposition 3.3 below.

Partitions into a planar graph and a clique. Just to illustrate the range of possibilities, we may define $\mathcal{S}$ to consist of all planar graphs and define $\mathcal{D}$ to consist of all complete graphs. Kuratowski's theorem implies that we can take $c = 4$.

In [8] the authors consider a number of partition problems with similar flavor, e.g., into a stable set and a tree (NP-complete), a stable set and a trivially perfect graph (NP-complete), or a stable set and a threshold graph (polynomial time solvable). The latter satisfies the conditions for a sparse-dense partition and can in fact be solved by our technique; cf. [8].

In most of our examples, the classes $\mathcal{S}, \mathcal{D}$ are recognizable in polynomial time. (Of the above examples, only the (a, b)-graphs with $a \geq 3$ or $b \geq 3$ are an exception.) It turns out that in such a case the existence of a sparse-dense partition can be decided in polynomial time. In fact, in such a case *all* sparse-dense partitions can be found in polynomial time.

THEOREM 3.1. *Let $\mathcal{S}, \mathcal{D}$ be classes of sparse and dense graphs respectively.*

A graph on n vertices has at most n^{2c} different sparse-dense partitions.

Furthermore, all these partitions can be found in time proportional to $n^{2c+2}T(n)$, where $T(n)$ is the time for recognizing sparse and dense graphs.

Proof. Let $V(G) = S \cup D$ be a particular sparse-dense partition. Then any other sparse-dense partition $V(G) = S' \cup D'$ has $|S' \cap D| \leq c$ and $|S \cap D'| \leq c$, so S' is obtained from S by deleting at most c vertices and inserting at most c new vertices. In fact, if we allow ourselves to insert back a vertex that has just been deleted, we can say that we make exactly c deletions and exactly c insertions. Each of these at most $2c$ operations can be made in at most n ways. This observation proves the first assertion and allows us to find *all* sparse-dense partitions if one such partition is known. It amounts to a $2c$-local search (the current S is changed in at most $2c$ vertices), during which one set is tested for sparseness and one set is tested for denseness; thus it can be performed in time n^{2c} times $2T(n)$.

It remains to explain how to find the first sparse-dense partition. The algorithm proceeds in two phases. The first phase attempts to find as large a sparse set as possible. This is based on the observation that if $V(G) = S \cup D$ is a sparse-dense partition and S' a sparse set smaller than S, then $S' \cap D$ has at most c vertices, and hence, as above, S' can be enlarged by removing some c vertices and inserting some $c + 1$ new vertices (recall that subsets of sparse sets are sparse). Thus, starting with any sparse set (for instance, the empty set), we can increase its size by performing a $(2c + 1)$-local search (making all possible c deletions and $c + 1$ insertions and testing if the result is sparse) in time $n^{2c+1}T(n)$. After performing this operation at most n times, we reach a situation where the current sparse set can no longer be enlarged in this way. Clearly, at this point our current sparse set S' has the same size as the (unknown) set S.

The second phase of the algorithm attempts to change S', without changing its size, until $V(G) - S'$ is dense. This is accomplished by a $2c$-local search, based on a very similar principle—namely, if $V(G) = S \cup D$ is a sparse-dense partition and $|S| = |S'|$, then S is obtained from S' by a deletion of c vertices and the insertion of

c other vertices. Thus we can test all n^{2c} possible new sets S' for sparseness and the corresponding $V(G) - S'$ for denseness, and if no sparse-dense partition is found we can be sure none exists.

The most time-consuming operation is the first phase of the algorithm, finding one sparse-dense partition—taking time $n^{2c+2}T(n)$. □

In cases where computing $T(n)$ is hard (such as (a, b)-graphs with a or b at least 3), it also turns out to be hard to decide if a sparse-dense partition exists.

PROPOSITION 3.2. *Suppose that the disjoint union of sparse graphs is also sparse. If testing for sparse graphs is NP-complete, then the partition problem into sparse and dense graphs is also NP-complete.*

Proof. Suppose we wish to test whether G is sparse. We can construct G' by taking the disjoint union of $c + 1$ copies of G. Then G' has a sparse-dense partition if and only if G is sparse. Indeed, if G is sparse, then G' is sparse by the assumption. On the other hand, if G' admits a partition, then our algorithm from the proof of Theorem 3.1 will find a pair (S, D) where D has at most c vertices. Therefore, one of the copies of G is contained in S, so G is sparse. □

We have defined sparse and dense subgraphs with respect to each other, since the definition depends on the existence of a constant c bounding their intersections. The next result shows that we can define sparse and dense graphs independently, under the assumption that stable sets are always sparse and cliques are always dense.

PROPOSITION 3.3. *Suppose $\mathcal{S}$ is a class of graphs sparse with respect to the cliques (with a constant a), and suppose $\mathcal{D}$ is a class of graphs dense with respect to the stable sets (with a constant b). Then $\mathcal{S}, \mathcal{D}$ are sparse and dense with respect to each other, with some associated constant c.*

Proof. The intersection of $S \in \mathcal{S}$ and $D \in \mathcal{D}$ is sparse, and hence cannot contain an $(a+1)$-clique, and is also dense and thus cannot contain a $(b+1)$-stable set. Such a graph has its number of vertices bounded by the Ramsey number $c = R(a+1, b+1)$ (cf. [46]). □

This result makes it easy to find additional examples of sparse and dense classes. Examples of sparse classes (with respect to cliques) are stable sets, bipartite graphs, $(c+1)$-clique-free graphs, planar graphs, and c-colorable graphs. (The last one is NP-complete for $c \geq 3$, and the remaining are polynomial time solvable.) Examples of dense classes (with respect to stable sets) can be obtained by taking complements, e.g., cliques, cobipartite graphs, graphs without $(c+1)$-stable sets, complements of planar graphs, and complements of c-colorable graphs. Combining any one of the former with any one of the latter produces a sparse-dense pair of families. In particular, this shows that the earlier example where sparse sets have clique-size at most a and dense sets have stable-set-size at most b satisfies the requirements.

4. Separators.

Some partition problems on G can be solved by considering all maximal cliques of G: for example, to decide if a graph is a split graph we can test the complements of all maximal cliques to see if any are stable. Indeed, if C is a clique and S a stable set, some maximal clique (or maximal clique with one vertex deleted) of G always "separates" C from S in the sense that it contains C and is disjoint from S. Unfortunately, in general the number of maximal cliques is exponential. (The graph $K_{2n} - nK_2$ has 2^n maximal cliques.) The following result of Lovász asserts that there always exists a *subexponential* family of sets that separate cliques and stable sets. Such separators turn out to be surprisingly useful for list M-partitions.

Let G be a graph. A family $\mathcal{E}$ of subsets of $V(G)$ is said to *separate* cliques and stable sets if for any pair of disjoint sets C, S, such that C is a clique and S is a stable set in G, some $E \in \mathcal{E}$ contains C and is disjoint from S.

Lovász's subexponential bound turns out to be quasi-polynomial, as do our generalizations of it. It is not known whether or not the bound can be improved to a polynomial.

THEOREM 4.1 (see [37]). *Every graph with n vertices has a family of $n^{\frac{1}{2}\log n}$ sets that separate cliques and stable sets.*

Moreover, such a family can be found in time $n^{\frac{1}{2}\log n}$ times a polynomial in n.

Proof. The bound actually given in [37] is $2^{\binom{1+\log(n+1)}{2}} \leq n^{\frac{1}{2}\log n}$. It is couched in terms of communication complexity (as a communication game); cf. also [26]. Here we describe a more combinatorial view of the proof. For simplicity we will prove only a weaker bound of $n^{\log n}$ sets (found in time $n^{\log n}$ times a polynomial in n). The bound as given in [37], and as claimed in the theorem, is a direct corollary of our Theorem 4.7, obtained by setting $t = 2$. We choose to give the proof (of the weaker bound) in detail because it will allow us to explain how the proof needs to be modified to obtain our generalizations.

The idea of the proof is to obtain a family of sets E which are sufficient to separate cliques and stable sets, and which can be described "concisely"—and hence are not too numerous.

Suppose C is a clique and S is a disjoint stable set in G. A *valid encoding* of the pair C, S will be a sequence $v_1, v_2, \dots, v_k$ of vertices of G obtained as follows.

Let $G_0 = G$. At any stage, n_i will denote the number of vertices of the graph G_i.

Suppose G_{i-1} has already been defined. Then we define v_i and G_i by either of the following two pairs of rules:

- v_i is a vertex of S whose degree in G_{i-1} is greater than $n_{i-1}/2$,
- and G_i is the graph obtained from G_{i-1} by deleting v_i and all its neighbors;

or

- v_i is a vertex of C whose degree in G_{i-1} is smaller than or equal to $n_{i-1}/2$,
- and G_i is the graph obtained from G_{i-1} by deleting v_i and all vertices which are *not* neighbors of v_i.

Since at each step we remove more than half of the vertices, G_i becomes empty for $i > \log n$, and we may assume that $k \leq \log n$. (All logarithms in this paper are base two.) At the end of the process all degrees in C are too high and all degrees in S too low.

We now claim that any valid encoding of a pair C, S determines a set E which contains C and is disjoint from S. Equivalently, we will find two complementary sets $E = C^+$ and $\overline{E} = S^+$ such that C^+ contains C and S^+ contains S. To obtain C^+, S^+ we *decode* the sequence $v_1, v_2, \dots, v_k$ as follows.

Let $C_0^+ = S_0^+ = \emptyset$. At any stage, C_i^+, S_i^+ will be disjoint, G_i will be the graph obtained from G by deleting C_i^+ and S_i^+, and n_i will denote the number of vertices of G_i.

If C_{i-1}^+, S_{i-1}^+ have already been defined, we consider the degree d of v_i in G_i.

- If $d > n_i/2$, then we add v_i to S_{i-1}^+ and all its neighbors to C_{i-1}^+, thus forming new C_i^+, S_i^+;
- otherwise ($d \leq n_i/2$), we add v_i to C_{i-1}^+ and all the vertices that are not its neighbors to S_{i-1}^+, creating in this way new C_i^+, S_i^+.

Once all v_i have been processed, we form C^+ by adding to C_k^+ all the remaining vertices of G_k of high degree in G_k, that is, of degree in G_k greater than $n_k/2$, and form S^+ by adding to S_k^+ all the other vertices (of degree at most $n_k/2$ in G_k). Note that S^+ is the complement of C^+.

Since the decoding process reverses the steps of the encoding, the resulting set C^+ contains C, and the resulting set S^+ contains S. Indeed, if $v = v_i \in C$ for some i, it was chosen as v_i since its degree in G_{i-1} was high, and hence is placed in C^+ in the decoding process. A similar argument applies if $v = v_i \in S$. Otherwise the degree of $v \in C$ in G_k is low, and the degree of $w \in S$ is high, so once again they are correctly placed in C^+, S^+, respectively.

Let $\mathcal{E}$ denote the set of all sets C^+ produced by this decoding process from all possible sequences $v_1, v_2, \ldots, v_k, k = \lceil \log n \rceil$. Then $\mathcal{E}$ separates cliques and stable sets, since for each clique C and stable set S some sequence is the encoding of C, S. Moreover, $\mathcal{E}$ has at most $n^k = n^{\log n}$ elements. □

We remark that to obtain the better bound $(2^{\binom{1+\log(n+1)}{2}} \leq n^{\frac{1}{2}\log n})$ we would describe the separators by binary sequences, as is explained in the proof of Theorem 4.7. (Recall that the better bound actually follows from Theorem 4.7 by letting $t = 2$.)

We have several generalizations of the theorem, which we will use to solve certain M-partition problems.

Let G be a graph. A *clique-pair* (or a *skew set*) in G is a pair of disjoint sets A, B of vertices of G such that each $a \in A$ is adjacent in G to each $b \in B$. A *stable-pair* (or a *disconnected set*) in G is a pair of disjoint sets A, B such that no $a \in A$ is adjacent in G to any $b \in B$. Note that when $A = A_i, B = A_j$ are parts of an M-partition, then a clique-pair A_i, A_j corresponds to $m_{i,j} = 1$ and a stable pair A_i, A_j to $m_{i,j} = 0$. Thus cliques and stable sets are 1's and 0's (respectively) *on the diagonal* of M, and clique-pairs and stable-pairs are 1's and 0's (respectively) *off the diagonal* of M.

Let G be a graph. We say that a family $\mathcal{E}$ of subsets of $V(G)$ *separates cliques and stable-pairs* if for any pair $C, (A, B)$, where C is a clique and (A, B) is a stable-pair, such that C and $A \cup B$ are disjoint, some $E \in \mathcal{E}$ contains C and is disjoint from A *or* disjoint from B. Similarly, we say that $\mathcal{E}$ *separates clique-pairs and stable-pairs* if, for any pair $(A, B), (C, D)$, where (A, B) is a stable-pair and (C, D) is a clique-pair, such that A, C are disjoint and B, D are disjoint, some $E \in \mathcal{E}$ contains C and is disjoint from A, or it contains D and is disjoint from B.

THEOREM 4.2. *Every graph with* n *vertices has a family of* $n^{\log n}$ *sets that separate clique-pairs and stable-pairs.*

Moreover, such a family can be found in time $n^{\log n}$ *times a polynomial in* n.

Proof. Suppose that A, B is a stable pair, and C, D is a clique pair, in a graph G, and that $A \cap C = B \cap D = \emptyset$. We again define a valid encoding. Having seen the complete details above, we make the description here more concise. Thus a valid encoding of the pair $(A, B), (C, D)$ will be a sequence $v_1, v_2, \ldots, v_k$ of vertices of G obtained as follows: There will be two auxiliary sets U, W of vertices, initially both equal to $V(G)$. At each stage i, we define the vertex $v_i \in U$ to be either

- a vertex of A of *high degree* in W, i.e., adjacent to more than one half of the vertices in W,
- and remove from U the vertex v_i, and remove from W all the neighbors of v_i;

or

- a vertex of C of *low degree* in W, i.e., adjacent to at most one half of the vertices in W,
- and remove from U the vertex v_i, and remove from W all its nonneighbors.

Since the size of the set W is halved at each stage, we may again assume that $k < \log n$. At the end of the process we again have all vertices of U in A have their degree in W too low and all vertices of U in C have their degree in W too high.

The decoding process is a little different. We shall be building two complementary pairs of sets, $A^+, B^+ = \overline{A^+}$ and $C^+, D^+ = \overline{C^+}$, such that $A \subseteq A^+, C \subseteq C^+$ or $B \subseteq B^+, D \subseteq D^+$. Initially all four sets A^+, B^+, C^+, D^+ are empty. We also have auxiliary sets U, W, both initially equal to $V(G)$, similar to the ones from the encoding procedure. We process the vertices $v_1, v_2, \ldots, v_k$ in this order. Once v_{i-1} has been processed, we consider the degree of v_i with respect to W.

- If v_i has high degree in W, we place it in A^+ and put all its neighbors in D^+, removing v_i from U and its neighbors from W.
- If v_i is of low degree, we place it in C^+ and put all its nonneighbors in B^+, removing v_i from U and its nonneighbors from W.

Note that A^+, C^+ are disjoint, and so are B^+, D^+ (but A^+ could have common elements with either B^+ or D^+).

Once all v_i have been processed, one of two things can happen: Either W has become empty, which means that every vertex is either in B^+ or in D^+, and so we have a pair of complementary sets B^+, D^+ with $B \subseteq B^+, D \subseteq D^+$, or W is still nonempty. In the latter case we know that we can place all vertices of high degree in W into the set C^+ and all vertices of low degree in W into the set A^+; thus every vertex belongs to either C^+ or A^+, and so we have a pair of complementary sets A^+, C^+ with $A \subseteq A^+, C \subseteq C^+$, as claimed.

Let $\mathcal{E}$ be the family of all sets C^+ and D^+ obtained from all sequences $v_1, v_2, \ldots, v_k$. It follows that $\mathcal{E}$ separates clique-pairs and stable-pairs. □

An argument similar to that given for Theorem 4.2 will show the following.

Theorem 4.3. *Every graph with n vertices has a family of $n^{\log n}$ sets that separate cliques and stable-pairs.*

Moreover, such a family can be found in time $n^{\log n}$ times a polynomial in n. □

There is an important special case of this last theorem. For cliques and stable-pairs that *partition* the vertices of G, there is a polynomial separating family.

Theorem 4.4. *For every graph G with n vertices there exists a family of n sets, which separates all cliques C and all stable-pairs (A, B) with the property that A, B, C partition $V(G)$.*

Moreover, such a family of separators can be found in polynomial time.

Proof. Let G' be a minimal chordal extension of G, and let $v_1, v_2, \ldots, v_n$ be a perfect elimination ordering of G'. A minimal chordal extension of an arbitrary graph, and a perfect elimination ordering of a chordal graph, can be found in polynomial time [32]. It follows from the definition of a perfect elimination ordering that, for each $i = 1, 2, \ldots, n$, the set E_i consisting of v_i and all $v_j, j \geq i$, adjacent to v_i induces a clique in G'. Moreover, each clique of G' is contained in one of the cliques E_i, namely one with i being the first subscript such that v_i is present in the clique. We claim that the family $\mathcal{E} = \{E_1, E_2, \ldots, E_n\}$ satisfies the statement of the theorem.

Thus suppose C is a clique in G and (A, B) is a stable-pair in G such that A, B, C partition $V(G)$. Since G' is a minimal chordal extension of G, it cannot have an edge joining a vertex of A to a vertex of B [42]. (Indeed, G' with all such edges deleted will still be a chordal extension of G, since any cycle in it that contains both a vertex of A and a vertex of B goes twice through the clique C, and hence has a chord.) Thus (A, B) is also a stable pair in G', and, of course, C is also a clique in G'. Thus some E_i contains C and is disjoint from A or from B. □

This result illustrates that it is sometimes possible to find *polynomial* separating families. It is not known whether the quasi-polynomial bounds in Theorems 4.1, 4.2, and 4.3 can be improved to polynomial bounds.

Here is how we can use these results to reduce certain complex list partition problems to simpler ones.

COROLLARY 4.5. *Suppose M has $XZ = 0$ and $YW = 1$. Then the list M-partition problem reduces to $n^{\log n}$ instances, each of which has no list containing $\{X, Y\}$ or no list containing $\{Z, W\}$.*

In the special case $Z = X, W = Y$, i.e., $X = 0$ and $Y = 1$, the number of instances can be reduced to $n^{\frac{1}{2}\log n}$ (Theorem 4.1).

In the special case $X = Y$, i.e., $XZ = 0$ and $XW = 1$, the number of instances can be reduced to just $n + 1$, with one instance having no list containing X and n instances having no list containing both Y and Z (Corollary 2.4).

Proof. Suppose first that X, Z, Y, W are all different. Then any M-partition of an input graph G contains the clique-pair (Y, W) and the disjoint stable-pair (X, Z). According to Theorem 4.2 there is a family of $n^{\log n}$ sets E that separate all clique-pairs from all stable-pairs. For each set E we obtain two instances—in one we remove X from all vertices in E and Y from all vertices not in E, and in the other we remove Z from all vertices in E and W from all vertices not in E. The other cases are treated similarly. □

We may define separators also in the sparse-dense model: A family $\mathcal{E}$ *separates dense sets and sparse sets* if for any pair of disjoint sets D (dense) and S (sparse) there is a set $E \in \mathcal{E}$ which contains D and is disjoint from S.

The next result concerns the case when sparse subgraphs are the a-colorable subgraphs and dense subgraphs are the complements of b-colorable subgraphs.

THEOREM 4.6. *Every graph with n vertices has a family of $n^{\frac{1}{2}ab\log n}$ sets that separate the a-colorable subgraphs and the complements of b-colorable subgraphs.*

Moreover, such a family can be found in time $n^{\frac{1}{2}ab\log n}$ times a polynomial in n.

Proof. We have already observed that a sparse graph can meet a dense graph in at most $c = ab$ vertices.

We know that $n^{\frac{1}{2}\log n}$ separators E are sufficient to separate each stable set from each clique. If we separate each of the a stable sets from a sparse subgraph and each of the b cliques from a dense subgraph, we obtain $c = ab$ such sets E. We can then construct a separator E' for the sparse and dense subgraphs by taking, for each of the b cliques, the intersection of the a separators E corresponding to the a stable sets and then letting E' be the union of the b intersections corresponding to the b cliques. Since there are at most $n^{\frac{1}{2}\log n}$ separators E, and the separator E' is constructed from $c = ab$ such separators, the $n^{\frac{1}{2}c\log n}$ bound follows. □

Our last generalization concerns the case when sparse subgraphs are the $(a+1)$-clique-free subgraphs, and dense subgraphs are the $(b+1)$-stable-set-free subgraphs. Note that if we know that all stable sets are sparse and all cliques are dense, then sparse graphs are $(c+1)$-clique-free, and dense graphs are $(c+1)$-stable-set-free. Thus the following result can be used in all such situations; in particular, it can be used when $a = b = 1$, i.e., for separating cliques and stable sets. We take this opportunity also to refine the arguments to obtain the better bounds.

Instead of using sequences of vertices to describe the separators, we shall be using binary sequences—we simply represent each vertex by a binary sequence. The number of such sequences is then 2 power the length of the sequence.

The bound in the theorem is less than $2^{[\log^{(a+b)} n]/[(a+b)!]}$, which for the case $a = b = 1$ equals $2^{[\log^2 n]/2} = n^{\frac{1}{2}\log n}$.

THEOREM 4.7. *Every graph with n vertices has a family of $2^{C(a+b,n)}$ sets that separate the $(a+1)$-clique-free subgraphs and the $(b+1)$-stable-set-free subgraphs,* *where*

$$C(t,n) \leq \binom{t+\log(n+1)}{t} - 1 - \log(n+1)$$

is the solution of the recurrence

$$C(t,0) = 0,$$

$$C(1,n) = 0,$$

$$C(t,n) = \log(n+1) + \max_{n/2<d<n} C(t-1,d) + C(t,n-d-1).$$

Moreover, such a family can be found in time $2^{C(a+b,n)}$ *times a polynomial in* n.

Proof. We shall encode the sparse-dense pairs by binary sequences. Equivalently, we may talk of sequences of vertices as before, but we count each vertex as having a certain length. In fact, for this proof the sequences will use one additional special symbol, %. Thus, together with the n vertices of the input graph, we will have $n+1$ different symbols, and we will encode these by giving each symbol a different binary sequence of length $\log(n+1)$. With this measure of length, we shall show how to represent separators by sequences of length $C(a+b,n)$.

The description is as follows. Suppose G is a given graph, and S, D is a disjoint pair of sets, where S is $(a+1)$-clique-free and D $(b+1)$-stable-set-free.

Suppose first that there is a vertex $v \in S$ of degree $d > n/2$. We shall describe the separator by first giving the binary sequence for v, followed by two binary sequences, one describing the pair $S' = S \cap N, D' = D \cap N$, where N is the set of neighbors of v, and the other describing the pair $S'' = S - S', D'' = D - D'$. In the decoding process, we will be able to tell that $v \in S$ and recursively decode the two subsequences to produce a correct separator for S, D in G. Note that the first sequence has length at most $C(a-1+b,d) \leq C(a-1+b,n)$, since S' must be a-clique-free (being a subset of S and completely adjacent to $v \in S$). On the other hand, the second sequence has length at most $C(a+b,n-d-1) \leq C(a+b,\lfloor n/2 \rfloor)$.

Similarly, if there is a vertex $w \in D$ of degree $d \leq n/2$, then the description will start with the binary sequence for w, followed by two sequences, one for the $n-d-1$ nonneighbors of w and the other one for the d neighbors of w. The lengths of these sequences are again $C(a+b-1,n-d-1) \leq C(a+b-1,n)$ and $C(a+b,d) \leq C(a+b,\lfloor n/2 \rfloor)$. Here we have used the fact that if w is not adjacent to any of the vertices in D, then removing it decreases the size of the largest stable set by one.

If neither v or w can be found, then we can define the separator to consist of all the vertices of degree greater than $n/2$. It is easy to see that this set contains D and is disjoint from S. We shall use the special symbol % to indicate in the sequence that this is the case. (We need such an indication when recursively decoding the sequence.)

If $a+b=1$, then $a=0$ and the sparse graph is empty, or $b=0$ and the dense graph is empty.

Thus, we have the recurrence stated in the theorem, which can be bounded by

$$C(t,n) \leq \log(n+1) + C(t-1,n) + C(t,\lfloor n/2 \rfloor).$$

The bound on $C(t,n)$ follows by induction, with base cases

$$\binom{t+\log 1}{t} - 1 - \log 1 = 0,$$
$$\binom{1+\log(n+1)}{1} - 1 - \log(n+1) = 0$$

and inductive case

$$\binom{t+\log(n+1)}{t} = \binom{t-1+\log(n+1)}{t-1} + \binom{t+\log(\frac{n+1}{2})}{t}$$

and $\log(n+1) = 1 + \log(\frac{n+1}{2})$. □

5. Example applications. In this section we shall illustrate the general techniques of the preceding sections on some important examples. In the following section we treat *all* the remaining partition problems with at most four parts. We give the proofs in this section in full detail, allowing us to abbreviate the similar proofs given in the next section.

5.1. The list version of generalized split graphs. We first return to the case of generalized split graphs. Recall that G is an (a,b)-graph if its vertices can be partitioned into a stable sets $A_1, A_2, \ldots, A_a$ and b cliques $A_{a+1}, A_{a+2}, \ldots, A_{a+b}$, i.e., if and only if G has an M-partition where M is an $(a+b)$ by $(a+b)$ matrix with all off-diagonal entries equal to $*$ and with the first a diagonal entries equal to 0 and the last b diagonal entries equal to 1. We shall show how Theorem 3.1 implies a polynomial time algorithm to recognize (a,b)-graphs when $a, b \leq 2$. In fact, we shall solve the list version of these problems.

COROLLARY 5.1. *If both $a \leq 2$ and $b \leq 2$, then the list M-partition problem is polynomial time solvable. Otherwise it is NP-complete.*

Proof. First we note that if $a \geq 3$, then the list M-partition problem is NP-complete, since we can decide whether or not an input graph G is 3-colorable by endowing all its vertices with the list $\{1,2,3\}$ and asking whether or not it has a list M-partition. (If $b \geq 3$, the proof is similar.)

Thus assume that both $a \leq 2$ and $b \leq 2$. Let $\mathcal{S}$ be the class of all a-colorable graphs, and let $\mathcal{D}$ be the class of all graphs with b-colorable complements. Note that both classes can be recognized in polynomial time. According to Theorem 3.1 we can generate, in polynomial time, all sparse-dense partitions of any input graph G.

Suppose G with lists L is an instance of the list M-partition problem. For each sparse-dense partition of G, we update the lists of the vertices as follows: If v belongs to the sparse part $(A_1 \cup A_2 \cup \cdots \cup A_a)$ we remove all elements of $\{a+1, a+2, \ldots, a+b\}$ from $L(v)$ (if present). If v belongs to the dense part $(A_{a+1} \cup \cdots \cup A_{a+b})$ we remove all elements of $\{1, 2, \ldots, a\}$ from $L(v)$ (if present). The resulting instance has all lists of size at most two, and hence can be solved by 2-satisfiability (see Proposition 2.1). (Note that it is possible that some lists have become empty.) It is clear that G has a list M-partition with respect to the original lists L if and only if it has a list M-partition with respect to at least one of the modified lists. □

The corollary yields algorithms for all the polynomial generalized split graph recognition problems [5, 8]. Specifically, it gives polynomial time algorithms for the recognition of split graphs, (2, 1)-graphs, (1, 2)-graphs, and (2, 2)-graphs. All other (a,b)-graph recognition problems are NP-complete [5, 8].

5.2. The list clique cutset problem. The best known polynomial time solvable three-part partition problem is the clique cutset problem, i.e., $C = 1, AB = 0$, all others $*$. In [26] we have shown that the list version of this problem can be reduced, in polynomial time, to the list-free version solved by polynomial time algorithms of Tarjan [42] and Whitesides [47, 48]. Here we give a direct polynomial time algorithm for the list clique cutset problem. It is motivated by the original algorithms [42, 47, 48], and it follows from one of our separator theorems—Theorem 4.4.

COROLLARY 5.2. *There is a polynomial time algorithm for the list clique cutset problem.*

Proof. The theorem yields a polynomial size family $\mathcal{E}$ such that whenever an input graph G has a partition A, B, C with $C = 1, AB = 0$, some $E \in \mathcal{E}$ contains C and is disjoint from A or from B. Thus for each E from the family we will do two tests: In both tests, we remove C from all the lists of the vertices that do not belong to E. For the vertices that belong to E, we remove A in the first test and B in the second test. This ensures that if a partition exists, one of the tests will succeed. Each test can be performed in polynomial time by Proposition 2.1. □

5.3. The list skew cutset problem. The best known four-part partition problem is the skew cutset problem, i.e., $AC = 0$, $BD = 1$, and all others $*$. The complexity of this problem was a well-known open problem in the theory of perfect graphs [13]. In [26] we presented the first subexponential algorithm for the problem, strongly suggesting that it was not NP-complete. Since then, a polynomial algorithm has been found by one of us (Klein), with de Figueiredo, Kohayakawa, and Reed [29]. It is worth noting that although the algorithm uses a different technique from the ones given here, it still very much uses the flexibility of recursively reducing the problem by modifying the lists. Our algorithm, presented below, is a simple application of Theorem 4.2. Conceivably, this simplicity could be exploited for a possible *subexponential* algorithm for perfect graph recognition based on [12].

COROLLARY 5.3. *The list skew cutset problem can be solved in time $n^{\log n}$ times a polynomial in n.*

Proof. Since (B, D) is a clique-pair, and (A, C) is a disjoint stable-pair, we can apply Theorem 4.2. For each E from the family $\mathcal{E}$ generated by the theorem we perform two tests: The first test checks whether E contains B and is disjoint from A, thus deleting A from all the lists of the vertices that belong to E and deleting B from the vertices that do not belong to E. The second test assumes that E contains D and is disjoint from C, also updating the lists accordingly. During the first test, no list has both A and B. If at any point a vertex has a list of size one, we eliminate it and restrict its neighbors and nonneighbors accordingly. If no list becomes empty, we arrive at a situation where every vertex has a list of size at least two but never contains both A and B. This means that every list contains C or D. But $C = D = CD = *$, so we can freely choose to put all vertices to either C or D, according to their lists. The second test is done similarly. If both tests fail for all E from the family $\mathcal{E}$, then there is no solution by Theorem 4.2. The tests treats $2n^{\log n}$ cases with restricted lists (no lists containing $\{A, B\}$ or no lists containing $\{C, D\}$), each case being polynomial time solvable. □

5.4. The list two-clique cutset problem. As an application of Theorem 4.3 we give a quasi-polynomial bound for the *two-clique cutset problem*, that is, $AC = 0, B = D = 1$, all others equal to $*$. This example is also interesting because it illustrates how we can use a separator theorems twice. (This is a recurring theme in the general classification of matrices with $k = 4$.)

COROLLARY 5.4. *The list two-clique cutset problem can be solved in time $n^{2\log n}$ times a polynomial in n.*

Proof. Let $\mathcal{E}$ be a family of $n^{\log n}$ sets that separates cliques and stable pairs, as guaranteed by Theorem 4.3. If the input graph G admits a partition as specified above, then some E_1 will contain B and be disjoint from A or C, and some E_2 will contain D and be disjoint from A or C. For *any two* elements E_1, E_2 of $\mathcal{E}$, we shall make four tests: In the first test we assume that E_1 contains B and is disjoint from A, while E_2 contains D and is also disjoint from A; in the second test we assume E_1 contains B and is disjoint from A, while E_2 contains D and is disjoint from C. The third and fourth tests are defined similarly (assuming that E_1 is disjoint from C). As before, the tests are performed by correspondingly modifying the lists of the elements inside or outside of E_1, E_2, respectively. The second test results in all lists of size two, as does one of the last two tests; these can be solved again by Proposition 2.1. Consider the first test (the remaining test is done similarly). No list has both A, B and no list has both A, D. If there are any lists of size three, they must be $\{B, C, D\}$. We can again assume that there are no lists of size one. We are now in the following situation: If a list has no C, then it must be $\{B, D\}$. Since $C = BC = BD = CD = *$, we can complete the test by placing all vertices that have C in their list to C and checking that those vertices with lists $\{B, D\}$ can be partitioned into two cliques, i.e., that the graph they induce has a bipartite complement. □

We have been informed [10] that a polynomial algorithm for this problem has been constructed along the lines of [29].

6. Classifications for small matrices M.

Recall that most of our motivating examples of M-partitions dealt with small values of k. Split graphs have $k = 2$; stable set partition, clique partition, and homogeneous set have $k = 3$; skew partition and the problem of Winkler have $k = 4$; and so on (cf. Figures 2 and 3). This suggests a systematic investigation of M-partition problems with small k. Here, we focus on the case $k \leq 4$.

When $k = 2$, all list M-partition problems are polynomial time solvable by Proposition 2.1.

THEOREM 6.1. *Suppose the size of M is $k = 3$. Then the list M-partition problem is NP-complete when M or its complement is the matrix of 3-coloring or the stable cutset problems (Figure 3), and it is polynomial time solvable otherwise.*

Proof. The NP-complete cases are standard results [31, 28].

Consider a matrix M with rows A, B, and C and connections AB, AC, and BC, which is different from the four exceptional matrices described in the theorem. We may assume that M is connected; thus at most one of the connections is 0, and, by complementation, at most one of the connections is 1. We may also assume that no row has both a 0 and a 1; cf. Corollary 2.4 and Proposition 2.1. In particular, we do not have both a connection of type 0 and a connection of type 1. Thus without loss of generality we may assume that $AC = BC = *$. In that case we may also assume that $C \neq *$; otherwise C dominates all other rows, and we can eliminate it as explained after Proposition 2.5.

If each part A, B, and C is of type either 0 or 1, and not all are the same type, then the problem is polynomial time solvable by Theorem 3.1, since after we have decided which vertices go to parts of type 0 and which to parts of type 1, we are left with each list of size one or two.

If $AB = *$ as well, then we also have $A, B \neq *$, and so we may assume that all three values A, B, and C are the same, either 0 or 1. This is impossible, as M is matrix different from the matrix of 3-coloring and its complement.

Up to complementation we may assume that $AB = 0$. We may also assume that $A, B \neq 1$; otherwise we have a row with both a 0 and a 1. If one of A and B dominates the other, we reduce all lists to size at most two. The only other possibility is that $A = B = *$. Since M is not the matrix of the stable cutset problem we conclude that $C = 1$; i.e., it is the matrix of the clique-cutset problem solved in the previous section. ∎

We now proceed to discuss matrices of size $k = 4$. It will be easier to deal first with matrices M which have no $*$'s on the main diagonal.

THEOREM 6.2. *Suppose M is of size $k = 4$, and assume it does not contain any $*$'s on the main diagonal.*

- *If M contains the matrix corresponding to 3-coloring or its complement, then the list M-partition problem is NP-complete.*
- *Otherwise the list M-partition problem is polynomial time solvable.*

Proof. The first statement follows from Propositions 2.6 and 2.7. Thus assume that M does not contain the matrix corresponding to 3-coloring or its complement.

If there are two parts of type 0 and two of type 1, we proceed as in the case of $k = 3$, solving, for each sparse-dense partition, the remaining problem by 2-satisfiability. If there are three parts of type 0 and one part of type 1 (a similar argument applies when there are three 1's and one 0), then we proceed the same way, since the three 0 parts do not yield 3-coloring and all other three-part problems without a diagonal 1 are polynomial time solvable. Once the vertices that go to the part of type 1 are known, we can remove them, modifying the remaining lists, and solve the three-part subproblem.

By complementation, we may now assume that all four parts are of type 0.

Suppose there are two disjoint connections of type 1, say $AB = CD = 1$. Then we may try to place two vertices—v into A and w into C: for vertices adjacent to both v and w we can remove A and C from the lists, for those nonadjacent to both we can remove B and D, for those adjacent to v but not to w we can remove A and D, and for those adjacent to w but not to v we can remove B and C. Thus we can reduce the problem to the following polynomial time solvable instances: One instance with lists without A, one instance with lists without C, and at most n^2 instances with lists of size at most two (corresponding to all possible choices of v, w).

Suppose next that three connections of type 1 are incident on the same part, say $AB = AC = AD = 1$. Then we can reduce the problem (by trying to place a vertex into A) to one instance with lists without A (polynomial time solvable since M does not contain 3-coloring), and at most n instances with all vertices having list $\{A\}$ or subset of $\{B, C, D\}$, which can be solved in polynomial time as a three-part subproblem.

On the other hand, if there are three connection of type 1 which form a triangle, say $AB = BC = AC = 1$, then we can reduce the problem to at most $n^2 + 2$ polynomial time solvable cases by trying to place one vertex in A and one in B.

Thus it remains to consider the case of at most two connections of type 1. If there are two such connections, then we may assume that they both touch on A, thus, say, $AB = AC = 1$. If $AD = 0$, then trying to place a vertex in A leads to $n+1$ polynomial time solvable instances. Thus we may assume that $AB = AC = 1, AD = *$. In this case D dominates B or C unless $BD = CD = 1$ or $BC = 1, BD = CD$. Thus we may assume that no list contains both B and D or no list contains both C and D. Now we can reduce each of these problems to $n + 1$ polynomial time solvable cases by trying to place a vertex in A. The same technique (trying to place a vertex to A) also works when $BD = CD = 1, BC = 0$, since then C dominates B. Note that we cannot have $BD = CD = BC = 1$, since M does not contain 3-coloring. Thus we are left with the case $BC = 1, BD = CD = 0$ (and $AB = AC = 1, AD = *$). In this case we can reduce the problem to one instance of lists without A, one instance of lists without B, and n^2 instances of a vertex v placed into A and a vertex w into B. The former two problems are polynomial time solvable; the latter problem can also be solved in polynomial time, since vertices adjacent to w and nonadjacent to v must map to A, while no other vertex can map to A—hence we can remove the vertices that map to A and solve the three-part subproblem.

Suppose there is only one connection of type 1, namely $AB = 1$. If $AX = BY = 0$ for some (possibly equal) X, Y, then we can again reduce the problem to $n^2 + 2$ polynomial time solvable instances by trying to place a vertex in A and a vertex in B. Thus we may assume that, say, $AB = 1, AC = AD = *$. Since M does not contain 3-coloring, we must have $CD = 0$, and it is easy to see that in the remaining cases one of C, D dominates the other; if no list contains both C and D, then the problem can be reduced to $n + 1$ instances of polynomial time solvable instances by trying to place a vertex in A or B. We note that a linear time algorithm for (the complement of) the problem $AB = 1, AC = AD = 1, BC = BD = 1, CD = 0$ (which is one of the problems considered in this paragraph) was given by Everett, Klein, and Reed [21].

If there are no connections of type 1, then we have a list homomorphism problem, solvable in polynomial time (see Proposition 2.8). □

Note that the M-partition problem (without lists) is trivial if there is a part of type $*$ (all vertices can be placed in it). Thus the theorem allows a complete classification of the M-partition problem without lists.

COROLLARY 6.3. *If the size of M is $k = 4$, then the M-partition problem (without lists) is*

- *NP-complete when M contains the matrix of 3-coloring or its complement, and no diagonal entry is $*$,*
- *and it is polynomial time solvable otherwise.*

Proof. The polynomial algorithms follow from the theorem and the above remark.

Suppose M contains the matrix of 3-coloring; say M is the matrix

$$\begin{pmatrix} 0 & * & * & x_1 \\ * & 0 & * & x_2 \\ * & * & 0 & x_3 \\ x_1 & x_2 & x_3 & y \end{pmatrix}$$

(When M contains the complement of the matrix of 3-coloring, the argument is similar.) By assumption, $y \neq *$.

Suppose first that $y = 1$. We prove the NP-completeness of M-partition by reducing to it the problem of 3-colorability. Thus suppose that G is a graph we would like to 3-color, and let G' consist of two disjoint copies of G. We claim that G is 3-colorable if and only if G' admits an M-partition. Indeed, if G is 3-colorable, then G' is also 3-colorable, and hence admits an M-partition (with all vertices in the first three parts). On the other hand, an M-partition of G' cannot place two vertices from different copies of G in the fourth part, since the fourth part is a clique. Thus all vertices of one copy of G are placed in the first three parts; i.e., G is 3-colorable.

Now suppose that $y = 0$. If any other $x_i = 0$, then the union of the ith part and the fourth part is a stable set, and a graph admits an M-partition if and only if it is

3-colorable. If any $x_i = 1$, then it is again the case that G is 3-colorable if and only if G' (from above) admits an M-partition. Indeed, an M-partition of G' cannot place both a vertex from the first copy of G to the ith part and a vertex from the second copy in the fourth part, since those parts are completely adjacent. Therefore at least one copy of G is 3-colored. □

We now turn to the general list M-partition problem, where M may have parts of type $*$. We are no longer able to classify the problems as NP-complete or polynomial, but we do prove that they all are NP-complete or quasi-polynomial. (Since the preliminary version of this paper [26], most of the quasi-polynomial cases have been proved to be polynomial [10].)

THEOREM 6.4. *Suppose the size of M is $k = 4$. Then the list M-partition problem is quasi-polynomial (possibly polynomial) or NP-complete.*

Proof. By Proposition 2.8, we can assume that M has at least one 0 and at least one 1. It turns out then that the only NP-complete problems are those mentioned earlier, namely those arising from matrices containing the matrix of 3-coloring or of stable cutset, or their complements. Recall that we use the shorthand XY to refer to the entry of M in row X and column Y (and write X for XX). The proofs below are written in an abbreviated style; the details can be filled out in a manner similar to the proof of Corollary 5.3.

The next two lemmas cover the cases where M has an off-diagonal 0 and an off-diagonal 1.

LEMMA 6.4.1. *Suppose $AC = 0$, $BD = 1$. Then list M-partition is quasi-polynomial or NP-complete.*

Proof. Note that this case includes the list skew cutset problem solved in quasi-polynomial time in Corollary 5.3, and the stable cutset problem proved NP-complete in [28]. The proof below is written in an abbreviated style; the full details could be written out in a manner similar to the proof of Corollary 5.3.

Since $AC = 0$ and $BD = 1$, we can assume that no list contains $\{A, B\}$ or no list contains $\{C, D\}$ ($n^{\log n}$ cases); also, we can assume no list contains $\{A, D\}$ or no list contains $\{B, C\}$ ($n^{\log n}$ cases). (These are obtained by applying Theorem 4.2 in two ways.) The four possibilities are similar, so say there is no $\{A, B\}$ and no $\{A, D\}$. (For instance the possibility that there is no $\{A, B\}$ and no $\{B, C\}$ corresponds to exactly the same situation for the complementary matrix $\overline{M}$.) In other words, all lists are contained in $\{A, C\}$ or in $\{B, C, D\}$. We then have the following:

$C \neq 1$, else place vertex in C, no $\{A, C\}$ (Corollary 2.4 applies, since $C = 1$ and $AC = 0$); hence drop A and solve the polynomial problem with the three parts B, C, D.

$C \neq 0$, else no $\{B, C\}$ or no $\{C, D\}$ ($n^{\log n}$ cases from Theorem 4.3, which applies as $C = 0$ and $BD = 1$), and we can solve these problems by 2-satisfiability; cf. Proposition 2.1.

Therefore we must have $C = *$. We also have the following:

$A \neq 1$, else place vertex in A, no $\{A, C\}$ (as $A = 1$ and $AC = 0$), drop A and consider the $\{B, C, D\}$ problem. If that problem is solvable in polynomial time, we have a quasi-polynomial algorithm; otherwise the restriction to $\{B, C, D\}$ is NP-complete, and we have an NP-complete problem.

$B \neq 0$, else place vertex in B, no $\{B, D\}$ (as $B = 0$ and $BD = 1$), solve the 2-satisfiability problem.

$D \neq 0$, else place vertex in D, no $\{B, D\}$ (as $D = 0$ and $BD = 1$), solve the 2-satisfiability problem.

$AB \neq 1$, else place vertex in A, no $\{B, C\}$ (as $AB = 1$ and $AC = 0$), solve the 2-satisfiability problem.

$AD \neq 1$, else place vertex in A, no $\{C, D\}$ (as $AC = 0$ and $AD = 1$), solve the 2-satisfiability problem.

$BC \neq 0$, else place vertex in B, no $\{C, D\}$ (as $BC = 0$ and $BD = 1$), solve the 2-satisfiability problem.

$CD \neq 0$, else place vertex in D, no $\{B, C\}$ (as $BD = 1$ and $CD = 0$), solve the 2-satisfiability problem.

So far, we have $AC = 0$, $BD = 1$, $C = *$, $A \neq 1$, $B \neq 0$, $D \neq 0$, $AB \neq 1$, $AD \neq 1$, $BC \neq 0$, $CD \neq 0$.

In addition we do not have both $BC = CD = *$; otherwise C dominates A, no $\{A, C\}$, drop A and consider the $\{B, C, D\}$ problem.

We may assume $BC = 1$ (by symmetry between B and D).

Now, we have no $\{A, C\}$ or no $\{B, C\}$ ($n^{\log n}$ cases as $AC = 0$ and $BC = 1$), so either drop A to get a $\{B, C, D\}$ problem or get a 2-satisfiability problem. ∎

Lemma 6.4.2. *Suppose $AB = 0$, $AD = 1$. Then list M-partition is quasi-polynomial or NP-complete.*

Proof. Place vertex in A, no $\{B, D\}$. Then we have the following:

$BC \neq 0$ and $CD \neq 1$, as the case of disjoint connections with value 0 and 1 was covered by the previous lemma.

$BC \neq 1$, else place vertex in B, no $\{A, C\}$ (as $AB = 0$ and $BC = 1$), solve the 2-satisfiability problem. So $BC = *$.

$CD \neq 0$, else place vertex in D, no $\{A, C\}$ (as $AD = *$ and $CD = 0$), solve the 2-satisfiability problem. So $CD = *$.

So far, we have $AB = 0$, $AD = 1$, $BC = *$, $CD = *$.

Suppose $C = *$. Then $AC \neq *$, else C dominates all and could be dropped. By symmetry under complementation, we can assume $AC = 0$. Also C dominates B, so no $\{B, C\}$. Place vertex in A, no $\{C, D\}$ (as $AC = 0$ and $AD = 1$). So a list can contain only one of $\{B, C, D\}$, solve the 2-satisfiability problem.

For the other case, $C \neq *$; by symmetry under complementation, we can assume $C = 0$. We also have the following:

No $\{A, C\}$ or no $\{C, D\}$ ($n^{\log n}$ cases as $C = 0$ and $AD = 1$). If no $\{A, C\}$, since no $\{B, D\}$, solve the 2-satisfiability problem. So no $\{C, D\}$. As a result, all lists are contained in $\{A, D\}$ or in $\{A, B, C\}$.

Now, no $\{A, D\}$ or no $\{A, B\}$ ($n^{\log n}$ cases as $AB = 0$, $AD = 1$), so either drop D to get a $\{A, B, C\}$ problem or get a 2-satisfiability problem. ∎

By these two lemmas, we can assume that 1 (or equivalently 0, by complementation) occurs only on the diagonal so that all off-diagonal entries are $0, *$.

We first consider the case where there are at least two 1s (on the main diagonal). For this case, we can assume that there is at least one 0 not on the main diagonal. Otherwise, if all off-diagonal entries are $*$, then if, say, $A = *$, then A dominates all other parts, and we obtain a size three problem on $\{B, C, D\}$; if none of A, B, C, D is $*$, then either they are two 0's and two 1's (polynomial by the sparse-dense technique) or we get an NP-complete problem by 3-colorability.

The next three lemmas consider the possible placements of the 0 connection with respect to the (at least two) 1 parts. Either the 0 connection is not incident on any of the two 1's or it is incident on one of them, or it is incident on both of them.

Lemma 6.4.3. *Suppose all off-diagonal entries are $0, *$, and $B = D = 1$, $AC = 0$. Then list M-partition is quasi-polynomial or NP-complete.*

Proof. There is no $\{A, B\}$ or no $\{B, C\}$ ($n^{\log n}$ cases as $B = 1$, $AC = 0$).

There is no $\{A, D\}$ or no $\{C, D\}$ ($n^{\log n}$ cases as $D = 1$, $AC = 0$).

If there is no list of size three, solve 2-satisfiability. A list of size three can be only $\{A, B, D\}$ or $\{B, C, D\}$. By symmetry, assume it is $\{B, C, D\}$. So all lists are contained in $\{A, C\}$ or in $\{B, C, D\}$.

We have the following:

$C \neq 1$, else place vertex in C, no $\{A, C\}$ (as $C = 1$ and $AC = 0$), drop A and get a three-part problem on $\{B, C, D\}$.

$C \neq 0$, else no $\{B, C\}$ ($n^{\frac{1}{2}\log n}$ cases as $C = 0$ and $B = 1$), solve 2-satisfiability. So $C = *$.

$BC \neq 0$, else place vertex in B, no $\{B, C\}$ (as $B = 1$ and $BC = 0$), solve 2-satisfiability. So $BC = *$.

$CD \neq 0$, else place vertex in D, no $\{C, D\}$ (as $D = 1$ and $CD = 0$), solve 2-satisfiability. So $CD = *$.

Now all $\{A, C\}$ vertices can be put in C without loss of generality, so drop A and get a three-part problem on $\{B, C, D\}$. □

Lemma 6.4.4. *Suppose all off-diagonal entries are* $0, *$*, and* $A = B = 1$*,* $AC = 0$*. Then list* M*-partition is quasi-polynomial or* NP*-complete.*

Proof. We can assume $D \neq 1$ and $CD \neq 0$ by the previous lemma, so $CD = *$.

Place a vertex in A, no $\{A, C\}$ (as $A = 1$ and $AC = 0$).

Also no $\{A, B\}$ or no $\{B, C\}$ ($n^{\log n}$ cases as $B = 1$, $AC = 0$).

Suppose first no $\{A, B\}$. All lists are contained in $\{A, D\}$ or in $\{B, C, D\}$.

We have the following:

$D \neq 0$, else no $\{A, D\}$ ($n^{\frac{1}{2}\log n}$ cases as $D = 0$, $A = 1$), so drop A and get a three-part $\{B, C, D\}$ problem. So $D = *$.

$AD \neq 0$, else place vertex in A, no $\{A, D\}$ (as $A = 1$ and $AD = 0$), so drop A and get a three-part $\{B, C, D\}$ problem. So $AD = *$.

$BD \neq 0$, else place vertex in B, no $\{B, D\}$ (as $B = 1$ and $BD = 0$), solve the 2-satisfiability problem. So $BD = *$.

But now, D dominates all vertices, so drop D and get a three-part $\{A, B, C\}$ problem.

Suppose instead no $\{B, C\}$. All lists are contained in $\{C, D\}$ or in $\{A, B, D\}$.

We have the following:

$D \neq 0$, else no $\{A, D\}$ ($n^{\frac{1}{2}\log n}$ cases as $D = 0$, $A = 1$), solve 2-satisfiability. So $D = *$.

$AD \neq 0$, else place vertex in A, no $\{A, D\}$ (as $A = 1$ and $AD = 0$), solve 2-satisfiability. So $AD = *$.

$BD \neq 0$, else place vertex in B, no $\{B, D\}$ (as $B = 1$ and $BD = 0$), solve 2-satisfiability. So $BD = *$.

But now, D dominates all vertices, so drop D and get a three-part $\{A, B, C\}$ problem. □

Lemma 6.4.5. *Suppose all off-diagonal entries are* $0, *$*, and* $A = C = 1$*,* $AC = 0$*. Then list* M*-partition is quasi-polynomial or* NP*-complete.*

Proof. By the last two lemmas, we can assume $AB = BC = AD = BD = CD = *$, and also $B \neq 1$, $D \neq 1$. If $B = *$, then B dominates all vertices, so drop B and solve $\{A, C, D\}$ problem. Similarly, if $D = *$, then D dominates all vertices, so drop D and solve $\{A, B, C\}$ problem. So $B = D = 0$. The problem is then the problem of recognizing $(2, 2)$-graphs, which is solved in polynomial time by Corollary 5.1. □

We are now left with the case where M has a single 1, and this 1 is on the diagonal, say $D = 1$.

For convenience, we define the *separating statement* for $x = A, B, C$ to be $xD = 0$ or $x = 0$. If this statement holds for x, then there is no $\{x, D\}$, either by placing a vertex in D (as $D = 1$ and $xD = 0$) or $n^{\frac{1}{2}\log n}$ cases for $D = 1$ and $x = 0$.

If all three separating statements hold, then there is no $\{x, D\}$ for $x = A, B, C$, so drop D and solve the three-part $\{A, B, C\}$ problem.

Suppose next exactly two separating statements hold, say, for A, B. So all lists are contained in $\{C, D\}$ or in $\{A, B, C\}$, and $C = CD = *$. The three possible cases are covered by the next three lemmas.

LEMMA 6.4.6. *Suppose there is a single* 1 *at* $D = 1$*, and* $AD = BD = 0$*,* $C = CD = *$*. Then list* M*-partition is quasi-polynomial or* NP*-complete.*

Proof. We may assume that not both $AC = BC = *$, else C dominates all vertices, so we can drop C and get a three-part $\{A, B, D\}$ problem.

Say $AC = 0$. Then $BC \neq 0$, else we have two components $\{A, B\}$ and $\{C, D\}$. So $BC = *$.

We have the following:

$A \neq 0$, else C dominates A, no $\{A, C\}$, solve 2-satisfiability problem. (So $A = *$.)

$AB \neq 0$, else C dominates B, no $\{B, C\}$, solve the 2-satisfiability problem. (So $AB = *$.)

$B \neq *$, else B dominates A, no $\{A, B\}$, solve 2-satisfiability problem. (So $B = 0$.)

The remaining problem on $\{A, B, C\}$ is the stable cutset problem, which is NP-complete. □

LEMMA 6.4.7. *Suppose there is a single* 1 *at* $D = 1$*, and* $A = B = 0$*,* $C = CD = *$*. Then list* M*-partition is quasi-polynomial or* NP*-complete.*

Proof. We may asume that not both $AC = BC = *$, else C dominates all vertices, drop C and get a three-part $\{A, B, D\}$ problem.

Say $AC = 0$. Then $AB = *$ and $BC = 0$, else C dominates A, no $\{A, C\}$, solve the 2-satisfiability problem.

Each connected component of the subgraph induced by the vertices with lists included in $\{A, B, C\}$ can go to $\{A, B\}$ or to $\{C\}$, but it can always be put in $\{C\}$ if C is in the lists. Solve the 2-satisfiability problem. □

LEMMA 6.4.8. *Suppose there is a single* 1 *at* $D = 1$*, and* $AD = B = 0$*,* $A = BD = *$*,* $C = CD = *$*. Then list* M*-partition is quasi-polynomial or* NP*-complete.*

Proof. $AC = 0$ and $AB = *$, else C dominates B, no $\{B, C\}$, solve 2-satisfiability problem.

If $BC = *$, the problem on $\{A, B, C\}$ is the stable cutset problem, which is NP-complete.

If $BC = 0$, then each connected component of the subgraph induced by the vertices with lists included in $\{A, B, C\}$ can go to $\{A, B\}$ or to $\{C\}$, but it can always be put in $\{C\}$ if C is in the lists. Solve the 2-satisfiability problem. □

The remaining case with a single 1 at $D = 1$ has at most one separating statement holding, say, for A.

LEMMA 6.4.9. *Suppose there is a single* 1 *at* $D = 1$*,* $B = BD = C = CD = *$*. Then list* M*-partition is quasi-polynomial or* NP*-complete.*

Proof. If $BC = *$, then one of B, C dominates all vertices and can be dropped, to obtain a three-part problem, unless $AB = AC = 0$ (and $A = *$), in which case the rows of B and C are identical, so B and C can be collapsed to a single part.

So $BC = 0$. We consider various cases of the values of (A, AB, AC):

$(0,*,*)$ is the stable cutset problem, which is NP-complete.

$(*,*,*)$: if also $AD=*$, A dominates all vertices and can be dropped to obtain a three-part problem on $\{B,C,D\}$. If $AD=0$, then place a vertex in D, no $\{A,D\}$ (as $D=1$ and $AD=0$). Also, we have no $\{B,D\}$ or no $\{C,D\}$ ($n^{\log n}$ cases, as $D=1$ and $BC=0$). Say there is no $\{B,D\}$. Then all lists are contained in $\{C,D\}$ or in $\{A,B,C\}$, and they can be assumed to be of size at last two. Now place all vertices with the list $\{C,D\}$ in C, and place all vertices with lists contained in $\{A,B,C\}$ in either A or C. Since $A=C=AC=*$, this is a solution.

$(*,0,*)$ with $AD=0$, $(0,0,0)$, and $(0,0,*)$: all three have no $\{A,D\}$ (place a vertex in D in the first case as $AD=0$ and $D=1$; in the other two cases we get $n^{\frac{1}{2}\log n}$ cases as $A=0$ and $D=1$). Also no $\{B,D\}$ or no $\{C,D\}$ ($n^{\log n}$ cases as $D=1$ and $BC=0$). So $\{A,B,C\}$ is the only 3-element list, but C dominates A in all three cases, no $\{A,C\}$, and we can solve the 2-satisfiability problem.

$(*,0,*)$ with $AD=*$ has the rows of A and C identical, so A and C can be collapsed to a single part.

$(*,0,0)$ has $n^{\log n}$ cases each of no $\{A,D\}$ or no $\{B,D\}$ (as $D=1$ and $AB=0$), no $\{A,D\}$ or no $\{C,D\}$ (as $D=1$ and $AC=0$), no $\{B,D\}$ or no $\{C,D\}$ (as $D=1$ and $BC=0$). So there is at most one of $\{A,D\}$, $\{B,D\}$, $\{C,D\}$, and all other lists contained in $\{A,B,C\}$. For lists contained in $\{A,B,C\}$, the connected components go to a single one of A, B, or C, and C can always be preferred if possible. Solve the resulting 2-satisfiability problem. □

This completes the proof of the theorem. □

7. Matrices of arbitrary size. Some results apply to arbitrarily sized matrices; e.g., Proposition 2.8 completely classifies list M-partition problems for matrices which do not contain any 0's or do not contain any 1's. We first observe that we can also deal with matrices which do not contain any $*$'s.

Corollary 7.1. *If M is a $(0,1)$-matrix, then the list M-partition problem is polynomial time solvable.*

Proof. Once we know from Theorem 3.1 which vertices are placed in parts of type 0 (stable sets) and which are placed in parts of type 1 (cliques), we can find the classes of the equivalence in which two vertices that are both in parts of type 0 are equivalent if they have the same open neighborhood, and two vertices that are both in parts of type 1 are equivalent if they have the same closed neighborhood. Having these equivalence classes in hand, we can easily check if a list partition exists. (Recall that M, and hence its size, is fixed.) □

Next, we shall discuss the possibility that Theorem 6.4 might extend to all matrices M.

In fact, based on the evidence of this paper, we make the following "quasi-dichotomy" conjecture:

Conjecture 7.1.1. *All list M-partition problems are quasi-polynomial or NP-complete.*

We have additional evidence supporting the conjecture. A list H-coloring problem can be thought of as a special case of a constraint satisfaction problem: Consider the fixed graph H as having one symmetric binary relation (the adjacency in H), and $2^{|V(H)|}$ unary relations, each corresponding to a subset of $V(H)$, and containing precisely the vertices in that subset. Then an instance G,L of the list H-coloring problem can be thought also as having one binary relation (the adjacency in G), and $2^{|V(H)|}$ unary relations, indexed by the subsets of $V(H)$, each containing all the vertices of G whose list is that subset. It is easy to see that a mapping of

$V(G)$ to $V(H)$ is a list H-coloring if and only if it preserves the binary relation as well as all the unary relations. Let $\mathcal{F}$ be a set of subsets of $V(H)$. The $\mathcal{F}$-*restricted* list H-coloring problem is the restriction of the standard list H-coloring problem in which the instances are restricted to G, L with $L(g) \notin \mathcal{F}$ for all $g \in V(G)$. (Some lists are forbidden.) The above translation still applies: The $\mathcal{F}$-restricted list H-coloring problem is polynomially equivalent to the corresponding constraint satisfaction problem with one binary and $2^{|V(H)|} - |\mathcal{F}|$ unary constraints. Indeed, any instance G of the $\mathcal{F}$-restricted list H-coloring problem gives rise, in the way described above, to an instance of the corresponding constraint satisfaction problem. For the converse, there is a slight technical complication not present in the unrestricted case: When an instance of the constraint satisfaction problem contains a vertex not included in any of the unary constraints, we would like to (but cannot) assign to it the list $V(H)$. This is easily resolved by modifying the corresponding constraint satisfaction problem to include an additional artificial vertex ∞ which does not belong to any of the unary constraints and is adjacent in the binary constraint to all other vertices, including itself.

THEOREM 7.2. *Each list M-partition problem can be reduced to at most $n^{c \log n}$ instances of restricted list H-coloring problems.*

Proof. Suppose $AC = 0, BD = 1$ in M. According to Corollary 4.5, we can reduce the list M-partition problem for an input graph with arbitrary lists L to $n^{\log n}$ (or fewer) instances, each of which contains no list containing both A and B or no list containing both C and D. Each of these can in turn be reduced to further $n^{\log n}$ (or fewer) instances in which there is no list containing both A' and B' or no list containing both C' and D' for some other pair $A'C' = 0, B'D' = 1$ in M. Since M is fixed, we obtain at most $(n^{\log n})^c = n^{c \log n}$ instances in which no pair of vertices v, v' has lists $L(v) \supseteq \{X, Y\}$ and $L(v') \supseteq \{Z, W\}$, where X, Y, Z, W are any parts such that $XZ = 0, YW = 1$ in M. Thus each of these instances has some set of forbidden lists.

Let M' be obtained from the matrix M by replacing all 1's by 0's. Now M' is a $(0, *)$-matrix, and hence corresponds to the adjacency matrix of a graph H (when $*$'s are replaced by 1's); the list M'-partition problem is precisely the list H-coloring problem.

Consider one such instance, a graph G with lists L. For any pair of vertices v, v' of G the pairs $X \in L(v), Y \in L(v')$ all have $XY = 0, *$, or they all have $XY = *, 1$. For vertices v, v' for which they are $0, *$, leave the edge or nonedge between v and v' unchanged. For those v, v' for which they are $*, 1$, revert an edge between v and v' to a nonedge, and a nonedge to an edge. It is easy to see that the original graph G has a list M-partition if and only if the modified graph has a list M'-partition, i.e., a list H-homomorphism. Note that the homomorphism problems inherit the list restrictions from the list M-partitions. $\square$

Feder and Vardi [27] proposed a dichotomy conjecture for all constraint satisfaction problems. In the context of restricted list H-coloring problems it can be stated as follows.

CONJECTURE 7.2.1. *All restricted list H-coloring problems are polynomial time solvable or NP-complete.*

Note that without the list restrictions, the conjecture is true by Proposition 2.8.

COROLLARY 7.3. *If Conjecture* 7.2.1 *is true, then Conjecture* 7.1.1 *is also true.*

Proof. If the above instances of restricted list H-coloring problems can be solved in polynomial time, we obtain a quasi-polynomial algorithm for list M-partition. If a restricted list H-coloring problem is NP-complete, then the original problem is also NP-complete. □

Acknowledgments. We are grateful to Donald Knuth, Jan Kratochvíl, and Romeo Rizzi for their interest and helpful suggestions.

Note added in proof. Using a recent result of A. Bulatov, the first two authors have now succeeded in proving Conjecture 7.2.1, and hence Conjecture 7.1.1 also follows (by Corollary 7.3). The authors of [10] have now verified that all quasi-polynomial list partition problems with $k = 4$ are in fact polynomial, with the sole exception of one "stubborn" problem for which no polynomial time algorithm is known. Finally, a polynomial time recognition algorithm for perfect graphs has now been obtained by M. Chudnovsky and P. Seymour, as well as by G. Cornuéjols, X. Liu, and K. Vuškovic.

REFERENCES

[1] N. Alon and M. Tarsi, *Colorings and orientations of graphs*, Combinatorica, 12 (1992), pp. 125–134.

[2] B. Aspvall, F. Plass, and R.E. Tarjan, *A linear time algorithm for testing the truth of certain quantified Boolean formulas*, Inform. Process. Lett., 8 (1979), pp. 121–123.

[3] C. Berge, *Farbung von Graphen deren samtliche bzw. deren ungerade Kreise starr sind (Zusammenfassung)*, Wiss. Zeit. der Martin-Luther Universitat Halle-Wittenberg Math. Natur. Reihe, 10 (1961), pp. 114–115.

[4] R.E. Bixby, *A composition for perfect graphs*, Ann. Discrete Math., 21 (1984), pp. 221–224.

[5] A. Brandstädt, *Partitions of graphs into one or two independent stable sets and cliques*, Discrete Math., 152 (1996), pp. 47–54.

[6] A. Brandstädt, Corrigendum to *Partitions of graphs into one or two independent stable sets and cliques*, Discrete Math., 186 (1998), p. 295.

[7] A. Brandstädt, *Partitions of Graphs into One or Two Stable Sets and Cliques*, Informatik-Berichte Nr. 105, 1/1991, FernUniversitat Hagen Technical Report, Hagen, Germany, 1991.

[8] A. Brandstädt, V.B. Le, and T. Szymczak, *The complexity of some problems related to graph 3-colourability*, Discrete Appl. Math., 89 (1998), pp. 59–73.

[9] M. Burlet and J. Fonlupt, *A polynomial algorithm to recognize a Meyniel graph*, Ann. Discrete Math., 21 (1984), pp. 225–252.

[10] K. Cameron, E.M. Eschen, C.T. Hoang, and R. Sritharan, *personal communication.*

[11] M. Conforti, G. Cornuéjols, G. Gasparyan, and K. Vušković, *Perfect graphs, partitionable graphs and cutsets*, Combinatorica, 22 (2002), pp. 19–33.

[12] M. Chudnovsky, N. Robertson, P. Seymour, and R. Thomas, *The Strong Perfect Graph Theorem*, manuscript, 2002.

[13] V. Chvátal, *Star-cutsets and perfect graphs*, J. Combin. Theory Ser. B, 39 (1985), pp. 189–199.

[14] V. Chvátal and N. Sbihi, *Bull-free Berge graphs are perfect*, Graphs Combin., 3 (1987), pp. 127–139.

[15] G. Conruéjols and W.H. Cunningham, *Compositions for perfect graphs*, Discrete Math., 55 (1985), pp. 245–254.

[16] G. Conruéjols and B. Reed, *Complete multi-partite cutsets in minimal imperfect graphs*, J. Combin. Theory Ser. B, 59 (1993), pp. 191–198.

[17] A. Cournier and M. Habib, *A new linear algorithm for modular decomposition*, in CAAP'94: International Colloquium, Lectures Notes in Comput. Sci. 787, Sophie Tison, ed., Springer-Verlag, Berlin, 1994, pp. 68–84.

[18] W.H. Cunningham, *A Combinatorial Decomposition Theory*, Ph.D. thesis, University of Waterloo, Waterloo, ON, Canada, 1973.

[19] W.H. Cunningham, *Decomposition of directed graphs*, SIAM J. Alg. Disc. Meth., 3 (1982), pp. 214–228.

[20] W.H. Cunningham and J. Edmonds, *A combinatorial decomposition theory*, Canad. J. Math., 32 (1980), pp. 734–765.

[21] H. Everett, S. Klein, and B. Reed, *An algorithm for finding clique-cross partitions*, Congr. Numer., 135 (1998), pp. 171–177.

[22] H. Everett, S. Klein, and B. Reed, *An algorithm for finding homogeneous pairs*, Discrete Appl. Math., 72 (1977), pp. 209–218.

[23] T. Feder and P. Hell, *List homomorphisms to reflexive graphs*, J. Combin. Theory Ser. B, 72 (1998), pp. 236–250.

[24] T. Feder, P. Hell, and J. Huang, *List homomorphisms and circular arc graphs*, Combinatorica, 19 (1999), pp. 487–505.

[25] T. Feder, P. Hell, and J. Huang, *Bi-arc graphs and the complexity of list homomorphisms*, J. Graph Theory, 42 (2003), pp. 61–80.

[26] T. Feder, P. Hell, S. Klein, and R. Motwani, *Complexity of graph partition problems*, in Proceedings of the 31st Annual ACM Symposium on Theory of Computing, Atlanta, GA, 1999, pp. 464–472.

[27] T. Feder and M.Y. Vardi, *The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory*, SIAM J. Comput., 28 (1998), pp. 57–104.

[28] C.M.H. de Figueiredo and S. Klein, *The NP-completeness of multipartite cutset testing*, Congr. Numer., 119 (1996), pp. 217–222.

[29] C.M.H. de Figueiredo, S. Klein, Y. Kohayakawa, and B. Reed, *Finding skew partitions efficiently*, J. Algorithms, 37 (2000), pp. 505–521.

[30] H. Fleischner and M. Stiebitz, *A solution of a colouring problem of P. Erdos*, Discrete Math., 101 (1992), pp. 39–48.

[31] M.R. Garey and D.S. Johnson, *Computers and Intractability*, W.H. Freeman, San Francisco, 1979.

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

[33] W. Gutjahr, E. Welzl, and G. Woeginger, *Polynomial graph-colorings*, Discrete Appl. Math., 35 (1992), pp. 29–45.

[34] P. Hell and J. Nešetřil, *On the complexity of H-colouring*, J. Combin. Theory Ser. B, 48 (1990), pp. 92–110.

[35] P. Hell, S. Klein, L. T. Nogueira, and F. Protti, *On generalized split graphs*, Electronic Notes in Discrete Mathematics, 7 (2001); available online from http://www.elsevier.nl/locate/jnlnr/05268.

[36] C.T. Hoang and V.B. Le, *Recognizing perfect 2-split graphs*, SIAM J. Discrete Math., 13 (2000), pp. 48–55.

[37] L. Lovász, *Communication complexity: A survey*, in Paths, Flows, and VLSI-Layout, B. Korte, L. Lovász, H. J. Prömel, and A. Schrijver, eds., Springer-Verlag, Berlin, 1990, pp. 235–265.

[38] L. Lovász, *Normal hypergraphs and the perfect graph conjecture*, Discrete Math., 2 (1972), pp. 253–267.

[39] G. MacGillivray and M.L. Yu, *Generalized partitions of graphs*, Discrete Appl. Math., 91 (1999), pp. 143–153.

[40] R.M. McConnell and J.P. Spinrad, *Linear-time modular decomposition and efficient transitive orientation of comparability graphs*, in Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, Arlington, VA, 1994, ACM, New York, 1994, pp. 536–545.

[41] L. Tito-Nogueira, *personal communication.*

[42] R.E. Tarjan, *Decomposition by clique separators*, Discrete Math., 55 (1985), pp. 221–232.

[43] A. Tucker, *Coloring graphs with stable sets*, J. Combin. Theory Ser. B, 34 (1983), pp. 258–267.

[44] K. Tucker-Nally, *List M-Partitions of Digraphs*, M.Sc. Thesis, Simon Fraser University, Burnaby, BC, Canada, 2003.

[45] N. Vikas, *Computational complexity of graph compaction*, in Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms, Baltimore, MD, 1999.

[46] D.G. West, *Introduction to Graph Theory*, Prentice-Hall, Upper Saddle River, NJ, 1996.

[47] S. Whitesides, *An algorithm for finding clique cutsets*, Inform. Process. Lett., 12 (1981), pp. 31–32.

[48] S. Whitesides, *A method for solving certain graph recognition and optimization problems, with applications to perfect graphs*, Ann. Discrete Math., 21 (1984), pp. 281–297.

SIAM J. Discrete Math.
Vol. 16, No. 3, pp. 479–498

LOCALIZED EIGENVECTORS FROM WIDELY SPACED MATRIX MODIFICATIONS*

XIANGWEI LIU[†], GILBERT STRANG[†], AND SUSAN OTT[†]

Abstract. We start with a large matrix A whose structure is simple, say, with unit entries on the first subdiagonal and superdiagonal. Its eigenvalues and eigenvectors are known. We modify A in M widely spaced rows and columns. Then the "new eigenvectors" are nearly a *sum of spikes* $x_j = t^{|j-r|}$ centered at the positions r of the modified rows. The new eigenvalues are given almost exactly by $\pm\sqrt{4+\mu^2}$, where μ is an eigenvalue of the M by M modification.

We extend this analysis to a larger class of structured matrices. For a banded Toeplitz matrix, our experiments show similar spikes centered around modified rows, and we have a conjecture on the structure of the new eigenvectors. For a single diagonal modification of the adjacency matrix of an infinite two-dimensional grid, we find the new eigenvalue from an elliptic integral (and we don't yet recognize the eigenvector).

Key words. localized eigenvectors, Toeplitz matrix, adjacency matrix

AMS subject classification. 15A12

PII. S0895480102409048

1. Introduction.

This paper is about the eigenvalues and eigenvectors of familiar structured matrices after changes in a small number of entries. The actual changes need not be small, so we refer to them as modifications rather than perturbations. The *number* of changes is small relative to the size of the matrix, because the modifications are required to be "widely spaced." They occur in entries that are far apart. They produce new eigenvectors that are localized near the components that correspond to the modified rows. By knowing the approximate form of those eigenvectors, we also determine a very close (and simple) approximation to the eigenvalues.

Imagine a large number of nodes around a circle. Edges go only to the two neighbors of every node. Each row of the adjacency matrix A of this cyclic graph has two 1's. The matrix is a circulant with 1's on the first subdiagonal and superdiagonal, coming from the neighbors to the left and right. Now add a few edges going "across" the circle so that the nodes involved are widely spaced. The modified graph has an adjacency matrix (symmetric if the added edges are undirected, but this is not required) with 1 in the i, j entry when an edge connects node i to node j. A typical example of our work is to find the "new" eigenvalues and eigenvectors of this modified matrix.

The second author mentioned in *SIAM News* (April 2000) the simplest case of this example. Only one undirected edge crosses the circle, from node i to a distant node j. This added edge modifies A by $a_{ij} = a_{ji} = 1$, in other words by a widely spaced submatrix with entries from

$$B = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

[*] Received by the editors May 29, 2002; accepted for publication (in revised form) January 16, 2003; published electronically May 22, 2003.

http://www.siam.org/journals/sidma/16-3/40904.html

[†] Department of Mathematics, MIT, Cambridge, MA 02139 (xwliu@alum.mit.edu, gs@math.mit.edu).

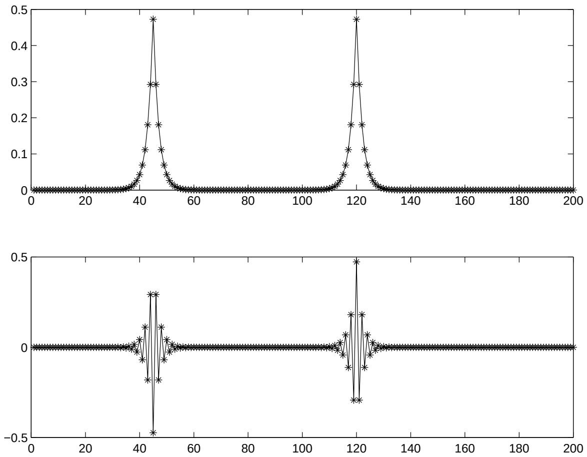


FIG. 1. *The eigenvectors for the maximal and minimal eigenvalues of the adjacency matrix of a* 200*-node cycle with an edge added between nodes* $i = 45$ *and* $j = 120$.

The two new 1's in the modified matrix are far from the main diagonal. *The two new eigenvalues are almost exactly* $\sqrt{5}$ *and* $-\sqrt{5}$. The corresponding eigenvectors show a sum or difference of two spikes, as in Figure 1, centered at the positions i and j connected by the "shortcut edge." The remaining eigenvalues stay in the interval $[-2, 2]$ that contains all eigenvalues of the original A. Their eigenvectors still oscillate like the original eigenvectors, but orthogonality to the new ones produces the pinching that is illustrated by Figure 2.

This brief report in *SIAM News* brought suggested proofs from three friends: Beresford Parlett, Bill Trench, and Jackie Shen. All four approaches, including ours, are different. Shen connected the problem to the theory of perturbed Schrödinger operators, and our work can be seen as a small contribution (possibly not new) to that established theory. The first section studies the effect of such a modification on

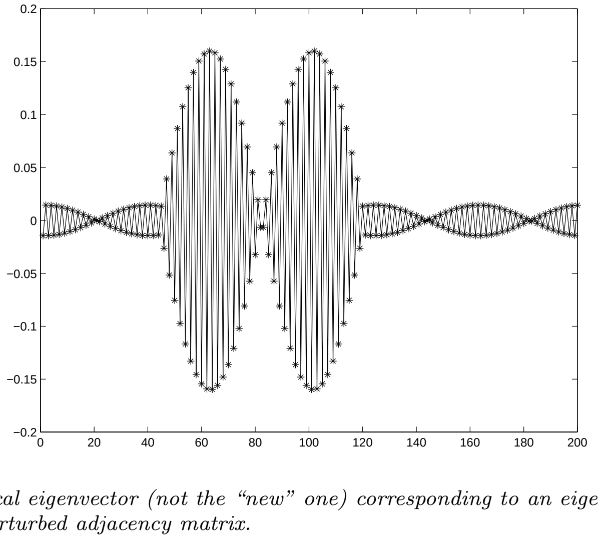


FIG. 2. *A typical eigenvector (not the "new" one) corresponding to an eigenvalue in the range of* $[-2, 2]$ *for the perturbed adjacency matrix.*

a string of nodes, and we find the following formula linking the (nearly exact) new eigenvalues λ to the eigenvalues μ of B:

$$\lambda = \operatorname{sign}(\mu)\ \sqrt{4+\mu^2}.$$

The result is equally true for a line or a circle of nodes.

In our first example, the rank two perturbation from B above has eigenvalues $\mu = 1$ and -1, confirming that $\lambda = \sqrt{5}$ and $-\sqrt{5}$. In the two localized eigenvectors, the heights of the "spikes" are given by the eigenvectors (c, c) and $(-c, c)$ of B. We also determine the ratio t between neighboring entries near the spikes (a smaller t means a sharper spike and a more localized eigenvector). This pattern extends to any widely spaced modification by a nonsingular B.

Sections 4 and 5 of this paper extend the theory beyond the line or circle of nodes and their particular adjacency matrix A. Similar isolated eigenvalues and "spiked" eigenvectors can also be found when the underlying matrix is a general Toeplitz matrix, or the adjacency matrix of a two-dimensional grid, provided that the modifications are widely separated in an appropriate way.

Related questions of perturbed Toeplitz eigenvalues have been investigated by Boettcher, Embree, and Sokolov [1], [2], who study the spectrum of $A + B$ when B has only one nonzero entry. Their work focuses on nonsymmetric finite and infinite-dimensional matrices. The variety of phenomena they have discovered is amazing. Unlike our present work, they do not consider the effect of the perturbation on eigenvectors.

2. The model problem. We start with an infinite line of nodes (the graph has a node at every integer). Its adjacency matrix A has 1's on the first subdiagonal and first superdiagonal: $a_{i,i-1} = a_{i,i+1} = 1$ for $-\infty < i < \infty$. The modification of A will be governed by an M by M matrix B, which need not be symmetric. We choose M widely spaced indices $r_1 < \cdots < r_M$; the differences between these indices all exceed a number $L \gg 1$. Then the i, j entry of B is added to the r_i, r_j entry of A. By a terrible abuse of notation, we call the modified matrix $A + B$. Our problem is to estimate the "new" eigenvalues and eigenvectors after the modification:

$$(A+B)x = \lambda x. \tag{1}$$

First, suppose that B is just a 1 by 1 matrix. The single real number b appears in the rth diagonal entry of the modified $A + B$. When A is finite, we always assume that modifications are far from the boundary ($r_1 \geq L$ and $r_m \leq n - L$).

Figure 3 shows the spectrum and the localized eigenvector corresponding to the isolated eigenvalue. This eigenvector decays very quickly and centers at the position $r = 50$ of the modification. From the logplot in Figure 4, we can see that the eigenvector is exponentially decaying with a constant exponent. Thus, we assume that the eigenvector is a spike centered at $r = r_1$ with $x_r = 1$. The "spike ratio" between neighboring entries is denoted t, with $|t| < 1$. Then the jth component of this eigenvector is $t^{|j-r|}$. Substitute this form of x into (1), and let $R = (A + B)x - \lambda x$ be the residual. There are three cases for the entries of R.

1. For nodes other than $j = 1$, r, and n, there is no contribution from B or from the boundary:

$$R_j = x_{j-1} + x_{j+1} - \lambda x_j = t^{|j-r|}\left(t + \frac{1}{t} - \lambda\right).$$

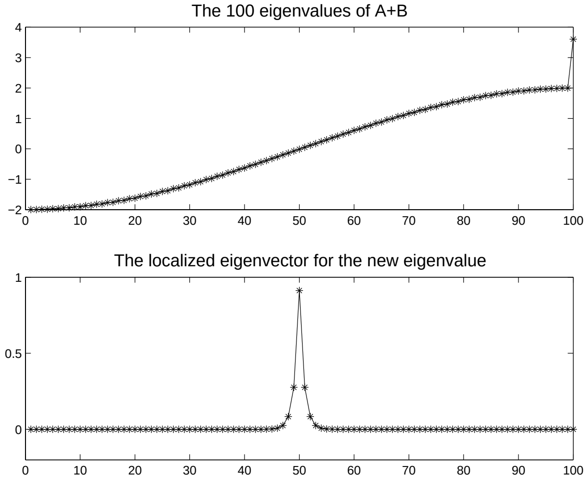


FIG. 3. *The spectrum of $A + B$ (notice λ_{100}) and the eigenvector corresponding to that isolated eigenvalue $\lambda \approx \sqrt{4+9}$ for $n = 100$, $r_1 = 50$, and $b = 3$.*

2. For node r, the entry has an extra term from B:

$$R_r = x_{r-1} + bx_r + x_{r+1} - \lambda x_r = 2t + b - \lambda.$$

3. Boundary nodes $j = 1$ and n have only one neighbor, and the residuals are $R_1 = t^{r-2} - \lambda t^{r-1}$ and $R_n = t^{n-r-1} - \lambda t^{n-r}$.

Since $|t| < 1$ and $1 \ll r \ll n$, the two boundary residuals will be of order t^L, $L = \min\{r - 1, n - r\}$. For $L \gg 1$, we focus on the typical case R_j and the special case R_r. These residuals are zero if

$$\lambda = t + \frac{1}{t} \quad \text{and} \quad \lambda = 2t + b \tag{2}$$

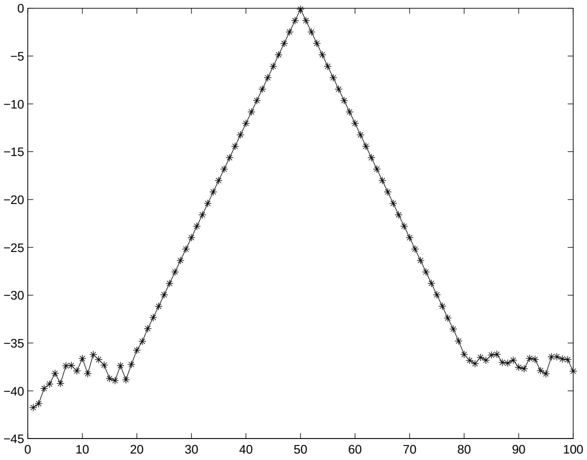


FIG. 4. *Logplot of the localized eigenvector.*

with the constraint $|t| < 1$. Equation (2) has the unique solution

$$\begin{cases} t = \frac{1}{2}(-b + \operatorname{sign}(b)\sqrt{4+b^2}), \\ \lambda = \operatorname{sign}(b)\sqrt{4+b^2}. \end{cases} \tag{3}$$

The spike vector x is only an approximation to a real eigenvector, since the residual terms R_1 and R_n are not zero at boundary nodes. But when L is large, those terms will be very small. We will prove that there is indeed an eigenvalue and eigenvector very close to our λ and x. Going back to the numerical experiment with $b = 3$ in Figure 3, the eigenvalue predicted by our construction is $\sqrt{13}$. The actual eigenvalue calculated using MATLAB agrees to 15 digits.

If we consider an *infinite* linear string with nodes numbered from $-\infty$ to ∞ and a single modification at entry $(0, 0)$, then the same construction will produce an exact eigenvalue and eigenvector. This is the only $\mathcal{L}^2$-finite eigenvector for the system (the following approach was suggested by David Ingerman, beginning with the Fourier transform of any such eigenvector):

$$f(y) = \sum_{n=-\infty}^{\infty} x_n e^{-iny}.$$

The eigenvector equation $(A+B)x = \lambda x$ can be rewritten as

$$x_{n-1} + x_{n+1} + b\delta(n)x_n = \lambda x_n, \tag{4}$$

where $\delta(n)$ is the discrete Dirac delta function. Fourier transform of (4) yields

$$f(y) = \frac{bx_0}{\lambda - e^{-iy} - e^{iy}}, \quad x_0 \neq 0. \tag{5}$$

The inverse transform recovers the eigenvector components

$$x_n = \frac{1}{2\pi}\int_0^{2\pi} f(y)e^{iny}\, dy. \tag{6}$$

At $n = 0$, (5) and (6) give the central component

$$x_0 = \frac{1}{2\pi}\int_0^{2\pi} \frac{bx_0\, dy}{\lambda - e^{-iy} - e^{iy}}. \tag{7}$$

By normalizing $x_0 = 1$ and substituting $z = e^{iy}$, (7) becomes an integral around the unit circle:

$$\frac{1}{2\pi i}\int_{S^1} \frac{b\, dz}{\lambda z - z^2 - 1} = 1. \tag{8}$$

Solving (8), we get

$$\lambda = \operatorname{sign}(b)\sqrt{4+b^2}. \tag{9}$$

Then (6) yields $x_n = x_0 t^{|n|}$ with t defined as in (3), which is our spike. This shows that the eigenvector we constructed is the unique localized eigenvector of $A + B$.

For a modification of M rows and columns, it is natural to expect that there will again be spiked eigenvectors. But now, instead of one spike, we expect M spikes

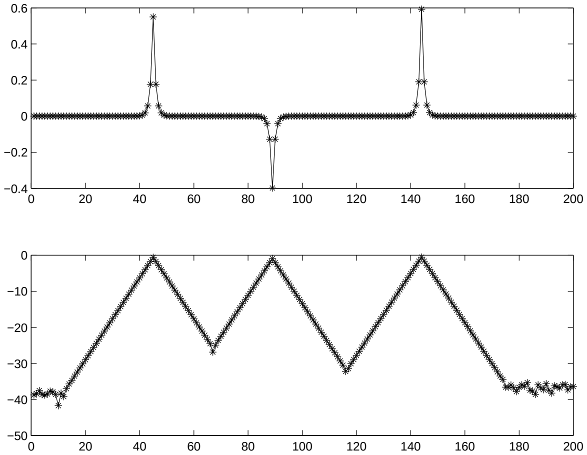


Fig. 5. *The localized eigenvector with three spikes and its logplot.*

of different heights. A typical localized eigenvector with three spikes is shown in Figure 5. The logplot shows the absolute values of the components.

To find the new eigenvalues and eigenvectors, we construct an approximate vector x that is now a sum of M spikes. Suppose the spike centered at the r_kth entry of x has height h_k, and the common spike ratio is t. Then the jth component of x has the form

$$x_j = \sum_{k=1}^{M} t^{|j-r_k|} h_k. \tag{10}$$

Substitute (10) into (1). We want the residual $R = (A+B)x - \lambda x$ to be small. As in section 2, we can divide the entries of R into three categories.

1. For nodes j other than $r_1, r_2, \ldots, r_M$ and the boundary nodes,

$$R_j = \left(t + \frac{1}{t}\right) x_j - \lambda x_j. \tag{11}$$

2. The spike centered at a modified row r_k contributes $2th_k + (Bh)_k - \lambda h_k$ to the r_k component of the residual. All other spikes contribute $\mathrm{O}(t^L)$ since they are separated by a distance at least L. Thus, we have

$$R_{r_k} = 2th_k + (Bh)_k - \lambda h_k + \mathrm{O}(t^L). \tag{12}$$

3. For boundary nodes $j = 1$ or n, every spike contributes $\mathrm{O}(t^L)$ to the residual and

$$R_j = \mathrm{O}(t^L), \quad j = 1,\ n. \tag{13}$$

Since $|t| < 1$, we ignore all errors of order t^L and set the R_j's to zero. Then (12) says that the vector h of spike heights is an eigenvector of B. If that eigenvector has an eigenvalue μ, (11) and (12) become

$$2t + \mu = \lambda = t + \frac{1}{t}. \tag{14}$$

Equation (14) is exactly the same system of equations we have in (2) with b replaced by μ. The unique solution is

$$\left\{\begin{aligned} t =& \ \tfrac{1}{2}(-\mu + \operatorname{sign}(\mu)\ \sqrt{4+\mu^2}),\\ \lambda =& \ \operatorname{sign}(\mu)\ \sqrt{4+\mu^2}. \end{aligned}\right. \tag{15}$$

Equation (15) is the (approximate) relation between the new eigenvalue λ of $A+B$ and the eigenvalue μ of B. Our next goal is to prove that the error in (15) is of the same order t^L as the terms that were dropped. In the remaining sections of this paper, unless specified otherwise, we will use L_2-norm for vector norms, and spectral norm for matrix norms.

When B is a symmetric matrix, the modified matrix $A+B$ will also be symmetric, and we can bound the eigenvalue using the following easy estimate.

THEOREM 2.1. *Suppose A is symmetric, x_0 is a unit vector, and $R = Ax_0 - \lambda_0 x_0$. Then there is an eigenvalue λ of A satisfying $|\lambda - \lambda_0| \leq \|R\|$.*

Proof. Assume λ_0 is not an eigenvalue of A. Let $\sigma = \|(A-\lambda_0 I)^{-1}\|$. Since A is symmetric, σ^{-1} is the smallest distance between λ_0 and eigenvalues of A. We have

$$1 = \|x_0\| = \|(A-\lambda_0 I)^{-1}R\| \leq \sigma\|R\|.$$

Thus, $\sigma^{-1} \leq \|R\|$. □

For λ in (15) and x in (10), the norm of the residual $\|R\| = \|(A+B)x - \lambda x\|$ is of order t^L. So an actual eigenvalue of $A+B$ is within $\mathrm{O}(t^L)$ of λ.

Now we can state the result for the nonsymmetric (and nondegenerate) case.

THEOREM 2.2. *If μ is a simple nonzero real eigenvalue of the M by M matrix B, with eigenvector h of norm one, then λ in* (15) *and x in* (10) *are within $O(t^L)$ of an exact eigenvalue-eigenvector pair for the modified matrix $A+B$, where t is defined in* (15).

For the proof, we want some bound on the determinant of the modification.

LEMMA 2.3. *If an n by n matrix A is modified by any matrix B, then*

$$|\det(A+B) - \det(A)| \leq n!2^n|A|^{n-1}|B|, \tag{16}$$

where $|A|$ and $|B|$ are the largest absolute values of the entries, and we assume $|A| \geq |B|$.

Proof. The determinant of $A+B$ is a sum of $n!$ terms. Each of these terms is a product of n entries of $A+B$. Expand that product into 2^n monomials, each one a product of a's from A and b's from B.

Cancel the monomial that uses only a's and contributes to $\det(A)$. The remaining monomials, with at least one factor from B, are bounded by $|A|^{n-1}|B|$. Then $\det(A+B) - \det(A)$ is bounded by (16). □

Proof of Theorem 2.2. Since we applied a modification matrix B to A at M widely spaced indices $1 \ll r_1 \ll r_2 \ll \cdots \ll r_M \ll n$, the eigenvector equation $(A+B)x = \lambda x$ can be divided naturally into $M+2$ parts. The first $M+1$ parts do not involve B. Each of those parts, between the modified rows r_i and r_{i+1}, has the form

$$\begin{bmatrix} 1 & -\lambda & 1 & & & \\ & 1 & -\lambda & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -\lambda & 1 \end{bmatrix} \begin{bmatrix} x_{r_i} \\ \vdots \\ x_{r_{i+1}} \end{bmatrix} = 0, \quad i = 0, \ldots, M. \tag{17}$$

To simplify our discussion, we added two indices $r_0 = 0$ and $r_{M+1} = n + 1$ with components $x_0 = x_{n+1} = 0$.

The only part that involves B is in the M modified rows:

$$(B - \lambda I_M)\begin{bmatrix} x_{r_1} \\ x_{r_2} \\ \vdots \\ x_{r_M} \end{bmatrix} + \begin{bmatrix} x_{r_1-1} + x_{r_1+1} \\ x_{r_2-1} + x_{r_2+1} \\ \vdots \\ x_{r_M-1} + x_{r_M+1} \end{bmatrix} = 0. \tag{18}$$

The key to our method is to express x_{r_i-1} and x_{r_i+1} in terms of $x_{r_1}, \dots, x_{r_M}$. This effectively converts the n by n eigenvalue problem to a root-finding problem based on an M by M matrix. Since M is fixed and is far smaller than n, we can (with patience) bound the actual eigenvalue.

To accomplish that, let A_i be the $(r_{i+1} - r_i - 1)$ by $(r_{i+1} - r_i + 1)$ matrix in (17). This A_i can be decomposed into two factors:

$$\begin{bmatrix} 1 & -\lambda & 1 & & & \\ & 1 & -\lambda & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -\lambda & 1 \end{bmatrix} = \begin{bmatrix} 1 & -t & & & \\ & 1 & -t & & \\ & & \ddots & \ddots & \\ & & & 1 & -t \end{bmatrix}\begin{bmatrix} 1 & -t^{-1} & & & \\ & 1 & -t^{-1} & & \\ & & \ddots & \ddots & \\ & & & 1 & -t^{-1} \end{bmatrix}, \tag{19}$$

where t satisfies the quadratic equation

$$t^2 - \lambda t + 1 = 0. \tag{20}$$

The first factor on the right side of (19) is $(r_{i+1} - r_i - 1)$ by $(r_{i+1} - r_i)$. The second factor is $(r_{i+1} - r_i)$ by $(r_{i+1} - r_i + 1)$.

Substituting this decomposition into (17), we have

$$\begin{bmatrix} 1 & -t & & & \\ & 1 & -t & & \\ & & \ddots & \ddots & \\ & & & 1 & -t \end{bmatrix}\begin{bmatrix} x_{r_i} - t^{-1}x_{r_i+1} \\ x_{r_i+1} - t^{-1}x_{r_i+2} \\ \vdots \\ x_{r_{i+1}-1} - t^{-1}x_{r_{i+1}} \end{bmatrix} = 0. \tag{21}$$

Solving (21), and regretting that the expressions appear awkward, we get

$$\begin{aligned} x_{r_i+1} &= \frac{t^{r_{i+1}-r_i-1} - t^{-(r_{i+1}-r_i-1)}}{t^{r_{i+1}-r_i} - t^{-(r_{i+1}-r_i)}}x_{r_i} + \frac{t - t^{-1}}{t^{r_{i+1}-r_i} - t^{-(r_{i+1}-r_i)}}x_{r_{i+1}}, \\ x_{r_{i+1}-1} &= \frac{t - t^{-1}}{t^{r_{i+1}-r_i} - t^{-(r_{i+1}-r_i)}}x_{r_i} + \frac{t^{r_{i+1}-r_i-1} - t^{-(r_{i+1}-r_i-1)}}{t^{r_{i+1}-r_i} - t^{-(r_{i+1}-r_i)}}x_{r_{i+1}}. \end{aligned} \tag{22}$$

To simplify our notation, let $\psi(t, n) = t^{-n} - t^n$. Then (22) can be written as

$$\begin{aligned} x_{r_i+1} &= \frac{\psi(t, r_{i+1} - r_i - 1)}{\psi(t, r_{i+1} - r_i)}x_{r_i} + \frac{\psi(t, 1)}{\psi(t, r_{i+1} - r_i)}x_{r_{i+1}}, \\ x_{r_{i+1}-1} &= \frac{\psi(t, 1)}{\psi(t, r_{i+1} - r_i)}x_{r_i} + \frac{\psi(t, r_{i+1} - r_i - 1)}{\psi(t, r_{i+1} - r_i)}x_{r_{i+1}}. \end{aligned} \tag{23}$$

Before proceeding to study (18), we observe that (20) has reciprocal roots. Since $\psi(t,n) = -\psi(t^{-1},n)$ and all the terms in (23) involve pairs of ψ's, it makes no difference which root we choose. Without loss of generality, we choose

$$t = \frac{1}{2}(\lambda - \operatorname{sign}(\lambda)\sqrt{\lambda^2-4}). \tag{24}$$

When λ is real with $|\lambda| > 2$, which is the case here based on our assumptions on μ, t will be a real number that lies in $(-1,1)$. As n increases, $\psi(1,n) = t^{-n}(1-t^{2n})$ approaches t^{-n}. Using this approximation,

$$\begin{aligned}\frac{\psi(t,n-1)}{\psi(t,n)} &= \frac{t^{-(n-1)}(1-t^{2(n-1)})}{t^{-n}(1-t^{2n})}\\ &= t(1-t^{2(n-1)})(1+t^{2n}+t^{4n}+\cdots)\\ &= t-(1-t^2)t^{2n-1}+\mathrm{O}(t^{4n})\end{aligned} \tag{25}$$

and

$$\begin{aligned}\frac{\psi(t,1)}{\psi(t,n)} &= \frac{t^{-1}-t}{t^{-n}(1-t^{2n})}\\ &= t^n(t^{-1}-t)(1+t^{2n}+t^{4n}+\cdots)\\ &= t^n(t^{-1}-t)+\mathrm{O}(t^{3n}).\end{aligned} \tag{26}$$

So when $|t| < 1$ and $n \gg 1$, the ratio $\psi(t,n-1)/\psi(t,n)$ is approximately t with error $\mathrm{O}(t^{2n})$ and $\psi(t,1)/\psi(t,n)$ is approximately zero with error $\mathrm{O}(t^n)$.

Substituting (23) into (18), we get

$$(B - \lambda I + \Delta(\lambda))x_M = 0, \tag{27}$$

where $x_M = (x_{r_1},\ldots,x_{r_M})^T$ is a subvector of x and $\Delta(\lambda)$ is a symmetric tridiagonal M by M matrix:

$$\Delta(\lambda) = \begin{bmatrix} \frac{\psi(t,r_1-1)}{\psi(t,r_1)} + \frac{\psi(t,r_2-r_1-1)}{\psi(t,r_2-r_1)} & \frac{\psi(t,1)}{\psi(t,r_2-r_1)} & 0 \\ \frac{\psi(t,1)}{\psi(t,r_2-r_1)} & \frac{\psi(t,r_2-r_1-1)}{\psi(t,r_2-r_1)} + \frac{\psi(t,r_3-r_2-1)}{\psi(t,r_3-r_2)} & \frac{\psi(t,1)}{\psi(t,r_3-r_2)} \\ \ddots & \ddots & \ddots \end{bmatrix}. \tag{28}$$

Δ is a function of λ because t is a function of λ in (24). When $|\lambda| > 2$, we have $|t(\lambda)| < 1$. Using the approximations (25) and (26),

$$\Delta(\lambda) = \begin{bmatrix} 2t & & & \\ & 2t & & \\ & & \ddots & \\ & & & 2t \end{bmatrix} + \begin{bmatrix} \mathrm{O}(t^{2L}) & \mathrm{O}(t^L) & & \\ \mathrm{O}(t^L) & \mathrm{O}(t^{2L}) & \mathrm{O}(t^L) & \\ & \ddots & \ddots & \ddots \\ & & \mathrm{O}(t^L) & \mathrm{O}(t^{2L}) \end{bmatrix}. \tag{29}$$

So when $|\lambda| > 2$, the difference between $\Delta(\lambda)$ and $2tI$ is an M by M symmetric tridiagonal matrix $\Omega(\lambda) = \Delta(\lambda) - 2tI$ whose nonzero entries are at most of order t^L.

3. The eigenvalues and eigenvectors of $\boldsymbol{A+B}$. Let us consider the M by M matrix $B-\lambda I+\Delta(\lambda)$ in (27). The roots of its determinant $\rho(\lambda)$ correspond to eigenvalues of the matrix $A+B$. Let λ_0 and t_0 be defined by (15). To prove that there is an eigenvalue of $A+B$ within $\mathrm{O}(t_0^L)$ of λ_0 is equivalent to proving that one root of $\rho(\lambda)$ is within $\mathrm{O}(t_0^L)$ of λ_0 for sufficiently large L. To simplify our discussion, we will assume that μ is a positive simple eigenvalue of B. The case when μ is a negative simple eigenvalue of B is exactly the same.

From (14) we know that $\lambda_0-2t_0=\mu$. Since μ is separated from other eigenvalues of B, we can always choose δ_1 such that for $\lambda\in(\lambda_0-\delta_1,\lambda_0+\delta_1)$, the distance between $\sqrt{\lambda^2-4}$ and any other eigenvalue of B is at least $\delta_1/2$, and $|t(\lambda)|\le 1-\delta_1/2$.

Recall that all the entries of $\Omega(\lambda)$ are of order $t(\lambda)^L$ or lower. Using Lemma 2.3, with $M!2^M|B|^{M-1}$ a fixed constant once B is chosen, there exist two constants C and L_1 such that for any $\lambda\in(\lambda_0-\delta_1,\lambda_0+\delta_1)$ and for all $L>L_1$,

$$|\rho(\lambda)-\det(B-\sqrt{\lambda^2-4}I)|<Ct(\lambda)^L. \tag{30}$$

Since $\lambda_0=\sqrt{\mu^2+4}$, and μ is an eigenvalue of B, we have $\det(B-\sqrt{\lambda_0^2-4}I)=0$. Substituting this into (30) yields $|\rho(\lambda_0)|<Ct_0^L$. Rewrite (30) as

$$\det(B-\sqrt{\lambda^2-4}I)-Ct(\lambda)^L<\rho(\lambda)<\det(B-\sqrt{\lambda^2-4}I)+Ct(\lambda)^L. \tag{31}$$

Consider the function $\rho_1(\lambda)=\det(B-\sqrt{\lambda^2-4}I)$. Let $\mu_1,\dots,\mu_M$ be the eigenvalues of B (with $\mu_1=\mu$):

$$\rho_1(\lambda)=\prod_{i=1}^{M}(\mu_i-\sqrt{\lambda^2-4}).$$

If $\nu=\sqrt{\lambda^2-4}$, then $\nu(\lambda_0)=\mu$. Since μ is a simple eigenvalue of B,

$$\frac{\partial\rho_1}{\partial\nu}\ \text{ and also }\ \frac{\partial\nu}{\partial\lambda}=\frac{\lambda}{\sqrt{\lambda^2-4}}$$

are nonzero at λ_0. Thus $\rho_1'(\lambda)$ is nonzero at λ_0. Without loss of generality, assume $\rho_1'(\lambda_0)>0$. Then $t(\lambda)$ defined in (24) has

$$t'(\lambda)=\frac{1}{2}-\frac{\lambda}{\sqrt{\lambda^2-4}}\le-\frac{1}{2}.$$

Since $\rho_1(\lambda)$ and $t'(\lambda)$ are smooth functions, there exists a $\delta_2<\delta_1$ such that for λ in $(\lambda_0-\delta_2,\lambda_0+\delta_2)$, $\rho_1'(\lambda)$ is bounded from below by a positive constant θ, and $|t'(\lambda)|$ is bounded from above by a positive constant ξ.

Let $\gamma=\sup\{\,t(\lambda)\mid\lambda\in(\lambda_0-\delta_2,\lambda_0+\delta_2)\}<1$. Both $L\gamma^{L-1}$ and t_0^L approach zero as L goes to infinity. Thus, there exists a constant $L_2>L_1$ such that for any $L>L_2$,

$$|L\gamma^{L-1}|<\frac{\theta}{2\xi C}\ \text{ and }\ t_0^L<\frac{\theta\delta_2}{2C}.$$

If $f_1(\lambda)=\rho_1(\lambda)+Ct(\lambda)^L$, then $f_1(\lambda_0)=Ct_0^L>0$. For $\lambda\in(\lambda_0-\delta_2,\lambda_0+\delta_2)$ and $L>L_2$ and $\epsilon=2Ct_0^L/\theta<\delta_2$,

$$\begin{aligned}f_1'(\lambda)&=\rho_1'(\lambda)-CLt(\lambda)^{L-1}t'(\lambda)\\&>\theta-\xi CL\gamma^{L-1}\\&>\theta-\frac{\theta}{2}=\frac{\theta}{2}.\end{aligned} \tag{32}$$

So

$$f_1(\lambda_0-\epsilon)=f_1(\lambda_0)-\int_{\lambda_0-\epsilon}^{\lambda_0} f_1'(\lambda)\,d\lambda < Ct_0^L-\frac{\theta}{2}\epsilon=0.$$

Since $f_1(\lambda_0)>0$ and $f_1(\lambda_0-\epsilon)<0$, there exists a $\lambda_1\in(\lambda_0-\epsilon,\lambda_0)$ such that $f_1(\lambda_1)=0$. A similar argument shows that for $f_2(\lambda)=\rho_1(\lambda)-Ct(\lambda)^L$, there exists a $\lambda_2\in(\lambda_0,\lambda_0+\epsilon)$ such that $f_2(\lambda_2)=0$. From (31), we know $\rho(\lambda_1)<0$ and $\rho(\lambda_2)>0$, so there exists a Λ in $(\lambda_1,\lambda_2)\subset(\lambda_0-\epsilon,\lambda_0+\epsilon)$ such that $\rho(\Lambda)=0$. This is the eigenvalue we are looking for. Since $|\Lambda-\lambda_0|<\epsilon$, which is of order t_0^L, we conclude that the distance of λ_0 from a real eigenvalue of $A+B$ is of order t_0^L.

The other part of the theorem is to prove that there is an eigenvector X corresponding to the actual eigenvalue Λ such that X is within $\mathrm{O}(t_0^L)$ of x defined by (10). Let V be the one-dimensional eigenspace of B corresponding to eigenvalue μ. Since μ is a simple eigenvalue of B, any nonzero vector β in the orthogonal complement $V^\perp$ has $(B-\mu)\beta\neq 0$. Set

$$\delta_3=\min\{\|(B-\mu)\beta\|\mid\beta\in V^\perp,\ \|\beta\|=1\}>0. \tag{33}$$

Let X be an eigenvector of $A+B$ corresponding to the eigenvalue Λ, and let H be the subvector $\{X_{r_1},\dots,X_{r_M}\}^T$ of X. We normalize X by $\|H\|=1$. Set $\eta=\sqrt{\Lambda^2-4}$ and $T=t(\Lambda)$. Since the difference between λ_0 and Λ is of order t_0^L and the derivatives of $t(\lambda)$ and $\sqrt{\lambda^2-4}$ do not vanish at λ_0, it is clear that η and T are within $\mathrm{O}(t_0^L)$ of μ and t_0. Recall that all the nonzero entries of the tridiagonal matrix $\Omega(\lambda)=\Delta(\lambda)-2t(\lambda)I$ are of order $t(\lambda)^L$ or lower, so $\|\Omega(\lambda)\|$ is of order $t(\lambda)^L$. Thus there exist two constants C_1 and $L_3>L_2$ such that for any $L>L_3$, $|\eta-\mu|<C_1t_0^L$, $|T-t_0|<C_1t_0^L$, $L\gamma^L<1$, and $\|\Omega(\Lambda)\|<C_1T^L$.

For any $L>L_3$,

$$\begin{aligned}|T^L-t_0^L|&=|T-t_0|\sum_{i=0}^{L-1}T^it_0^{L-i-1}\le|T-t_0|L\gamma^L\\&<|T-t_0|\le C_1t_0^L.\end{aligned}\tag{34}$$

This implies that $T^L<(C_1+1)t_0^L$ for $L>L_3$. With $C_2=C_1(C_1+1)$ and $L>L_3$, we have $\|\Omega(\Lambda)\|<C_1T^L<C_2t_0^L$.

There exist two unique vectors $\alpha\in V$ and $\beta\in V^\perp$ such that $H=\alpha+\beta$. For any $L>L_3$, from (27) we have

$$\begin{aligned}0&=\|(B-\Lambda+\Delta(\Lambda))H\|=\|(B-\eta)H+\Omega(\Lambda)H\|\\&=\|(B-\mu)\alpha+(B-\mu)\beta+(\mu-\eta)H-\Omega(\Lambda)H\|\\&\ge\delta_3\|\beta\|-|\mu-\eta|-\|\Omega(\Lambda)\|\\&>\delta_3\|\beta\|-(C_1+C_2)t_0^L.\end{aligned}\tag{35}$$

From (35) we know

$$\|\beta\|<\frac{(C_1+C_2)t_0^L}{\delta_3}. \tag{36}$$

Recall that the construction of X in (10) is based on a norm 1 eigenvector of B corresponding to eigenvalue μ. Since the eigenspace V is of dimension 1, we can assume $\alpha/\|\alpha\| = h$ (otherwise multiply X by -1). Thus,

$$\begin{aligned} \|h - H\| &\le \|h - \alpha\| + \|\beta\| \\ &= (1 - \|\alpha\|) + \|\beta\| \le 2\|\beta\| \\ &< \frac{2(C_1 + C_2)t_0^L}{\delta_3} = C_3 t_0^L. \end{aligned} \tag{37}$$

In (23) we solved for X_{r_i+1} and $X_{r_{i+1}-1}$ in terms of X_{r_i} and $X_{r_{i+1}}$. In the same way, we can solve for other terms between indices r_i and r_{i+1}, and the result is

$$X_{r_i+k} = \frac{\psi(t, r_{i+1} - r_i - k)}{\psi(t, r_{i+1} - r_i)} X_{r_i} + \frac{\psi(t, k)}{\psi(t, r_{i+1} - r_i)} X_{r_{i+1}}. \tag{38}$$

The eigenvector we constructed in (10) is a combination of M spikes,

$$x = \sum_{i=1}^{M} u_i, \tag{39}$$

where the jth entry of spike u_i is

$$u_{i,j} = h_i t_0^{|j-r_i|}. \tag{40}$$

From (38), we know that the actual eigenvector X can also be written as a combination of M spikes:

$$X = \sum_{i=1}^{M} U_i, \tag{41}$$

where the jth entry of U_i has the following form:

$$U_{i,j} = \begin{cases} \frac{\psi(T, r_i - r_{i-1} - |j - r_i|)}{\psi(T, r_i - r_{i-1})} H_i & r_{i-1} \le j \le r_i, \\ \frac{\psi(T, r_{i+1} - r_i - |j - r_i|)}{\psi(T, r_{i+1} - r_i)} H_i & r_i \le j \le r_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{42}$$

Since

$$\|x - X\| = \left\| \sum_{i=1}^{M} u_i - \sum_{i=1}^{M} U_i \right\| \le \sum_{i=1}^{M} \|u_i - U_i\|, \tag{43}$$

we have $\|x - X\| = \mathrm{O}(t_0^L)$ if each $\|u_i - U_i\|$ is of order t_0^L. We look at $\|u_1 - U_1\|^2$. Substituting (40) and (42), this is

$$\sum_{i=1}^{r_1} (t_0^{r_1 - i} h_1 - \frac{\psi(T, i)}{\psi(T, r_1)} H_1)^2 + \sum_{i=r_1+1}^{r_2} \left(t_0^{i - r_1} h_1 - \frac{\psi(T, r_2 - i)}{\psi(T, r_2 - r_1)} H_1 \right)^2 + \sum_{i=r_2+1}^{n} (t_0^{i - r_1} h_1)^2. \tag{44}$$

From (37), we know that

$$|h_1 - H_1| \le \|h - H\| < C_3 t_0^L \quad \text{for } L > L_3. \tag{45}$$

Also notice that for sufficiently large n and for any $0 \le k \le n$,

$$\left|T^{n-k} - \frac{\psi(T,k)}{\psi(T,n)}\right| = \frac{T^{n+k} - T^{3n-k}}{1 - T^{2n}} \le \frac{T^{n+k}}{1 - T^{2n}} \le 2T^{n+k}. \tag{46}$$

Using (34), (45), and (46), the first term of (44) can be bounded by a constant C_4 for $L > L_3$:

(47)

$$\begin{aligned}
&\sum_{i=1}^{r_1} \left(t_0^{r_1-i} h_1 - \frac{\psi(T,i)}{\psi(T,r_1)} H_1 \right)^2 \\
&\le 2 \sum_{i=1}^{r_1} \left[(t_0^{r_1-i} h_1 - T^{r_1-i} h_1)^2 + (T^{r_1-i} h_1 - T^{r_1-i} H_1)^2 + \left(T^{r_1-i} H_1 - \frac{\psi(T,i)}{\psi(T,r_1)} H_1 \right)^2 \right] \\
&< 2C_1^2 t_0^{2L} \sum_{i=0}^{r_1-1} (i\gamma^i)^2 + 2C_3^2 t_0^{2L} \sum_{i=0}^{r_1-1} T^{2i} + 8T^{2r_1} \sum_{i=1}^{r_1} T^{2i} \\
&< 2C_1^2 t_0^{2L} \sum_{i=0}^{\infty} (i\gamma^i)^2 + 2C_3^2 t_0^{2L} \sum_{i=0}^{\infty} \gamma^{2i} + 8(C_2+1)^2 t_0^{2L} \sum_{i=0}^{\infty} \gamma^{2i} \\
&= C_4 t_0^{2L}.
\end{aligned}$$

The second term of (44) can be bounded in exactly the same way, and we have

$$\sum_{i=r_1+1}^{r_2} \left(t_0^{i-r_1} h_1 - \frac{\psi(T, r_2 - i)}{\psi(T, r_2 - r_1)} H_1 \right)^2 < C_5 t_0^{2L}. \tag{48}$$

For the third term of (44), we get

$$\sum_{i=r_2+1}^{n} (t_0^{i-r_1} h_1)^2 < t_0^{2L} \sum_{i=0}^{\infty} t_0^{2i} = C_6 t_0^{2L}. \tag{49}$$

Combining (47), (48), and (49) yields

$$\|u_1 - U_1\|^2 < (C_4 + C_5 + C_6) t_0^{2L}. \tag{50}$$

The same is true for each $\|u_i - U_i\|$, $i \le M$. We conclude that $\|X - x\|$ is of order t_0^L. □

Remark 1. When B is a diagonal matrix, simple arguments show that the bound in (30) can be improved to $\mathrm{O}(t(\lambda)^{2L})$. This in turn will translate to a $\mathrm{O}(t_0^{2L})$ bound for the eigenvalue. The eigenvector bound from (44) will remain $\mathrm{O}(t_0^L)$.

Remark 2. The condition that μ is a real number is not necessary. Our construction and bound remain valid as long as the spike ratio t has magnitude less than one. This is true for any μ outside the $[-2i, 2i]$ line segment in the complex plane. The formulas for t and λ still apply. Take $B = \left[\begin{smallmatrix} 0 & 3 \\ -3 & 0 \end{smallmatrix}\right]$ as an example, with eigenvalues $\pm 3i$. If we apply a widely spaced modification governed by B to A, experiments show that $A + B$ will have two isolated eigenvalues that are approximately $\pm\sqrt{4-9} = \pm\sqrt{5}i$. One of the localized eigenvectors is very close to our construction with $h = [1, i]$ and $t = (\sqrt{5} - 3)i/2$. The other localized eigenvector is the complex conjugate.

In Theorem 2.2, the underlying matrix is the adjacency matrix of an n-node linear chain. Similar conclusions can be drawn when the underlying matrix is the adjacency

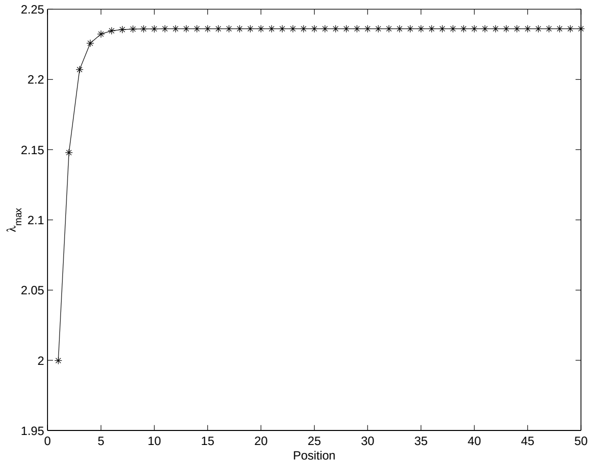


FIG. 6. *The special eigenvalue of the perturbed matrix as the added* 1 *moves from* $(1,1)$ *to* $(50,50)$ *in a matrix of order* 100.

matrix of an n-node circle. The difference is that in the n-node circle case, $\Delta(\lambda)$ in (27) has $\psi(t,1)/\psi(t,n+r_1-r_M+1)$ in the $(1,n)$ and $(n,1)$ entries. Repeat the arguments in the proof of Theorem 2.2 with this new $\Delta(\lambda)$. In the same way, λ and x in (15) and (10) are within $\mathrm{O}(t^L)$ of an eigenvalue and eigenvector of the modified matrix.

Example 1 (now confirmed). Suppose we change only a single entry on the main diagonal from 0 to 1. The modifying matrix is just $B = [1]$. For the (infinite) modified matrix, the localized eigenvector is exact. The eigenvalue is $\sqrt{5}$ and the spike ratio $t = (-1+\sqrt{5})/2$ is the reciprocal of the golden mean. If the single 1 is the $(0,0)$ entry, the jth component of the eigenvector is $t^{|j|}$.

For a finite matrix, this eigenvalue-eigenvector pair is only approximate. The approximation is good when the modified entry is near the center of the finite matrix and poor (see Figure 6) as it approaches the ends of the diagonal (where the limiting eigenvalue is 2).

Example 2. Connect three widely spaced nodes i,j,k by three undirected edges. In this case the modifying matrix is

$$B = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}.$$

Its eigenvalues are $\mu = 2, -1, -1$. The eigenvalues of the large matrix $A+B$ are approximately $\lambda = \sqrt{4+2^2} = \sqrt{8}$ and $\lambda = -\sqrt{4+(-1)^2} = -\sqrt{5}$ (twice).

The eigenvector $h = (1,1,1)$ of the small matrix is correctly reflected in the eigenvector of $A+B$ for $\lambda = \sqrt{8}$. It is very nearly a sum of three equal spikes.

The other eigenvalue $\mu = -1$ is repeated. The eigenvalue $\lambda = -\sqrt{5}$ is nearly but not exactly repeated. Theorem 2.2 cannot apply as it stands to the eigenvectors, because the small matrix has a plane of eigenvectors for $\mu = -1$. Since $\lambda = -\sqrt{5}$ is not exactly repeated, there is no corresponding plane for $A+B$. Experiment shows that its eigenvectors are sums of spikes at i,j,k with weights $h = (-1,0,1)$ and $h = (-1,2,-1)$.

To extend this example, suppose the modification adds a complete graph on M nodes to the starting graph (which is still an infinite line of nodes connected only to their neighbors). The M by M matrix B has 0's on the diagonal and 1's everywhere else, as above. Its largest eigenvalue $\mu = M-1$ has eigenvector $h = (1,1,\ldots,1)$. Then the modified matrix $A+B$ has largest eigenvalue $\lambda = \sqrt{4+(M-1)^2}$ with $x =$ sum of equal spikes.

The $M-1$ remaining eigenvalues of the small matrix B all equal -1. Again this produces $M-1$ eigenvalues of the modified matrix, all close to $\lambda = -\sqrt{5}$ but not repeated. Each of them has a corresponding eigenvector that is a sum of M spikes, with the weights adding to zero.

4. Beyond tridiagonal matrices. We turn to cases in which the original matrix A is no longer tridiagonal.

Suppose A is an n by n symmetric Toeplitz matrix, with $(0, a_1, \ldots, a_q, 0, \ldots, 0)$ as its first row. Apply a rank one modification by choosing an index $1 \ll r \ll n$ and changing the (r,r) entry of A from 0 to b. Numerical experiments tell us that one isolated eigenvalue will appear for $A+B$, together with a localized eigenvector. Figure 7 is typical, with $n = 200$ and modified row $r = 100$, the matrix $A = \text{Toeplitz}\{0,1,1,0,\ldots,0\}$, and $b = 1$. MATLAB computed the new isolated eigenvalue

$$\lambda = 4.05956284882943. \tag{51}$$

In Figure 7, the plot on the left shows the localized eigenvector. On the right is its logplot. Comparing to Figure 4, there is no longer a single spike ratio near the position $r = 100$ of the modification. For $q > 1$, the localized eigenvector is approximately the sum of q spikes with different weights and different spike ratios $t_1, \ldots, t_q$.

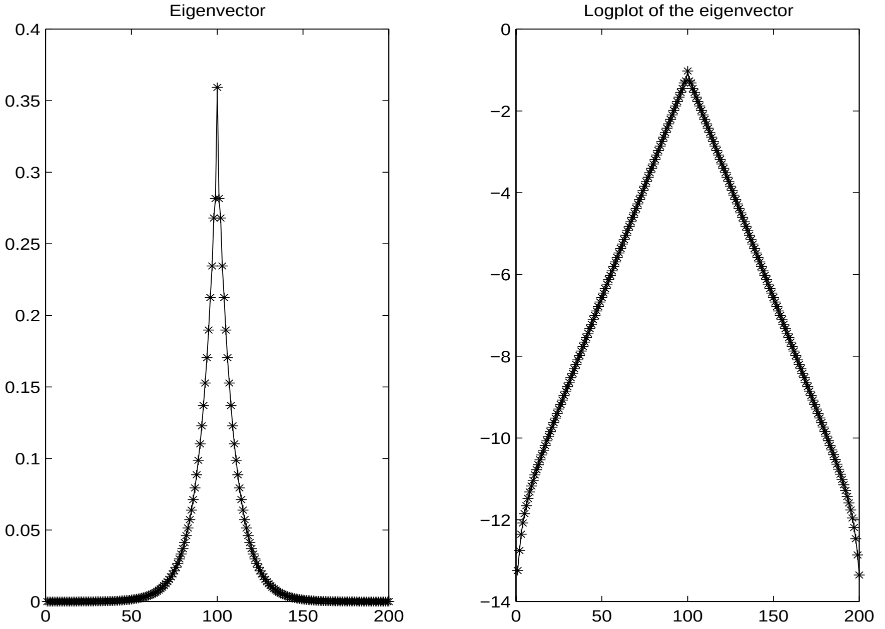


FIG. 7. *Localized eigenvector of a modified Toeplitz matrix*

Take $q = 2$ as an example. To simplify the discussion, assume that A is an infinite matrix indexed from $-\infty$ to ∞. (If the matrix is finite, the boundary terms will be of order t^L or lower.) Suppose A has all ones on the *first and second* subdiagonal and superdiagonal, and zeros on the main diagonal (as in Figure 7). Apply a unit modification ($b = 1$) to the $(0, 0)$ entry, and look for a new eigenvector x that is the sum of two spikes centered at 0:

$$x_k = c_1 t_1^{|k|} + c_2 t_2^{|k|}. \tag{52}$$

We normalize x by $x_0 = 1$, which means

$$c_1 + c_2 = 1. \tag{53}$$

Now consider the eigenvector equation $(A + B)x = \lambda x$ a row at a time:

$$(Ax)_k + (Bx)_k = \lambda x_k. \tag{54}$$

Substituting (52) into (54), we have the cases $k \geq 2$, $k = 1$, $k = 0$ (symmetry accounts for $k < 0$).

1. For $k \geq 2$, we have

$$c_1 t_1^{k-2}(1 + t_1 + t_1^3 + t_1^4 - \lambda t_1^2) + c_2 t_2^{k-2}(1 + t_2 + t_2^3 + t_2^4 - \lambda t_2^2) = 0. \tag{55}$$

It is clear that (55) will be satisfied if

$$1 + t_1 + t_1^3 + t_1^4 - \lambda t_1^2 = 0 \text{ and } 1 + t_2 + t_2^3 + t_2^4 - \lambda t_2^2 = 0. \tag{56}$$

2. For $k = 1$, we have

$$c_1 t_1 + c_2 t_2 + 1 + c_1 t_1^2 + c_2 t_2^2 + c_1 t_1^3 + c_2 t_2^3 = \lambda(c_1 t_1 + c_2 t_2). \tag{57}$$

3. For $k = 0$, we have

$$2(c_1 t_1 + c_2 t_2 + c_1 t_1^2 + c_2 t_2^2) + 1 = \lambda. \tag{58}$$

The four equations in (56), (57), and (58) with the normality constraint (53) yield five equations for the unknowns λ, t_1, t_2, c_1, and c_2. Using MATLAB, we compute

$$\begin{aligned} &\lambda = 4.05956284889808, \quad t_1 = -0.37994846989306, \quad t_2 = 0.89676509565825, \\ &c_1 = 0.088390260956474, \quad c_2 = 0.91160973904353. \end{aligned} \tag{59}$$

This λ agrees with (51) to 10 digits. That confirms our conjecture (52) about the structure of the localized eigenvector. Since $c_2 t_2^k$ will dominate $c_1 t_1^k$ for large k, this explains the nearly single spike appearance of Figure 7.

Now we consider the general band Toeplitz case, with the simplest modification by $b = 1$. The underlying matrix A is a doubly infinite symmetric Toeplitz matrix (Laurent matrix) with $a_1, a_2, \ldots, a_q$ on the first q upper and lower diagonals, and zero elsewhere. We conjecture that the localized eigenvector x of $A + B$ is the sum of q spikes centered at $k = 0$, with different weights c_j and spike ratios t_j:

$$x_k = \sum_{j=1}^{q} c_j t_j^{|k|}. \tag{60}$$

For $k \geq q$, substituting (60) into (54) yields

$$\sum_{i=1}^{q}\sum_{j=1}^{q} a_i c_j (t_j^{k-i} + t_j^{k+i}) = \lambda \sum_{j=1}^{q} c_j t_j^k. \tag{61}$$

This equation will be satisfied if $\sum c_j = 1$ (normalization) and

$$\sum_{i=1}^{q} a_i (t_j^i + t_j^{-i}) - \lambda = 0 \quad \text{for } j = 1, \dots, q. \tag{62}$$

We get q more equations by substituting (60) into (54) for $k = 0, 1, \dots, q-1$. The $2q$ equations and the normalization match the number of unknowns $t_1, \dots, t_q, c_1, \dots, c_q, \lambda$. The eigenvector is exact when A is an infinite matrix and B is 1 by 1.

For completeness we generalize to an arbitrary widely spaced modification. Denote the localized eigenvector after a single point modification of magnitude b at index r by $x_{r,b}$. We know that $x_{r,b}$ is the sum of q spikes centered at r, with eigenvalue λ_b and largest spike ratio t_b. Pick M indices r_i separated by L or more. Modify A at these indices by an M by M matrix B. Let μ be a simple eigenvalue of B with eigenvector h. Then

$$x = \sum_{i=1}^{M} h_i x_{r_i,\mu} \ \text{ yields } \ \|(A+B)x - \lambda_\mu x\| = \mathrm{O}(t_\mu^L).$$

Briefly, a modified row leads to

$$h_i (A x_{r_i,\mu})_{r_i} + (Bh)_i = \lambda_\mu h_i + \mathrm{O}(t_\mu^L). \tag{63}$$

The $\mathrm{O}(t_\mu^L)$ term comes from the other $M-1$ vectors $x_{r_j,\mu}$ with $j \neq i$. From $(Bh)_i = \mu h_i$, we get

$$(A x_{r_i,\mu})_{r_i} + \mu = \lambda_\mu + \mathrm{O}(t_\mu^L). \tag{64}$$

By the definition of $x_{r_i,b}$ we know that $(A x_{r_i,\mu})_{r_i} + \mu = \lambda_\mu$. So (64) is satisfied up to $\mathrm{O}(t_\mu^L)$.

Figure 8 shows a localized eigenvector of a modified 400 by 400 Toeplitz matrix $A = \text{Toeplitz}\{0, 1, 1, 1, 0, \dots, 0\}$. B is a 3 by 3 random matrix, and the plot shows three "triple spikes" centered at different positions. Each triple spike is a combination of three spikes with different weights and spike ratios.

For a single point modification, say $b = 1$, another way to calculate the new isolated eigenvalue is by Fourier transform. Let x be an $\mathcal{L}^2$-finite eigenvector of the modified $A + B$, corresponding to a new λ. The eigenvector equation becomes

$$\lambda x_n = \sum_{k=1}^{q} a_k (x_{n-k} + x_{n+k}) + \delta(n) x_n. \tag{65}$$

Its Fourier transform is

$$\lambda f(y) = \sum_{k=1}^{q} a_k f(y) (e^{-iky} + e^{iky}) + x_0, \tag{66}$$

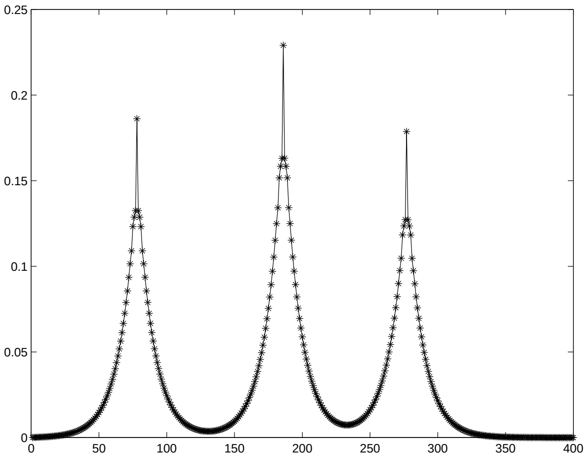


FIG. 8. *Localized eigenvector of a modified Toeplitz matrix with three spikes.*

which is equivalent to

$$f(y) = \frac{x_0}{\lambda - \sum_{k=1}^{q} a_k(e^{-iky} + e^{iky})}. \tag{67}$$

The inverse Fourier transform gives

$$x_0 = \frac{1}{2\pi}\int_0^{2\pi} \frac{x_0\ dy}{\lambda - \sum_{k=1}^{q} a_k(e^{-iky} + e^{iky})}. \tag{68}$$

Normalizing $x_0 = 1$ and substituting $z = e^{iy}$, we get

$$\frac{1}{2\pi i}\int_{S^1} \frac{z^{q-1}\ dz}{\lambda z^q - \sum_{k=1}^{q} a_k(z^{q+k} + z^{q-k})} = 1. \tag{69}$$

The solutions of (69) correspond to the eigenvalues of $A + B$ with localized eigenvectors.

5. Two-dimensional grids and localized eigenvectors. Suppose G is a regular M by N grid, with edges connecting to the (usually four) nearest neighbors. The MN by MN symmetric adjacency matrix A represents node (i, j) by the $((i-1)N + j)$th row and column. Numerical experiment tells us that a single point modification, when applied to a node (i, j) with $1 \ll i \ll M$ and $1 \ll j \ll N$, produces a localized eigenvector. Figure 9 shows one example from a 30 by 30 grid, modified at node $(16, 15)$. The three-dimensional plot uses MATLAB's "surf" function. The eigenvector in Figure 9 corresponds to the largest eigenvalue $\lambda = 3.9881$ of $A + B$. The logplot on the right shows that the spike ratio is no longer a constant.

For this case, we do not have an explicit formula for the localized eigenvector (is it a known special function?). We can still estimate the eigenvalue by Fourier transform on an infinite grid. Define V_G to be the space of all mappings from nodes of G to $\mathbb{R}$, and let $u_{i,j}$ be the value of node (i, j) under mapping (vector) u. The adjacency matrix of a graph is seen in [3] as an operator that maps V_G to V_G, $Au = v$, such that $v_{i,j}$ is the weighted sum of all $u_{k,l}$. In our case, the weight is one or zero depending on whether the two nodes are connected. Apply a unit modification on G by adding

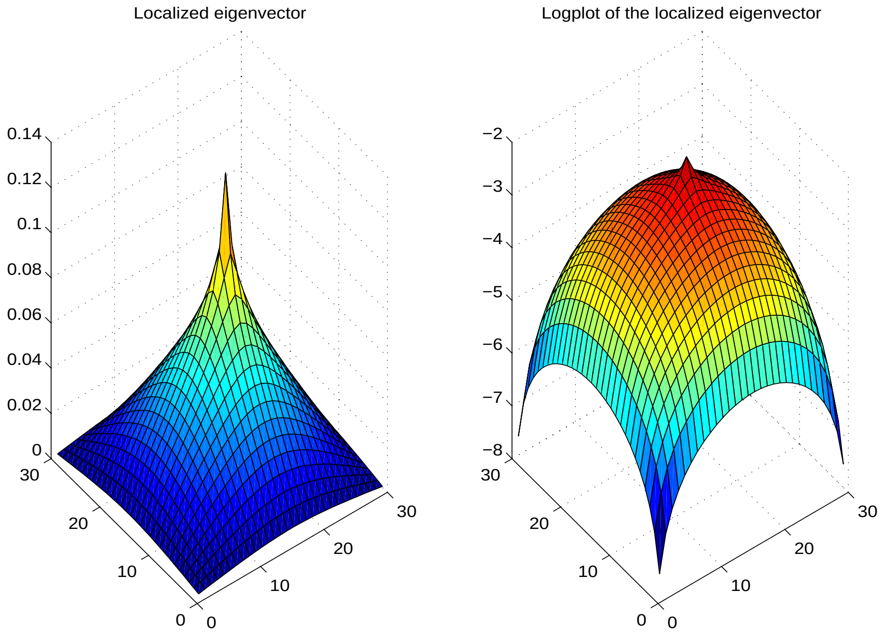


FIG. 9. *Localized eigenvector of the modified adjacency matrix of a two-dimensional grid.*

an edge from node $(0,0)$ to itself. Let x be a localized eigenvector of the modified adjacency matrix $A+B$:

$$x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1} + \delta(i,j)x_{i,j} = \lambda x_{i,j}, \tag{70}$$

where $\delta(i,j)$ is a two-variable delta function. Let $f(y,z)$ be the Fourier transform

$$f(y,z) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x_{k,l} e^{-i(ky+lz)} \ . \tag{71}$$

Transforming both sides of (70), we get

$$f(y,z) = \frac{x_{0,0}}{\lambda - e^{-iy} - e^{iy} - e^{-iz} - e^{iz}} \ . \tag{72}$$

Using the inverse two-dimensional transform and cancelling $x_{0,0}$, the integral equation for λ is

$$\frac{1}{(2\pi)^2} \int_0^{2\pi} \int_0^{2\pi} \frac{1}{\lambda - e^{-iy} - e^{iy} - e^{-iz} - e^{iz}} \, dy \, dz = 1. \tag{73}$$

By the result in (8), this simplifies to

$$E(\lambda) = \int_0^{2\pi} \frac{1}{\sqrt{(\lambda - e^{-iy} - e^{iy})^2 - 4}} \, dy = 2\pi. \tag{74}$$

Equation (74) involves an elliptic integral. The software package PARI/GP [4] produced the graph of $E(\lambda)$ in Figure 10. The dashed horizontal line $y = 2\pi$ intersects at the solution $\lambda = 4.000111576954677619$. This is a much smaller shift of λ_{max} than in the one-dimensional case of nodes along a line.

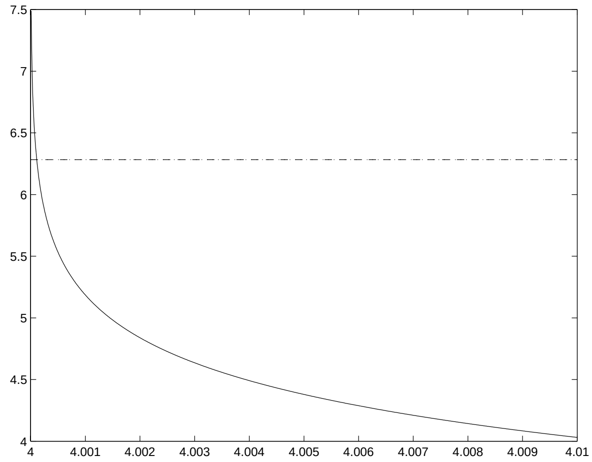


Fig. 10. *Plot of the elliptic integral $E(\lambda)$ in* (74).

REFERENCES

[1] A. Boettcher, M. Embree, and V.I. Sokolov, *Infinite Toeplitz and Laurent matrices with localized impurities*, Linear Algebra Appl., 343/344 (2002), pp. 101–118.

[2] A. Boettcher, M. Embree, and V.I. Sokolov, *The Spectra of Large Toeplitz Band Matrices with a Randomly Perturbed Entry*, Oxford University Computing Laboratory Numerical Analysis Report 01/16, Oxford, UK, 2001.

[3] F. Chung, *Spectral Graph Theory*, CBMS Reg. Conf. Ser. Math. 92, AMS, Providence, RI, 1997.

[4] H. Cohen and K. Belabas, *PARI-GP Software*, http://www.parigp-home.de.

SIAM J. DISCRETE MATH.
Vol. 16, No. 3, pp. 499–510

FRAMEPROOF CODES[*]

SIMON R. BLACKBURN[†]

Abstract. Frameproof codes were first introduced by Boneh and Shaw in the context of digital fingerprinting. Variants of these codes have been studied by several authors, and several similar definitions of frameproof codes exist in the literature. The paper considers frameproof codes from a combinatorial point of view, where we define frameproof codes as follows.

Let F be a (finite) set, and let $P \subseteq F^\ell$ be a set of words of length ℓ over the alphabet F. The *set of descendants* of P, $\mathrm{desc}(P)$, is the set of all words $x \in F^\ell$ such that for all $i \in \{1, 2, \ldots, \ell\}$, the ith component of x agrees with the ith component of some member of P. Let c be an integer such that $c \geq 2$. A *c-frameproof code* is a subset $C \subseteq F^\ell$ such that for all $P \subseteq C$ with $|P| \leq c$, we have that $\mathrm{desc}(P) \cap C = P$.

The paper considers the following question: What is the largest cardinality n of a c-frameproof code of length ℓ, over an alphabet of size q? The paper concentrates on the case when q is large. The paper shows that $n = \ell(q-1)$ in the case when $2 \leq \ell \leq c$ and shows that if $c = 2$, then n is approximately $tq^{\lceil \ell/2 \rceil}$, where $t = 1$ when ℓ is odd and $t = 2$ if ℓ is even. The paper establishes improved upper bounds on n by applying techniques from extremal set theory (namely, a generalization of the Erdős–Ko–Rado theorem).

Key words. frameproof codes, digital fingerprinting, watermarking

AMS subject classification. 68R05

DOI. 10.1137/S0895480101384633

1. Introduction.

Frameproof codes were first introduced by Boneh and Shaw [3] in the context of digital fingerprinting. There is more than one definition of frameproof codes in the literature; we use the following version.

Let F be a (finite) set of cardinality q and let ℓ be a positive integer. For a q-ary codeword $x \in F^\ell$ and an integer $i \in \{1, 2, \ldots, \ell\}$ we write x_i for the ith component of x. Let $P \subseteq F^\ell$ be a set of codewords of length ℓ. The *set of descendants* of P, $\mathrm{desc}(P)$, is the set of all words $x \in F^\ell$ such that for all $i \in \{1, 2, \ldots, \ell\}$, there exists $y \in P$ such that $x_i = y_i$. Let c be an integer such that $c \geq 2$. A *c-frameproof code* is a subset $C \subseteq F^\ell$ such that for all $P \subseteq C$ with $|P| \leq c$, we have that $\mathrm{desc}(P) \cap C = P$.

Boneh and Shaw use a different definition of descendant. The definition for frameproof codes we use is explicitly given by Fiat and Tassa [9], who credit Chor, Fiat, and Naor [4] with its first use. See Stinson and Wei [13] and Staddon, Stinson, and Wei [12] for constructions of binary frameproof codes and for a discussion of the relationship between frameproof codes and such concepts as traceability codes and codes with the identifiable parent property.

Inspired by an open question of Staddon, Stinson, and Wei [12, Section 5], we ask the following: What is the largest cardinality $M_{\ell,c}(q)$ of a q-ary c-frameproof code of length ℓ? Let ℓ and c be fixed. We are interested in how $M_{c,\ell}(q)$ behaves as a function of q.

When $\ell \leq c$, we give a simple argument (Corollary 3) to show that $M_{c,\ell}(q) = \ell(q-1)$ for $q \geq 2$. The more interesting and difficult case is when $\ell > c$. As a first approximation, previous results (see Theorem 1 and Construction 2 below) imply that

[*]Received by the editors February 6, 2001; accepted for publication (in revised form) February 6, 2003; published electronically June 25, 2003.
http://www.siam.org/journals/sidma/16-3/38463.html

[†]Department of Mathematics, Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK (s.blackburn@rhul.ac.uk).

$M_{c,\ell}(q) = \Theta(q^{\lceil \ell/c \rceil})$, where the constants hidden by the notation may depend on ℓ and c. This suggests that we examine the behavior of the ratio $R_{c,\ell}(q)$ defined by $R_{c,\ell}(q) = M_{c\ell}(q)/q^{\lceil \ell/c \rceil}$. Define t to be the unique integer such that $1 \leq t \leq c$ and $t = \ell \bmod c$. Again, it follows from known results that $\overline{\lim}_{q\to\infty} R_{c,\ell}(q) \leq \max\{1, t\}$ and that $\underline{\lim}_{q\to\infty} \geq 1$. When $\ell = 1 \bmod c$, these results imply (Corollary 5) that $\lim_{q\to\infty} R_{c,\ell}(q)$ exists and is equal to 1.

One case not covered by the above is the case $c = 2$ and ℓ even, where the above results show that $1 + o(1) \leq R_{c,\ell}(q) \leq 2 + o(1)$. In section 4, we give a construction that matches the upper bound, thus establishing that $\lim_{q\to\infty} R_{c,\ell}(q) = 2$ in this case.

In sections 5 and 6, we turn to improving the upper bound. Defining t as above, we show that $R_{c,\ell}(q) \leq \ell/(\ell - (t-1)\lceil \ell/c \rceil)$ by relating the problem of providing an upper bound to a problem in extremal set theory. In the two cases when $t = 1$ and $t = c$, this bound is essentially the same as the upper bound of $t + o(1)$ given by Theorem 1, but for any other values of t and c it gives an improvement. Indeed, when c is fixed and ℓ is large, then our new upper bound is approximately $c/(c - t + 1)$ which is generally much less than t.

In general there is still a gap between the upper bounds we have given for $R_{c,\ell}(q)$ and the lower bounds that follow from known large q-ary c-frameproof codes of length ℓ. In section 7 we close this gap in one case: when $\ell = 5$ and $c = 3$. By constructing a code of size $(5/3)q^2 + O(q)$, we show that our upper bound on $R_{c,\ell}(q)$ of $5/3 + o(1)$ is tight when $q \to \infty$ by establishing that $R_{c,\ell}(q) = 5/3 + o(1)$.

Note that any set of length 1 vectors is a c-frameproof code for any c; thus the length 1 case is trivial. For the remainder of the paper we consider codes of length ℓ, where $\ell \geq 2$.

The paper is organized as follows. Section 2 proves an upper bound on the size of a c-frameproof code. This bound is a slight modification of the bound given in Staddon, Stinson, and Wei [12, Theorem 3.7]. Section 3 contains two constructions of q-ary c-frameproof codes of length ℓ (one of these constructions has been given before, in Cohen and Encheva [5, Proposition 1]). Section 4 contains a third, more complicated, construction of 2-frameproof codes. The constructions of sections 3 and 4 show that the leading term of the upper bound has the correct order of magnitude; moreover, the leading coefficient of the upper bound is tight when $c = 2$, when $\ell \leq c$, or when $\ell = 1 \bmod c$. Section 5 improves the bound of section 2 by relating the problem to a question in the theory of intersecting systems of finite sets. This set theoretic question is investigated further in section 6. Section 7 constructs a family of 3-frameproof codes of length 5 to show that the improved upper bound given in sections 5 and 6 is tight in this case. Finally, the paper ends with a brief discussion of open problems.

2. An upper bound.

THEOREM 1. *Let ℓ, q, and c be positive integers such that $c \geq 2$ and $\ell \geq 2$. Let C be a q-ary c-frameproof code of length ℓ with cardinality n greater than q. Define the integer $r \in \{0, 1, \ldots, c-1\}$ to be the remainder of ℓ on division by c. Then*

$$n \leq \max\left\{ q^{\lceil \ell/c \rceil}, r\left(q^{\lceil \ell/c \rceil} - 1\right) + (c - r)\left(q^{\lfloor \ell/c \rfloor} - 1\right) \right\}. \tag{1}$$

We remark that for almost all parameter sets, the second term on the right-hand side of (1) is the largest.

Proof. Let C be a q-ary length ℓ c-frameproof code of cardinality n. We show that the bound (1) holds. For any subset $S \subseteq \{1, 2, \ldots, \ell\}$, define U_S by

$$U_S = \{x \in C : \text{ there exists no } y \in C \setminus \{x\} \text{ such that } x_i = y_i \text{ for all } i \in S\}.$$

Note that $|U_S| \leq q^{|S|}$, since every codeword $x \in U_S$ is uniquely identified by the subword $(x_i : i \in S)$. Moreover, if $n > q^{|S|}$ then $|U_S| \leq q^{|S|} - 1$, since at least one choice of the subword $(x_i : i \in S)$ must correspond to two or more codewords in C.

Let $S_1, S_2, \ldots, S_c \subseteq \{1, 2, \ldots, \ell\}$ be disjoint subsets, where $|S_j| = \lceil \ell/c \rceil$ whenever $1 \leq j \leq r$ and $|S_j| = \lfloor \ell/c \rfloor$ whenever $r + 1 \leq j \leq c$. So $\cup_{j=1}^{c} S_j = \{1, 2, \ldots, \ell\}$. The bound of the theorem follows if we can show that $C = \cup_{j=1}^{c} U_{S_j}$.

Suppose, for a contradiction, that $x \in C \setminus \cup_{j=1}^{c} U_{S_j}$. So there exist $x^1, x^2, \ldots, x^c \in C \setminus \{x\}$ such that x^j and x agree in their ith components for all $i \in S_j$. But then $x \in \operatorname{desc}(\{x^1, x^2, \ldots, x^c\})$, which contradicts the c-frameproof property of C. This contradiction shows that $C = \cup_{j=1}^{c} U_{S_j}$, as required. □

COROLLARY 2. *A q-ary c-frameproof code of length ℓ contains at most*

$$tq^{\lceil \ell/c \rceil} + O(q^{\lceil \ell/c \rceil - 1})$$

codewords, where t is the unique integer such that $t \in \{1, 2, \ldots, c\}$ and $t = \ell \bmod c$.

3. Two constructions. This section presents two constructions of frameproof codes; the second of these constructions is given in Cohen and Encheva [5, Proposition 1].

CONSTRUCTION 1. *Let $F = \{0, 1, \ldots, q - 1\}$. The set C of all words of length ℓ and weight exactly 1 (i.e., the elements of F^ℓ with exactly one nonzero component) forms a c-frameproof code of cardinality $\ell(q - 1)$.*

Proof. Let $x \in C$ be a weight 1 vector, and suppose its ith component is nonzero. Now, any set $P \subseteq C$ such that $x \in \operatorname{desc}(P)$ must contain a codeword y such that $y_i = x_i$. But since a codeword of weight 1 is uniquely determined by its nonzero component, we must have that $x = y$. Hence C is c-frameproof for any c. □

Theorem 1 and Construction 1 combine to show the following result.

COROLLARY 3. *Let q, ℓ, and c be positive integers such that $q \geq 2$ and $2 \leq \ell \leq c$. Then the largest q-ary length ℓ c-frameproof code has cardinality $\ell(q - 1)$.*

CONSTRUCTION 2. *Let integers ℓ and c be such that $\ell \geq 2$ and $c \geq 2$. Let q be a prime power such that $q \geq \ell$. Let F be the finite field of cardinality q and let $\alpha_1, \alpha_2, \ldots, \alpha_\ell \in F$ be distinct. Define a length ℓ code C over F by*

$$C = \{(f(\alpha_1), f(\alpha_2), \ldots, f(\alpha_\ell)) : f \in F[X] \text{ and } \deg f < \lceil \ell/c \rceil\}.$$

Then C is a c-frameproof code of cardinality $q^{\lceil \ell/c \rceil}$.

We remark that the restriction $q \geq \ell$ may be weakened to $q + 1 \geq \ell$ by also allowing a polynomial f to be evaluated at a "point at infinity": $f(\infty)$ is defined to be the coefficient of $X^{\lceil \ell/c \rceil - 1}$ in f.

Proof. There are $q^{\lceil \ell/c \rceil}$ choices for a polynomial f of degree less than $\lceil \ell/c \rceil$ as there are q choices for each of its coefficients.

If $x, y \in C$ agree in $\lceil \ell/c \rceil$ positions, then $x = y$ (since we may recover the polynomial associated with a codeword by interpolation by considering just the positions where x and y agree). In particular, each distinct choice for the polynomial f gives rise to a distinct codeword, since f is determined by specifying $f(\alpha)$ at $\lceil \ell/c \rceil$ points α. Hence $|C| = q^{\lceil \ell/c \rceil}$. Now, let $x \in C \cap \operatorname{desc}(P)$, where $P \subseteq C$ has cardinality at most c. Each component of x must agree with the corresponding component of one of the codewords in P, and so there is a codeword $y \in P$ that agrees with x in at least $\lceil \ell/c \rceil$ positions. But then $x = y \in P$, and so the code is c-frameproof. □

Corollary 2 and Construction 2 combine to show the following two results.

COROLLARY 4. *Let ℓ and c be fixed integers such that $\ell \geq 2$ and $c \geq 2$. Let $M_{c,\ell}(q)$ be the largest cardinality of a q-ary c-frameproof code of length ℓ. Then*

$$\lim_{q \to \infty} \log_q M_{c,\ell}(q) = \lceil \ell/c \rceil.$$

Proof. We have that $\log_q M_{c,\ell}(q) \leq \lceil \ell/c \rceil + o(1)$ by Corollary 2.

For a given value of q, let q' be the largest prime power such that $q' \leq q$. By the prime number theorem, $q'/q = 1 - o(1)$. By Construction 2, we have that $\log_{q'} M_{c,\ell}(q') \geq \lceil \ell/c \rceil$ whenever q' is sufficiently large. Hence

$$\log_q M_{c,\ell}(q) \geq \log_q M_{c,\ell}(q') \geq \log_{q'} M_{c,\ell}(q') - o(1) \geq \lceil \ell/c \rceil - o(1).$$

These bounds on $\log_q M_{c,\ell}(q)$ imply that $\lim_{q\to\infty} \log_q M_{c,\ell}(q)$ exists and is equal to $\lceil \ell/c \rceil$, as required. □

The proof of the following corollary is similar to the proof of Corollary 4.

COROLLARY 5. *Let ℓ and c be fixed integers such that $\ell \geq 2$, $c \geq 2$, and $\ell = 1 \bmod c$. Let $M_{c,\ell}(q)$ be defined as in Corollary 4. Then*

$$\lim_{q \to \infty} M_{c,\ell}(q)/q^{\lceil \ell/c \rceil} = 1.$$

4. 2-frameproof codes of even length. We aim to construct a family of 2-frameproof codes of length ℓ, where ℓ is even. This construction, when combined with Construction 2, will show that the leading term of the upper bound given in Theorem 1 is tight in the case when $c = 2$.

We define two subcodes as part of our final construction. Let ℓ be an even integer such that $\ell \geq 4$. Let m be a prime power such that $m \geq \ell + 1$ and set $q = m^2 + 1$. Let $\mathbb{F}_m$ be the finite field of order m, and define F to be the disjoint union $F = \{\infty\} \cup (\mathbb{F}_m)^2$. Let $\beta_0, \beta_1, \alpha_1, \alpha_2, \ldots, \alpha_{\ell-1}$ be distinct elements of $\mathbb{F}_m$. For polynomials $f, g \in \mathbb{F}_m[X]$, we write $(f,g)(\alpha_i)$ for the element $(f(\alpha_i), g(\alpha_i)) \in F$. Define $C_1 \subseteq F^\ell$ by

$$C_1 = \{(\infty, (f,g)(\alpha_1), (f,g)(\alpha_2), \ldots, (f,g)(\alpha_{\ell-1}))\}, \tag{2}$$

where $f, g \in \mathbb{F}_m[X]$ are such that $\deg f = (\ell/2) - 1$ and $\deg g \leq (\ell/2) - 1$. Define $C_2 \subseteq F^\ell$ by

$$C_2 = \{((t(\beta_0), t(\beta_1)), (s,t)(\alpha_1), (s,t)(\alpha_2), \ldots, (s,t)(\alpha_{\ell-1}))\}, \tag{3}$$

where $s, t \in \mathbb{F}_m[X]$ are such that $\deg s \leq (\ell/2) - 2$ and $\deg t \leq (\ell/2)$.

CONSTRUCTION 3. *Let ℓ be an even integer such that $\ell \geq 4$. Let m be a prime power such that $m \geq \ell + 1$ and set $q = m^2 + 1$. Define C_1 and C_2 as above. Then the code C defined by $C = C_1 \cup C_2$ is a 2-frameproof code of cardinality $2(q-1)^{\ell/2}(1 - 1/(2\sqrt{q-1}))$.*

Proof. By considering their first components, it is clear that C_1 and C_2 are disjoint. A polynomial of degree at most $(\ell/2) - 1$ is determined by its values at $\ell/2$ distinct points, and hence the polynomials f and g in (2) are uniquely determined by a codeword $x \in C_1$. There are $m^{\ell/2} - m^{(\ell/2)-1}$ choices for f and there are

$m^{\ell/2}$ choices for g, and so $|C_1| = (m^2)^{\ell/2}(1 - 1/m)$. The polynomial s in (3) is determined by $(\ell/2)$ of the final $\ell - 1$ components of a codeword $x \in C_2$. Similarly, the polynomial t is determined by $(\ell/2) + 1$ of these components. Hence $|C_2|$ is equal to the number of choices for s and t and so $|C_2| = m^{(\ell/2)-1}m^{(\ell/2)+1} = (m^2)^{\ell/2}$. Summing our expressions for $|C_1|$ and $|C_2|$ and using the fact that $m = \sqrt{q-1}$ shows that $|C| = 2(q-1)^{\ell/2}(1 - 1/(2\sqrt{q-1}))$, as required.

It remains to show that C is a 2-frameproof code. To this end, we claim that codewords $x \in C_1$ and $y \in C_2$ can agree in at most $(\ell/2) - 1$ components. The first components of x and y are never equal. If $\ell/2$ of the remaining positions agree, then the definitions of C_1 and C_2 imply that a polynomial f of degree exactly $(\ell/2) - 1$ and a polynomial s of degree at most $(\ell/2) - 2$ agree at $\ell/2$ points. This contradiction establishes our claim.

Let $P \subseteq C$ be such that $|P| = 2$. Let $x \in \mathrm{desc}(P) \cap C$. We must show that $x \in P$.

Suppose that $x \in C_1$. Excluding the first coordinate, there are $\ell - 1$ coordinates, and so x must agree with some member $y \in P$ in $\lceil \ell/2 \rceil = \ell/2$ positions other than the first. Since x and y agree in more than $(\ell/2) - 1$ positions, we must have that $y \in C_1$. But any $\ell/2$ of the last $\ell - 1$ components determine a codeword in C_1, and so $x = y$. Hence $x = y \in P$, as required.

Now suppose that $x \in C_2$. Let $y \in P$ be such that $x_1 = y_1$ (and so $y \in C_2$). If x and y agree on $(\ell/2) - 1$ or more of the last $\ell - 1$ components, then the components on which x and y agree include $(\ell/2) - 1$ values of s and $(\ell/2) + 1$ values of t, and so $x = y$. Thus $x = y \in P$ in this case. Now suppose that x and y agree on less than $(\ell/2) - 1$ of the last $\ell - 1$ components. If we define z to be the element of P not equal to y, we have that x and z agree in at least $(\ell/2) + 1$ components. This implies that $z \in C_2$, and since the components on which x and z agree include at least $\ell/2$ values of s and $(\ell/2) + 1$ values of t, we have that $x = z$. Hence $x = z \in P$ in this case also, and so C is a 2-frameproof code. □

COROLLARY 6. *In the notation of Corollary* 4,

$$\lim_{q\to\infty} M_{2,\ell}(q)/q^{\lceil \ell/2 \rceil} = 1 \textit{ when } \ell \textit{ is odd},$$

$$\lim_{q\to\infty} M_{2,\ell}(q)/q^{\lceil \ell/2 \rceil} = 2 \textit{ when } \ell \textit{ is even}.$$

5. An improved upper bound.

Given Corollaries 3 and 6, it might be tempting to conjecture that the leading term of Theorem 1 is always tight. However, this is not the case. This section reduces the problem of providing an improved upper bound to a problem in extremal set theory. This latter problem will be considered in section 6.

Let ℓ and k be fixed integers, where $1 \leq k \leq \ell$. Let D be a set, and let

$$(V_S \subseteq D : S \subseteq \{1, 2, \ldots, \ell\}, |S| = k)$$

be a family of subsets of D indexed by the subsets of $\{1, 2, \ldots, \ell\}$ of cardinality k. We say that this family is a $(k, \ell; b, t)$-*frameproof code set system* (FPCSS) if $|V_S| \leq b$ for all subsets S of $\{1, 2, \ldots, \ell\}$ of cardinality k, and if

$$V_{S_1} \cup V_{S_2} \cup \cdots \cup V_{s_t} = D \tag{4}$$

whenever $S_1, S_2, \ldots, S_t$ are pairwise disjoint subsets of $\{1, 2, \ldots, \ell\}$ of cardinality k. We define the *size* of a $(k, \ell; b, t)$-FPSS to be $|D|$.

We aim to show (see Lemma 7) that a frameproof code gives rise to an FPCSS of comparable size. If we can determine the largest size of an FPCSS, then this will provide an upper bound on the size of a frameproof code.

LEMMA 7. *Let q, c, and ℓ be positive integers, and suppose that $\ell > c$. Let C be a q-ary c-frameproof code of length ℓ containing n codewords. Let $t \in \{1, 2, \dots, c\}$ be such that $t = \ell \bmod c$. Let $k = \lceil \ell/c \rceil$. Then there exists a $(k, \ell; q^k, t)$-FPCSS of size at least*

$$n - \tbinom{\ell}{k-1} q^{k-1}.$$

Proof. As in section 2, for any $S \subseteq \{1, 2, \dots, \ell\}$ we define U_S to be the set of codewords $x \in C$ which are uniquely determined by the ordered subset $(x_i : i \in S)$ of their components. Just as in the proof of Theorem 1, we may show that

$$C = U_{S_1} \cup U_{S_2} \cup \cdots \cup U_{S_c},$$

whenever $S_1, S_2, \dots, S_c$ are subsets of $\{1, 2, \dots, \ell\}$ with the property that $S_1 \cup S_2 \cup \cdots \cup S_c = \{1, 2, \dots, \ell\}$.

We define an FPCSS as follows. Let

$$D = C \setminus \left(\bigcup_S U_S \right),$$

where S runs through all subsets of $\{1, 2, \dots, \ell\}$ of cardinality $k - 1$. We observed in the proof of Theorem 1 that $|U_S| \le q^{|S|}$, and so

$$|D| \ge n - \tbinom{\ell}{k-1} q^{k-1}.$$

For any subset $S \subseteq \{1, 2, \dots, \ell\}$ such that $|S| = k$, we define

$$V_S = U_S \cap D.$$

Clearly, $|V_S| \le |U_S| \le q^k$.

It remains to show that the subsets V_S do indeed form a $(k, \ell; q^k, t)$-FPCSS. Let $S_1, S_2, \dots, S_t$ be a set of pairwise disjoint subsets of $\{1, 2, \dots, \ell\}$ of cardinality k. We need to show that $V_{S_1} \cup V_{S_2} \cup \cdots \cup V_{S_t} = D$. The number of elements of $\{1, 2, \dots, \ell\}$ which are not contained in $S_1 \cup S_2 \cup \cdots \cup S_t$ is $\ell - tk = (c - t)(k - 1)$. Hence there exist subsets $S_{t+1}, S_{t+2}, \dots, S_c$ of cardinality $k - 1$ such that

$$S_1 \cup S_2 \cup \cdots \cup S_c = \{1, 2, \dots, \ell\}.$$

By our definition of D, we have that $U_{S_i} \cap D = \emptyset$ whenever $i \ge t + 1$. Hence

$$\begin{aligned} V_{S_1} \cup V_{S_2} \cup \cdots \cup V_{S_t} &= (U_{S_1} \cup U_{S_2} \cup \cdots \cup U_{S_t}) \cap D \\ &= (U_{S_1} \cup U_{S_2} \cup \cdots \cup U_{S_c}) \cap D \\ &= C \cap D \\ &= D. \end{aligned}$$

Thus our sets form an FPCSS as claimed, and so the lemma follows. □

We now introduce the problem in extremal set theory that we will be concerned with. We say that a family $\mathcal{S}$ of subsets of a set is *t-colliding* if $\mathcal{S}$ does not contain t pairwise disjoint subsets.

Let t, k, and ℓ be positive integers such that $1 \leq k \leq \ell$. We define $m(t,k,\ell)$ to be the maximum number of subsets in a t-colliding family $\mathcal{S}$ of subsets of $\{1,2,\ldots,\ell\}$, where $|S| = k$ for all $S \in \mathcal{S}$. Note that $m(t,k,\ell) = \binom{\ell}{k}$ when $tk > \ell$, and $m(t,k,\ell) < \binom{\ell}{k}$ otherwise.

THEOREM 8. *Let t, k, ℓ, and b be positive integers such that $tk \leq \ell$. Then a $(k,\ell;b,t)$-FPCSS has size at most*

$$\left(\frac{1}{1 - m(t,k,\ell)/\binom{\ell}{k}}\right) b.$$

We remark that when $tk > \ell$, the condition (4) becomes trivial and so there is no bound on the size of a $(k,\ell;b,t)$-FPCSS.

Proof. Let D be a set, and let (V_S) be a collection of subsets of D that forms a $(k,\ell;b,t)$-FPCSS. We prove our upper bound on $|D|$ by counting, in two ways, the elements of the set

$$K = \{(x,S) : x \in V_S\}, \tag{5}$$

where $S \subseteq \{1,2,\ldots,\ell\}$ is such that $|S| = k$, and where $x \in D$.

There are $\binom{\ell}{k}$ choices for the subset S. Once S is chosen, there are at most b choices for x since $|V_S| \leq b$ by the definition of an FPCSS. Hence $|K| \leq \binom{\ell}{k}b$.

We claim that an element $x \in D$ is contained in V_S for at least $\binom{\ell}{k} - m(t,k,\ell)$ subsets S of cardinality k. Let $\mathcal{S}$ be defined by

$$\mathcal{S} = \{S \subseteq \{1,2,\ldots,\ell\} : |S| = k \text{ and } x \notin V_S\}.$$

Now, $\mathcal{S}$ is t-colliding, for if there exist pairwise disjoint subsets $S_1, S_2, \ldots, S_t \in \mathcal{S}$, then $x \notin V_{S_1} \cup V_{S_2} \cup \cdots \cup V_{S_t}$, which would contradict the FPCSS property (4). Since $\mathcal{S}$ is t-colliding, $|\mathcal{S}| \leq m(t,k,\ell)$, and so our claim follows.

There are $|D|$ choices for the element x in (5), and our claim implies that once x is fixed, there are at least $\binom{\ell}{k} - m(t,k,\ell)$ choices for S such that $(x,S) \in K$. Hence $|K| \geq |D|(\binom{\ell}{k} - m(t,k,\ell))$. But now

$$|D|(\tbinom{\ell}{k} - m(t,k,\ell)) \leq |K| \leq \tbinom{\ell}{k}b,$$

and so the theorem follows. □

The bound of Theorem 8 is tight, as the following example shows. Let t, k, and ℓ be positive integers, and suppose that $tk \leq \ell$. Let $\mathcal{S}$ be a t-colliding family of subsets of $\{1,2,\ldots,\ell\}$ with the property that $|S| = k$ for all $S \in \mathcal{S}$, and suppose that $\mathcal{S}$ consists of $m(t,k,\ell)$ subsets. Define $D = \mathrm{Sym}(\ell)$, the symmetric group on ℓ letters. For any subset $S \subseteq \{1,2,\ldots,\ell\}$ such that $|S| = k$, define

$$V_S = \{\pi \in D : \pi(S) \notin \mathcal{S}\}.$$

Let $S_1, S_2, \ldots, S_t$ be pairwise disjoint subsets of $\{1,2,\ldots,\ell\}$ with $|S_i| = k$ for all $i \in \{1,2,\ldots,t\}$. Let $\pi \in D$ and suppose that $\pi \notin V_{S_1} \cup V_{S_2} \cup \cdots \cup V_{S_t}$. Then $\pi(S_i) \in \mathcal{S}$ for all $i \in \{1,2,\ldots,t\}$ by the definition of V_S. But this implies that $\pi(S_1), \pi(S_2), \ldots, \pi(S_t)$ form a set of t pairwise disjoint subsets in $\mathcal{S}$, contradicting the fact that $\mathcal{S}$ is t-colliding. Hence $\pi \in V_{S_1} \cup V_{S_2} \cup \cdots \cup V_{S_t}$ for all $\pi \in D$, and condition (4) follows.

It is easy to see that $|D| = \ell!$ and that the sets V_S all have cardinality $b = (\binom{\ell}{k} - m(t,k,\ell))k!(\ell-k)!$. Hence D is a $(k,\ell;b,t)$-FPCSS that meets the bound of Theorem 8, as required.

Corollary 9. *Let c and ℓ be integers, and suppose that $c \geq 2$ and $\ell \geq 2$. Let $t \in \{1,2,\ldots,c\}$ be such that $t = \ell \bmod c$. Let C be a q-ary c-frameproof code of length ℓ. As $q \to \infty$ with c and ℓ fixed, we have that*

$$|C| \leq \kappa q^{\lceil \ell/c \rceil} + O(q^{\lceil \ell/c \rceil - 1}),$$

where κ is the constant defined by

$$\kappa = \frac{1}{1 - m(t, \lceil \ell/c \rceil, \ell)/\binom{\ell}{\lceil \ell/c \rceil}}.$$

Proof. The corollary follows by Lemma 7 and Theorem 8 after observing that $t\lceil \ell/c \rceil \leq \ell$. □

6. Intersecting set systems. Recall from the previous section that a family of subsets is t-colliding if it does not contain a set of t pairwise disjoint subsets. Let t, k, and ℓ be positive integers such that $tk \leq \ell$. Define, as before, $m(t,k,\ell)$ to be the maximum size of a t-colliding family $\mathcal{S}$ of subsets of $\{1,2,\ldots,\ell\}$ such that $|S| = k$ for all $S \in \mathcal{S}$. This section proves an upper bound on $m(t,k,\ell)$.

Note that the case when $t = 1$ is trivial: no nonempty family of subsets can be 1-colliding, and so $m(1,k,\ell) = 0$ in this case. We will therefore assume that $t \geq 2$.

The family $\mathcal{M}$ defined by

$$\mathcal{M} = \{S \subseteq \{1,2,\ldots,\ell\} : |S| = k \text{ and } S \cap \{1,2,\ldots,t-1\} \neq \emptyset\}$$

is clearly t-colliding, and $|\mathcal{M}| = \binom{\ell}{k} - \binom{\ell-(t-1)}{k}$. This family provides a lower bound on $m(t,k,\ell)$, which we would expect to be realistic. Indeed, much of the literature on this problem has been concerned with showing that $\mathcal{M}$ is optimal (in the sense that $m(t,k,\ell) = |\mathcal{M}|$) when certain conditions on t, k, and ℓ are met. The famous theorem of Erdős, Ko, and Rado [8] (see Anderson [1]) asserts in our notation that $m(2,k,\ell) = \binom{\ell-1}{k-1}$, and so $\mathcal{M}$ is optimal in the case when $t = 2$. Erdős [7] was the first to consider the problem when $t > 2$; he proves that there exists a constant κ depending only on k such that $\mathcal{M}$ is optimal whenever $\ell > \kappa t$. Bollobás, Daykin, and Erdős [2] show that $\ell > 2k^3 t$ will suffice. In Deza and Frankl [6, section 4], a result of Frankl is mentioned that shows that $\mathcal{M}$ is optimal whenever $\ell > \kappa' k t^2$ for some constant κ'. Deza and Frankl conjecture that $\mathcal{M}$ is optimal whenever $\ell > \kappa'' k t$ for some constant κ''.

Rather than proving that $m(t,k,\ell) = \binom{\ell}{k} - \binom{\ell-(t-1)}{k}$ for certain values of t, k, and ℓ, we would like an upper bound on $m(t,k,\ell)$ that holds for any values of t, k, and ℓ. Such a bound is given in Theorem 11 below. This bound is inspired by Katona's proof [11] of the Erdős–Ko–Rado theorem and is a special case of a bound of Gronau [10]; we include a proof here for the sake of completeness.

Before proving Gronau's bound, we will first consider a simpler situation. Let $\mathbb{Z}_\ell$ denote the integers modulo ℓ. For $a \in \mathbb{Z}_\ell$, define $T_\ell(a) \subseteq \mathbb{Z}_\ell$ by

$$T_\ell(a) = \{a, a+1, a+2, \ldots, a+(k-1)\}.$$

Write $\mathcal{T} = \{T_\ell(a) : a \in \mathbb{Z}_\ell\}$.

Lemma 10. *Let t, k, and ℓ be positive integers such that $\ell \geq tk$. Define the sets $T_\ell(a)$ and the family $\mathcal{T}$ as above. Suppose that $\mathcal{S}$ is contained in $\mathcal{T}$ and is t-colliding. Then $|\mathcal{S}| \leq (t-1)k$.*

We remark that the family $\mathcal{S} = \{T_\ell(a) : 0 \leq a \leq (t-1)k - 1\}$ is t-colliding and meets the bound of Lemma 10.

Proof. We prove the lemma by induction on ℓ. Suppose that $\ell = tk$. In this case, we may partition $\mathcal{T}$ into parts $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_k$, where

$$\mathcal{T}_i = \{T_\ell(i), T_\ell(i+k), T_\ell(i+2k), \ldots, T_\ell(i+(t-1)k)\}.$$

Since $\mathcal{T}_i$ consists of t pairwise disjoint sets, $\mathcal{T}_i$ is not contained in $\mathcal{S}$ and so $|\mathcal{T}_i \cap S| \leq t-1$. Hence

$$\begin{aligned} |\mathcal{S}| &= |(\mathcal{T}_1 \cup \mathcal{T}_2 \cup \cdots \cup \mathcal{T}_k) \cap \mathcal{S}| \\ &= \sum_{i=1}^{k} |\mathcal{T}_i \cap \mathcal{S}| \\ &\leq (t-1)k, \end{aligned}$$

and so the lemma follows when $\ell = tk$.

Assume, as an inductive hypothesis, that $\ell > tk$ and the lemma holds for all smaller values of ℓ. Certainly $\mathcal{S} \neq \mathcal{T}$, and so there exists $c \in \mathbb{Z}_\ell$ such that $T_\ell(c) \notin \mathcal{S}$. We may define a family $\overline{\mathcal{S}}$ of subsets of $\mathbb{Z}_{\ell-1}$ by

$$\begin{aligned} \overline{\mathcal{S}} = &\{T_{\ell-1}(a) : a \in \{0, 1, \ldots, c-1\}, T_\ell(a) \in S\} \\ &\cup \{T_{\ell-1}(a-1) : a \in \{c+1, c+2, \ldots, \ell-1\}, T_\ell(a) \in S\}. \end{aligned}$$

Clearly there is a one-to-one correspondence between the subsets in $\mathcal{S}$ and the subsets in $\overline{\mathcal{S}}$, and so $|\mathcal{S}| = |\overline{\mathcal{S}}|$. Moreover, the cardinality of the intersection of a pair of subsets in $\overline{\mathcal{S}}$ is at least as great as the cardinality of the intersection of the corresponding pair of subsets in $\mathcal{S}$. Hence the fact that $\mathcal{S}$ is t-colliding implies that $\overline{\mathcal{S}}$ is t-colliding. Our inductive hypothesis now implies that $|\overline{\mathcal{S}}| \leq (t-1)k$, and so $|\mathcal{S}| \leq (t-1)k$ as required. The lemma now follows by induction on ℓ. □

Theorem 11. *Let t, k, and ℓ be positive integers, where $tk \leq \ell$. Let $\mathcal{S}$ be a t-colliding family of subsets of $\{1, 2, \ldots, \ell\}$, where $|S| = k$ for all $S \in \mathcal{S}$. Then*

$$|\mathcal{S}| \leq \binom{\ell}{k} \frac{(t-1)k}{\ell}.$$

So Theorem 11 states that $m(t,k,\ell) \leq \binom{\ell}{k}\frac{(t-1)k}{\ell}$. The bound of Theorem 11 is best possible when $t = 1$ (as the problem is trivial) and $t = 2$ (the t-colliding family $\mathcal{M}$ defined near the start of this section provides the appropriate example). In the case when $tk = \ell$, the t-colliding family

$$\mathcal{N} = \{S \subseteq \{1, 2, \ldots, \ell\} : |S| = k \text{ and } 1 \notin S\}$$

contains $\binom{\ell}{k}\frac{(t-1)k}{\ell}$ sets. So Theorem 11 is also best possible in the case when $tk = \ell$.

When t and k are fixed with $\ell \to \infty$, the upper bound on $m(t,k,\ell)$ provided by Theorem 11 has the form $(t-1)\ell^{k-1}/(k-1)! + O(\ell^{k-2})$. But the lower bound on $m(t,k,\ell)$ provided by the t-colliding family $\mathcal{M}$ at the start of the section also has this form, as can be easily seen from the expression $|\mathcal{M}| = \sum_{i=1}^{t-1}\binom{t-1}{i}\binom{\ell-(t-1)}{k-i}$. In particular, the ratio between the upper and lower bounds on $m(t, k, \ell)$ tends to 1 as $\ell \to \infty$ with t and k fixed. So the upper bound of Theorem 11 is the right order of magnitude when ℓ is large.

Proof of Theorem 11. Define $\mathcal{T}$ as above, and let Q be the set of pairs (α, S), where $S \in \mathcal{S}$ and $\alpha : \{1, 2, \ldots, \ell\} \to \mathbb{Z}_\ell$ is a bijection such that $\alpha(S) \in \mathcal{T}$. We will count the elements of Q in two ways.

There are $|\mathcal{S}|$ choices for $S \in \mathcal{S}$. Once S has been chosen, there are ℓ choices for $\alpha(S) \in \mathcal{T}$ and then $k!(\ell - k)!$ choices for a suitable bijection α. Hence

$$|Q| = \ell\,|\mathcal{S}|\,k!(\ell - k)!.$$

We now count the elements of Q in a different way. There are $\ell!$ choices for α. Suppose now that α is fixed. The number of choices for S is $|\mathcal{X}|$, where $\mathcal{X} = \{S \in \mathcal{S} : \alpha(S) \in \mathcal{T}\}$. Now, $\mathcal{X}$ is t-colliding because it is a subfamily of $\mathcal{S}$. Hence the corresponding subfamily $\alpha(\mathcal{X})$ of $\mathcal{T}$ (where $\alpha(\mathcal{X}) = \{\alpha(S) : S \in \mathcal{X}\}$) is t-colliding. Hence $|\mathcal{X}| = |\alpha(\mathcal{X})| \leq (t-1)k$ by Lemma 10. So

$$|Q| \leq \ell!(t-1)k,$$

and therefore

$$\begin{aligned} |\mathcal{S}| &= |Q| / \left(k!(\ell - k)!\ell\right) \\ &\leq \ell!(t-1)k / \left(k!(\ell-k)!\ell\right) \\ &= \binom{\ell}{k} \frac{(t-1)k}{\ell}, \end{aligned}$$

as required. □

Corollary 12. *Let c and ℓ be integers, and suppose that $c \geq 2$ and $\ell \geq 2$. Let $t \in \{1, 2, \ldots, c\}$ be such that $t = \ell \bmod c$. Let C be a q-ary c-frameproof code of length ℓ. Then*

$$|C| \leq \left(\frac{\ell}{\ell - (t-1)\lceil \ell/c \rceil}\right) q^{\lceil \ell/c \rceil} + O(q^{\lceil \ell/c \rceil - 1}).$$

Proof. The corollary follows by combining Corollary 9 with Theorem 11. □

7. A 3-frameproof code of length 5. The first case where the upper bound of section 5 improves on the bound of section 2 is when we are considering q-ary 3-frameproof codes of length 5. The upper bound of section 5 shows that such a q-ary 3-frameproof code has cardinality at most $\frac{5}{3}q^2 + O(q)$. We will now show that the leading term of the bound is tight in this case by constructing a 3-frameproof code of length 5 of sufficiently large cardinality.

We define five sets X_1, X_2, X_3, X_4, and X_5 of words of length 5 over the alphabet $\mathbb{F}_3 \cup \{\infty\}$ as follows:

$$\begin{array}{rclcccccl} X_1 & = & \{(& \infty, & a, & a, & a, & a &) : a \in \mathbb{Z}_3\} \\ X_2 & = & \{(& a, & \infty, & a, & a+1, & a+2 &) : a \in \mathbb{Z}_3\} \\ X_3 & = & \{(& a, & a, & \infty, & a+2, & a+1 &) : a \in \mathbb{Z}_3\} \\ X_4 & = & \{(& a, & a+1, & a+2, & \infty, & a &) : a \in \mathbb{Z}_3\} \\ X_5 & = & \{(& a, & a+2, & a+1, & a, & \infty &) : a \in \mathbb{Z}_3\} \end{array}$$

The sets X_i are clearly pairwise disjoint and have cardinality 3. Moreover, it is not difficult to check that a codeword in $X_1 \cup X_2 \cup X_3 \cup X_4 \cup X_5$ is uniquely determined by specifying two of its components.

Let m be a prime power such that $m \geq 4$. Let $\alpha_1, \alpha_2, \alpha_3$, and α_4 be distinct elements in $\mathbb{F}_m$. Define five sets Y_1, Y_2, Y_3, Y_4, and Y_5 of words of length 5 over the alphabet $\mathbb{F}_m \cup \{\infty\}$ by

$$\begin{aligned}
Y_1 &= \{(\infty, f(\alpha_1), f(\alpha_2), f(\alpha_3), f(\alpha_4)) : f \in \mathbb{F}_m[X], \deg f \leq 1\}\\
Y_2 &= \{(f(\alpha_1), \infty, f(\alpha_2), f(\alpha_3), f(\alpha_4)) : f \in \mathbb{F}_m[X], \deg f \leq 1\}\\
Y_3 &= \{(f(\alpha_1), f(\alpha_2), \infty, f(\alpha_3), f(\alpha_4)) : f \in \mathbb{F}_m[X], \deg f \leq 1\}\\
Y_4 &= \{(f(\alpha_1), f(\alpha_2), f(\alpha_3), \infty, f(\alpha_4)) : f \in \mathbb{F}_m[X], \deg f \leq 1\}\\
Y_5 &= \{(f(\alpha_1), f(\alpha_2), f(\alpha_3), f(\alpha_4), \infty) : f \in \mathbb{F}_m[X], \deg f \leq 1\}
\end{aligned}$$

Clearly the sets Y_i are disjoint and have cardinality m^2. Moreover, if elements $x, y \in Y_i$ agree on two components not including the ith, then $x = y$.

Define sets of words C_1, C_2, C_3, C_4, and C_5 of length 5 over the alphabet $(\mathbb{F}_3 \times \mathbb{F}_m) \cup \{(\infty, \infty)\}$ by

$$\begin{aligned}
C_i = \{&((x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5)) :\\
&(x_1, x_2, x_3, x_4, x_5) \in X_i \text{ and } (y_1, y_2, y_3, y_4, y_5) \in Y_i\}
\end{aligned}$$

for all $i \in \{1, 2, 3, 4, 5\}$. Note that $|C_i| = |X_i| \times |Y_i| = 3m^2$.

Construction 4. *Let q be of the form $3m + 1$, where m is a prime power and $m \geq 4$. Define sets C_1, C_2, C_3, C_4, and C_5 of words of length 5 over the alphabet $F = (\mathbb{F}_3 \times \mathbb{F}_m) \cup \{(\infty, \infty)\}$ as above. Then the code C defined by*

$$C = C_1 \cup C_2 \cup C_3 \cup C_4 \cup C_5$$

is a 3-frameproof code of length 5 and cardinality $\frac{5}{3}q^2 - \frac{10}{3}q + \frac{5}{3}$.

Proof. The subsets C_i are pairwise disjoint and $|C_i| = 3m^2 = \frac{1}{3}(q^2 - 2q + 1)$. Hence the code C has the claimed cardinality. It remains to show that C is 3-frameproof.

For a codeword $x \in C$, let $\pi_1(c)$ be the word in $X_1 \cup X_2 \cup X_3 \cup X_4 \cup X_5$ obtained by replacing each component $(a, b) \in F$ of c by the element $a \in \mathbb{F}_3 \cup \{\infty\}$. Note that $\pi_1(c) \in X_i$ if and only if $c \in C_i$. Similarly, define $\pi_2(c)$ to be the word in $Y_1 \cup Y_2 \cup Y_3 \cup Y_4 \cup Y_5$ obtained by replacing each component $(a, b) \in F$ of c by the element $b \in \mathbb{F}_m \cup \{\infty\}$.

Suppose $x \in C$ and let $P \subseteq C$ be such that $|P| \leq 3$ and $x \in \operatorname{desc}(P)$. We must show that $x \in P$. Now $x \in C_j$ for some $j \in \{1, 2, 3, 4, 5\}$. So the jth component of x is (∞, ∞) and $\pi_1(x) \in X_j$. Since $|P| \leq 3$, there exists $y \in P$ that agrees with x in 2 or more components other than the jth. We aim to show that $x = y$.

Since x and y agree in two or more components, the same is true for $\pi_1(x)$ and $\pi_1(y)$. Hence $\pi_1(y) = \pi_1(x)$. In particular, we have that $\pi_1(y) \in X_j$ and so $y \in C_j$.

Since $x, y \in C_j$, we have that $\pi_2(x), \pi_2(y) \in Y_j$. Moreover, since x and y agree in two components not including the jth, the same is true for $\pi_2(x)$ and $\pi_2(y)$. This implies that $\pi_2(x) = \pi_2(y)$. Since $\pi_1(x) = \pi_1(y)$ and $\pi_2(x) = \pi_2(y)$ we find that $x = y \in P$ as required. □

We remark that the condition $m \geq 4$ in the statement of Construction 4 can be weakened to $m \geq 3$ if we set $\alpha_4 = \infty$ in the definition of the sets Y_i. (See the remark after the statement of Construction 2.)

8. Discussion. Two questions suggest themselves for further work. First, can the upper bound of Corollary 9 be made more explicit by determining the constant $m(t, k, \ell)$ exactly in all cases? Erdős [7] warns that this does not seem easy. Second, is it the case that the upper bound of Corollary 9 is tight? The most tempting cases to consider are when we know the explicit value of $m(t, k, \ell)$ used in Corollary 2, namely when $t = 1$, $t = 2$, and $\ell = tk$. The case $t = 1$ occurs when $\ell = 1 \bmod c$ and has already been dealt with by Corollary 5. The case $t = 2$ occurs when $\ell = 2 \bmod c$. So is there a c-frameproof code of cardinality approximately $(\ell/(\ell - \lceil \ell/c \rceil))q^{\lceil \ell/c \rceil}$ when $\ell = 2 \bmod c$? The case $\ell = tk$ occurs when ℓ is a multiple of c. So is there a c-frameproof code of cardinality approximately $cq^{\ell/c}$ when ℓ is a multiple of c?

Acknowledgments. Many thanks to Jessica Staddon and Rob Waiser for their careful readings of an earlier manuscript. Thanks to Dominic Welsh and Lucia Moura for pointing me in the direction of some literature on intersecting set systems, and thanks to Peter Wild for some useful discussions. Also, thanks to an anonymous referee for suggested improvements to the paper (including a strengthening of Corollaries 4 and 5) and for pointing out the reference [10].

REFERENCES

[1] I. Anderson, *Combinatorics of Finite Sets*, Oxford University Press, Oxford, UK, 1987.

[2] B. Bollobás, D.E. Daykin, and P. Erdős, *Sets of independent edges in a hypergraph*, Quart. J. Math. Oxford Ser. 2, 27 (1976), pp. 25–32.

[3] D. Boneh and J. Shaw, *Collision-secure fingerprinting for digital data*, IEEE Trans. Inform. Theory, 44 (1998), pp. 1897–1905.

[4] B. Chor, A. Fiat, and M. Naor, *Tracing traitors*, in Advances in Cryptology—CRYPTO '94, Y.G. Desmedt, ed., Lecture Notes in Comput. Sci. 839, Springer, Berlin, 1994, pp. 257–270.

[5] G. Cohen and S. Encheva, *Efficient constructions of frameproof codes*, Electron. Lett., 36 (2000), pp. 1840–1842.

[6] M. Deza and P. Frankl, *Erdős–Ko–Rado theorem—22 years later*, SIAM J. Algebraic Discrete Methods, 4 (1983), pp. 419–431.

[7] P. Erdős, *A problem on independent r-tuples*, Ann. Univ. Sci. Budapest Eőtvős Sect. Math., 8 (1965), pp. 93–95.

[8] P. Erdős, C. Ko, and R. Rado, *Intersection theorems for systems of finite sets*, Quart. J. Math. Oxford Ser. 2, 12 (1961), pp. 313–320.

[9] A. Fiat and T. Tassa, *Dynamic traitor tracing*, in Advances in Cryptology—CRYPTO '99, M. Weiner, ed., Lecture Notes in Comput. Sci. 1666, Springer, Berlin, 1999, pp. 354–371.

[10] H.-D.O.F. Gronau, *An extremal problem for set families*, Ann. Inst. Mat. Univ. Nac. Autonoma Mexico, 25 (1985), pp. 1–10.

[11] G.O.H. Katona, *A simple proof of the Erdős–Ko–Rado theorem*, J. Combin. Theory Ser. B, 13 (1972), pp. 183–184.

[12] J.N. Staddon, D.R. Stinson, and R. Wei, *Combinatorial properties of frameproof and traceability codes*, IEEE Trans. Inform. Theory, 47 (2001), pp. 1042–1049.

[13] D.R. Stinson and R. Wei, *Combinatorial properties and constructions of traceability schemes and frameproof codes*, SIAM J. Discrete Math., 11 (1998), pp. 41–53.

SIAM J. Discrete Math.
Vol. 16, No. 3, pp. 511–516

ON THE b-STABLE SET POLYTOPE OF GRAPHS WITHOUT BAD K_4*

DION GIJSWIJT† AND ALEXANDER SCHRIJVER‡

Abstract. We prove that for a graph $G = (V, E)$ without bad K_4 subdivision, and for $b \in \mathbf{Z}_+^{V\cup E}$, the b-stable set polytope is determined by the system of constraints determined by the vertices, edges, and odd circuits. We also prove that this system is totally dual integral. This relates to t-perfect graphs.

Key words. t-perfect, graph, polytope, stable set

AMS subject classifications. 05C69, 90C27, 90C57

DOI. 10.1137/S0895480102417343

Let $G = (V, E)$ be a graph and let $b \in \mathbf{Z}_+^{V\cup E}$. Then a *$b$-stable set* in G is a vector $x \in \mathbf{Z}_+^V$ satisfying $x_v \le b_v$ for every vertex v and $x_u + x_v \le b_{uv}$ for every edge uv. The *b-stable set polytope* of G is defined as the convex hull of the b-stable sets in G.

We will use the following notation. For sets $B \subseteq A$ and a vector $x \in \mathbf{R}^A$, let χ^B be the characteristic vector of B and let $x(B) := x^T\chi^B$. For an edge $\{u, v\}$ we will use the shorthand notation uv.

The vectors in the b-stable set polytope obviously satisfy the following system of inequalities:

$$\text{(1)}\qquad \begin{aligned} &\text{(i)}\quad 0 \le x_v \le b_v \quad \text{for each } v \in V;\\ &\text{(ii)}\quad x_u + x_v \le b_{uv} \quad \text{for each edge } uv \in E;\\ &\text{(iii)}\quad x(VC) \le \lfloor \tfrac{1}{2} b(EC) \rfloor \quad \text{for each odd circuit } C. \end{aligned}$$

We call a graph G *t-perfect with respect to b* if the b-stable set polytope is determined by (1). Since each integral vector satisfying (1) is a b-stable set, the polytope determined by (1) equals the b-stable set polytope if and only if it is integral. We call a graph G *strongly t-perfect with respect to b* if system (1) is totally dual integral.

For any weight function $w \in \mathbf{Z}_+^V$ and any $b \in \mathbf{Z}_+^{V\cup E}$, denote by $\alpha(G, b, w)$ the maximum *w-weight* w^Tx of a b-stable set x in G. Define a *w-cover* as a family of vertices, edges, and odd circuits in G that covers each vertex v at least w_v times. The *b-cost* of a w-cover is defined as the sum of the costs of its elements, where the cost of a vertex v equals b_v, the cost of an edge e equals b_e, and the cost of an odd circuit C equals $\lfloor \frac{1}{2} b(EC) \rfloor$. Denote by $\tilde{\rho}(G, b, w)$ the minimum cost of a w-cover. Strong t-perfection can now be characterized equivalently as follows: a graph $G = (V, E)$ is strongly t-perfect with respect to b if and only if $\alpha(G, b, w) = \tilde{\rho}(G, b, w)$ for every weight function $w \in \mathbf{Z}_+^V$.

Call a subdivision of K_4 *odd* if each triangle of K_4 has become an odd circuit. An odd subdivision of K_4 is called *bad* if there are no two disjoint edges e, f of K_4 such

*Received by the editors November 8, 2002; accepted for publication (in revised form) March 21, 2003; published electronically June 25, 2003.

http://www.siam.org/journals/sidma/16-3/41734.html

†Department of Mathematics, University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands (gijswijt@science.uva.nl).

‡CWI, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands, and Department of Mathematics, University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands (lex@cwi.nl).

that e and f are not subdivided and the other four edges have become even length paths. We say that a graph *has a bad K_4 subdivision* if it has a subgraph that is a bad K_4 subdivision.

In [4], it was proved that a graph has no bad K_4 subdivision if and only if each subgraph is t-perfect with respect to the all-one vector. Here the "if" part follows from the fact that a bad K_4 subdivision is not t-perfect with respect to the all-one vector (see [1]). In [5], it was proved that graphs without bad K_4 subdivision are also strongly t-perfect with respect to the all-one vector. In this paper we prove that graphs having no bad K_4 subdivision are strongly t-perfect with respect to every $b \in \mathbf{Z}_+^{V\cup E}$, which implies our theorem.

THEOREM. *Let $G = (V, E)$ be a graph. Then the following are equivalent:*

(i) *G has no bad K_4 subdivision.*

(ii) *G is* t*-perfect with respect to each $b \in \mathbf{Z}_+^{V\cup E}$.*

(iii) *G is strongly* t*-perfect with respect to each $b \in \mathbf{Z}_+^{V\cup E}$.*

Proof. If G satisfies (ii), then also each subgraph of G satisfies (ii). So the implication (ii) $\Longrightarrow$ (i) follows from the fact that a bad K_4 subdivision is not t-perfect with respect to the all-one vector (see [1]).

The implication (iii) $\Longrightarrow$ (ii) follows from the fact that any totally dual integral system with integral right-hand side determines an integral polyhedron.

To prove the implication (i) $\Longrightarrow$ (iii), it will be convenient to first prove the implication (i) $\Longrightarrow$ (ii). Let $G = (V, E)$ be a graph without bad K_4 subdivision, and let $b \in \mathbf{Z}_+^{V\cup E}$. We show that the polytope P determined by (1) is integral. Suppose that x is a nonintegral vertex of P. Let x' be defined by $x'_v := x_v - \lfloor x_v \rfloor$ for every vertex v, and let b' be defined by $b'_v := b_v - \lfloor x_v \rfloor$ for every vertex v and $b'_e := b_e - \lfloor x_u \rfloor - \lfloor x_v \rfloor$ for every edge $e = uv$. Then x' is a nonintegral vertex of the polytope determined by (1) with b replaced by b'. Let $G' := (V, F)$, where $F := \{e \in E |\ b'_e = 1\}$. Since G' has no bad K_4 subdivision and x' satisfies the constraints (1) for the graph G' and the all-one vector $\chi^{V\cup F}$ instead of b, x' is a convex combination of incidence vectors of stable sets in G' by [5]. Each of these incidence vectors is a b'-stable set. Hence x' is a convex combination of b'-stable sets in G, a contradiction. This proves the implication (i) $\Longrightarrow$ (ii).

The remainder of this proof consists of showing the implication (i) $\Longrightarrow$ (iii). The idea is to reduce the general statement to the case in which b is the all-one vector.

Suppose the implication (i) $\Longrightarrow$ (iii) is false. Let the graph $G = (V, E)$ and $b \in \mathbf{Z}_+^{V\cup E}$ form a counter example with (first) $|V| + |E|$ minimal and (second) $b(V)$ minimal. Let $w \in \mathbf{Z}_+^V$ be any weight function for which $\alpha(G, b, w) < \tilde{\rho}(G, b, w)$. Note that by the minimality of G, we know that G has no isolated vertices. We observe the following facts about w and b.

CLAIM 1.

$$\text{(2)}\qquad \begin{aligned} &\text{(i)} \quad \alpha(G, b, w) < \alpha(G - e, b|_{G-e}, w) \quad \textit{for each edge } e \in E,\\ &\text{(ii)} \quad b_{uv} < b_u + b_v \quad \textit{for each edge } uv \in E,\\ &\text{(iii)} \quad 1 \le b_u \le b_{uv} \quad \textit{for each edge } uv \in E,\\ &\text{(iv)} \quad 1 \le w_v \quad \textit{for each vertex } v \in V. \end{aligned}$$

Proof. By the minimality of G, we know that

$$\alpha(G, b, w) < \tilde{\rho}(G, b, w) \le \tilde{\rho}(G - e, b|_{G-e}, w) = \alpha(G - e, b|_{G-e}, w).$$

This gives (i). If for some edge uv we have $b_{uv} \ge b_u + b_v$, then every $b|_{G-uv}$-stable set in $G - uv$ is a b-stable set in G, contradicting (i). Hence we have (ii). Suppose that $b_u > b_{uv}$ for some edge uv. Let $b' := b - \chi^u$. Now we have

$$\alpha(G, b', w) = \alpha(G, b, w) < \tilde{\rho}(G, b, w) = \tilde{\rho}(G, b', w),$$

contradicting the minimality of b. Hence we have $0 \le b_{uv} - b_v < b_u \le b_{uv}$, and (iii) follows. Suppose that $w_v = 0$ for some vertex v. Let $b' := b|_{G-v}$ and $w' := w|_{G-v}$. Then

$$\alpha(G - v, b', w') = \alpha(G, b, w) < \tilde{\rho}(G, b, w) \le \tilde{\rho}(G - v, b', w'),$$

contradicting the minimality of G. Hence we have (iv). □

For the b-stable sets of maximum weight we have the following.

CLAIM 2. *Let x be a b-stable set of w-weight $w^T x = \alpha(G, b, w)$. Then $x_v \le 1$ for each $v \in V$.*

Proof. To see this, suppose that $x_v > 1$ for some vertex v. Let $x' := x - \chi^v$ and $b' := b - \chi^{\{v\} \cup \delta(v)}$. For any b'-stable set $\tilde{x}$ in G, we have

$$w^T \tilde{x} = w^T(\tilde{x} + \chi^v) - w_v \le \alpha(G, b, w) - w_v = w^T x',$$

and hence x' is a maximum w-weight b'-stable set in G. By minimality of b, there exists a w-cover F of b'-cost $\tilde{\rho}(G, b', w) = \alpha(G, b', w)$.

Since $x_v > 1$, we have $x'_v > 0$, and hence by “complementary slackness” v is covered exactly w_v times by F. This implies that F has b-cost

$$\tilde{\rho}(G, b', w) + w_v = \alpha(G, b', w) + w_v = \alpha(G, b, w),$$

a contradiction. □

CLAIM 3. *For every edge $f \in E$ we have $b_f \le 2$.*

Proof. Suppose that Claim 3 is not true and that we have $b_f \ge 3$ for some edge $f = uv$. Let $w' := w + N \cdot \chi^f$, where $N := w(V) + 1$. Then

$$\alpha(G, b, w') = \tilde{\rho}(G, b, w'),$$

since otherwise by Claim 2 applied to w' we have for any maximum w'-weight b-stable set x the inequality

$$w'^T x = w^T x + N(x_u + x_v) \le N - 1 + 2N < 3N,$$

while $x' := b_u \chi^u + (b_f - b_u)\chi^v$ is a b-stable set of w'-weight

$$w'^T x' \ge N \cdot b_f \ge 3N,$$

contradicting the optimality of x.

So we can choose w such that

$$\begin{aligned} \alpha(G, b, w) &< \tilde{\rho}(G, b, w), \\ \alpha(G, b, w + \chi^f) &= \tilde{\rho}(G, b, w + \chi^f). \end{aligned} \tag{3}$$

Let $F := \{v_1, \dots, v_r, e_1, \dots, e_s, C_1, \dots, C_t\}$ be a minimum b-cost $w + \chi^f$-cover, where the v_i are vertices, the e_i are edges, and the C_i are odd circuits. Note that none of the e_i is the edge f, since otherwise $\tilde{\rho}(G, b, w) \le \tilde{\rho}(G, b, w + \chi^f) - b_f$, which would imply that $\tilde{\rho}(G, b, w) \le \alpha(G, b, w + \chi^f) - b_f \le \alpha(G, b, w)$. Let $G' := G - f$, let

$b' := b|_{G'}$, and let x' be a maximum w-weight b'-stable set in G'. Then $\alpha(G', b', w) \geq \alpha(G, b, w) + 1$ by Claim 1, and hence

$$x'(f) > b_f. \tag{4}$$

For any odd circuit C traversing f, we have

$$x'(VC) \leq \lfloor \tfrac{1}{2} b(EC) \rfloor + \tfrac{1}{2}(x'(f) - b_f + 1), \tag{5}$$

since $2x'(VC) \leq x'(f) + b(EC - f) = x'(f) + b(EC) - b_f$. Now let l be the number of circuits in F traversing f. We obtain

$$\begin{aligned}\tilde{\rho}(G, b, w + \chi^f) &= \alpha(G, b, w + \chi^f) \leq \alpha(G, b, w) + b_f \leq \alpha(G', b', w) - 1 + b_f \\ &= w^T x' - 1 + b_f = (w + \chi^f)^T x' - (x'(f) - b_f + 1) \\ &\leq -(x'(f) - b_f + 1) + \sum_{i=1}^{r} x'(v_i) + \sum_{i=1}^{s} x'(e_i) + \sum_{i=1}^{t} x'(VC_i) \\ &\leq (\tfrac{1}{2}l - 1)(x'(f) - b_f + 1) + \sum_{i=1}^{r} b_{v_i} + \sum_{i=1}^{s} b_{e_i} + \sum_{i=1}^{t} \lfloor \tfrac{1}{2} b(EC_i) \rfloor \\ &= (\tfrac{1}{2}l - 1)(x'(f) - b_f + 1) + \tilde{\rho}(G, b, w + \chi^f).\end{aligned} \tag{6}$$

Hence we have $(l-2)(x'(f) - b_f + 1) \geq 0$. Since $x'(f) - b_f + 1 > 0$ by (4), we have $l \geq 2$.

We may assume that C_1 and C_2 traverse f. Decompose the cycle $EC_1 \Delta EC_2$ into circuits $C'_1, \ldots, C'_q$, where $C'_1, \ldots, C'_p$ are odd and $C'_{p+1}, \ldots, C'_q$ are even. Choose in each C'_i with $i = p+1, \ldots, q$ a perfect matching M_i with $b(M_i) \leq \frac{1}{2} b(EC'_i)$. Now the circuits C_1 and C_2 are removed from the cover F, and the circuits $C'_1, \ldots, C'_p$, the edges in the matchings $M_{p+1}, \ldots, M_q$, and the edges in $EC_1 \cap EC_2$ are added to the cover. This gives a $w + \chi^f$-cover F' of b-cost

$$\begin{aligned}&\tilde{\rho}(G, b, w + \chi^f) - \lfloor \tfrac{1}{2} b(EC_1) \rfloor - \lfloor \tfrac{1}{2} b(EC_2) \rfloor \\ &\qquad + b(EC_1 \cap EC_2) + \sum_{i=1}^{p} \lfloor \tfrac{1}{2} b(EC'_i) \rfloor + \sum_{i=p+1}^{q} b(M_i) \\ &\leq \tilde{\rho}(G, b, w + \chi^f) - \tfrac{1}{2}(b(EC_1) + b(EC_2) - 2) + \tfrac{1}{2}(b(EC_1 \Delta EC_2)) + b(EC_1 \cap EC_2) \\ &\qquad = \tilde{\rho}(G, b, w + \chi^f) + 1.\end{aligned} \tag{7}$$

Hence $F' - f$ is a w-cover of b-cost at most $\tilde{\rho}(G, b, w + \chi^f) + 1 - b_f$. This implies that

$$\begin{aligned}\alpha(G, b, w) \leq \tilde{\rho}(G, b, w) - 1 &\leq \tilde{\rho}(G, b, w + \chi^f) - b_f \\ &= \alpha(G, b, w + \chi^f) - b_f \leq \alpha(G, b, w).\end{aligned} \tag{8}$$

So we have equality throughout and, in particular, we obtain

$$\alpha(G, b, w + \chi^f) = \alpha(G, b, w) + b_f.$$

Let x be a maximum $w + \chi^f$-weight b-stable set in G. Then

$$\alpha(G, b, w) + b_f = (w + \chi^f)^T x = w^T x + x(f) \leq \alpha(G, b, w) + b_f,$$

and hence $x(f) = b_f$ and x is a maximum w-weight b-stable set. However, $x(f) = b_f \geq 3$ implies that $x_u > 1$ or $x_v > 1$, contradicting Claim 2. □

Partition the vertex set V into $V_1 := \{v \in V | \ b_v = 1\}$ and $V_2 := \{v \in V | \ b_v = 2\}$. Thus by Claim 1, we know that the edges e spanned by V_1 have $b_e = 1$ and the other edges have $b_e = 2$. We now prove the following claim.

CLAIM 4. *Either* $V_1 = \emptyset$ *or* $V_2 = \emptyset$.

Proof. To prove the claim, take w with $\alpha(G, b, w) < \tilde{\rho}(G, b, w)$ such that $w(V)$ is minimal. We first prove the following:

$$\begin{aligned}&\text{If } b_v = 1 \text{ for some vertex } v,\\&\text{then there exists a maximum } w\text{-weight } b\text{-stable set } x \text{ with } x_v = 0.\end{aligned} \tag{9}$$

Indeed, let $w' := w - \chi^v$. By the minimality of w, we have

$$\alpha(G, b, w') + 1 = \tilde{\rho}(G, b, w') + 1 \geq \tilde{\rho}(G, b, w) \geq \alpha(G, b, w) + 1.$$

Hence $\alpha(G, b, w') = \alpha(G, b, w)$, implying that there exists a maximum w-weight b-stable set x satisfying $x_v = 0$.

Similarly, we have the following:

$$\begin{aligned}&\text{If } b_e = 1 \text{ for some edge } e,\\&\text{then there exists a maximum } w\text{-weight } b\text{-stable set } x \text{ with } x(e) = 0.\end{aligned} \tag{10}$$

To see this, let $w' := w - \chi^e$. By the minimality of w, we have

$$\alpha(G, b, w') + 1 = \tilde{\rho}(G, b, w') + 1 \geq \tilde{\rho}(G, b, w) \geq \alpha(G, b, w) + 1.$$

Hence $\alpha(G, b, w') = \alpha(G, b, w)$, implying that there exists a maximum w-weight b-stable set x satisfying $x(e) = 0$.

Consider an edge $e = uv$ with $u \in V_1$ and $v \in V_2$. By (9), there is a maximum w-weight b-stable set x with $x_u = 0$. By Claim 2, we know that $x_v \leq 1$. Hence $x(e) \leq 1 < 2 = b_e$. So we have that

$$\begin{aligned}&\text{for each edge } e \in \delta(V_1),\\&\text{there is a maximum } w\text{-weight } b\text{-stable set } x \text{ with } x(e) < b_e.\end{aligned} \tag{11}$$

Next consider an odd circuit traversing an edge in $\delta(V_1)$. We have that

$$\begin{aligned}&\text{for each odd circuit } C \text{ traversing an edge in } \delta(V_1)\text{, there is a}\\&\text{maximum } w\text{-weight } b\text{-stable set } x \text{ with } x(VC) < \lfloor \tfrac{1}{2} b(EC) \rfloor.\end{aligned} \tag{12}$$

Indeed, let C be an odd circuit traversing an edge in $\delta(V_1)$ and suppose that C does not traverse an edge spanned by V_1. Let $u \in V_1$ be a vertex traversed by C. By (9), there is a maximum w-weight b-stable set x with $x_u = 0$. By Claim 2 we have $x(VC) \leq |VC| - 1 < |VC| = \lfloor \frac{1}{2} b(EC) \rfloor$.

Thus we may assume that C traverses an edge spanned by U_1. Then C has three consecutive vertices t, u, and v with $t, u \in V_1$, and $v \in V_2$. By (10) there is a maximum w-weight b-stable set x with $x(tu) = 0 = b_{tu} - 1$. By Claim 2 we have $x_v \leq 1$, and hence $x(uv) \leq 1 \leq b_{uv} - 1$. Thus $2x(VC) \leq b(EC) - 2$, and hence $x(VC) < \lfloor \frac{1}{2} b(EC) \rfloor$.

Now suppose that V_1 and V_2 are nonempty. By minimality of G, we know that there is at least one edge $e \in \delta(V_1)$. Let $G' := G - e$, let $b' := b|_{G'}$, and let x' be a maximum w-weight b'-stable set in G'. Let x maximize $w^T x$ over the b-stable set polytope of G such that x is in general position on the face of optimal solutions. Then by (11) and (12), $x(e) < b_e$ and $x(VC) < \lfloor \frac{1}{2} b(EC) \rfloor$ for each odd circuit traversing e. Hence there is a $0 < \lambda \leq 1$ such that $\tilde{x} := (1-\lambda)x + \lambda x'$ satisfies the system of constraints (1). By the implication (i) $\Longrightarrow$ (ii), $\tilde{x}$ belongs to the b-stable set polytope of G. However, $w^T \tilde{x} > w^T x$, since $w^T x' = \alpha(G', b', w) > \alpha(G, b, w) = w^T x$ by Claim 1. This contradicts the optimality of x. So either V_1 or V_2 is empty. □

If b is the all-one vector, the total dual integrality of (1) follows from [5]. So V_1 is empty, and hence $b_e = 2$ for every edge e and $b_v = 2$ for every vertex v. Denote by a $2w$-edge cover a vector $y \in \mathbf{Z}_+^E$ with $y(\delta(v)) \geq 2w_v$ for every vertex $v \in V$. It is easy to see that for any $2w$-edge cover y and any 2-stable set x, we have $w^T x \leq \sum_{e \in E} y_e \frac{1}{2} \sum_{v \in e} x(v) \leq y(E)$. By a theorem of Gallai (see [2]), G has a 2-stable set x and a $2w$-edge cover such that $w^T x = y(E)$. Denote by U_y the set of vertices v for which $y(\delta(v))$ is odd. Let x be a 2-stable set and let y be a $2w$-edge cover such that $w^T x = y(E)$ and $|U_y|$ is minimal.

If $U_y \neq \emptyset$, then there is a simple path P connecting two vertices in U_y with $y_e \geq 1$ for each $e \in EP$. Let M be a maximum size matching in P. Then $y' := y + \chi^{EP} - 2\chi^M$ is a $2w$-edge cover with $y'(E) \leq y(E)$ and $|U_{y'}| = |U_y| - 2$, a contradiction. So $y(\delta(v))$ is even for every vertex v and we can write $y = \chi^{EC_1} + \cdots + \chi^{EC_r} + \chi^{EC'_1} + \cdots + \chi^{EC'_s}$ for odd circuits $C_1, \ldots, C_r$ and even circuits $C'_1, \ldots, C'_s$. Let M_i be a perfect matching in C'_i for $i = 1, \ldots, s$. Then $C_1, \ldots, C_r$ together with the edges in the matchings $M_1, \ldots, M_s$ give a w-cover of b-cost $y(E) = w^T x$. Since x is a b-stable set, this implies that $\tilde{\rho}(G, b, w) \leq \alpha(G, b, w)$, contradicting the choice of w. This concludes the proof of the theorem. □

Remark. Let $G = (V, E)$ be a graph with $E \times V$ incidence matrix M. In [3] it was proved that the matrix

$$\begin{pmatrix} I \\ -I \\ M \\ -M \end{pmatrix}$$

has Chvátal rank at most 1 if and only if G has no odd K_4 subdivision. The equivalence of (i) and (ii) of the theorem above has the following reformulation in terms of the Chvátal rank: the matrix

$$\begin{pmatrix} I \\ -I \\ M \end{pmatrix}$$

has Chvátal rank at most 1 if and only if G has no *bad* K_4 subdivision.

REFERENCES

[1] F. Barahona and A.R. Mahjoub, *Composition of graphs and polyhedra* III: *Graphs with no W_4 minor*, SIAM J. Discrete Math., 7 (1994), pp. 372–389.

[2] T. Gallai, *Maximum-minimum Sätze über Graphen*, Acta Math. Acad. Sci. Hungar., 9 (1958), pp. 395–434.

[3] A.M.H. Gerards and A. Schrijver, *Matrices with the Edmonds-Johnson property*, Combinatorica, 6 (1986), pp. 365–379.

[4] A.M.H. Gerards and F.B. Shepherd, *The graphs with all subgraphs* t-*perfect*, SIAM J. Discrete Math., 11 (1998), pp. 524–545.

[5] A. Schrijver, *Strong* t-*perfection of bad-K_4-free graphs*, SIAM J. Discrete Math., 15 (2002), pp. 403–415.

SIAM J. DISCRETE MATH.
Vol. 16, No. 4, pp. 517–523

BIPARTITE DOMINATION AND SIMULTANEOUS MATROID COVERS*

C. W. KO† AND F. B. SHEPHERD‡

Abstract. Damaschke, Müller, and Kratsch [*Inform. Process. Lett.*, 36 (1990), pp. 231–236] gave a polynomial-time algorithm to solve the minimum dominating set problem in *convex* bipartite graphs $B = (X \cup Y, E)$, that is, where the nodes in Y can be ordered so that each node of X is adjacent to a contiguous sequence of nodes. Gamble et al. [*Graphs Combin.*, 11 (1995), pp. 121–129] gave an extension of their algorithm to weighted dominating sets. We formulate the dominating set problem as that of finding a minimum weight subset of elements of a graphic matroid, which covers each fundamental circuit and fundamental cut with respect to some spanning tree T. When T is a directed path, this *simultaneous covering problem* coincides with the dominating set problem for the previously studied class of convex bipartite graphs. We describe a polynomial-time algorithm for the more general problem of simultaneous covering in the case when T is an arborescence. We also give NP-completeness results for fairly specialized classes of the simultaneous cover problem. These are based on connections between the domination and induced matching problems.

Key words. matroids, network matrices, induced matching, domination, linear programming

AMS subject classifications. 05C50, 05C75

DOI. 10.1137/S089548019828371X

1. The problem and some background.

We are interested in minimum *covering problems* of the form $\min\{wx : Ax \geq 1,\ x_i = 0 \text{ or } 1,\ i = 1, \ldots, n\}$, where $w \in \mathbf{Z}^n$ and A is a $0,1$ matrix. For a graph $G = (V, E)$, a *dominating set* is a set $S \subset V$ such that for each node $u \in V$, u either is in S or is adjacent to a node in S. The dominating set problem is the covering problem associated with the $0,1$ matrix whose rows are the incidence vectors of closed neighborhoods (i.e., $N[u] = \{u\} \cup \{v : uv \in E\}$). Computing the size of a minimum dominating set is NP-hard in general graphs; in fact, it is hard to approximate to within a $\log(n)$ factor [6].

In [2], Damaschke, Müller, and Kratsch outline a polynomial-time algorithm to solve the minimum dominating set problem in bipartite graphs $G = (X \cup Y, E)$, where the nodes of Y can be labeled as $1, 2, \ldots$ so that for each node $u \in X$, $N[u] = \{i, i+1, i+2, \ldots, j\}$ for some choice of $i < j$. Such graphs are called *convex bipartite graphs*. The algorithm of Damaschke, Müller, and Kratsch was extended to solve the minimum weight dominating set problem in [5], where convex graphs arose in the study of finding maximum right-angle-free subsets of points in the plane. Both of these algorithms are rather complex and do not elucidate the reasons for polynomial-time solvability in this class. Attempting to rectify this, we give an edge-based formulation of the problem. This formulation leads naturally to a more general version of the problem for which a simpler algorithm can be described. This approach originates from considering the integer and linear programming formulations for domination; we thus start our exposition this way as well.

Consider the (natural) integer program for domination in a convex bipartite graph. Let B be the $X \times Y$ $0,1$ matrix such that $B_{ij} = 1$ if the edge $ij \in E(B)$. Then

*Received by the editors February 2, 1998; accepted for publication (in revised form) February 26, 2003; published electronically July 30, 2003.

http://www.siam.org/journals/sidma/16-4/28371.html

†Citigroup, 390 Greenwich St., New York, NY 10043 (chun.ko@citigroup.com).

‡Mathematical Sciences, Bell Laboratories, Murray Hill, NJ 07974 (bshep@research.bell-labs.com).

B has the consecutive ones property (in each row) and hence is totally unimodular. It follows that the matrix

$$A = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix} \tag{1}$$

is also totally unimodular. The *covering problem associated with* A is thus solvable by linear programming since the region $\{x \in \mathbf{Q}^n : Ax \geq 1,\ x \geq 0\}$ has integral extreme points (cf. [7]). The dominating set problem in convex graphs, however, corresponds to the covering problem for the matrix $A + I$, and it is not the case that $A + I$ is totally unimodular whenever A is.

Nevertheless, we briefly explore the nature of covering problems associated with such matrices as $A + I$. A matrix $[I\ B]$ can be signed to be totally unimodular if and only if its rows are the incidence vectors of fundamental cuts of some *regular matroid* $\mathcal{M} = (E, C)$, relative to the basis consisting of the first m columns (where B has m rows). In this case, we also have that $[B^T\ I]$ is the incidence matrix of the fundamental circuits of $\mathcal{M}$. Thus the covering problem for $A + I$ (where A is symmetric and totally unimodular) can be viewed as the problem of covering (over $\mathbf{Z}_2$) all fundamental circuits and cuts for a regular matroid. We formulate our dominating set problem as a special case of this covering problem, which may be of independent interest.

SIMULTANEOUS MATROID COVER PROBLEM (SMC).

Given: A matroid $\mathcal{M} = (E, C)$ and a basis B.

Find: A minimum (weight) subset of E with nonempty intersection with each fundamental (with respect to B) circuit and each cocircuit of $\mathcal{M}$.

This problem is NP-hard in general, although we now see that domination in convex graphs is a special case of the problem for graphic matroids. Recall that a *network matrix* (Tutte [8]) is derived from a *network pair* (D, T), where D is a digraph (V, E) and T a spanning subtree. In the network matrix $A_{D,T}$, there is a row for each arc in T and a column for each arc not in T. The element in row a, column (u, v) contains a 0 if a does not appear on the path in T between u and v. Otherwise, if a appears in the forward direction on the path, then this entry is 1 and, if a appears in the reverse direction, then the element is -1. The tree T is *compatible* with D (and (D, T) is said to be *compatible*) if $A_{D,T}$ has only $0, 1$ entries.[1] We then also call $A_{D,T}$ a compatible matrix.

The convex dominating set problem arises as the special case of simultaneous covering, where we restrict our attention to network pairs (D, T), where T is a directed path and D is an acyclic digraph; i.e., a matrix B with the consecutive ones property arises as the network matrix for such a pair. (Note that B^T need not be a network matrix, however.) We extend this class to consider *single-source* instances of SMC, i.e., those consisting of a compatible pair D, T, where T is an arborescence rooted at some node r. In section 3 we prove the main algorithmic result.

THEOREM 1.1. *Single-source* SMC *is solvable by a polynomial-time algorithm.*

On the negative side, we show bipartite domination is NP-hard even if restricted to compatible matrices B in (1). This is a byproduct of the following result proved in section 2.

[1] Given an arbitrary oriented matroid, one may ask whether it has a compatible basis. Fonlupt and Raco [4] give a polynomial-time algorithm to transform *any* totally unimodular matrix into a $0, 1$ matrix by pivoting and multiplying rows or columns by -1 (i.e., by changing the basis and re-orienting the elements of the matroid).

THEOREM 1.2. SMC *is NP-hard when restricted to graphic matroid pairs* (D, T), *where* $D - E(T)$ *is a planar bipartite graph of maximum degree* 4 *and* T *is a star.*

We mention that we do not know the complexity of bipartite domination in the case in which the adjacency matrix (1) has both B and B^T compatible. In this case, a result of Whitney implies that these matrices arise from a graphic matroid in a self-dual planar digraph D. We remark that the techniques from section 3 likely extend to the case of a fixed number of sources; the main challenge is to determine whether there is an algorithm whose running time is independent of the number of sources. Along the way, we also give a short proof, and strengthening, of an earlier complexity result due to Cameron [1] on the induced matching problem.

2. The complexity results. A subset of the nodes of a graph $G = (V, E)$ is a *dominating set* if each node either is in the set or is adjacent to an element in the set. We denote by $\gamma(G)$ the minimum cardinality of a dominating set of G. We denote by $\sigma(G)$ the maximum size of an *induced matching* of G, i.e., a subset of edges whose endpoints induce a 1-regular graph. The induced matching problem is NP-hard for planar graphs of maximum degree 4. To see this, suppose first that G is planar, and consider the graph G' obtained by hanging a pendant leaf from each node of G. Clearly G' is planar. It is also straightforward to reason that $\sigma(G')$ is equal to the size of a maximum stable set of G. Since the maximum stable set problem is NP-hard for 3-regular planar graphs, the induced matching problem is NP-hard for planar graphs of maximum degree 4 as follows:

$$\text{The parameter } \sigma \text{ is NP-hard to compute} \\ \text{for the class of planar graphs of maximum degree 4.} \tag{2}$$

It is well known that the dominating set problem is NP-hard for bipartite graphs. A more surprising result states that finding a maximum induced matching is NP-hard for bipartite graphs (Cameron [1]). The previous trick of reducing the stable set problem does not work, as the stable set problem is polynomially solvable for bipartite graphs. In this sense, induced matching is even harder than the stable set problem. We later give a strengthening of Cameron's result that relies only on statement (2) and a formula which we give relating the parameters for the dominating set and induced matching problems.

For a graph G and integer $k > 0$, let G_k denote the graph obtained from G by recursively subdividing each edge k times; i.e., each edge is replaced by a path of 2^k edges. Then we have the following.

LEMMA 2.1. *For any graph* G, $\gamma(G_1) + \sigma(G) = |V_G|$.

Proof. First suppose that M is a maximum induced matching of G. If we let D consist of those degree 2 nodes of G_1 which lie on the edges of M, together with the nodes of G which are not incident to any edge of M, then we check that D is a dominating set of G_1. We also have $\gamma(G_1) \leq |D| = (|V_G| - 2|M|) + |M| = |V_G| - \sigma(G)$. So let D be a minimum dominating set of G_1 with a minimum number of degree 2 nodes. Let x_1, x_2 be two degree 2 nodes in D and let e_1, e_2 be the edges of G which correspond to them. If $\{e_1, e_2\}$ is not an induced matching of size two in G, then there is some edge $e_3 = v_1v_2 \in E_G$ such that v_i is incident to e_i (or e_1 and e_2 are incident edges; we consider this case next). Let x_3 be the node of G_1 corresponding to e_3. Since x_3 must be dominated, we have by minimality of degree 2 nodes that $|D \cap \{v_1, v_2\}| > 0$. Suppose that $v_1 \in D$ and let v' be the other endpoint of e_1. Then $D \cup \{v'\} \setminus \{x_1\}$ is also a dominating set of G_1, contradicting minimality. If e_1 and e_2 are incident, say $e_1 = v_1v_2$ and $e_2 = v_2v_3$, then one can show that $v_3 \notin D$, and $(D \setminus x_3) + v_3$ is a minimum dominating set that violates the minimality of X-nodes. Thus if X is the set of degree 2 nodes of D, then $\sigma(G) \geq |X| = |V_G| - |D| = |V_G| - \gamma(G_1)$, completing the proof. □

We can also interchange the roles of the parameters. We leave the proof for the reader (a proof for bipartite graphs is given in [5]).

LEMMA 2.2. *For any graph G, $\sigma(G_1) + \gamma(G) = |V_G|$.*

We denote by $\gamma_k(G)$ (respectively, $\sigma_k(G)$) the value $\gamma(G_k)$ ($\sigma(G_k)$). We can now deduce that all the parameters are determined by the values on the two graphs G and G_1.

LEMMA 2.3. *For any graph G and $k \geq 0$, we have*

- $\gamma_{2k+1} + \sigma = |V| + s_k|E| = \sigma_{2k+1} + \gamma$,
- $\gamma_{2k} + \sigma_1 = |V| + t_k|E| = \sigma_{2k} + \gamma_1$,

where $s_0 := 0$, $t_0 := 0$, $t_k := \sum_{i=0}^{k-1} 2^{2i}$, and $s_k := 2^{2k} - t_k - 1$ for $k > 0$.

This also immediately implies the following.

COROLLARY 2.4. *For any graph G and $k \geq 0$,*

[1] $\sigma_{2k} - \sigma = t_k|E| = \gamma_{2k} - \gamma$,

[2] $\gamma_{2k+1} - \gamma_1 = s_k|E| = \sigma_{2k+1} - \sigma_1$.

The upshot of this is that the parameters $\{\gamma, \gamma_2, \gamma_4, \ldots\} \cup \{\sigma_1, \sigma_3, \ldots\}$ all share the same fate with respect to NP-completeness (and similarly for the class obtained by swapping parities). We thus obtain the following result.

THEOREM 2.5. *For each $k \geq 1$, both σ (induced matching) and γ (domination) are NP-hard to compute for the class of planar, bipartite graphs of maximum degree 4 with one side of the bipartition consisting only of degree 2 nodes and with each cycle having length $\cong 0$ mod 2^k.*

Proof. We have seen trivially that σ is NP-hard to compute for maximum degree 4 planar graphs. Thus γ_1 is also NP-hard for this class. However, hanging a path with three edges from each degree 2 node of G_1 results in a graph G' such that $\gamma(G') = \gamma_1(G) + |E|$. Thus γ is also NP-hard to compute for planar graphs of maximum degree 4. By Corollary 2.4, for each j, γ_j, σ_j are NP-hard to compute for this class of graphs, and the result follows. □

COROLLARY 2.6 (see Cameron [1]). *It is NP-hard to compute the induced matching number for the class of bipartite graphs.*

Results of Corneil and Perl [3] have a flavor similar to Theorem 2.5, but for the independent domination number. An *independent dominating set* is a dominating set of mutually nonadjacent nodes. Denote by $\iota(G)$ the cardinality of a smallest independent dominating set in G. It is an easy exercise to show that for each $k > 0$, $\gamma(G_k) = \iota(G_k)$, and so we have the NP-completeness of independent domination for the class of planar bipartite graphs given in Theorem 2.5. A related result in [3] shows that it is NP-hard to compute the independent domination number for the class of bipartite graphs of maximum degree 3 with one side of the bipartition consisting only of degree 2 nodes.

We close our discussion of induced matchings by mentioning that we know of no class of graphs for which exactly one of γ, σ is polynomially computable.

We now return to the proof of Theorem 1.2.

DEFINITION 2.7. *For an arbitrary planar bipartite graph $B = (V_1 \cup V_2, E_B)$ we construct a network pair $(D_B := (V_1 \cup V_2 \cup \{v^*\}, E(D_B)), T_B)$, where $E(T_B) := \{(x, v^*) : x \in V_1\} \cup \{(v^*, x) : x \in V_2\}$ and $E(D_B) := E(T_B) \cup \{(x, y) \in E_B : x \in V_1, y \in V_2\}$. Note that $D_B - \{v^*\}$ is simply the planar graph B with arcs oriented from*

V_1 to V_2 and that each $x \neq v^*$ is a leaf of T_B.

LEMMA 2.8. *The minimum cardinality of an* SMC *for (D_B, T_B) is $|V(B)| - \sigma(B)$.*

Proof. Suppose that C is a minimum cardinality SMC for (D_B, T_B) whose subset C' of tree arcs is maximized. Let $C' = L_1 \cup L_2$, where $L_1 = \{(x_1, v^*), (x_2, v^*), \ldots, (x_k, v^*)\}$ and $L_2 = \{(v^*, y_1), \ldots, (v^*, y_l)\}$, and set $R_1 = V_1 - \{x_i\}_{i=1}^k$, $R_2 = V_2 - \{y_i\}_{i=1}^l$. Note that for each arc (x, y) with $x \in R_1, y \in R_2$, the edges of the (undirected) fundamental circuit $(x, v^*), (x, y), (v^*, y)$ are not covered by a tree edge of C, and hence $(x, y) \in C$. We claim first that $C = C' \cup C''$, where $C'' = \{(x, y) \in E(D_B) : x \in R_1, y \in R_2\}$. If, say, $(x_i, y) \in C$ for some $y \in R_2$, then this arc is needed only to cover the fundamental cocircuit associated with y, and so we may replace it by (v^*, y) to obtain another minimum size SMC, contradicting the maximality of C'. We next claim that C'' is an induced matching. If there exists $(x, y), (w, y) \in C''$ say, then we could replace the edge (w, y) by (w, v^*), again contradicting maximality of C'. Thus $|C| = |V(B)| - |I|$, where I is the associated induced matching on B. Moreover, given any induced matching I of B, there is clearly an SMC for (D_B, T_B) of size $|V(B)| - |I|$. The result now follows. □

This together with Theorem 2.5 now implies the following strong version of Theorem 1.2.

THEOREM 2.9. *For each $k \geq 1$,* SMC *is NP-hard even when restricted to network pairs (D, T), where T is a star with center node v^* and $D - v^*$ is a planar bipartite graph $B = (V_1 \cup V_2, E_B)$ of maximum degree 4 such that V_2 contains only degree 2 nodes and each cycle of B has length congruent to 0 (mod 2^k).*

3. A polynomial-time algorithm for single-source compatible network matrices.

In this section we give a proof of Theorem 1.1. The presentation will be in terms of an algorithm to solve SMC for graphic matroids with a basis consisting of an *arborescence*, i.e., an oriented tree T with a single source r. This is essentially a dynamic programming algorithm. We adopt, however, a simple, recursive description devised by Gerards.

More specifically, we assume that we are given a compatible pair (D, T), where D has no loops except possibly a collection $\lambda(r)$ at the root r of the arborescence T, along with a weight function w on the edges of D. Let $\delta(r)$ denote the nonloop edges incident with r. For $e \in T$, $C_T(e)$ denotes the fundamental cut containing e with respect to T; for $e \notin T$, $C_T(e)$ denotes the fundamental circuit containing e with respect to T. If e is a loop, then $C_T(e) := \{e\}$.

We are also given two sets $X \subseteq (\delta(r) \cup \lambda(r)) \setminus E(T)$ and $Y \subseteq \delta(r) \cup \lambda(r)$. As is standard, we say a set Z *covers* a set Y if $Z \cap Y \neq \emptyset$. We then solve for

$$\begin{aligned} w(D, T, X, Y) := {} & \text{the minimum weight of a subset of } E(D) \text{ that covers the set } Y \\ & \text{and all the sets } C_T(e) \text{ with } e \in E(D) \setminus X. \end{aligned}$$

Note that finding $w(D, T, \emptyset, \delta(r))$ with $\lambda(r) = \emptyset$ solves SMC for (D, T). The algorithm to solve for $w(D, T, X, Y)$ has three recursive steps: *loop cleaning*, *root contraction*, and *root splitting.* These are used to reduce the problem to a single node r incident to some loop edges $\lambda(r)$. If (D, T, X, Y) is such a base instance, then a minimum cover C is obtained easily. First, set $C := \lambda(r) \setminus X$. If C covers Y, then C is an optimal set. Otherwise, add the minimum weight member of Y to C.

We now define three operations that may be used to reduce any instance to one of these base cases. We make repeated use of the fact that there is always an edge $e \in E(T) \cap (\delta(r) \setminus X)$. Thus any feasible solution must cover $C_T(e)$, and hence $\delta(r)$

is always covered by any such solution. In the following, we define $w(D, T, X, \emptyset) = \infty$ for any D, T, X.

Loop cleaning. This operation reduces the problem to one without loops.

First, suppose that $Y \cap (\lambda(r) \setminus X) \neq \emptyset$. Then $w(D, T, X, Y) = w(D \setminus \lambda(r), T, X \cap \delta(r), \delta(r)) + w(\lambda(r) \setminus X)$ holds as well. We repeatedly apply the two ideas used in this case. First, by the definition of $w(D, T, X, Y)$, $\lambda(r) \setminus X$ must be in any feasible cover. Clearly then the $\leq$-inequality holds. Second, there is an edge $e \in E(T) \cap (\delta(r) \setminus X)$, so any solution for $w(D, T, X, Y)$ must cover $C_T(e)$ and thus $\delta(r)$. Thus the $\geq$-inequality holds as well.

Next, suppose that $Y \cap (\lambda(r) \setminus X) = \emptyset$. Then $w(D, T, X, Y)$ is the minimum of the following expressions:

- $w(D \setminus \lambda(r), T, X \cap \delta(r), \delta(r)) + w(\lambda(r) \setminus X) + w_{\min}(Y \cap \lambda(r))$,
- $w(D \setminus \lambda(r), T, X \cap \delta(r), Y \cap \delta(r)) + w(\lambda(r) \setminus X)$.

Here we define $w_{\min}(A)$ as the minimum of $w()$ over A, with $w_{\min}(\emptyset) := \infty$.

The first case corresponds to covering Y by a loop edge, whereas in the second case it is covered by an edge in $Y \cap \delta(r)$.

Root contracting. If D has no loops and r has exactly one out-neighbor r' in T, then the following recursive formula transforms the problem into two problems on the digraph D/rr', the digraph obtained by contracting rr' and then deleting the loop. In this case, $w(D, T, X, Y)$ is the minimum of

- $w(D/rr', T/rr', \delta(r) \cap \delta_{D/rr'}(r'), Y') + w_{rr'}$, where $Y' := \delta_{D/rr'}(r')$ if $rr' \in Y$, and $Y' := Y$ if $rr' \notin Y$.
- $w(D/rr', T/rr', X, Y \cap \delta_{D/rr'}(r'))$.

The first case corresponds to those covers that include the edge rr', and hence we no longer need to cover any of the fundamental circuits for arcs in $\delta(r)$. Thus X becomes $\delta(r) \cap \delta_{D/rr'}(r')$. The second case corresponds to covers that do not include rr'.

Root splitting. If D has no loops and r has two or more out-neighbors $r_1, \ldots, r_k$ in T, then clearly D has k edge-disjoint components that are connected at r only. The problem can be essentially restricted to these components separately as long as we ensure that at least one of the subproblems also covers Y. More rigorously, $w(D, T, X, Y)$ is defined as the minimum of the following k values:

$$\begin{aligned} &w(D_i, T_i, X \cap \delta_{D_i}(r), Y \cap \delta_{D_i}(r)) \\ &\qquad + \sum_{j=1, j\neq i}^{k} w(D_j, T_j, X \cap \delta_{D_j}(r), \delta_{D_j}(r)) \quad \text{for each } i = 1, 2, \ldots, k. \end{aligned}$$

Here, T_i consists of rr_i together with the maximal rooted subtree of T rooted at r_i, and D_i is the subgraph of D induced by $V(T_i)$. The interpretation is that the ith of these values represents the minimum weight of a covering where the ith component is responsible for covering Y.

THEOREM 3.1. *There is a polynomial-time algorithm that solves the* SMC *problem for compatible pairs (D, T), where T is an arborescence.*

Proof. It is routine to check that the recursion defined above yields the minimum weight cover. To show there is a polynomial-time algorithm, we first show that the recursion generates a polynomially bounded number of subproblems $w(D', T', X', Y')$. Note that for each subproblem, D' and T' are obtained from D and T by contracting the rs-path P (where s is the root of T') in T and then deleting some edges. One also sees (by checking each reduction) that every such generated Y' is the intersection of some collection of fundamental cuts of the original directed graph D with respect to T. In addition, each of these cuts is induced by an arc in the path P associated with the subproblem. Similarly, each X' is also the intersection of such a set with the nontree edges. Note that the intersection of some fundamental cuts induced by arcs $e_1, e_2, \ldots, e_l$ say, on a dipath P in T, is the same as intersecting the fundamental cuts for e_1, e_l alone (assuming P traverses the e_i's in the order of their subscripts). Thus the algorithm generates at most a quadratic number of such sets, and hence at most a polynomial number $(O(n^4))$ of subproblems is generated.

Each of these subproblems could be needed more than once, of course. Still, dynamic programming can be performed in polynomial time as follows. Note that there is a natural acyclic digraph H whose nodes are the encountered subproblems and whose arcs are determined by whether one instance "spawns" another. Each such arc has an associated weight set by our three operations. The sinks in this digraph are just the base cases, i.e., single nodes with some loops. The "in-degree" of a node in this digraph is just the number of other subproblems which spawn it. Once H is constructed, we can compute the values one by one. Simply choose a sink v in H and compute its value (since all of its original out-neighbors have had their values computed). Then delete v from H and repeat. □

4. Conclusions.

We do not know the complexity status of the dominating set problem for bipartite graphs whose adjacency matrices (1) are themselves network matrices. Thus both B, B^T are network matrices and so a result of Whitney states that these arise from compatible pairs, where $D \cup T$ is a planar graph. Finally, we ask whether one may find a compact (extended) formulation for dominating set polyhedra for convex bipartite graphs.

Acknowledgments. The authors thank Kathy Cameron for stimulating comments. They are also grateful to A.M.H. Gerards for many helpful improvements and especially for sharing his ingenious recursion.

REFERENCES

[1] K. Cameron, *Induced matchings*, Discrete Appl. Math., 24 (1989), pp. 97–102.

[2] P. Damaschke, H. Müller, and D. Kratsch, *Domination in convex and chordal bipartite graphs*, Inform. Process. Lett., 36 (1990), pp. 231–236.

[3] D.G. Corneil and Y. Perl, *Clustering and domination in perfect graphs*, Discrete Math., 9 (1984), pp. 27–39.

[4] J. Fonlupt and M. Raco, *Orientation of matrices*, Math. Programming Stud., 22 (1984), pp. 86–98.

[5] B. Gamble, W. Pulleyblank, B. Reed, and B. Shepherd, *Right angle free subsets in the plane*, Graphs Combin., 11 (1995), pp. 121–129.

[6] R. Raz and S. Safra, *A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP*, in Proceedings of the 29th Annual ACM Symposium on Theory of Computing, ACM, New York, 1997, pp. 475–484.

[7] A. Schrijver, *The Theory of Linear and Integer Programming*, Wiley, New York, 1986.

[8] W.T. Tutte, *Lectures on matroids*, J. Res. Nat. Bur. Standards Sect. B, 69B (1965), pp. 1–47.

SIAM J. DISCRETE MATH.
Vol. 16, No. 4, pp. 524–532

EQUITABLE COLORING OF k-UNIFORM HYPERGRAPHS[*]

RAPHAEL YUSTER[†]

Abstract. Let H be a k-uniform hypergraph with n vertices. A *strong r-coloring* is a partition of the vertices into r parts such that each edge of H intersects each part. A strong r-coloring is called *equitable* if the size of each part is $\lceil n/r \rceil$ or $\lfloor n/r \rfloor$. We prove that for all $a \geq 1$, if the maximum degree of H satisfies $\Delta(H) \leq k^a$, then H has an equitable coloring with $\frac{k}{a \ln k}(1 - o_k(1))$ parts. In particular, every k-uniform hypergraph with maximum degree $O(k)$ has an equitable coloring with $\frac{k}{\ln k}(1 - o_k(1))$ parts. The result is asymptotically tight. The proof uses a double application of the nonsymmetric version of the Lovász local lemma.

Key words. hypergraph, coloring

AMS subject classification. 05C15

DOI. 10.1137/S089548010240276769

1. Introduction. Let H be a k-uniform hypergraph with n vertices. (All hypergraphs considered here are finite. For standard terminology the reader is referred to [5].) A *strong r-coloring* is a partition of the vertices of H into r parts such that each edge of H intersects each part. (A *weak* r-coloring is a coloring where no edge is monochromatic.) A strong r-coloring is called *equitable* if the size of each part is $\lceil n/r \rceil$ or $\lfloor n/r \rfloor$. The study of equitable colorings is motivated by scheduling applications in which some tasks are required to perform at the same time. A good survey on equitable colorings is given in [8]. See also [4, 7] for other related results in the graph-theoretic case. Let $c(H)$ denote the maximum possible number of parts in a strong coloring of H. Let $ec(H)$ denote the maximum possible number of parts in an equitable coloring of H. Trivially, $1 \leq ec(H) \leq c(H) \leq k$. In general, k could be large and still $ec(H) = c(H) = 1$ if we do not impose upper bounds on the maximum degree. Consider the complete k-uniform hypergraph on $2k$ vertices. Trivially, it has $c(H) = 1$, and the maximum degree is less than 4^k. In this paper we prove that $c(H)$ and $ec(H)$ are quite large if the maximum degree is bounded by a polynomial in k. In fact, we get the following asymptotically tight result.

THEOREM 1.1. *If $a \geq 1$, and H is a k-uniform hypergraph with maximum degree at most k^a, then $ec(H) \geq \frac{k}{a \ln k}(1 - o_k(1))$. The lower bound is asymptotically tight. For all $a \geq 1$, there exist k-uniform hypergraphs H with maximum degree at most k^a and $c(H) \leq \frac{k}{a \ln k}(1 + o_k(1))$.*

The tightness is shown by exhibiting a random hypergraph with appropriate parameters. Alon [1] has shown that there exist k-uniform hypergraphs with n vertices and maximum degree at most k that do not have a vertex cover (transversal) of size less than $(n \ln k/k)(1-o_k(1))$. In particular, no strong coloring (moreover an equitable one) could have more than $(k/\ln k)(1 + o_k(1))$ parts. For completeness, in section 3 we give a general argument valid for all $a \geq 1$. The proof of the main result appears in section 2. The final section contains some concluding remarks.

[*] Received by the editors February 20, 2002; accepted for publication (in revised form) February 6, 2003; published electronically July 30, 2003.
http://www.siam.org/journals/sidma/16-4/40276.html

[†] Department of Mathematics, University of Haifa at Oranim, Tivon 36006, Israel (raphy@research.haifa.ac.il).

2. Proof of the main result. In the proof of Theorem 1.1 we need to use the Lovász local lemma [6] in its strongest form, known as the *nonsymmetric version.* We state it here, following the notation in [2] (which also contains a simple proof of the lemma). Let $A_1, \dots, A_n$ be events in an arbitrary probability space. A directed graph $D = (V, E)$ on the set of vertices $V = [n]$ is called a *dependency digraph* for the events $A_1, \dots, A_n$ if for each i, $i = 1, \dots, n$, the event A_i is mutually independent of all the events $\{A_j \; : \; (i, j) \notin E\}$.

LEMMA 2.1 (the local lemma, nonsymmetric version). *If $x_1, \dots, x_n$ are real numbers so that $0 \le x_i < 1$ and $\Pr[A_i] \le x_i \prod_{(i,j)\in E}(1 - x_j)$ for all $i = 1, \dots, n$, then with positive probability no event A_i occurs.* □

If the maximum outdegree in D is at most $d \ge 1$ and each A_i has $\Pr[A_i] \le p$, then by assigning $x_i = 1/(d + 1)$ we immediately obtain the following.

COROLLARY 2.2 (the local lemma, symmetric version). *If $p(d + 1) \le 1/e$, then with positive probability no event A_i occurs.* □

Proof of Theorem 1.1. Let $a \ge 1$ be any real number, and let $\epsilon > 0$ be small. Throughout the proof we assume k is sufficiently large as a function of a and ϵ. Let k be so large that there is an integer between $\frac{k}{(1+\epsilon^2/4)a \ln k}$ and $\frac{k}{(1+\epsilon^2/8)a \ln k}$. Thus, for some $\gamma \in [\epsilon^2/8, \; \epsilon^2/4]$, the number $t = \frac{k}{(1+\gamma)a \ln k}$ is an integer. Now, let $H = (V, E)$ be a hypergraph with n vertices and $\Delta(H) \le k^a$. We will show that there exists an equitable coloring of H with $\frac{k}{(1+\gamma)a \ln k} - \lceil \sqrt{\gamma} \frac{k}{a \ln k} \rceil > (1 - \epsilon) \frac{k}{a \ln k}$ colors.

Assume that we have the set of colors $\{1, \dots, t\}$. It will be convenient to deal with the finite set of hypergraphs having $n < 2k \ln k$ separately. We begin with the general case.

2.1. The general case $n > 2k \ln k$.

In the first phase of the proof we color most of the vertices (that is, we obtain a partial coloring) such that certain specific properties hold. In the second phase we color the vertices that were not colored in the first phase and show that we can do it carefully enough to obtain a strong t-coloring. In the third phase we show how to modify our coloring to obtain an equitable coloring.

2.1.1. First phase.

Our goal in this phase is to achieve a partial coloring with several essential properties shown below.

LEMMA 2.3. *There exists a partial coloring of H with the colors $\{1, \dots, t\}$ such that the following four conditions hold:*

1. *Every edge contains at least $k\gamma/5$ uncolored vertices.*
2. *Every edge has at most $\lceil 10/\gamma \rceil$ colors that do not appear on its vertex set.*
3. *Put $z = \lceil k^{1-a\gamma/4} \rceil$. For each $v \in V$, for each sequence of z* **distinct** *colors $c_1, \dots, c_z$, and for each sequence of z* **distinct** *edges containing v denoted $f_1, \dots, f_z$, at least one f_i has an element colored c_i.*
4. *Every color appears on at least $n\frac{(1+\gamma/4)a \ln k}{k}$ vertices.*

Proof. We let each vertex $v \in V$ choose a color from $\{1, \dots, t\}$ randomly. The probability of choosing color i is $p = \frac{(1+\gamma/2)a \ln k}{k}$ for $i = 1, \dots, t$ and the probability of it remaining uncolored is, therefore, $q = 1 - pt = \frac{\gamma}{2(1+\gamma)}$. For an edge f, let A_f denote the event that f contains less than $k\gamma/5$ uncolored vertices. Let B_f denote the event that f has more than $\lceil 10/\gamma \rceil$ colors missing from its vertex set. For a vertex v, let C_v denote the event that there exist z distinct edges $f_1, \dots, f_z$ where each f_i contains v, and there exist z distinct colors $c_1, \dots, c_z$ such that c_i is missing from f_i for each $i = 1, \dots, z$. For a color c, let D_c denote the event that the color c appears on less than $n\frac{(1+\gamma/4)a \ln k}{k}$ vertices. We must show that with positive probability, none of the $2|E| + |V| + t$ events above hold. The following four claims provide upper bounds for the probabilities of the events A_f, B_f, C_v, D_c.

CLAIM 2.4. $\Pr[A_f] < \frac{1}{k^{5a}}$.

Proof. Let X_f denote the random variable counting the uncolored elements of f. The expectation of X_f is $E[X_f] = kq$. Since each vertex chooses its color independently we have by a common Chernoff inequality (cf. [2])

$$\Pr\left[A_f\right] = \Pr\left[X_f < \frac{k\gamma}{5}\right] \le \Pr\left[X_f < \frac{kq}{2}\right] = \Pr\left[X_f < \frac{E[X_f]}{2}\right]$$

$$< e^{-2(E[X_f]/2)^2/k} = e^{-k^2q^2/(2k)} = e^{-kq^2/2} << \frac{1}{k^{5a}}. \qquad \square$$

CLAIM 2.5. $\Pr[B_f] < \frac{1}{k^{5a}}$.

Proof. Fix $s = \lceil 10/\gamma \rceil$ distinct colors. The probability that none of them appear on f is precisely $(1-sp)^k$. Now,

$$(1-sp)^k = \left(1 - \frac{s(1+\frac{\gamma}{2})a\ln k}{k}\right)^k < \frac{1}{k^{as+as\gamma/2}}.$$

As there are $\binom{t}{s} < k^s$ possible sets of s distinct colors, we get that

$$\Pr[B_f] < \binom{t}{s}\frac{1}{k^{as+as\gamma/2}} < \frac{1}{k^{as\gamma/2}} \le \frac{1}{k^{5a}}. \qquad \square$$

CLAIM 2.6. $\Pr[C_v] < \frac{1}{k^{5a}}$.

Proof. If the degree of v is less than z, there is nothing to prove. Otherwise, fix a set of z distinct colors $\{c_1, \dots, c_z\}$ and z distinct edges containing v, denoted $\{f_1, \dots, f_z\}$. We begin by computing the probability that for each $i = 1, \dots, z$, c_i does not appear on an element of f_i. Denote this probability by $\rho = \rho(v, f_1, \dots, f_z, c_1, \dots, c_z)$. For every vertex u let d_u be the number of edges f_i, $1 \le i \le z$, which contain u. By the definition of the event C_v we know that if C_v holds, then there is a set of d_u colors, none of which was assigned to u. The probability of this is $1 - d_u p$. Thus

$$\rho = \prod_u (1 - d_u p) \le e^{-p\Sigma_u d_u} = e^{-p\Sigma_i |f_i|} = e^{-pkz} = \frac{1}{k^{a(1+\gamma/2)z}}.$$

There are exactly $(t)_z < (k/\ln k)^z$ ordered sets of z distinct colors. Thus, the probability that each edge of $f_1, \dots, f_z$ misses a distinct color is less than $(k/\ln k)^z / k^{a(1+\gamma/2)z}$. There are at most $\binom{\lfloor k^a \rfloor}{z}$ distinct subsets of z edges containing v. This, together with Stirling's formula, gives

$$\begin{aligned}\Pr[C_v] &< \binom{\lfloor k^a \rfloor}{z}\frac{k^z}{(\ln k)^z\, k^{a(1+\gamma/2)z}} < \left(\frac{ek^a}{z}\frac{k}{k^{a(1+\gamma/2)}\ln k}\right)^z \\ &\le \left(\frac{e}{k^{a\gamma/4}\ln k}\right)^z << \frac{1}{k^{5a}}. \qquad \square\end{aligned}$$

Claim 2.7. $\Pr[D_c] < \frac{1}{e^{n/k}}$.

Proof. Let X_c denote the number of vertices which received the color c. Clearly, $E[X_c] = pn = n\frac{(1+\gamma/2)a\ln k}{k}$. Put $\beta = n\frac{a\gamma\ln k}{4k}$. We shall use the Chernoff inequality (cf. [2])

$$\Pr[X_c - pn < -\beta] < e^{-\beta^2/(2pn)}.$$

In our case

$$\Pr[D_c] = \Pr[X_c - pn < -\beta] < e^{-\beta^2/(2pn)} = e^{-\frac{na\ln k}{k}(\frac{\gamma^2}{32(1+\gamma/2)})}$$

$$< e^{-\frac{na\ln k}{k}(\frac{\gamma^2}{33})} = \frac{1}{k^{(an/k)(\gamma^2/33)}} \le \frac{1}{k^{(n/k)(\gamma^2/33)}} < \frac{1}{e^{n/k}}. \qquad \Box$$

We now construct a dependency digraph for all the events of the forms A_f, B_f, C_v, D_c (we refer to the events as type A, type B, type C, and type D, respectively). Consider an event A_f. Let $E(f)$ denote the set of edges of H which are disjoint from f. Let $V(f)$ denote the set of vertices of H which do not appear on any edge that intersects f. Clearly A_f is mutually independent of all the $2|E(f)| + |V(f)|$ events of the form A_g, B_g, or C_v which correspond to the elements of $E(f)$ and $V(f)$. Since there are at most k^{a+1} edges intersecting f and since there are at most k^{a+2} vertices in these edges, the outdegree in the dependency graph from A_f to other events of type A is at most k^{a+1}. Similarly, the outdegree in the dependency graph from A_f to other events of type B is at most k^{a+1}, and to events of type C it is at most k^{a+2}. Since A_f depends on all events of type D, we have that the outdegree is t. This explains the first line of Table 1 (the dependency table). The other elements in the table are determined similarly. Note that events of type D depend on all other events (the fourth line in Table 1).

Table 1
The maximum possible outdegrees in the dependency digraph.

Source\target	A_f	B_f	C_v	D_t
A_f	k^{a+1}	k^{a+1}	k^{a+2}	t
B_f	k^{a+1}	k^{a+1}	k^{a+2}	t
C_v	k^{2a+1}	k^{2a+1}	k^{2a+2}	t
D_t	$\lvert E\rvert$	$\lvert E\rvert$	n	t

In order to apply Lemma 2.1 we need to assign a coefficient to each event in the dependency digraph (the coefficients correspond to the x_i in Lemma 2.1). To each event of type A, B, or C we assign the coefficient $3/k^{5a}$. To each event of type D we assign the coefficient $1/e^{n/2k}$. It remains to show that the conditions in Lemma 2.1 hold for each event. For events of type A we must show that

$$(1) \qquad \Pr[A_f] < \frac{3}{k^{5a}}\left(1-\frac{3}{k^{5a}}\right)^{k^{a+1}}\left(1-\frac{3}{k^{5a}}\right)^{k^{a+1}}\left(1-\frac{3}{k^{5a}}\right)^{k^{a+2}}\left(1-\frac{1}{e^{n/2k}}\right)^{t}.$$

Indeed, recall that $n > 2k\ln k$ so $(1-1/e^{n/2k})^{k-1} > e^{-1}$. Using Claim 2.4 and the relation $t < k-1$, we find that the right side of (1) exceeds

$$\frac{3}{k^{5a}}\left(1-\frac{3}{k^{5a}}\right)^{3k^{a+2}} e^{-1} > \frac{3}{k^{5a}} \cdot 0.99 \cdot e^{-1} > \frac{1}{k^{5a}} > \Pr[A_f].$$

The analogous inequalities for events of types B and C follow similarly from Claims 2.5 and 2.6, respectively. Finally, consider events of type D. We must show that

$$\Pr[D_c] < \frac{1}{e^{n/2k}}\left(1-\frac{3}{k^{5a}}\right)^{2|E|+n}\left(1-\frac{1}{e^{n/2k}}\right)^{t}. \tag{2}$$

In any k-uniform hypergraph we have $|E| \le n\Delta/k$. Thus, in our case, $2|E|+n \le 3k^{a-1}n$. Using Claim 2.7 and again the relation $(1-1/e^{n/2k})^{k-1} > e^{-1}$, we find that the right side of (2) exceeds

$$\frac{1}{e^{n/2k}}\left(1-\frac{3}{k^{5a}}\right)^{3k^{a-1}n} e^{-1} > \frac{1}{e^{n/2k}}\left(1-\frac{3}{k^{5a}}\right)^{(\frac{k^{5a}}{3}-1)\frac{18n}{k^{4a+1}}} e^{-1}$$

$$> \frac{1}{e^{n/2k}}e^{-\frac{18n}{k^{4a+1}}-1} > \frac{1}{e^{n/2k}}\frac{1}{e^{n/2k}} = \frac{1}{e^{n/k}} > \Pr[D_c].$$

According to Lemma 2.1, with positive probability, none of the events in the dependency digraph hold. We have completed the proof of Lemma 2.3.

2.1.2. Second phase. Fix a partial coloring satisfying the four conditions in Lemma 2.3. For an edge f, let $M(f)$ denote the set of missing colors from f. By Lemma 2.3 we know that $|M(f)| \le \lceil 10/\gamma \rceil$. For a vertex v, let $S(v) = \cup_{v\in f} M(f)$. We claim that $|S(v)| \le \lceil 10/\gamma\rceil(z-1) \le 11z/\gamma$. To see this, notice that if $|S(v)| > \lceil 10/\gamma\rceil(z-1)$, then there must be at least z distinct edges containing v, say, $f_1,\ldots,f_z$, and z distinct colors $c_1,\ldots,c_z$ such that c_i does not appear on f_i for $i=1,\ldots,z$. However, this is impossible by the third requirement in Lemma 2.3. In the second phase we only color the vertices that are uncolored after the first phase. Let v be such a vertex. We let v choose a random color from $S(v)$ with uniform distribution. The choices made by distinct vertices are independent. (In case $S(v)=\emptyset$ we can assign an arbitrary color to v.) Let $f \in E$ be any edge, and let $c \in M(f)$. Let $A_{f,c}$ denote the event that, after the second phase, c still does not appear as a color on a vertex of f. Our goal is to show that, with positive probability, none of the events $A_{f,c}$ for $f \in E$ and $c \in M(f)$ hold. This will give a strong t-coloring of H (although not necessarily an equitable one).

Let $T(f)$ be the subset of vertices of f which are uncolored after the first phase. By Lemma 2.3 we have $|T(f)| \ge k\gamma/5$. If $c \in M(f)$, we have that for each $u \in T(f)$ the color c appears on $S(u)$. Hence,

$$\Pr[A_{f,c}] = \Pi_{u\in T(f)}\left(1-\frac{1}{|S(u)|}\right) \le \Pi_{u\in T(f)}\left(1-\frac{\gamma}{11z}\right)$$

$$\le \left(1-\frac{\gamma}{11z}\right)^{k\gamma/5} < e^{-\frac{k\gamma^2}{55z}} < e^{-k^{a\gamma/4}\frac{\gamma^2}{110}} << \frac{1}{k^{a+2}}.$$

Since each event $A_{f,c}$ is mutually independent of all other events except those that correspond to edges that intersect f, we have that the dependency digraph of the events has maximum outdegree at most $\lceil 10/\gamma\rceil k^{a+1} < k^{a+2}/e-1$. Since $\frac{1}{k^{a+2}}((k^{a+2}/e-1)+1) = 1/e$ we have, by Corollary 2.2, that with positive probability none of the events of the form $A_{f,c}$ hold. In particular, there exists a strong t-coloring of H.

2.1.3. Third phase. Assume the color classes of the strong t-coloring obtained after the second phase are $V_1, \dots, V_t$, where $|V_i| \ge |V_{i+1}|$, $i = 1, \dots, t-1$. By Lemma 2.3 we know that $|V_i| \ge n\frac{(1+\gamma/4)a\ln k}{k}$, $i = 1, \dots, t$. Let $s = \lceil \sqrt{\gamma} k/(a \ln k)\rceil$ and let $W = V_1 \cup \cdots \cup V_s$. Clearly

$$n-|W| = |V \backslash W| = |V_{s+1} \cup \cdots \cup V_t| \ge (t-s)n\frac{(1+\frac{\gamma}{4})a\ln k}{k} = n\left(\frac{1+\frac{\gamma}{4}}{1+\gamma}\right) - \frac{sn(1+\frac{\gamma}{4})a\ln k}{k}.$$

Hence,

$$|W| \le n\left(1-\frac{1+\frac{\gamma}{4}}{1+\gamma}\right) + \frac{sn(1+\frac{\gamma}{4})a\ln k}{k} < \gamma n + \frac{sn(1+\frac{\gamma}{4})a\ln k}{k}.$$

In particular, $|V_s| \le |W|/s < \gamma n/s + n(1+\gamma/4)a\ln k/k$. It follows that $||V_i| - |V_j|| < \gamma n/s$ for all $s+1 \le i < j \le t$. Hence, it suffices to show that $|W| \ge (t-s)\gamma n/s$ since we can then transfer all the vertices in the color classes $V_1, \dots, V_s$ to the color classes $V_{s+1}, \dots, V_t$ such that after the transfer, the $t-s$ remaining classes form an equitable partition (the strong coloring stays proper, of course). Indeed,

$$|W| > sn\frac{a\ln k}{k} = s^2 n\frac{a\ln k}{sk} \ge n\gamma\frac{k^2}{a^2(\ln k)^2}\frac{a\ln k}{sk} = n\gamma\frac{k}{sa\ln k} > n\frac{t\gamma}{s} > (t-s)\frac{n\gamma}{s}.$$

We have shown how to obtain an equitable coloring with $t-s = \frac{k}{(1+\gamma)a\ln k} - \lceil\sqrt{\gamma}\frac{k}{a\ln k}\rceil > (1-\epsilon)\frac{k}{a\ln k}$ colors.

2.2. The finite case $n < 2k\ln k$. As in the proof for the general case, let each vertex choose a color randomly and independently, with each color having probability p where $p = \frac{(1+\gamma/2)a\ln k}{k}$ for $i = 1, \dots, t$ and whose probability of remaining uncolored is $q = 1 - pt = \frac{\gamma}{2(1+\gamma)}$. As in the proof of Claim 2.4, the probability that an edge contains less than $k\gamma/5$ uncolored vertices is less than $1/k^{5a}$. There are $|E| \le nk^a/k \le 2k^a \ln k$ edges. Hence, the expected number of edges with less than $k\gamma/5$ edges is less than $1/k^3$. Thus, with probability at least $1 - 1/k^3$, all edges have at least $k\gamma/5$ uncolored vertices. As in the proof of Claim 2.7, the probability that a color appears on less than $na\ln k(1+\gamma/4)/k$ vertices is less than $\frac{1}{k^{(n/k)(\gamma^2/33)}}$. Unlike Claim 2.7, we cannot bound this number from above by $e^{-n/k}$; instead, since $n \ge k$ (otherwise there are no edges at all), we can bound it with $k^{-\gamma^2/33}$. Since there are $t < k$ colors, the expected number of colors which appear on less than $na\ln k(1+\gamma/4)/k$ vertices is less than $k^{1-\gamma^2/33}$. Thus, with probability at least $2/3$ there are less than $3k^{1-\gamma^2/33}$ such colors. Finally, let X count the number of pairs (e, c), where $e \in E$ and c is a color that is missing from e. Clearly,

$$E[X] = |E|t(1-p)^k < 2k^a \ln k \cdot k \cdot k^{-a(1+\gamma/2)} = 2k^{1-a\gamma/2}\ln k < 2k^{1-\gamma/4} < \frac{k\gamma}{15}.$$

Hence, with probability at least $2/3$, $X < k\gamma/5$.

We have proved that with probability at least $1 - 1/k^3 - 1/3 - 1/3 > 0$ all of the following occur simultaneously:

1. All edges have at least $k\gamma/5$ uncolored vertices.
2. At least $t - 3k^{1-\gamma^2/33}$ colors appear, each on at least $na\ln k(1+\gamma/4)/k$ vertices.
3. The number of pairs (e, c) of edges e and colors c such that c is missing from e is less than $k\gamma/5$.

Fix a partial coloring with all these properties. Trivially we can make it a strong coloring by assigning a color c that is missing from an edge e to one of the uncolored vertices of e, and we can do it greedily to all such (e, c) pairs. We therefore obtain a strong t-coloring of H, where, in addition, at least $t - 3k^{1-\gamma^2/33}$ colors appear, each on at least $na \ln k(1+\gamma/4)/k$ vertices. We can now use the same arguments as in the third phase of the general case and obtain an equitable coloring. The only difference is that instead of t we only use $t - r$ colors, where r is the number of color classes having less than $na \ln k(1+\gamma/4)/k$ vertices. Thus, $t - r \geq t - 3k^{1-\gamma^2/33} > t(1-\gamma/33)$, and it is easily seen that all computations in the third phase hold when replacing t with $t(1-\gamma/33)$. □

3. A random hypergraph construction.

3. A random hypergraph construction. Let $a \geq 1$ and let $\epsilon > 0$. Let $n = k^{2a}$. For simplicity we assume n is an integer in order to avoid floors and ceilings. We select k sufficiently large to justify this assumption and the assumptions that follow. Let $m = (1-\epsilon)k^{3a-1}$ (again, assume m is an integer). Consider the random k-uniform hypergraph on the vertex set $[n]$ with m randomly selected edges $f_1, \ldots, f_m$. Each edge f_i is chosen uniformly from all $\binom{n}{k}$ possible edges. The m choices are independent (thus, the same edge can be selected more than once). The expected degree of a vertex v (including multiplicities) is $mk/n = (1-\epsilon)k^a$. Notice that for k sufficiently large we have, using a Chernoff inequality, that the degree of v is greater than k^a with probability less than $1/(2k^{2a}) = 1/(2n)$. Hence, with probability greater than 0.5 the maximum degree is at most k^a. Put $t = (1-2\epsilon)na \ln k/k$. Again, we assume t is an integer. We show that with probability greater than 0.5, no t-subset of vertices is a vertex cover. This proves the existence of hypergraphs H with $\Delta(H) \leq k^a$ and $c(H) \leq (1+o_k(1))k/(a \ln k)$.

Fix $X \subset [n]$ with $|X| = t$. For each edge f_i we have, assuming k is sufficiently large,

$$\begin{aligned}\Pr[f_i \cap X = \emptyset] &= \frac{(n-t)(n-t-1)\cdots(n-t-k+1)}{n(n-1)\cdots(n-k+1)} > \left(1 - \frac{t}{n-k+1}\right)^k \\ &> \left(1 - \frac{t}{(1-\epsilon)n}\right)^k = \left(1 - \frac{(1-2\epsilon)a\ln k}{(1-\epsilon)k}\right)^k > \left(1 - \frac{(1-\epsilon)a \ln k}{k}\right)^k \\ &> \frac{1}{2}e^{-(1-\epsilon)a\ln k} = \frac{1}{2k^{a(1-\epsilon)}}.\end{aligned}$$

Since each edge is selected independently we have

$$\Pr[X \text{ is a vertex cover}] < \left(1 - \frac{1}{2k^{a(1-\epsilon)}}\right)^m.$$

There are $\binom{n}{t}$ possible choices for X. It suffices to show that

$$\binom{n}{t}\left(1 - \frac{1}{k^{2a(1-\epsilon)}}\right)^m < \frac{1}{2}.$$

Indeed, for k sufficiently large

$$\begin{aligned}
\binom{n}{t}\left(1-\frac{1}{2k^{a(1-\epsilon)}}\right)^{m} &< \left(\frac{en}{t}\right)^{t}\left(1-\frac{1}{2k^{a(1-\epsilon)}}\right)^{(1-\epsilon)k^{3a-1}}\\
&= \left(\frac{ek}{(1-2\epsilon)a\ln k}\right)^{(1-2\epsilon)k^{2a-1}\ln k}\left(1-\frac{1}{2k^{a(1-\epsilon)}}\right)^{(1-\epsilon)k^{3a-1}}\\
&= \left(\left(\frac{ek}{(1-2\epsilon)a\ln k}\right)^{(1-2\epsilon)\ln k}\left(1-\frac{1}{2k^{a(1-\epsilon)}}\right)^{(1-\epsilon)k^{a}}\right)^{k^{2a-1}}\\
&< \left(e^{\ln^2 k}e^{-k^{a\epsilon}(1-\epsilon)/2}\right)^{k^{2a-1}} << \frac{1}{2}. \qquad \square
\end{aligned}$$

4. Concluding remarks. In the proof of Theorem 1.1 we require that $\Delta(H) \le k^a$ for some fixed $a \ge 1$. It is possible (although the computations get somewhat more complicated) to prove Theorem 1.1 when a is not necessarily a constant but satisfies $a = a(k) = o(k/\ln k)$. In other words, $\Delta(H)$ is allowed to be any subexponential function of k.

The proof of Theorem 1.1 is not algorithmic. It is, however, possible to obtain a polynomial time (in the number of vertices of the hypergraph, and not in its uniformity) algorithm that yields an equitable partition with $(1-o_k(1))ck/(a\ln k)$ parts, where c is a fixed small constant (depending only on a). This can be done by using the method of Beck for the 2-coloring of hypergraphs [3] and generalizing it to more colors. We also need to take care that the coloring obtained is equitable (Beck's algorithm does not guarantee this). However, Beck's algorithm can be modified to guarantee that all colors use *roughly* the same number of colors, and then we can use the approach from the third phase of our proof to show that by sacrificing only a small fraction of the colors we can make the partition equitable using the remaining colors. Notice that the third phase can be easily implemented in polynomial time.

A special case of Theorem 1.1 yields an interesting result about graphs. Let G be a k-regular graph. If k is sufficiently large, then G has an equitable coloring with $(1-o_k(1))(k/\ln k)$ colors such that each color class is a total dominating set (a total dominating set D is a subset of the vertices that has the property that each vertex $v \in G$ has a neighbor in D). To see this, we can construct a hypergraph H from the graph G as follows. For each vertex $v \in G$, let $N(v)$ denote the neighborhood of v. The vertices of H are those of G and the edges are all the sets $N(v)$. Note that H is k-uniform and $\Delta(H) = k$. Theorem 1.1 applied to H gives the desired result about G.

Acknowledgment. I wish to thank the referees for their useful suggestions.

REFERENCES

[1] N. Alon, *Transversal numbers of uniform hypergraphs*, Graphs Combin., 6 (1990), pp. 1–4.

[2] N. Alon and J. H. Spencer, *The Probabilistic Method*, John Wiley and Sons Inc., New York, 1991.

[3] J. Beck, *An algorithmic approach to the Lovász local lemma*, Random Structures Algorithms, 2 (1991), pp. 343–365.

[4] B. Bollobás and R. K. Guy, *Equitable and proportional coloring of trees*, J. Combin. Theory Ser. B, 34 (1983), pp. 177–186.

[5] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, Macmillan Press, London, 1976.

[6] P. Erdös and L. Lovász, *Problems and results on 3-chromatic hypergraphs and some related questions*, in Infinite and Finite Sets, A. Hajnal, R. Rado, and V. T. Sós, eds., North-Holland, Amsterdam, 1975, pp. 609–628.

[7] S. V. Pemmaraju, K. Nakprasit, and A. V. Kostochka, *Equitable colorings with constant number of colors*, in Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (Baltimore, MD, 2003), SIAM, Philadelphia, 2003, pp. 458–459.

[8] K.-W. Lih, *The equitable coloring of graphs*, in Handbook of Combinatorial Optimization, Vol. 3, Kluwer, Boston, MA, 1998, pp. 543–566.

SIAM J. DISCRETE MATH.
Vol. 16, No. 4, pp. 533–544

AMORTIZING RANDOMNESS IN PRIVATE MULTIPARTY COMPUTATIONS*

EYAL KUSHILEVITZ†, RAFAIL OSTROVSKY‡, AND ADI ROSÉN§

Abstract. We study the relationship between the number of rounds needed to repeatedly perform a private computation (i.e., where there are many sets of inputs sequentially given to the players on which the players must compute a function privately) and the overall randomness needed for this task. For the XOR function we show that, by re-using the same ℓ random bits, we can significantly speed up the round-complexity of each computation compared to what is achieved by the naive strategy of partitioning the ℓ random bits between the computations. Moreover, we prove that our protocols are optimal in the amount of randomness they require.

Key words. private distributed computations, randomness, round-complexity, amortization

AMS subject classifications. 94A60, 68Q99, 68R99

DOI. 10.1137/S089548010135274X

1. Introduction.

A very basic question in the theory of computation is the *direct-sum* question defined as follows: Can the complexity of solving k independent instances of a problem be smaller than the cost of independently solving the k instances? This general question was studied in various scenarios and with respect to various complexity measures, e.g., in [10, 20, 21, 23, 27, 40, 43]. To answer such a question, one typically needs to consider a problem whose complexity in the single-instance case is reasonably well understood.

In this work, we consider a direct-sum question related to the randomness complexity of private multiparty protocols. A 1-*private* (or simply, *private*) protocol $\mathcal{A}$ for computing a function f is a protocol that allows n players P_i, $P_1, \ldots, P_n$, each possessing an individual secret input x_i, to compute the value of $f(\vec{x})$ in a way that no *single* player learns about the initial inputs of other players more than what is revealed by the value of $f(\vec{x})$ and its own input.[1] We consider the setting in which the players have unlimited computational power; however, they do not deviate from their prescribed protocol (i.e., the information theoretical setting with honest-but-curious players, as opposed to byzantine parties). Private computations in this setting were the subject of a considerable amount of work; e.g., [5, 13, 2, 3, 15, 16, 17, 18, 21, 33, 37].[2] In

*Received by the editors January 15, 2001; accepted for publication (in revised form) September 18, 2002; published electronically July 30, 2003. A preliminary version of this paper appeared in *Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing*, ACM, New York, 1998, pp. 81–90.

http://www.siam.org/journals/sidma/16-4/35274.html

†Department of Computer Science, Technion, Haifa, Israel (eyalk@cs.technion.ac.il, http://www.cs.technion.ac.il/∼eyalk). Part of this author's research was done while visiting ICSI Berkeley. This author's research was supported by the MANLAM Fund.

‡Telcordia Technologies, MCC-1C357B, 445 South Street, Morristown, NJ 07960-6438 (rafail@research.telcordia.com, http://www.argreenhouse.com/bios/rafail/index.shtml).

§Department of Computer Science, Technion, Haifa 32000, Israel (adiro@cs.technion.ac.il). Part of this author's research was done while he was with the Department of Computer Science, University of Toronto, Toronto, Ontario, Canada; with the Department of Computer Science, Tel-Aviv University, Tel-Aviv, Israel; and visiting ICSI Berkeley.

[1] In the literature a more general definition of t-privacy is given. The above definition is the case $t = 1$.

[2] This *information theoretic privacy* setting is different from the *computational privacy* setting [46, 24] where players are limited to polynomial-time computations in their attempts to learn additional information.

this paper, we consider this setting for the basic XOR (exclusive-or) function and show direct-sum-type results relating the round-complexity and the randomness-complexity of such computations.

Randomness is an important resource in computation. As a result, various methods for saving in randomness were studied [1, 6, 14, 19, 26, 28, 29, 30, 38, 39, 42, 44, 45, 47]. In addition, the role of randomness in specific contexts was studied in, e.g., [41, 36, 4, 11, 8, 9]. One such case is the study of randomness in private multiparty computations [7, 12, 31, 34, 35, 22]. In particular, in [7, 35, 31] the amount of randomness required for private computations of XOR was considered (this function was the subject of previous research in the area of privacy due to its being a basic linear operation and its relative simplicity [21, 16]).

In this paper too we concentrate on the XOR function. We chose this function since its single-instance case is relatively well understood, and thus it enables us to derive direct-sum-type results. Indeed, the randomness-complexity of private computations of the XOR function on a single instance (that is, where each player has a single input bit) was previously investigated and the following results are known:

- There is no *deterministic* solution for the problem for $n \geq 3$;
- with a single random bit the problem requires $\Theta(n)$ rounds (time) [35];
- with $\ell \geq 2$ random bits the problem requires $\Theta(\frac{\log n}{\log \ell})$ rounds [35, 22].

None of the above mentioned works addressed multiple inputs.[3]

Our results. Before we make a precise statement of our general result, we start with a statement of a somewhat weaker version of our result which is simpler to state. Let us consider the case where n players are sequentially given n sets of inputs, of a single bit each, and for each such set the players wish to privately compute the XOR of these bits. In this case, using the results in [35], if the players use only a single independent random bit for each set of inputs, they can compute the XOR for each of the n sets of inputs in $n/2$ rounds for each set of inputs. It is impossible to compute the XOR's privately using the single random bit (per computation) with less than $\Omega(n)$ rounds per computation [35]. If the players wish to compute n independent XOR's with $O(1)$ rounds per computation, using independent random bits for each computation, they will need at least $\Omega(n^{1+\varepsilon})$ random bits overall (i.e., at least $\Omega(n^{\varepsilon})$ for each input set) [22]. In this paper we show how to re-use the *same* $O(n)$ random bits for all computations and achieve optimal rounds-performance each time; i.e., each computation will be performed in $O(1)$ rounds. Moreover, we accompany this result with a corresponding lower bound showing that in order to privately compute the n XOR's the players need $\Omega(n)$ random bits regardless of the number of rounds (for any $n \geq 3$). Thus, using the minimum number of random bits possible, we achieve by a recycling procedure the optimal round-complexity.

More generally, we consider the setting in which the players are sequentially given k input bits each, and the goal is to sequentially compute the XOR function k times immediately after the input bits for each set are provided. If one uses $\ell \geq k$ random bits, the naive solution would be to partition these bits into sets of $d \cong \ell/k$ random bits and to use the best single-input solution for d random bits to compute XOR for each sequential input. For example, if $\ell = k$, then $d = 1$ and this solution requires $\Theta(n)$ rounds and, if $\ell = \Theta(k)$, then $d = O(1)$ and this solution requires $\Theta(\log n)$ rounds [35, 22]. In this paper we present much better solutions than the above. In particular, we show that if $\ell = k = c \cdot (n-1)$, then the problem can be solved in $O(1)$ rounds

[3] Amortization of the *communication-complexity* in private computation of XOR was shown in [21].

per computation (rather than $\Theta(n)$), and for $\ell = O(k)$ we can solve the problem in $O(\frac{\log n}{\log k})$ rounds (rather than $\Theta(\log n)$). In addition, we prove that in order to privately compute the function XOR on k sequential inputs, at least $(1 - \frac{2}{n})k$ random bits are required. That is, for any fixed number $n \geq 3$ of players, $\Omega(k)$ random bits are required. This generalizes the claim that for the single-input case there is no deterministic private protocol if $n \geq 3$.

Our techniques. For the upper bounds, we present protocols that re-use the random bits and obtain the desired round-complexity. The lower bounds require more technical work. We first use techniques from [35] in order to reduce the problem of proving lower bounds on the amount of randomness used in such private computations to a problem of proving a lower bound on the number of *views* that the players may have in a *deterministic* protocol for computing the same function. Then, the main technical part is to give a lower bound on the number of such views. We note that the measure of the number of *views* is different from the measure of the number of *histories* (or projected histories), which is usually used in the study of (multiparty) communication complexity (cf. [32]) but is insufficient for our purposes.[4]

Organization. In section 2 we provide the required definitions, including the model and the definition of privacy. In section 3 we present our protocols that show our technique for recycling the random bits. Section 4 includes the lower bound.

2. Preliminaries.

In this section we give a description of the protocols we consider and define the *privacy* property of protocols as well as the required complexity measures.

A set of n players P_i $(1 \leq i \leq n)$, each receiving sequentially k input bits $x_i^1, \ldots, x_i^k$ (known *only* to that player), collaborate in a protocol to compute the k values

$$\begin{gathered} \text{XOR}(x_1^1, x_2^1, \ldots, x_n^1), \\ \text{XOR}(x_1^2, x_2^2, \ldots, x_n^2), \\ \vdots \\ \text{XOR}(x_1^k, x_2^k, \ldots, x_n^k). \end{gathered}$$

(In general, we may be interested in computing any function f.) More specifically, a protocol works in k *phases.* In phase j each player P_i gets the input bit x_i^j. Then, the players have to compute the jth value $\text{XOR}(x_1^j, x_2^j, \ldots, x_n^j)$ and only after this computation is completed do they get the $(j+1)$st input bit. The computation in each phase operates in *rounds.* In each round each player P_i may toss some coins and then sends messages to the other players. All messages are sent over private channels so that other than the intended receiver of each message no other player can listen to it. The content of each message may depend on all the information available to the sender: its input (in the current and previous phases), its random coins, and the messages it received so far (in the current and previous phases). Player P_i then

[4] To clarify the terminology, the *history* is the transcript of the communication exchanged in the protocol; the *projected history* is that part of the history seen by some player P_i (i.e., only those messages sent from/to P_i); the *view* of P_i includes all the information to which P_i has access, i.e., the projected history as well as its input and random input.

receives the messages sent to it by the other players. During the execution of each phase j, each player P_i produces its output value. The *correctness* of the protocol requires that this value will always be equal to $\text{XOR}(x_1^j, x_2^j, \ldots, x_n^j)$. We use C_i to denote the sequence of messages that player P_i receives during the execution of the protocol. We also use x_i to denote the input seen by P_i during the whole protocol, i.e., $x_i = x_i^1, \ldots, x_i^k$, and use $\vec{x}$ to denote the vector of inputs seen by all players, i.e., $\vec{x} = (x_1, \ldots, x_n)$. We denote by x^j the vector of inputs received by all players in the jth phase, i.e., $x^j = (x_1^j, \ldots, x_n^j)$. Finally, we use $f^k(\vec{x})$ to denote the k-tuple of the function values, i.e., $f^k(\vec{x}) = (f(x^1), \ldots, f(x^k))$.

Informally, *privacy* with respect to player P_i means that player P_i cannot learn anything (in particular, the inputs of the other players) from C_i except what is implied by its input bits, and the value of the function computed. Formally, we have the following definition.[5]

Definition 1 (privacy). *A (k-phase) protocol $\mathcal{A}$ for computing a function f is* private *with respect to player P_i if for any two input vectors $\vec{x}$ and $\vec{y}$, such that $f^k(\vec{x}) = f^k(\vec{y})$ and $x_i = y_i$, for any sequence of messages C, and for any random coins, R_i, tossed by P_i,*

$$\Pr[C_i = C | R_i, \vec{x}] \quad = \quad \Pr[C_i = C | R_i, \vec{y}],$$

where the probability is over the random coin tosses of all other *players.*

A protocol is called *private* if it is private with respect to every P_i.

To measure the amount of randomness used by a protocol we use the next definition.

Definition 2. *An ℓ-random protocol is a protocol such that for every input assignment $\vec{x}$ the total number of coins tossed by all players in* every *execution (during all phases) is at most ℓ.*

Next, we define the round-complexity of a protocol. Note that while in the case of the randomness-complexity it makes sense to measure the *total* number of coins tossed in the protocol over *all* phases, the definition of round-complexity considers each phase separately (that is, it measures the number of rounds that it takes from the time that the input to the phase is given and until the time that the output of this phase is computed).

Definition 3. *An r-round protocol is a protocol such that for every input assignment $\vec{x}$, and every sequence of coin tosses, the number of rounds in each phase j is at most r.*

We emphasize that the definitions allow, for example, that in different executions different players will toss the coins. This may depend both on the input of the players and on the previous coin tosses.

3. Upper bound. In this section we present our positive results. First, we consider the case $k = n - 1$. By the lower bound of section 4, at least $n - 2$ random bits are needed for such a computation, regardless of the number of rounds per phase. The protocol below uses $n-1$ random bits. Of course, there is a naive way to perform this computation using only $n - 1$ random bits, and that is to use a single random bit for each of the $n - 1$ phases. However, such computation takes $\Theta(n)$ rounds per phase, and this is the best one can do with a single random bit (per phase) [35]. Our protocol takes a different direction that allows it to use only $O(1)$ rounds per phase.

[5] The formalization below, which is the most common in the literature on information theoretic privacy, is *perfect* in the sense that it requires the relevant probability distributions to be *equal*. Certain weaker definitions can be found, e.g., in [15].

LEMMA 4. *There is a private, k-phase protocol that computes* XOR *on $k = n - 1$ inputs with $\ell = n - 1$ random bits overall and $r = 2$ rounds per phase. It requires an additional initialization round before the k phases.*

Proof. For the proof we present an appropriate k-phase protocol.

Initialization. Player P_n chooses $n-1$ random bits, denoted $r_1, \dots, r_{n-1}$. It sends bit r_i to player P_i.

Phase j.

1. Each player P_i, $1 \le i \le n-1$, sends a bit $b_{i,j} = x_i^j + r_i$ to player P_j. In addition, player P_n sends to P_j the bit $b_{n,j} = x_n^j + \sum_{m=1}^{n-1} r_m$. (The summations here and elsewhere are all modulo 2.)
2. Player P_j sums the n bits $b_{i,j}$ it received in the previous step. It announces $\sum_{i=1}^{n} b_{i,j}$ as the output for the jth phase.

For the correctness, consider the sum computed by player P_j in step 2. This sum equals

$$\begin{aligned}\sum_{i=1}^{n} b_{i,j} &= \sum_{i=1}^{n-1}(x_i^j + r_i) + \left(x_n^j + \sum_{m=1}^{n-1} r_m\right) \\ &= \text{XOR}(x_1^j, x_2^j, \dots, x_n^j).\end{aligned}$$

For the privacy, note that player P_n receives no message (except the output values) during the protocol, and hence the privacy with respect to P_n certainly holds. Also, observe that during phase j only player P_j receives any message (other than the output of the phase). The communication received by P_j in phase j is the sequence of messages $b_{1,j}, \dots, b_{n-1,j}, b_{n,j}$; the only additional message received by P_j is r_j. Now, observe that for every input $\vec{x}^j = (x_1^j, x_2^j, \dots, x_n^j)$, every communication $r_j, b_{1,j}, \dots, b_{n-1,j}, b_{n,j}$ which is consistent with the output (and the input x_j^j of P_j) has the same probability, $2^{-(n-1)}$. This is because each of $b_{1,j}, \dots, b_{n-1,j}$ determines one of the $n-1$ random bits (which are all independent), $b_{j,j}$ determines r_j, and $b_{n,j}$ is determined by the value of the function and the previously determined values. The privacy of the protocol follows. □

The main idea in the above protocol is that we can compute the XOR of each of the $n-1$ inputs, using $n-1$ random bits, in a way that allows using the same $n-1$ random bits for all the $n-1$ inputs. We can use the same idea, with a bit more precaution, to achieve similar savings for other parameters. The following is a simple corollary of the previous construction.

LEMMA 5. *There is a private, k-phase protocol that computes* XOR *on $k = d(n-1)$ inputs with $\ell = d(n-1)$ random bits overall and $r = O(1)$ rounds per phase.*

Proof. Simply partition the $d(n-1)$ inputs into d sets of size $n-1$. For each set of $n-1$ inputs use the $(n-1)$-phase protocol of Lemma 4 that requires $n-1$ random bits and $O(1)$ rounds. If we do this each time with new and independent random bits, we get the desired result. □

Again note that, by the results of section 4, the above lemma is (almost) optimal in terms of the number of random bits required for this computation. Moreover, if k (the number of inputs) is not divisible by $n-1$ we can always add some dummy inputs to each player to make the number of inputs some k' which is divisible by $n-1$. For example, if $\frac{n-1}{2} \le k < n-1$, then $\ell = n-1$ random bits and $r = O(1)$ rounds suffice. The only case in which this is inefficient is when $k \ll n-1$; in such a case increasing the number of inputs to $k' = n-1$ would be wasteful. For such cases, we use the following construction.

LEMMA 6. *Let s be an integer $(1 \leq s \leq n)$. There is a private, k-phase protocol that computes* XOR *on k inputs, $k < (n-1)/2$ with $\ell = 2k + s$ random bits, and $r \leq \log n / \log s$ rounds per phase. The protocol requires an additional initialization round before the k phases.*

Proof. As in the previous protocols, P_n will be the player that makes the random choices. We partition the other $n-1$ players into g groups of size k. Assume for simplicity that $n-1$ is divisible by k. Moreover, assume that $g = (n-1)/k$ is even (later we describe the modifications required when this is not the case).

Initialization. Player P_n chooses k random bits, denoted $r_1, \dots, r_k$. It sends the bit r_i to the ith player of each of the g groups. P_n chooses s additional random bits $\alpha_1, \dots, \alpha_s$ to be used later.

Phase j.

1. The ith player in each group t $(1 \leq i \leq k,\ 1 \leq t \leq g)$, denoted $P_{i,t}$, sends a bit $b^j_{i,t} = x^j_{i,t} + r_i$ to player $P_{j,t}$ (where $x^j_{i,t}$ denotes the input of $P_{i,t}$ in the current phase, j).
2. Each player $P_{j,t}$ computes $Y^j_t = \sum_{i=1}^{k} b^j_{i,t}$.
3. The jth players of all groups together with P_n participate in a private protocol to compute the sum of $g+1$ bits: $Y^1_j, \dots, Y^g_j$ and x^j_n. They announce the output as the XOR of the jth input. The players do this computation using the protocol of [35]. This protocol, when using s random bits, terminates within $\log(g+1)/\log s$ rounds. In addition, all the random bits in this protocol are chosen by one player who receives no message during the protocol; we choose this player to be P_n and choose $\alpha_1, \dots, \alpha_s$ to be these random bits (note that P_n uses the *same* s random bits in all k phases).

For the correctness note that

$$\begin{aligned}\sum_{t=1}^{g} Y^t_j \ + \ x^j_n &= \sum_{t=1}^{g}\sum_{i=1}^{k} b^j_{i,t} + \ x^j_n \\ &= \sum_{t=1}^{g}\sum_{i=1}^{k} (x^j_{i,t} + r_i) + \ x^j_n \\ &= \sum_{i=1}^{n} x^j_i + g \cdot \left(\sum_{i=1}^{k} r_i\right).\end{aligned}$$

Since we assumed that g is even, the last term contributes 0 to the sum (modulo 2) and so the jth output is $\sum_{i=1}^{n} x^j_i$, as needed. The number of random bits used is $k + s$.

For the privacy, note again that P_n only sends messages during the whole protocol and that in phase j only the jth player of each group receives messages. In step 1 each of the players $P_{j,t}$ receives from the members of its group k bits $b^j_{1,t}, \dots, b^j_{k,t}$ which are distributed uniformly and independently. Then, each player $P_{j,t}$ becomes involved in a private protocol which guarantees that, no matter what is the input to the protocol, each player sees the same distribution of communications for all the possible inputs that $P_{j,t}$ may have and all possible outputs; moreover, $P_{j,t}$ receives messages only in the jth phase of the protocol (other than initialization messages and output values). Also note that the s random bits used for this subprotocol are independent of the random bits used in step 1. Altogether, the privacy follows.

Now, there are some technical issues that we still need to address. First, if the number of groups g is odd, then P_n can always make sure that there will be no contribution of random bits to the result by using as its input in step 3 of phase j the bit x_n^j xored with these random bits. Another technical issue that has to be dealt with is the case in which $n-1$ is not divisible by k. In this case the gth group is of size $k' < k$ and hence cannot use the above protocol. We solve this by letting P_n choose for the members of the gth group k' special random bits. The messages $b_{i,g}^j$ will be sent to the jth player of group 1 instead of the jth player of group g (which may not exist). Since this is done with new random bits, the privacy still holds and the total number of random bits is still at most $2k+s$. □

Combining Lemmas 5 and 6 we get the following theorem.

THEOREM 7. *Let s be an integer $(1 \leq s \leq n)$. There is a private k-phase protocol that computes* XOR *on $k = d(n-1)+q$ inputs, with $\ell = d(n-1)+2q+s$ random bits and $r \leq \max\{\log n/\log s, 2\}$ rounds per phase. The protocol requires an additional initialization round before the k phases.*

An interesting case, obtained by setting $s=k$, is the following.

COROLLARY 8. *There is a private, k-phase protocol that computes* XOR *on k inputs with $\ell = O(k)$ random bits overall and $r \leq \max\{\log n/\log k, 2\}$ rounds per phase. The protocol requires an additional initialization round before the k phases.*

This means that we can use $O(k)$ random bits overall and, in each of the k phases, compute the function in the optimal time for computing XOR on a single input using k bits (i.e., $\Theta(\log n/\log k)$ rounds [35, 22]).

4. Lower bound. In this section we prove a lower bound on the number of random bits required for a k-phase, private computation of XOR (on k instances). This lower bound holds for any number of rounds used by the protocol. We prove the following theorem.

THEOREM 9. *Let $\mathcal{A}$ be a d-random, k-phase, n-player, private protocol computing the function* XOR. *Then $d \geq (1-\frac{2}{n}) \cdot k$.*

(Note that for $n=2$ there is a deterministic protocol for computing XOR; indeed, in this case, the above theorem is trivialized, as it states that $d \geq 0$.) To prove the theorem, we first present the following definition and two technical lemmas. We then show how to derive the theorem from the lemmas. The proofs of these lemmas are deferred to the next subsection.

DEFINITION 10. *Denote by $View_i^t(\vec{x}, \vec{R})$ the view of player P_i at time t on input $\vec{x}$ and vector of random tapes $\vec{R}$. This view consists of the inputs to player P_i received so far, the random tape of player P_i, i.e., R_i, and the communication received by player P_i up to and including time $t-1$.*

In our proof we argue about the number of different views that players can see in various executions of the protocol. The following lemmas are useful for this argument. The proof of Theorem 9 is based on the following two lemmas. The first lemma is very similar to a lemma in [35] and its proof appears (in the next subsection) mainly for self-containment.

LEMMA 11 (see [35]). *Consider a private d-random, k-phase protocol $\mathcal{A}$ computing a Boolean function f. Fix the random tapes of the players to be $\vec{R}$. Then, for any P_i, the view $View_i^t(\cdot, \vec{R})$ can assume at most 2^{2k+d} different values (over the 2^{kn} input assignments).*

LEMMA 12. *Let $\mathcal{A}$ be a deterministic (possibly nonprivate), k-phase, n-player protocol computing the function* XOR. *Then, there is at least one player that can see at least $2^{(3-\frac{2}{n})k}$ views over the 2^{kn} input assignments.*

Proof of Theorem 9. By Lemma 11, if we fix the random tapes of the players, then each player can see (over the different inputs) at most 2^{2k+d} different views. But, by Lemma 12, for the protocol to be correct there must be at least one player that sees at least $2^{(3-\frac{2}{n})k}$ views. Thus $2^{(3-\frac{2}{n})k} \leq 2^{2k+d}$, and the theorem follows. □

4.1. Proofs of the lemmas. To complete the proof of the main theorem of this section, we give below the proofs of Lemmas 11 and 12.

Proof of Lemma 11. In the first step of the proof, we fix an arbitrary input $\vec{x}$ and consider the possible values $View_i^t(\vec{x}, \vec{R})$ over all different choices of random tapes $\vec{R} = (R_1, \ldots, R_n)$. The d-randomness of the protocol implies that the total number of coins tossed is at most d; however, in different executions these coins can be tossed by different players. Nevertheless, we claim that the number of different values $View_i^t(\vec{x}, \vec{R})$ is at most 2^d. For each execution we can order the coin tosses of all players (i.e., the readings from the local random tapes) according to the phases of the protocol, within each phase according to the rounds, and within each round according to the index of the player that tosses them. The identity of the player who tosses the first coin is fixed by $\vec{x}$. The identity of the player who tosses any following coin is determined by $\vec{x}$ and by the outcome of the previous coins. Therefore, the different executions on input $\vec{x}$ can be described using the following binary tree: In each node of the tree we have a name of a player P_j that tosses a coin. The two outgoing edges from this node, labeled 0 and 1 according to the outcome of the coin, lead to two nodes labeled P_k and P_ℓ, respectively (j, k, and ℓ need not be distinct) which are the identities of the players who toss the next coin depending on the outcome of the random choice made by P_j. If no additional coin toss occurs, the node is labeled "nil"; there are no outgoing edges from a nil node. By the d-randomness property of the protocol, the depth of the above tree is at most d, and hence it has at most 2^d root-to-leaf paths. Every possible run of the protocol is described by one root-to-leaf path. Such a path determines all the messages sent in the protocol, which player tosses coins and when, and the outcome of these coins. In particular each such path determines the view for any P_i. Hence, $View_i^t(\vec{x}, \cdot)$ can assume at most 2^d different values.

In the second step of the proof, we first fix a vector of random tapes for the players $\vec{R} = (R_1, \ldots, R_n)$. We now consider the deterministic protocol $\mathcal{A}_{\vec{R}}$ derived from the private protocol $\mathcal{A}$ by fixing these random tapes. We partition the input assignments $\vec{x}$ into 2^{2k} groups according to the input value of x_i (0 or 1) in each of the k phases and according to the output value (0 or 1) in each of the k phases. We argue that the number of different values that $View_i^t(\cdot, \vec{R})$ can assume in $\mathcal{A}_R$, on the different input assignments within each such group, is at most 2^d. For this, fix $\vec{x}$ in one of these 2^{2k} groups and consider any other $\vec{y}$ pertaining to the same group. If the value $View_i^t(\vec{y}, \vec{R})$ is some ξ_i (which includes the input of player P_i, its random input, and the communication it observes), then by the privacy requirement (with respect to player P_i), the view ξ_i must also occur (in $\mathcal{A}$) when the input is $\vec{x}$, and the random tapes are some $\vec{R}' = (R_1', \ldots, R_n')$, where $R_i' = R_i$. However, by the first step of the proof, for a fixed $\vec{x}$, $View_i^t(\vec{x}, \cdot)$ can assume at most 2^d values (over the choice of random tapes). Since this is true for each group, the lemma follows. □

Before proving Lemma 12, we need the following technical claim.

CLAIM 13. *For any nonnegative values $a_{i,j}$ $(1 \leq j \leq q,\ 1 \leq i \leq p)$,*

$$\prod_{j=1}^{q} \left(\sum_{i=1}^{p} a_{i,j} \right) \geq p^q \min_{1 \leq i \leq p} \left\{ \prod_{j=1}^{q} a_{i,j} \right\}.$$

Proof. We have

$$\begin{aligned}\prod_{j=1}^{q}\sum_{i=1}^{p} a_{i,j} &= p^q \prod_{j=1}^{q}\left(\frac{\sum_{i=1}^{p} a_{i,j}}{p}\right)\\ &\geq p^q \prod_{j=1}^{q}\left(\prod_{i=1}^{p} a_{i,j}\right)^{\frac{1}{p}}\\ &= p^q \left(\prod_{i=1}^{p}\prod_{j=1}^{q} a_{i,j}\right)^{\frac{1}{p}}\\ &\geq p^q \min_{1\leq i\leq p}\left(\prod_{j=1}^{q} a_{i,j}\right),\end{aligned}$$

where the first inequality uses the theorem of the arithmetic and geometric means (cf. [25, p. 17]). □

It remains to prove Lemma 12. To this end we turn to the main technical part of this section. For the purpose of the proof, we extend the set of protocols that we look at: we consider k-phase (deterministic, possibly nonprivate) protocols that compute XOR with the modification that for the first instance to be computed, only m of the n players get inputs (alternatively, we can assume that the input of $n - m$ of the players for the first instance is 0). For $k \geq 1$ and $1 \leq m \leq n$ let $\mathcal{A}(k, m)$ be the set of k-phase protocols that correctly compute XOR with the above restriction. We prove the following lemma that applies to this extended class of protocols (the extension of the class of protocols makes the proof by induction easier).

LEMMA 14. *Let $\mathcal{A} \in \mathcal{A}(k, m)$. Let $V_s^{\mathcal{A}}$ be the number of different views player P_s can see (in protocol $\mathcal{A}$) over the $2^{(k-1)n+m}$ inputs. Then,*

$$\Pi_{s=1}^{n} V_s^{\mathcal{A}} \geq 2^{(3n-2)(k-1)+n+2(m-1)}.$$

Proof. We prove the claim by induction on both k and m, where the base case is $k = 1$, $m = 1$. Let $\mathcal{A} \in \mathcal{A}(1, 1)$. That is, one player has an input bit and $\mathcal{A}$ has to ensure that all players "compute" the value of this bit. Obviously for all P_s we have $V_s^{\mathcal{A}} \geq 2$ (as there are two output values), which gives $\Pi_{s=1}^{n} V_s^{\mathcal{A}} \geq 2^n$, as required. For the induction step, let $\mathcal{A} \in \mathcal{A}(k, m)$ for $k > 1$ or $m > 1$. We consider two cases $m > 1$ and $m = 1$.

Case $m > 1$ (and $k \geq 1$). Before the first XOR value is computed by any player there must be at least one nonconstant message sent in the protocol. That is, there must be at least one player P_i that sends a message to some player P_j, and this message is not constant over all input assignments. Consider the first round in which at least one such nonconstant message is sent, and consider one of the nonconstant messages sent in this round. Denote this message by M. Without loss of generality, let M be sent from player P_i to player P_j. Since no nonconstant message is received by P_i before M is sent, M can only depend on the first input of P_i. Without loss of generality, assume that P_i sends the value of its input bit. Let ℓ_s^0 (resp., ℓ_s^1) be the number of possible views of player P_s given that the value of M is 0 (resp., 1). We get that

- $V_i^{\mathcal{A}} = \ell_i^0 + \ell_i^1$.
- $V_j^{\mathcal{A}} = \ell_j^0 + \ell_j^1$.
- For all $k \neq i, j$, $V_k^{\mathcal{A}} \geq \max(\ell_k^0, \ell_k^1)$.

Therefore,

$$\begin{aligned}\Pi_{s=1}^{n} V_s^{\mathcal{A}} &\geq (\ell_i^0 + \ell_i^1)(\ell_j^0 + \ell_j^1)\Pi_{s\neq i,j}\max(\ell_s^0, \ell_s^1)\\ &\geq 4\min(\ell_i^0\ell_j^0, \ell_i^1\ell_j^1)\Pi_{s\neq i,j}\max(\ell_s^0, \ell_s^1)\\ &= 4\ell_i^0\ell_j^0\Pi_{s\neq i,j}\max(\ell_s^0, \ell_s^1)\\ &\geq 4\Pi_{s=1}^{n}\ell_s^0,\end{aligned}$$

where the second inequality follows by Claim 13 and the equality follows by assuming, without loss of generality, that $\ell_i^0\ell_j^0 \leq \ell_i^1\ell_j^1$.

Now, consider a protocol $\mathcal{A}_0$ defined as follows. It is the protocol $\mathcal{A}$ with the modification that P_i has no input, and it behaves as if its input is 0. Since we assume that $\mathcal{A}$ is a correct protocol, $\mathcal{A}_0$ is a correct protocol as well in the class $\mathcal{A}(k, m-1)$.[6] Also, we know that $\mathcal{A}_0$ sends 0 as the value of M. Therefore for any s, $1 \leq s \leq n$, we have $V_s^{\mathcal{A}_0} = \ell_s^0$. We get

$$\begin{aligned}\Pi_{s=1}^{n} V_s^{\mathcal{A}} &\geq 4\Pi_{s=1}^{n}\ell_s^0\\ &= 4\Pi_{s=1}^{n} V_s^{\mathcal{A}_0}\\ &\geq 4\cdot 2^{(3n-2)(k-1)+n+2(m-2)},\end{aligned}$$

where the last inequality follows from the induction hypothesis. We get that

$$\Pi_{s=1}^{n} V_s^{\mathcal{A}} \geq 2^{(3n-2)(k-1)+n+2(m-1)},$$

which concludes the proof of the first case.

Case $m = 1$ (and $k > 1$). This is the case where, in the first phase, there is a single player who has an input bit. The value of the function on this input has to be computed by all players before they receive the next input to be computed. Therefore, the first step of the protocol must be that all players receive messages from which each player can conclude whether this first input is 0 or 1. It follows for each P_s that $V_s^{\mathcal{A}} = \ell_s^0 + \ell_s^1$, where ℓ_s^0 (resp., ℓ_s^1) is the number of different views of player P_s given that the first input bit is 0 (resp., 1). Also note that all players agree on the output. We get

$$\Pi_{s=1}^{n} V_s^{\mathcal{A}} = \Pi_{s=1}^{n}(\ell_s^0 + \ell_s^1).$$

Using Claim 13, we have

$$\Pi_{s=1}^{n}(\ell_s^0 + \ell_s^1) \geq 2^n \min(\Pi_{s=1}^{n}\ell_s^0, \Pi_{s=1}^{n}\ell_s^1),$$

and assuming, without loss of generality, that $\Pi_{s=1}^{n}\ell_s^0 \leq \Pi_{s=1}^{n}\ell_s^1$, we get

$$\Pi_{s=1}^{n} V_s^{\mathcal{A}} \geq 2^n \Pi_{s=1}^{n}\ell_s^0.$$

By the same arguments as those for the first case, we now consider a protocol $\mathcal{A}_0 \in \mathcal{A}(k-1, n)$ defined using protocol $\mathcal{A}$, and we have that $V_s^{\mathcal{A}_0} = \ell_s^0$ for any P_s. Using the induction hypothesis we have

$$\begin{aligned}\Pi_{s=1}^{n} V_s^{\mathcal{A}} &\geq 2^n \Pi_{s=1}^{n}\ell_s^0\\ &= 2^n \Pi_{s=1}^{n} V_s^{\mathcal{A}_0}\\ &\geq 2^n 2^{(3n-2)(k-2)+n+2(n-1)}\\ &= 2^{(3n-2)(k-1)+n},\end{aligned}$$

[6] In case $\ell_i^0\ell_j^0 > \ell_i^1\ell_j^1$ we consider a protocol $\mathcal{A}_1$ that behaves as if the input to P_i is 1, but also negates the outputs of the first set of inputs.

which concludes the proof of the second case. □

We can now complete the proof of Lemma 12.

Proof of Lemma 12. Let $\mathcal{A} \in \mathcal{A}(k,n)$. Then, by Lemma 14,

$$\Pi_{s=1}^{n} V_i^{\mathcal{A}} \geq 2^{(3n-2)(k-1)+n+2(n-1)} = 2^{(3n-2)k}.$$

Therefore there is at least one player P_i such that $V_i^{\mathcal{A}} \geq 2^{(3-\frac{2}{n})k}$. □

Acknowledgments. We wish to thank Pino Persiano for several interactions regarding this paper and Yuval Ishai for discussions on the definitions. We also wish to thank the anonymous PODC '98 referees for useful comments.

REFERENCES

[1] N. Alon, O. Goldreich, J. Hastad, and R. Peralta, *Simple constructions of almost k-wise independent random variables*, Random Structures Algorithms, 3 (1992), pp. 289–304. (*Addendum*, 4 (1993), pp. 119–120.)

[2] J. Bar-Ilan and D. Beaver, *Non-cryptographic fault-tolerant computing in constant number of rounds of interaction*, in Proc. of the 8th Annual ACM Symposium on Principles of Distributed Computing, ACM, New York, 1989, pp. 201–209.

[3] D. Beaver, *Perfect Privacy for Two-Party Protocols*, Technical Report TR-11-89, Harvard University, Cambridge, MA, 1989.

[4] M. Bellare, O. Goldreich, and S. Goldwasser, *Randomness in interactive proofs*, Comput. Complexity, 3 (1993), pp. 319–354.

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

[6] M. Blum and S. Micali, *How to generate cryptographically strong sequences of pseudo-random bits*, SIAM J. Comput., 13 (1984), pp. 850–864.

[7] C. Blundo, A. De-Santis, G. Persiano, and U. Vaccaro, *On the number of random bits in totally private computations*, in Automata, Languages and Programming (Szeged 1995), Lecture Notes in Comput. Sci. 944, Springer, Berlin, pp. 171–182.

[8] C. Blundo, A. Giorgio Gaggia, and D. R. Stinson, *On the dealer's randomness required in secret sharing schemes*, Des. Codes Cryptogr., 11 (1997), pp. 235–259.

[9] C. Blundo, A. De-Santis, and U. Vaccaro, *Randomness in distribution protocols*, Inform. and Comput., 131 (1996), pp. 111–139.

[10] N. H. Bshouty, *On the extended direct sum conjecture*, in Proc. of the 21st Annual ACM Symposium on Theory of Computing, 1989, pp. 177–185.

[11] R. Canetti and O. Goldreich, *Bounds on tradeoffs between randomness and communication complexity*, Comput. Complexity, 3 (1993), pp. 141–167.

[12] R. Canetti, E. Kushilevitz, R. Ostrovsky, and A. Rosén, *Randomness versus fault-tolerance*, J. Cryptography, 13 (2000), pp. 107–142.

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

[14] B. Chor and O. Goldreich, *Unbiased bits from sources of weak randomness and probabilistic communication complexity*, SIAM J. Comput., 17 (1988), pp. 230–261.

[15] B. Chor and E. Kushilevitz, *A zero-one law for Boolean privacy*, SIAM J. Discrete Math., 4 (1991), pp. 36–47.

[16] B. Chor and E. Kushilevitz, *A communication-privacy tradeoff for modular addition*, Inform. Process. Lett., 45 (1993), pp. 205–210.

[17] B. Chor, M. Geréb-Graus, and E. Kushilevitz, *Private computations over the integers*, SIAM J. Comput., 24 (1995), pp. 376–386.

[18] B. Chor, M. Geréb-Graus, and E. Kushilevitz, *On the structure of the privacy hierarchy*, J. Cryptology, 7 (1994), pp. 53–60.

[19] A. Cohen and A. Wigderson, *Dispersers, deterministic amplification, and weak random sources*, in Proc. of the 30th IEEE Symposium on the Foundations of Computer Science, 1989, pp. 14–19.

[20] T. Feder, E. Kushilevitz, M. Naor, and N. Nisan, *Amortized communication complexity*, SIAM J. Comput., 24 (1995), pp. 736–750.

[21] M. Franklin and M. Yung, *Communication complexity of secure computation*, in Proc. of the 24th Annual ACM Symposium on the Theory of Computing, 1992, pp. 699–710.

[22] A. Gál and A. Rosén, *A theorem on sensitivity and applications in private computation*, SIAM J. Comput., 31 (2002), pp. 1424–1437.

[23] G. Galibati and M. J. Fischer, *On the complexity of* 2*-output Boolean networks*, Theoret. Comput. Sci., 16 (1981), pp. 177–185.

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

[25] G. Hardy, J. Littlewood, and G. Pólya, *Inequalities*, Cambridge University Press, Cambridge, UK, 1988.

[26] R. Impagliazzo and D. Zuckerman, *How to recycle random bits*, in Proc. of the 30th IEEE Symposium on the Foundations of Computer Science, 1989, pp. 248–253.

[27] M. Karchmer, E. Kushilevitz, and N. Nisan, *Fractional covers and communication complexity*, SIAM J. Discrete Math., 8 (1995), pp. 76–92.

[28] D. Karger and D. Koller, *(De)randomized construction of small sample spaces in NC*, J. Comput. System Sci., 55 (1997), pp. 402–413.

[29] D. Koller and N. Megiddo, *Constructing small sample spaces satisfying given constraints*, SIAM J. Discrete Math., 7 (1994), pp. 260–274.

[30] H. Karloff and Y. Mansour, *On construction of k-wise independent random variables*, Combinatorica, 17 (1997), pp. 91–107.

[31] E. Kushilevitz and Y. Mansour, *Randomness in private computations*, SIAM J. Discrete Math., 10 (1997), pp. 647–661.

[32] E. Kushilevitz and N. Nisan, *Communication Complexity*, Cambridge University Press, Cambridge, UK, 1997.

[33] J. Kilian, E. Kushilevitz, S. Micali, and R. Ostrovsky, *Reducibility and completeness in private computations,* SIAM J. Comput., 29 (2000), pp. 1189–1208.

[34] E. Kushilevitz, R. Ostrovsky, and A. Rosén, *Characterizing linear size circuits in terms of privacy*, J. Comput. System Sci., 58 (1999), pp. 129–136.

[35] E. Kushilevitz and A. Rosén, *A randomness-rounds tradeoff in private computation*, SIAM J. Discrete Math., 11 (1998), pp. 61–80.

[36] D. Krizanc, D. Peleg, and E. Upfal, *A time-randomness tradeoff for oblivious routing*, in Proc. of the 20th Annual ACM Symposium on the Theory of Computing, 1988, pp. 93–102.

[37] E. Kushilevitz, *Privacy and communication complexity*, SIAM J. Discrete Math., 5 (1992), pp. 273–284.

[38] J. Naor and M. Naor, *Small-bias probability spaces: Efficient constructions and applications*, SIAM J. Comput., 22 (1993), pp. 838–856.

[39] N. Nisan, *Pseudorandom generators for space bounded computation*, Combinatorica, 12 (1992), pp. 449–461.

[40] W. Paul, *Realizing Boolean function on disjoint sets of variables*, Theoret. Comput. Sci., 2 (1976), pp. 383–396.

[41] P. Raghavan and M. Snir, *Memory versus randomization in on-line algorithms*, J. Assoc. Comput. Mach., 40 (1993), pp. 421–453.

[42] L. J. Schulman, *Sample spaces uniform on neighborhoods*, in Proc. of the 24th Annual ACM Symposium on the Theory of Computing, 1992, pp. 17–25.

[43] Q. F. Stout, *Meshes with multiple buses*, in Proc. of the 27th IEEE Symposium on Foundations of Computer Science, 1986, pp. 264–273.

[44] U. Vazirani and V. Vazirani, *Random polynomial time is equal to slightly-random polynomial time*, in Proc. of the 26th IEEE Symposium on the Foundations of Computer Science, 1985, pp. 417–428.

[45] A. C. Yao, *Theory and applications of trapdoor functions*, in Proc. of the 23rd IEEE Symposium on the Foundations of Computer Science, 1982, pp. 80–91.

[46] A. Yao, *Protocols for secure computation*, in Proc. of the 23rd IEEE Symposium on the Foundations of Computer Science, 1982, pp. 160–164.

[47] D. Zuckerman, *Simulating BPP using a general weak random source*, Algorithmica, 16 (1996), pp. 367–391.

SIAM J. DISCRETE MATH.
Vol. 16, No. 4, pp. 545–554

FOURIER ANALYSIS OF A CLASS OF FINITE RADON TRANSFORMS*

FABIO SCARABOTTI†

Abstract. We develop a Fourier analysis for Radon transforms between multiplicity-free permutation representations. Statistical applications of such Radon transforms were given by Diaconis and Rockmore in [*Groups and Computation* (New Brunswick, NJ, 1991), DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 11, AMS, Providence, RI, 1993, pp. 87–104].

Key words. finite Radon transforms, intertwining functions, Gelfand pairs, symmetric group, finite general linear group

AMS subject classifications. Primary 44A12; Secondary 43A90

DOI. 10.1137/S0895480102416398

Introduction. Let G be a finite group acting transitively on two finite sets X and Y. Let $L(X)$ and $L(Y)$ denote the vector spaces of complex valued functions defined, respectively, on X and Y. If I is a *G-invariant incidence relation* between Y and X (i.e., I is a subset of $Y \times X$ such that if $(y, x) \in I$ and $g \in G$, then $(gy, gx) \in I$), we can define the *associated Radon transform* R from $L(Y)$ to $L(X)$ by

$$(Rf)(x) = \sum_{y \in Y : (y,x) \in I} f(y)$$

for every $x \in X$ and $f \in L(Y)$; compare with [13, Appendix C]. Clearly, R is a linear operator that intertwines the permutation representations of G on $L(Y)$ and $L(X)$. The aim of this paper is to develop a suitable Fourier analysis for the Radon transform R in the case where both $L(X)$ and $L(Y)$ are multiplicity-free permutation representations of the group G; we also assume that every irreducible representation in $L(Y)$ is also in $L(X)$. Our investigation follows a suggestion at the end of [4]. In that paper, Diaconis and Rockmore developed, for statistical applications (see also [2]), algorithms for computing projections onto the isotypic subspaces of a permutation representation of G on a space $L(X)$. More precisely, for the case in which K is the isotropy subgroup of a fixed element $x_0 \in X$, they give general algorithms that use a set of generalized spherical functions obtained by projecting the irreducible characters onto the space of bi-K-invariant functions on G; when the permutation representation $L(X)$ is multiplicity free, i.e., (G, K) is a *Gelfand pair*, their algorithms involve ordinary spherical functions. For the particular Gelfand pair $G = S_n$ (the symmetric group) and $K = S_{n-k} \times S_k$, they give another algorithm, which is based on the inversion of a natural Radon transform from $S_n/(S_{n-j} \times S_j)$ to $S_n/(S_{n-k} \times S_k)$, $j \leq k$. R is defined as follows: If $S_n/(S_{n-j} \times S_j)$ and $S_n/(S_{n-k} \times S_k)$ are identified, respectively, with the space of all j-subsets and of all k-subsets of $\{1, 2, \ldots, n\}$, then for every $B \in S_n/(S_{n-k} \times S_k)$ and every $f \in L(S_n/(S_{n-j} \times S_j))$

$$(Rf)(B) = \sum_{A \in S_n/(S_{n-j} \times S_j) : A \subset B} f(A).$$

*Received by the editors October 19, 2002; accepted for publication (in revised form) February 26, 2003; published electronically July 30, 2003.

http://www.siam.org/journals/sidma/16-4/41639.html

†Universitá degli Studi di Roma "La Sapienza," Dipartimento di Metodi e Modelli Matematici per Le Scienze Applicate, Via A. Scarpa 16, I-00161 Roma, Italy (scarabot@dmmm.uniroma1.it).

At the end of their paper, Diaconis and Rockmore posed the problem of finding the connection between Radon transforms and character theory. In the present paper we show that every Radon transform R is a convolution operator whose kernel is a K-invariant function defined on Y; the natural Fourier analysis of R consists in the decomposition of R as a linear combination of operators acting on irreducible subspaces and may be obtained using the intertwining functions studied in [6]. This Fourier analysis tells us whether R is injective and gives a left inverse if it exists (the inversion of the Radon transform on $S_n/(S_{n-j}\times S_j)$, taken from [8], is the key fact in the last algorithm in [4]). We also formalize the method used in [8] and [3] to compute the left inverse of a Radon transform by means of a direct computation; in concrete examples, this method seems simpler than Fourier analysis.

1. Intertwining operators and Radon transforms. Let G be a finite group acting transitively on a finite set X. Fix a point $x_0 \in X$ and define $K = \{g \in G : gx_0 = x_0\}$; thus $X = G/K$. The permutation representation of G on the vector space $L(X)$ of all complex-valued functions on X is defined by setting $(gf)(x) = f(g^{-1}x)$ for every $f \in L(X)$, $x \in X$, and $g \in G$. We suppose $L(X)$ endowed with the natural scalar product: for $f_1, f_2 \in L(X)$ we set $\langle f_1, f_2\rangle = \sum_{x\in X} f_1(x)\overline{f_2(x)}$. The space $L(X)$ is isomorphic to the space of right-K-invariant functions defined on G; the isomorphism is given by the map $f \to \tilde{f}$, where

$$\tilde{f}(g) = f(gx_0).$$

In what follows, especially in the definitions of convolution operators, f and $\tilde{f}$ will be identified.

Now suppose that Y is another homogeneous space for G. Fix a point $y_0 \in Y$ and define $H = \{g \in G : gy_0 = y_0\}$. Suppose that R is an intertwining operator from $L(Y)$ to $L(X)$. That is, R is linear and $R(gf) = g(Rf)$ for every $f \in L(Y)$ and every $g \in G$. Then we may define the function f_0 on Y by setting, for every $y \in Y$,

$$f_0(y) = (R\delta_y)(x_0),$$

where δ_y is the dirac function at y. Note that f_0 is *K-invariant*: if $k \in K$, then $f_0(ky) = (R\delta_{ky})(x_0) = [R(k\delta_y)](x_0) = [k(R\delta_y)](x_0) = (R\delta_y)(k^{-1}x_0) = (R\delta_y)(x_0) = f_0(y)$. Moreover, if $f \in L(Y)$, then $f = \sum_{y\in Y} f(y)\delta_y$, and therefore

$$\begin{aligned}(Rf)(gx_0) &= (g^{-1}Rf)(x_0) = (Rg^{-1}f)(x_0) = \sum_{y\in Y} f(y)(Rg^{-1}\delta_y)(x_0)\\ &= \sum_{y\in Y} f(y)(R\delta_{g^{-1}y})(x_0) = \sum_{y\in Y} f(y)f_0(g^{-1}y).\end{aligned}$$

Conversely, if f_0 is a K-invariant function defined on Y, then the formula

$$(Rf)(gx_0) = \sum_{y\in Y} f(y)f_0(g^{-1}y) \tag{1}$$

defines an operator R that intertwines $L(Y)$ with $L(X)$; it is easy to check that the correspondence $f_0 \to R$ is a linear isomorphism between the space of K-invariant functions on $L(Y)$ and the space of the operators that intertwine $L(X)$ with $L(Y)$. Note that (1) may be written in the following form of convolution:

$$(Rf)(gx_0) = \frac{1}{|H|}\sum_{s\in G}\tilde{f}(gs)\omega(s^{-1}) = \frac{1}{|H|}(\tilde{f} * \omega)(g),$$

where $\tilde{f}(g) = f(gy_0)$ and $\omega(s) = f_0(s^{-1}y_0)$; ω is a function defined on G left-H-invariant and right-K-invariant (thus we will say that ω is H-K-invariant, as in [6]). In what follows, R will be called *the intertwining operator associated to the K-invariant function f_0 (or to the H-K-invariant convolution kernel ω).* Now suppose that $\Omega_0, \Omega_1, \dots, \Omega_j$ are the orbits of K on Y. Their characteristic functions $\mathbf{1}_{\Omega_0}, \mathbf{1}_{\Omega_1}, \dots, \mathbf{1}_{\Omega_j}$ form a basis for the space of K-invariant functions on Y; the kernels $\omega_0, \omega_1, \dots, \omega_j$, defined by $\omega_l(s) = \mathbf{1}_{\Omega_l}(s^{-1}y_0)$, $l = 0, 1, \dots, j$, form a basis for the space of H-K-invariant functions defined on G; and the associated intertwining operators $R_0, R_1, \dots, R_j$, defined as in (1), form a basis for the space of operators that intertwine $L(Y)$ with $L(X)$. If for $l = 0, 1, \dots, j$ we choose an element g_l in G such that $g_l y_0$ belongs to Ω_l, then we have $\mathbf{1}_{\Omega_l}(sy_0) = \mathbf{1}_{Kg_lH}(s)$, and this tells us that the kernel ω_l is the characteristic function $\mathbf{1}_{Hg_l^{-1}K}$ of the double coset $Hg_l^{-1}K$ (clearly $g_0, g_1, \dots, g_l$ is a set of representatives for the double cosets KgH).

If I is a G-invariant incidence relation between Y and X as described at the beginning of the introduction, a moment of reflection shows that the associated Radon transform R may be written as the sum of some of the operators $R_0, R_1, \dots, R_j$; in the most important cases (see the examples in the final part of the paper) R coincides with one of the R_l (in general the most simple). Thus in what follows, we restrict our attention to the operators R_l, which will be called Radon transforms.

Remark. Formula (1) may be written in two other, different ways. Since f_0 is constant on the orbits Ω_l, following [4, p. 99], we may define $f_g(l) = \sum_{y \in \Omega_l} f(gy)$ (note that $g \to f_g(l)$ is right-K-invariant) and write

$$(Rf)(gx_0) = \sum_{l=0}^{j} f_g(l) f_0(\Omega_l).$$

Since f_0 is K-invariant, we may also define the matrix $(r(y,x))_{y \in Y, x \in X}$ by $r(y, gx_0) = f_0(g^{-1}y)$. Thus (1) may be written in the matrix form

$$(Rf)(x) = \sum_{y \in Y} f(y) r(y,x).$$

Note also that $r(sy, sx) = r(y,x)$ for every $y \in Y$, $x \in X$, and $s \in G$ and that the map $f_0 \to r$ is a linear bijection between the space of K-invariant functions on $L(Y)$ and the space of the matrices satisfying this condition; see also [1, pp. 38–39]. From a theoretical point of view, the main difference between the algorithms in [4, p. 99] and [4, p. 102] is the use of these two different ways of writing formula (1) (see also the remark at the end of section 5).

2. Intertwining functions. In what follows we suppose that both $L(Y)$ and $L(X)$ decompose without multiplicity and that every irreducible representation contained in $L(Y)$ is also contained in $L(X)$: $L(X) = \bigoplus_{i=0}^{k} V_i$ and $L(Y) = \bigoplus_{i=0}^{j} V_i$, where V_i, $i = 0, 1, \dots, k$, are distinct irreducible representations, and $j \le k$. In every V_i we choose a normalized K-invariant vector v_i and, if $i \le k$, a normalized H-invariant vector u_i, and we set $\psi_i(g) = \langle gv_i, u_i \rangle$, $\varphi_i(g) = \langle gu_i, v_i \rangle$, $\eta_i(g) = \langle gu_i, u_i \rangle$, $\sigma_i(g) = \langle gv_i, v_i \rangle$ (by Frobenius reciprocity, u_i and v_i exist and are unique up to a multiplicative complex constant of modulus one). Clearly, η_i and σ_i are the spherical functions of the Gelfand pairs (G,H) and (G,K), while φ_i and ψ_i are the intertwining functions of [6]. Intertwining and spherical functions take the place of the characters in our cases: $\frac{1}{|K|}\sum_{k\in K} k$ is the projection on the space of K-invariant functions on G and, if χ_i is the character of the irreducible representation V_i, then

$$\frac{1}{|K|}\sum_{k\in K}\chi_i(gk) = \sigma_i(g)$$

and (we denote by e the identity of G)

$$\begin{aligned}\frac{1}{|K||H|}\sum_{h\in H}\sum_{k\in K}\chi_i(kgh) = \frac{1}{|H|}\sum_{\sigma\in H}\sigma_i(gh) = \frac{1}{|H|}\sum_{h\in H}\langle ghv_i, v_i\rangle \\ = \langle v_i, u_i\rangle\langle gu_i, v_i\rangle = \psi_i(e)\varphi_i(g)\end{aligned}$$

because the projection of a vector $v \in V_i$ on the space of H-invariant vectors is given by $\langle v, u_i\rangle u_i$. The intertwining functions are bi-invariant; thus

$$\psi_i = \sum_{l=0}^{j}\psi_i(g_l^{-1})\mathbf{1}_{Hg_l^{-1}K}, \qquad \varphi_i = \sum_{l=0}^{j}\varphi_i(g_l)\mathbf{1}_{Kg_lH}.$$

We recall that $\{\psi_0, \psi_1, \dots, \psi_j\}$ and $\{\varphi_0, \varphi_1, \dots, \varphi_j\}$ are orthogonal bases for the space of H-K- and K-H-invariant functions on G and $\|\psi_i\|^2 = \sum_{g\in G}|\psi_i(g)|^2 = \frac{|G|}{d_i} = \|\varphi_i\|^2$ [6]. Thus

$$\begin{aligned}(2)\qquad \mathbf{1}_{Hg_l^{-1}K} = \sum_{i=0}^{j}\frac{d_i}{|G|}\left\langle \mathbf{1}_{Hg_l^{-1}K}, \psi_i\right\rangle\psi_i = \sum_{i=0}^{j}\frac{d_i}{|G|}\|\mathbf{1}_{Hg_l^{-1}K}\|^2\overline{\psi_i(g_l^{-1})}\psi_i \\ = \frac{|H||K|}{|G||g_l^{-1}Kg_l\cap H|}\sum_{i=0}^{j}d_i\varphi_i(g_l)\psi_i.\end{aligned}$$

3. Fourier analysis of the Radon transforms $R_0, R_1, \dots, R_j$. For both the space of K-H-invariant functions and the space of H-K-invariant functions we have two bases, the first made up of characteristic functions and the second made up of matrix coefficients of irreducible representations. Now we introduce the corresponding bases for the spaces of intertwining operators (we have already defined the operators R_l). We define the Radon transforms $D_0, D_1, \dots, D_j$ from $L(X)$ to $L(Y)$ and the intertwining operators $T_0, T_1, \dots, T_j$ from $L(Y)$ to $L(X)$ and $S_0, S_1, \dots, S_j$ from $L(X)$ to $L(Y)$ by setting

$$D_l\tilde{f} = \frac{1}{|K|}\tilde{f} * \mathbf{1}_{Kg_lH} \quad \text{for} \quad l = 0, 1, \dots, j \quad \text{and} \quad f \in L(X),$$

$$T_i\tilde{f} = \frac{1}{|H|}\tilde{f} * \psi_i \quad \text{for} \quad i = 0, 1, \dots, j \quad \text{and} \quad f \in L(Y),$$

$$S_i\tilde{f} = \frac{1}{|K|}\tilde{f} * \varphi_i \quad \text{for} \quad i = 0, 1, \dots, j \quad \text{and} \quad f \in L(X).$$

We recall that the convolution by the kernel $\frac{d_i}{|G|}\eta_i$ gives the projection from $L(Y)$ to V_i (this is the key fact in the algorithm in [4, pp. 98–100]). The following proposition gives a simple generalization of this fact.

Proposition 3.1. *The kernel of T_i (resp., of S_i) is $\bigoplus_{t\neq i} V_t$ and its range is the subspace of $L(X)$ (resp., of $L(Y)$) isomorphic to V_i.*

Proof. If $t \neq i$, and f belongs to the subspace of $L(Y)$ isomorphic to V_t, then $T_i\tilde{f} = T_i \frac{d_t}{|H|}\tilde{f} * \eta_t = \frac{d_t}{|H|^2}\tilde{f} * \eta_t * \psi_i = 0$: η_t and ψ_i are matrix coefficients of irreducible nonequivalent representations, so $\eta_t * \psi_i = 0$. Moreover, T_i is nontrivial (resp., ψ_i is nontrivial); thus its range is the V_i in $L(X)$. □

Clearly

$$T_i = \sum_{l=0}^{j} \psi_i(g_l^{-1})R_l, \qquad S_i = \sum_{l=0}^{j} \varphi_i(g_l)D_l. \tag{3}$$

Moreover, formula (2) may be translated in terms of intertwining operators obtaining

$$R_l = \frac{|K||H|}{|G||g_l^{-1}Kg_l \cap H|} \sum_{i=0}^{j} d_i\varphi_i(g_l)T_i, \tag{4}$$

which gives the Fourier analysis of the Radon transform R_l.

Remark. In the case $X = Y$ (and $K = H$) (4) becomes the usual spectral analysis of the invariant operator R_l obtained by means of the spherical Fourier transform (now T_i is a multiple of the orthogonal projection onto the irreducible subspace V_i of $L(Y)$).

4. On the inversion of the Radon transforms. The main problems are to know when the Radon transform R_l is injective and, if it is injective, to compute a left inverse, which is also an intertwining operator. The Fourier analysis of the preceding section gives a solution to these problems. In the following lemma, δ_{it} is the usual Kronecker symbol.

Lemma 4.1. (i) $\psi_i * \varphi_t = \delta_{it}\frac{|G|}{d_i}\eta_i$.

(ii) *S_iT_t is $\delta_{it}\frac{|G|^2}{|K||H|d_i^2}$ times the projection from $L(Y)$ onto the subspace V_i.*

Proof. If $i \neq t$, then $\psi_i * \varphi_t = 0$ because ψ_i and φ_t are matrix coefficients of irreducible nonequivalent representations. If $i = t$, the convolution $\psi_i * \varphi_i$ is a bi-H-invariant function and, as a function of $L(Y)$, it belongs to V_i (it is the convolution of two matrix coefficients of V_i), and thus it must be a multiple of η_i; but from the orthogonality relations for the matrix coefficients of an irreducible representation it follows that $(\psi * \varphi)(e) = \sum_{s\in G}\psi(s)\varphi(s^{-1}) = \sum_{s\in G}|\langle sv, u\rangle|^2 = \frac{|G|}{d_i}$, and so (i) is proved. (ii) follows easily from (i). □

As a consequence of Lemma 4.1 and (4) we have the following theorem.

Theorem 4.2. *The operator R_l is injective if and only if all the numbers $\{\varphi_i(g_l) : i = 0, 1, \ldots, j\}$ are nonzero. If it is injective, it has a left inverse, which is also an intertwining operator, and such inverse is given by the formula*

$$\sum_{i=0}^{j} \frac{|g_l^{-1}Kg_l \cap H|}{|G|} \cdot \frac{d_i}{\varphi_i(g_l)} S_i.$$

Note that this formula is a kind of inverse Fourier transform that must be evaluated using (3) in order to express the inverse of R_l in terms of the operators $D_0, D_1, \ldots, D_j$. In concrete examples, the computation of this Fourier transform may be very complicated. Thus we describe another possible way to compute the inverse of a Radon transform R (if it is injective). Choose a set $\{s_0, s_1, \ldots, s_j\}$ of representatives for the double cosets HgH (we suppose that s_0 is the identity of G)

and define the "Laplace operators" by setting $\Delta_r\tilde{f} = \frac{1}{|H|}\tilde{f} * \mathbf{1}_{Hs_rH}$ (thus Δ_0 is the identity). The operators $\{\Delta_0, \Delta_1, \dots, \Delta_j\}$ on $L(Y)$ form a basis for the space of all operators that intertwine $L(Y)$ with itself. Thus there exists a set of coefficients $\{\alpha_{lr} : l = 0, 1, \dots, j\}$ such that

$$D_lR = \sum_{r=0}^{j} \alpha_{lr}\Delta_r. \tag{5}$$

We may look for a left inverse of R in the form $\sum_{l=0}^{j} \beta_l D_l$, i.e., we may look for coefficients $\beta_0, \beta_1, \dots, \beta_j$ such that $(\sum_{l=0}^{j} \beta_l D_l)R = \sum_{r=0}^{j}(\sum_{l=0}^{j} \beta_l\alpha_{lr})\Delta_r$ is equal to Δ_0. Thus we have to solve the system

$$\begin{cases} \sum_{l=0}^{j} \beta_l\alpha_{l0} = 1, \\ \sum_{l=0}^{j} \beta_l\alpha_{lr} = 0 \quad \text{for } r = 1, 2, \dots, j. \end{cases} \tag{6}$$

Clearly, R is injective if and only if the matrix $(\alpha_{lr})_{l,r=0,\dots,j}$ is nonsingular. In some important examples (see the next sections) (α_{lr}) is a triangular matrix and the system (6) may be solved using standard binomial (or q-binomial) identities (see also [1] for the case $j = 1$). Moreover, if $X = Y$ is a finite distance transitive graph and R is the associated Laplace operator, then (α_{lr}) is a tridiagonal matrix; see [3, pp. 334–338] for an example, [10] for the use of the Fourier transform in this example, and [12] for general background.

Remark. If R is injective, it has a unique left inverse R' in the space of intertwining operators, and we will call it *the* left inverse. Moreover, RR' is always the orthogonal projection onto the range of R, as in the case in [4, pp. 101–102] (if T is an injective linear operator between two vector spaces with inner product, and S is a left inverse of T, then TS is the orthogonal projection onto the range of T Ran$T$ if and only if the kernel of S is the orthogonal complement of Ran$T$).

5. Radon transforms between the Gelfand pairs $(S_n, S_{n-k} \times S_k)$. Let S_n be the symmetric group on $\{1, 2, \dots, n\}$. Let j, k be two nonnegative integers such that $0 \le j < k \le n/2$. We identify $Y = S_n/(S_{n-j} \times S_j)$ and $X = S_n/(S_{n-k} \times S_k)$, respectively, with the space of all j-subsets and with the space of all k-subsets of $\{1, 2, \dots, n\}$. We suppose that $H = S_{n-j} \times S_j$ and $K = S_{n-k} \times S_k$ are the isotropy subgroups, respectively, of the subsets $\{1, 2, \dots, j\}$ and $\{1, 2, \dots, k\}$. In this case the orbits of K on Y, the orbits of H on X, and the orbits of H on Y are given, respectively, by the subsets

$$\begin{aligned} \Omega_l &= \{A \in Y : |A \cap \{1, 2, \dots, k\}| = j - l\}, && l = 0, 1, \dots, j; \\ \Theta_l &= \{B \in X : |B \cap \{1, 2, \dots, j\}| = j - l\}, && l = 0, 1, \dots, j; \\ \Pi_r &= \{A \in Y : |B \cap \{1, 2, \dots, j\}| = j - r\}, && r = 0, 1, \dots, j. \end{aligned}$$

Proposition 5.1. *The intertwining operators R_l, D_l, and Δ_r associated to the characteristic functions of the orbits Ω_l, Θ_l, and Π_r are given by the following formulas: For $A \in Y$ and $B \in X$*

$$R_l\delta_A = \sum_{C \in X : |C \cap A| = j-l} \delta_C, \qquad D_l\delta_B = \sum_{C \in Y : |C \cap B| = j-l} \delta_C, \qquad \Delta_r\delta_A = \sum_{C \in Y : |C \cap A| = j-r} \delta_C.$$

Proof. If R_l is the intertwining operator associated to the orbit Ω_l, $A \in Y$, and $B = g\{1, 2, \dots, k\} \in X$, $g \in S_n$, then

$$(R_l\delta_A)(B) = \mathbf{1}_{\Omega_l}(g^{-1}A) = \begin{cases} 1 & \text{if } |B \cap A| = j - l, \\ 0 & \text{otherwise,} \end{cases}$$

and this proves the first formula. The others may be proved analogously. □

The most important operator is R_0: it is the natural Radon transform from $L(Y)$ and $L(X)$. Its left inverse R' is given in [8]; in our notation it is given by the formula

$$R' = (k-j)\sum_{l=0}^{j} \frac{(-1)^l}{k-j+l}\cdot\frac{1}{\binom{n-j}{k-j+l}}D_l.$$

If $A, C \in Y$ and $|A \cap C| = j - r$, then the number of $B \in X$ such that $A \subset B$ and $|C \cap B| = j - l$ is equal to $\binom{r}{l}\binom{n-j-r}{k-j+l-r}$. Thus in this case (5) is

$$D_l R_0 = \sum_{r=l}^{\min\{k-j+l,j\}} \binom{r}{l}\binom{n-j-r}{k-j+l-r}\Delta_r$$

and the system (6) is triangular; by means of elementary manipulations and of the identity (5.24) in [7], it is simple to verify that R' is the left inverse of R_0. Clearly, R' is the intertwining operator associated to the K-H-invariant convolution kernel $\omega_0 = (k-j)\sum_{l=0}^{j}\frac{(-1)^l}{k-j+l}\cdot\frac{1}{\binom{n-j}{k-j+l}}\tilde{\mathbf{1}}_{\Theta_l}$.

Now we want to connect the formula for ω_0 with the formula in Theorem 4.2 and the theory of the $(S_{n-j}\times S_j - S_{n-k}\times S_k)$-invariant functions on S_n developed in [6]. We use the results in [6] and the notation of the preceding sections of this paper. In this case $L(Y)$ decomposes into $j+1$ distinct irreducible representations that are also contained in $L(X)$. The intertwining functions φ_i are given by the following formula (in [6] a different normalization is used):

$$\varphi_i = \left(\frac{(n-k)!(n-i-j)!j!(k-i)!}{(n-j)!(n-k-i)!(j-i)!k!}\right)^{1/2}\sum_{l=0}^{j} Q_i(l;-(n-k)-1,-k-1,j)\tilde{\mathbf{1}}_{\Theta_l},$$
$$i = 0, 1, \dots, j,$$

where Q_i is the Hahn polynomial [12]

$$Q_i(l;-(n-k)-1,-k-1,j) = \frac{1}{\binom{j}{i}}\sum_{s=0}^{i}(-1)^s\frac{\binom{k-i+s}{s}}{\binom{n-k}{s}}\binom{j-l}{i-s}\binom{l}{s}.$$

Note that the value of φ_i on Ω_0 is $\left(\frac{(n-k)!(n-i-j)!j!(k-i)!}{(n-j)!(n-k-i)!(j-i)!k!}\right)^{1/2}$, and thus it is nonzero for every i. By Theorem 4.2, this confirms that R_0 is injective. In this case the operator given by Theorem 4.2 is the intertwining operator associated to the K-H-invariant convolution kernel ω_1 given by

$$\omega_1 = \frac{j!(k-j)!(n-k)!}{n!}\sum_{i=0}^{j}\left(\frac{(n-k)!(n-i-j)!j!(k-i)!}{(n-j)!(n-k-i)!(j-i)!k!}\right)^{-1/2}\left[\binom{n}{i}-\binom{n}{i-1}\right]\varphi_i.$$

In fact the dimension of the irreducible representation corresponding to φ_i is $[\binom{n}{i} - \binom{n}{i-1}]$, $|K \cap H| = j!(k-j)!(n-k)!$, and $|G| = n!$. However, R_0 has a unique left inverse in the space of intertwining operators, and thus we have $\omega_0 = \omega_1$; a simple calculation shows that this fact is equivalent to the following formula for the Hahn polynomials:

(7)
$$\sum_{i=0}^{j}\left[\binom{n}{i}-\binom{n}{i-1}\right]Q_i(l;-n+k-1,-k-1,j) = \binom{n}{j}\binom{n-j}{k-j}\frac{(-1)^l}{k-j+l}\frac{k-j}{\binom{n-j}{k-j+l}}.$$

An analytic proof of (7) may be easily obtained using the orthogonality relations for the Hahn polynomials and (5.26) in [7].

Remark. As noted in the remark of section 1, the main theoretical difference between the algorithms in [4, p. 99] and [4, pp. 101–102] is the different way to write (1). Moreover, the second algorithm contains implicitly an expression for the spherical functions of the Gelfand pair $(S_n, S_{n-k} \times S_k)$, which is different that in the literature [6], [12].

6. A Radon transform from $S_{2n}/(S_n wr S_2)$ to $S_{2n}/(S_2 wr S_n)$. Let wr denote the wreath product of finite groups (see [9]). Then $Y = S_{2n}/(S_n wr S_2)$ may be identified with the space of all partitions of $\{1, 2, \dots, 2n\}$ in two parts of size n, while $X = S_{2n}/(S_2 wr S_n)$ may be identified with the space of all partitions of $\{1, 2, \dots, 2n\}$ in n parts of size two (in both cases there is no order between or within parts). It is known [11] that the permutation representations of S_{2n} on $L(Y)$ and $L(X)$ are multiplicity free and that every irreducible subrepresentation of $L(Y)$ is also contained in $L(X)$ (but in this case $L(X)$ is much bigger than $L(Y)$). If $y \in Y, x \in X, y = \{A, B\}$, and $x = \{A_1, A_2, \dots, A_n\}$ we define $w(y, x) = |\{s : A_s \text{ is contained in } A \text{ or in } B\}|$; i.e., $w(y, x)$ is the number of parts of x that are contained in a part of y. Clearly, $w(y, x)$ is an even number. If $S_n wr S_2$ $(S_2 wr S_n)$ is the isotropy group of y_0 (of x_0), then the orbits of $S_n wr S_2$ (of $S_2 wr S_n$) on X (resp., on Y) are given by the subsets $\{x \in X : w(y_0, x) = 2l\}$ (resp., $\{y \in Y : w(y, x_0) = 2l\}$), $l = 0, 1, \dots, [n/2]$. The Radon transforms associated to these orbits are given by the formulas

$$R_l \delta_y = \sum_{x : w(y,x) = 2l} \delta_x, \qquad R_l \text{ from } L(Y) \text{ to } L(X),$$

$$D_l \delta_x = \sum_{y : w(y,x) = 2l} \delta_y, \qquad D_l \text{ from } L(X) \text{ to } L(Y).$$

Then we define the invariant operators Δ_r, $0 \le r \le [n/2]$, setting, for $y = \{A, B\} \in Y$,

$$\Delta_r \delta_y = \sum \delta_z,$$

where the sum is over all $z = \{A', B'\} \in Y$ such that $\{|A \cap A'|, |A \cap B'|\} = \{r, n - r\}$ (thus z may be obtained from y by moving r elements from A to B and r elements from B to A). The set $\{\Delta_r : r = 0, 1, \dots, [n/2]\}$ is a basis for the space of all operators that intertwine $L(Y)$ with itself; it is formed by the operators associated to the orbits of $S_n wr S_2$ on Y.

The most simple Radon transform is R_0; we want to invert it, solving the corresponding system (6). If $y, z \in Y$, $y = \{A, B\}$, $z = \{A', B'\}$, and $\{|A \cap A'|, |A \cap B'|\} = \{n, n - r\}$, $0 \le r \le n/2$, then the number of $x \in X$ such that $w(y, x) = 0$ and $w(z, x) = 2l$ is equal to $\binom{n-r}{l}^2 \binom{r}{l}^2 (l!)^2 (r - l)!(n - r - l)! = \binom{n-r}{l}\binom{r}{l} r!(n - r)!$. Therefore in this case (5) is

$$D_l R_0 = \sum_{r=l}^{\min\{[n/2], n-l\}} r!(n - r)! \binom{n - r}{l} \binom{r}{l} \Delta_r.$$

Thus the matrix of the system (6) is triangular and its diagonal elements are different from zero: it follows that R is injective. Moreover, this system may be easily solved:

by using formula (5.25) in [7] it is not hard to prove that the left inverse R' of R_0 is given by the formula

$$R' = \sum_{l=0}^{[n/2]} \frac{(-1)^l}{n!\binom{n-1}{l}} D_l.$$

7. The Radon transform on the finite Grasmann manifold $GL_n(F_q)/(GL_{n-k}(F_q) \times GL_k(F_q))$. In this section we treat the inversion of the q-analogue of the Radon transform of section 5 (the study of this Radon transform is suggested in [4, p. 103]). Let $GL_n(F_q)$ be the group of all nondegenerate linear transformations of F_q^n, which is the n-dimensional vector space over the finite field F_q of q elements. Then the finite homogeneous space $GL_n(F_q)/(GL_{n-k}(F_q) \times GL_k(F_q))$ may be identified with the Grasmann manifold $X^k(q)$ of all k-dimensional subspaces of F_q^n. Let $0 \leq j < k \leq [n/2]$; then it is known [5] that $L(X^j(q))$ decompose into $j+1$ nonequivalent irreducible representations of $GL_n(F_q)$ and that every irreducible representation in $L(X^j(q))$ is also in $L(X^k(q))$. The natural Radon transform from $L(X^j(q))$ to $L(X^k(q))$ is defined by

$$(Rf)(x) = \sum_{y \in X^j(q): y \subset x} f(y) \quad \forall x \in X^k(q), \qquad f \in L(X^j(q)).$$

In this case the operator associated to the orbits of $GL_{n-j}(F_q) \times GL_j(F_q)$ on $X^k(q)$ and on $X^j(q)$ is given by

$$D_l \delta_x = \sum_{y \in X^j(q): \dim(x \cap y) = j-l} \delta_y \quad \forall x \in X^k(q), \qquad l = 0, 1, \ldots, j;$$

$$\Delta_r \delta_y = \sum_{z \in X^j(q): \dim(z \cap y) = j-r} \delta_z \quad \forall y \in X^j(q), \qquad l = 0, 1, \ldots, j.$$

We recall that if $y, z \in X^j(q)$ and $\dim(y \cap z) = j - r$, then the number of $x \in X^k(q)$ such that $z \subset x$ and $\dim(y \cap x) = j - l$ is equal to [5, p. 13].

$$\binom{r}{l}_q \binom{n-j-r}{k-j+l-r}_q q^{l(k-j+l-r)},$$

where $\binom{x}{k}_q = \frac{(q^x-1)(q^{x-1}-1)\cdots(q^{x-k+1}-1)}{(q^k-1)(q^{k-1}-1)\cdots(q-1)}$ if k is positive, $\binom{x}{0}_q = 1$, and $\binom{x}{k}_q = 0$ if k is a negative integer. It follows that, in this case, formula (5) is

$$D_l R = \sum_{r=l}^{\min\{j,k-j+l\}} \binom{r}{l}_q \binom{n-j-r}{k-j+l-r}_q q^{l(k-j+l-r)} \Delta_r.$$

Again the matrix of the system (6) is triangular and its diagonal elements are different from zero. Therefore R is injective. Its left inverse R' is given by the formula

$$R' = (1 - q^{k-j}) \sum_{l=0}^{j} \frac{(-1)^l}{1 - q^{k-j+l}} \frac{1}{\binom{n-j}{k-j+l}_q} q^{-l(k-j)+(l-l^2)/2} D_l.$$

This may be proved using the following q-analogue of (5.24) in [7]:

$$\sum_k \binom{l}{m+k}_q \binom{s+k}{n}_q (-1)^k q^{(k^2+k)/2+k(m-n-1)}$$
$$= (-1)^{l+m} q^{(l-m)(l-2n+m-1)/2} \binom{s-m}{n-l}_q, \qquad l \geq 0,$$

which may be easily proved by induction on l.

REFERENCES

[1] E. Bolker, *The finite Radon transform*, Contemp. Math., 63 (1987), pp. 27–50.

[2] P. Diaconis, *Group Representation in Probability and Statistics*, Institute of Mathematical Statistics, Hayward, CA, 1988.

[3] P. Diaconis and R. L. Graham, *The Radon transform on* $\mathbf{Z}_2^k$, Pacific J. Math., 118 (1985), pp. 323–345.

[4] P. Diaconis and D. Rockmore, *Efficent computation of isotypic projection for the symmetric group*, in Groups and Computation (New Brunswick, NJ, 1991), DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 11, AMS, Providence, RI, 1993, pp. 87–104.

[5] C. Dunkl, *An addition theorem for some q-Hahn polynomials*, Monatsh. Math., 85 (1977), pp. 5–37.

[6] C. Dunkl, *Spherical functions on compact groups and applications to special functions*, Sympos. Math., 22 (1979), pp. 145–161.

[7] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, Addison-Wesley, Reading, MA, 1989.

[8] R. L. Graham, S.-Y. R. Li, and W.-C. W. Li, *On the structure of t-designs*, SIAM J. Algebraic Discrete Methods, 1 (1980), pp. 8–14.

[9] G. D. James and A. Kerber, *The Representation Theory of the Symmetric Group*, Encyclopedia of Mathematics and Its Applications 16, Addison-Wesley, Reading, MA, 1981.

[10] J. A. Morrison, *Weighted averages of Radon transforms on* $\mathbf{Z}_2^k$, SIAM J. Algebraic Discrete Methods, 7 (1986), pp. 404–413.

[11] J. Saxl, *On multiplicity-free permutation representations*, in Finite Geometries and Designs, London Math. Soc. Lecture Note Ser. 49, Cambridge University Press, Cambridge, UK, 1981, pp. 337–353.

[12] D. Stanton, *Orthogonal polynomials and Chevalley groups*, in Special Functions: Group Theoretical Aspects and Applications, R. Askey, T. Koornwinder, and W. Schempp, eds., D. Reidel, Dordrecht, 1984, pp. 87–128.

[13] S. Sternberg, *Group Theory and Physics*, Cambridge University Press, Cambridge, UK, 1994.

SIAM J. DISCRETE MATH.
Vol. 16, No. 4, pp. 555–590

OPTIMAL ONLINE ALGORITHMS FOR MINIMAX RESOURCE SCHEDULING*

BRADY HUNSAKER†, ANTON J. KLEYWEGT†, MARTIN W. P. SAVELSBERGH†, AND CRAIG A. TOVEY†

Abstract. We consider a very general online scheduling problem with an objective to minimize the maximum level of resource allocated. We find a simple characterization of an optimal deterministic online algorithm. We develop further results for the two, more specific problems of single resource scheduling and hierarchical line balancing. We determine how to compute optimal online algorithms for both problems using linear programming and integer programming, respectively. We show that randomized algorithms can outperform deterministic algorithms, but only if the amount of work done is a nonconcave function of resource allocation.

Key words. online algorithms, competitive analysis, worst-case analysis, single-machine scheduling, multiprocessor scheduling, line balancing

AMS subject classifications. 68Q25, 90B18, 90B35, 90C47

DOI. 10.1137/S0895480101397761

1. Introduction.

Consider the following problem: Work with different deadlines arrives over time and has to be performed using a resource. The quantities of work that arrive as well as their deadlines become known only at the times of arrival. At a given set of time points, the decision maker decides how much resource to allocate and what part of the available work to perform at that time. The objective is to minimize the maximum amount of resource allocated at any time during the planning period. This problem is called the *online resource minimization problem* (ORMP). It occurs in production scheduling settings in which the major cost component is energy consumption (Kleywegt et al. [13]). The decision maker would like to spread the workload out as evenly as possible over time but faces the dilemma of uncertainty about future work. That is, the decision maker must make a trade-off between allocating too much resource early and postponing too much work to be completed with work that arrives later.

Consider another problem: Work with different requirements arrives over time and has to be assigned to a collection of machines with different capabilities. The machines form a linear hierarchy based on their capabilities; i.e., machine j has at least the same capabilities as machine $j - 1$. The amount of work that arrives as well as the required machine capabilities become known only at the time of arrival. At a given set of decision points, the decision maker decides how to assign to the machines the work that has arrived since the previous decision point. The objective is to minimize the maximum amount of work assigned to any machine. This problem is called the *hierarchical line balancing problem* (HLBP).

*Received by the editors November 12, 2001; accepted for publication (in revised form) February 26, 2003; published electronically July 30, 2003. Some of these results were reported in A. J. Kleywegt, V. S. Nori, M. W. P. Savelsbergh, and C. A. Tovey, "Online Resource Minimization," in *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '99)*, Baltimore, MD, January 1999, SIAM, Philadelphia, pp. 576–585.

http://www.siam.org/journals/sidma/16-4/39776.html

†School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205 (hunsacker@isye.gatech.edu, Anton.Kleywegt@isye.gatech.edu, mwps@isye.gatech.edu, ctovey@isye.gatech.edu). The second author was supported by the National Science Foundation under grant DMI-9875400.

In both the ORMP and the HLBP, the quality of an algorithm is evaluated by its competitive ratio, i.e., the worst-case ratio over all possible instances of the value of the solution produced by the algorithm and the value of the optimal solution with perfect information.

In this paper, we introduce a simple parameterized deterministic algorithm, called the α-policy, with parameter α and competitive ratio α, provided it produces a feasible solution. We show that with an appropriate choice of parameter α, the α-policy has as good a competitive ratio as any other deterministic algorithm. Under a convexity assumption, which holds for both the ORMP and the HLBP, the α-policy is also optimal among all randomized algorithms. However, we show that randomized algorithms can outperform deterministic algorithms in other cases.

We also investigate how the optimal parameter can be computed for the ORMP and the HLBP. For the ORMP, we construct linear programs to compute the optimal parameters, which depend on the number of time periods. For the HLBP, the optimal parameters are computed with linear programming if the number of time periods is greater than or equal to the number of machines and computed with integer programming for the remaining cases. Interestingly, the resulting parameter values (and hence the optimal competitive ratios) for realistic finite numbers of machines and time periods are substantially lower than the asymptotic values.

The ORMP and the HLBP are special cases of a more general problem, the *online min-max problem* (OMMP), and several of our results can be extended to the OMMP.

The ORMP appears to be a new problem, as we have not been able to find any existing literature discussing it. On the other hand, the HLBP is a known problem and has been studied, for example, by Bar-Noy, Freund, and Naor [6], who distinguish a fractional variant and an integral variant. In the integral variant work must be assigned in its entirety to a machine; in the fractional variant work may be split among eligible machines. Our results are for the fractional variant. Bar-Noy, Freund, and Naor [6] give an algorithm with asymptotically optimal competitive ratio e in the limiting case, where the number of machines goes to infinity. When all machines have the same capabilities, i.e., there is no linear hierarchy among machines, the fractional variant of the HLBP is trivial: divide arriving work equally among the machines. On the other hand, when all machines have the same capabilities, the integral variant of the HLBP remains difficult, as it is equivalent to minimizing the makespan on a set of parallel identical machines. Already in 1966, Graham [10] had presented a greedy algorithm with a competitive ratio of $2 - 1/m$ for online makespan minimization on m identical parallel machines. This problem has continued to attract researchers; see, for example, the recent papers by Albers [1], Bartal et al. [7], Fleischer and Wahl [9], and Seiden [17]. Fleischer and Wahl [9] present the current best deterministic online algorithm, and Albers [1] presents the current best randomized algorithm. In Aspnes et al. [2] an 8-competitive algorithm is given for online makespan minimization on related parallel machines, i.e., where the processing requirement of a job is determined not only by the length of the job but also by the speed of the machine. This was improved by Berman, Charikar, and Karpinski [8]. Azar, Naor, and Rom [5] consider a more general online load balancing problem in which each job can be handled only by a subset of machines and requires a different level of service. In load balancing problems a distinction is made between permanent and temporary jobs. Permanent jobs continue forever after they arrive and load the machine indefinitely, while temporary jobs load the machine only during the interval in which they are active. Azar, Broder, and Karlin [3] and Azar et al. [4] extend the work mentioned above to handle temporary jobs. Finally, we want to mention the work by Hoogeveen and Vestjens [11]. They consider the problem of minimizing the maximum delivery time on a single machine and present an optimal deterministic online algorithm. This is one of the few cases that we are aware of in which an optimal online algorithm is presented. Note that our α-policy, with an appropriate choice of α, is also an optimal policy.

The paper is organized as follows. In section 2, we define the OMMP, the ORMP, and the HLBP. In section 3, we introduce an optimal online algorithm called the α-policy. In section 4, we establish optimality results for the ORMP, the HLBP, and we generalize these results for the OMMP. In section 5, we show that randomized algorithms can outperform deterministic algorithms, but only if the amount of work done is a nonconcave function of resource allocation. Finally, in section 6, we point out future research directions.

2. Problem definition.

In this section we define the general problem, OMMP, as well as two special cases of it, the ORMP and the HLBP.

2.1. OMMP.

DEFINITION 2.1 (OMMP). *An* instance ω *of the OMMP is a finite sequence* $(a(1),\ldots,a(T))$ *of length* T*. Each* $a(t)$ *could be a number, vector, function, set, problem instance, or any other object, and it could be different for different* t*. The nature of each* $a(t)$ *is determined by the particular type of OMMP, as illustrated in the examples of sections* 2.2 *and* 2.3*. The set of all instances is denoted by* Ω*. Often it is of interest to explore the characteristics of the OMMP as a function of problem parameters. For that purpose, the set of all instances with parameter* β *is denoted by* Ω_β*. For example, the set of all instances of length* T *is denoted by* Ω_T*. For any instance* $\omega = (a(1),\ldots,a(T)) \in \Omega_\beta$*, and any* $t \in \{1,\ldots,T\}$*, let* $\omega^t \equiv (a(1),\ldots,a(t))$ *denote the first* t *elements of instance* ω*; that is,* ω^t *denotes the history of instance* ω *up to time* t*. Let* $\Omega_\beta^t \equiv \{\omega^t : \omega \in \Omega_\beta\}$ *denote the set of all such partial instances of the first* t *elements. For any instance of length* T*, a* solution r *of the OMMP is a sequence* $(r(1),\ldots,r(T)) \in \mathbb{R}_+^T$ *of* T *nonnegative real numbers. Let* $\mathbb{R}_+^\infty \equiv \bigcup_{T=1}^\infty \mathbb{R}_+^T$ *denote the set of all solutions. For any instance* ω*, let* $\mathcal{R}(\omega)$ *denote the set of feasible solutions. A* deterministic algorithm π *for the OMMP is a function* $\pi : \Omega \mapsto \mathbb{R}_+^\infty$*, such that for any* $\omega \in \Omega_T$*,* $\pi(\omega) \in \mathbb{R}_+^T$*; i.e., if* ω *is of length* T*, then* $\pi(\omega)$ *is also of length* T*. A deterministic algorithm* π *is called* feasible *if, for every* ω*,* $\pi(\omega) \in \mathcal{R}(\omega)$*. Let* $\mathcal{B}_+^T$ *denote the Borel sets on* $\mathbb{R}_+^T$*, let* $\mathcal{P}^T$ *denote the set of probability measures on* $\mathcal{B}_+^T$*, and let* $\mathcal{P} \equiv \bigcup_{T=1}^\infty \mathcal{P}^T$*. A* randomized algorithm π *for the OMMP is a function* $\pi : \Omega \mapsto \mathcal{P}$ *such that for any* $\omega \in \Omega_T$*,* $\pi(\omega) \in \mathcal{P}^T$*. The probability that the solution is in a set* $B \in \mathcal{B}_+^T$ *is denoted by* $\pi(\omega)[B]$*. We assume that* $\mathcal{R}(\omega) \in \mathcal{B}_+^T$ *for* $\omega \in \Omega_T$*, and a randomized algorithm* π *is called* feasible *if, for every* ω*,* $\pi(\omega)[\mathcal{R}(\omega)] = 1$*. Also, for any* $t \in \{1,\ldots,T\}$*, we will use* $\pi(\omega)(t)$ *to denote the decision at time* t *for instance* ω *under algorithm* π*. For a deterministic algorithm* π*,* $\pi(\omega)(t)$ *is deterministic, and for a randomized algorithm* π*,* $\pi(\omega)(t)$ *is a random variable, where the tuple* $(\pi(\omega)(1),\ldots,\pi(\omega)(T))$ *is distributed according to the probability measure* $\pi(\omega)$*. When the instance* ω *has been fixed, we also use* $r^\pi(t)$ *to denote the decision* $\pi(\omega)(t)$ *under algorithm* π *at time* t*. A* deterministic (randomized) online algorithm π *for the OMMP is a deterministic (randomized) algorithm such that, for each* ω *and each* t*, (the probability distribution of)* $\pi(\omega)(t)$ *depends on* ω^t *only; i.e., it depends only on the history of instance* ω *up to time* t *and not on the whole instance* ω*. Let* Π^{DO} *denote the set of all deterministic online algorithms, and let* $\Pi^{RO} \supseteq \Pi^{DO}$ *denote the set of all randomized online algorithms for the OMMP.*

For any instance $\omega \in \Omega_T$, and any deterministic algorithm π, the value $v^\pi(\omega)$ *is the maximum norm of $\pi(\omega)$; i.e.,*

$$v^\pi(\omega) \quad \equiv \quad \max\Big\{r^\pi(1), \ldots, r^\pi(T)\Big\}$$

if $\pi(\omega) \in \mathcal{R}(\omega)$, and $v^\pi(\omega) = \infty$ otherwise. Similarly, for any instance $\omega \in \Omega_T$, and any randomized algorithm π, the value $v^\pi(\omega)$ is the expected maximum norm under $\pi(\omega)$; i.e.,

$$v^\pi(\omega) \quad \equiv \quad E^{\pi(\omega)}\Big[\max\Big\{r^\pi(1), \ldots, r^\pi(T)\Big\}\Big]$$

if $\pi(\omega)[\mathcal{R}(\omega)] = 1$, and $v^\pi(\omega) = \infty$ otherwise.

For any instance $\omega \in \Omega_T$, the optimal value with perfect information, *$v^*(\omega)$, is defined by*

$$v^*(\omega) \quad \equiv \quad \inf_{r \in \mathcal{R}(\omega)} \Big\{\max\{r(1), \ldots, r(T)\}\Big\}.$$

We assume that $v^*(\omega) < \infty$ for all $\omega \in \Omega$, so $\mathcal{R}(\omega) \neq \varnothing$.

In this paper, the quality of an algorithm π for an instance ω is evaluated by the ratio of the value of the algorithm and the optimal value with perfect information, i.e., $v^\pi(\omega)/v^*(\omega)$. For any class Ω_β of instances, and any algorithm π, the competitive ratio or worst-case ratio ρ^π_β denotes the largest ratio of the value of algorithm π and the optimal value with perfect information over all instances in Ω_β, i.e.,

$$\rho^\pi_\beta \quad \equiv \quad \inf\Big\{\rho \geq 1 \;:\; v^\pi(\omega) \leq \rho v^*(\omega) \quad \forall\, \omega \in \Omega_\beta\Big\}. \tag{2.1}$$

The convention is that $\inf \varnothing = \infty$. *Note that, if $v^*(\omega) > 0$ for all $\omega \in \Omega_\beta$, then*

$$\rho^\pi_\beta \quad = \quad \sup_{\omega \in \Omega_\beta} \left\{\frac{v^\pi(\omega)}{v^*(\omega)}\right\}.$$

Also note that, if $\rho^\pi_\beta < \infty$, then the infimum in (2.1) *is attained, in the sense that $v^\pi(\omega) \leq \rho^\pi_\beta v^*(\omega)$ for all $\omega \in \Omega_\beta$. This criterion is standard in the literature for online algorithms; see, for example, McGeoch and Sleator* [15] *and Irani and Karlin* [12]. *The competitive ratio ρ^π of algorithm π over all instances is given by*

$$\rho^\pi \quad \equiv \quad \inf\Big\{\rho \geq 1 \;:\; v^\pi(\omega) \leq \rho v^*(\omega) \quad \forall\, \omega \in \Omega\Big\}.$$

If $\mathbf{B}$ *is the set of parameters β, i.e., $\Omega = \bigcup_{\beta \in \mathbf{B}} \Omega_\beta$, then $\rho^\pi = \sup_{\beta \in \mathbf{B}} \rho^\pi_\beta$.*

The optimal competitive ratio ρ^*_β over all deterministic online algorithms, and over all instances in Ω_β, is given by

$$\rho^*_\beta \quad \equiv \quad \inf_{\pi \in \Pi^{DO}} \rho^\pi_\beta.$$

The optimal competitive ratio ρ^* over all deterministic online algorithms, and over all instances, is given by

$$\rho^* \quad \equiv \quad \inf_{\pi \in \Pi^{DO}} \rho^\pi.$$

The optimal competitive ratios over all randomized online algorithms are defined similarly.

Alternatively, one may want to define $\rho^* \equiv \sup_{\beta\in\mathbf{B}} \rho^*_\beta$. *The question is whether the two definitions of* ρ^* *are equal, that is, whether* $\sup_{\beta\in\mathbf{B}} \inf_{\pi\in\Pi^{RO}} \rho^\pi_\beta = \inf_{\pi\in\Pi^{RO}} \sup_{\beta\in\mathbf{B}} \rho^\pi_\beta$. *In general, for any real valued function* $f(x,y)$, *it holds that* $\sup_x \inf_y f(x,y) \leq \inf_y \sup_x f(x,y)$, *and the inequality may be strict. Thus* $\sup_{\beta\in\mathbf{B}} \inf_{\pi\in\Pi^{RO}} \rho^\pi_\beta \leq \inf_{\pi\in\Pi^{RO}} \sup_{\beta\in\mathbf{B}} \rho^\pi_\beta$. *Lemma* 2.2 *establishes that if the parameter* β *is known beforehand by the decision maker, then the two definitions of* ρ^* *are in fact equal. If the parameter* β *is not known beforehand, then the inequality may be strict, because for each* β, *there may be an algorithm* π_β *that performs particularly well for instances in* Ω_β, *but there may not exist a single algorithm* π *that does not depend on* β *(because* β *is not known beforehand) that performs well for all* β. *For the special cases of the OMMP and the choices of parameter* β *considered in this paper,* β *is known beforehand.*

LEMMA 2.2. *If* β *is known beforehand by the decision maker, then*

$$\rho^* \quad = \quad \sup_{\beta\in\mathbf{B}} \rho^*_\beta.$$

An algorithm $\pi^* \in \Pi^{DO}$ *is called* optimal *over deterministic online algorithms if* $\rho^{\pi^*}_\beta = \rho^*_\beta$ *for all* $\beta \in \mathbf{B}$.

For any $r_1, r_2 \in \mathbb{R}^T$, *we denote* $r_1 \leq r_2$ *if* $r_1(t) \leq r_2(t)$ *for all* t. *Since the objective is to minimize* $\max\{r(1),\ldots,r(T)\}$, *it is natural to assume that increasing* r *does not adversely impact the feasibility of the solution, so there is a trade-off between smaller values of* r *for improving the objective value and larger values of* r *for improving the feasibility. It is therefore assumed that* $\mathcal{R}$ *has the* feasibility monotonicity property; *that is, for any* $\omega \in \Omega$ *and any* $r_1 \in \mathcal{R}(\omega)$, *it holds that* $r_2 \in \mathcal{R}(\omega)$ *for all* $r_2 \geq r_1$.

2.2. ORMP. In this section we define the ORMP and show that it is a special case of the OMMP.

DEFINITION 2.3 (ORMP). *Work with different deadlines arrives over time and has to be performed using a costly resource. The amount of work that arrives at each point in time as well as its deadlines become known only at the time of arrival. At a given set of decision points, indexed with* $t = 1,\ldots,T$, *the decision maker decides how much resource to allocate and what part of the available work to perform at that time. The objective is to minimize the maximum amount of resource allocated at any time during the planning period. Let* $a_u(t) \in \mathbb{R}_+$ *denote the amount of work that arrives at time* t *with deadline* u, *with* $t, u \in \{1,\ldots,T\}$, *and let* $a(t) \equiv (a_1(t),\ldots,a_T(t))$. *Assume that* $a_u(t) = 0$ *for* $u < t$; *i.e., work does not arrive after its deadline. Let* $r(t) \in \mathbb{R}_+$ *denote the total amount of resource allocated at decision point* t, *and let* $r \equiv (r(1),\ldots,r(T))$. *Let* $q_u(t) \in \mathbb{R}_+$ *denote the amount of work with deadline* u *that is performed at time* t, *and let* $q(t) \equiv (q_1(t),\ldots,q_T(t))$ *and* $q = (q(1),\ldots,q(T))$. *Thus* $(r(t), q(t))$ *denotes the decision made at time* t. *For* (r,q) *to be feasible,* $(r(t), q(t))$ *must satisfy the following for all* t*:*

$$\sum_{u=1}^{T} q_u(t) \leq r(t), \tag{2.2}$$

$$q_u(t) = 0 \quad \textit{for all } u < t, \tag{2.3}$$

$$\sum_{t'=1}^{t} q_t(t') = \sum_{t'=1}^{t} a_t(t'), \tag{2.4}$$

$$\sum_{t'=1}^{t} q_u(t') \leq \sum_{t'=1}^{t} a_u(t') \quad \textit{for all } u. \tag{2.5}$$

Constraint (2.2) *states that the total amount of work performed at time t cannot exceed the amount of work that can be accomplished with $r(t)$ amount of resource. Constraint* (2.3) *states that no work can be performed after its deadline. Constraint* (2.4) *states that all work must be performed by the respective deadlines. Constraint* (2.5) *states that work cannot be performed before it has arrived.*

A more general version of the ORMP stated above incorporates a productivity function $\eta_t(r)$ which represents the amount of work that can be performed at time t with r amount of resource. Thus the ORMP stated above has productivity function $\eta_t(r) = r$ for all t and r. The ORMP with productivity function $\eta_t(r)$ is the same as the ORMP stated above, except that constraint (2.2) *is replaced with*

$$\sum_{u=1}^{T} q_u(t) \quad \leq \quad \eta_t(r(t)). \tag{2.6}$$

It is clear that as long as η_t is nondecreasing for all t, $\mathcal{R}$ has the feasibility monotonicity property. It will be stated clearly which results hold for the ORMP with productivity function $\eta_t(r) = r$ and which results hold for more general productivity functions.

The decision maker has to make a sequence of decisions $(r(t), q(t))$ over time, using information as it becomes available. Because the decision maker has no information about future arrivals, decision $(r(t), q_t(t), \ldots, q_T(t))$ can depend on past arrivals only and not on any arrivals after time t. Thus algorithms are required to be online. Because the objective is to minimize the maximum amount of resource allocated at any time during the planning period, the algorithm evaluation criteria of the ORMP are the same as those of the OMMP. Thus it seems that the ORMP fits into the framework of the OMMP, except that the OMMP includes only a single decision $r(t) \in \mathbb{R}_+$ at each time t, whereas the ORMP includes both resource quantity decision $r(t) \in \mathbb{R}_+$ and resource allocation decision $q(t) \in \mathbb{R}_+^T$ at each time t. However, it is clear that one should give preference to available work with earlier deadlines above work with later deadlines in the allocation q of the chosen amounts of resource r. This decision rule for the allocation of the chosen amounts of resource r is called the *earliest deadline first* rule (EDF). Specifically, the EDF rule works as follows. For any time t and any chosen amount of resource $r(t)$, $q_u(t)$ is determined inductively by $q_u(t) = 0$ for all $u < t$, and

$$q_u(t) \quad = \quad \min\left\{\sum_{\tau=1}^{t} a_u(\tau) - \sum_{\tau=1}^{t-1} q_u(\tau),\ \eta_t(r(t)) - \sum_{v=t}^{u-1} q_v(t)\right\} \tag{2.7}$$

for all $u \geq t$. It is easy to see that for any instance ω and any feasible solution (r, q), the solution (r, q'), where q' denotes the resource allocation decisions according to the EDF rule, is both feasible and has the same objective value as solution (r, q). Because EDF performs at least as well as any other allocation rule, attention is restricted to algorithms that use the EDF rule. Thus a solution is specified by r only, and the ORMP is a special case of the OMMP.

It follows from the definition of the EDF rule that constraints (2.6) (or (2.2)), (2.3), and (2.5) cannot be violated by the EDF rule. Thus the only constraint that can be violated by the EDF rule is (2.4); that is, the algorithm can fail to perform all work by the deadlines, in which case the algorithm is infeasible.

The problem parameter β of interest for the ORMP is the length T of the time horizon. Note that ρ_T^* is nondecreasing in T, because for any instance $\omega_T = (a(1), \ldots, a(T))$ of length T there is an instance $\omega_{T+1} = (0, a(1), \ldots, a(T))$ of length

$T+1$, such that the optimal value with perfect information is the same for both instances, $v^*(\omega_T) = v^*(\omega_{T+1})$, and for any feasible solution $r = (r(1), \ldots, r(T+1))$ for ω_{T+1}, the solution $r' = (r'(1), \ldots, r'(T))$ for ω_T with $r'(t) = r(t+1)$ for all $t \in \{1, \ldots, T\}$ is feasible, and $\max\{r'(1), \ldots, r'(T)\} \leq \max\{r(1), \ldots, r(T+1)\}$.

The version of the ORMP in which all deadlines are equal to the planning horizon T is called the single deadline ORMP, and the version with different deadlines is called the multiple deadline ORMP.

2.3. HLBP. In this section we define the HLBP, which is another interesting problem that is a special case of the OMMP.

Definition 2.4 (HLBP). *Work arrives over time and has to be performed using a set of m machines. The machines can be ordered in a linear hierarchy, that is, the machines can be indexed with the integers 1 through m, so that machine $j \in \{1, \ldots, m\}$ is at least as versatile as any machine $i \in \{1, \ldots, j\}$. Each quantity of work has a specification of the least versatile machine on which the work can be performed. That is, if a quantity of work requires at least machine i for its completion, then the work can be assigned to any one or more than one of machines $i, \ldots, m$. The quantities of work that arrive as well as the specifications of their least versatile machines become known only at the time of arrival. At a given set of decision points, indexed with $t = 1, \ldots, T$, the decision maker decides how to assign to the eligible machines the work that has arrived since the previous decision point. The objective is to minimize the maximum amount of work assigned to any machine. Let $a_i(t) \in \mathbb{R}_+$ denote the amount of work that arrives at time t that requires at least machine i, with $i \in \{1, \ldots, m\}$ and $t \in \{1, \ldots, T\}$, and let $a(t) \equiv (a_1(t), \ldots, a_m(t))$. Let $q_i(t) \in \mathbb{R}_+$ denote the amount of work assigned to machine i at time t, and let $q(t) \equiv (q_1(t), \ldots, q_m(t))$ and $q = (q(1), \ldots, q(T))$. Thus $q(t)$ denotes the decision made at time t. For q to be feasible, $q(t)$ must satisfy the following for all t:*

$$\sum_{j=i}^{m} q_j(t) \quad \geq \quad \sum_{j=i}^{m} a_j(t) \qquad \textit{for all } i. \tag{2.8}$$

Let

$$r_i(t) \quad \equiv \quad \sum_{\tau=1}^{t} q_i(\tau) \tag{2.9}$$

denote the total amount of work assigned to machine i up to time t. Let

$$r(t) \quad \equiv \quad \max\{r_1(t), \ldots, r_m(t)\} \tag{2.10}$$

denote the maximum amount of work assigned to any machine up to time t, and let $r \equiv (r(1), \ldots, r(T))$. Clearly, $\max\{r(1), \ldots, r(T)\} = r(T)$.

As in the OMMP, algorithms are required to be online because the decision maker has no information about future arrivals. Because the objective is to minimize the maximum amount of work assigned to any machine, $\max\{r_1(T), \ldots, r_m(T)\} \equiv r(T) = \max\{r(1), \ldots, r(T)\}$, the algorithm evaluation criteria of the HLBP are the same as those of the OMMP. Similarly to the ORMP, it seems that the HLBP fits into the framework of the OMMP, except that the OMMP describes a decision $r(t) \in \mathbb{R}_+$ at each time t, whereas the HLBP describes a work assignment decision $q(t) \in \mathbb{R}_+^m$ at each time t. However, without loss of optimality one can obtain a decision q from r as follows. Intuitively it is clear that for any chosen value of $r(t)$, which is the maximum amount of work assigned to any machine up to time t, one should assign work to the least versatile machine which qualifies for that work, and which has less than amount $r(t)$ of work assigned to it, until an amount $r(t)$ of work has been assigned to that machine or all the work has been assigned and, if any work remains, one continues to assign it in this fashion. This decision rule for the assignment of work is called the *least versatile first* (LVF) rule. Specifically, the LVF rule works as follows. For any time t and any chosen value of $r(t)$, $q_i(t)$ is determined inductively by

$$(2.11) \qquad q_i(t) \equiv \min\left\{\sum_{j=1}^{i} a_j(t) - \sum_{j=1}^{i-1} q_j(t),\ r(t) - \sum_{\tau=1}^{t-1} q_i(\tau)\right\}$$

for all $i = 1, \ldots, m$. It is easy to see that for any instance ω and any feasible solution q, the solution q' obtained by choosing $r_i(t) \equiv \sum_{\tau=1}^{t} q_i(\tau)$, and then determining q' according to the LVF rule, is both feasible and has as good an objective value as solution q. Thus, for any given r, the LVF rule performs at least as well as any other work assignment rule. Therefore, attention is restricted to algorithms that use the LVF rule so that a solution is specified by r only. With the LVF rule, it is easy to see that $\mathcal{R}$ has the feasibility monotonicity property. Thus we have established that the HLBP is a special case of the OMMP.

Although $r(t)$ is treated as a decision from here on, it is clear that one can assume without loss of optimality that $r(t)$ satisfies (2.10). It follows from the definition of the LVF rule that no work is assigned to a machine for which the machine is not eligible and that $\sum_{\tau=1}^{t} q_i(\tau) \leq r(t)$ for all i and all t. Thus a solution r, with q determined by the LVF rule, is feasible if and only if

$$(2.12) \qquad \sum_{j=1}^{m} q_j(t) = \sum_{j=1}^{m} a_j(t)$$

for all t; that is, the LVF rule assigns all the work. Thus the system (2.8) of Tm constraints can be replaced with the system (2.12) of T constraints.

The problem parameter β of interest for the HLBP is the length T of the time horizon as well as the number m of machines; thus $\beta = (T, m)$. Note that $\rho^*_{T,m}$ is nondecreasing in T and in m because for any instance $\omega_{T,m} = (a(1), \ldots, a(T))$ of length T with m machines there is an instance $\omega_{T+1,m} = (0, a(1), \ldots, a(T))$ of length $T+1$ with m machines, and there is an instance $\omega_{T,m+1} = (a'(1), \ldots, a'(T))$ of length T with $m+1$ machines, with $a'_1(t) = 0$ and $a'_{i+1}(t) = a_i(t)$ for all $i \in \{1, \ldots, m\}$ and all $t \in \{1, \ldots, T\}$, such that $v^*(\omega_{T,m}) = v^*(\omega_{T+1,m}) = v^*(\omega_{T,m+1})$; and for any feasible solution for $\omega_{T+1,m}$ or $\omega_{T,m+1}$ there is a feasible solution for $\omega_{T,m}$ with at least as good an objective value.

3. An optimal algorithm.

In this section we introduce a simple parameterized algorithm, called the α-policy, with parameter α_β and competitive ratio α_β, provided that it is feasible. The intuition behind the α-policy is as follows. Suppose that at time t, the minimum value over all instances $\omega \in \Omega_\beta$ which start with the part ω^t of the instance observed so far, of the optimal value with perfect information, is $v_\beta(\omega^t)$. (It is easy to compute $v_\beta(\omega^t)$ for both the ORMP and the HLBP, as is shown in sections 4.1 and 4.2.) If one wants to choose $r(t)$ in such a way that it is guaranteed that the eventual objective value will not exceed the optimal value with perfect information by more than a factor of α, then one must choose $r(t) \leq$

$\alpha v_\beta(\omega^t)$. However, if one chooses α or $r(t)$ too small, the resulting solution may not be feasible.

We show that with an appropriate choice of parameters α_β, the α-policy has as good a competitive ratio as any other deterministic algorithm, and that under mild conditions an optimal parameter value exists. Hence, the α-policy is optimal.

For any partial instance $\omega^t \in \Omega^t_\beta$, let $\Omega_\beta(\omega^t) \equiv \{\tilde{\omega} \in \Omega_\beta : \tilde{\omega}^t = \omega^t\}$ denote the set of all instances in Ω_β with first t elements equal to ω^t. Let

$$v_\beta(\omega^t) \quad \equiv \quad \inf_{\tilde{\omega} \in \Omega_\beta(\omega^t)} v^*(\tilde{\omega})$$

denote the best value with perfect information over all instances in Ω_β which start with ω^t.

Definition 3.1 (α-policy). *An α-policy is an algorithm $\pi_\alpha \in \Pi^{DO}$, with parameters $\alpha_\beta \in [1, \infty)$ for each $\beta \in \mathbf{B}$, such that for any instance $\omega \in \Omega_\beta$ and any t,*

$$\pi_\alpha(\omega)(t) \quad \equiv \quad \alpha_\beta v_\beta(\omega^t).$$

Recall that, for any instance $\omega \in \Omega_\beta$ and any t, $v_\beta(\omega^t) \le v^*(\omega)$. Thus, for any instance $\omega \in \Omega_\beta$ and any t, $\pi_\alpha(\omega)(t) \le \alpha_\beta v^*(\omega)$, and hence

$$v^{\pi_\alpha}(\omega) \quad \equiv \quad \max\Big\{\pi_\alpha(\omega)(1), \ldots, \pi_\alpha(\omega)(T)\Big\} \quad \le \quad \alpha_\beta v^*(\omega).$$

Therefore, if $\pi_\alpha(\omega)$ is feasible for all $\omega \in \Omega_\beta$, then $\rho^{\pi_\alpha}_\beta \le \alpha_\beta$. It follows from the definition of $v_\beta(\omega^t)$ that if $v^*(\omega) > 0$ and $\pi_\alpha(\omega)$ is feasible for all $\omega \in \Omega_\beta$, then this bound is tight, and thus $\rho^{\pi_\alpha}_\beta = \alpha_\beta$.

Note that the feasibility monotonicity property implies that if $\pi_{\alpha_1}(\omega) \in \mathcal{R}(\omega)$ for some $\omega \in \Omega$ and some $\alpha_1 \in [1, \infty)$, then $\pi_{\alpha_2}(\omega) \in \mathcal{R}(\omega)$ for all $\alpha_2 \ge \alpha_1$.

Next we show that for any algorithm $\pi \in \Pi^{DO}$, there is an α-policy, with appropriate parameters α_β, that performs as well as π.

Theorem 3.2. *For any algorithm $\pi \in \Pi^{DO}$, if $\rho^\pi_\beta < \infty$, then the α-policy π_α with parameter $\alpha_\beta = \rho^\pi_\beta$ achieves the same competitive ratio, $\rho^{\pi_\alpha}_\beta = \rho^\pi_\beta$.*

Proof. Choose parameter $\alpha_\beta = \rho^\pi_\beta$. We show that the α-policy leads to feasible solutions for all instances $\omega \in \Omega_\beta$ by showing that $\pi_\alpha(\omega)(t) \ge \pi(\omega)(t)$ for all $\omega \in \Omega_\beta$ and all t. This is shown by contradiction; suppose that $\pi_\alpha(\omega)(t) < \pi(\omega)(t)$ for some $\omega \in \Omega_\beta$ and some t. Choose any instance $\omega' \in \Omega_\beta(\omega^t)$ such that

$$\begin{aligned} & v^*(\omega') < v_\beta(\omega^t) + \frac{\pi(\omega)(t) - \pi_\alpha(\omega)(t)}{\alpha_\beta} \\ \Rightarrow \quad & \alpha_\beta v^*(\omega') < \alpha_\beta v_\beta(\omega^t) + \pi(\omega)(t) - \pi_\alpha(\omega)(t) \\ & \qquad = \pi(\omega)(t) \quad = \quad \pi(\omega')(t) \\ & \qquad \le v^\pi(\omega'). \end{aligned}$$

The last equality follows because instances ω and ω' have the same history up to time t, and π is an online algorithm. It follows that $v^\pi(\omega') > \rho^\pi_\beta v^*(\omega')$, which contradicts algorithm π having competitive ratio $\rho^\pi_\beta < \infty$. Hence $\pi_\alpha(\omega)(t) \ge \pi(\omega)(t)$ for all $\omega \in \Omega_\beta$ and all t, and thus it follows from feasibility monotonicity and $\rho^\pi_\beta < \infty$ that the α-policy with $\alpha_\beta = \rho^\pi_\beta$ leads to feasible solutions for all $\omega \in \Omega_\beta$. Therefore $\rho^{\pi_\alpha}_\beta \le \alpha_\beta = \rho^\pi_\beta$. Also, $\pi_\alpha(\omega)(t) \ge \pi(\omega)(t)$ for all $\omega \in \Omega_\beta$ and all t implies that $\rho^{\pi_\alpha}_\beta \ge \rho^\pi_\beta$. Thus $\rho^{\pi_\alpha}_\beta = \rho^\pi_\beta$. □

COROLLARY 3.3. *To determine ρ^*_β (and ρ^*), it is sufficient to consider only the α-policy. That is,*

$$\rho^*_\beta \;=\; \inf_{\alpha \geq 1} \rho^{\pi_\alpha}_\beta .$$

Also,

$$\rho^*_\beta \;=\; \inf \left\{ \alpha \geq 1 : \rho^{\pi_\alpha}_\beta < \infty \right\},$$

where $\inf \varnothing = \infty$.

However, the α-policy with parameter $\alpha_\beta = \rho^*_\beta$ may not be feasible for all $\omega \in \Omega_\beta$, in which case there is no optimal algorithm, as stated in Corollary 3.4.

COROLLARY 3.4. *If there exists an algorithm that is optimal for instances in Ω_β, and $\rho^*_\beta < \infty$, then*

1. *ρ^*_β is the least parameter for which the α-policy is feasible for instances in Ω_β, and ρ^*_β is therefore the optimal parameter for the α-policy, and*
2. *the α-policy with parameter $\alpha_\beta = \rho^*_\beta$ is optimal for instances in Ω_β among all deterministic online algorithms.*

Next it is natural to ask under which conditions an optimal algorithm exists, that is, under which conditions the α-policy with $\alpha_\beta = \rho^*_\beta$ is feasible. Proposition 3.5 shows that if $\mathcal{R}(\omega)$ is closed for all $\omega \in \Omega_\beta$, then the set $\{\alpha \geq 1 : \rho^{\pi_\alpha}_\beta < \infty\}$ of feasible α-values is closed, and thus the α-policy with $\alpha_\beta = \rho^*_\beta$ is feasible.

PROPOSITION 3.5. *If $\rho^*_\beta < \infty$ and for some $\omega \in \Omega_\beta$, $\mathcal{R}(\omega)$ is closed, then $\pi_{\rho^*_\beta}(\omega) \in \mathcal{R}(\omega)$. Thus, if $\mathcal{R}(\omega)$ is closed for all $\omega \in \Omega_\beta$, then the α-policy with parameter $\alpha_\beta = \rho^*_\beta$ is feasible for all $\omega \in \Omega_\beta$.*

Proof. Consider $\omega \in \Omega_\beta$. Choose any sequence $\{\alpha_n\}$ such that $\alpha_n > \rho^*_\beta$ for all n and $\alpha_n \to \rho^*_\beta$ as $n \to \infty$. Thus $\pi_{\alpha_n}(\omega) = (\alpha_n v_\beta(\omega^1), \ldots, \alpha_n v_\beta(\omega^T)) \to (\rho^*_\beta v_\beta(\omega^1), \ldots, \rho^*_\beta v_\beta(\omega^T)) = \pi_{\rho^*_\beta}(\omega)$ as $n \to \infty$. It follows from Corollary 3.3 and feasibility monotonicity that $\pi_{\alpha_n}(\omega) \in \mathcal{R}(\omega)$ for all n. Then it follows from $\mathcal{R}(\omega)$ being closed that $\pi_{\rho^*_\beta}(\omega) \in \mathcal{R}(\omega)$. □

4. Optimal competitive ratios.

The results in section 3 are all existential in nature. They do not show how to compute ρ^*_β, and therefore the optimal parameters α_β, for the α-policy. In this section we show how the optimal parameters for the α-policy can be computed, first for the ORMP in section 4.1, then for the HLBP in section 4.2, and then we show how some of these results generalize for the OMMP in section 4.3.

4.1. ORMP.

In this section we investigate the application of the α-policy to the ORMP, including the calculation of the optimal competitive ratios and optimal parameters for the α-policy, ρ^*_T and ρ^*. (The problem parameter β of interest for the ORMP is T.)

Recall that, for a given instance $\omega \in \Omega_T$, the decisions under the α-policy are given by $\pi_\alpha(\omega)(t) \equiv \alpha_T v_T(\omega^t)$. To implement the α-policy, one has to determine the optimal value of α_T, that is, ρ^*_T, as well as $v_T(\omega^t)$. These two issues are addressed next.

4.1.1. Calculation of $\boldsymbol{v_T(\omega^t)}$.

It is easy to compute $v_T(\omega^t) \equiv \inf_{\tilde{\omega} \in \Omega_T(\omega^t)} v^*(\tilde{\omega})$ for the ORMP. The optimal value $v^*(\omega)$ with perfect information can be computed for any ω using Proposition 4.1. Next, for any partial instance ω^t, the best instance

$\tilde{\omega} \in \Omega_T$ that starts with ω^t can be determined, and $v_T(\omega^t)$ can be computed, as shown in Proposition 4.4.

PROPOSITION 4.1. *For any instance $\omega = (a(1), \ldots, a(T))$ of the ORMP with nondecreasing productivity function $\eta_t(r)$, the optimal value $v^*(\omega)$ with perfect information is given by*

$$v^*(\omega) \quad = \quad \inf \left\{ r \geq 0 \,:\, \sum_{t=i}^{j} \eta_t(r) \geq \sum_{t=i}^{j} \sum_{u=t}^{j} a_u(t) \quad \forall\, i,j \in \{1, \ldots, T\}, i \leq j \right\},$$

where $\inf \varnothing = \infty$.

The sum $\sum_{t=i}^{j} \eta_t(r)$ is the total amount of work that can be done between time periods i and j inclusive, while the sum $\sum_{t=i}^{j} \sum_{u=t}^{j} a_u(t)$ is the total amount of work that arrives at or after time i and is due at or before time j. Clearly, for feasibility the former sum must be at least as great as the latter for all pairs i, j. The proof of Proposition 4.1 consists of a straightforward verification that this requirement is not only necessary but also sufficient for feasibility.

COROLLARY 4.2. *For any instance $\omega = (a(1), \ldots, a(T))$ of the ORMP with productivity function $\eta_t(r) = r$, the optimal value $v^*(\omega)$ with perfect information is given by*

$$v^*(\omega) \quad = \quad \max_{\{i,j \in \{1,\ldots,T\} \,:\, i \leq j\}} \left\{ \frac{1}{j-i+1} \sum_{t=i}^{j} \sum_{u=t}^{j} a_u(t) \right\}.$$

For any $\omega = (a(1), \ldots, a(T)), \omega' = (a'(1), \ldots, a'(T))$, we denote $\omega \leq \omega'$ if $a_u(t) \leq a'_u(t)$ for all t and u.

COROLLARY 4.3. *The optimal value $v^*(\omega)$ with perfect information of the ORMP with productivity function $\eta_t(r)$ is nondecreasing; that is, for any $\omega, \omega' \in \Omega_T$ with $\omega \leq \omega'$, it holds that $v^*(\omega) \leq v^*(\omega')$.*

It follows from Corollary 4.3 that for any partial instance $\omega^t = (a(1), \ldots, a(t))$, instance $(a(1), \ldots, a(t), 0, \ldots, 0) \in \Omega_T$ has the best optimal value $v^*(\tilde{\omega})$ with perfect information among all $\tilde{\omega} \in \Omega_T$ that start with ω^t. This result makes it easy to compute $v_T(\omega^t) \equiv \inf_{\omega \in \Omega_T(\omega^t)} v^*(\omega)$.

PROPOSITION 4.4. *For any partial instance $\omega^t = (a(1), \ldots, a(t))$ of the ORMP with productivity function $\eta_t(r)$, $v_T(\omega^t) = v^*(a(1), \ldots, a(t), 0, \ldots, 0)$. Specifically, for the ORMP with nondecreasing productivity function $\eta_t(r)$,*

$$v_T(\omega^t) = \inf \left\{ r \geq 0 \,:\, \sum_{\tau=i}^{j} \eta_\tau(r) \geq \sum_{\tau=i}^{\min\{j,t\}} \sum_{u=\tau}^{j} a_u(\tau) \quad \forall\, i \in \{1, \ldots, t\}, j \in \{i, \ldots, T\} \right\}$$

and for the ORMP with productivity function $\eta_t(r) = r$,

$$v_T(\omega^t) \quad = \quad \max_{\{i \in \{1,\ldots,t\}, j \in \{i,\ldots,T\}\}} \left\{ \frac{1}{j-i+1} \sum_{\tau=i}^{\min\{j,t\}} \sum_{u=\tau}^{j} a_u(\tau) \right\}.$$

Corollary 4.3 and Proposition 4.4 lead to the following result.

COROLLARY 4.5. *For any partial instances $\omega_1^t, \omega_2^t \in \Omega_T^t$ of the ORMP with productivity function $\eta_t(r)$, with $\omega_1^t \leq \omega_2^t$, it holds that $v_T(\omega_1^t) \leq v_T(\omega_2^t)$. Specifically,* *for any instance $\omega \in \Omega_T$, $v_T(\omega^t)$ is nondecreasing in t. It follows that for any α-policy and any instance $\omega \in \Omega_T$, $\pi_\alpha(\omega)(t) \equiv \alpha_T v_T(\omega^t)$ is nondecreasing in t.*

Thus the α-policy takes full advantage of resource that has already been allocated, since it allocates at least as much at time $t+1$ as at time t.

4.1.2. Optimal parameter values. Next we address the determination of the optimal value of α_T. For any $\omega \in \Omega$, the feasible region $\mathcal{R}(\omega)$ is determined by linear constraints (2.2), (2.3), (2.4), and (2.5). Thus $\mathcal{R}(\omega)$ is a polyhedron and is closed. Therefore it follows from Corollary 3.4 and Proposition 3.5 that the optimal value of α_T is ρ_T^* and that the α-policy with parameter $\alpha_T = \rho_T^*$ is feasible and optimal among all deterministic online algorithms. (It is shown in section 5.2, Proposition 5.3, that if the productivity function η_t is concave for all t, then the α-policy with parameters $\alpha_T = \rho_T^*$ is optimal among all randomized online algorithms.)

First we show that, to determine ρ_T^* for the multiple deadline ORMP, it is sufficient to consider the single deadline ORMP. This result simplifies the calculation of ρ_T^*.

We introduce the following notation to distinguish between the single deadline ORMP and the multiple deadline ORMP. Consider the function $\theta : \Omega \mapsto \Omega$ that postpones the deadlines of all work to the end of the time horizon. That is, for any instance $\omega = (a_1(1), a_2(1), \ldots, a_T(T)) \in \Omega_T$ of the multiple deadline ORMP, $\theta(\omega) = (a'_1(1), a'_2(1), \ldots, a'_T(T))$, where

$$a'_u(t) \quad = \quad \begin{cases} 0 & \text{if} \quad u < T, \\ \sum_{v=t}^{T} a_v(t) & \text{if} \quad u = T. \end{cases}$$

When considering the single deadline ORMP, we simplify the notation slightly by letting $a(t) \in \mathbb{R}_+$ denote the amount of work arriving at time t (with deadline T) and letting $q(t) \in \mathbb{R}_+$ denote the amount of work performed at time t.

Recall that Ω_T denotes the set of instances of length T for the multiple deadline ORMP. Note that the set of instances of length T for the single deadline ORMP is given by $\theta(\Omega_T) \subset \Omega_T$. Also note that, for any instance $\theta(\omega)$, the set of feasible solutions of the single deadline ORMP is the same as the set $\mathcal{R}(\theta(\omega))$ of feasible solutions for the same instance of the multiple deadline ORMP. In addition, for any instance $\theta(\omega)$, the optimal value with perfect information of the single deadline ORMP is equal to the optimal value $v^*(\theta(\omega))$ with perfect information for the same instance of the multiple deadline ORMP. The following corollary for the single deadline ORMP follows from Proposition 4.1 and Corollary 4.2.

Corollary 4.6. *For any instance $\omega = (a(1), \ldots, a(T))$ of the single deadline ORMP, the optimal value $v^*(\omega)$ with perfect information is given by the following: With nondecreasing productivity function $\eta_t(r)$,*

$$v^*(\omega) \quad = \quad \inf \left\{ r \geq 0 \, : \, \sum_{t=i}^{T} \eta_t(r) \geq \sum_{t=i}^{T} a(t) \ \ \forall \ i \in \{1, \ldots, T\} \right\}$$

and with productivity function $\eta_t(r) = r$,

$$v^*(\omega) \quad = \quad \max_{\{i \in \{1,\ldots,T\}\}} \left\{ \frac{1}{T-i+1} \sum_{t=i}^{T} a(t) \right\}.$$

It follows from Proposition 4.4 that, for any partial instance $\theta(\omega)^t$, the best value with perfect information, over all instances in $\theta(\Omega_T)$ of the single deadline ORMP that start with $\theta(\omega)^t$, is equal to the best value $v_T(\theta(\omega)^t)$ with perfect information, over all instances in Ω_T of the multiple deadline ORMP that start with $\theta(\omega)^t$.

COROLLARY 4.7. *For any partial instance $\omega^t = (a(1),\dots,a(t))$ of the single deadline ORMP with nondecreasing productivity function $\eta_t(r)$,*

$$v_T(\omega^t) \quad = \quad \inf\left\{ r \geq 0 \,:\, \sum_{\tau=i}^{T} \eta_\tau(r) \geq \sum_{\tau=i}^{t} a(\tau) \;\;\forall\, i \in \{1,\dots,t\}\right\}$$

and for the single deadline ORMP with productivity function $\eta_t(r) = r$,

$$v_T(\omega^t) \quad = \quad \max_{\{i\in\{1,\dots,t\}\}} \left\{ \frac{1}{T-i+1} \sum_{\tau=i}^{t} a(\tau) \right\}.$$

Thus, for any instance $\theta(\omega) \in \theta(\Omega_T)$ of the single deadline ORMP, the α-policy prescribes exactly the same decisions for the single deadline ORMP as for the multiple deadline ORMP:

$$\pi_\alpha(\theta(\omega))(t) \quad \equiv \quad \alpha_T v_T(\theta(\omega)^t).$$

Also note that, for all $\omega \in \Omega_T$, $v^*(\theta(\omega)) \leq v^*(\omega)$, and $v_T(\theta(\omega)^t) \leq v_T(\omega^t)$ for all $t \in \{1,\dots,T\}$.

We want to show that ρ_T^* is the same for the multiple deadline ORMP and the single deadline ORMP, and thus that the optimal parameter α_T is the same for the multiple deadline ORMP and the single deadline ORMP. That is, we want to show that

$$\begin{aligned}\rho_T^* &= \inf_{\alpha\geq 1} \inf\left\{\rho \geq 1 \,:\, v^{\pi_\alpha}(\omega) \leq \rho v^*(\omega) \;\;\forall\, \omega \in \Omega_T\right\} \\ &= \inf_{\alpha\geq 1} \inf\left\{\rho \geq 1 \,:\, v^{\pi_\alpha}(\omega) \leq \rho v^*(\omega) \;\;\forall\, \omega \in \theta(\Omega_T)\right\}.\end{aligned}$$

The first equality follows from Corollary 3.3, and it remains to establish the second equality. We do that by recalling that ρ_T^* is nondecreasing in T, and we show in Theorem 4.9 that if $\alpha \in (\rho_{T-1}^*, \rho_T^*)$, so that the α-policy with parameter α is infeasible for some instance $\omega \in \Omega_T$, then the α-policy with parameter α is also infeasible for instance $\theta(\omega) \in \theta(\Omega_T)$. Thereafter, Theorem 4.10 establishes the second equality, and thus that ρ_T^* is the same for the multiple deadline ORMP and the single deadline ORMP. The performance of the α-policy on instances in Ω_{T-1} enters into the evaluation, and we introduce the following notation for that purpose.

Consider the function $\vartheta : \Omega \mapsto \Omega$ that removes all work with deadline equal to T. That is, for any instance $\omega = (a_1(1), a_2(1), \dots, a_T(T)) \in \Omega_T$, $\vartheta(\omega) = (a_1''(1), a_2''(1), \dots, a_{T-1}''(T-1))$, where $a_u''(t) = a_u(t)$ for all $t \in \{1,\dots,T-1\}$ and $u \in \{t,\dots,T-1\}$. Note that for any $\omega \in \Omega_T$, $\vartheta(\omega) \in \Omega_{T-1}$. Thus, for any $\omega \in \Omega_T$, the α-policy prescribes decisions $\pi_\alpha(\vartheta(\omega))(t) = \alpha_{T-1} v_{T-1}(\vartheta(\omega)^t)$ for $\vartheta(\omega) \in \Omega_{T-1}$. For any instance $\omega = (a_1(1), a_2(1), \dots, a_T(T)) \in \Omega_T$, let $\tilde\omega = (\tilde a_1(1), \tilde a_2(1), \dots, \tilde a_T(T)) \in \Omega_T$ be given by $\tilde a_u(t) = a_u(t)$ for all $t \in \{1,\dots,T-1\}$ and $u \in \{t,\dots,T-1\}$, and $\tilde a_T(t) = 0$ for all $t \in \{1,\dots,T\}$. Note that $\tilde\omega \leq \omega$. Thus, if $\eta_t(r) \geq 0$ for all $r, t \geq 0$, then $v^*(\vartheta(\omega)) = v^*(\tilde\omega) \leq v^*(\omega)$, and $v_{T-1}(\vartheta(\omega)^t) = v_T(\tilde\omega^t) \leq v_T(\omega^t)$ for all $t \in \{1,\dots,T-1\}$.

For any $r \in \mathbb{R}_+^T$, let r^{T-1} denote the first $T-1$ components of r.

Lemma 4.8. *For the ORMP with productivity function $\eta_t(r)$, any $\omega \in \Omega_T$, and any $r \in \mathbb{R}^T_+$, $r \in \mathcal{R}(\theta(\omega))$, and $r^{T-1} \in \mathcal{R}(\vartheta(\omega))$ imply that $r \in \mathcal{R}(\omega)$.*

Proof. Consider any instance $\omega = (a_1(1), a_2(1), \ldots, a_T(T))$, $\theta(\omega) = (a'_1(1), a'_2(1), \ldots, a'_T(T))$, and $\vartheta(\omega) = (a''_1(1), a''_2(1), \ldots, a''_{T-1}(T-1))$. Consider any r such that $r \in \mathcal{R}(\theta(\omega))$ and $r^{T-1} \in \mathcal{R}(\vartheta(\omega))$. As usual, available work is performed in EDF order. Let $w_u(t) \equiv a_u(1) - q_u(1) + a_u(2) - q_u(2) + \cdots + a_u(t)$, $w'_T(t) \equiv a'_T(1) - q'_T(1) + a'_T(2) - q'_T(2) + \cdots + a'_T(t)$, and $w''_u(t) \equiv a''_u(1) - q''_u(1) + a''_u(2) - q''_u(2) + \cdots + a''_u(t)$ denote the remaining amount of work at time t with deadline u for instances ω, $\theta(\omega)$, and $\vartheta(\omega)$, respectively.

It is shown by induction on t that $w_u(t) = w''_u(t)$ and $q_u(t) = q''_u(t)$ for all $t = 1, \ldots, T-1$, $u = t, \ldots, T-1$, and $w'_T(t) = \sum_{u=t}^T w_u(t)$ and $q'_T(t) = \sum_{u=t}^T q_u(t)$ for all $t = 1, \ldots, T$. The hypothesis holds for $t = 1$. Suppose that the hypothesis holds for t. Note that $q_t(t) = q''_t(t) = w''_t(t) = w_t(t)$ from the assumption that $r^{T-1} \in \mathcal{R}(\vartheta(\omega))$. Then $w_u(t+1) = w_u(t) - q_u(t) + a_u(t+1) = w''_u(t) - q''_u(t) + a''_u(t+1) = w''_u(t+1)$ for all $u = t+1, \ldots, T-1$. Because available work is performed in EDF order, $q_u(t+1) = q''_u(t+1)$ for all $u = t+1, \ldots, T-1$. Also, $w'_T(t+1) = w'_T(t) - q'_T(t) + a'_T(t+1) = \sum_{u=t}^T w_u(t) - \sum_{u=t}^T q_u(t) + \sum_{u=t+1}^T a_u(t+1) = \sum_{u=t+1}^T (w_u(t) - q_u(t) + a_u(t+1)) = \sum_{u=t+1}^T w_u(t+1)$, and the hypothesis has been established.

Recall that the EDF rule ensures that constraints (2.6), (2.3), and (2.5) are satisfied. Thus, to show that $r \in \mathcal{R}(\omega)$, it remains to verify that solution r satisfies (2.4) for instance ω. From the assumption that $r^{T-1} \in \mathcal{R}(\vartheta(\omega))$, it follows that r^{T-1} satisfies (2.4) for $\vartheta(\omega)$, and thus it follows from the hypothesis established above that r satisfies (2.4) for ω, for all $t = 1, \ldots, T-1$. It remains to be shown that r satisfies (2.4) for ω at $t = T$. From the assumption that $r \in \mathcal{R}(\theta(\omega))$ and the induction hypothesis, it follows that $w_T(T) = w'_T(T) \le \eta_T(r(T))$, and thus r satisfies (2.4) for ω at $t = T$. □

Theorem 4.9. *Suppose that the productivity function $\eta_t(r)$ is nondecreasing. If the α-policy with parameter α gives an infeasible solution for some instance ω and a feasible solution for instance $\vartheta(\omega)$, then the α-policy with parameter α gives an infeasible solution for instance $\theta(\omega)$.*

Proof. Consider any $\omega \in \Omega_T$. Recall that $v_{T-1}(\vartheta(\omega)^t) \le v_T(\omega^t)$, and thus $\pi_\alpha(\vartheta(\omega))(t) = \alpha v_{T-1}(\vartheta(\omega)^t) \le \alpha v_T(\omega^t) = \pi_\alpha(\omega)(t)$, for all $t \in \{1, \ldots, T-1\}$. From the assumption that $\pi_\alpha(\vartheta(\omega)) \in \mathcal{R}(\vartheta(\omega))$ and from feasibility monotonicity, it follows that $\pi_\alpha(\omega)^{T-1} \in \mathcal{R}(\vartheta(\omega))$. From the assumption that $\pi_\alpha(\omega) \notin \mathcal{R}(\omega)$ and the contrapositive of the result in Lemma 4.8, it follows that $\pi_\alpha(\omega) \notin \mathcal{R}(\theta(\omega))$. Recall that $v_T(\theta(\omega)^t) \le v_T(\omega^t)$, and thus $\pi_\alpha(\theta(\omega))(t) = \alpha v_T(\theta(\omega)^t) \le \alpha v_T(\omega^t) = \pi_\alpha(\omega)(t)$, for all $t \in \{1, \ldots, T\}$. It follows from feasibility monotonicity and $\pi_\alpha(\omega) \notin \mathcal{R}(\theta(\omega))$ that $\pi_\alpha(\theta(\omega)) \notin \mathcal{R}(\theta(\omega))$. Thus the α-policy with parameter α gives an infeasible solution for instance $\theta(\omega)$. □

Next, Theorem 4.10 establishes that ρ^*_T is the same for the multiple deadline ORMP and the single deadline ORMP.

Theorem 4.10. *Suppose that the productivity function $\eta_t(r)$ is nondecreasing. Then*

$$\rho^*_T \quad = \quad \inf_{\alpha \ge 1} \inf \left\{ \rho \ge 1 \; : \; v^{\pi_\alpha}(\omega) \le \rho v^*(\omega) \quad \forall \; \omega \in \theta(\Omega_T) \right\}.$$

That is, to determine ρ^*_T (and ρ^*), it is sufficient to consider only the α-policy and only the instances in $\theta(\Omega_T)$.

Proof. Corollary 3.3 established that

$$\rho_T^* \quad = \quad \inf_{\alpha \geq 1} \inf \left\{ \rho \geq 1 \, : \, v^{\pi_\alpha}(\omega) \leq \rho v^*(\omega) \ \ \forall \, \omega \in \Omega_T \right\}.$$

Let

$$\rho_T^\theta \quad \equiv \quad \inf_{\alpha \geq 1} \inf \left\{ \rho \geq 1 \, : \, v^{\pi_\alpha}(\omega) \leq \rho v^*(\omega) \ \ \forall \, \omega \in \theta(\Omega_T) \right\}.$$

Note that, because $\theta(\Omega_T) \subset \Omega_T$, it follows that $\rho_T^\theta \leq \rho_T^*$. Also note that, similar to ρ_T^*, ρ_T^θ is nondecreasing in T. We show by induction on T that $\rho_T^\theta = \rho_T^*$. For $T = 1$, $\theta(\Omega_1) = \Omega_1$, and thus $\rho_1^\theta = \rho_1^*$. Suppose that $\rho_{T-1}^\theta = \rho_{T-1}^*$. Then $\rho_{T-1}^* = \rho_{T-1}^\theta \leq \rho_T^\theta \leq \rho_T^*$. Thus, if $\rho_{T-1}^* = \rho_T^*$, then $\rho_T^\theta = \rho_T^*$.

Otherwise, if $\rho_{T-1}^* < \rho_T^*$, then consider any $\alpha \in (\rho_{T-1}^*, \rho_T^*)$. Then there exists an $\omega \in \Omega_T$ such that the α-policy with parameter α gives an infeasible solution for instance ω, i.e., $\pi_\alpha(\omega) \notin \mathcal{R}(\omega)$. Also, the α-policy with parameter α gives a feasible solution for instance $\vartheta(\omega)$, i.e., $\pi_\alpha(\vartheta(\omega)) \in \mathcal{R}(\vartheta(\omega))$, because $\alpha > \rho_{T-1}^*$ and $\vartheta(\omega) \in \Omega_{T-1}$. Then it follows from Theorem 4.9 that $\pi_\alpha(\theta(\omega)) \notin \mathcal{R}(\theta(\omega))$, and thus $v^{\pi_\alpha}(\theta(\omega)) = \infty$. Hence $\{\rho \geq 1 : v^{\pi_\alpha}(\omega) \leq \rho v^*(\omega) \ \forall \, \omega \in \theta(\Omega_T)\} = \varnothing$. Thus by feasibility monotonicity $\inf_{\alpha < \rho_T^*} \inf\{\rho \geq 1 : v^{\pi_\alpha}(\omega) \leq \rho v^*(\omega) \ \forall \, \omega \in \theta(\Omega_T)\} = \infty$. Next consider any $\alpha > \rho_T^*$. It follows from Theorem 3.2 and feasibility monotonicity that $v^{\pi_\alpha}(\omega) = \alpha v^*(\omega)$ for all $\omega \in \Omega_T$, and thus for all $\omega \in \theta(\Omega_T) \subset \Omega_T$. Hence, noting that $v^*(\omega) > 0$ for some $\omega \in \theta(\Omega_T)$, it follows that $\inf\{\rho \geq 1 : v^{\pi_\alpha}(\omega) \leq \rho v^*(\omega) \ \forall \, \omega \in \theta(\Omega_T)\} = \alpha$. Thus $\rho_T^\theta = \inf_{\alpha > \rho_T^*} \inf\{\rho \geq 1 : v^{\pi_\alpha}(\omega) \leq \rho v^*(\omega) \ \forall \, \omega \in \theta(\Omega_T)\} = \inf_{\alpha > \rho_T^*} \alpha = \rho_T^*$. □

Next we show how ρ_T^* is given by the optimal value of a linear program for the ORMP with a linear productivity function.

Theorem 4.10 simplifies the calculation of ρ_T^* by establishing that it is sufficient to consider the single deadline ORMP. Theorem 4.12 further simplifies the calculation of ρ_T^* by establishing that the parameter α_T is too small if and only if there exists an instance $\omega = (a(1), \ldots, a(T))$ such that the total amount of resource allocated under the α-policy, $\alpha_T \sum_{t=1}^T v_T(\omega^t)$, is less than the total amount of work to be performed, $\sum_{t=1}^T a(t)$. Such instances can be identified with a linear program, as shown later. Theorem 4.12 follows directly from Lemma 4.11.

LEMMA 4.11. *Suppose that the α-policy with parameter α_T gives an infeasible solution with instance $\omega' = (a'(1), \ldots, a'(T)) \in \Omega_T$. Then there exists an instance $\omega = (a(1), \ldots, a(T))$ such that*

$$\alpha_T \sum_{t=1}^T v_T(\omega^t) \quad < \quad \sum_{t=1}^T a(t). \tag{4.1}$$

Proof. Let $q'(t)$ denote the amount of work performed at time t on instance ω'. Let $\tau \equiv \max\{t : \sum_{l=1}^t q'(l) = \sum_{l=1}^t a'(l)\}$. Since ω' is infeasible, $\tau < T$. For $t > \tau$, $q'(t) = \alpha_T v_T(\omega'^t)$, so if $\tau = 0$, we have obtained the desired inequality with $\omega = \omega'$.

If $\tau > 0$, then the part of instance ω' before τ does not contribute to the solution being infeasible. Define $\omega = (a(1), \ldots, a(T))$ by

$$a(t) \quad = \quad \left\{ \begin{array}{cl} 0 & \text{if} \quad t \leq \tau, \\ a'(t) & \text{if} \quad t > \tau. \end{array} \right.$$

Note that $\omega \leq \omega'$, so $v_T(\omega^t) \leq v_T(\omega'^t)$ for all t.

Since $\sum_{t=1}^{T} q'(t) < \sum_{t=1}^{T} a'(t)$ and $\sum_{t=1}^{\tau} q'(t) = \sum_{t=1}^{\tau} a'(t)$, it follows that $\sum_{t=\tau+1}^{T} q'(t) < \sum_{t=\tau+1}^{T} a'(t)$. Thus

$$\alpha_T \sum_{t=\tau+1}^{T} v_T(\omega^t) \;\leq\; \alpha_T \sum_{t=\tau+1}^{T} v_T(\omega'^t) \;=\; \sum_{t=\tau+1}^{T} q'(t) \;<\; \sum_{t=\tau+1}^{T} a'(t) \;=\; \sum_{t=\tau+1}^{T} a(t).$$

For $t \leq \tau$, it follows from the definition of ω that $v_T(\omega^t) = a(t) = 0$. Thus, $\alpha_T \sum_{t=1}^{T} v_T(\omega^t) < \sum_{t=1}^{T} a(t)$. □

THEOREM 4.12. *Parameter α_T for the α-policy is too small ($\alpha_T < \rho_T^*$) if and only if there exists an instance $\omega = (a(1), \ldots, a(T))$ such that $\alpha_T \sum_{t=1}^{T} v_T(\omega^t) < \sum_{t=1}^{T} a(t)$. Also,*

$$\rho_T^* = \inf\left\{\rho \in [1,\infty) \,:\, \rho \sum_{t=1}^{T} \max_{\{i\in\{1,\ldots,t\}\}} \left\{\frac{1}{T-i+1}\sum_{j=i}^{t} a(j)\right\} \geq \sum_{t=1}^{T} a(t) \;\; \forall\ (a(1),\ldots,a(T)) \in \mathbb{R}_+^T \right\}. \tag{4.2}$$

The α-policy provides the optimal solution with perfect information for the zero instance $\omega = (0, \ldots, 0)$. Thus, to determine ρ_T^*, one can restrict attention to instances $(a(1), \ldots, a(T)) \in \mathbb{R}_+^T$ for which $\sum_{t=1}^{T} a(t) > 0$. It follows from (4.2) that ρ_T^* is determined by instances $(a(1), \ldots, a(T)) \in \mathbb{R}_+^T$ that minimize

$$\sum_{t=1}^{T} \max_{\{i\in\{1,\ldots,t\}\}} \left\{\frac{1}{T-i+1}\sum_{j=i}^{t} \frac{a(j)}{\sum_{k=1}^{T} a(k)}\right\}.$$

It follows that one can restrict attention to instances $(a(1), \ldots, a(T)) \in \mathbb{R}_+^T$ for which $\sum_{t=1}^{T} a(t) = 1$.

COROLLARY 4.13. *The optimal competitive ratio ρ_T^* for the ORMP can be calculated by solving the following linear program (LP) with decision variables $a(t)$ and $x(t)$, $t \in \{1, \ldots, T\}$:*

$$\begin{aligned}
\text{(LP)}\quad & \textit{Minimize} \sum_{t=1}^{T} x(t) \\
& \textit{subject to} \sum_{t=1}^{T} a(t) \;=\; 1, \\
\text{(4.3)}\quad & x(t) \;\geq\; \frac{1}{T-i+1}\sum_{j=i}^{t} a(j) \;\; \forall\ t \in \{1,\ldots,T\},\ \forall\ i \in \{1,\ldots,t\}, \\
& a(t) \;\geq\; 0\ \forall\ t \in \{1,\ldots,T\}.
\end{aligned}$$

Proof. In an optimal solution (a^*, x^*) of the LP, $a^* = (a^*(1), \ldots, a^*(T))$ represents a worst-case instance of length T for the α-policy, and each $x^*(t)$ represents the corresponding value of $v_T(a^{*t})$. Also, it follows from (4.2) that

$$\rho_T^* \;=\; \inf\left\{\rho \in [1,\infty) \,:\, \rho \sum_{t=1}^{T} x^*(t) \geq 1\right\}.$$

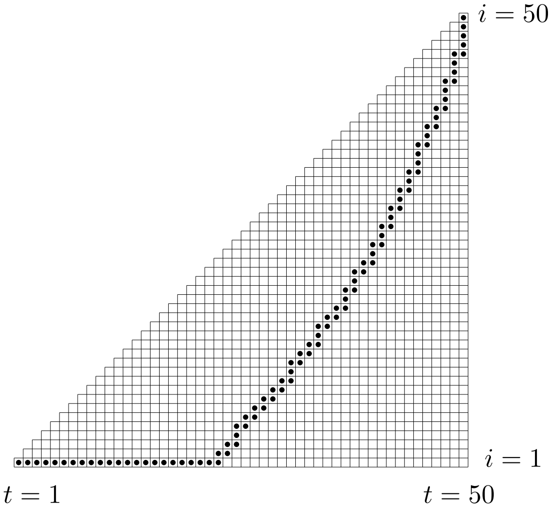


FIG. 4.1. *Incidence of active constraints at optimality for $T = 50$.*

That is, $\rho_T^* = \max\{1, 1/\sum_{t=1}^T x^*(t)\}$. Actually, $\rho_T^* = 1/\sum_{t=1}^T x^*(t)$, as shown next. It is easily checked that (a, x) with $a(t) = 1/T$ and $x(t) = t/T^2$ is a feasible solution for the LP. Thus $\sum_{t=1}^T x^*(t) \leq \sum_{t=1}^T t/T^2 = (1 + 1/T)/2$. Hence $\rho_T^* = 1/\sum_{t=1}^T x^*(t) \geq 2/(1 + 1/T) \geq 1$. □

There are $2T$ decision variables, but quadratically many constraints, because of the $T(T + 1)/2$ allocation constraints (4.3). As the number of periods T increases, working with the full LP becomes prohibitive, in terms of memory requirements as well as computation times. Fortunately, most of the allocation constraints are inactive at an optimal solution, which suggests that a cutting plane method may provide an effective solution approach. To determine a good initial set of constraints, we analyzed the active allocation constraints for the instance with $T = 50$. In Figure 4.1, each grid cell (t, i) corresponds to an allocation constraint from $(4.3{:}t, i)$, and each of the black dots shows an active allocation constraint at optimality.

Let $\gamma(t) \equiv \arg\max_{\{i\in\{1,\ldots,t\}\}}\{\sum_{j=i}^t a(j)/(T - i + 1)\}$. Then the black dots in Figure 4.1 also can be viewed as a map of $\gamma(t)$ versus t. (Note that $\gamma(t)$ is a set valued function.) Although the precise pattern of $\gamma(t)$ versus t is quite complicated, it does follow a crude pattern. For $1 \leq t \leq \delta T \approx T/3$, $\gamma(t) = 1$. This means that for the first approximately $T/3$ time periods t, the optimum value $v_T(\omega^t)$ of the truncated instance ω^t is the amount of resource required to do all the work that has arrived since the first time period. For larger t, the slightly smaller amount of work that has arrived since a later time period $\gamma(t)$, divided by the smaller value $T - \gamma(t) + 1$, provides a larger bound on the required resource. As $t \to T$, the value of $\gamma(t) \to T$ also. This means that in a worst-case instance, so much work arrives in the last time periods that the amount of resource required is determined by the few most recent work arrivals only.

By estimating δ and lower and upper bounds on $\gamma(t)$, we extrapolated the pattern in Figure 4.1 to larger values of T and thereby identified relatively small subsets of allocation constraints, which we hoped would contain all the active allocation constraints for an optimal solution. We solved the relaxed LP. Then we checked the remaining allocation constraints to see if our guess was correct. If not, we added

Table 4.1
Values of ρ_T^* for the ORMP.

T	ρ_T^*
1	1
2	4/3
3	3/2
4	44/27
5	1.71329
6	1.77778
7	1.82765
8	1.86880
9	1.90547
10	1.93576
25	2.14951
50	2.26470
75	2.31800
100	2.35061
200	2.41585
300	2.44663
400	2.46592
500	2.47956
750	2.501833

all the violated allocation constraints that were initially left out, and repeated the procedure. For $T \leq 200$, no cut generation was necessary, but for $T > 200$, several rounds of cut generation were needed prior to finding the optimal solution.

Table 4.1 gives the values of ρ_T^* for some values of T, as determined by the LP.

4.1.3. Asymptotic behavior. Sometimes it is interesting to determine $\rho^* = \sup_{\beta \in \mathbf{B}} \rho_\beta^*$. Thus, for the ORMP, we are interested in $\rho^* = \sup_{T \in \mathbb{Z}_+} \rho_T^*$. As pointed out in section 2.2, ρ_T^* is nondecreasing in T, and thus $\rho^* = \lim_{T \to \infty} \rho_T^*$. It is convenient to study ρ^* with a continuous time model for the ORMP, as in Kleywegt et al. [14]. Also, by Theorem 4.10, to determine ρ^*, it is sufficient to consider the single deadline version of the ORMP.

In the continuous time model, $t \in [0,1]$. An instance is given by an integrable function $a : [0,1] \mapsto [0,\infty)$, with $a(t)$ denoting the rate at which work (with deadline 1) arrives at time t. Let Ω denote the set of all such instances. Thus $A(t) \equiv \int_0^t a(\tau)\, d\tau$ is the total amount of work that has arrived by time t. A solution is given by an integrable function $r : [0,1] \mapsto [0,\infty)$, with $r(t)$ denoting the rate at which resources are allocated at time t. Let $W(t)$ denote the amount of unfinished work at time t. Thus W satisfies the differential equation

$$\frac{dW}{dt}(t) \quad = \quad \begin{cases} a(t) - r(t) & \text{if} \quad W(t) > 0, \\ \max\{a(t) - r(t),\, 0\} & \text{if} \quad W(t) = 0 \end{cases}$$

with boundary condition $W(0) = 0$. A solution r is feasible for an instance a if $W(1) = 0$, for which it is necessary that $\int_0^1 a(t)\, dt \leq \int_0^1 r(t)\, dt$. The value of a solution r for an instance a is $\sup_{t \in [0,1]} r(t)$ if r is feasible for a, and the value is ∞ if r is not feasible for a. Algorithms are defined similar to those for the discrete time ORMP. It is easy to verify that, for any instance a, the set $\mathcal{R}(a)$ of feasible solutions is convex, and it will follow from Theorem 5.2 that it is sufficient to restrict attention to deterministic algorithms.

For any instance a, the optimal value $v^*(a)$ with perfect information is given by

$$v^*(a) \quad = \quad \sup_{\gamma \in [0,1)} \frac{1}{1-\gamma} \int_\gamma^1 a(t)\, dt.$$

As before, a^t denotes the instance a truncated at time t, and the optimal value $v_\infty(a^t)$ with perfect information among all completions of truncated instance a^t is given by

$$v_\infty(a^t) \quad = \quad \sup_{\gamma \in [0,t)} \frac{1}{1-\gamma} \int_\gamma^t a(x)\, dx \quad = \quad \sup_{\gamma \in [0,t)} \frac{A(t) - A(\gamma)}{1-\gamma} \quad \equiv \quad \sup_{\gamma \in [0,t)} f_t(\gamma).$$

The α-policy allocates resource at rate

$$r^\alpha(t) \quad = \quad \alpha v_\infty(a^t). \tag{4.4}$$

For the α-policy with parameter α to be feasible, it is necessary that

$$\int_0^1 a(t)\, dt \quad \leq \quad \int_0^1 r^\alpha(t)\, dt \quad = \quad \alpha \int_0^1 v_\infty(a^t)\, dt$$

for all instances a. Suppose the α-policy with parameter α_∞ gives an infeasible solution with instance a', that is, $W'(1) > 0$. Let $\tau \equiv \sup\{t \in [0,1] \,:\, W'(t) = 0\}$. Let

$$a(t) \quad = \quad \begin{cases} 0 & \text{if} \quad t \in [0,\tau), \\ a'(t) & \text{if} \quad t \in [\tau, 1]. \end{cases}$$

Then, similar to Lemma 4.11, it follows that

$$\alpha_\infty \int_0^1 v_\infty(a^t)\, dt \quad < \quad \int_0^1 a(t)\, dt.$$

Thus, for the α-policy with parameter α to be feasible, it is necessary and sufficient that

$$\int_0^1 a(t)\, dt \quad \leq \quad \alpha \int_0^1 v_\infty(a^t)\, dt \tag{4.5}$$

for all instances a. Therefore

$$\rho^* \quad = \quad \inf \left\{ \rho \in [1, \infty) \,:\, \rho \int_0^1 \sup_{\gamma \in [0,t)} \frac{1}{1-\gamma} \int_\gamma^t a(y)\, dy\, dt \geq \int_0^1 a(t)\, dt \;\; \forall\, a \in \Omega \right\}. \tag{4.6}$$

Thus ρ^* can be calculated by solving the following optimal control problem with integrable controls $a(t)$ and $x(t)$, $t \in [0,1]$:

$$\begin{aligned}
\text{Minimize} \quad & \int_0^1 x(t)\, dt \\
\text{subject to} \quad & \int_0^1 a(t)\, dt = 1 \\
& x(t) \geq \frac{1}{1-\gamma} \int_\gamma^t a(y)\, dy \qquad \forall\, t \in [0,1], \;\; \forall\, \gamma \in [0,t), \\
& a(t) \geq 0 \qquad \forall\, t \in [0,1].
\end{aligned} \tag{4.7}$$

Rewriting (4.5) gives

$$\rho^* \quad \geq \quad \frac{\int_0^1 a(t)\,dt}{\int_0^1 v_\infty(a^t)\,dt} \tag{4.8}$$

for all instances a such that $\int_0^1 a(t)\,dt > 0$. To obtain a lower bound on ρ^*, one can substitute particular instances a into (4.8). Consider instances $a : [0,1] \mapsto [0,\infty)$ that are continuous and nondecreasing. Consider any continuous nondecreasing extension $a : (-\infty,1] \mapsto [0,\infty)$, so that $\lim_{t\to-\infty} a(t) = 0$. Then, for any $t \in (0,1)$, it follows from a being continuous that f_t is differentiable, and it follows from a being nondecreasing that f_t is at first nondecreasing and then nonincreasing on $(-\infty,t)$, and f_t attains a maximum at a point $\gamma(t) \in (-\infty,t)$, where $f'_t(\gamma) = 0$. That is, for any $t \in (0,1)$, $\gamma(t)$ is a solution of

$$A(t) \quad = \quad A(\gamma) + (1-\gamma)a(\gamma) \quad \equiv \quad g(\gamma). \tag{4.9}$$

It follows from a being nondecreasing that g is nondecreasing. Also, A is nondecreasing, and thus, although the solution of (4.9) is not necessarily unique, γ can be chosen to be nondecreasing. Note that $\gamma(t) > 0$ for t sufficiently close to 1. If $\gamma(t) < 0$ for some $t \in (0,1)$, let t' be a crossing point of $\gamma(t) = 0$, that is, $\gamma(t) \leq 0$ for all $t < t'$ and $\gamma(t) \geq 0$ for all $t > t'$; otherwise, let $t' \equiv 0$. Then, for all $t < t'$, $v_\infty(a^t) = \sup_{\gamma\in[0,t)} f_t(\gamma) = f_t(0) = A(t)$. Also, for all $t > t'$, $v_\infty(a^t) = f_t(\gamma(t)) = [A(t) - A(\gamma(t))]/[1-\gamma(t)] = a(\gamma(t))$. Hence,

$$\int_0^1 v_\infty(a^t)\,dt \quad = \quad \int_0^{t'} A(t)\,dt + \int_{t'}^1 a(\gamma(t))\,dt.$$

Therefore,

$$\rho^* \quad \geq \quad \frac{\int_0^1 a(t)\,dt}{\int_0^{t'} A(t)\,dt + \int_{t'}^1 a(\gamma(t))\,dt}.$$

For example, consider a linear arrival rate $a(t) = t$. Then $A(t) = t^2/2$, $\gamma(t) = 1 - \sqrt{1-t^2}$, $t' = 0$, and thus $\rho^* \geq (\int_0^1 t\,dt)/(\int_0^1 \left(1-\sqrt{1-t^2}\right)\,dt) = 0.5/(1-\pi/4) \approx 2.3298$ (Kleywegt et al. [14]). As another example, one can consider $a(t;k) = e^{k(1-t)^{1/k}}$. In the context of the HLBP, Bar-Noy, Freund, and Naor [6] arrive at the same optimal control problem to determine ρ^*_{HLBP}. Using a function similar to $a(t;k)$ they prove a lower bound $\rho^*_{HLBP} \geq e$. They also showed that this bound is tight by demonstrating an algorithm with a competitive ratio of e in the continuous model. Thus, $\rho^* = \lim_{T\to\infty} \rho^*_T = e$ for the single deadline version of the ORMP. Applying Theorem (4.10), we conclude that the same ρ^* value applies to the ORMP with multiple deadlines.

Convergence to the limit e is quite slow, as shown earlier in Table 4.1. For example, values such as $\rho^*_{50} \approx 2.26470$ and $\rho^*_{100} \approx 2.35061$ offer considerably better performance than e.

4.2. HLBP. In this section we present optimality results for the HLBP; we investigate the application of the α-policy to the HLBP, including the calculation of the optimal competitive ratios and optimal parameters for the α-policy, $\rho^*_{T,m}$ and ρ^*. (The problem parameter β of interest for the HLBP is (T,m).) We show that if

$T \geq m$, then $\rho^*_{T,m}$ for the HLBP is equal to ρ^*_m for the ORMP, but if $T < m$, then $\rho^*_{T,m}$ for the HLBP can be strictly between ρ^*_T and ρ^*_m for the ORMP.

As before, for a given instance $\omega \in \Omega_{T,m}$, the decisions under the α-policy are given by $\pi_\alpha(\omega)(t) \equiv \alpha_{T,m} v_{T,m}(\omega^t)$. To implement the α-policy, one has to determine the optimal value of $\alpha_{T,m}$, that is, $\rho^*_{T,m}$, as well as $v_{T,m}(\omega^t)$. These two issues are addressed next.

4.2.1. Computing $\boldsymbol{v_{T,m}(\omega^t)}$. Similar to the ORMP, it is easy to compute $v_{T,m}(\omega^t) \equiv \inf_{\omega \in \Omega_{T,m}(\omega^t)} v^*(\omega)$ for the HLBP. The optimal value $v^*(\omega)$ with perfect information can be computed for any ω using Proposition 4.14. Next, for any partial instance ω^t, the best instance $\tilde{\omega} \in \Omega_{T,m}$ that starts with ω^t can be determined, and $v_{T,m}(\omega^t)$ can be computed, as shown in Proposition 4.16.

Proposition 4.14. *For any instance $\omega = (a(1), \ldots, a(T))$ of the HLBP, the optimal value $v^*(\omega)$ with perfect information is given by*

$$v^*(\omega) \quad = \quad \max_{\{i \in \{1,\ldots,m\}\}} \left\{ \frac{1}{m-i+1} \sum_{t=1}^{T} \sum_{j=i}^{m} a_j(t) \right\}.$$

The proof of Proposition 4.14 is similar to that of Proposition 4.1 and is omitted.

Corollary 4.15. *The optimal value $v^*(\omega)$ with perfect information for the HLBP is nondecreasing; that is, for any $\omega, \omega' \in \Omega_{T,m}$ with $\omega \leq \omega'$, it holds that $v^*(\omega) \leq v^*(\omega')$.*

It follows from Corollary 4.15 that for any partial instance $\omega^t = (a(1), \ldots, a(t))$, instance $(a(1), \ldots, a(t), 0, \ldots, 0)$ has the best optimal value $v^*(\tilde{\omega})$ with perfect information among all $\tilde{\omega} \in \Omega_{T,m}$ that start with ω^t. This leads to the following result, which lets us compute $v_{T,m}(\omega^t) \equiv \inf_{\tilde{\omega} \in \Omega_{T,m}(\omega^t)} v^*(\tilde{\omega})$.

Proposition 4.16. *For any partial instance $\omega^t = (a(1), \ldots, a(t)) \in \Omega^t_{T,m}$ of the HLBP, $v_{T,m}(\omega^t) = v^*(a(1), \ldots, a(t), 0, \ldots, 0)$. That is,*

$$v_{T,m}(\omega^t) \quad = \quad \max_{\{i \in \{1,\ldots,m\}\}} \left\{ \frac{1}{m-i+1} \sum_{\tau=1}^{t} \sum_{j=i}^{m} a_j(\tau) \right\}.$$

Note that $v_{T_1,m}(\omega^t) = v_{T_2,m}(\omega^t)$ for any $T_1, T_2 \geq t$ and any ω^t. Corollary 4.15 and Proposition 4.16 lead to the following result, which is used to compute $\rho^*_{T,m}$.

Corollary 4.17. *For any partial instances $\omega^t_1, \omega^t_2 \in \Omega^t_{T,m}$ with $\omega^t_1 \leq \omega^t_2$, it holds that $v_{T,m}(\omega^t_1) \leq v_{T,m}(\omega^t_2)$. Specifically, for any instance $\omega \in \Omega_{T,m}$, $v_{T,m}(\omega^t)$ is nondecreasing in t. It follows that, for any α-policy and any instance $\omega \in \Omega_{T,m}$, $\pi_\alpha(\omega)(t) \equiv \alpha_{T,m} v_{T,m}(\omega^t)$ is nondecreasing in t.*

4.2.2. Optimal parameter values. Next we address the determination of the optimal value of $\alpha_{T,m}$. For any $\omega \in \Omega$, the feasible region $\mathcal{R}(\omega)$ is determined by linear constraints (2.8), (2.9), and (2.10). Thus $\mathcal{R}(\omega)$ is a polyhedron, and is closed. Therefore it follows from Corollary 3.4 and Proposition 3.5 that the optimal value of $\alpha_{T,m}$ is $\rho^*_{T,m}$ and that the α-policy with parameter $\alpha_{T,m} = \rho^*_{T,m}$ is feasible and optimal among all deterministic online algorithms. (It is shown in section 5.2, Proposition 5.4, that the α-policy with parameters $\alpha_{T,m} = \rho^*_{T,m}$ is optimal among all randomized online algorithms.)

Next we show how $\rho^*_{T,m}$ is given by the optimal value of an integer program and by a linear program in the case with $T \geq m$. For any instance $\omega = (a(1), \ldots, a(T)) \in$

$\Omega_{T,m}$ and any machine i, let

$$\tau(i) \quad \equiv \quad \max\left\{t \,:\, \sum_{j=1}^{i} a_j(t) > 0\right\}$$

denote the last time that work arrives for which machine i is eligible. $\tau(i) = 0$ if there is no such time. Then the maximum amount of work that can be assigned to machine i under the α-policy is $\alpha_{T,m} v_{T,m}(\omega^{\tau(i)})$, and thus the maximum amount of work that can be assigned under the α-policy is $\alpha_{T,m} \sum_{i=1}^{m} v_{T,m}(\omega^{\tau(i)})$. In Lemma 4.18 we show that, if the parameter $\alpha_{T,m}$ is too small, then there exists an instance $\omega = (a(1), \ldots, a(T))$ such that the maximum amount of work that can be assigned under the α-policy, $\alpha_{T,m} \sum_{i=1}^{m} v_{T,m}(\omega^{\tau(i)})$, is less than the total amount of work to be performed, $\sum_{t=1}^{T} \sum_{i=1}^{m} a_i(t)$.

LEMMA 4.18. *Suppose that the α-policy with parameter $\alpha_{T,m}$ gives an infeasible solution with instance $\omega' \in \Omega_{T,m}$. Then there exists an instance $\omega = (a(1), \ldots, a(T)) \in \Omega_{T,m}$ such that*

$$\alpha_{T,m} \sum_{i=1}^{m} v_{T,m}(\omega^{\tau(i)}) \quad < \quad \sum_{t=1}^{T} \sum_{i=1}^{m} a_i(t), \tag{4.10}$$

where $\tau(i) \equiv \max\{t \,:\, \sum_{j=1}^{i} a_j(t) > 0\}$.

Proof. Instance ω is constructed from ω' by discarding work that does not contribute to infeasibility. This is done by inductively constructing a sequence of instances $\omega^m, \ldots, \omega^1$. Let $a_j^i(t)$ denote the amount of work in instance ω^i that arrives at time t that requires at least machine j. As before, $\omega^{i,t}$ denotes the first t components of instance ω^i, and $v_{T,m}(\omega^{i,0}) \equiv 0$. Let $q_j^i(t)$ denote the amount of work assigned to machine j at time t under the α-policy with parameter $\alpha_{T,m}$ and for instance ω^i. Let $\tau^i(j) \equiv \max\{t \,:\, \sum_{k=1}^{j} a_k^i(t) > 0\}$ denote the last time in instance ω^i that work arrives for which machine j is eligible, with $\tau^i(j) \equiv 0$ if $\sum_{t=1}^{T} \sum_{k=1}^{j} a_k^i(t) = 0$. We know that $\sum_{t=1}^{T} \sum_{j=1}^{m} q_j'(t) < \sum_{t=1}^{T} \sum_{j=1}^{m} a_j'(t)$, so we will create a sequence of instances where $\sum_{t=1}^{T} q_j(t) = \alpha_{T,m} v_{T,m}(\omega^{\tau(j)})$ for each $j = m, \ldots, 1$. Specifically, it is shown that

$$\sum_{t=1}^{T} \sum_{j=1}^{i-1} q_j^i(t) + \alpha_{T,m} \sum_{j=i}^{m} v_{T,m}(\omega^{i,\tau^i(j)}) \quad < \quad \sum_{t=1}^{T} \sum_{j=1}^{m} a_j^i(t) \tag{4.11}$$

for all $i = m, \ldots, 1$, which implies (4.10) with $\omega = \omega^1$.

Let

$$t(m) \quad \equiv \quad \min\left\{t' \,:\, \sum_{j=1}^{m} q_j'(t') < \sum_{j=1}^{m} a_j'(t')\right\}.$$

Such a t' must exist because the α-policy with parameter $\alpha_{T,m}$ gives an infeasible solution with instance ω'.

Since at time $t(m)$ the allocation $r(t(m))$ is insufficient to complete all work available at $t(m)$, it must be that machine m is being "fully used"; i.e., $q_m'(t(m)) = \alpha_{T,m} v_{T,m}(\omega'^{t(m)}) - \sum_{t=1}^{t(m)} q_m'(t)$.

Thus,

$$\sum_{t=1}^{t(m)} \sum_{j=1}^{m-1} q'_j(t) + \alpha_{T,m} v_{T,m}(\omega'^{t(m)}) \quad < \quad \sum_{t=1}^{t(m)} \sum_{j=1}^{m} a'_j(t).$$

Instance ω^m is obtained from ω' by discarding work arriving after time $t(m)$; that is, $a_j^m(t) = 0$ for all j and all $t > t(m)$, and $a_j^m(t) = a'_j(t)$ for all j and all $t \leq t(m)$. Hence $\tau^m(m) = t(m)$, and

$$\sum_{t=1}^{T} \sum_{j=1}^{m-1} q_j^m(t) + \alpha_{T,m} v_{T,m}(\omega^{m,\tau^m(m)}) \quad < \quad \sum_{t=1}^{T} \sum_{j=1}^{m} a_j^m(t),$$

which shows (4.11) for $i = m$.

As an induction hypothesis, assume that (4.11) holds for some $i \in \{2, \ldots, m\}$. Let

$$t(i-1) \quad \equiv \quad \max \left\{ t \, : \, \sum_{\tau=1}^{t} q_{i-1}^i(\tau) = \alpha_{T,m} v_{T,m}(\omega^{i,t}) \right\}$$

if such a t exists; otherwise $t(i-1) \equiv 0$. Note that $t(i-1)$ is the latest time that machine $i-1$ did the maximum amount of work it could, so later work that requests machines $i-1$ or lower does not contribute to the infeasibility of the instance. Instance ω^{i-1} is obtained from ω^i by discarding work arriving after time $t(i-1)$ that requests machines $j \leq i-1$; that is, $a_j^{i-1}(t) = 0$ for all $j \leq i-1$ and $t > t(i-1)$, and $a_j^{i-1}(t) = a_j^i(t)$ for all other j and t.

In the new instance, machine $i-1$ is fully used, so we have $\sum_{t=1}^{T} q_{i-1}^{i-1}(t) = \alpha_{T,m} v_{T,m}(\omega^{i,t(i-1)})$. Note that the total amount of work done by machines 1 through $i-1$ was reduced by the amount actually done after time $t(i-1)$, so we have

$$\sum_{t=1}^{T} \sum_{j=1}^{i-2} q_j^{i-1}(t) + \alpha_{T,m} v_{T,m}(\omega^{i,t(i-1)}) \quad = \quad \sum_{t=1}^{T} \sum_{j=1}^{i-1} q_j^i(t) - \sum_{t=t(i-1)+1}^{T} \sum_{j=1}^{i-1} a_j^i(t).$$

The final ingredients in the proof are inequalities relating $v_{T,m}$ for the various instances and truncated instances under consideration. Note that $\omega^{i-1} \leq \omega^i$ implies $\omega^{i-1,t} \leq \omega^{i,t}$ as well as $\tau^{i-1}(j) \leq \tau^i(j)$. In addition, we know that $\tau^{i-1}(i-1) \leq t(i-1)$. Putting all of this together with Corollary 4.17 gives

$$v_{T,m}(\omega^{i-1,\tau^{i-1}(j)}) \quad \leq \quad v_{T,m}(\omega^{i-1,\tau^i(j)}) \quad \leq \quad v_{T,m}(\omega^{i,\tau^i(j)})$$

and

$$v_{T,m}(\omega^{i-1,\tau^{i-1}(i-1)}) \quad \leq \quad v_{T,m}(\omega^{i-1,t(i-1)}) \quad \leq \quad v_{T,m}(\omega^{i,t(i-1)}).$$

Therefore, we can establish the induction hypothesis as follows:

$$\begin{aligned}
&\sum_{t=1}^{T}\sum_{j=1}^{i-2} q_j^{i-1}(t) + \alpha_{T,m}\sum_{j=i-1}^{m} v_{T,m}(\omega^{i-1,\tau^{i-1}(j)})\\
&\le \sum_{t=1}^{T}\sum_{j=1}^{i-2} q_j^{i-1}(t) + \alpha_{T,m} v_{T,m}(\omega^{i,t(i-1)}) + \alpha_{T,m}\sum_{j=i}^{m} v_{T,m}(\omega^{i,\tau^{i}(j)})\\
&= \sum_{t=1}^{T}\sum_{j=1}^{i-1} q_j^{i}(t) - \sum_{t=t(i-1)+1}^{T}\sum_{j=1}^{i-1} a_j^{i}(t) + \alpha_{T,m}\sum_{j=i}^{m} v_{T,m}(\omega^{i,\tau^{i}(j)})\\
&< \sum_{t=1}^{T}\sum_{j=1}^{m} a_j^{i}(t) - \sum_{t=t(i-1)+1}^{T}\sum_{j=1}^{i-1} a_j^{i}(t)\\
&= \sum_{t=1}^{T}\sum_{j=1}^{m} a_j^{i-1}(t). \qquad \square
\end{aligned}$$

THEOREM 4.19. *Parameter $\alpha_{T,m}$ for the α-policy is too small ($\alpha_{T,m} < \rho^*_{T,m}$) if and only if there exists an instance $\omega = (a(1),\dots,a(T)) \in \Omega_{T,m}$ such that $\alpha_{T,m}\sum_{i=1}^{m} v_{T,m}(\omega^{\tau(i)}) < \sum_{t=1}^{T}\sum_{i=1}^{m} a_i(t)$. Also,*

$$\begin{aligned}
\rho^*_{T,m} = \inf\Bigg\{\rho \in [1,\infty) \,:\, \rho\sum_{i=1}^{m}\max_{\{j\in\{1,\dots,m\}\}}\left\{\frac{1}{m-j+1}\sum_{t=1}^{\tau(i)}\sum_{k=j}^{m} a_k(t)\right\}&\\
\ge \sum_{t=1}^{T}\sum_{i=1}^{m} a_i(t) \ \ \forall\ (a(1),\dots,a(T)) \in \mathbb{R}_+^{Tm}\Bigg\}.& \qquad (4.12)
\end{aligned}$$

The α-policy provides the optimal solution with perfect information for the zero instance $\omega = (0,\dots,0)$. Thus, to determine $\rho^*_{T,m}$, one can restrict attention to instances $(a(1),\dots,a(T)) \in \mathbb{R}_+^{Tm}$ for which $\sum_{t=1}^{T}\sum_{i=1}^{m} a_i(t) > 0$. It follows from (4.12) that $\rho^*_{T,m}$ is determined by instances $(a(1),\dots,a(T)) \in \mathbb{R}_+^{Tm}$ that minimize

$$\sum_{i=1}^{m}\max_{\{j\in\{1,\dots,m\}\}}\left\{\frac{1}{m-j+1}\sum_{t=1}^{\tau(i)}\sum_{k=j}^{m}\frac{a_k(t)}{\sum_{t'=1}^{T}\sum_{i'=1}^{m} a_{i'}(t')}\right\}.$$

It is also clear that one can restrict attention to instances $(a(1),\dots,a(T)) \in \mathbb{R}_+^{Tm}$ for which $\sum_{t=1}^{T}\sum_{i=1}^{m} a_i(t) = 1$.

COROLLARY 4.20. *The optimal competitive ratio $\rho^*_{T,m}$ for HLBP can be calculated by solving the following integer linear program (IP) with decision variables $a_i(t)$, $x(t)$,*

y_i, *and* $z_i(t)$, $i \in \{1, \dots, m\}$, $t \in \{1, \dots, T\}$*:*

$$\begin{array}{lll}
\text{(IP)} \quad \textit{Minimize} & \displaystyle\sum_{i=1}^{m} y_i & \\
\phantom{\text{(IP)}} \quad \textit{subject to} & \displaystyle\sum_{t=1}^{T} \sum_{i=1}^{m} a_i(t) = 1, & \\
& \displaystyle x(t) \geq \frac{1}{m-i+1} \sum_{\tau=1}^{t} \sum_{j=i}^{m} a_j(\tau) & \forall\ i, \quad \forall\ t, \\
& y_i \geq x(t) + z_i(t) - 1 & \forall\ i, \quad \forall\ t, \\
& \displaystyle z_i(t) \geq \sum_{j=1}^{i} a_j(t) & \forall\ i, \quad \forall\ t, \\
& a_i(t) \geq 0 & \forall\ i, \quad \forall\ t, \\
& y_i \geq 0 & \forall\ i, \\
& z_i(t) \in \{0, 1\} & \forall\ i, \quad \forall\ t.
\end{array}$$

Proof. In an optimal solution (a^*, x^*, y^*, z^*) of the IP, $a^* = (a^*(1), \dots, a^*(T))$ represents a worst-case instance in $\Omega_{T,m}$ for the α-policy. Without loss of generality we can assume that $x^*(t) = v_{T,m}(a^*(t))$, which is at most 1 for all t. It follows from Corollary 4.17 that $x^*(t)$ is nondecreasing in t, and thus each y_i^* is equal to $x^*(t)$ for the largest value of t for which $z_i^*(t) = 1$; that is, y_i^* represents the corresponding value of $v_{T,m}(a^*(\tau(i)))$. Also, it follows from (4.12) that

$$\rho_{T,m}^* \quad = \quad \inf \left\{ \rho \in [1, \infty) \ : \ \rho \sum_{i=1}^{m} y_i^* \geq 1 \right\}.$$

That is, $\rho_{T,m}^* = \max\{1, 1/\sum_{i=1}^{m} y_i^*\}$. Actually, $\rho_{T,m}^* = 1/\sum_{i=1}^{m} y_i^*$, as shown next. If $T \geq m$, then it is easily checked that (a, x, y, z) with

$$\begin{aligned}
a_i(i) &= 1/m \text{ for } i \in \{1, \dots, m\}, \\
a_i(t) &= 0 \text{ otherwise}, \\
x(t) &= \begin{cases} t/m^2 & \text{for } t \leq m, \\ 1/m & \text{for } t \geq m, \end{cases} \\
y_i &= i/m^2 \text{ for } i \in \{1, \dots, m\}, \\
z_i(t) &= \begin{cases} 1 & \text{for } t \leq i, \\ 0 & \text{for } t > i \end{cases}
\end{aligned}$$

is a feasible solution for the IP. Thus $\sum_{i=1}^{m} y_i^* \leq \sum_{i=1}^{m} i/m^2 = (1 + 1/m)/2$. Similarly, if $T < m$, then it is easily checked that (a, x, y, z) with

$$\begin{aligned}
a_{m-T+t}(t) &= 1/T \text{ for } t \in \{1, \dots, T\}, \\
a_i(t) &= 0 \text{ otherwise}, \\
x(t) &= t/T^2 \text{ for } t \in \{1, \dots, T\}, \\
y_i &= \begin{cases} 0 & \text{for } i \in \{1, \dots, m-T\}, \\ (i - m + T)/T^2 & \text{for } i \in \{m - T + 1, \dots, m\}, \end{cases} \\
z_i(t) &= \begin{cases} 1 & \text{for } t \leq i - m + T, \\ 0 & \text{for } t > i - m + T \end{cases}
\end{aligned}$$

Table 4.2
Values of $\rho^*_{T,m}$ for the HLBP.

T	m	$\rho^*_{T,m}$
2	4	4/3
2	5	4/3
2	6	4/3
3	4	3/2
3	5	1.511629
3	6	1.511629
3	9	1.520549
3	14	1.522063
3	19	1.522063
4	5	1.629631
4	6	1.629631
4	7	1.630138
4	8	1.630138

is a feasible solution for the IP. Thus $\sum_{i=1}^m y_i^* \le \sum_{i=m-T+1}^m (i-m+T)/T^2 = \sum_{t=1}^T t/T^2 = (1+1/T)/2$. Therefore $\rho^*_{T,m} = 1/\sum_{i=1}^m y_i^* \ge 2/(1+1/\min\{T,m\}) \ge 1$. □

Table 4.2 gives the values of $\rho^*_{T,m}$ for some values of T and m, as determined by the IP.

There are a total of $2Tm+T+m$ decision variables and $3Tm+1$ constraints, so the IP grows fairly rapidly. Next it is shown that if $T \ge m$, then $\rho^*_{T,m}$ for the HLBP is equal to ρ^*_m for the ORMP, which can be computed with the LP given in section 4.1.2. First, Lemma 4.21 shows that, to determine $\rho^*_{T,m}$, one can restrict attention to instances in which at most one machine is requested in each time period and in which machines are requested from least versatile to most versatile.

Lemma 4.21. *Suppose that the α-policy with parameter $\alpha_{T,m}$ gives an infeasible solution with instance $\omega' = (a'(1), \ldots, a'(T)) \in \Omega_{T,m}$. Then there exists an instance $\hat{\omega} = (\hat{a}(1), \ldots, \hat{a}(\lambda)) \in \Omega_{\lambda,m}$ such that the following hold:*

1. *For each t, let $m'(t) \equiv \min\{i : a'_i(t) > 0\}$ (ignoring all t such that $\sum_i a'_i(t) = 0$). Then $\lambda \equiv |\{m'(1), \ldots, m'(T)\}| \le \min\{T, m\}$.*
2. *For all $t \in \{1, \ldots, \lambda\}$, $\hat{a}_i(t) > 0$ for exactly one i.*
3. *Let $\hat{m}(t) \equiv \min\{i : \hat{a}_i(t) > 0\}$; that is, $\hat{m}(t)$ is the unique i such that $\hat{a}_i(t) > 0$. Then $\hat{m}$ is strictly increasing.*
4. *We have*

$$\alpha_{T,m} \sum_{i=1}^m v_{\lambda,m}(\hat{\omega}^{\hat{\tau}(i)}) \quad < \quad \sum_{t=1}^{\lambda} \hat{a}_{\hat{m}(t)}(t),$$

where $\hat{\tau}(i) \equiv \max\{t : \sum_{j=1}^i \hat{a}_j(t) > 0\}$.

Proof. It was shown in Lemma 4.18 that if the α-policy with parameter $\alpha_{T,m}$ gives an infeasible solution with instance $\omega' \in \Omega_{T,m}$, then there exists an instance $\omega = (a(1), \ldots, a(T)) \in \Omega_{T,m}$ such that

$$\alpha_{T,m} \sum_{i=1}^m v_{T,m}(\omega^{\tau(i)}) \quad < \quad \sum_{t=1}^T \sum_{i=1}^m a_i(t).$$

For each t, let $m(t) \equiv \min\{i : a_i(t) > 0\}$ denote the least versatile machine requested at time t (simply ignore all t such that $\sum_{i=1}^m a_i(t) = 0$). For each t, let $\tilde{a}_{m(t)}(t) \equiv$

$\sum_{i=1}^m a_i(t)$, and $\tilde{a}_i(t) = 0$ for all $i \neq m(t)$; that is, at each time t, all the work requests machine $m(t)$. Note that, for each i, $\tilde{\tau}(i) = \tau(i)$, and for all i and all t, $\sum_{j=i}^m \tilde{a}_j(t) \leq \sum_{j=i}^m a_j(t)$. Thus it follows that $v_{T,m}(\tilde{\omega}^{\tilde{\tau}(i)}) \leq v_{T,m}(\omega^{\tau(i)})$ for all i. Also, $\sum_{t=1}^T \sum_{i=1}^m \tilde{a}_i(t) = \sum_{t=1}^T \sum_{i=1}^m a_i(t)$, from which we have

$$\alpha_{T,m} \sum_{i=1}^m v_{T,m}(\tilde{\omega}^{\tilde{\tau}(i)}) \quad < \quad \sum_{t=1}^T \sum_{i=1}^m \tilde{a}_i(t).$$

It follows from the construction of $\tilde{\omega}$ that $\tilde{m}(t) \equiv \min\{i : \tilde{a}_i(t) > 0\} = m'(t)$; that is, $\tilde{m}(t) = m'(t)$ is the unique i such that $\tilde{a}_i(t) > 0$. For each $t \in \{1, \ldots, \lambda\}$, let $i(t)$ be the tth smallest element in $\{\tilde{m}(1), \ldots, \tilde{m}(T)\}$. Let $\hat{a}_{i(t)}(t) \equiv \sum_{\tau=1}^T \tilde{a}_{i(t)}(\tau)$, and $\hat{a}_i(t) = 0$ for all $i \in \{1, \ldots, m\}$ and all $t \in \{1, \ldots, \lambda\}$ with $i \neq i(t)$. Thus $\hat{m}(t) = i(t)$, which is by definition strictly increasing in t. For any $i, j \in \{1, \ldots, m\}$,

$$\sum_{t=1}^{\hat{\tau}(i)} \sum_{k=j}^m \hat{a}_k(t) \quad = \quad \sum_{t=1}^{\lambda} \sum_{k=j}^i \hat{a}_k(t) \quad = \quad \sum_{t=1}^{T} \sum_{k=j}^i \tilde{a}_k(t) \quad = \quad \sum_{t=1}^{\tilde{\tau}(i)} \sum_{k=j}^i \tilde{a}_k(t) \quad \leq \quad \sum_{t=1}^{\tilde{\tau}(i)} \sum_{k=j}^m \tilde{a}_k(t)$$

and thus

$$\begin{aligned} v_{\lambda,m}(\hat{\omega}^{\hat{\tau}(i)}) &= \max_{\{j \in \{1,\ldots,m\}\}} \left\{ \frac{1}{m-j+1} \sum_{t=1}^{\hat{\tau}(i)} \sum_{k=j}^m \hat{a}_k(t) \right\} \\ &\leq \max_{\{j \in \{1,\ldots,m\}\}} \left\{ \frac{1}{m-j+1} \sum_{t=1}^{\tilde{\tau}(i)} \sum_{k=j}^m \tilde{a}_k(t) \right\} \\ &= v_{T,m}(\tilde{\omega}^{\tilde{\tau}(i)}). \end{aligned}$$

Therefore

$$\alpha_{T,m} \sum_{i=1}^m v_{\lambda,m}(\hat{\omega}^{\hat{\tau}(i)}) \leq \alpha_{T,m} \sum_{i=1}^m v_{T,m}(\tilde{\omega}^{\tilde{\tau}(i)}) < \sum_{t=1}^T \sum_{i=1}^m \tilde{a}_i(t) = \sum_{t=1}^{\lambda} \hat{a}_{\hat{m}(t)}(t). \qquad \square$$

It follows from Lemma 4.21 that if the α-policy with parameter $\alpha_{T,m}$ gives an infeasible solution with instance $\omega' \in \Omega_{T,m}$, then there exists an instance $\omega = (a(1), \ldots, a(m)) \in \Omega_{m,m}$ such that $a_i(t) = 0$ for all $i \neq t$, and

$$\alpha_{T,m} \sum_{i=1}^m v_{m,m}(\omega^i) \quad < \quad \sum_{i=1}^m a_i(i).$$

This observation leads to Theorem 4.22.

THEOREM 4.22. *Parameter $\alpha_{T,m}$, with $T \geq m$, for the α-policy is too small ($\alpha_{T,m} < \rho^*_{T,m}$) if and only if there exists an instance $\omega = (a(1), \ldots, a(m)) \in \Omega_{m,m}$ such that $a_i(t) = 0$ for all $i \neq t$, and $\alpha_{T,m} \sum_{i=1}^m v_{m,m}(\omega^i) < \sum_{i=1}^m a_i(i)$. Also, for $T \geq m$,*

$$\begin{aligned} \rho^*_{T,m} \quad = \quad \rho^*_{m,m} = \inf \Bigg\{ \rho \in [1, \infty) \, : \, \rho \sum_{i=1}^m \max_{\{j \in \{1,\ldots,i\}\}} \left\{ \frac{1}{m-j+1} \sum_{k=j}^i a(k) \right\} \\ \geq \sum_{i=1}^m a(i) \quad \forall \ (a(1), \ldots, a(m)) \in \mathbb{R}_+^m \Bigg\}. \qquad (4.13) \end{aligned}$$

Corollary 4.23 follows by comparing (4.13) and (4.2).

Corollary 4.23. *For $T \geq m$, $\rho^*_{T,m}$ for the HLBP is equal to ρ^*_m for the ORMP, which can be computed with the LP.*

4.2.3. Asymptotic behavior. For the HLBP, we are interested in $\rho^* = \sup_{T\in\mathbb{Z}_+, m\in\mathbb{Z}_+} \rho^*_{T,m}$, as well as $\rho^*_{\infty,m} \equiv \sup_{T\in\mathbb{Z}_+} \rho^*_{T,m}$, and $\rho^*_{T,\infty} \equiv \sup_{m\in\mathbb{Z}_+} \rho^*_{T,m}$. As pointed out in section 2.3, $\rho^*_{T,m}$ is nondecreasing in T and m, and thus $\rho^* = \lim_{m\to\infty} \rho^*_{m,m}$. Thus it follows from Corollary 4.23 that ρ^* for the HLBP is equal to ρ^* for the ORMP, which is equal to e. It also follows that $\sup_{T\in\mathbb{Z}_+} \rho^*_{T,m} = \lim_{T\to\infty} \rho^*_{T,m} = \rho^*_{m,m}$, which is equal to ρ^*_m for the ORMP. Also, $\sup_{m\in\mathbb{Z}_+} \rho^*_{T,m} = \lim_{m\to\infty} \rho^*_{T,m}$; however, this asymptotic behavior is not well understood yet, except for the bounds $\lim_{m\to\infty} \rho^*_{T,m} \geq \rho^*_T$ for the ORMP, and $\lim_{m\to\infty} \rho^*_{T,m} \leq \rho^* = e$. By using a discrete time, continuous machine model of the HLBP, it can be shown that $\lim_{m\to\infty} \rho^*_{2,m} = \rho^*_2 = 4/3$. It also follows from a comparison of Tables 4.1 and 4.2 that $\lim_{m\to\infty} \rho^*_{T,m} > \rho^*_T$ for some T. We conjecture that $\lim_{m\to\infty} \rho^*_{T,m} < \lim_{m\to\infty} \rho^*_{T+1,m}$, which would imply that $\lim_{m\to\infty} \rho^*_{T,m} < \rho^* = e$ for all T.

4.3. Generalizations for the OMMP. In this section we briefly show how some of the results in section 4.1 for the ORMP can be generalized for the OMMP.

Let $\theta : \Omega \mapsto \Omega$ be any function such that for any $\omega \in \Omega_T$, $\theta(\omega) \in \Omega_T$. For any partial instance $\theta(\omega)^t$, let $\Omega^\theta_T(\theta(\omega)^t)$ denote the set of all instances in $\theta(\Omega_T)$ with first t elements equal to $\theta(\omega)^t$. Let

$$v^\theta_T(\theta(\omega)^t) \quad \equiv \quad \inf_{\omega\in\Omega^\theta_T(\theta(\omega)^t)} v^*(\omega).$$

Assume that $v^\theta_T(\theta(\omega)^t) \leq v_T(\omega^t)$ for all $t \in \{1,\ldots,T\}$.

Let $\vartheta : \Omega \mapsto \Omega$ be any function such that for any $\omega \in \Omega_T$, $\vartheta(\omega) \in \Omega_{T-1}$. For any partial instance $\vartheta(\omega)^t$, $t \leq T-1$, let $\Omega^\vartheta_T(\vartheta(\omega)^t)$ denote the set of all instances in $\vartheta(\Omega_T)$ with first t elements equal to $\vartheta(\omega)^t$. Let

$$v^\vartheta_T(\vartheta(\omega)^t) \quad \equiv \quad \inf_{\omega\in\Omega^\vartheta_T(\vartheta(\omega)^t)} v^*(\omega).$$

Assume that $v^\vartheta_T(\vartheta(\omega)^t) \leq v_T(\omega^t)$ for all $t \in \{1,\ldots,T-1\}$.

Thus both θ and ϑ map an instance ω to instances $\theta(\omega)$ and $\vartheta(\omega)$ that are "smaller" than ω. As before, for any $r \in \mathbb{R}^T_+$, r^{T-1} denotes the first $T-1$ components of r.

Definition 4.24 (extensibility property). *We say that $(\mathcal{R},\theta,\vartheta)$ has the extensibility property if, for any $\omega \in \Omega_T$ and any $r \in \mathbb{R}^T_+$, $r \in \mathcal{R}(\theta(\omega))$ and $r^{T-1} \in \mathcal{R}(\vartheta(\omega))$ imply that $r \in \mathcal{R}(\omega)$.*

Intuitively, the extensibility property states that although instances $\theta(\omega)$ and $\vartheta(\omega)$ are smaller than ω, together the feasibility of a solution r for $\theta(\omega)$ and r^{T-1} for $\vartheta(\omega)$ are sufficient to establish the feasibility of r for ω.

Theorem 4.25 follows along the lines of Theorem 4.9.

Theorem 4.25. *Suppose that $(\mathcal{R},\theta,\vartheta)$ has the extensibility property. If the α-policy with parameter α gives an infeasible solution for some instance ω and a feasible solution for instance $\vartheta(\omega)$, then the α-policy with parameter α gives an infeasible solution for instance $\theta(\omega)$.*

Next it is shown that if $(\mathcal{R},\theta,\vartheta)$ has the extensibility property, then to determine ρ^*_T, it is sufficient to consider only the instances in the image $\theta(\Omega_T)$ of θ.

THEOREM 4.26. *Suppose that $(\mathcal{R}, \theta, \vartheta)$ has the extensibility property and that ρ_T^* is nondecreasing in T. Then*

$$\rho_T^* \;=\; \inf_{\alpha \geq 1} \inf \left\{ \rho \geq 1 \,:\, v^{\pi_\alpha}(\omega) \leq \rho v^*(\omega) \;\; \forall\, \omega \in \theta(\Omega_T) \right\}.$$

That is, to determine ρ_T^* (and ρ^*), it is sufficient to consider only the α-policy and only the instances in $\theta(\Omega_T)$.

5. Randomized algorithms.

So far we have analyzed only deterministic algorithms. However, for some problems, randomized algorithms can have better competitive ratios than deterministic algorithms (Motwani and Raghavan [16] and Hoogeveen and Vestjens [11]), and we investigate that possibility here. First, we show that a randomized algorithm can have better competitive ratios for the OMMP than any deterministic algorithm. Second, we show that if the feasible set $\mathcal{R}(\omega)$ is convex for all ω, then randomized algorithms do not have better competitive ratios than deterministic algorithms.

5.1. Randomized algorithms may have better competitive ratios.

We give an example of an OMMP for which a randomized algorithm has a better competitive ratio than any deterministic algorithm. The example is for an ORMP with strictly convex productivity function η.

PROPOSITION 5.1. *Consider the single deadline ORMP with a stationary productivity function of the form $\eta_t(r) = cr^p$, $c > 0$, $p > 1$, for all t. For this problem there exists a randomized algorithm $\pi \in \Pi^{RO}$ such that $v^\pi(\omega) < v^{\pi_\alpha}(\omega)$ for every $\omega \in \Omega_2$, $\omega > 0$.*

The idea is to choose a randomized algorithm π that prescribes almost the same decisions as the α-policy, except that it randomly chooses to do $\pm\varepsilon$ extra work in the first time period and $\mp\varepsilon$ extra work in the second time period, where ε is a sufficiently small value. The solution is still feasible since the total amount of work performed is the same, and the maximum amount of work performed will be equal to the maximum amount under the α-policy, $\mp\epsilon$. When one takes η^{-1} to determine the amount of resource required to perform the work, and averages over the $+\varepsilon$ and $-\varepsilon$ outcomes, one gets a number that is lower than the maximum amount of resource under the α-policy, due to the strict convexity of η.

5.2. A sufficient condition for optimality of deterministic algorithms.

In this section we present a sufficient condition for the (deterministic online) α-policy to be optimal among all randomized online algorithms for the OMMP. Thereafter we apply the result to show that the α-policy is optimal among all randomized online algorithms for the ORMP with concave productivity functions η_t as well as for the HLBP.

THEOREM 5.2. *Suppose that $\mathcal{R}(\omega)$ is convex for all $\omega \in \Omega_\beta$. For any algorithm $\pi \in \Pi^{RO}$, if $\rho_\beta^\pi < \infty$, then the α-policy with parameter $\alpha_\beta = \rho_\beta^\pi$ achieves the same competitive ratio, $\rho_\beta^{\pi_\alpha} = \rho_\beta^\pi$.*

Proof. We show that the α-policy with parameter $\alpha_\beta = \rho_\beta^\pi$ leads to feasible solutions for all instances $\omega \in \Omega_\beta$ by showing that $\pi_\alpha(\omega)(t) \geq E[\pi(\omega)(t)]$ for all $\omega \in \Omega_\beta$ and all t. For this, we mimic the first part of the proof of Theorem 3.2, replacing $\pi(\omega)(t)$ with $E[\pi(\omega)(t)]$.

Because $\rho_\beta^\pi < \infty$, it holds for all $\omega \in \Omega_\beta$ that $\pi(\omega)[\mathcal{R}(\omega)] = 1$; that is, with probability 1, the solution $(\pi(\omega)(1), \ldots, \pi(\omega)(T))$ is in $\mathcal{R}(\omega)$. Then, because $\mathcal{R}(\omega)$ is convex and $E[\pi(\omega)(t)] \leq \rho_\beta^\pi < \infty$ for all $\omega \in \Omega_\beta$ and all t, the solution $(E[\pi(\omega)(1)], \ldots,$

$E[\pi(\omega)(T)])$ is in $\mathcal{R}(\omega)$. Thus it follows from feasibility monotonicity that $(\pi_\alpha(\omega)(1), \ldots, \pi_\alpha(\omega)(T)) \in \mathcal{R}(\omega)$; that is, the α-policy with $\alpha_\beta = \rho^\pi_\beta$ leads to feasible solutions for all $\omega \in \Omega_\beta$. Therefore $\rho^{\pi_\alpha}_\beta \leq \alpha_\beta = \rho^\pi_\beta$. Also, it follows from the definition of the α-policy and $\rho^\pi_\beta < \infty$ that $\rho^{\pi_\alpha}_\beta \geq \rho^\pi_\beta$. Thus $\rho^{\pi_\alpha}_\beta = \rho^\pi_\beta$. □

Next we show that the α-policy is optimal among all randomized online algorithms for the ORMP with quasi-concave productivity functions η_t, as well as for the HLBP. These results follow directly from applying Theorem 5.2 to the ORMP and the HLBP. Recall that for any quasi-concave function $f : \mathbb{R}^n \mapsto \mathbb{R}$ and for any l, the set $\{x \in \mathbb{R}^n : f(x) \geq l\}$ is convex.

Proposition 5.3. *If the ORMP productivity functions η_t are quasi-concave for all t, then*

1. *the set of feasible solutions $\mathcal{R}(\omega)$ is convex for all ω, and*
2. *the α-policy with parameters $\alpha_T = \rho^*_T$ is optimal among all randomized online algorithms.*

Proposition 5.4. *For the HLBP,*

1. *the set of feasible solutions $\mathcal{R}(\omega)$ is convex for all ω, and*
2. *the α-policy with parameters $\alpha_{T,m} = \rho^*_{T,m}$ is optimal among all randomized online algorithms.*

6. Conclusions and further research.

The α-policy theory developed in this paper is a powerful tool for finding worst-case optimal algorithms for online min-max problems. It makes analysis easier by turning optimization questions into feasibility questions and by focusing on properties such as feasibility monotonicity and extensibility.

Our work suggests several questions and possible extensions. For the ORMP, the solutions to the LP show that the convergence of ρ^*_T to ρ^* is quite slow. It would be nice to have a theoretical explanation of this. For the HLBP, the asymptotic values of $\rho^*_{T,m}$ as $m \to \infty$ are not known. Given the slow convergence for the ORMP, and the computational limitations of the IP, it may be difficult to make inferences about asymptotic values from known values of $\rho^*_{T,m}$. The HLBP can be generalized to nonlinear server hierarchies [6] such as rooted in-trees and general partial orders. Values of $\rho^*_{T,m}$ for these cases are unknown.

If the future is completely unknown, and the decision maker is risk-averse, then the online criterion optimized in this paper would be quite appropriate. If future arrivals are known in distribution, the problem would become a Markov decision process and could be attacked with dynamic programming methods. We suspect that the most realistic models would involve an intermediate level of information. It would be interesting to see whether an algorithm which is a blend of the α-policy and other algorithms could perform well on such a model or whether altogether new algorithms are needed.

Appendix A. Proof of Lemma 2.2.

Lemma 2.2. *If β is known beforehand, then*

$$\rho^* \quad = \quad \sup_{\beta \in \mathbf{B}} \rho^*_\beta.$$

Proof. It remains to be shown that

$$\rho^* \;\equiv\; \inf_{\pi \in \Pi^{RO}} \sup_{\beta \in \mathbf{B}} \rho^\pi_\beta \;\leq\; \sup_{\beta \in \mathbf{B}} \inf_{\pi \in \Pi^{RO}} \rho^\pi_\beta \;\equiv\; \hat{\rho}.$$

If $\hat{\rho} = \infty$, the result follows immediately. Suppose $\hat{\rho} < \infty$. Thus $\inf_{\pi \in \Pi^{RO}} \rho^\pi_\beta \leq \hat{\rho} < \infty$ for all β. Hence, for any $\beta \in \mathbf{B}$ and any $\varepsilon > 0$, there exists an algorithm $\pi^\varepsilon_\beta \in \Pi^{RO}$

such that $\rho_\beta^{\pi_\beta^\varepsilon} < \hat{\rho} + \varepsilon$. Choose algorithm π^ε to be the same as algorithm π_β^ε if $\omega \in \Omega_\beta$. Because β is known in advance, π^ε is an online algorithm. Then $\rho_\beta^{\pi^\varepsilon} < \hat{\rho} + \varepsilon$ for all $\beta \in \mathbf{B}$. Thus $\sup_{\beta\in\mathbf{B}} \rho_\beta^{\pi^\varepsilon} \le \hat{\rho} + \varepsilon$, and hence $\inf_{\pi\in\Pi^{RO}} \sup_{\beta\in\mathbf{B}} \rho_\beta^\pi \le \hat{\rho} + \varepsilon$ for any $\varepsilon > 0$. Therefore $\inf_{\pi\in\Pi^{RO}} \sup_{\beta\in\mathbf{B}} \rho_\beta^\pi \le \hat{\rho}$. □

Appendix B. Proposition B.1.

PROPOSITION B.1. *For the ORMP, if the productivity function η_t is upper semicontinuous for all $t \in \{1, \dots, T\}$, then the set of feasible solutions $\mathcal{R}(\omega)$ is closed for all instances $\omega \in \Omega_T$.*

Proof. Consider any instance $\omega \in \Omega_T$, and any limit point $\hat{r} = (\hat{r}(1), \dots, \hat{r}(T))$ of $\mathcal{R}(\omega)$. It is to be shown that $\hat{r} \in \mathcal{R}(\omega)$. Because of the EDF assignment of resources to work, the only constraint that may be violated by solution $\hat{r}$ is (2.4). Consider any $\varepsilon > 0$. Because η_t is upper semicontinuous, there exists $\delta_t > 0$ such that $\eta_t(\hat{r}(t)) > \eta_t(r(t)) - \varepsilon/T$ for all $r(t)$ with $|\hat{r}(t) - r(t)| < \delta_t$. Let $\delta \equiv \min\{\delta_1, \dots, \delta_T\}$. There exists $r^\delta = (r^\delta(1), \dots, r^\delta(T)) \in \mathcal{R}(\omega)$ such that $\|\hat{r} - r^\delta\| < \delta$. Thus $|\hat{r}(t) - r^\delta(t)| < \delta_t$ and $\eta_t(\hat{r}(t)) > \eta_t(r^\delta(t)) - \varepsilon/T$ for all t. Let $\hat{w}_u(t)$ ($w_u^\delta(t)$) denote the amount of work with deadline u waiting at time t to be performed under solution $\hat{r}$ (r^δ) and EDF assignment, after the arrivals at time t have taken place, but before any work has been performed at time t. Let $\hat{W}_u(t) \equiv \hat{w}_1(t) + \cdots + \hat{w}_u(t)$, and let $W_u^\delta(t) \equiv w_1^\delta(t) + \cdots + w_u^\delta(t)$. It is shown by induction on t that $\hat{W}_u(t) < W_u^\delta(t) + \varepsilon t/T$ for all $t, u \in \{1, \dots, T\}$, and that $\hat{w}_u(t) = 0$ for all $u < t$. For $t = 1$, $\hat{W}_u(1) = a_1(1) + \cdots + a_u(1) = W_u^\delta(1) < W_u^\delta(1) + \varepsilon/T$. As an induction hypothesis, suppose that $\hat{W}_u(t) < W_u^\delta(t) + \varepsilon t/T$ for all $u \in \{1, \dots, T\}$. For all $u < t$, $w_u^\delta(t) = 0$ because $r^\delta \in \mathcal{R}(\omega)$, and thus for all $u < t$, $\hat{W}_u(t) < W_u^\delta(t) + \varepsilon t/T = \varepsilon t/T$. Hence, for all $u < t$, $\hat{w}_u(t) < \varepsilon t/T$ for all $\varepsilon > 0$. Thus $\hat{w}_u(t) = 0$ for all $u < t$, $\hat{W}_u(t) = 0$, and $\hat{W}_u(t+1) = 0 < W_u^\delta(t+1) + \varepsilon(t+1)/T$ for all $u < t$. Consider $u \ge t$. If $\hat{W}_u(t) \le \eta_t(\hat{r}(t))$, then $\hat{W}_u(t+1) = a_{t+1}(t+1) + \cdots + a_u(t+1) \le W_u^\delta(t+1) < W_u^\delta(t+1) + \varepsilon(t+1)/T$. Otherwise, if $\hat{W}_u(t) > \eta_t(\hat{r}(t))$, then $\hat{W}_u(t+1) = \hat{W}_u(t) - \eta_t(\hat{r}(t)) + a_{t+1}(t+1) + \cdots + a_u(t+1) < W_u^\delta(t) + \varepsilon t/T - \eta_t(r^\delta(t)) + \varepsilon/T + a_{t+1}(t+1) + \cdots + a_u(t+1) \le W_u^\delta(t+1) + \varepsilon(t+1)/T$, and the induction hypothesis has been established. Hence $\hat{w}_u(t) = 0$ for all $u < t$, and therefore $\hat{r} \in \mathcal{R}(\omega)$. □

Appendix C. Proof of Proposition 4.1.

PROPOSITION 4.1. *For any instance $\omega = (a(1), \dots, a(T))$ of the ORMP with nondecreasing productivity function $\eta_t(r)$, the optimal value $v^*(\omega)$ with perfect information is given by*

$$v^*(\omega) \quad = \quad \inf\left\{ r \ge 0 \,:\, \sum_{t=i}^{j} \eta_t(r) \ge \sum_{t=i}^{j} \sum_{u=t}^{j} a_u(t) \quad \forall\ i, j \in \{1, \dots, T\}, i \le j \right\},$$

where $\inf \varnothing = \infty$.

Proof. Let

$$\gamma^*(\omega) \quad \equiv \quad \inf\left\{ r \ge 0 \,:\, \sum_{t=i}^{j} \eta_t(r) \ge \sum_{t=i}^{j} \sum_{u=t}^{j} a_u(t) \quad \forall\ i, j \in \{1, \dots, T\}, i \le j \right\}.$$

It follows from the productivity function $\eta_t(r)$ being nondecreasing for all t that $v^*(\omega) \ge \gamma^*(\omega)$. It remains to show that $v^*(\omega) \le \gamma^*(\omega)$. If $\gamma^*(\omega) = \infty$, the result follows immediately. Otherwise, fix any $\varepsilon > 0$. It follows from the definition of $\gamma^*(\omega)$ that for any $\varepsilon > 0$ there exists an $r^\varepsilon \ge 0$ such that $r^\varepsilon < \gamma^*(\omega) + \varepsilon$ and

$\sum_{t=i}^{j} \eta_t(r^{\varepsilon}) \geq \sum_{t=i}^{j} \sum_{u=t}^{j} a_u(t)$ for all $i, j \in \{1, \ldots, T\}, i \leq j$. Consider the solution that makes r^{ε} amount of resource available at each time t. Let $w(t)$ denote the total amount of work waiting to be processed after the arrivals at time t have taken place, but before any processing at time t. Thus if $w(t) \leq \eta_t(r^{\varepsilon})$, then all $w(t)$ amount of work is performed at time t. Otherwise, if $w(t) > \eta_t(r^{\varepsilon})$, then $\eta_t(r^{\varepsilon})$ amount of work is performed at time t, according to the EDF rule. Hence this solution never uses more than r^{ε} amount of resource at a time. The solution is feasible and has objective value $r^{\varepsilon} < \gamma^*(\omega) + \varepsilon$ if and only if all work is completed by the deadlines, which is established next.

For any time t, let $\ell(t)$ denote the last time $\tau \in \{1, \ldots, t\}$ that there is no work with deadlines less than or equal to t waiting to be processed after the processing at time τ has taken place. If there is no such time $\tau \in \{1, \ldots, t\}$, then let $\ell(t) = 0$ (which will turn out never to be the case). Thus the objective is to show that $\ell(t) = t$, which implies that all work with deadlines less than or equal to t has been processed at the end of time t. Suppose $\ell(t) < t$. Then the amount of work with deadlines less than or equal to t that has to be processed in $\{\ell(t)+1, \ldots, t\}$ is $\sum_{\tau=\ell(t)+1}^{t} \sum_{u=\tau}^{t} a_u(\tau)$, which is less than or equal to $\sum_{\tau=\ell(t)+1}^{t} \eta_\tau(r^{\varepsilon})$ from the definition of r^{ε}. However, from the definition of $\ell(t)$, there is work with deadlines less than or equal to t remaining at the end of each of the times in $\{\ell(t)+1, \ldots, t\}$, and thus from the definition of the solution with the EDF rule, $\sum_{\tau=\ell(t)+1}^{t} \eta_\tau(r^{\varepsilon})$ amount of work with deadlines less than or equal to t is processed in $\{\ell(t)+1, \ldots, t\}$. Thus the amount of work with deadlines less than or equal to t that is processed in $\{\ell(t)+1, \ldots, t\}$ is at least as much as the amount that needs to be processed to finish all such work by time t. Thus all work is finished by the deadlines, and $v^*(\omega) \leq r^{\varepsilon} < \gamma^*(\omega) + \varepsilon$ for arbitrary $\varepsilon > 0$. Therefore $v^*(\omega) \leq \gamma^*(\omega)$. □

Appendix D. Proof of Lemma 4.8.

LEMMA 4.8. *For the ORMP with productivity function $\eta_t(r)$, $(\mathcal{R}, \theta, \vartheta)$ has the extensibility property.*

Proof. Consider any instance $\omega = (a_1(1), a_2(1), \ldots, a_T(T))$, $\theta(\omega) = (a'_1(1), a'_2(1), \ldots, a'_T(T))$, and $\vartheta(\omega) = (a''_1(1), a''_2(1), \ldots, a''_{T-1}(T-1))$. Consider any r such that $r \in \mathcal{R}(\theta(\omega))$ and $r^{T-1} \in \mathcal{R}(\vartheta(\omega))$. As usual, available work is performed in EDF order. Let $w_u(t) \equiv a_u(1) - q_u(1) + a_u(2) - q_u(2) + \cdots + a_u(t)$, $w'_T(t) \equiv a'_T(1) - q'_T(1) + a'_T(2) - q'_T(2) + \cdots + a'_T(t)$, and $w''_u(t) \equiv a''_u(1) - q''_u(1) + a''_u(2) - q''_u(2) + \cdots + a''_u(t)$.

It is shown by induction on t that $w_u(t) = w''_u(t)$ and $q_u(t) = q''_u(t)$ for all $t = 1, \ldots, T-1$, $u = t, \ldots, T-1$, and $w'_T(t) = \sum_{u=t}^{T} w_u(t)$ and $q'_T(t) = \sum_{u=t}^{T} q_u(t)$ for all $t = 1, \ldots, T$. Clearly the hypothesis holds for $t = 1$. Suppose the hypothesis holds for t. Note that $q_t(t) = q''_t(t) = w''_t(t) = w_t(t)$ from the assumption that $r^{T-1} \in \mathcal{R}(\vartheta(\omega))$. Then $w_u(t+1) = a_u(1) - q_u(1) + \cdots + a_u(t+1) = a''_u(1) - q''_u(1) + \cdots + a''_u(t+1) = w''_u(t+1)$ for all $u = t+1, \ldots, T-1$. Because available work is performed in EDF order, $q_u(t+1) = q''_u(t+1)$ for all $u = t+1, \ldots, T-1$. Also, $w'_T(t+1) = w'_T(t) - q'_T(t) + a'_T(t+1) = \sum_{u=t}^{T} w_u(t) - \sum_{u=t}^{T} q_u(t) + \sum_{u=t+1}^{T} a_u(t+1) = \sum_{u=t+1}^{T} (w_u(t) - q_u(t) + a_u(t+1)) = \sum_{u=t+1}^{T} w_u(t+1)$, and the hypothesis has been established.

From the assumption that $r^{T-1} \in \mathcal{R}(\vartheta(\omega))$, it follows that constraints (2.6), (2.3), (2.4), and (2.5) are satisfied by $\vartheta(\omega)$ and r^{T-1} and thus by ω and r for all $t = 1, \ldots, T-1$. It remains to be shown that constraints (2.6), (2.3), (2.4), and (2.5) are satisfied by ω and r at $t = T$. From the hypothesis and the assumption that $r \in \mathcal{R}(\theta(\omega))$, it follows that $w_T(T) = w'_T(T) \leq \eta_T(r(T))$, and thus constraints (2.6), (2.3), (2.4), and (2.5) are satisfied by ω and r at $t = T$. □

Appendix E. Proof of Proposition 5.1.

The proof of Proposition 5.1 is based on the use of a strictly convex productivity function with the ORMP. It is important to the proof that our techniques for determining the optimal competitive ratio ρ_T^* can be adapted for a certain class of such functions. Those results are presented in the next section, which is followed by the proof of Proposition 5.1.

E.1. Other productivity functions for the ORMP.

In section 4.1.2, the determination of the optimal competitive ratio for the ORMP applied only to the case with $\eta_t(r) = r$ for all $t = 1, \dots, T$. In general, the analysis does not easily extend to other productivity functions, though there are a few easy cases. Assume that η_t is increasing and therefore invertible.

The general equation for $q(t)$ is

$$q(t) = \eta_t(\alpha_T v_T(\omega^t)).$$

To "solve" for α_T as we did in section 4.1.2, we would like to factor it out of the sum

$$\sum_{t=1}^{T} \eta_t(\alpha_T v_T(\omega^t)),$$

which is not possible in general, even if $\eta_t = \eta$ is stationary.

It is possible, however, for stationary productivity functions that satisfy $\eta(kr) = f(k)\eta(r)$ for some function $f(k)$ and all $k, r > 0$. This class of functions includes functions of the form $\eta(r) = cr^p$, where $p > 0$ (with $f(k) = k^p$). In these cases, we have

$$\sum_{t=1}^{T} \eta(\alpha_T v_T(\omega^t)) = f(\alpha_T) \sum_{t=1}^{T} \eta(v_T(\omega^t)).$$

Note that for these productivity functions, which are stationary and nondecreasing, one can derive from Proposition 4.1 that

$$v_T(\omega^t) = \eta^{-1}\left(\max_{\{i \in \{1,\dots,t\}\}} \frac{1}{T-i+1} \sum_{t=i}^{t} a(t) \right).$$

Therefore, $\eta(v_T(\omega^t))$ is just the maximization expression in the parentheses above.

If f is invertible (as it is for this class of functions), then one can write

$$\alpha_T \geq f^{-1}\left(\frac{\sum_{t=1}^{T} a(t)}{\sum_{t=1}^{T} \max_{\{i \in \{1,\dots,t\}\}} \frac{1}{T-i+1} \sum_{\tau=i}^{t} a(\tau)} \right).$$

Assuming f^{-1} is monotonically increasing (as it is for this class of functions), one can still use the LP formulation to find the optimal value of ρ_T^*:

$$\rho_T^* = f^{-1}\left(\frac{1}{\text{optimal solution of LP}} \right).$$

In particular, this means that *any* stationary linear productivity function ($\eta(r) = cr$) has the same ρ_T^* values as previously determined. For a stationary productivity function of the form $\eta(r) = cr^p$, ρ_T^* is the pth root of the reciprocal of the optimal LP value. We know from [6] that the limiting LP value reciprocal is $\rho^* = e$, which means that as $p \to \infty$, the corresponding value of ρ^* goes to 1. Similarly, as $p \to 0$, the corresponding value of ρ^* increases without bound.

Table E.1
Algorithm π for Proposition 5.1 *($\varepsilon = a(1)/15$).*

	Decision 1	
	Period 1	Period 2
$a(1) \leq a(2)$	$\eta^{-1}\left(\frac{2}{3}a(1) + \varepsilon\right)$	$\eta^{-1}\left(\frac{4}{3}a(2) - \varepsilon\right)$
$3\varepsilon \leq a(2) < a(1)$		$\eta^{-1}\left(\frac{2}{3}a(1) + \frac{2}{3}a(2) - \varepsilon\right)$
$a(2) < 3\varepsilon$		$\eta^{-1}\left(\frac{2}{3}a(1) + \varepsilon\right)$

Decision 2	
Period 1	Period 2
$\eta^{-1}\left(\frac{2}{3}a(1) - \varepsilon\right)$	$\eta^{-1}\left(\frac{4}{3}a(2) + \varepsilon\right)$
	$\eta^{-1}\left(\frac{2}{3}a(1) + \frac{2}{3}a(2) + \varepsilon\right)$
	$\eta^{-1}\left(\frac{2}{3}a(1) - \varepsilon\right)$

E.2. Proof of Proposition 5.1.

Proposition 5.1. *Consider the single deadline ORMP with a stationary productivity function of the form $\eta_t(r) = cr^p$, $c > 0$, $p > 1$, for all t. For this problem there exists a randomized algorithm $\pi \in \Pi^{RO}$ such that $v^{\pi}(\omega) < v^{\pi_\alpha}(\omega)$ for every $\omega \in \Omega_2$, $\omega > 0$.*

Proof. Note that $\eta(r) = cr^p$ with $c > 0$ and $p > 1$ is strictly convex and strictly increasing on $[0, \infty)$.

It follows from section E.1 that for η of this form, $\rho_2^* = f^{-1}(4/3)$, where $f(k) = k^p$, so the optimal α-policy (and thus the optimal deterministic algorithm) will allocate $f^{-1}(4/3)\eta^{-1}(a(1)/2) = \eta^{-1}(2a(1)/3)$ in the first time period. Similarly, the optimal α-policy allocates $\max\left\{\eta^{-1}\left((4/3)(a(1) + a(2))/2\right),\ \eta^{-1}\left(4a(2)/3\right)\right\}$ in the second time period.

To show that the α-policy is not optimal among all randomized online algorithms for this problem, we construct a randomized algorithm π that achieves a better value than the α-policy on any instance.

Algorithm π randomly chooses one of two decisions, each with probability 1/2. If Decision 1 is chosen, more work than under the α-policy is done in the first time period, and if decision 2 is chosen, less work than under the α-policy is done in the first time period. Specifically, algorithm π allocates resources according to Table E.1, with $\varepsilon = a(1)/15$.

Next we prove that π is always feasible and results in a better value than the α-policy.

First note that the *work attempted* under algorithm π in the first time period, by which we mean η of the resource allocated, is at most $2a(1)/3+\varepsilon = 2a(1)/3+a(1)/15 < a(1)$, so π does not attempt to do work before it arrives. We consider the following three cases that come up in the second time period: $a(2) \geq a(1)$, $3\varepsilon \leq a(2) < a(1)$, and $a(2) < 3\varepsilon$.

Consider the first two cases together. It is easy to see that in these two cases, the total work attempted is the same as the total work attempted under the α-policy. Since we already showed that all work attempted in the first time period is actually accomplished, this means that algorithm π is feasible for both of these two cases (since the α-policy is feasible).

We claim that in both these cases, the maximum resource allocation occurs in the second time period. Recall that this statement is true for the α-policy. Therefore, our only concern is Decision 1: specifically, the possibility that adding ε work to the first time period gives a greater total than subtracting ε work from the second time period. Thus it will suffice to show that $\eta(v_T(\omega^2))$, whose value depends on whether $a(2) \geq a(1)$, is at least 2ε more than $\eta(v_T(\omega^1))$, which is always $2a(1)/3$. If $a(2) \geq a(1)$, then $\eta(v_T(\omega^2)) = 4a(2)/3$ and we have

$$\frac{4}{3}a(2) - \frac{2}{3}a(1) \geq \frac{4}{3}a(1) - \frac{2}{3}a(1) = \frac{2}{3}a(1) > \frac{2}{15}a(1) = 2\varepsilon.$$

If $3\varepsilon \leq a(2) < a(1)$, then $\eta(v_T(\omega^2)) = 2a(1)/3 + 2a(2)/3$ and we have

$$\frac{2}{3}a(1) + \frac{2}{3}a(2) - \frac{2}{3}a(1) = \frac{2}{3}a(2) \geq \frac{2}{3}(3\varepsilon) = 2\varepsilon.$$

Therefore, the value of algorithm π on any instance in which $a(1) \leq a(2)$, is the expected value of the second time period's allocation, which is

$$\frac{1}{2}\eta^{-1}\left(\frac{4}{3}\frac{(a(1)+a(2))}{2} + \varepsilon\right) + \frac{1}{2}\eta^{-1}\left(\frac{4}{3}\frac{(a(1)+a(2))}{2} - \varepsilon\right) < \eta^{-1}\left(\frac{4}{3}\frac{(a(1)+a(2))}{2}\right),$$

where we have strict inequality because of the strict concavity of η^{-1} (due to the strict convexity of η). Thus, the expected value of algorithm π is always less than the value of the α-policy.

Similarly, if $3\varepsilon \leq a(2) < a(1)$, the value of algorithm π is

$$\frac{1}{2}\eta^{-1}\left(\frac{4}{3}a(2) + \varepsilon\right) + \frac{1}{2}\eta^{-1}\left(\frac{4}{3}a(2) - \varepsilon\right) < \eta^{-1}\left(\frac{4}{3}a(2)\right),$$

so algorithm π has a better value in this case as well.

It now remains to consider the third case: $a(2) < 3\varepsilon$. For feasibility, note that Decision 1 allocates more at both times than Decision 2, so it suffices to show that Decision 2 allocates enough resources to meet the deadline. The total work attempted is

$$\begin{aligned} 2\left(\frac{2}{3}a(1) - \varepsilon\right) &= \frac{4}{3}a(1) - 2\varepsilon = \frac{4}{3}a(1) - \frac{2}{15}a(1) \\ &= \frac{18}{15}a(1) = a(1) + \frac{3}{15}a(1) \geq a(1) + a(2). \end{aligned}$$

This shows that at least as much work is attempted as is available, and the work attempted in the first time period does not exceed $a(1)$, so the algorithm is feasible.

Now note that the value of algorithm π in this case is

$$\frac{1}{2}\eta^{-1}\left(\frac{2}{3}a(1) + \varepsilon\right) + \frac{1}{2}\eta^{-1}\left(\frac{2}{3}a(1) - \varepsilon\right) < \eta^{-1}\left(\frac{2}{3}a(1)\right).$$

So the value of algorithm π is strictly less than the first time period's allocation by the α-policy, which must be no more than the value of the α-policy (since the second time period's allocation is at least as great as the first time period's).

So, in all three cases, algorithm π has a value strictly less than the α-policy, which completes the proof. $\square$

REFERENCES

[1] S. Albers, *On randomized online scheduling*, in Proceedings of the 34th Annual ACM Symposium on the Theory of Computing (STOC), ACM, New York, 2002, pp. 134–143.

[2] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts, *On-line routing of virtual circuits with applications to load balancing and machine scheduling*, J. ACM, 44 (1997), pp. 486–504.

[3] Y. Azar, A. Z. Broder, and A. R. Karlin, *On-line load balancing*, Theoret. Comput. Sci., 130 (1994), pp. 73–84.

[4] Y. Azar, B. Kalyanasundaram, S. Plotkin, K. Pruhs, and O. Waarts, *Online load balancing of temporary tasks*, J. Algorithms, 22 (1997), pp. 93–110.

[5] Y. Azar, J. Naor, and R. Rom, *The competitiveness of on-line assignments*, J. Algorithms, 18 (1995), pp. 221–237.

[6] A. Bar-Noy, A. Freund, and J. Naor, *On-line load balancing in a hierarchical server topology*, in Algorithms—ESA '99, 7th Annual European Symposium, Prague, Czech Republic, July 16–18, J. Nesetril, ed., Lecture Notes in Comput. Sci. 1643, Springer-Verlag, Berlin, 1999, pp. 77–88.

[7] Y. Bartal, S. Leonardi, A. Marchetti-Spaccamella, J. Sgall, and L. Stougie, *Multiprocessor scheduling with rejection*, SIAM J. Discrete Math., 13 (2000), pp. 64–78.

[8] P. Berman, M. Charikar, and M. Karpinski, *On-line load balancing for related machines*, in Proceedings of the 5th Workshop on Algorithms and Data Structures, Lecture Notes in Comput. Sci. 1272, Springer-Verlag, Berlin, 1997, pp. 116–125.

[9] R. Fleischer and M. Wahl, *Online scheduling revisited*, in Algorithms—ESA 2000, Proceedings of the 8th Annual European Symposium, Lecture Notes in Comput. Sci. 1879, Springer-Verlag, Berlin, 2000, pp. 202–210.

[10] R. L. Graham, *Bounds for certain multiprocessing anomalies*, Bell System Tech. J., 45 (1966), pp. 1563–1581.

[11] J. A. Hoogeveen and A. P. A. Vestjens, *A best possible deterministic on-line algorithm for minimizing maximum delivery time on a single machine*, SIAM J. Discrete Math., 13 (2000), pp. 56–63.

[12] S. Irani and A. R. Karlin, *Online computation*, in Approximation Algorithms for NP-Hard Problems, D. S. Hochbaum, ed., PWS Publishing Company, Boston, MA, 1997, pp. 521–564.

[13] A. J. Kleywegt, V. S. Nori, M. W. P. Savelsbergh, and C. A. Tovey, *Dynamic and Stochastic Resource Minimization*, Tech. Report 97-05, The Logistics Institute, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 1997.

[14] A. J. Kleywegt, V. S. Nori, M. W. P. Savelsbergh, and C. A. Tovey, *Online resource minimization*, in Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms, Baltimore, MD, January 1999, SIAM, Philadelphia, 1999, pp. 576–585.

[15] L. A. McGeoch and D. D. Sleator, eds., *On-line Algorithms*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science 7, AMS, Providence, RI, ACM, New York, 1992.

[16] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, Cambridge, UK, 1995.

[17] S. Seiden, *Online randomized multiprocessor scheduling*, Algorithmica, 28 (2000), pp. 173–216.

SIAM J. DISCRETE MATH.
Vol. 16, No. 4, pp. 591–603

PROJECTIONS OF BINARY LINEAR CODES ONTO LARGER FIELDS*

JON-LARK KIM[†], KEITH E. MELLINGER[‡], AND VERA PLESS[§]

Abstract. We study certain projections of binary linear codes onto larger fields. These projections include the well-known projection of the extended Golay $[24, 12, 8]$ code onto the hexacode over GF(4) and the projection of the Reed–Muller code $R(2, 5)$ onto the unique self-dual $[8, 4, 4]$ code over GF(4). We give a characterization of these projections, and we construct several binary linear codes which have best known optimal parameters, for instance, $[20, 11, 5]$, $[40, 22, 8]$, $[48, 21, 12]$, and $[72, 31, 16]$. We also relate the automorphism group of a quaternary code to that of the corresponding binary code.

Key words. additive codes, projection onto larger fields

AMS subject classification. 94B35

DOI. 10.1137/S0895480102404367

1. Introduction. The construction of good binary (linear) codes from shorter codes has been widely studied by coding theorists. One of the main reasons in this direction is to lower the decoding complexity of the original code. The $(u|u + v)$ construction [17], the projection of Z_4-linear codes onto nonlinear binary codes [14], and the projection of codes over GF(p^m) onto codes over GF(p) are such examples. Each of these constructions applies to a large class of binary codes.

We recall a projection construction that is quite different from those mentioned above. In the mid 1980s the third author [18] showed that the Golay code of length 24 (as well as the ternary Golay code of length 12) can be easily constructed from the Hexacode of length 6 over GF(4) (resp., the tetracode of length 4 over GF(3)). It was expected [18, p. 565] that one can construct, in a somewhat analogous fashion, good large binary codes whose decoding can be reduced, in part, to the decoding of a good quaternary code. However, only a few codes had the above type of projection construction.

Recently Gaborit, Kim, and Pless [12, 16] showed that the three singly even self-dual binary $[32, 16, 8]$ codes and three of the five doubly even self-dual $[32, 16, 8]$ codes have a similar projection. The construction of Amrani and Be'ery [1] of binary Reed–Muller codes is also an interesting generalization of a projection. These projections regard a binary linear code of length $4m$ as a set of $4 \times m$ arrays and then *project* these arrays onto a quaternary code of length m. A projection onto GF(16) was suggested by Esmaeili, Gulliver, and Khandani [10] to investigate whether the $[48, 24, 12]$ quadratic residue code has such a projection.

The purpose of our paper is to give a uniform characterization of these projections. We provide many examples of binary linear codes having these projections. In particular, we construct several binary linear codes that have best known optimal

*Received by the editors March 22, 2002; accepted for publication (in revised form) February 17, 2003; published electronically July 30, 2003.
http://www.siam.org/journals/sidma/16-4/40436.html

[†] Department of Mathematics and Statistics, University of Nebraska-Lincoln, Lincoln, NE 68588-0323 (jlkim@math.unl.edu).

[‡] Department of Mathematics, Mary Washington College, Fredericksburg, VA 22401 (kmelling@mwc.edu).

[§] Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, Chicago, IL 60607-7045 (pless@math.uic.edu).

parameters, for instance, $[20,11,5]$, $[40,22,8]$, $[48,21,12]$, and $[72,31,16]$. We also relate the automorphism group of a quaternary code to that of the corresponding binary code. Sections 2 and 3 survey the basic facts about projections onto GF(4) and additive codes over GF(4). In section 4 we characterize which binary linear codes have a projection onto GF(4). In section 5 we apply results of section 4 to extremal binary self-dual codes, and section 6 discusses a projection onto GF(16). Finally, in section 7 we construct two codes having the best known parameters $[48,21,12]$ and $[72,31,16]$.

2. Projection. We begin with the projection of binary linear codes into quaternary codes (i.e., codes over GF(4)) as explained in [18]. Consider a $4 \times m$ array with zeros and ones in it. Label the four rows with the elements of GF(4): $0, 1, \omega, \overline{\omega}$. Recall that $\overline{\omega} = \omega^2$, $\overline{\omega}^2 = \omega$, and $\overline{\omega} = 1 + \omega$. If we take the inner product of a column of our array with the row labels, we obtain an element of GF(4). In this way we have a correspondence between binary vectors of length $4m$ and quaternary vectors of length m. For example, let $\mathbf{v} = (1,0,1,1,0,0,0,0,1,0,0,1,0,0,1,1,0,1,0,1,0,1,0,0,1,1,0,1,0,1,0,1,1,0)$ be the binary vector of length 32. Then

$$\mathbf{v} = \begin{array}{c cccccccc} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ \omega & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ \overline{\omega} & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ \hline & 1 & 0 & \overline{\omega} & 1 & \omega & 1 & \omega & \overline{\omega} \end{array}$$

corresponds to (or projects onto) the quaternary vector $\mathbf{w} = (1,0,\overline{\omega},1,\omega,1,\omega,\overline{\omega})$ of length 8. We denote this projection by $\mathrm{Proj}(\mathbf{v}) = \mathbf{w}$. The columns of such an array associated with vector $\mathbf{v}$ will be referred to as the *columns of* $\mathbf{v}$ and the top row of the array will be referred to as the *top row of* $\mathbf{v}$. Note that Proj is a GF(2)-linear map from the set of binary vectors of length $4m$ to the set of quaternary vectors of length m.

Let the *parity of a column* be either even or odd, respectively, if an even or an odd number of ones exists in the column. Define the *parity of the top row* in a similar fashion. Thus the first column of the 4×8 array of the above vector has odd parity, and the rest have even parity. The top row also has even parity. By a quaternary additive code $\mathcal{C}_4$ of length m we mean a set of vectors in $\mathrm{GF}(4)^m$ that is closed under addition.

DEFINITION 2.1. *Let S be a set of binary vectors of length $4m$ and $\mathcal{C}_4$ a quaternary additive code of length m. Then S is said to have* projection O onto $\mathcal{C}_4$ *if the following conditions are satisfied:*

(P1) *For any vector $\mathbf{v} \in S$, Proj$(\mathbf{v}) \in \mathcal{C}_4$. Conversely, for any vector $\mathbf{w} \in \mathcal{C}_4$, all vectors $\mathbf{v}$ such that Proj$(\mathbf{v}) = \mathbf{w}$ are in S.*

(P2) *The columns of the array of any vector of S are either all even or all odd.*

(P3) *The parity of the top row of the array of any vector of S is the same as the column parity of the array.*

It is easy to see that the above set S is in fact a binary linear code of length $4m$. It is well known [18] that the extended Golay $[24,12,8]$ code has projection O onto the $[6,3,4]$ Hexacode. The main advantage of this projection is its ability to decode a binary code by decoding the projected code. Generally this lowers the decoding complexity. Hard decision decoding by hand using this projection was done in [18] and soft decision decoding was done by several authors [7, 21, 22, 23].

The Reed–Muller $[32, 16, 8]$ code $R(2, 5)$ has a similar projection [12]. We define such a projection, called projection E, as follows.

Definition 2.2. *Using the same notation as Definition* 2.1*, S is said to have* projection E onto $\mathcal{C}_4$ *if conditions* (P1) *and* (P2)*, as well as the following third condition* (P3′)*, are satisfied:*

(P3′) *The parity of the top row of the array of any vector of S is always even.*

3. Introduction to additive codes over GF(4).

In this section we give some basic definitions and preliminaries related to additive codes, and we refer the reader to [4, 11] for more details. As before, an *additive code $\mathcal{C}_4$ over $GF(4)$ of length n* is an additive subgroup of $\mathrm{GF}(4)^n$. As $\mathcal{C}_4$ is a free GF(2)-module, it has size 2^k for some $0 \leq k \leq 2n$. We call $\mathcal{C}_4$ an $(n, 2^k)$ code. It has a basis, as a GF(2)-module, consisting of k basis vectors; a *generator matrix* of $\mathcal{C}_4$ will be a $k \times n$ matrix with entries in GF(4) whose rows form a basis of $\mathcal{C}_4$. Interest in additive codes over GF(4) has arisen because of their correspondence to quantum codes, as described in [4]. There is a natural inner product arising from the trace map. If we let $\mathrm{GF}(4) = \{0, 1, \omega, \overline{\omega}\}$, where $\overline{\omega} = \omega^2 = 1 + \omega$, the *trace* map $\mathrm{Tr} : \mathrm{GF}(4) \to \mathrm{GF}(2)$ is given by

$$\mathrm{Tr}(x) = x + x^2.$$

In particular $\mathrm{Tr}(0) = \mathrm{Tr}(1) = 0$ and $\mathrm{Tr}(\omega) = \mathrm{Tr}(\overline{\omega}) = 1$. The *conjugate* of $x \in \mathrm{GF}(4)$, denoted $\overline{x}$, is the image of x under the Frobenius automorphism; hence, $\overline{0} = 0$, $\overline{1} = 1$, and $\overline{\overline{\omega}} = \omega$. We now define the *trace inner product* of two vectors $\mathbf{x} = (x_1 x_2 \cdots x_n)$ and $\mathbf{y} = (y_1 y_2 \cdots y_n)$ in $\mathrm{GF}(4)^n$ to be

$$\mathbf{x} \star \mathbf{y} = \sum_{i=1}^{n} \mathrm{Tr}(x_i \overline{y_i}).$$

Example 3.1. Let $\mathcal{G}_6$ be the $[6, 3, 4]$ *hexacode* whose generator matrix as a linear GF(4)-code is

$$\begin{bmatrix} 1 & 0 & 0 & 1 & \omega & \omega \\ 0 & 1 & 0 & \omega & 1 & \omega \\ 0 & 0 & 1 & \omega & \omega & 1 \end{bmatrix}.$$

This is also an additive $(6, 2^6, 4)$ code; thinking of $\mathcal{G}_6$ as an additive code, we see that it has generator matrix

$$\begin{bmatrix} 1 & 0 & 0 & 1 & \omega & \omega \\ \omega & 0 & 0 & \omega & \overline{\omega} & \overline{\omega} \\ 0 & 1 & 0 & \omega & 1 & \omega \\ 0 & \omega & 0 & \overline{\omega} & \omega & \overline{\omega} \\ 0 & 0 & 1 & \omega & \omega & 1 \\ 0 & 0 & \omega & \overline{\omega} & \overline{\omega} & \omega \end{bmatrix}.$$

If $\mathcal{C}_4$ is an additive code, its *dual*, denoted $\mathcal{C}_4^\perp$, is the additive code $\{\mathbf{x} \in \mathrm{GF}(4)^n \mid \mathbf{x} \star \mathbf{c} = 0 \text{ for all } \mathbf{c} \in \mathcal{C}_4\}$. If $\mathcal{C}_4$ is an $(n, 2^k)$ code, then $\mathcal{C}_4^\perp$ is an $(n, 2^{2n-k})$ code. As usual, $\mathcal{C}_4$ is *self-orthogonal* if $\mathcal{C}_4 \subseteq \mathcal{C}_4^\perp$ and *self-dual* if $\mathcal{C}_4 = \mathcal{C}_4^\perp$. In particular, if $\mathcal{C}_4$ is self-dual, $\mathcal{C}_4$ is an $(n, 2^n)$ code. The code $\mathcal{G}_6$ in Example 3.1 is self-dual as an additive code. (Any GF(4)-linear code that is self-orthogonal under the Hermitian inner product is a self-orthogonal additive code under the trace inner product.)

As usual, the *weight* $\mathrm{wt}(\mathbf{c})$ of $\mathbf{c} \in \mathcal{C}_4$ is the number of nonzero components of $\mathbf{c}$. The minimum weight d of $\mathcal{C}_4$ is the smallest weight of any nonzero codeword in $\mathcal{C}_4$. If $\mathcal{C}_4$ is an $(n, 2^k)$ additive code of minimum weight d, $\mathcal{C}_4$ is called an $(n, 2^k, d)$ code. We say $\mathcal{C}_4$ is *Type II* if $\mathcal{C}_4$ is self-dual and all codewords have even weight. It can be shown that Type II codes of length n exist if and only if n is even [11]. If $\mathcal{C}_4$ is self-dual but some codeword has odd weight (in which case the code cannot be GF(4) linear), we say the code is *Type I* (see [20, section 4.2]). There exists a bound on the minimum weight of an additive self-dual code [20, Theorem 33]. If d_I and d_{II} are the minimum distances of additive self-dual Type I and Type II codes, respectively, of length $n > 1$, then

$$d_I \leq \begin{cases} 2\left\lfloor \frac{n}{6} \right\rfloor + 1 & \text{if } n \equiv 0 \pmod 6, \\ 2\left\lfloor \frac{n}{6} \right\rfloor + 3 & \text{if } n \equiv 5 \pmod 6, \\ 2\left\lfloor \frac{n}{6} \right\rfloor + 2 & \text{otherwise,} \end{cases} \tag{3.1}$$

$$d_{II} \leq 2\left\lfloor \frac{n}{6} \right\rfloor + 2. \tag{3.2}$$

A code that meets the appropriate bound is called *extremal.* Note that (3.2) is the same as saying that $d = 2m + 2$ if $n = 6m + 2(i - 1)$, with $i = 1, 2$, or 3. Type II codes meeting the bound d_{II} have a unique weight enumerator. This property is not true for Type I extremal codes. A self-dual (with respect to the Hermitian inner product) linear code over GF(4) also satisfies bound (3.2), and an extremal code is a $[6m, 3m, 2m + 2]$ code.

We say that two additive codes $\mathcal{C}_4$ and $\mathcal{C}_4'$ are *equivalent* provided there is a map sending the codewords of $\mathcal{C}_4$ onto the codewords of $\mathcal{C}_4'$, where the map consists of a permutation of coordinates, followed by a scaling of coordinates by elements of GF(4), possibly followed by conjugation of some of the coordinates. Notice that permuting coordinates, scaling coordinates, and conjugating some coordinates of a self-orthogonal (or self-dual) code do not change self-orthogonality (or self-duality). The *automorphism group* of $\mathcal{C}_4$, denoted Aut($\mathcal{C}_4$), consists of all bijections on codewords in $\mathcal{C}_4$ to codewords in $\mathcal{C}_4$, which permute coordinates, scale coordinates, and conjugate coordinates.

4. Projection of binary linear codes onto GF(4).

In this section we characterize binary linear codes of length $4m$ having projection O or projection E onto GF(4). We let $\mathcal{C}$ (resp., $\mathcal{C}'$) be the set of binary vectors satisfying (P2) and (P3) (resp., (P2) and (P3$'$)). A standard counting argument shows that $\mathcal{C}$ (resp., $\mathcal{C}'$) is a linear $[4m, 3m]$ code. If we look at all the vectors in $\mathcal{C}$ that project to the zero vector, we obtain a subcode of $\mathcal{C}$, which we denote $\mathcal{D}$. The subcode $\mathcal{D}$ is generated by all even sums of weight 4 vectors, all of whose ones appear in the same column together with the one additional vector $f_1 = (1000\ 1000\ \cdots\ 1000\ 1000)$ if m is odd, or $f_2 = (1000\ 1000\ \cdots\ 1000\ 0111)$ if m is even. Similarly $\mathcal{C}'$ has such a subcode $\mathcal{D}$, which contains f_1 when m is even and contains f_2 when m is odd. A counting argument again shows that $\mathcal{D}$ has dimension m.

LEMMA 4.1. *Let $\mathcal{D}$ and $\mathcal{C}$ ($\mathcal{C}'$) be defined as above. Let $\mathbf{v}_1$ and $\mathbf{v}_2$ be two vectors in $\mathcal{C}$ (resp., $\mathcal{C}'$) such that $\mathbf{v}_1 \not\equiv \mathbf{v}_2 \pmod{\mathcal{D}}$. Then Proj($\mathbf{v}_1$) $\neq$ Proj($\mathbf{v}_2$).*

It easily follows that there is a one-to-one correspondence between the cosets of $\mathcal{D}$ in $\mathcal{C}$ ($\mathcal{C}'$) and GF$(4)^m$ given by Proj($\mathbf{v} + \mathcal{D}$) = Proj($\mathbf{v}$).

LEMMA 4.2. *Let $\mathcal{C}_2$ be a binary linear subcode of $\mathcal{C}$ that also contains the subcode $\mathcal{D}$. Suppose that there are r linearly independent vectors $\mathbf{v}_{m+1}, \ldots, \mathbf{v}_{m+r}$ in $\mathcal{C}_2$ such that any nontrivial linear combination of them is not in $\mathcal{D}$. Then Proj($\mathbf{v}_{m+1}$), ..., Proj($\mathbf{v}_{m+r}$) are linearly independent over GF(2).*

We can now give a characterization of a binary linear code $\mathcal{C}_2$ of length $4m$ that has either projection O or projection E onto an additive code over GF(4). The following results are easy to prove and will be used in future arguments.

PROPOSITION 4.3. *Let $\mathcal{C}_2$ be a binary linear $[4m, k, d]$ code with projection O (or projection E) onto an additive code $\mathcal{C}_4$ over $GF(4)$. Then*

1. *$d \le d(\mathcal{D}) \le 8$, where $d(\mathcal{D})$ is the minimum weight of $\mathcal{D}$, and $\mathcal{C}_4$ has dimension $r = k - m \ge 0$ over $GF(2)$;*
2. *there exist $(k-m)$ linearly independent vectors $\mathbf{v}_{m+1}, \ldots, \mathbf{v}_{m+(k-m)} = \mathbf{v}_k$ of $\mathcal{C}_2$ whose projection forms a basis for $\mathcal{C}_4$ as an additive code;*
3. *the vectors in part 2 above can be chosen so that $wt(\mathbf{v}_i) = 2wt(Proj(\mathbf{v}_i))$ for $i = m+1, \ldots, k$, and $wt(\mathbf{v}_i \cap \mathbf{v}_j) \equiv Proj(\mathbf{v}_i) \star Proj(\mathbf{v}_j) \pmod 2$ for $m+1 \le i, j \le k, i \ne j$.*

Proof. We prove only the projection O case. Clearly $d \le d(\mathcal{D}) \le 8$, as $\mathcal{D}$ is a subcode of $\mathcal{C}_2$. Since $\mathcal{C}_2$ has dimension k and $\mathcal{D}$ has dimension m, we know there exist $k - m$ linearly independent vectors $\mathbf{v}_{m+1}, \ldots, \mathbf{v}_{m+(k-m)} = \mathbf{v}_k$ in $\mathcal{C}_2$ such that any nontrivial linear combination of them is not in $\mathcal{D}$. Hence, by Lemma 4.2, $\mathrm{Proj}(\mathbf{v}_{m+1}), \ldots, \mathrm{Proj}(\mathbf{v}_k)$ are linearly independent over GF(2). Therefore $\mathcal{C}_4$ has dimension $k - m$ over GF(2) with basis $\{\mathrm{Proj}(\mathbf{v}_{m+1}), \ldots, \mathrm{Proj}(\mathbf{v}_k)\}$. This proves parts 1 and 2.

We can assume that the columns of the above $k - m$ linearly independent vectors $\mathbf{v}_{m+1}, \ldots, \mathbf{v}_k$ all have even parity by adding $f_1(m : \text{odd})$ or $f_2(m : \text{even})$ to those vectors of odd column parity. Furthermore, we may assume that the top row of each vector $\mathbf{v}_{m+1}, \ldots, \mathbf{v}_k$ consists of zeros of length m by adding proper codewords from $\mathcal{D}$. Hence, the columns of any vector from $\mathbf{v}_{m+1}, \ldots, \mathbf{v}_k$ have only one of the following four forms: $(0000), (0011), (0101), (0110)$. Thus for $m + 1 \le i, j \le k$, and $i \ne j$, $\mathrm{wt}(\mathbf{v}_i \cap \mathbf{v}_j) \equiv \mathrm{Proj}(\mathbf{v}_i) \star \mathrm{Proj}(\mathbf{v}_j) \pmod 2$ and $\mathrm{wt}(\mathbf{v}_i) = 2\mathrm{wt}(\mathrm{Proj}(\mathbf{v}_i))$, $i = m + 1, \ldots, k$. This proves part 3. □

We give an explicit construction of a binary linear code, which has projection O or projection E onto a given additive code $\mathcal{C}_4$. Suppose now that $\mathcal{C}_4$ is an additive $(m, 2^r)$ code, and let $\widehat{\mathcal{C}_4}$ be the binary linear $[4m, r]$ code obtained from $\mathcal{C}_4$ by replacing each $GF(4)$ component with a 4-tuple in $\mathrm{GF}(2)^4$ as follows : $0 \to 0000$, $1 \to 0011$, $\omega \to 0101$, $\overline{\omega} \to 0110$.

Construction O: $\rho_O(\mathcal{C}_4) = \widehat{\mathcal{C}_4} + \mathcal{D}$, where $\mathcal{D}$ contains f_1 when m is odd and f_2 when m is even.

Construction E: $\rho_E(\mathcal{C}_4) = \widehat{\mathcal{C}_4} + \mathcal{D}$, where $\mathcal{D}$ contains f_2 when m is odd and f_1 when m is even.

The above constructions were known [11] for additive self-dual codes. The next result follows from Proposition 4.3.

COROLLARY 4.4. *Let $\mathcal{C}_4$ be an additive $(m, 2^r)$ code with $0 \le r \le 2m$. Then,*

1. *$\rho_O(\mathcal{C}_4)$ and $\rho_E(\mathcal{C}_4)$ are binary linear $[4m, m + r]$ codes having projection O and projection E onto $\mathcal{C}_4$, respectively.*
2. *Any binary linear code having projection O or projection E onto $\mathcal{C}_4$ can be constructed in this way.*

Next we consider the natural question of whether two equivalent additive codes could be constructed from two inequivalent binary linear codes via projection O or projection E. We label the positions in a 4-tuple with the integers 1, 2, 3, and 4. With this notation, under the above mapping of each GF(4) component to a 4-tuple in $\mathrm{GF}(2)^4$, the multiplication of $x \in \mathrm{GF}(4)$ by ω corresponds to the cycle permutation (234) of each binary 4-tuple of x. Also the conjugation of $x \in \mathrm{GF}(4)$ corresponds to the transposition (34) of the binary 4-tuple of x. Trivially the permutation of coordinates of additive codes corresponds to the column permutation of their associated binary arrays. Hence we have shown the following.

Table 4.1
Projection of binary linear codes onto GF(4).

(m,r)	Linear codes over GF(4) [3]	Parameters for binary codes via construction O or E	Highest minimum weight d_B [3]
$(7,8)$	$[7,4,3]$	$[28,15,6]$	$d_B=6$
$(8,10)$	$[8,5,3]$	$[32,18,6]$	$d_B=6-7$
$(9,12)$	$[9,6,3]$	$[36,21,6]$	$d_B=7-8$
$(10,14)$	$[10,7,3]$	$[40,24,6]$	$d_B=7-8$
$(7,6)$	$[7,3,4]$	$[28,13,7]$	$d_B=8$
$(8,8)$	$[8,4,4]$	$[32,16,8]$	$d_B=8$
$(9,10)$	$[9,5,4]$	$[36,19,8]$	$d_B=8$
$(10,12)$	$[10,6,4]$	$[40,22,8]$	$d_B=8$
$(11,14)$	$[11,7,4]$	$[44,25,8]$	$d_B=8-9$
$(12,16)$	$[12,8,4]$	$[48,28,8]$	$d_B=8-10$
$(13,18)$	$[13,9,4]$	$[52,31,8]$	$d_B=8-10$
$(14,20)$	$[14,10,4]$	$[56,34,8]$	$d_B=8-10$
$(15,22)$	$[15,11,4]$	$[60,37,8]$	$d_B=8-10$
$(16,24)$	$[16,12,4]$	$[64,40,8]$	$d_B=9-11$
$(17,26)$	$[17,13,4]$	$[68,43,8]$	$d_B=9-12$

Lemma 4.5. *Let $\mathcal{C}_4$ and $\mathcal{C}_4'$ be additive codes that are equivalent via maps defined in section* 3. *Then $\rho_O(\mathcal{C}_4)$ and $\rho_O(\mathcal{C}_4')$ are equivalent by some coordinate permutation. Similarly $\rho_E(\mathcal{C}_4)$ and $\rho_E(\mathcal{C}_4')$ are equivalent.*

Corollary 4.6. *Let $\mathcal{C}_4$ be an additive code. The automorphism group of $\mathcal{C}_4$ is isomorphic to a subgroup of the automorphism group of $\rho_O(\mathcal{C}_4)$ (resp., $\rho_E(\mathcal{C}_4)$).*

4.1. Examples.

Example 4.7. Let P_5 be the Pentacode [21], an additive self-dual $(5,2^5,3)$ code over GF(4). Ran and Snyders [21, Lemma 4] showed that a binary linear $[20,10,5]$ code P_{20}^b has projection O onto P_5. If we define $P_{20}^c=\rho_E(P_5)$, then P_{20}^c is also a binary linear $[20,10,5]$ code. The software package Magma [5] was used to show that P_{20}^b and P_{20}^c have the same weight distribution and isomorphic automorphism groups of order 1920 and that they are not equivalent. We remark that P_{20}^b and P_{20}^c have minimum weight, which is one less than the optimal [3] binary $[20,10,6]$ codes.

Example 4.8. Consider the case when $m=5$. There exists a linear $[5,3,3]$ code $\mathcal{C}_4$ over GF(4) [3]. It has parameters $(5,2^6,3)$ as an additive code. By Corollary 4.4, $\rho_O(\mathcal{C}_4)$ and $\rho_E(\mathcal{C}_4)$ are both binary linear $[20,11]$ codes. It is not difficult to prove that the minimum weights of these binary codes is 5. It is known [3] that binary $[20,11,5]$ codes are optimal. Hence we have shown that some such codes have projection O or projection E.

Example 4.9. Let $\mathcal{C}_4$ be any additive $(m,2^r,3)$ code, where $m\geq 6$. Then $\rho_O(\mathcal{C}_4)$ and $\rho_E(\mathcal{C}_4)$ have minimum weight 6. In this way we obtain optimal binary $[28,15,6]$ codes having projection O or projection E onto a linear $[7,4,3]$ code over GF(4). See Table 4.1 for more codes, where the fourth column denotes the highest minimum weight of the corresponding binary $[n,k]$ code together with the theoretical upper bound.

Example 4.10. Consider the case when $m=10$. There exists a linear $[10,6,4]$ code $\mathcal{C}_4$ over GF(4) [3]. It has parameters $(10,2^{12},4)$ as an additive code. By Corollary 4.4, both $\rho_O(\mathcal{C}_4)$ and $\rho_E(\mathcal{C}_4)$ are binary linear $[40,22]$ codes. We want to show that the minimum weight of these binary codes is 8 in order to obtain optimal [3] binary $[40, 22, 8]$ codes. Without loss of generality, let $\mathbf{w}$ be a codeword in $\mathcal{C}_4$ whose first four coordinates are nonzero. Such a vector $\mathbf{w}$ necessarily exists, as the minimum weight of $\mathcal{C}_4$ is 4. Then in the case of even parity columns, the columns corresponding to the nonzero coordinates each contain two 1's. In the case of odd parity columns, there is at least one 1 in *every* column. Hence the minimum weight of $\rho_O(\mathcal{C}_4)$ and $\rho_E(\mathcal{C}_4)$ is 8. We have shown that there exist binary optimal $[40, 22, 8]$ codes that have projection O or projection E.

Generalizing this example, let $\mathcal{C}_4$ be any additive $(m, 2^r, 4)$ code, where $m \geq 7$. Then (i) if $m = 7$, the minimum weight of $\rho_O(\mathcal{C}_4)$ and $\rho_O(\mathcal{C}_4)$ is 7 and (ii) if $m \geq 8$, the minimum weight of $\rho_O(\mathcal{C}_4)$ and $\rho_O(\mathcal{C}_4)$ is 8. We get several optimal binary codes having projection O or projection E on $\mathcal{C}_4$. See Table 4.1 for more examples.

5. Projections of binary self-dual codes onto GF(4). In this section, we characterize binary self-dual codes of length $8k$ that have either projection O or projection E. The following proposition will be useful when we determine which binary self-dual codes have projection O or projection E.

PROPOSITION 5.1. *Let $\mathcal{C}_2$ be a binary self-dual $[4m, 2m, d]$ code with projection O (or projection E) onto a quaternary additive code $\mathcal{C}_4$. Then*

1. *m is even.*
2. *$\mathcal{C}_4$ has dimension m over $GF(2)$.*
3. *$\mathcal{C}_4$ is a self-dual code under the trace inner product. Furthermore when $\mathcal{C}_2$ is doubly even, $\mathcal{C}_4$ is even.*

Proof. We prove the claim only for projection O. By definition, $\mathcal{D}$ is a subcode of $\mathcal{C}_2$. We take f_1 or f_2 in $\mathcal{D}$, depending on (P3). Since $\mathcal{C}_2$ is self-dual, $\text{wt}(f_1)$ and $\text{wt}(f_2)$ are even. As $\text{wt}(f_1) = m$ and $\text{wt}(f_2) = m + 2$, it follows that m is even. This proves part 1. Part 2 follows from part 1 of Proposition 4.3. Part 3 follows from part 3 of Proposition 4.3. □

We can say a little more about the relationship between the automorphism group of an even additive code $\mathcal{C}_4$ and its associated binary linear code in the case when the binary linear code is self-orthogonal.

PROPOSITION 5.2. *Let $\mathcal{C}_4$ be an even additive $(m, 2^r)$ code that lifts to a self-orthogonal binary linear code $\mathcal{C}_2$ of length $4m$ via construction O or E given above. Then $Aut(\mathcal{C}_2)$ contains a subgroup of order 2^r, which is not induced by a subgroup of $Aut(\mathcal{C}_4)$.*

Proof. We consider only construction E, as the proof for construction O is similar. Let $\mathbf{v}$ be a vector of $\widehat{\mathcal{C}_4}$ whose columns have even parity. We associate a unique coordinate permutation $p_{\mathbf{v}}$ with the vector $\mathbf{v}$ in the following way. If a column of $\mathbf{v}$ contains all 1's or all 0's, then every position in that column is fixed under $p_{\mathbf{v}}$. If a column of $\mathbf{v}$ contains exactly two 1's, then the permutation $p_{\mathbf{v}}$ interchanges the coordinate positions in that column which contain 1's and also interchanges the coordinate positions which contain 0's. For instance, the permutation associated with the vector

$$\mathbf{v} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

is given by the coordinate permutation $(1, 2)(3, 4)(9, 12)(10, 11)(13, 14)(15, 16)(17, 19)$ $(18, 20)$. We claim that such a coordinate permutation leaves the code invariant and

TABLE 5.1
Automorphism group orders of some self-dual codes.

$\mathcal{C}$	$\lvert\mathrm{Aut}(\mathcal{C})\rvert$	$\lvert\mathrm{Aut}(\rho_E(\mathcal{C}))\rvert$	$\lvert\mathrm{Aut}(\rho_O(\mathcal{C}))\rvert$
$\mathcal{G}_6$	$2^4 \cdot 3^3 \cdot 5$	$2^{10} \cdot 3^3 \cdot 5$	$2^{10} \cdot 3^3 \cdot 5 \cdot 7 \cdot 11 \cdot 23$
$\mathcal{C}_1$	$2^7 \cdot 3^2$	$2^{15} \cdot 3^2 \cdot 5 \cdot 7$	$2^{15} \cdot 3^2$
$\mathcal{C}_2$	$2^4 \cdot 3 \cdot 7$	$2^{12} \cdot 3 \cdot 7$	$2^{12} \cdot 3 \cdot 7$
$\mathcal{C}_3$	$2^7 \cdot 3^2 \cdot 7$	$2^{15} \cdot 3^2 \cdot 5 \cdot 7 \cdot 31$	$2^{15} \cdot 3^2 \cdot 7$

hence is part of the full automorphism group of the binary linear code $\rho_E(\mathcal{C}_4)$. Hence, we need to show that the image of any codeword under such a permutation is still in the code.

Let $\mathbf{w}$ be any binary codeword in $\rho_E(\mathcal{C}_4)$ with even column parity and let $p_{\mathbf{v}}$ be a permutation associated with vector $\mathbf{v}$ as above. If $\mathbf{w}$ is fixed under $p_{\mathbf{v}}$, then we are done. Otherwise, we consider the columns of $\mathbf{w}$ whose coordinates are not fixed under $p_{\mathbf{v}}$. Let c_i be any such column of $\mathbf{w}$. Then c_i contains exactly two 1's and, since this column of $\mathbf{w}$ is not fixed under $p_{\mathbf{v}}$, we know that c_i meets the corresponding column of $\mathbf{v}$ in exactly one position. Because of the self-orthogonality condition, there must be another column of $\mathbf{w}$, say c_j, with the same property. Letting $d_{i,j}$ be the element of the subcode $\mathcal{D}$ with all 1's in the ith and jth columns and 0's everywhere else, we see that the action of $p_{\mathbf{v}}$ on the ith and jth columns of $\mathbf{w}$ is the same as adding $d_{i,j}$ to $\mathbf{w}$. We conclude that the image of the codeword $\mathbf{w}$ under the coordinate permutation $p_{\mathbf{v}}$ is equal to $\mathbf{w} + \mathbf{d}$, where $\mathbf{d}$ is some element of $\mathcal{D}$.

Now let $\mathbf{u}$ be any binary codeword in $\rho_E(\mathcal{C}_4)$ with odd column parity. Any such vector can be written as $f_1 + \mathbf{w}$ for some vector $\mathbf{w}$ with even column parity. Hence, it is sufficient to check that the image of f_1 under $p_{\mathbf{v}}$ is still in the code $\mathcal{C}_2$. Since $\mathcal{C}_4$ is an even code, the action of $p_{\mathbf{v}}$ on f_1 will only permute the positions in an even number of columns of f_1. Let $\mathbf{d}_{\mathbf{v}}$ be the element of $\mathcal{D}$ that has all 1's in the columns where $\mathbf{v}$ has weight 2. Then, one can easily check that the image of f_1 under the permutation $p_{\mathbf{v}}$ is equal to $f_1 + \mathbf{v} + \mathbf{d}_{\mathbf{v}}$.

Hence, we have shown that the permutation $p_{\mathbf{v}}$ leaves the code $\rho_E(\mathcal{C}_4)$ invariant. Note that any nontrivial permutation as described above does not permute columns, but does permute the top position of any column on which it does not act trivially. This shows that every such permutation cannot be induced from an element of $Aut(\mathcal{C}_4)$. Since the number of codewords of $\widehat{\mathcal{C}_4}$ is exactly 2^r, this completes the proof. □

Note that the action of a permutation $p_{\mathbf{v}}$ on a particular column can be viewed as an element of the Klein 4 group, that is, a cycle permutation corresponding to (1,2)(3,4), (1,3)(2,4), or (1,4)(2,3). This observation can be used to show that for any two permutations $p_{\mathbf{v_1}}$ and $p_{\mathbf{v_2}}$, the composition gives the permutation $p_{\mathbf{v_1}+\mathbf{v_2}}$.

COROLLARY 5.3. *Let $\mathcal{C}_4$ be an even additive $(m, 2^r)$ code that lifts to a self-orthogonal binary linear code $\mathcal{C}_2$ of length $4m$ via construction O or E given above. Then $2^r \cdot |Aut(\mathcal{C}_4)|$ divides $|Aut(\mathcal{C}_2)|$.*

We note that this result about automorphism groups partially explains the size of the automorphism groups of the binary codes given in Table 5.1, which originally appeared in [11, section 5, p. 149].[1] Here, $\mathcal{G}_6$ is the $(6, 2^6, 4)$ hexacode, and $\mathcal{C}_1, \mathcal{C}_2$, and $\mathcal{C}_3$ are the three $(8, 2^8, 4)$ Type II codes. Note that the orders of the binary linear codes all satisfy the relationship given in the corollary above. In fact, the entire automorphism group is completely determined for those cases in which the binary

[1] Table reprinted with permission of the American Mathematical Society, Providence, RI.

code is singly even. This is the case for only one of the doubly even codes, namely $\rho_E(\mathcal{C}_2)$.

5.1. Examples. In the following we consider an extremal Type II self-dual $[8k, 4k, 4\lfloor\frac{n}{24}\rfloor+4]$ code.

Example 5.4. When $k=1$ we get the unique Hamming $[8,4,4]$ code $\mathcal{H}_3$. Let i_2 be the self-dual linear $[2,1,2]$ code over GF(4) with generator matrix $[1\ \ 1]$. Then the set of vectors satisfying conditions (P1), (P2), and (P3) with $\mathcal{C}_4 = i_2$ in Definition 2.1 gives $\mathcal{H}_3$. In other words, $\mathcal{H}_3$ has projection O onto i_2.

Example 5.5. When $k=2$, there are exactly two Type II $[16,8,4]$ binary codes $A_8 \oplus A_8$ and E_{16} in the notation of [19]. By using exactly two Type II additive quaternary $(4,2^4,2)$ codes from [15, Table 1] or [13], we see that $A_8 \oplus A_8$ and E_{16} have projection E onto $(4,2^4,2)$ codes. The Type I $[16,8,4]$ binary code F_{16} has projection E onto the Type I $(4,2^4,2)$ code from [15, Table 2] or [13].

Example 5.6. When $k=3$, it is well known [18] that the extended Golay code has projection O onto the hexacode. If we consider projection E onto the hexacode, we get the Type I $[24,12,6]$ code [12].

Example 5.7. When $k=4$, we consider the five Type II $[32,16,8]$ codes given in [6]. Several authors [1, 11, 12, 24] are interested in a projection construction for some of these codes. It is known [11, Example 5.4] that applying construction E to the three Type II additive $(8,2^8,4)$ codes produces three of these five, i.e., $2g_{16}, 8f_4$, and r_{32} in the notation of [6].

It is claimed in [24] that the extended quadratic residue code q_{32} has projection O onto a quaternary linear $[8,4,4]$ code B given by Yuan, Chen, and Ma [24, p. 410]. In an example, they construct a singly even [32,16,8] code, which they claim is the quadratic residue code. However, the latter code is doubly even. Their example contains a weight 14 vector, which was claimed to be in q_{32}. We note that the code B in [24, p. 410] is equivalent to the unique linear self-dual $[8,4,4]$ code over GF(4) with generator matrix of the binary Hamming $[8,4,4]$ code. So the set of vectors in [24, Definition 1] is actually r_{32}, one of the three Type I $[32,16,8]$ codes given in [8].

Furthermore we prove here that q_{32} does not have projection E onto an additive code over GF(4). It is easy to see that Type II $[32,16,8]$ codes do not have projection O.

PROPOSITION 5.8. *Exactly three Type II* $[32,16,8]$ *codes out of the five Type II codes, namely* $2g_{16}, 8f_4$, *and* r_{32}, *have projection E onto the three Type II additive* $(8,2^8,4)$ *codes.*

Proof. Let $\mathcal{C}_2$ be one of the five Type II $[32,16,8]$ codes which have projection E onto one of the three Type II additive $(8,2^8,4)$ codes. Then by part 1 and part 2 of Corollary 4.4, we note that at most three Type II $[32,16,8]$ codes are constructed. From the discussion in Example 5.7, these three codes are in fact $2g_{16}, 8f_4$, and r_{32}. This completes the proof. □

There is an alternative way to prove Proposition 5.8. Suppose that $\mathcal{C}_2$ is one of the five Type II $[32,16,8]$ codes which have projection E onto one of the three Type II additive $(8,2^8,4)$ codes. By Proposition 4.3, $\mathcal{C}_2$ contains the set $\mathcal{D}_0$ of all even sums of weight 4 vectors with all four 1's in a column. The set $\mathcal{D}_0$ gives rise to an *octet*, that is, a weight 4 coset of $\mathcal{C}_2$ containing exactly eight weight 4 vectors (see [8, p. 1328]). As codes q_{32} and $16f_2$ have no octets while codes $2g_{16}, 8f_4$, and r_{32} have one or more [8], the above proposition follows.

Example 5.9. For $k=5$, there are at most 19 Type II $[40,20,8]$ codes having projection E onto additive $(10,2^{10},4)$ codes, as there are exactly 19 Type II $(10,2^{10},4)$

codes given in [2, 11].

6. Projections of binary codes onto GF(16). So far we have investigated projections of binary linear codes onto GF(4) using arrays with four rows. It is natural to consider a generalization to arrays with more rows. In this case we need other field extensions of GF(2) apart from GF(4). Esmaeili, Gulliver, and Khandani [10] first studied a projection of binary linear codes onto GF(16) as follows.

Let GF(16) be generated by α such that $\alpha^4 + \alpha + 1 = 0$, where α is a primitive element of GF(16). We write a binary vector of length $6m$ as a $6 \times m$ array whose rows are indexed by $0, 1, \alpha, \alpha^2, \alpha^3, \beta$, where $\beta = \alpha^{12} = 1 + \alpha + \alpha^2 + \alpha^3$. As before, we take the inner product of a column of our array with the row labels, producing an element of GF(16). It is easy to see that for any element x in GF(16), there are exactly two columns of odd parity and two columns of even parity which project to x. For example, let $x = \alpha^4$. Then $(111000)^t$, and its complement are two odd columns projecting to α^4. Similarly $(011000)^t$, and its complement are two even columns projecting to α^4.

Now we can define projection O and projection E onto GF(16) as we defined them onto GF(4) in section 2. It is clear that the binary $[48, 24, 12]$ quadratic residue code q_{48} does not have projection O or projection E onto any additive code over GF(4) since the minimum weight of q_{48} is greater than 8. It is also shown [10, Theorem 2] by computer search that q_{48} does not have projection O onto any linear code over GF(16). We show this without a computer search. Suppose that q_{48} has projection O or projection E onto an additive code of length 8 over GF(16). Then q_{48} would have a subcode generated by all even sums of weight 6 vectors, all of whose ones appear in the same column. This subcode gives rise to a weight 6 coset of q_{48} containing exactly eight weight 6 vectors. However, it is known [9, Table I] that there is no such coset of q_{48}. Therefore q_{48} cannot have projection O or projection E onto any additive code of length 8.

Furthermore we can prove that any $[48, 24, 12]$ binary code $\mathcal{C}_2$ does not have projection O or projection E onto an additive code of length 8 over GF(16). If it did, then $\mathcal{C}_2$ would be projected onto an additive $(8, 2^{16}, d \geq 6)$ code $\mathcal{C}_{16}$ over GF(16). It is well known [3, p. 299] that any q-ary (n, M, d) code has at most q^{n-d+1} vectors in it. Applying this to $\mathcal{C}_{16}$ we get $2^{16} \leq 16^{8-d+1}$, so $d \leq 5$. This is a contradiction.

PROPOSITION 6.1. *No binary* $[48, 24, 12]$ *code has projection O or projection E onto GF*(16).

7. Projections of codes with large minimum weight. We note that projection O and projection E are very useful when the minimum weight of the binary code is at most 8. In what follows, we generalize projection E so that we can construct a binary $[48, 21, 12]$ code and a $[72, 31, 16]$ code which both have a projection onto an additive GF(4) code. Interestingly, these codes are optimal [3].

Apart from the projection of a binary 4-tuple to an element of GF(4) from section 2, we recall two other maps **TOP** and **PAR** defined in [1, p. 2562]. **TOP** is the mapping of a binary 4-tuple (v_1, v_2, v_3, v_4) to v_1. **PAR** is the mapping of a binary 4-tuple (v_1, v_2, v_3, v_4) to $v_1 + v_2 + v_3 + v_4$. Both of these maps are linear. We extend these maps onto a $4 \times m$ binary array, operating on every column of the array.

Under this notation, we define a projection as follows.

DEFINITION 7.1. *Let S be a set of binary vectors of length $4m$ written as $4 \times m$ arrays as before. Let $\mathcal{P}$ and $\mathcal{T}$ be binary codes of length m and let $\mathcal{C}_4$ be a quaternary additive code of length m. Then S is said to have* projection G onto $\mathcal{C}_4$ *if the following conditions are satisfied:*

(G1) *For any vector* $\mathbf{v} \in S$, $Proj(\mathbf{v}) \in \mathcal{C}_4$*. Conversely, for any vector* $\mathbf{w} \in \mathcal{C}_4$*, all vectors* $\mathbf{v}$ *such that* $Proj(\mathbf{v}) = \mathbf{w}$ *are in* S*.*
(G2) **PAR** *of any vector of* S *is in* $\mathcal{P}$*.*
(G3) **TOP** *of any vector of* S *is in* $\mathcal{T}$*.*
We call codes $\mathcal{P}$ *and* $\mathcal{T}$ *a* parity code *and a* top code*, respectively.*

Taking the parity code as the repetition $[m, 1, m]$ code and the top code as the even $[m, m-1, 2]$ code, projection G is the same as projection E. Now we give properties of projection G. Since its proof is similar to that of Proposition 4.3, we omit the details.

PROPOSITION 7.2. *Let* $\mathcal{C}_2$ *be a binary linear* $[4m, k, d]$ *code with projection G onto an additive code* $\mathcal{C}_4$ *over* $GF(4)$*. Let* $\mathcal{P}$ *be a parity code with dimension* k_1 *and* $\mathcal{T}$ *a top code with dimension* k_2*. Then*

1. $\mathcal{C}_4$ *has dimension* $r = k - (k_1 + k_2) \geq 0$ *over* $GF(2)$*.*
2. *There exist* r *linearly independent vectors* $\mathbf{v}_{k_1+k_2+1}, \ldots, \mathbf{v}_{k_1+k_2+r} = \mathbf{v}_k$ *of* $\mathcal{C}_2$ *whose projection forms a basis for* $\mathcal{C}_4$ *as an additive code.*

We remark that part 3 of Proposition 4.3 does not hold in general.

7.1. Examples.

Example 7.3. It was shown in Theorem 1 and Corollary 2 of [1] that the binary Reed–Muller $R(r, m)$ code, where $r \geq 1$ and $m > r + 1$, has a projection onto $R(r - 1, m - 2)$ over GF(4). This fact can be described in terms of projection G by taking $\mathcal{C}_2 = R(r, m)$, $\mathcal{C}_4 = R(r - 1, m - 2)$, $\mathcal{P} = R(r - 2, m - 2)$, and $\mathcal{T} = R(r, m - 2)$. In the case of the first-order Reed–Muller $R(1, m)$ code for $m > 2$, we understand $\mathcal{P}$ as the zero code of length 2^{m-2}.

Example 7.4. We will construct a binary $[48, 21, 12]$ code having projection G onto the unique self-dual additive $(12, 2^{12}, 6)$ code over GF(4) called the dodecacode [4, 11]. For the top code, we consider a binary optimal $[12, 8, 3]$ code, which is easy to construct. We also take the repetition $[12, 1, 12]$ code as the parity code. Then by Proposition 7.2 we construct a binary $[48, 21, 12]$ code having projection G onto the dodecacode. See Table 7.1 for the generator matrix of the binary $[48, 21, 12]$ code and Table 7.2 for its weight distribution. This code has an automorphism group of order 2 generated by the following transposition found by Magma:

$$(1, 29)(2, 31)(3, 30)(4, 32)(5, 33)(6, 36)(7, 35)(8, 34)(9, 25)(10, 26)(11, 28)$$
$$(12, 27)(13, 21)(14, 24)(15, 22)(16, 23)(37, 45)(38, 48)(39, 46)(40, 47)$$

Example 7.5. We can similarly construct a binary $[72, 31, 16]$ code having projection G onto the quaternary linear $[18, 9, 8]$ code S_{18} [20]. We take as the top code a binary $[18, 12, 4]$ code and as the parity code the repetition code of length 18. Then by Proposition 7.2 we get a binary $[72, 31, 16]$ code having projection G onto S_{18}.

7.2. Decoding. We sketch a hard decision decoding algorithm for binary linear codes having projection G. The decoding idea is analogous to the syndrome decoding algorithm [12] and, generally, the decoding algorithm given in [16].

Let $\mathcal{C}_2$ have projection G onto $\mathcal{C}_4$ with the parity code $\mathcal{P}$ and the top code $\mathcal{T}$. In order to make the situation simple we assume that $\mathcal{P}$ is the repetition code of proper length. Suppose $\mathbf{v}$ is a received vector. First we compute the parities of the columns of $\mathbf{v}$ and take the majority parity among them. We regard the columns of $\mathbf{v}$ with this parity as correct columns. Then we project $\mathbf{v}$ onto a vector $\mathbf{w}$ over GF(4). We find a closest codeword $\mathbf{x}$ in $\mathcal{C}_4$ to $\mathbf{w}$ by solving a syndrome equation with respect to H_4, the parity check matrix of $\mathcal{C}_4$. See [16] for more details. We then lift $\mathbf{x}$ to a

Table 7.1
Generator matrix of the binary $[48, 21, 12]$ *code.*

$$\begin{bmatrix}
1111\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 1111\ 1111\ 0000\ 0000 \\
0000\ 1111\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 1111\ 0000\ 1111\ 0000 \\
0000\ 0000\ 1111\ 0000\ 0000\ 0000\ 0000\ 0000\ 1111\ 0000\ 0000\ 1111 \\
0000\ 0000\ 0000\ 1111\ 0000\ 0000\ 0000\ 0000\ 0000\ 1111\ 1111\ 0000 \\
0000\ 0000\ 0000\ 0000\ 1111\ 0000\ 0000\ 0000\ 0000\ 1111\ 0000\ 1111 \\
0000\ 0000\ 0000\ 0000\ 0000\ 1111\ 0000\ 0000\ 0000\ 0000\ 1111\ 1111 \\
0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 1111\ 0000\ 1111\ 1111\ 1111\ 0000 \\
0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 1111\ 1111\ 0000\ 1111\ 1111 \\
1000\ 0111\ 0111\ 0111\ 0111\ 0111\ 0111\ 0111\ 1000\ 1000\ 0111\ 0111 \\
0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0011\ 0011\ 0011\ 0011\ 0011\ 0011 \\
0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0101\ 0101\ 0101\ 0101\ 0101\ 0101 \\
0011\ 0011\ 0011\ 0011\ 0011\ 0011\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000 \\
0101\ 0101\ 0101\ 0101\ 0101\ 0101\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000 \\
0000\ 0000\ 0000\ 0011\ 0101\ 0110\ 0000\ 0000\ 0000\ 0011\ 0101\ 0110 \\
0000\ 0000\ 0000\ 0101\ 0110\ 0011\ 0000\ 0000\ 0000\ 0101\ 0110\ 0011 \\
0011\ 0110\ 0101\ 0000\ 0000\ 0000\ 0011\ 0110\ 0101\ 0000\ 0000\ 0000 \\
0101\ 0011\ 0110\ 0000\ 0000\ 0000\ 0101\ 0011\ 0110\ 0000\ 0000\ 0000 \\
0000\ 0000\ 0000\ 0011\ 0110\ 0101\ 0101\ 0110\ 0011\ 0000\ 0000\ 0000 \\
0000\ 0000\ 0000\ 0101\ 0011\ 0110\ 0011\ 0101\ 0110\ 0000\ 0000\ 0000 \\
0011\ 0101\ 0110\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0110\ 0101\ 0011 \\
0110\ 0011\ 0101\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0011\ 0110\ 0101
\end{bmatrix}.$$

Table 7.2
Weight distribution of the binary $[48, 21, 12]$ *code.*

Weights	No.	Weights	No.	Weights	No.	Weights	No.
0	1	18	56832	26	203264	34	3072
12	2065	20	374012	28	373142	36	1884
14	2944	22	201984	30	56192	40	4
16	49254	24	722548	32	49953	44	1

binary vector $\mathbf{v}'$. There are often several choices for $\mathbf{v}'$. When the syndrome of $\mathbf{v}'$ with respect to H, the parity check matrix of $\mathcal{T}$, is zero, we take $\mathbf{v}'$ as a codeword of $\mathcal{C}_2$. Otherwise we go back to the previous step, finding a closest codeword in $\mathcal{C}_4$ to $\mathbf{w}$ by solving another syndrome equation. We repeat this step until we get a binary vector $\mathbf{v}'$ whose syndrome with respect to H is zero.

We can apply this algorithm to the second-order Reed–Muller code $R(2, m)$, as it has projection G with the repetition code as the parity code. We remark that a soft decision decoding for the first-order Reed–Muller code $R(1, m)$ was explained in [1]. It appears that a soft decision decoding for binary linear codes having projection G is possible in a similar fashion; see [1, 7, 10, 21, 22, 23, 24]. It would be interesting to find a fast hard or fast soft decision decoding algorithm for projection G.

Acknowledgments. We would like to thank T. A. Gulliver for providing his preprint [10] and Y. Be'ery for providing [24]. The first author would also like to thank O. Amrani for his helpful discussion.

REFERENCES

[1] O. Amrani and Y. Be'ery, *Reed-Muller codes: Projections on* $GF(4)$ *and multilevel construction*, IEEE Trans. Inform. Theory, 47 (2001), pp. 2560–2565.

[2] C. Bachoc and P. Gaborit, *On extremal additive* $GF(4)$*-codes of lengths* 10 *to* 18, J. Théorie Nombres Bordeaux, 12 (2000), pp. 225–271.

[3] A. E. Brouwer, *Bounds on the size of linear codes*, in Handbook of Coding Theory, V. S. Pless and W. C. Huffman, eds., Elsevier, Amsterdam, 1998, pp. 295–461.

[4] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, *Quantum error correction via codes over* $GF(4)$, IEEE Trans. Inform. Theory, 44 (1998), pp. 1369–1387.

[5] J. Cannon and C. Playoust, *An Introduction to Magma*, University of Sydney, Sydney, Australia, 1994.

[6] J. H. Conway and V. Pless, *On the enumeration of self-dual codes*, J. Combin. Theory Ser. A, 28 (1980), pp. 26–53.

[7] J. H. Conway and N. J. A. Sloane, *Decoding techniques for codes and lattices, including the Golay code and the Leech lattice*, IEEE Trans. Inform. Theory, 32 (1986), pp. 41–50.

[8] J. H. Conway and N. J. A. Sloane, *A new upper bound on the minimal distance of self-dual codes*, IEEE Trans. Inform. Theory, 36 (1990), pp. 1319–1333.

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

[10] M. Esmaeili, T. A. Gulliver, and A. K. Khandani, *On the Pless construction and ML decoding of the* $(48, 24, 12)$ *quadratic residue code*, IEEE Trans. Inform. Theory, 49 (2003), pp. 1527–1535.

[11] P. Gaborit, W. C. Huffman, J.-L. Kim, and V. Pless, *On additive* $GF(4)$ *codes*, in Proceedings of the DIMACS Workshop on Codes and Association Schemes, DIMACS Ser. Discrete Math. Theoret. Comput. Sci. 56, AMS, Providence, RI, 2001, pp. 135–149.

[12] P. Gaborit, J.-L. Kim, and V. Pless, *Decoding binary* $R(2, 5)$ *by hand*, Discrete Math., 264 (2003), pp. 55–73.

[13] T. A. Gulliver and J.-L. Kim, *Circulant based extremal additive self-dual codes over* $GF(4)$, IEEE Trans. Inform. Theory, submitted.

[14] A. R. Hammons, Jr., P. V. Kumar, A. R. Calderbank, N. J. A. Sloane, and P. Solé, *The* Z_4*-linearity of Kerdock, Preparata, Goethals, and related codes*, IEEE Trans. Inform. Theory, 40 (1994), pp. 301–319.

[15] G. Höhn, *Self-Dual Codes over the Kleinian Four Group*, preprint, 1996. Updated version available online at http://xxx.lanl.gov/ps/math.CO/0005266.

[16] J.-L. Kim and V. Pless, *Decoding some doubly-even self-dual* $[32, 16, 8]$ *codes by hand*, in Proceedings of a Conference Honoring Professor Dijen Ray-Chaudhuri on the Occasion of His 65th Birthday, Ohio State Univ. Math. Res. Inst. Publ. 10, Ohio State University, Columbus, OH, 2002, pp. 165–178.

[17] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, New York, 1977.

[18] V. Pless, *Decoding the Golay codes*, IEEE Trans. Inform. Theory, 32 (1986), pp. 561–567.

[19] V. Pless, *A classification of self-orthogonal codes over* $GF(2)$, Discrete Math., 3 (1972), pp. 209–246.

[20] E. Rains and N. J. A. Sloane, *Self-dual codes*, in Handbook of Coding Theory, V. S. Pless and W. C. Huffman, eds., Elsevier, Amsterdam, 1998, pp. 177–294.

[21] M. Ran and J. Snyders, *Constrained designs for maximum likelihood soft decoding of* $RM(2, m)$ *and the extended Golay codes*, IEEE Trans. Comm., 43 (1989), pp. 812–820.

[22] J. Snyders and Y. Be'ery, *Maximum likelihood soft decoding of binary block codes and decoders for the Golay codes*, IEEE Trans. Inform. Theory, 35 (1989), pp. 963–975.

[23] A. Vardy and Y. Be'ery, *More efficient soft decoding of the Golay codes*, IEEE Trans. Inform. Theory, 37 (1991), pp. 667–672.

[24] J. Yuan, C. S. Chen, and S. Ma, *Two-level decoding of* $(32, 16, 8)$ *quadratic residue code*, Proc. IEEE, 140 (1993), pp. 409–414.

SIAM J. DISCRETE MATH.
Vol. 16, No. 4, pp. 604–615

ON PLAYING GOLF WITH TWO BALLS[*]

IOANA DUMITRIU[†], PRASAD TETALI[‡], AND PETER WINKLER[§]

Abstract. We analyze and solve a game in which a player chooses which of several Markov chains to advance, with the object of minimizing the expected time (or cost) for *one* of the chains to reach a target state. The solution entails computing (in polynomial time) a function γ—a variety of "Gittins index"—on the states of the individual chains, the minimization of which produces an optimal strategy.

It turns out that γ is a useful cousin of the expected hitting time of a Markov chain but is defined, for example, even for random walks on infinite graphs. We derive the basic properties of γ and consider its values in some natural situations.

Key words. Gittins index, Markov chain, Markov decision theory, random walk, hitting time, game theory

AMS subject classifications. 60J10, 66C99

DOI. 10.1137/S0895480102408341

1. Introduction.

Everyone has encountered situations where there is more than one way to accomplish some task and where it may be desirable to change strategies from time to time depending on the outcome of various actions. In trying to contact a colleague, for example, one might first try telephoning, and depending on the result, telephone again later or perhaps try sending electronic mail. A dating strategy for someone who is seeking a mate might call for trying a new prospect, or retrying an old one, if things are going badly with the current one. In these situations, if one knows the best *first* move from any state, one can behave optimally.

Suppose you are invited to play the following game. Tokens begin on vertices 2 and 5 of a path connecting vertices $0, \ldots, 5$ (see Figure 1). A valuable gift awaits you if either token reaches vertex 3. At any time you may pay \$1 and point to a token; that token will then make a random move (with equal probability to its left or right neighboring vertex if it has two neighbors, otherwise to its only neighbor). Which token should you move first?

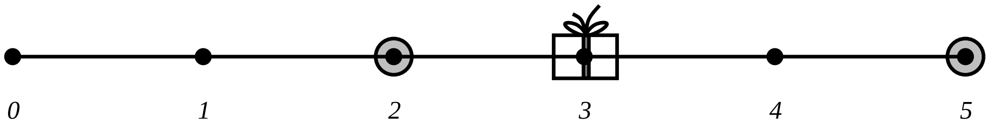


FIG. 1. *What's the fastest way to the gift?*

It is not difficult to see that by moving the token at 2 first, then switching permanently to the other if the game does not end immediately, your expected cost to reach the prize is \$3; this is the unique optimal strategy. Contrast this with a similar

[*] Received by the editors May 28, 2002; accepted for publication (in revised form) February 26, 2003; published electronically July 30, 2003.
http://www.siam.org/journals/sidma/16-4/40834.html

[†] Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA 02139 (dumitriu@math.mit.edu).

[‡] School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332-0160 (tetali@math.gatech.edu).

[§] Bell Labs, Lucent Technologies, Inc., 2C-365, Murray Hill, NJ 07974-0636 (pw@lucent.com).

"loyal" game in which you must choose one token and stick with it. Then, choosing the token at vertex 5 costs \$4 on average, and choosing the other token costs \$5 on average.

Now fix any graph G with distinguished target node t, and write $u \le v$ if, in the first-described game with tokens at u and v, there is an optimal strategy which calls for moving the token at u first. Is this relation transitive—that is, if $u \le v$ and $v \le w$, does that imply $u \le w$? (Note that two tokens may occupy the same vertex; in fact there is no loss or gain of generality if each token has its own graph and its own target).

In the loyal game, the corresponding statement is trivially true because there is a quantity (the expected length of a random walk from u to t, or the *hitting time*) which measures the desirability of choosing the token at u, regardless of where the other token may be.

When one is permitted to switch tokens the situation becomes more subtle. Nonetheless, it does turn out that there is a measure of first-move desirability which can be applied to a single token, and therefore transitivity does hold. This measure (our function γ) is polynomial-time computable, and it is related both to what Markov decision theorists know as the *Gittins index* of a single-armed bandit and to expected hitting times in a *different* Markov chain. The development here, however, will be mostly self-contained.

The main theorem will be stated in a more general, but by no means the *most* general, form. The graph is replaced by two (or more) "Markov systems," one for each token; each system consists of a finite-state Markov chain with a starting state and a target state, and a positive real move-cost at each state. Further generalizations are considered along the way.

2. Markov systems. We call the pieces from which we construct our games *Markov systems.* A Markov system $\mathcal{S} = \langle V, P, C, s, t\rangle$ consists of a state space V (which will be assumed finite unless otherwise specified), a transition matrix $P = \{p_{u,v}\}$ indexed by V, a positive real move-cost C_v for each state v, a starting state s, and a target state t. We will assume usually that t is accessible (ultimately) from every state in V.

The cost of a "trip" $v(0), \dots, v(k)$ on $\mathcal{S}$ is the sum $\sum_{i=0}^{k-1} C_{v(i)}$ of the costs of the exited states. The (finite) expected cost of a trip from v to the target t is denoted $\mathrm{E}_v[\mathcal{S}]$, with the subscript sometimes omitted when the trip begins at s. Since we never exit the target state, we may arbitrarily set $C_t = 0$ and $p_{t,t} = 1$.

3. Games and strategies. The games we consider are all played by a single player against a "bank" and consist of a series of moves chosen and paid for by the player, with random effect. We imagine that the player is forced to play until termination, which occurs mercifully in finite expected time.

The *cost* $\mathbf{E}[\mathcal{G}]$ of a game $\mathcal{G}$ is the minimum expected cost (to the player) of playing $\mathcal{G}$, taken over all possible strategies. A strategy which achieves expected cost $\mathbf{E}[\mathcal{G}]$ is said to be *optimal.*

Let $\mathcal{S}_1, \dots, \mathcal{S}_k$ be k Markov systems, each of which has a token on its starting state and an associated cost function C_i. A *simple multitoken Markov game* $\mathcal{S}_1 \circ \mathcal{S}_2 \circ \cdots \circ \mathcal{S}_k$ consists of a succession of steps in which we choose one of the k tokens, which takes a random step in its system (i.e., according to its P_i). After choosing a token i (on state u, say), we pay the cost $C_i(u)$ associated with the state u of the system $\mathcal{S}_i$ whose token we have chosen. As soon as one of the tokens reaches its target state for the first time, we stop.

We define the *terminator* $\mathcal{T}_g$ as the Markov system $\langle \{s,t\}, P, g, s, t\rangle$, where $p_{s,t}=1$. The terminator always hits its target in exactly one step, at cost g. The *token vs. terminator* game, in which we play a simple two-token game of systems $\mathcal{S}$ (for some $\mathcal{S}$) and $\mathcal{T}_g$ (for some g), will play a critical role in the analysis of general Markov games.

It will also be useful to define the *join* $\mathcal{G} = \mathcal{G}_1 \Box \mathcal{G}_2 \Box \cdots \Box \mathcal{G}_n$ of games $\mathcal{G}_1, \ldots, \mathcal{G}_n$ as follows: at each step the player chooses one of the n games, then pays for and makes a move in that game. $\mathcal{G}$ terminates when any of its component games is finished. We will employ the join in order to analyze the Markov game $\mathcal{S}_1 \circ \mathcal{S}_2 \circ \cdots \circ \mathcal{S}_k$.

Throughout the paper, we will be using (sometimes without making explicit reference to) the following two classical theorems from the general theory of Markov game strategies; the reader is referred to [6] for more detail.

The first theorem enables us to look for optimal strategies in a finite set.

Theorem 3.1. *Every Markov game (in our sense) has a pure optimal strategy.*

From a given *state* u of a Markov game, an *action* α produces an immediate expected cost $C_u(\alpha)$ and a probability distribution $\{p_{u,\cdot}\}$ of new states. (Note that in the present context, a *state* of a Markov game consisting of k Markov systems would be a specific configuration of the k tokens, and an *action* would correspond to the choice of a particular token to move.) Thus a strategy σ which takes action α at the state u satisfies

$$E_u[\sigma] = C_u(\alpha) + \sum_v p_{u,v}(\alpha) E_v[\sigma] \ .$$

If among all possible actions at state u, α minimizes the right-hand side of this expression, σ is said to be *consistent* at u.

Theorem 3.2. *A strategy σ is optimal if and only if it is consistent at every state.*

Proof. Let τ be an optimal strategy and U the set of states v on which $E_v[\tau]-E_v[\sigma]$ attains its minimal value x. Since $t \notin U$, we can find a state $u \in U$ from which some state not in U can be reached in one step by τ. But then, if α is the action taken by τ at u,

$$E_u[\sigma] = E_u[\tau] + x = C_u(\alpha) + \sum_v p_{u,v}(\alpha)(E_v[\tau]+x) < C_u(\alpha) + \sum_v p_{u,v}(\alpha) E_v[\sigma],$$

contradicting the fact that σ is consistent at u. □

4. The grade. We say that an optimal player is *indifferent* among some set of moves if for each of those moves there is an optimal strategy which employs it. Going back to the *token vs. terminator* game from the preceding section, we define the *grade* $\gamma(\mathcal{S})$ of a system $\mathcal{S} = \langle V, P, C, s, t\rangle$ to be the unique value of g at which an optimal player is indifferent between the two possible first moves in the game $\mathcal{G}_g = \mathcal{S} \circ \mathcal{T}_g$. Thus, $\gamma(\mathcal{S})$ is the least value of g such that if, at any time, we can pay g to quit the system $\mathcal{S}$, we are still willing to try one move in $\mathcal{S}$. (To be consistent with our notation below, we should be denoting $\gamma(\mathcal{S})$ by $\gamma_s(\mathcal{S})$, indicating the start state of the game.)

To see that $\gamma = \gamma(\mathcal{S})$ is a well-defined quantity, we will make use of Theorem 3.1. Any pure strategy σ is defined by the set $Q(\subset V)$ of states in which it chooses to move in $\mathcal{T}_g$. Suppose $\mathcal{S}$ is run until either t or a state in Q is reached; let the first event be represented by R, and let X be the final cost of the run in $\mathcal{S}$. Put $p = \Pr[R]$, $A = \mathrm{E}[X|R]$, and $B = \mathrm{E}[X|\neg R]$. Then

$$\mathrm{E}[\sigma] = pA + (1-p)(B+g),$$

i.e., $E[\sigma]$ is linear in g for fixed strategy σ. Optimizing over σ, it follows that $\mathbf{E}[\mathcal{G}_g]$ is the maximum of a set of linear functions and is therefore continuous in g. For g less than the cost of the first move, $\mathbf{E}[\mathcal{G}_g] = g$ (because we choose to move in $\mathcal{T}_g$). On the other hand, if g exceeds the expected cost $\mathrm{E}_v[\mathcal{S}]$ from any state v, then we will always choose to move in $\mathcal{S}$, hence $\mathbf{E}[\mathcal{G}_g] = \mathrm{E}_s[\mathcal{S}]$. Figure 2 illustrates a typical shape for the graph of $\mathbf{E}[\mathcal{G}_g]$.

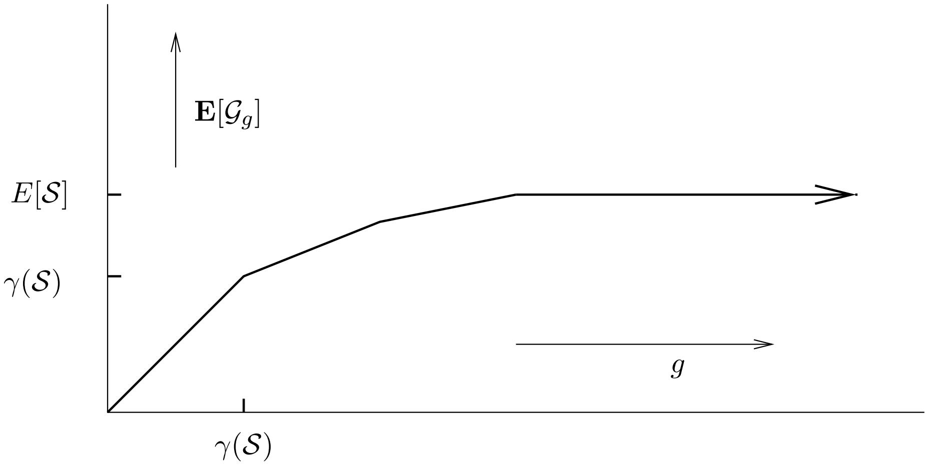


FIG. 2. *The expected cost of $\mathcal{G}_g = \mathcal{S} \circ \mathcal{T}_g$.*

The grade of $\mathcal{S}$ is marked on the figure as the highest value of g at which the strategy "play in $\mathcal{T}_g$" is optimal, i.e., the coordinate of the top end of the line segment of slope 1.

We use $\gamma_u(\mathcal{S})$ or just γ_u (when $\mathcal{S}$ is fixed except for its starting state) to denote the grade of $\mathcal{S}_u = \langle V, P, C, u, t\rangle$. Hence we can formulate the following theorem.

THEOREM 4.1. *A strategy for $\mathcal{S}\circ\mathcal{T}_g$ is optimal if and only if it chooses $\mathcal{S}$ whenever the current state u of $\mathcal{S}$ satisfies $\gamma_u < g$ and it chooses $\mathcal{T}_g$ whenever $\gamma_u > g$.*

Remark 4.2. Note that in system $\mathcal{S} = \langle V, P, C, s, t\rangle$ there is positive probability of moving from s to a state of strictly lower grade. Otherwise, in the *token vs. terminator* game $\mathcal{S} \circ \mathcal{T}(\gamma(\mathcal{S}))$ the strategy of paying for the first move in $\mathcal{S}$ and then terminating would be optimal, yet more costly than terminating immediately.

5. An optimal strategy for the simple multitoken game. The surprising and fundamental discovery of Gittins, first proved by Gittins and Jones [2], was that in many Markov games, options could be "indexed" separately and then numerically compared to determine an optimal strategy. This is indeed the case for our games, the index being the "grade" defined above.

THEOREM 5.1. *A strategy for the game $\mathcal{G} = \langle \mathcal{S}_1 \circ \cdots \circ \mathcal{S}_n\rangle$ is optimal if and only if it always plays in a system whose current grade is minimal.*

Proof. We will employ a modified version of the very elegant proof given by Weber [5] for Gittins' theorem. Our "grade" differs from the Gittins index in several minor respects, among them that our games terminate and our costs are not subject to discounting (about which more later). These differences are not sufficient to regard the grade as other than a special case, or variation, of the Gittins index.

The proof will proceed using a sequence of easy lemmas. We begin by considering a "reward game" $\mathcal{S}_i(g)$ based on the system $\mathcal{S}_i$, in which we play and pay as in $\mathcal{S}$

but may quit at any time; as incentive to play, however, there is a reward of g at the target which we may claim when and if the target is reached.

LEMMA 5.2. *$\mathcal{S}_i(\gamma(\mathcal{S}_i))$ is a fair game (that is, the expectation $\mathbf{E}[\mathcal{S}_i(\gamma(\mathcal{S}_i))] = 0$) and a strategy for $\mathcal{S}_i(\gamma(\mathcal{S}_i))$ is optimal if and only if the player quits whenever his current state u satisfies $\gamma_u > g$ and plays on when $\gamma_u < g$.*

Proof. The reward game is no different from a terminator game $\mathcal{S} \circ \mathcal{T}_{\gamma(\mathcal{S}_i)}$ in which the player is provided with an initial stake of $\gamma(\mathcal{S}_i)$, hence the characterization of optimality follows from Theorem 4.1. Since quitting immediately is among the optimal strategies, $\mathbf{E}[\mathcal{S}_i(\gamma(\mathcal{S}_i))] = 0$. □

Suppose the game $\mathcal{S}_i(\gamma_u(\mathcal{S}_i))$ is amended in the following teasing manner: whenever the player reaches a state u with $\gamma_u > g$, the reward at the target is boosted up to γ_u—just enough to tempt the player to continue. (Note that the reward is never lowered.) It might seem that this game, which we will denote simply by $\mathcal{S}'_i$, is better than fair, but we have the following lemma.

LEMMA 5.3. *$\mathcal{S}'_i$ is fair, and a strategy for $\mathcal{S}'_i$ is optimal if and only if the player never quits when the current grade is below the current reward value.*

Proof. To see that $\mathbf{E}[\mathcal{S}'_i] = 0$, note that a session with $\mathcal{S}'_i$ can be broken up into a series of smaller games, each ending either upon reaching a state U whose grade exceeds the current reward value, or upon reaching the final target. Since each of these games is fair, so is $\mathcal{S}'_i$. Note that the antipodal strategies of quitting immediately, and of playing until the target is hit, are in particular both optimal. □

Now we consider the join $\mathcal{G}' := \mathcal{S}'_1 \square \cdots \square \mathcal{S}'_n$, in which we play the teaser game of our choice, paying as we go, until we quit or hit one of the targets (in which case we claim the current reward at that target).

LEMMA 5.4. *$\mathcal{G}'$ is a fair game.*

Proof. Any combination (simultaneous, sequential, or interleaved) of the independent fair games $\mathcal{S}'_1, \ldots, \mathcal{S}'_n$ is still fair. The join $\mathcal{G}'$ can be no better than such a combination, since it differs only in having additional restrictions on the player; hence it is *at best* fair. However, $\mathcal{G}'$ cannot be worse than fair, since, e.g., the player can simply quit at the start or play one game to its finish and ignore the others. □

Among the strategies for $\mathcal{G}'$ is one we call the "Gittins strategy" Γ: always play from a system which is currently of minimal grade. This is the strategy we claim is optimal for the original game $\mathcal{G}$, but first we observe two properties of Γ relative to the game $\mathcal{G}'$.

LEMMA 5.5. *The Gittins strategy Γ is optimal for $\mathcal{G}'$.*

Proof. If a move by Γ results in the grade of a component game $\mathcal{S}'_i$ dropping below its reward value, then since its grade has just gone down it is now the *unique* lowest-grade component and therefore Γ will again move that token. Hence no component system will ever be stranded in a state u with γ_u less than the reward on target t_i, thus all the components $\mathcal{S}'_i$ are played optimally. □

LEMMA 5.6. *Of all strategies for $\mathcal{G}'$ which play until a target is hit, Γ reaps the smallest expected reward at the end. In other words, if the move-costs are waived, then Γ actually is the* worst *(in terms of reward collected) possible nonquitting strategy for $\mathcal{G}'$. Furthermore, among nonquitting strategies which are optimal for the unaltered game $\mathcal{G}'$, Γ is the only one with this property.*

Proof. Imagine that the course of each individual system $\mathcal{S}_i$ is fixed. Then each teaser game $\mathcal{S}'_i$ terminates, if played all the way to its target, with a certain reward g_i (equal to the largest γ_u over all states u hit en route). Every nonquitting strategy will claim one of the rewards g_i at the end, but the Gittins strategy gets the *smallest* one; the reason is that if it collected a nonminimal reward (say g_j) when teaser game $\mathcal{S}'_i$, $i \neq j$, was headed for a final reward of $g_i < g_j$, then at the time of termination of $\mathcal{G}'$ the reward for $\mathcal{S}'_i$ was g_i or less, hence $\gamma_u(\mathcal{S}_i) \leq g_i$ where u is its last state. But this is impossible because the final run of plays of $\mathcal{S}_j$ began at a state v where $\gamma_v(\mathcal{S}_j) = g_j$ or more, and Γ should have preferred to play in $\mathcal{S}_i$ at that time. From the proof it is clear that Γ is unique in the sense of the last assertion in the statement of the lemma. □

We are finally ready to show that Γ is optimal for the original game $\mathcal{G}$. For any nonquitting strategy Δ for $\mathcal{G}'$, let $C(\Delta)$ be its expected cost and $R(\Delta)$ its expected reward; thus $\mathrm{E}[\Delta] = R(\Delta) - C(\Delta) \leq 0$ since $\mathcal{G}'$ is fair. But then since $\mathrm{E}[\Gamma] = 0$,

$$C(\Gamma) = R(\Gamma) \leq R(\Delta) \leq C(\Delta),$$

so Γ incurs the least cost among all nonquitting strategies for $\mathcal{G}'$, and this says exactly that it is optimal for $\mathcal{G}$.

If Δ is also optimal for $\mathcal{G}$, then the above inequalities are both tight, hence Lemmas 5.5 and 5.6 both hold for Δ. If Δ is not a Gittins strategy, then we may assume that Δ makes a non-Gittins move already at the start of the game, playing $\mathcal{S}_2$ even though $\mathcal{S}_1$ has smaller grade. This will not necessarily cause it to miss the smallest reward in $\mathcal{G}'$, because there may be 0 probability of that system hitting its target immediately and Δ can return to $\mathcal{S}_1$ before it's too late. However, it follows from Remark 4.2 above that there is *always* a positive probability that any system will reach its target along a path whose grade is strictly declining. If this is fated to happen to both $\mathcal{S}_1$ and $\mathcal{S}_2$, then Δ will either end up accepting the larger reward of $\mathcal{S}_2$ (thus failing to have minimal reward) or leave one of the systems in a "grade below reward" state (thus failing to be optimal for $\mathcal{G}'$).

We conclude that Δ is optimal for $\mathcal{G}$ if and only if it is a Gittins strategy, and the proof of Theorem 5.1 is complete. □

6. The grade and the Gittins index. Both the history and the range of applicability of the Gittins index are rather complex subjects; the reader is referred to Gittins' modestly written book [1] for some appreciation of the former. It appears that the mathematical and statistical communities took some time to appreciate that the notorious "multi-armed bandit" problem had been solved; then they took additional time to find new, cleaner proofs and to uncover some very nice disguised consequences. The experience of this paper's authors suggests that the Gittins index is still not widely known in the mathematical community, especially among researchers in combinatorics and in the theory of computing. We hope to make a start at rectifying the situation with this work.

Framed in our terms, the circumstances to which the Gittins index was originally applied comprise a collection of Markov systems $\mathcal{S}_1, \ldots, \mathcal{S}_n$ such as those we have considered but without target states and with rewards instead of costs. When a system is chosen (say at time t) a (possibly random) nonnegative and uniformly bounded reward R_t, dependent on the state of that system, is collected. The object is to maximize $\sum_{t=0}^{\infty} \beta^t R_t$, where β is a "discount" strictly between 0 and 1.

"Gittins' theorem" asserts the existence of an index depending on system and state whose maximization at each stage produces an optimal strategy. Readers are referred to [7] and [4] as well as [2], [5] and Gittins' book [1] for various proofs.

The discount β is ubiquitous in Markov decision theory and in economics, financial, and actuarial research as well. It is necessary in the multi-armed bandit formulation to make the objective function finite. Discounts are less natural and familiar to pure mathematicians and are obviated in our presentation, where the presence of terminating targets keeps things finite. The elimination of discounts, particularly in the context of job scheduling problems, is discussed in section 6.2 of [1]; one approach, which can be used to deduce Theorem 5.1 from Gittins' theorem, is to let targets represent cycling states of zero reward and allow the discount factor to approach 1.

One benefit of our formulation is its natural application to the problem of minimizing the time needed to reach a goal, for example, for some token on a graph to reach a target node via random walk. As a result we can represent our "grade" γ as a hitting time, or more generally a hitting cost.

Let $\mathcal{S} = \langle V, P, C, s, t\rangle$ be a Markov system and let $U \subset V \backslash \{s, t\}$. Define a new system $\mathcal{S}\restriction U = \langle U \cup \{s, t\}, P', C, s, t\rangle$ by putting

$$p'_{u,s} = p_{u,s} + \sum_{v \in V \backslash \{U \cup \{s,t\}\}} p_{u,v}$$

and $p'_{u,w} = p_{u,w}$ for $w \in U$ and $u \in U$. In effect, $\mathcal{S}\restriction U$ is the restriction of $\mathcal{S}$ to U, where the state-marking token is sent back to s whenever it tries to leave U.

THEOREM 6.1. *With $\mathcal{S}$ and U as above, $\gamma_s(\mathcal{S}) \le \mathrm{E}_s[\mathcal{S}\restriction U]$, with equality if U contains all states of grade lower than γ_s and no states of grade higher than γ_s.*

Proof. Let σ be the strategy for playing the "reward game" $\mathcal{S}(\gamma_s(\mathcal{S}))$ which entails playing until the target is hit or some state $v \in V \backslash U$ is reached, in which case the game is terminated. Since $\mathcal{S}(\gamma_s(\mathcal{S}))$ is a fair game, σ has nonpositive expectation. Suppose we are permitted to restart a new $\mathcal{S}(\gamma_s(\mathcal{S}))$ and continue with strategy σ, whenever there is a voluntary termination. The resulting sequence of games still has nonpositive expectation but is equivalent to playing the reward game $\mathcal{S}\restriction U(\gamma_s(\mathcal{S}))$ until the target is hit. Since this will always result in collecting $\gamma_s(\mathcal{S})$ at the end, the expected total move-cost must be at least $\gamma_s(\mathcal{S})$.

On the other hand, we know from Theorem 4.1 that σ is optimal (thus has zero expected reward) when U fulfills the additional conditions; in that case we get that the expected total move-cost is precisely $\gamma_s(\mathcal{S})$. □

Note that the $U = \emptyset$ case yields the rather obvious fact that $\gamma_x \le \mathrm{E}_x[\mathcal{S}]$ for all x.

It might be argued that Theorem 6.1 is circular since it reduces computing the grade to computing a hitting cost, but only if we know which states have grade less than γ_s, and which have grade more than γ_s. However, in the next section we use the theorem recursively to compute grades one by one.

7. Computing the grade.

Like (most variations of) the Gittins index, our "grade" can be determined in time bounded by a polynomial in the length of description of a system $\mathcal{S}$. We will now present and analyze an algorithm which calculates the grade γ_u of all the states u of $\mathcal{S}$, one state at a time.

Let U be the set of states in V whose grades have already been calculated. We add one more state to U, namely, the state of smallest grade in $V \backslash U$. Let $N(U)$ denote the set of states x in $V \backslash U$ that are reachable directly from a state in U (i.e., $N(U) := \{v \in V | p_{u,v} > 0$ for some $u \in U\}$).

As before, $\mathrm{E}_x[\mathcal{S}]$—the "hitting cost"—denotes the expected cost of a trip to t from x.

The algorithm is given in pseudocode below.

```
1. $U = \{t\}$, $\gamma_t = 0$;
2. While $V \backslash U \neq \phi$
    (a) $\mathit{CheckedStates} = \phi$;
```

```
(b) While $\mathit{CheckedStates} \neq N(U)$
    i. Choose $v \in N(U) \setminus \mathit{CheckedStates}$;
    ii. Let $P' = \{p'_{u,v}\}$ be the transition matrix obtained from $P = \{p_{u,v}\}$
        in the following way:
        • $P'$ disregards all states not in $U \cup \{v\}$;
        • $p'_{u,v} = p_{u,v} + \sum_{w \in V \setminus \{U \cup \{v\}\}} p_{u,w} \ \forall u \in U \cup \{v\}$;
        • $p'_{u,u'} = p_{u,u'} \ \forall u \in U \cup \{v\}$ and $u' \in U$.
    iii. Compute $h_v = \mathrm{E}_v[\mathcal{S}']$, where $\mathcal{S}' = \langle U, P', C, v, t \rangle$;
    iv. $\mathit{CheckedStates} = \mathit{CheckedStates} \cup \{v\}$;
(c) Find $x$ such that $h_x = \min\{h_v : v \in \mathit{CheckedStates}\}$;
(d) $U = U \cup \{x\}$, $\gamma(x) = h_x$.
```

It is evident from Theorem 6.1 that if the selected state x always has minimum grade among the states in $V \setminus U$, then the algorithm correctly computes the grades of all states in V.

We first note that a minimum grade $x \notin U$ is indeed to be found among the neighbors of U, because Remark 4.2 implies that there is a path of decreasing grade from x to t.

It remains only to establish that if $v \in V \setminus U$ is *not* of minimum grade, then $h_v = \mathrm{E}_v[\mathcal{S}']$ is at least as large as γ_v. But this is exactly the content of Theorem 6.1 of the preceding section.

Let us now analyze the running time for the algorithm.

Let n be the initial number of states. At step i, $N(U)$ has $\mathrm{O}(n{-}i)$ states. For any state in $N(U)$, the greatest workload is done to compute $E_v(P')$. It involves solving an $(i{+}1) \times (i{+}1)$ system of equations; this can be done by an *LU* factorization followed by a backward substitution, and it represents $\mathrm{O}(i^3)$ work. Therefore, we can compute all the grades in $\mathrm{O}(\sum_{i=1}^{n} (n{-}i) i^3) = \mathrm{O}(n^5)$ time.

8. States of maximum grade. If the starting state of system $\mathcal{S}$ has maximum grade, then "never quitting" is an optimal strategy for the *token vs. terminator* game $\mathcal{S} \circ \mathcal{T}_\gamma$. Hence we have the following lemma.

LEMMA 8.1. *Let z be a state of maximum grade in a system $\mathcal{S}$. Then $\gamma_z = \mathrm{E}_z[\mathcal{S}]$.*

The converse of Lemma 8.1 fails for the uninteresting reason that states of higher grade than z may exist but not be accessible from z. More interesting is the question of maximum grade versus maximum hitting cost (that is, maximum expected cost of hitting t).

THEOREM 8.2. *In any system $\mathcal{S}$ the states of maximum grade and the states of maximum hitting cost are the same.*

Proof. Suppose that x maximizes $\mathrm{E}_x[\mathcal{S}]$, that is, x incurs the greatest expected cost $h_x = \mathrm{E}_x[\mathcal{S}]$ of hitting t assuming best strategy. Then we claim that $\gamma_x = h_x$. To see this, we let U be the set of states u in V such that $\gamma_u > \gamma_x$ and compute h_x by considering the effect of the event "A" that a walk from x hits U before it reaches t. Then

$$\begin{aligned} h_x &= \Pr[\neg A]\mathrm{E}_x[\mathcal{S}|\neg A] + \Pr[A]\left(\mathrm{E}_x[U] + \mathrm{E}_U[\mathcal{S}]\right) \\ &\le \Pr[\neg A]\mathrm{E}_x[\mathcal{S}|\neg A] + \Pr[A]\left(\mathrm{E}_x[U] + h_x\right), \end{aligned}$$

where $\mathrm{E}_x[U]$ is the expected cost of hitting U from x and $\mathrm{E}_U[\mathcal{S}]$ is the expected cost of hitting t from the random point in U which is hit first. Solving, we get

$$h_x(1 - \Pr[A]) \le \Pr[\neg A]\mathrm{E}_x[\mathcal{S}|\neg A] + \Pr[A]\mathrm{E}_x[U] \ .$$

However, if we compute $\gamma_x = \mathrm{E}_x[\mathcal{S}\restriction U]$ in the same fashion, we get

$$\mathrm{E}_x[\mathcal{S}\restriction U](1-\Pr[A]) = \Pr[\neg A]\mathrm{E}_x[\mathcal{S}|\neg A] + \Pr[A]\mathrm{E}_x[U]$$

so that $h_x \le \gamma_x$; thus they are equal. In particular, $\gamma_y \le h_y \le h_x = \gamma_x$ for all y so x also has maximal grade.

Suppose, on the other hand, that z has maximal grade, but not maximal hitting cost; let x have maximal hitting cost. But then we have seen that $\gamma_x = h_x > h_z \ge \gamma_z$, a contradiction. The theorem follows. □

Remark 8.3. Theorems 6.1 and 8.2 provide an algorithm for computing grades from highest to lowest, as opposed to the one we presented earlier. The idea is to find the state x_1 of largest hitting cost (hence highest grade), then the state x_2 which maximizes $\mathrm{E}_{x_2}[\mathcal{S}\restriction(V\backslash\{x_1\})]$, etc. Although we are not able to take advantage here of the neighborhood structure, the running time for this algorithm is of the same order as before, relative to the number of states.

9. Grades and graphs.

The hitting time (from, say, x to y) for a simple random walk on a graph G has many beautiful properties, including ties to electrical networks; our analogue, the "grade," has the additional advantage of being finite even when G is infinite. Below we illustrate some calculations and theorems concerning the grades of vertices of some symmetric graphs.

We have assumed up until now that our Markov chains have finite state spaces, and indeed it would appear that there are problems with the expected outcome of our basic game when the expected number of steps to hit a target is infinite; or even worse, when there is positive probability that the target will never be hit. However, the simple multitoken game makes sense as long as at least one of the systems it deals with has a finite hitting time to the target, and of course the "terminator" system has this property. It is not difficult to prove that.

THEOREM 9.1. *Let $\mathcal{M}$ be an infinite, locally finite Markov chain, with designated target state t. Then*

1. *every state u of $\mathcal{M}$ has a grade $\gamma_u = \gamma_u(\mathcal{M}) < \infty$;*
2. *for all real k, the set $S_k = \{v \in \mathcal{M} : \gamma_v < k\}$ is finite;*
3. *for all $u \in \mathcal{M}$, there exists a finite chain $\mathcal{M}'$, obtained via suppressing all but a finite number of states in $\mathcal{M}$, for which $\gamma_u(\mathcal{M}) = \gamma_u(\mathcal{M}')$.*

We will sketch the proof of Theorem 9.1; it is left to the reader to fill in the details.

Proof.

1. Since the Markov chain is locally finite, it follows that from any state u there is a (finite) shortest path to t. Let u be an arbitrary state, let $k < \infty$ be the length of a shortest path from u to t, and let p be the probability of this path. (Since the chain is locally finite, it follows that $p > 0$.) Let $g_* = k/p$, and consider the *token vs. terminator* $\mathcal{T}_{g_*}$ game, with the following strategy: starting from u, move k times "blindly" on $\mathcal{M}$, paying before each move; if after k steps the token is not on the target, pay the terminator and end the game in a step.
   It is immediate to verify that this strategy, though perhaps suboptimal, breaks even: the expected profit/loss from it is 0. But if γ_u were infinite, then for any finite g (in particular for g_*), any strategy for the *token vs. terminator* T_g game that does not choose the terminator T_g immediately would guarantee a positive loss! Hence γ_u must be finite. Moreover, it also follows that $g_* \ge \gamma_u$.

2. A state v whose distance from t is at least k will also (necessarily) have a grade of at least k; this is equivalent to saying that for any real k, $S_k \subseteq D_k = \{v \in \mathcal{M} : dist(v,t) < k\}$. Due to the local finiteness of the chain, for any real k, the set D_k is finite; hence for any real k, S_k is finite.
3. This follows directly from 1 and 2: given a state u, let $k = \gamma_u$, and suppress all states of $\mathcal{M}$ but for those in S_k. In the newly obtained finite chain $\mathcal{M}'$, $\gamma_u(\mathcal{M}) = \gamma_u(\mathcal{M}')$. □

In the following subsections, we consider the grade function for the simple random walk on each of the following graphs: the hypercube, the Cayley tree, the plane square grid, and the cubic grid in three-space. The last three are immediately relevant to the above, as infinite, locally finite chains; the first is finite, but interesting in itself.

9.1. The hypercube. We begin with a finite graph, the n-dimensional hypercube Q^n, whose vertices are binary sequences $u = (u_1, \dots, u_n)$ with $u \sim v$ when they differ in just one coordinate. The "kth level" of Q^n consists of the vertices with exactly k 1's. If the target vertex is fixed at the origin, then the grade γ_k of a point in level k is the hitting time from level k to level 0 in the truncated hypercube Q^n_k, defined as follows: all vertices at level greater than k are deleted, and each vertex at level k is provided with $n-k$ loops so that its total degree is n.

Let T_j be the time it takes to get from level j to level $j-1$ in Q^n_k. Clearly $\gamma_k = \sum_{j=1}^k T_j$; we derive the following recursion for T_j:

$$T_k = \frac{n}{k}, \quad \text{and}$$
$$T_j = 1 + \frac{n-j}{n}\Big(T_j + T_{j+1}\Big).$$

It is straightforward to verify that

$$T_j = \frac{1}{\binom{n-1}{n-j}} \sum_{i=n-j}^{n} \binom{n}{i};$$

this yields

$$\gamma_k = \sum_{i=1}^{k} \frac{1}{\binom{n-1}{n-j}} \sum_{i=n-j}^{n} \binom{n}{i}.$$

9.2. The Cayley tree. The d-regular Cayley tree is the unique connected, cycle-free infinite graph T^d whose vertices each has degree d. Again, this is a symmetric graph so we may assume the target vertex is an arbitrary "root" t.

The case $d = 2$ is the doubly infinite path, in which the grade of a vertex v at distance k from t is easily seen to be $k(k+1)$.

In general, the grade γ_k of a vertex v at distance k from the root is the hitting time from v to t in the graph T^d_k consisting of the first k levels of the tree (the root being at level 0), in which every vertex on the last level has $d-1$ loops (instead of $d-1$ children). This leads to a recurrence to which the solution, for $d > 2$, is:

$$\gamma_k = \frac{d\Big((d-1)^{k+1} - 1 - (k+1)(d-2)\Big)}{(d-2)^2}. \tag{9.1}$$

Interestingly, there is another way to compute the grade on T^d which works on any finite tree and shows that on trees, grades and hitting times are always integers. Let T be any tree, possibly with loops. Fix a target vertex t, and let v be any other vertex. Order the edges (including loops) incident to each $u \neq t$ arbitrarily subject to the edge on the path from u to t being last. Now walk from v by choosing each exiting edge in round-robin fashion, in accordance with the edge-order at the current vertex, until t is reached. For example, if the edges incident to some degree-3 vertex u are ordered e_1, e_2, e_3, then the first time u is reached it is exited via e_1, the second time by e_2, the fourth time by e_1 again, etc. We call such a walk a "whirling tour"; an example is provided in Figure 3.

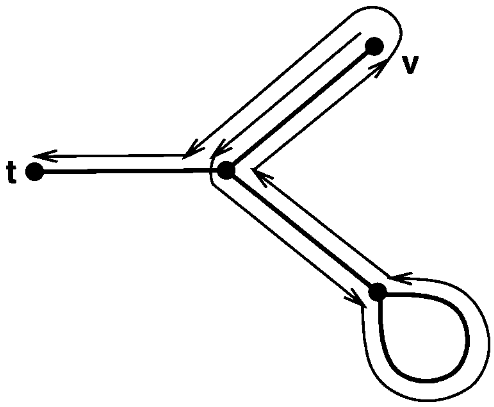


FIG. 3. *A whirling tour.*

THEOREM 9.2. *In any finite tree (possibly with some loops) the length of any whirling tour from v to t is exactly the expected hitting time from v to t.*

We leave the proof to the amusement of the reader.

We will denote by g_k the length of such a walk from the kth level to the root for every $k \in \mathbb{N}, k \geq 1$.

In order to walk from level k to level 0 (the root), we have to first execute a walk from level k to level 1 and then walk from there to the root. The rest of the walk will be a depth-first search of a $(d-1)$-ary tree with loops for leaves, plus the final edge. The length of the depth-first search is easily computed: we have

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

edges and $(d-1)^{k-1}$ loops; each edge is walked twice (once forward, once backward) and each loop is walked once for a total length of

$$2\Big(\frac{(d-1)^k - 1}{d-2} - 1\Big) + (d-1)^k = \frac{d(d-1)^k - 2d}{d-2}.$$

This sets up the recurrence

$$g_k = \frac{d(d-1)^k - 2d + 2}{d-2} + g_{k-1} + 1,$$

where $g_0 = 0,\ g_1 = d$. Thus,

$$g_k = \sum_{j=1}^{n} \frac{d\Big((d-1)^j - 1\Big)}{d-2} = \frac{d\Big((d-1)^{k+1} - 1 - (k+1)(d-2)\Big)}{(d-2)^2}$$

in accordance with (9.1), as expected.

9.3. Grids. The d-dimensional grid $\mathbb{Z}^d$ is the graph whose vertices are d-tuples of integers, with $u \sim v$ if u and v are at Euclidean distance 1. Since simple random walks on $\mathbb{Z}^d$ behave approximately symmetrically with respect to rotation, one would expect that the Gittins index of a node of $\mathbb{Z}^d$, with the origin as target, depends largely on its distance from the origin. This and more has recently been verified by Janson and Peres [3]; we quote their results below. To prove these, Janson and Peres employ a general lemma bounding the grade of each state of a Markov chain on both sides. The bounds are provided by integrals which depend on some harmonic function defined on the states.

THEOREM 9.3. *For simple random walk on* $\mathbb{Z}^2$,

$$\gamma(x,0) = 2|x|^2 \ln|x| + (2\gamma + 3\ln 2 - 1)|x|^2 + O(|x|\ln|x|), \qquad |x| \geq 2,$$

where γ *on the right-hand side is Euler's constant,* $\lim_{n\to\infty}(-\log_e n + \sum_{i=1}^{n} 1/i)$.

THEOREM 9.4. *For simple random walk on* $\mathbb{Z}^d$, $d \geq 3$,

$$\gamma(x,0) = \frac{\omega_d}{p_d}|x|^d + O(|x|^{d-1}),$$

where $\omega_d = \pi^{d/2}/\Gamma(d/2+1)$ *is the volume of the unit ball in* $\mathbb{R}^d$ *and* p_d *is the escape probability of the simple random walk, i.e., the probability that the random walk never returns to its starting point.*

From these theorems it follows that for each dimension d there is a constant $C = C(d)$, independent of the starting position x, such that the optimal strategy is to restart from every position y with $|y| > |x| + C$ but never when $|y| < |x| + C$.

REFERENCES

[1] J.C. GITTINS, *Multi-Armed Bandit Allocation Indices*, John Wiley, New York, 1989.

[2] J.C. GITTINS AND D.M. JONES, *A dynamic allocation index for the design of experiments*, in Progress in Statistics, Colloq. Math. Soc. János Bolyai 9, J. Gani, K. Sarkadi, and I. Vince, eds., North–Holland, Amsterdam, 1974, pp. 241–266.

[3] S. JANSON AND Y. PERES, *Hitting Times for Random Walks with Restarts*, preprint, Department of Statistics, U.C. Berkeley, Berkeley, CA, 2001.

[4] J.N. TSITSIKLIS, *A short proof of the Gittins index theorem*, Ann. Appl. Probab., 4 (1994), pp. 194–199.

[5] R. WEBER, *On the Gittins index for multiarmed bandits*, Ann. Appl. Probab., 2 (1992), pp. 1024–1033.

[6] D.J. WHITE, *Markov Decision Processes*, John Wiley, New York, 1993.

[7] P. WHITTLE, *Multi-armed bandits and the Gittins index*, J. Roy. Statist. Soc. Ser. B, 42 (1980), pp. 143–149.

SIAM J. DISCRETE MATH.
Vol. 16, No. 4, pp. 616–629

NONSEPARATING CYCLES IN 4-CONNECTED GRAPHS[*]

SEAN CURRAN[†] AND XINGXING YU[†]

Abstract. We prove that given any fixed edge ra in a 4-connected graph G, there exists a cycle C through ra such that $G - (V(C) - \{r\})$ is 2-connected. This will provide the first step in a decomposition for 4-connected graphs. We also prove that, for any given edge e in a 5-connected graph G, there exists an induced cycle C through e in G such that $G - V(C)$ is 2-connected. This provides evidence for a conjecture of Lovász.

Key words. nonseparating cycle, k-connected graph, disjoint paths, Hamilton cycle

AMS subject classifications. 05C38, 05C40

DOI. 10.1137/S089548010139518X

1. Introduction and notation.

Throughout the paper, we consider only simple graphs. We let $G = (V(G), E(G))$ be the graph with *vertex set* $V(G)$ and *edge set* $E(G)$. We use the shorthand notation xy (or yx) for an edge in $E(G)$ whose ends are x and y. For two subgraphs G and H of a graph, we use $G \cup H$ and $G \cap H$ to denote their union and intersection, respectively. For convenience, we use $A := B$ to rename B as A or to define A as B.

Let G be a graph. Given $x \in V(G)$, let $N_G(x) := \{y \in V(G) : yx \in E(G)\}$. Given $S \subseteq V(G)$, we let $N_G(S) := \{x \in V(G) - S : xy \in E(G) \text{ for some } y \in S\}$. For a subgraph H of G, we write $N_G(H) := N_G(V(H))$. When ambiguity is not a concern, we may simply use $V, E, N(x), N(S)$, and $N(H)$. Let P be a path between vertices u and v in G; then P is called a u-v path, and u and v are called the *ends* of P. Given vertices x, y on P, we let xPy denote the path in P with ends x and y. Let X be a set of 2-element subsets of $V(G)$; then $G + X$ denotes the graph with vertex set $V(G)$ and edge set $E(G) \cup X$.

Again, let G be a graph. Given $S \subseteq V(G)$, let $G[S]$ denote the subgraph of G induced by S, and let $G - S := G[V(G) - S]$. For any $S \subseteq E(G)$, we let $G - S$ denote the graph with vertex set $V(G)$ and edge set $E(G) - S$. If $S = \{s\} \subseteq V(G) \cup E(G)$, we let $G - s := G - S$. For any $\{u, v\} \subseteq V(G)$, $G - uv := G$ if $uv \notin E(G)$, and $G - uv := G - \{uv\}$ if $uv \in E(G)$. A cycle C in G is an *induced* cycle if $G[V(C)] = C$, and it is *nonseparating* if $G - V(C)$ is connected.

A *plane graph* is a graph which is drawn in the plane with no pair of edges crossing. The *faces* of a plane graph are the connected components (in topological sense) of its complement in the plane. The *infinite face* of a plane graph is its unbounded face. The boundary of a face is called a *facial walk*, or *facial cycle* if it is a cycle. A graph is *planar* if it is isomorphic to a plane graph.

An *ear decomposition* of a connected graph G is a sequence $\mathcal{E}_G = (P_0, P_1, P_2, \ldots, P_k)$ which satisfies the following three conditions: (1) P_0 is a cycle in G; (2) for each $1 \leq i \leq k$, P_i is a path in G with ends x and y, $(\bigcup_{j=0}^{i-1} E(P_j)) \cap E(P_i) = \emptyset$, and $(\bigcup_{j=0}^{i-1} V(P_j)) \cap V(P_i) = \{x, y\}$; and (3) $G = (\bigcup_{j=0}^{k} V(P_j), \bigcup_{j=0}^{k} E(P_j))$. The elements of $\mathcal{E}_G$ are called *ears* of G.

[*] Received by the editors September 14, 2001; accepted for publication (in revised form) February 28, 2003; published electronically July 30, 2003.
http://www.siam.org/journals/sidma/16-4/39518.html

[†] School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 (curran@math.gatech.edu, yu@math.gatech.edu). The research of the second author was partially supported by NSF grant DMS 9970527.

Let $T_1, T_2, \ldots, T_m$ be spanning trees of a graph G, and let $r \in V(G)$. We say that $T_1, \ldots, T_m$ are *independent spanning trees of G rooted at r* if, for any $x \in V(G)$ and for any distinct $i, j \in \{1, \ldots, m\}$, the r-x paths in T_i and T_j are vertex-disjoint in G except at r and x. Given any vertex r in a 2-connected graph G, it is known that G contains two independent spanning trees rooted at r; Itai and Rodeh [5] constructed these trees using an ear decomposition of G. In [13], Zehavi and Itai showed that if G is a 3-connected graph and $r \in V(G)$, then G contains three independent spanning trees rooted at r. Their proof relied on the property that every 3-connected graph with at least five vertices contains a *contractible* edge—one whose contraction results in a new 3-connected graph. Since this property is unique to 3-connected graphs, there is little hope of generalizing their approach to cases with higher connectivity. Cheriyan and Maheshwari [3] independently showed the 3-connected result; however, they used an ear decomposition of the graph, albeit a more restrictive type called a *nonseparating* ear decomposition. This nonseparating ear decomposition $(P_0, P_1, \ldots)$ imposes connectivity conditions between P_i and $G - (\bigcup_{j=1}^{i} V(P_j))$ and also on $G - (\bigcup_{j=1}^{i} V(P_j))$. The first ear P_0 of a nonseparating ear decomposition is guaranteed by the following result of Tutte [12].

Theorem 1.1. *Let G be a 3-connected graph, let $st \in E(G)$, and let $r \in V(G) - \{s, t\}$. Then G contains a nonseparating induced cycle through st and avoiding r.*

In [13], it is conjectured that, for any vertex r in a k-connected graph G, there exist k independent spanning trees of G rooted at r. The 4-connected case is the first case where the existence of a contractible edge is not guaranteed. Huck [4] has shown that every 4-connected planar graph contains four independent spanning trees rooted at any given vertex. We would like to devise a 4-connected version of the nonseparating ear decomposition which could be used to construct four independent spanning trees (rooted at any given vertex r) in 4-connected graphs. The first step in building such a decomposition is to find a cycle C through the "root" r which leaves a certain degree of connectivity in $G - (V(C) - \{r\})$. Our construction of such a cycle is the main result of this paper.

Theorem 1.2. *Let G be a 4-connected graph, and let $ra \in E(G)$. Then G contains a cycle C through ra such that $G - (V(C) - \{r\})$ is 2-connected.*

While motivated by the search for an ear decomposition, this result is of its own interest. For example, variations of our proof give the following two results.

Theorem 1.3. *Let G be a 5-connected graph, and let $e \in E(G)$. Then G contains an induced cycle C through e such that $G - V(C)$ is 2-connected.*

Theorem 1.4. *Let G be a planar 4-connected graph, and let C be a nonseparating induced cycle in G. Then, for any $r \in V(C)$, $G - (V(C) - \{r\})$ is 2-connected.*

Note that Theorem 1.3 closely parallels Theorem 1.1, and a 6-connected version was shown by Kriesell [6]. As a consequence of Theorem 1.3, we can deduce the following result (also proved in [6] and [2]): for any 5-connected graph G and $\{a, b\} \subseteq V(G)$, G contains an induced a-b path P such that $G - V(P)$ is 2-connected. This result in turn provides some evidence for the following conjecture of Lovász [7]: Given any positive integer k, there exists some positive integer $f(k)$ with the property that, for any given vertices x and y in a $f(k)$-connected graph G, there exists an induced x-y path P in G such that $G - V(P)$ is k-connected.

We note that a cycle in a 3-connected plane graph is nonseparating and induced if and only if it is a facial cycle. Therefore, Theorem 1.4 says that if G is a 4-connected plane graph and C is any facial cycle of G, then, for each $r \in V(C)$, $G - (V(C) - \{r\})$ is 2-connected.

Our paper will progress as follows. In section 2, we establish some convenient definitions and state some known results. Three technical lemmas will be shown: two are deduced from well-known results on paths in graphs, and the last is an independent lemma necessary for proving Theorems 1.2 and 1.3. In section 3, we prove Theorem 1.2; in fact, we will prove a stronger result, Theorem 3.1. Its proof constructs a nonseparating cycle C for Theorem 1.2 and reveals some structure which will be useful in constructing nonseparating ear decompositions of 4-connected graphs. In section 4, we modify our proof of Theorem 3.1 to deduce Theorem 1.3. We also prove Theorem 1.4. In section 5, we offer some concluding remarks.

2. Preliminary results. For notational convenience, we begin this section with the following definition. Let G be a graph with distinct vertices $a, b, c,$ and d. We say that the ordered quintuple (G, a, b, c, d) is *planar* if G can be drawn in a closed disc in the plane with no pair of edges crossing such that a, b, c, d occur on the boundary of the disc in that cyclic order.

Establishing planarity of certain subgraphs will be critical in the proof of Theorem 3.1 in section 3. To this end, we use a well-known result of Seymour [9]. Different versions of this result were obtained independently by Chakravarti and Robertson [1] and by Thomassen [10].

THEOREM 2.1. *Let u_1, v_1, u_2, v_2 be distinct vertices of a graph $G = (V, E)$. Then exactly one of the following is true:*

(1) *There are vertex-disjoint paths joining u_1 to v_1 and u_2 to v_2, respectively.*

(2) *For some integer $k \geq 0$, there are pairwise disjoint sets $A_1, A_2, \dots, A_k \subseteq V - \{u_1, u_2, v_1, v_2\}$ such that*

(a) *for $1 \leq i \neq j \leq k$, $N(A_i) \cap A_j = \emptyset$,*

(b) *for $1 \leq i \leq k$, $|N(A_i)| \leq 3$,*

(c) *if G' is the graph obtained from G by, for each i, deleting A_i and adding new edges joining every pair of distinct vertices in $N(A_i)$, and also for $j = 1, 2$ adding an edge e_j joining u_j to v_j, then G' may be drawn in the plane with no pairs of edges crossing except e_1, e_2, which cross once.*

The following corollary is a simpler version of Theorem 2.1, attained by imposing some connectivity conditions.

COROLLARY 2.2. *Let $u_1,\ u_2,\ v_1,\ v_2$ be distinct vertices of a graph G. Suppose that for any $T \subseteq V(G)$ with $|T| \leq 3$, every component of $G - T$ contains at least one element of $\{u_1,\ u_2,\ v_1,\ v_2\}$. Then exactly one of the following is true:*

(1) *There are vertex-disjoint paths joining u_1 to v_1 and u_2 to v_2, respectively.*

(2) *$(G, u_1,\ u_2,\ v_1,\ v_2)$ is planar.*

Proof. Clearly, (1) and (2) are mutually exclusive because of planarity. We know that either (1) or (2) of Theorem 2.1 must hold. If (1) of Theorem 2.1 holds, then (1) of Corollary 2.2 holds. So assume (2) of Theorem 2.1 holds. Then $\{u_1,\ u_2,\ v_1,\ v_2\} \cap A_i = \emptyset$ for all $1 \leq i \leq k$. Hence $G[A_i]$ consists of those components of $G - N(A_i)$ containing no element of $\{u_1,\ u_2,\ v_1,\ v_2\}$, contradicting our hypothesis. So no A_i may exist. Let $G', e_1,$ and e_2 be described as in (c) of Theorem 2.1. Observe that $(G' - \{e_1, e_2\}, u_1, u_2, v_1, v_2)$ is planar. But $G' - \{e_1, e_2\} = G$. □

Let P be a subgraph of G. Then a *P-bridge* of G is a subgraph of G which is induced by either (1) an edge in $E(G) - E(P)$ with both ends on P or (2) edges of a component of $G - V(P)$ and edges of G from that component to P. For any P-bridge B of G, the set $V(B \cap P)$ is the set of *attachments* of B on P.

In the proof of Theorem 3.1, we will reroute paths through planar subgraphs. To this end, we need a well-known theorem of Thomassen [11].

THEOREM 2.3. *Let G be a 2-connected plane graph, F be a facial cycle of G, $x \in V(F)$, $e \in E(F)$, and $y \in V(G) - \{x\}$. Then G contains an x-y path P through e such that*

(1) *every P-bridge of G has at most three attachments on P, and*

(2) *every P-bridge of G containing an edge of F has two attachments on P.*

Note that if G is 4-connected and $|V(P)| \geq 4$, then P is a Hamilton path in G. We will apply Theorem 2.3 to certain planar subgraphs of a 4-connected graph. Therefore, it will be convenient to have the following corollary.

COROLLARY 2.4. *Let (G, a, c, b, d) be planar such that $G - \{c, d\}$ contains an $a - b$ path. Assume that, for any $T \subseteq V(G)$ with $|T| \leq 3$, every component of $G - T$ contains an element of $\{a, c, b, d\}$. Then $G - \{c, d\}$ contains an a-b Hamilton path.*

Proof. Let $G' := (G - d) + \{\{b, c\}, \{a, c\}\}$. We first show that G' is 2-connected. Suppose on the contrary that G' is not 2-connected. Let x be a cut vertex of G'. Because $G - \{c, d\}$ contains an a-b path, $\{a, b, c\}$ is contained in a cycle of G'. Therefore, $\{a, b, c\}$ is contained in an x-bridge of G', and G' has another x-bridge B such that $(V(B) - \{x\}) \cap \{a, b, c\} = \emptyset$. Hence $B - x$ is a component of $G - T$, where $T := \{x, d\}$, and $V(B - x) \cap \{a, b, c, d\} = \emptyset$, a contradiction.

Observe that G' is planar and may be drawn in the plane so that ac, bc, and $N(d)$ are on the cycle F which bounds its infinite face. Applying Theorem 2.3 (with G', a, c, bc as G, x, y, e, respectively), G' has an a-c path P through bc satisfying (1) and (2) of Theorem 2.3. Note that $ac \notin E(P)$ because $bc \in E(P)$.

We proceed to show that every P-bridge of G' is induced by a single edge, and so P must be a Hamilton path in G'. Let B be a P-bridge of G' such that $V(B) - V(P) \neq \emptyset$, and let $T := V(B) \cap V(P)$. Since a, b, and c are all on P, $\{a, b, c\} \cap V(B) \subseteq T$. Thus $B - T$ is a component of $G - (\{d\} \cup T)$ containing no element of $\{a, b, c, d\}$. If $|T| \leq 2$, then $|\{d\} \cup T| \leq 3$, contradicting our hypothesis. Since P must satisfy (1) of Theorem 2.3, we may assume $|T| = 3$. Then by (2) of Theorem 2.3, $E(B) \cap E(F) = \emptyset$, and hence $(V(B) - T) \cap N(d) = \emptyset$. Therefore, $B - T$ is a component of $G - T$ for which $V(B - T) \cap \{a, b, c, d\} = \emptyset$, a contradiction.

Thus $P - c$ is an a-b Hamilton path in $G - \{c, d\}$, as required. □

Finally, we prove the following important technical lemma. We rely heavily on this result in the proof for Theorems 1.3 and 3.1.

LEMMA 2.5. *Let G be a connected graph, let $S \subseteq V(G)$, and let $a, a', b, b' \in S$. Suppose*

(i) *G contains vertex-disjoint paths joining a to a' and b to b', respectively, and*

(ii) *for any $T \subseteq V(G)$ with $|T| \leq 2$, every component of $G - T$ contains an element of S.*

Then $G - \{b, b'\}$ contains an induced a-a' path P such that

(1) *$\{b, b'\}$ is contained in a component of $G - V(P)$, and*

(2) *every component of $G - V(P)$ contains an element of S.*

Proof. Let $\mathcal{P}$ be the set of those induced a-a' paths P in $G - \{b, b'\}$ such that $\{b, b'\}$ is contained in a component of $G - V(P)$. By (i), $\mathcal{P} \neq \emptyset$. For each $P \in \mathcal{P}$, let B_P denote the component of $G - V(P)$ containing $\{b, b'\}$, and let T_P denote the union of those components C of $G - V(P)$ for which $V(C) \cap S = \emptyset$.

Select $P \in \mathcal{P}$ such that (a) $|V(B_P)|$ is maximum and then (b) $|V(T_P)|$ is minimum. If $|V(T_P)| = 0$, then Lemma 2.5 holds. So assume $|V(T_P)| \neq 0$.

Let $\mathcal{C} = \{C_1, C_2, \ldots, C_n\}$ be the set of components of $G - V(P)$ such that $C_i \subseteq T_P$. For $i = 1, \ldots, n$, we let a_i and a_i' be the elements of $N(C_i) \cap V(P)$ such that $a_i P a_i'$ is maximal. Let the notation be chosen so that a, a_i, a_i', a' occur on P in the order

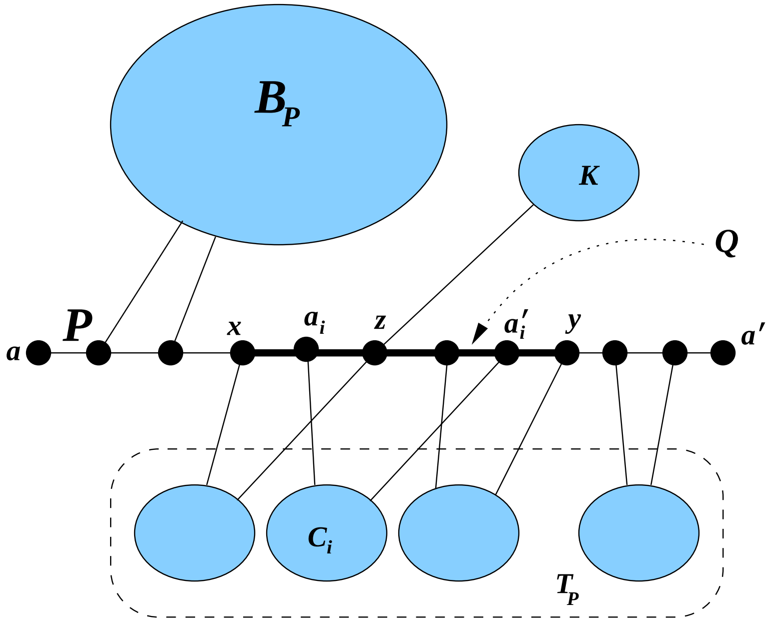


FIG. 1. *Lemma* 2.5.

listed. Let $\mathcal{K}$ be the auxiliary graph such that $V(\mathcal{K}) = \mathcal{C}$, and $C_iC_j \in E(\mathcal{K})$ if and only if $E(a_iPa_i') \cap E(a_jPa_j') \neq \emptyset$. Let $\mathcal{F}$ be a component of $\mathcal{K}$. From construction, $Q := \bigcup_{C_i \in V(\mathcal{F})} a_iPa_i'$ is a subpath of P. Let x and y be the ends of Q. See Figure 1 for an illustration.

Note that $V(Q) \neq \{x, y\}$, and there must exist some component K of $G - V(P)$ such that $V(K) \cap S \neq \emptyset$ and $N(K) \cap (V(Q) - \{x, y\}) \neq \emptyset$. For otherwise, the subgraph H of G induced by $(\bigcup_{C_i \in V(\mathcal{F})} V(C_i)) \cup (V(Q) - \{x, y\})$ is a union of components of $G - \{x, y\}$. But H contains no element of S, so $T := \{x, y\}$ violates hypothesis (ii).

Let $z \in N(K) \cap (V(Q) - \{x, y\})$. Then there exists some $C_i \in V(\mathcal{F})$ such that $z \in V(a_iPa_i') - \{a_i, a_i'\}$. Otherwise, for any $C_j \in V(\mathcal{F})$, either $\{a_j, a_j'\} \subseteq V(xPz)$ or $\{a_j, a_j'\} \subseteq V(zPy)$. Let $\mathcal{F}_x$ be the subgraph of $\mathcal{F}$ induced by those C_j such that $\{a_j, a_j'\} \subseteq V(xPz)$, and let $\mathcal{F}_y$ be the subgraph of $\mathcal{F}$ induced by those C_j with $\{a_j, a_j'\} \subseteq V(zPy)$. Then, for any $C_k \in V(\mathcal{F}_x)$ and $C_l \in V(\mathcal{F}_y)$, $E(a_kPa_k') \cap E(a_lPa_l') = \emptyset$. Hence $\mathcal{F}$ is not connected, a contradiction.

Choose any induced a_i-a_i' path R in $G\,[V(C_i) \cup \{a_i, a_i'\}]$, and let $X := aPa_i \cup R \cup a_i'Pa'$. Clearly, X is an induced a-a' path in G, and B_P is contained in a component of $G - V(X)$. Hence $X \in \mathcal{P}$ and $V(B_P) \subseteq V(B_X)$. But $V(T_X) \subseteq V(T_P) - V(R \cap C_i)$, contradicting (a) or (b). □

3. 4-connected Graphs.

We prove our main result in this section. For the sake of the proof and for the application to independent trees (as described in section 1), we prove the following stronger result.

THEOREM 3.1. *Let G be a 4-connected graph, let $r \in V(G)$, and let $e \in E(G)$ such that e is incident with r. Then there exists a cycle C through e in G such that $G - (V(C) - \{r\})$ is 2-connected. Moreover, for some integer $m \geq 0$, there exist edge-disjoint subpaths P_t of $C - r$ with ends a_t and b_t, $1 \leq t \leq m$, such that*

(i) *every chord of C has both ends on some P_t, and*

(ii) *for each* $t \in \{1, \ldots, m\}$, *there exist distinct* $c_t, d_t \in V(G) - V(C)$ *such that* $G[V(P_t) - \{a_t, b_t\}]$ *is a component of* $G - \{a_t, b_t, c_t, d_t\}$ *and* $(G[V(P_t) \cup \{c_t, d_t\}] - c_t d_t, a_t, c_t, b_t, d_t)$ *is planar.*

Proof. Let $\mathcal{D}$ denote the set of those induced cycles D in G for which $e \in E(D)$, $G - (V(D) - \{r\})$ is connected, and r is contained in a unique nontrivial block of $G - (V(D) - \{r\})$.

By Theorem 1.1, G contains a nonseparating induced cycle D through e. Since G is 4-connected, r must have at least four neighbors, and since D is induced, exactly two of those neighbors lie on D. Thus, $G - (V(D) - \{r\})$ is connected. Further, since $G - V(D)$ is connected, r is contained in a unique nontrivial block of $G - (V(D) - \{r\})$. Hence $\mathcal{D} \neq \emptyset$.

For each $D \in \mathcal{D}$, let B_D denote the unique nontrivial block of $G - (V(D) - \{r\})$ containing r. So B_D is 2-connected. We choose $D \in \mathcal{D}$ so that

(a) $|V(B_D)|$ is maximum.

For convenience, let $H := G - (V(D) - \{r)\}$, let $P := D - r$, and let a, b be the ends of P. If H is 2-connected, then $C := D$ gives the desired cycle, and in this case, $m = 0$ and no P_t may exist. So assume that H is not 2-connected. Let $X := \{v_1, v_2, \ldots, v_n\}$ be the set of cut vertices of H which are contained in B_D. Observe that $r \notin X$. Let $B_i^1, B_i^2, \ldots, B_i^{n_i}$ denote the v_i-bridges of H other than B_D, where $n_i \geq 1$ because v_i is a cut vertex of H. Let $\mathcal{B} := \{B_i^j \ : \ 1 \leq i \leq n, 1 \leq j \leq n_i\}$. Note that $r \notin V(B_i^j) \cup N(B_i^j - v_i)$ for every $B_i^j \in \mathcal{B}$.

Because G is 4-connected, $B_i^j - v_i$ has at least three neighbors on P. Let a_i^j, b_i^j be the neighbors of $B_i^j - v_i$ on P such that $a_i^j P b_i^j$ is maximal and a, a_i^j, b_i^j, b occur on P in this order. See Figure 2. For convenience, let $P_i^j := a_i^j P b_i^j$, and let $Q_i^j = P_i^j - \{a_i^j, b_i^j\}$. Because G is 4-connected, we have the following two observations.

(b) $V(Q_i^j) \neq \emptyset$ and $N(Q_i^j) \cap V(B_i^j - v_i) \neq \emptyset$.

(c) $N(Q_i^j) \not\subseteq V(B_i^j) \cup V(D)$.

CLAIM 1. *For each* B_i^j, *there exists a* $D_i^j \in \mathcal{D}$ *such that*

(i) $V(D_i^j) \cap (V(H) - V(B_i^j)) = \{r\}$,

(ii) $v_i \notin V(D_i^j)$, *and*

(iii) $V(D_i^j) \cap V(Q_i^j) = \emptyset$.

Proof of claim. Consider the graph $G_i^j := G[V(B_i^j) \cup \{a_i^j, b_i^j\}]$. Let $S = \{v_i, a_i^j, b_i^j\} \cup (N(Q_i^j) \cap V(B_i^j))$. Since G is 4-connected, for any $T \subseteq V(G_i^j)$ with $|T| \leq 3$, every component of $G_i^j - T$ must contain an element of S. Further, since B_i^j is a v_i-bridge of H, there must exist an a_i^j-b_i^j path in $G_i^j - v_i$. Applying Lemma 2.5 (with G_i^j, a_i^j, b_i^j, v_i as $G, a, a', b = b'$, respectively), there must exist an induced a_i^j-b_i^j path S_i^j in $G_i^j - v_i$ such that if F is a component of $G_i^j - V(S_i^j)$, then F contains some element of S.

Let $D_i^j := (D - V(Q_i^j)) \cup S_i^j$. Then D_i^j is a cycle in G. By construction, (i) $V(D_i^j) \cap (V(H) - V(B_i^j)) = \{r\}$, (ii) $v_i \notin V(D_i^j)$, and (iii) $V(D_i^j) \cap V(Q_i^j) = \emptyset$. Note that $e \in E(D_i^j)$. It remains to show that $D_i^j \in \mathcal{D}$.

Because D and S_i^j are induced subgraphs of G and by the definitions of a_i^j and b_i^j, it is easy to see that D_i^j is an induced cycle in G. So we need to show that $G - (V(D_i^j) - \{r\})$ is connected, and r is contained in a unique nontrivial block of $G - (V(D_i^j) - \{r\})$. Since $V(B_D \cap D_i^j) = V(B_D \cap D) = \{r\}$, $B_D - r \subseteq G - V(D_i^j)$. Since $B_D - r$ is connected and D_i^j is induced, it suffices to show that, for each $x \in V(G) - V(D_i^j)$, $G - V(D_i^j)$ has a path from x to $V(B_D) - \{r\}$.

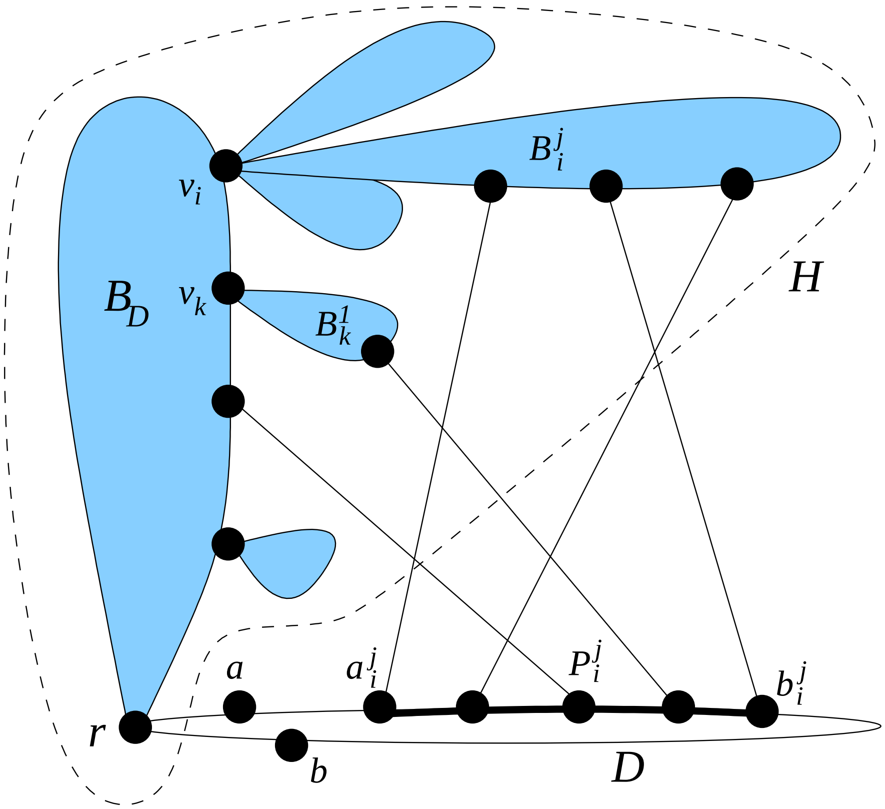


FIG. 2. *Theorem* 3.1.

Suppose $x \in V(B_k^l)$ for some $B_k^l \neq B_i^j$. By construction, $V(B_k^l) \cap V(D_i^j) = \emptyset$. Thus, B_k^l (and hence $G - V(D_i^j)$) has a path from x to $v_k \in V(B_D) - \{r\}$. So assume $x \in V(B_i^j) \cup V(Q_i^j)$.

If $x \in V(Q_i^j)$, then, since $N(Q_i^j) \not\subseteq V(B_i^j) \cup V(D)$ (by (c)) and $V(D_i^j) \cap V(Q_i^j) = \emptyset$ (by (iii)), $G - V(D_i^j)$ has a path from x to $V(B_D) - \{r\}$.

So let $x \in V(B_i^j)$. Let F denote the component of $G_i^j - V(S_i^j)$ containing x. If $v_i \in V(F)$, then F (and hence $G - V(D_i^j)$) contains a path from x to $v_i \in V(B_D) - \{r\}$. So assume that F has a neighbor of Q_i^j. Since $N(Q_i^j) \not\subseteq V(B_i^j) \cup V(D)$ (again, by (c)) and $V(D_i^j) \cap V(Q_i^j) = \emptyset$ (again, by (iii)), $G - V(D_i^j)$ must have a path from x to $V(B_D) - \{r\}$. □

For each $x \in V(H)$, we define x^* as follows. If $x \in V(B_i^j)$ for some $B_i^j \in \mathcal{B}$, then let $x^* = v_i$. If $x \in V(B_D)$, then define $x^* = x$.

CLAIM 2. *For any $B_i^j \in \mathcal{B}$ and for any $x, y \in (N(Q_i^j) \cap V(H)) - V(B_i^j)$, $x^* = y^*$.*

Proof of claim. Suppose that there are $x, y \in (N(Q_i^j) \cap V(H)) - V(B_i^j)$ such that $x^* \neq y^*$. Then G contains disjoint paths X and Y joining x to x^* and y to y^*, respectively, such that both X and Y are also disjoint from $D \cup (B_i^j - v_i) \cup (B_D - \{x^*, y^*\})$. Let $x', y' \in V(Q_i^j)$ such that $xx', yy' \in E(G)$. By Claim 1, there is some $D_i^j \in \mathcal{D}$ such that $V(D_i^j) \cap (V(H) - V(B_i^j)) = \{r\}$, $v_i \notin V(D_i^j)$, and $V(D_i^j) \cap V(Q_i^j) = \emptyset$. Then both B_D and the x^*-y^* path $X \cup xx'Q_i^j y'y \cup Y$ are contained in $B_{D_i^j}$. Hence

$|V(B_{D_i^j})| > |V(B_D)|$, and so D_i^j contradicts (a). □

Define a new graph $\mathcal{K}$ such that $V(\mathcal{K}) = \mathcal{B}$, and $B_i^j B_k^l \in E(\mathcal{K})$ if and only if $E(P_i^j) \cap E(P_k^l) \neq \emptyset$. Let $\mathcal{A}_1, \dots, \mathcal{A}_m$ be the components of $\mathcal{K}$. For each $t \in \{1, \dots, m\}$, let $V_t := \{v_i : B_i^j \in V(\mathcal{A}_t)$ for some $1 \leq j \leq n_i\}$, $P'_t := \bigcup_{B_i^j \in V(\mathcal{A}_t)} P_i^j$, and $B_t := \bigcup_{B_i^j \in V(\mathcal{A}_t)} B_i^j$. By definition, each P'_t is a subpath of P, $E(P'_s) \cap E(P'_t) = \emptyset$ for all $s \neq t$. Without loss of generality, assume that $P'_1, \dots, P'_m$ occur on P from a to b in the order listed. Let a_t and b_t be the ends of P'_t such that a, a_t, b_t, b occur on P in that order, and let $Q_t := P'_t - \{a_t, b_t\}$. See Figure 3 for an example with $t = 3$.

CLAIM 3. *For each* $t \in \{1, \dots, m\}$, $|V_t| \leq 2$.

Proof of claim. Assume for a contradiction that $|V_t| \geq 3$.

Case 1 ($\mathcal{A}_t$ contains an induced path $B_i^j B_k^l B_p^q$ with $i \neq p$). Then $E(P_i^j) \cap E(P_p^q) = \emptyset$, $E(P_i^j) \cap E(P_k^l) \neq \emptyset$, and $E(P_k^l) \cap E(P_p^q) \neq \emptyset$. Hence we may assume that the vertices a, a_i^j, b_i^j, a_p^q, b_p^q, b occur on P in the order listed and that the vertices a, a_k^l, b_i^j, a_p^q, b_k^l, b occur on P in the order listed. Moreover, $a_k^l \neq b_i^j$ and $a_p^q \neq b_k^l$. Let $x \in V(B_i^j) - \{v_i\}$ such that $xb_i^j \in E(G)$, and let $y \in V(B_p^q) - \{v_p\}$ such that $ya_p^q \in E(G)$. Then $x, y \in \big(N(Q_k^l) \cap V(H)\big) - V(B_k^l)$ and $x^* = v_i \neq v_p = y^*$, contradicting Claim 2.

Case 2 ($\mathcal{A}_t$ contains a triangle $B_i^j B_k^l B_p^q B_i^j$ with $i \neq k \neq p \neq i$). First, we prove that one of the following must be true: $N(Q_i^j) \cap (V(B_k^l) - \{v_k\}) \neq \emptyset$ and $N(Q_i^j) \cap (V(B_p^q) - \{v_p\}) \neq \emptyset$, or $N(Q_k^l) \cap (V(B_i^j) - \{v_i\}) \neq \emptyset$ and $N(Q_k^l) \cap (V(B_p^q) -$

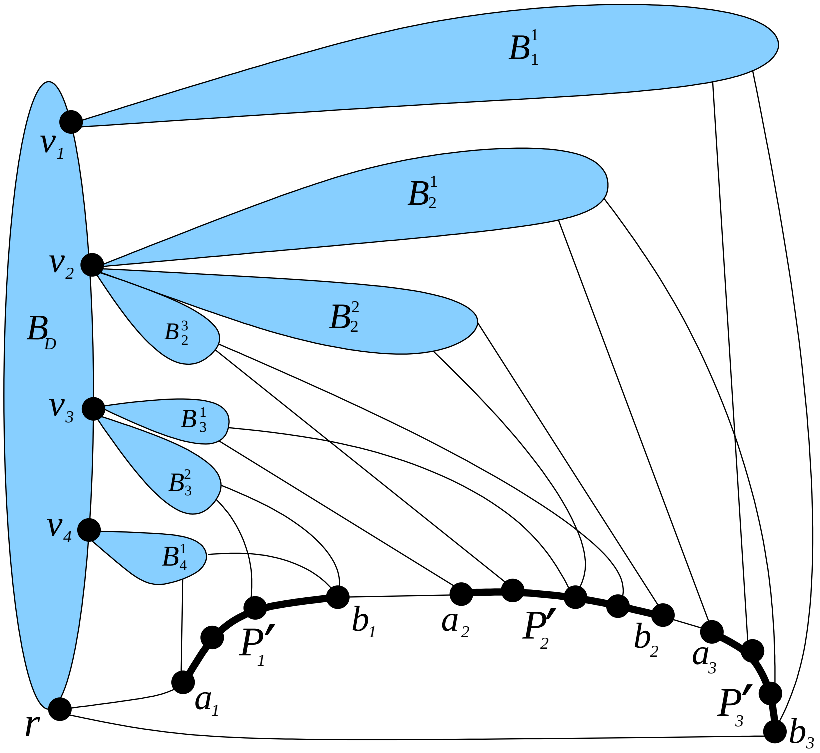


FIG. 3. *Claims* 2 *and* 3.

$\{v_p\}) \neq \emptyset$, or $N(Q_p^q) \cap (V(B_i^j) - \{v_i\}) \neq \emptyset$ and $N(Q_p^q) \cap (V(B_k^l) - \{v_k\}) \neq \emptyset$. Assume from symmetry that a, a_i^j, a_k^l, a_p^q, b, not necessarily distinct, occur on P in that order. Because $E(P_i^j) \cap E(P_p^q) \neq \emptyset$, then $b_i^j \in V(a_p^q Pb - a_p^q)$. Similarly, $b_k^l \in V(a_p^q Pb - a_p^q)$. If $a_i^j \neq a_k^l$, then $\emptyset \neq N(a_k^l) \cap (V(B_k^l) - \{v_k\}) \subseteq N(Q_i^j) \cap (V(B_k^l) - \{v_k\})$ and $\emptyset \neq N(a_p^q) \cap (V(B_p^q) - \{v_p\}) \subseteq N(Q_i^j) \cap (V(B_p^q) - \{v_p\})$. So assume $a_i^j = a_k^l$. Then from symmetry we may assume that $b_i^j \in V(a_p^q P b_k^l)$. Hence $Q_i^j \subseteq Q_k^l$. If $a_k^l \neq a_p^q$, then from (b), $\emptyset \neq N(Q_i^j) \cap (V(B_i^j) - \{v_i\}) \subseteq N(Q_k^l) \cap (V(B_i^j) - \{v_i\})$ and $\emptyset \neq N(a_p^q) \cap (V(B_p^q) - \{v_p\}) \subseteq N(Q_k^l) \cap (V(B_p^q) - \{v_p\})$. So assume $a_k^l = a_p^q$. We may now assume from symmetry that $a, a_i^j, b_i^j, b_k^l, b_p^q, b$, not necessarily distinct, occur on P in that order. Then $Q_i^j \subseteq Q_k^l \subseteq Q_p^q$, and from (b), $\emptyset \neq N(Q_i^j) \cap (V(B_i^j) - \{v_i\}) \subseteq N(Q_p^q) \cap (V(B_i^j) - \{v_i\})$ and $\emptyset \neq N(Q_k^l) \cap (V(B_k^l) - \{v_k\}) \subseteq N(Q_p^q) \cap (V(B_k^l) - \{v_k\})$.

By symmetry, we may assume $N(Q_k^l) \cap (V(B_i^j) - \{v_i\}) \neq \emptyset$ and $N(Q_k^l) \cap (V(B_p^q) - \{v_p\}) \neq \emptyset$. Then there exist $x \in N(Q_k^l) \cap (V(B_i^j) - \{v_i\})$ and $y \in N(Q_k^l) \cap (V(B_p^q) - \{v_p\})$. Hence $x, y \in \big(N(Q_k^l) \cap V(H)\big) - V(B_k^l)$ and $x^* = v_i \neq v_p = y^*$, contradicting Claim 2.

Case 3 (neither Case 1 nor Case 2). Because Case 1 does not occur, for any induced path $B_i^j B_k^l B_p^q$ in $\mathcal{A}_t$, we must have $i = p$. Because Case 2 does not occur and since $|V_t| \geq 3$, $\mathcal{A}_t$ is not complete. Further, for any induced path R in $\mathcal{A}_t$, R contains no subpath $B_i^j B_k^l B_p^q$ with $i \neq p$. Hence R may only take on two forms: (i) $V(R)$ may be composed of B_i^j's for a fixed i, or (ii) R may be alternating between B_i^j's and B_k^l's for fixed i, k. Since we assume $|V_t| \geq 3$, $\mathcal{A}_t$ contains B_i^j, B_k^l, B_p^q with $i \neq k \neq p \neq i$. Choose an induced path R_1 in $\mathcal{A}_t$ from B_i^j to B_k^l and another induced path R_2 in $\mathcal{A}_t$ from B_k^l to B_p^q. Clearly these paths cannot be of type (i) and so must be of type (ii). But then $R_1 \cup R_2$ contains a subpath $B_i^{j_0} B_k^l B_p^{q_0}$ such that $i \neq k \neq p \neq i$, and we would have Case 1 or Case 2, a contradiction. $\square$

Claim 4. *For each* $t \in \{1, \ldots, m\}$ *with* $|V_t| = 1$, *there exists* $d_t \in V(B_D) - (V_t \cup \{r\})$ *such that* $N(Q_t) - (V(B_t) \cup V(D)) = \{d_t\}$.

Proof of claim. Suppose $|V_t| = 1$. Because G is 4-connected and D is induced in G, Q_t must have a neighbor $x \in V(B_D) - (V_t \cup \{r\})$. By the definition of Q_t, we may assume that $x \in N(Q_i^j)$, and we choose such Q_i^j to be maximal. If $N(Q_t) - (V(B_t) \cup V(D)) = \{x\}$, then $d_t := x$ is the desired vertex. So we assume that there is some $y \in N(Q_t) - (V(B_t) \cup V(D))$ such that $y \neq x$. Then $y \in V(B_D) - (V_t \cup \{r, x\})$. Because $x, y \in V(B_D)$, $x^* = x$ and $y^* = y$. By Claim 2 and because $x^* = x \neq y = y^*$, $y \notin N(Q_i^j)$. Hence $|\mathcal{A}_t| \geq 2$, and so there exists some $B_k^l \in V(\mathcal{A}_t) - \{B_i^j\}$ such that $E(P_k^l) \cap E(P_i^j) \neq \emptyset$. By the maximality of Q_i^j, Q_i^j is not a proper subpath of Q_k^l, so either $a_k^l \in V(Q_i^j)$ or $b_k^l \in V(Q_i^j)$ or $Q_k^l = Q_i^j$. By (b), $N(Q_k^l) \cap (V(B_k^l) - \{v_k\}) \neq \emptyset$. Hence Q_i^j has a neighbor $z \in V(B_k^l) - \{v_k\}$. Then $x, z \in (N(Q_i^j) \cap V(H)) - V(B_i^j)$, $z^* = v_k$, and $x^* = x$. But since $v_k \in V_t$ and $x \notin V_t$, we have $z^* \neq x^*$. This contradicts Claim 2. $\square$

Claim 5. *For each* $t \in \{1, \ldots, m\}$ *with* $|V_t| = 2$, $N(Q_t) \cap V(B_D) \subseteq V_t$.

Proof of claim. Suppose $|V_t| = 2$, and assume that there is some $x \in (N(Q_t) \cap V(B_D)) - V_t$. Then $x^* = x \notin V_t$. By definition of Q_t, $x \in N(Q_i^j)$ for some $Q_i^j \in V(\mathcal{A}_t)$, and we may choose such Q_i^j to be maximal. Because $|V_t| \geq 2$, $|\mathcal{A}_t| \geq 2$. Hence there exists some $B_k^l \in V(\mathcal{A}_t) - \{B_i^j\}$ such that $E(P_k^l) \cap E(P_i^j) \neq \emptyset$. By the maximality of Q_i^j, Q_i^j is not a proper subpath of Q_k^l, so either $a_k^l \in V(Q_i^j)$ or $b_k^l \in V(Q_i^j)$ or $Q_k^l = Q_i^j$. From (b), $N(Q_k^l) \cap (V(B_k^l) - \{v_k\}) \neq \emptyset$. Hence Q_i^j has a neighbor

$y \in V(B_k^l) - \{v_k\}$. Note that $y^* = v_k \in V_t$ and $x^* = x \notin V_t$. Hence $x^* \neq y^*$. But $x, y \in (N(Q_i^j) \cap V(H)) - V(B_i^j)$, contradicting Claim 2. □

From Claims 3, 4, and 5, we may now identify the paths $P_1, \ldots, P_m$ and vertices a_t, b_t, c_t, d_t, $1 \leq t \leq m$, given in the statement of Theorem 3.1. We will then verify conditions (i) and (ii) in the conclusion of this theorem.

If $|V_t| = 2$, then let $V_t := \{c_t, d_t\}$ and let $G_t := G[V(B_t) \cup V(P_t')] - c_t d_t$. If $|V_t| = 1$, then by Claim 4, $N(Q_t) - (V(B_t) \cup V(D)) = \{d_t\} \subseteq V(B_D)$, and so, let $V_t := \{c_t\}$ and $G_t := G\,[V(B_t) \cup \{d_t\} \cup V(P_t')] - c_t d_t$. From Claims 4 and 5, $G_t - \{a_t, b_t, c_t, d_t\}$ is a component of $G - \{a_t, b_t, c_t, d_t\}$. We proceed to prove (i) and (ii). To do so, we will replace P_t' with an a_t-b_t Hamilton path P_t in $G_t - \{c_t, d_t\}$. First, we establish the following fact.

Claim 6. *The ordered quintuple $(G_t, a_t, c_t, b_t, d_t)$ is planar.*

Proof of claim. Since G is 4-connected, if $T \subseteq V(G_t)$ with $|T| \leq 3$, then any component of $G_t - T$ must contain an element of $\{a_t,\ b_t,\ c_t,\ d_t\}$. We may apply Corollary 2.2 to G_t, a_t, b_t, c_t, d_t (as G, u_1, v_1, u_2, v_2, respectively). Then either (1) G_t has disjoint paths joining a_t to b_t and c_t to d_t, respectively, or (2) $(G_t, a_t, c_t, b_t, d_t)$ is planar. If (2) holds, then we have our claim. So assume that (1) holds.

We may apply Lemma 2.5 to G_t, a_t, b_t, c_t, d_t (as G, a, a', b, b', respectively), letting $S = \{a_t, b_t, c_t, d_t\}$, and find an induced a_t-b_t path R in $G_t - \{c_t, d_t\}$ such that every component of $G_t - V(R)$ contains an element of S. Let $D' := (D - V(Q_t)) \cup R$. Then D' is an induced cycle in G and $G - V(D')$ is connected. It is then easy to see that $D' \in \mathcal{D}$. But both B_D and a c_t-d_t path in $G_t - V(R)$ are contained in $B_{D'}$. Thus $|V(B_{D'})| > |V(B_D)|$, contradicting (a). □

Since G is 4-connected, if $T \subseteq V(G_t)$ with $|T| \leq 3$, then every component of $G_t - T$ must contain an element of $\{a_t,\ b_t,\ c_t,\ d_t\}$. We may now apply Corollary 2.4 (with $(G_t, a_t, c_t, b_t, d_t)$ as (G, a, c, b, d)) to create an a_t-b_t Hamilton path P_t in $G_t - \{c_t, d_t\}$ for each $t \in \{1, \ldots, m\}$. By construction, $P_1, \ldots, P_m$ are all edge-disjoint paths. We let $C := (D - (\bigcup_{t=1}^m V(Q_t))) \cup (\bigcup_{t=1}^m P_t)$. Then C is a cycle in G, $e \in E(C)$, and $G - (V(C) - \{r\}) = B_D$ is 2-connected. Note that $G_t = G[V(P_t) \cup \{c_t, d_t\}] - c_t d_t$, and $G_t - \{a_t, b_t, c_t, d_t\} = G[V(P_t) - \{a_t, b_t\}]$ is a component of $G - \{a_t, b_t, c_t, d_t\}$. Hence $P_t,\ a_t,\ b_t,\ c_t,\ d_t, 1 \leq t \leq m$, satisfy condition (ii). We may easily see that condition (i) is also satisfied. Suppose there is a chord xy of C with $\{x, y\} \not\subseteq V(P_t)$ for all $1 \leq t \leq m$. If $x, y \notin V(Q_t)$ for any t, then xy is a chord of D. But D is induced in G, and this is a contradiction. So assume that $y \in V(Q_t)$ for some t, and then $x \notin V(P_t)$, contradicting the fact that $G_t - \{a_t, b_t, c_t, d_t\}$ is a component of $G - \{a_t, b_t, c_t, d_t\}$.

This completes the proof of Theorem 3.1. □

As a corollary, we have Theorem 1.2 by setting $e = ra$.

4. 5-connected graphs and planar graphs. In the proof of Theorem 3.1, we choose a cycle D to maximize a block B_D of $H = G - (V(D) - \{r\})$. After a sequence of five claims, we showed that any v_i-bridge other than B_D in H could be enclosed within a subgraph associated with a 4-cut. In a 5-connected graph, these 4-cuts cannot exist; this is the inspiration for Theorem 1.3. However, since we are now interested in the connectivity of $G - V(C)$, we must ensure that a nontrivial block exists in $G - V(C)$.

Since the proof of Theorem 1.3 closely parallels the proof of Theorem 3.1, we give only an outline and refer the reader to section 3, where possible. Following the proof, we demonstrate the relation of Theorem 1.3 to Lovász's conjecture.

Proof. Let G be a 5-connected graph, and let $e = ab$.

Claim 0. *There exists a nonseparating induced cycle D through e in G such that $G - V(D)$ contains a nontrivial block.*

Proof of claim. By Theorem 1.1, there exists a nonseparating induced cycle F through e in G such that $G - V(F)$ is connected. Note that $|V(G) - V(F)| \geq 2$ since F is an induced cycle and the minimum degree of G is at least five. If $G - V(F)$ contains a nontrivial block, then $D := F$ gives the desired cycle for the claim. So assume that $G - V(F)$ does not contain a nontrivial block; then $G - V(F)$ is a tree.

Let $T := G - V(F)$ and $P := F - e$. Since $|V(T)| \geq 2$, T has a leaf, say x. Then $|N(x) \cap V(F)| \geq 4$. Let a', b' be the neighbors of x on F such that $a'Pb'$ is maximal and a, a', b', b occur on P in that order. Let $P' := a'Pb'$, $Q' := P' - \{a', b'\}$, and $D := ((F - V(Q')) \cup \{x\}) + \{\{x, a'\}, \{x, b'\}\}$. Since $|N(x) \cap V(F)| \geq 4$, $V(Q') \neq \emptyset$. It follows from the choice of a' and b', D is an induced cycle in G.

Since $|N(v) \cap V(F)| = 2$, $|N(v) \cap V(T)| \geq 3$ and hence $|N(v) \cap (V(T) - \{x\})| \geq 2$. Therefore, since both Q' and $T - x$ are connected, $G - V(D) = G[(V(T) - \{x\}) \cup V(Q')]$ is connected. So D is nonseparating in G.

Let v_1, v_2 be distinct neighbors of v in $T - x$. Since $T - x$ is connected, $T - x$ has a v_1-v_2 path. This path together with v, vv_1, and vv_2 forms a cycle in $G - V(D)$. Hence $G - V(D)$ contains a nontrivial block. □

Let $\mathcal{D}$ denote the set of those nonseparating induced cycles D in G for which $e \in E(D)$ and $G - V(D)$ contains a nontrivial block. By Claim 0, $\mathcal{D} \neq \emptyset$. For any $D \in \mathcal{D}$, let B_D denote a block of $G - V(D)$ where $|V(B_D)|$ is maximum. We may choose $D \in \mathcal{D}$ so that

(a) $|V(B_D)|$ is maximum.

For convenience, let $H := G - V(D)$ and $P := D - e$. If H is 2-connected, then $C := D$ is the desired cycle. So assume that H is not 2-connected. Let $X := \{v_1, v_2, \ldots, v_n\}$ be the set of cut vertices of H which are contained in B_D. Let $B_i^1, B_i^2, \ldots, B_i^{n_i}$ denote the v_i-bridges of H other than B_D, where $n_i \geq 1$ because v_i is a cut vertex of H. Let $\mathcal{B} := \{B_i^j \ : \ 1 \leq i \leq n, 1 \leq j \leq n_i\}$.

Because G is 5-connected, $B_i^j - v_i$ has at least four neighbors on P. Let a_i^j, b_i^j be the neighbors of $B_i^j - v_i$ on P such that $a_i^j P b_i^j$ is maximal and a, a_i^j, b_i^j, b occur on P in this order. As in the proof of Theorem 3.1, we let $P_i^j := a_i^j P b_i^j$ and $Q_i^j := P_i^j - \{a_i^j, b_i^j\}$. We have the following two observations:

(b) $V(Q_i^j) \neq \emptyset$ and $N(Q_i^j) \cap V(B_i^j - v_i) \neq \emptyset$.

(c) $N(Q_i^j) \not\subseteq V(B_i^j) \cup V(D)$.

Claim 1. *For any B_i^j, there exists a $D_i^j \in \mathcal{D}$ such that* (i) $V(D_i^j) \cap (V(H) - V(B_i^j)) = \emptyset$, (ii) $v_i \notin V(D_i^j)$, *and* (iii) $V(D_i^j) \cap V(Q_i^j) = \emptyset$.

Proof of claim. Showing such a cycle exists is nearly identical to the proof of Claim 1 in section 3. Apply Lemma 2.5 (with $G[V(P_i^j) \cup V(B_i^j)], a_i^j, b_i^j$, and v_i as G, a, a', and $b = b'$, respectively) to create the cycle D_i^j through e. The only difference is that $V(D_i^j) \cap (V(H) - V(B_i^j)) = \emptyset$ in conclusion (i), since $V(H) \cap V(D) = \emptyset$. □

For any $x \in V(H)$, we may define x^* as in section 3. Similarly, we define the auxiliary graph $\mathcal{K}$, its components $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_m$, the sets $V_1, V_2, \ldots, V_m$, the subgraphs $B_1, B_2, \ldots, B_m$, and the paths $P'_1, P'_2, \ldots, P'_m, Q_1, Q_2, \ldots, Q_m$ as in section 3. With the same proofs for Claims 2, 3, 4, and 5 in the proof of Theorem 3.1, appealing to Claim 1 above where necessary, we have the following claims.

Claim 2. *For each $B_i^j \in \mathcal{B}$ and for any $x, y \in (N(Q_i^j) \cap V(H)) - V(B_i^j)$, $x^* = y^*$.*

Claim 3. *For each $t \in \{1, \ldots, m\}$, $|V_t| \leq 2$.*

Claim 4. *For each $t \in \{1, \ldots, m\}$ such that $|V_t| = 1$, there exists $d_t \in V(B_D) -$*

$(V_t \cup \{r\})$ such that $N(Q_t) - (V(B_t) \cup V(D)) = \{d_t\}$.

CLAIM 5. *For each $t \in \{1, \dots, m\}$ such that $|V_t| = 2$, $N(Q_t) \cap V(B_D) \subseteq V_t$.*

If $|V_t| = 2$, then let $V_t := \{c_t, d_t\}$, and let $G_t := G[V(B_t) \cup V(P_t')]$. If $|V_t| = 1$, then by Claim 4, $N(Q_t) - (V(B_t) \cup V(D)) = \{d_t\} \subseteq V(B_D)$, and so, let $V_t := \{c_t\}$ and $G_t := G\left[V(B_t) \cup \{d_t\} \cup V(P_t')\right]$. From Claims 4 and 5 above, $G_t - \{a_t, b_t, c_t, d_t\}$ is a component of $G - \{a_t, b_t, c_t, d_t\}$. This is a contradiction, since G is 5-connected.

Hence H is 2-connected, completing our proof. □

As a consequence of Theorem 1.3, we derive the following result of [6] and [2].

COROLLARY 4.1. *Let G be a 5-connected graph and $x, y \in V(G)$ be distinct. Then G contains an induced x-y path P such that $G - V(P)$ is 2-connected.*

Proof. If $xy \in E(G)$, then let P be the x-y path with $E(P) = \{xy\}$. Since G is 5-connected, $G - V(P) = G - \{x, y\}$ is 2-connected. So assume that $xy \notin E(G)$. Let $G' := G + xy$ and let $e = xy$. Note that G' is 5-connected. By Theorem 1.3, G' has an induced cycle C through e such that $G' - V(C)$ is 2-connected. Let $P := C - e$. Then P is an induced path in G. Since $G - V(P) = G' - V(C)$, then $G - V(P)$ is 2-connected. □

Corollary 4.1 shows that if $f(k)$ (of Lovász's conjecture, mentioned in section 1) exists, then $f(2) \leq 5$. The following example shows equality. Let G be the graph obtained from a cycle C on four or more vertices by adding two vertices x and y along with edges xa and ya for all $a \in V(C)$. Then G is 4-connected, but deleting any x-y path leaves only a path.

We proceed to prove Theorem 1.4.

Proof. Let G be a 4-connected planar graph, let C be a nonseparating induced cycle in G, and let $r \in V(C)$. Since G is 4-connected, r must have at least four neighbors, and since C is induced, exactly two of those neighbors lie on C. Thus, $G - (V(C) - \{r\})$ is connected. Further, since $G - V(C)$ is connected, r is not a cut vertex of $G - (V(C) - \{r\})$.

Let B denote the block of $G - (V(C) - \{r\})$ containing r. Clearly, B is 2-connected. For convenience, let $P := C - r$, and let $H := G - V(P)$. Suppose that H is not 2-connected. Let $v \in V(B)$ such that v is a cut vertex of H (and hence $v \neq r$), and let B' be a v-bridge of H such that $B' \neq B$. Let $x, y \in V(P) \cap N(B' - v)$ such that xPy is maximal. Since G is 4-connected, $G - \{x, y, v\}$ is connected; hence $G - \{x, y, v\}$ has a path P' from $V(B' \cup xPy) - \{x, y, v\}$ to $V(B) - \{v\}$. Because C is an induced cycle in G, B' is a v-bridge of H, and r is not a cut vertex of H, then P' is a path from $V(xPy) - \{x, y\}$ to some $w \in V(B) - \{v, r\}$ which is also disjoint from $(V(B) - \{w\}) \cup V(B') \cup (V(C) - (V(xPy) - \{x, y\}))$. Let z be the end of P' in $V(xPy) - \{x, y\}$. See Figure 4.

Since B is 2-connected, there exist a v-r path R_1 and v-w path P'' in B such that $V(R_1 \cap P'') = \{v\}$. Let $P_1 := P'' \cup P'$, $P_2 := zPx$, $P_3 := zPy$, let R_3 be the subpath of $C - x$ between y and r, and let R_2 be the subpath of $C - y$ between x and r.

Let $x', y' \in V(B') - \{v\}$ such that $xx', yy' \in E(G)$. Since $B' - v$ is connected, $B' - v$ contains a path Q from x' to y'. Note that B' contains a path Q_1 from v to some $s \in V(Q)$ such that $V(Q_1 \cap Q) = \{s\}$. Let $Q_2 := sQx'x$ and $Q_3 := sQy'y$.

Then $(\bigcup_{i=1}^{3} P_i) \cup (\bigcup_{i=1}^{3} Q_i) \cup (\bigcup_{i=1}^{3} R_i)$ is a subdivision of $K_{3,3}$. Hence G is not planar, contradicting our hypothesis. □

5. Concluding remarks. Theorem 1.3 suggests that Theorem 1.1 might be generalized.

CONJECTURE. *For any positive integer k, there exists some positive integer $f(k)$ such that if graph G is $f(k)$-connected, then, for any $e \in E(G)$, there exists an induced*

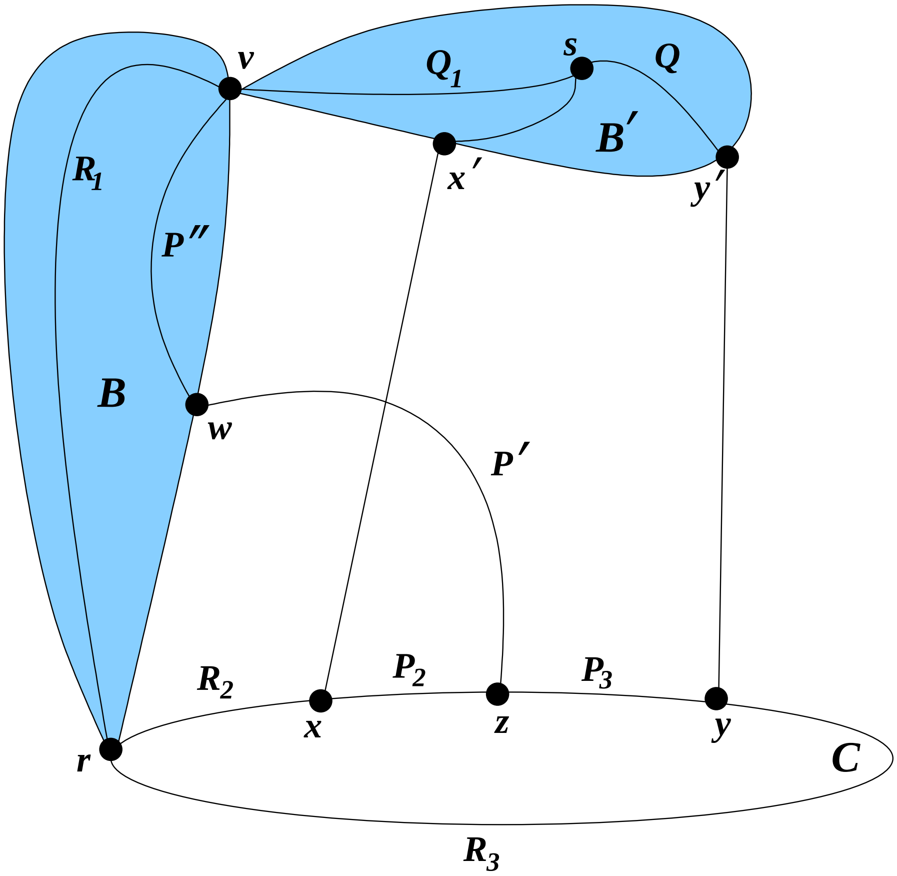


FIG. 4. *Theorem* 1.4.

cycle C through e in G such that $G - V(C)$ is k-connected.

This would imply Lovász's conjecture in exactly the same way that Theorem 1.3 implies the case for $k = 2$. Our proof for $k = 2$ relied on the highly useful block decomposition of connected graphs. Therefore, a natural problem is how to generalize block decomposition to k-connected graphs.

Our short-term goal is to find a nonseparating ear decomposition for 4-connected graphs which will yield four independent spanning trees rooted at a vertex. Theorem 3.1 provides the first ear. It is not incidental that the cycle in Theorem 3.1 has planar sections. Huck in [4] proved the existence of four independent spanning trees in every 4-connected planar graph. (Miura et al. [8] gave a linear time algorithm for finding four independent spanning trees in 4-connected planar graphs.) We intend to produce four independent trees in any 4-connected graph by building an ear decomposition with numerous planar sections and applying Huck's result. We hope that this will lend insight to an approach which could work for higher connectivity.

Acknowledgment. We would like to thank the referees for helpful suggestions. The proof of Claim 0 in the proof of Theorem 1.3 was suggested by one referee. Reference [8] was brought to us by another referee.

REFERENCES

[1] K. Chakravarti and N. Robertson, *Covering three edges with a bond in a nonseparable graph*, in Combinatorics 79, Part I, Ann. Discrete Math. 8, North–Holland, Amsterdam, 1980, p. 247.

[2] G. Chen, R. Gould, and X. Yu, *Graph connectivity after path removal*, Combinatorica, to appear.

[3] J. Cheriyan and S. N. Maheshwari, *Finding non-separating induced cycles and independent spanning trees in* 4*-connected graphs*, J. Algorithms, 9 (1988), pp. 507–537.

[4] A. Huck, *Independent trees in graphs*, Graphs Combin., 10 (1994), pp. 29–45.

[5] A. Itai and M. Rodeh, *The multi-tree approach to reliability in distributed networks,* in Proceedings of the 25th Annual IEEE Sympos. on Foundations of Computer Science, IEEE, Piscataway, NJ, 1984, pp. 137–147.

[6] M. Kriesell, *Induced paths in* 5*-connected graphs*, J. Graph Theory, 36 (2001), pp. 52–58.

[7] L. Lovász, *Problems*, in Recent Advances in Graph Theory, M. Fiedler, ed., Academia, Prague, 1975.

[8] K. Miura, S. Nakano, T. Nishizeki, and D. Takahashi, *A linear-time algorithm to find four independent spanning trees in four connected planar graphs,* Internat. J. Found. Comput. Sci., 10 (1999), pp. 195–210.

[9] P. D. Seymour, *Disjoint paths in graphs*, Discrete Math., 29 (1980), pp. 293–309.

[10] C. Thomassen, 2*-linked graphs*, European J. Combin., 1 (1980), pp. 371–378.

[11] C. Thomassen, *A theorem on paths in planar graphs*, J. Graph Theory, 7 (1983), pp. 169–176.

[12] W. T. Tutte, *How to draw a graph*, Proc. London Math. Soc. (3), 13 (1963), pp. 743–767.

[13] A. Zehavi and A. Itai, *Three three-paths*, J. Graph Theory, 13 (1989), pp. 175–188.

SIAM J. Discrete Math.
Vol. 16, No. 4, pp. 630–634

AN EXPLICIT CONSTRUCTION OF LOWER-DIAMETER CUBIC GRAPHS[*]

M. CAPALBO[†]

Abstract. The expected diameter of a cubic undirected N-vertex graph is no more than $\lg N + \lg\lg N + 10$. However, the problem of explicitly constructing an infinite family of N-vertex graphs with maximum degree 3 and diameter $\lg N + o(\lg N)$ is open; the best construction known for graphs on N vertices has diameter $1.47 \lg N$. Here we present an explicit construction of an infinite family $\mathcal{H}$ of cubic graphs Λ with diameter $1.413 \lg |V(\Lambda)| + O(1)$.

Key words. diameter, shuffle-exchange graph

AMS subject classification. 05C35

DOI. 10.1137/S0895480197330266

1. Introduction. Let $X = (V, E)$ be an undirected connected graph. For any $u, v \in V$, we define $d_X(u, v)$ to be the number of edges in the shortest path from u to v. The *diameter* $d(X)$ of X is defined to be the quantity

$$d(X) = \max_{u,v \in V} d_X(u, v).$$

It is well known (see [2]) that the expected diameter of a randomly constructed cubic graph on N vertices is no greater than $\lg N + \lg\lg N + 10$. However, the problem of explicitly constructing an infinite family of cubic graphs with N vertices and diameter $\lg N + o(\lg N)$ appears to be a very difficult open question. (See [1] and [4, p. 754].) The best construction known for such graphs of N vertices has diameter slightly greater than $1.47 \lg N$ and is presented in [2] and [3]. Low-diameter bounded-degree graphs are important in the design of networks that allow for fast broadcasting and routing. Explicit constructions of such networks are much preferred over random constructions; among other things, it is much easier to describe, store, and reproduce an explicit construction than a random construction. It also tends to be much easier to exploit a network which has a known, easily describable structure.

We progress toward a solution of the problem of explicitly constructing a low-diameter cubic graph by presenting an infinite family $\mathcal{H}$ of graphs such that the diameter of each $\Lambda \in \mathcal{H}$ is less than $1.414 \lg |V(\Lambda)| + O(1)$. Like the construction in [2], $\mathcal{H}$ is also constructed from the family of 3-ary shuffle-exchange graphs; however, we make more nontrivial modifications.

2. Construction of $\mathcal{H}$. Let N be an arbitrarily large positive integer of the form $N = 3^{9r/10}3^{-10}$, where r is a positive integer that is a multiple of 10. We construct $\mathcal{H}$ by presenting the construction of a graph $\Lambda \in \mathcal{H}$ such that Λ has at least N vertices. Then we prove that Λ has maximum degree 3 and that the diameter of Λ is no greater than $1.413 \lg N$, where $\lg N$ denotes $\log_2 N$.

We now present the notation that we will use for the rest of this paper. Let $\{0, 1, 2\}^r$ and W denote the linear space of vectors v of length r such that each

[*] Received by the editors November 17, 1997; accepted for publication (in revised form) September 18, 2002; published electronically August 19, 2003.

http://www.siam.org/journals/sidma/16-4/33026.html

[†] Department of Mathematical Sciences, The Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21218 (mrc@ias.edu).

coordinate of v is in $\{0,1,2\}$, with componentwise addition and subtraction done mod 3. Also, for each $v \in W$, let v_i denote the ith component of v, where $i \in \{0,\ldots,r-1\}$, so $v = v_{r-1}v_{r-2}\ldots v_1v_0$. Next, let X be the subspace of W consisting of the vectors x such that $x_{10n+l'} = x_{10n+l}$ for all nonnegative integers n, l, and l' such that $n < r/10$ and $l, l' < 10$. Also, let C be the subspace of W consisting of the vectors c such that $c_l = c_{10n+l}$ for all nonnegative integers n and l such that $n < r/10$ and $l < 10$. So $|X| = 3^{r/10}$ and $|C| = 3^{10}$. Then let $C \oplus X$ be the linear space generated by X and C. Thus $|C \oplus X| = 3^{10}3^{r/10}$. Next let $\sim$ be the equivalence relation on W, where $u \sim v$ if and only if $u - v \in C \oplus X$. For each $v \in W$, let $[v]$ be the equivalence class of W with respect to $\sim$ that contains v, and let $[W]$ denote the set of equivalence classes of W with respect to $\sim$. So $|[W]| = |W|/|C \oplus X|$, which is $3^{-10}3^{9r/10} = N$. Furthermore, for each $k \in \{0,1,\ldots,r-1\}$, let $\mathbf{e}^k$ denote the vector in W such that $\mathbf{e}^k_k = 1$, but every other coordinate of $\mathbf{e}^k$ is 0, and for each $\iota \in \{0,1,2\}$, let $\iota\mathbf{e}^k$ denote the vector in W such that $\iota\mathbf{e}^k_k = \iota$, but every other coordinate of $\iota\mathbf{e}^k$ is 0. Finally, for each $u \in W$, let $\lambda_{10}(u)$ denote the vector $u_9u_8\ldots u_1u_0u_{r-1}u_{r-2}\ldots u_{11}u_{10}$.

Construction of Λ. We construct Λ from a graph Λ' on $10|[W]|$ vertices, which we next specify. The vertex-set of Λ' is $[W] \times \{0,1,\ldots,9\}$. For any (not necessarily distinct) $u\ u' \in W$, vertices $\nu = \langle [u], k\rangle$ and $\nu' = \langle [u'], k'\rangle$ are adjacent in Λ' if and only if ν and ν' satisfy either condition (I) or (II), stated next.

(I) Both $k = k'$, and also $[u'] \in [u+\mathbf{e}^k], [u+2\mathbf{e}^k]$.

(II) ν and ν' satisfy one of (A), (B), (C), stated next.

(A) $[u] = [u']$, and k is either $k'+1$ or $k'-1$.

(B) $[u'] = [\lambda_{10}(u)]$, and $k = 9$ and $k' = 0$.

(C) $[u] = [\lambda_{10}(u')]$, and $k' = 9$ and $k = 0$.

Having specified Λ', we next construct Λ from Λ'. For each $u \in W$ and each $k \in \{0,1,\ldots,9\}$, the induced subgraph of Λ' on the set $I_{[u],k} = \{\langle [u],k\rangle, \langle [u+\mathbf{e}^k],k\rangle, \langle [u+2\mathbf{e}^k],k\rangle\}$ is a 3-cycle $T_{[u],k}$. For each $u \in W$ and each $k \in \{0,1,\ldots,9\}$, (1) remove each of the three edges of $T_{[u],k}$, and then (2) put an edge between $\nu_{I_{[u],k}}$ and each of the three vertices in $I_{[u],k}$. The resulting graph is Λ. $\square$

Clearly Λ' has $|[W]| = N$ vertices, and $|V(\Lambda')| \leq |V(\Lambda)|$. So Λ has at least N vertices, as claimed in the beginning of this section. So to prove that $\mathcal{H}$ is as claimed, it suffices to prove Theorem 2.1.

Theorem 2.1. *Λ has diameter no greater than* $1.413 \lg |V(\Lambda)|$ *and is* 3*-regular.*

We devote the rest of this paper to proving Theorem 2.1. We first show that Λ has maximum degree 3. The graph Λ' is 4-regular. Indeed, for each vertex $\nu \in \Lambda'$, there are exactly two vertices ν' such that ν and ν' satisfy (I) and exactly two other vertices ν' such that ν and ν' satisfy (II). Then Λ is 3-regular. Indeed, on one hand, every vertex ν in $V(\Lambda')$ is in exactly one $I_{[u],k}$, so from (1) and (2), the degree of ν in Λ is exactly one less than the degree of ν in Λ'. On the other hand, each $I_{[u],k}$ has exactly three vertices, so every vertex ν in $V(\Lambda) \setminus V(\Lambda')$ has degree exactly 3 by (2). We have proved the following lemma.

Lemma 2.1. *Λ is* 3*-regular.*

Having shown that Λ is 3-regular, we show in the next section that Λ has diameter no greater than $1.413 \lg |V(\Lambda)|$. From this Theorem 2.1 will follow.

3. Bound on the diameter of Λ. In this section, we bound the diameter of Λ. To do this, we first introduce notation. Set E'_1 to be the set of edges $\{\nu,\nu'\}$ of Λ', where the endpoints ν and ν' satisfy (I), and set E'_2 to be the set of edges $\{\nu,\nu'\}$ of Λ', where the endpoints ν and ν' satisfy (II). So E'_1 and E'_2 partition the edge-set of Λ'. We note the following.

(*) Suppose that ν and ν' are adjacent vertices in Λ'. If $\{\nu, \nu'\}$ is an edge in E'_1, then $d_\Lambda(\nu, \nu')$ is no greater than 2. But if ν and ν' are linked by an edge in E'_2, then ν and ν' are still adjacent in Λ or, equivalently, $d_\Lambda(\nu, \nu')$ is 1.

To see **(*)**, note from (1) and (2) that the only edges we removed from Λ' to construct Λ are those in the $T_{[u],k}$'s. But each such edge is in E'_1. So if ν and ν' were linked by an edge in E'_2, then ν and ν' still would be adjacent in Λ. On the other hand, if ν and ν' were linked by an edge in E'_1, then they would be in the same $I_{[u],k}$. So by (2), ν and ν' share a common neighbor in Λ and **(*)** follows.

Finally, for arbitrary vectors v and v' in W, let $h(v, v')$ denote the number of indices i such that $v_i \neq v'_i$.

To show that Λ has the low diameter claimed, we do the following. Let v and v' be arbitrary elements of W. We first show in Lemma 3.2 that there is a path in Λ' from $\nu = \langle [v], 0\rangle$ to $\nu' = \langle [v'], 0\rangle$ that uses at most r edges of E'_2 and $h(v, v')$ edges of E'_1. We next show in Lemma 3.4 (which uses Lemma 3.3) that there is some $x \in C \oplus X$ such that $h(v + x, v') \leq .507342r$. Then, because $[v + x] = [v]$ if $x \in C \oplus X$, Lemmas 3.4 and 3.2 will imply that there is a path in Λ' from ν to ν' that has no more than r edges of E'_2 and $.507342r$ edges of E'_1, and thus $r + .507342r$ edges total. But then this and **(*)** imply that (*) there is from ν to ν' a path in Λ having no more than $r + (2 \times .507342r)$ edges. However, for every vertex ν'' in Λ, there is a $v \in W$ such that ν'' is within distance 7 in Λ of the vertex $\langle [v], 0\rangle$. (Indeed, if ν'' is also in $V(\Lambda')$, then $\nu'' = \langle [v''], k\rangle$ for some $v'' \in W$ and $k \in \{0, 1, \ldots, 9\}$. So there is a path of exactly k edges (all of E'_2) from ν'' to $\langle [v''], 0\rangle$ and a path of exactly $10 - k$ edges from ν'' to $\langle [\lambda_{10}(v)], 0\rangle$. But if ν'' is not in $V(\Lambda')$, then from (2) ν'' is adjacent in Λ to a vertex in $V(\Lambda')$.) This and (*) imply that the diameter $d(\Lambda)$ of Λ is $r + (2 \times .507342r) + O(1)$, which is $1.413 \lg |[W]| + O(1)$, because $|[W]|$ is $3^{-10}3^{9r/10}$. As the number of vertices in Λ is at least $|[W]|$, the next lemma follows.

Lemma 3.1. *Theorem* 2.1 *will follow from Lemmas* 2.1, 3.2, *and* 3.4.

We next establish Lemma 3.2, and then Lemma 3.4, and then we are done.

Lemma 3.2. *Let ν and ν' be two vertices in Λ' of the form $\nu = \langle [v], 0\rangle$ and $\nu' = \langle [v'], 0\rangle$. Then there is a path P in Λ' from ν to ν' containing at most $r + h(v, v')$ edges that satisfies both of the following simultaneously:*

(A) *P contains at most $h(v, v')$ edges of E'_1.*

(B) *P contains at most r edges of E'_2.*

Proof. The path

$$\begin{aligned}\langle [v], 0\rangle &= \langle [v_{r-1} \ldots v_1 v_0], 0\rangle \to \langle [v_{r-1} \ldots v_1 v'_0], 0\rangle \to \langle [v_{r-1} \ldots v_1 v'_0], 1\rangle \\ &\to \langle [v_{r-1} \ldots v'_1 v'_0], 1\rangle \to \cdots \to \langle [v_{r-1} \ldots v_{10} v'_9 v'_8 \ldots v'_1 v'_0], 9\rangle \\ &\to \langle [v'_9 \ldots v'_1 v'_0 v_{r-1} \ldots v_{10}], 0\rangle \to \cdots \to \langle [v'_{r-11} \ldots v'_1 v'_0 v'_{r-1} \ldots v'_{r-10}], 9\rangle \\ &\to \langle [v'_{r-1} \ldots v'_1 v'_0], 0\rangle = \langle [v'], 0\rangle\end{aligned}$$

uses exactly r edges from E'_2, and $h(v, v')$ edges from E'_1, and $r + h(v, v')$ edges total. □

We next present Lemma 3.3, which we will use to prove Lemma 3.4.

Lemma 3.3. *For each $w, y \in \{0, 1, 2\}^{10}$, let $f(w, y)$ denote the number of indices i such that the ith coordinate w_i of w differs from the ith coordinate y_i of y. Then let $\hat{f}(w, y)$ denote the minimum of $f(w, y)$, $f(w + \mathbf{1}, y)$, $f(w + \mathbf{2}, y)$, where $\mathbf{1}$ and $\mathbf{2}$ denote the vectors in $\{0, 1, 2\}^{10}$ of all 1's and 2's, respectively (addition is componentwise and* *mod 3). Then for each $w \in \{0,1,2\}^{10}$,*

$$\sum_{y\in\{0,1,2\}^{10}} \hat{f}(w,y) \leq 3^{10} \times 5.07342. \tag{3.1}$$

Proof. For each $\iota \in \{0,1,2\}$, let j_ι denote the number of indices l such that $w_l - y_l = \iota$ (mod 3). Then $\hat{f}(w,y)$ can be no greater than $10 - \max\{j_0, j_1, j_2\}$. Thus j can be no greater than 6. So, with w fixed, $\hat{f}(w,y) = 5$ for exactly $3\binom{10}{5}$ $(2^5 - 2)$ $+ 3\binom{10}{5} = 23436$ of the y's. Also, $\hat{f}(w,y) = 4$ for exactly $3\binom{10}{6}2^4 = 10080$ of the y's. Futhermore, $\hat{f}(w,y) = 3$ for exactly $3\binom{10}{7}2^3 = 2880$ of the y's; $\hat{f}(w,y) = 2$ for exactly 540 of the y's; $\hat{f}(w,y) = 1$ and 0 for exactly 60 and 3 of the y's, respectively. Then $\hat{f}(w,y) = 6$ for the remaining 22050 y's. Summing up gives Lemma 3.3. □

LEMMA 3.4. *Let v and v' be two vectors in W. Then there exists an $x \in C \oplus X$ such that $h(v+x, v') \leq .507342r$.*

Proof. For each nonnegative integer $n < r/10$, let $v^{(n)}$ and $v'^{(n)}$ denote the following vectors in $\{0,1,2\}^{10}$. For each $i \in \{0,1,\ldots,9\}$, the ith coordinate of $v^{(n)}$ is v_{10n+i} (equivalently, the $(10n+i)$th coordinate of v) and the ith coordinate of $v'^{(n)}$ is v'_{10n+i}. Then (using the notation in Lemma 3.3), we claim that

$$\min_{x\in C\oplus X} h(v+x, v') \leq \min_{y\in\{0,1,2\}^{10}} \sum_{n=0}^{n<r/10} \hat{f}(v^{(n)}+y, v'^{(n)}). \tag{3.2}$$

Indeed, let $\mathbf{0}$, $\mathbf{1}$, $\mathbf{2}$ be the vectors in $\{0,1,2\}^{10}$ of all 0's, 1's, and 2's, respectively. Then let y' be a vector in $\{0,1,2\}^{10}$, and let $z'^{(0)}$,…,$z'^{(r/10-1)}$ be vectors in $\{\mathbf{0},\mathbf{1},\mathbf{2}\}$, such that

$$\sum_{n=0}^{n<r/10} f(v^{(n)}+y'+z'^{(n)}, v'^{(n)}) = \min_{y\in\{0,1,2\}^{10}} \sum_{n=0}^{n<r/10} \hat{f}(v^{(n)}+y, v'^{(n)}). \tag{3.3}$$

Such y' and $z'^{(n)}$ exist by definition of $\hat{f}$. Then let $\hat{x}$ denote the vector in C such that $\hat{x}_{10n+l}$ equals the lth coordinate of y' for all nonnegative integers n and l such that $n < r/10$ and $l < 10$. Also, let z' denote the vector in X such that z'_{10n+l} equals the lth coordinate of $z'^{(n)}$ for all nonnegative integers n and l such that $n < r/10$ and $l < 10$. (Since each $z'^{(n)}$ is either $\mathbf{0}$, $\mathbf{1}$, or $\mathbf{2}$, the vector z' is in X.) Then set $x = \hat{x} + z'$. Thus by definition of h, we see that $h(v+x, v')$ equals the quantity on the left-hand side of (3.3) which, by (3.3), equals the quantity on the right-hand side of (3.2). Therefore, (3.2) follows, since $x \in C \oplus X$.

If we show that the right-hand side of (3.2) is no greater than $.507342r$, then Lemma 3.4 follows, so we show this next. To this end, we now use Lemma 3.3. By Lemma 3.3,

$$\sum_{y\in\{0,1,2\}^{10}} \sum_{n=0}^{n<r/10} \hat{f}(v^{(n)}+y, v'^{(n)}) \leq 3^{10} \times 5.07342 \times r/10. \tag{3.4}$$

So there must exist some $y \in \{0,1,2\}^{10}$ such that

$$\sum_{n=0}^{n<r/10} \hat{f}(v^{(n)}+y, v'^{(n)}) \leq 5.07342 \times r/10, \tag{3.5}$$

and so the right-hand side of (3.2) is indeed no greater than $.507342r$, and thus Lemma 3.4 follows. □

Theorem 2.1 follows from Lemmas 2.1, 3.2, and 3.4 by Lemma 3.1. □

REFERENCES

[1] B. Bollobás and F. R. K. Chung, *The diameter of a cycle plus a random matching*, SIAM J. Discrete Math., 1 (1988), pp. 328–333.

[2] F. R. K. Chung, *Diameters of graphs: Old problems and new results*, in Eighteenth Southeastern Conference on Combinatorics, Graph Theory, and Computing, Congr. Numer. 60, Utilitas Math., Winnipeg, Manitoba, 1987, pp. 295–317.

[3] M. Jerrim and S. Skylum, *Families of Fixed Degree Graphs for Processor Interconnection*, Internal Report, Department of Computer Science, University of Edinburgh, Edinburgh, Scotland, 1983.

[4] F. T. Leighton, *Introduction to Parallel Algorithms and Architectures. Arrays, Trees, Hypercubes*, Morgan Kaufmann, San Mateo, CA, 1992.

SIAM J. Discrete Math.
Vol. 16, No. 4, pp. 635–650

GRAPH SUBCOLORINGS: COMPLEXITY AND ALGORITHMS[*]

JIŘÍ FIALA[†], KLAUS JANSEN[‡], VAN BANG LE[§], AND EIKE SEIDEL[‡]

Abstract. In a graph coloring, each color class induces a disjoint union of isolated vertices. A graph subcoloring generalizes this concept, since here each color class induces a disjoint union of complete graphs. Erdős and, independently, Albertson et al., proved that every graph of maximum degree at most 3 has a 2-subcoloring. We point out that this fact is best possible with respect to degree constraints by showing that the problem of recognizing 2-subcolorable graphs with maximum degree 4 is *NP*-complete, even when restricted to triangle-free planar graphs. Moreover, in general, for fixed k, recognizing k-subcolorable graphs is *NP*-complete on graphs with maximum degree at most k^2. In contrast, we show that, for arbitrary k, k-SUBCOLORABILITY can be decided in linear time on graphs with bounded treewidth and on graphs with bounded cliquewidth (including cographs as a specific case).

Key words. graph subcoloring, computational complexity, cograph, cliquewidth, treewidth

AMS subject classifications. 05C15, 05C85, 68R10

DOI. 10.1137/S0895480101395245

1. Introduction and results.

A *k-coloring* of a graph G is a partition of the vertices into k pairwise disjoint sets $V_1, \dots, V_k$ such that for every $i = 1, 2, \dots, k$, each color class V_i consists of isolated vertices; i.e., it forms a stable set. The smallest k for which the graph G has a k-coloring is called the *chromatic number* of G, denoted by $\chi(G)$. Graph coloring is well studied for both its theoretic and algorithmic aspects. It is well known that testing 3-COLORABILITY is *NP*-complete for triangle-free graphs with maximum degree 4 (see [24]) and for planar graphs with maximum degree 4 (see [19]). Testing 3-COLORABILITY is easy for graphs with maximum degree 3 (by the Brooks theorem) and for triangle-free planar graphs (by the Grötzsch theorem). 2-colorable graphs can be recognized in linear time.

Graph coloring has been generalized in several ways and by a number of authors; see [2] for a comprehensive survey. In this paper we address one of these generalized colorings. A partition $V_1, \dots, V_k$ of the vertex set of a graph G is called a *k-subcoloring* of G if each color class V_i induces in G a disjoint union of complete subgraphs (of various sizes). The *subchromatic number* $\chi_s(G)$ of G is the smallest integer k for which G has a k-subcoloring. Subcolorings have been discussed in [2, 7, 8, 25]. It turns out that subcolorings have many interesting properties similar to colorings, and every k-coloring is also a k-subcoloring; hence $\chi(G) \geq \chi_s(G)$. Among other results we would like to mention the following properties of graph subcolorings:

(1) For every $k \geq 1$, there is a triangle-free graph G_k with $\chi_s(G_k) = k$ [2, 25].

[*] Received by the editors September 17, 2001; accepted for publication (in revised form) February 26, 2003; published electronically August 19, 2003. This research was supported in part by EU ARACNE project HPRN-CT-1999-00112 and EU APPOL project IST-1999-14084. This research was done while the first author was in Kiel. An extended abstract of this paper was presented at the 27th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2001).

http://www.siam.org/journals/sidma/16-4/39524.html

[†] KAM, DIMATIA, and ITI, Charles University, Prague, Czech Republic (fiala@kam.mff.cuni.cz). The research of this author was supported by the Ministry of Education of the Czech Republic as project LN00A056.

[‡] Institut für Informatik und Praktische Mathematik, Christian-Albrechts-Universität zu Kiel, 24098 Kiel, Germany (kj@informatik.uni-kiel.de, ese@informatik.uni-kiel.de).

[§] Institut für Theoretische Informatik, Fachbereich Informatik, Universität Rostock, 18051 Rostock, Germany (le@informatik.uni-rostock.de).

(2) For every planar graph G, $\chi_s(G) \leq 4$. In addition, if G is outerplanar, $\chi_s(G) \leq 3$. These bounds are tight [7].

(3) For every graph G with maximum degree Δ, $\chi_s(G) \leq \lfloor \frac{\Delta}{2} \rfloor + 1$ [2].

Gimbel showed in [20] that the subchromatic number is dominated by the 1-defective coloring number. Cowen, Cowen, and Woodall showed, without using the 4-color theorem, that the 1-defective coloring of all planar graphs is at most 4 (see [15]). Therefore, the 4-subcolorability of planar graphs (2) is independent on the 4-color theorem as well.

By (3), every graph with maximum degree at most 3 is 2-subcolorable. Actually, this fact follows also from a theorem due to Erdős [16] which says that every graph G has a bipartite spanning subgraph H such that the degree in H of every vertex is at least one-half of its degree in G. Thus, if the maximum degree in G is at most 3, every bipartition of H defines a subbipartition of G. Moreover, such a bipartite spanning subgraph H of G can be found easily in polynomial time by a “local improvement” technique.

Albertson et al. [2] point out the difficulties involved in characterizing 2-subcolorable graphs by giving a number of examples. In this paper we prove the following theorems in sections 2.1 and 2.2.

THEOREM 1. *Recognizing* 2*-subcolorable graphs of maximum degree* 4 *is NP-complete, even on triangle-free planar graphs.*

Albertson informed us that, independently, Gimbel also proved Theorem 1 with a completely different reduction [20].

Formally we define k-SUBCOLORABILITY as a decision problem whose input is a graph G, and we question whether $\chi_s(G) \leq k$. Notice that the *NP*-completeness of k-SUBCOLORABILITY for the class of *all* graphs follows from a theorem by Achlioptas [1]. We prove the following theorem in section 2.4.

THEOREM 2. *For every fixed $k \geq 2$, k-*SUBCOLORABILITY *is NP-complete for graphs of maximum degree at most k^2.*

For constant k, k-SUBCOLORABILITY can be expressed as a monadic second order logic formula, and hence can be tested in linear time for graphs with bounded treewidth. Due to the fact that for these graphs $\chi_s(G) \leq c$ for a constant c, we get that there exists an algorithm that in linear time determines $\chi_s(G)$ for graphs with bounded treewidth. On the other hand, the general algorithm is unnecessarily complicated for our purpose, and in section 3.1 we present a simpler algorithm solving this problem.

On the other hand, $\chi_s(G)$ can be arbitrarily large for graphs with constant cliquewidth (or cographs that have cliquewidth ≤ 2).

THEOREM 3. *For every k, the k-*SUBCOLORABILITY *problem can be decided in time $O(n2^c k^{c+1})$ on graphs of treewidth bounded by a constant c, on cographs in time $O(nk^3)$, and finally in time $O(nk^{3^{2^c+1}})$ on graphs with cliquewidth bounded by c (assuming that a construction tree is given in all three cases).*

These algorithms can be fine tuned to compute the subchromatic number on these three graph classes. Note that our result on graphs with bounded cliquewidth is new because, for arbitrary k, k-SUBCOLORABILITY cannot be expressed in the so-called monadic second order logic.

2. The *NP*-completeness of the subcoloring problem.

We prove Theorem 1 by reducing the not-all-equal 3-satisfiability problem, which was proved to be *NP*-complete by Schaefer [27] (see also [18, Problem LO3]).

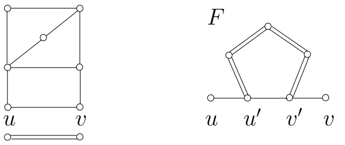


FIG. 1. *Two connector graphs.*

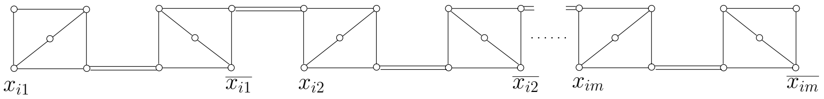


FIG. 2. *The variable gadget H_i.*

PROBLEM. *Let $\mathcal{C}$ be a Boolean formula consisting of m clauses such that every clause has exactly three distinct literals. Then the decision problem of whether there exists a satisfying assignment for $\mathcal{C}$ such that each clause in $\mathcal{C}$ has at least one false (and at least one true) literal is NP-complete.*

We denote the class of all formulas that allow such a not-all-equal assignment by *NAE3SAT*.

2.1. Triangle-free graphs of maximum degree 4. We first prove in this section the *NP*-completeness of 2-SUBCOLORABILITY for nonplanar triangle-free graphs of maximum degree 4 (the problem is clearly in *NP*). Let $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$ be a Boolean formula consisting of m clauses over variables $x_1, x_2, \dots, x_n$ such that every clause C_j of $\mathcal{C}$ contains exactly three literals, $C_j = (l'_j \vee l''_j \vee l'''_j)$. We will construct a triangle-free graph $G = G(\mathcal{C})$ of maximum degree 4 such that G has a 2-subcoloring if and only if $\mathcal{C} \in NAE3SAT$.

Before we describe the construction of G, observe the two connector graphs depicted in Figure 1.

The first graph has the property that, under any 2-subcoloring, vertices u and v have distinct colors. Its symbolic representation is shown below, and we will use this simplified drawing when a larger graph contains this connector as a subgraph. Such an example is depicted in the connector graph F (in the right part of Figure 1). Observe that under any 2-subcoloring of F the pair u, v is always colored by the same color, distinct from the color used on the pair u', v'.

The graph G consists of three parts: clause gadgets, variable gadgets, and connectors. The clause gadget is very simple: For each clause C_j, we insert into G a unique path P_3 of length 2, with vertices labeled l'_j, l''_j, and l'''_j. Observe that every clause gadget allows all possible 2-subcolorings such that both colors are used and that (by the definition of 2-subcoloring) it is impossible to color the P_3 by one color.

For each variable x_i we insert in G a copy of graph H_i, depicted in Figure 2.

LEMMA 4. *The graph H_i is 2-subcolorable. Any 2-subcoloring contains vertices $x_{i1}, x_{i2}, \dots, x_{im}$ in the same color class, and vertices $\overline{x_{i1}}, \overline{x_{i2}}, \dots, \overline{x_{im}}$ are colored by the other color.*

We complete the construction of the graph G by connecting clause and variable gadgets by inserting a copy of the connector graph F for each literal l^α_j of C, where the vertex u of F is merged with the corresponding vertex l^α_j in a clause gadget, and the vertex v is merged either with the vertex x_{ij} if the literal l^α_j equals x_i or with the vertex $\overline{x_{ij}}$ if $l^\alpha_j = \neg x_i$. (We make the construction for all possible $\alpha \in \{{\prime}, {\prime\prime}, {\prime\prime\prime}\}$.)

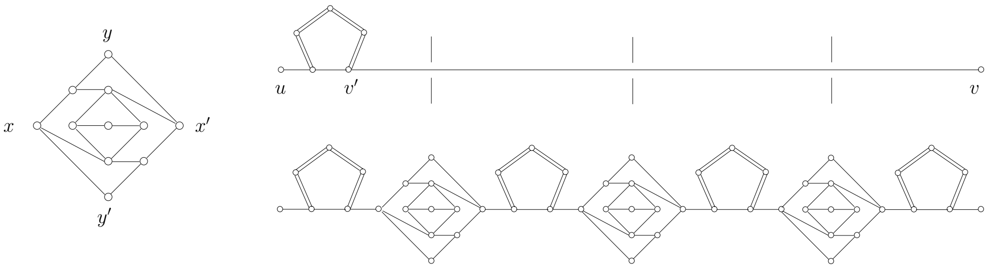


FIG. 3. *The "crossover" graph and the crossing replacement.*

Observe that all graphs involved in the construction of the graph G are triangle free and of maximum degree 4, and even the final composition does not violate this property.

We now show that $\mathcal{C} \in NAE3SAT$ if and only if G has a 2-subcoloring.

Let ϕ be a truth assignment for $\mathcal{C}$ in which every clause has at least one true and at least one false valued literal. We define a 2-subcoloring of G as follows: For every variable x_i, color all x_{ij} red if and only if $\phi(x_i) = \mathsf{true}$, and use the blue color otherwise. Then extend this subcoloring to a unique 2-subcoloring of H_i. This is possible, as we have seen by Lemma 4. Next extend this 2-subcoloring for all connectors F. Since ϕ was a feasible $NAE3SAT$ assignment, every clause gadget (path of length 3) is also properly 2-subcolored. Observe also that there is no conflict due to the vertex merging in the construction of G since, in every F, vertices u and u' and also v and v' have different colors.

In the opposite direction, suppose that there is a 2-subcoloring of G in red and blue. We define the assignment ϕ for $\mathcal{C}$ as follows: $\phi(x_i) = \mathsf{true}$ if x_{ij} is red for some j; otherwise $\phi(x_i) = \mathsf{false}$. By Lemma 4, this assignment is well defined. Due to the properties of connectors F it holds that in every clause gadget two of the three vertices l'_j, l''_j, l''_j have different colors. Therefore, each clause C_j has at least one true and at least one false literal by the truth assignment ϕ.

2.2. Planar graphs of maximum degree 4. In this section, we construct a triangle-free planar graph G' from the graph G obtained in the previous section, such that G is 2-subcolorable if and only if G' is 2-subcolorable.

Note that G can be embedded in the plane, in polynomial time, such that every edge is a straight line, all edge crossings occur only on (v, v') edges of the connector graphs F, and each crossing point meets exactly two edges. This makes possible the use of the "crossover" technique, described among others in [18], in proving *NP*-completeness of PLANAR GRAPH 3-COLORABILITY.

The "crossover" in our construction is the graph depicted in Figure 3 on the left side and has the following properties:

- In any 2-subcoloring, vertices x and x' belong to the same color class, and also vertices y and y' have the same color (not necessarily the same as x, x').
- There exists a 2-subcoloring such that x and y belong to the same color class, and also another 2-subcoloring such that x and y are colored by different colors.

The construction of the planar graph G' from G is very similar to the construction for PLANAR GRAPH 3-COLORABILITY. We replace each crossing point with the "crossover" graph and join these crossovers by connectors F (see Figure 3, right).

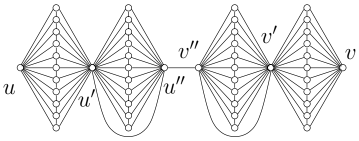


FIG. 4. *The edge replacement graph.*

Observe that the graph G' has a 2-subcoloring if and only if G does.

Suppose that G has a 2-subcoloring. Such a subcoloring of G can be extended to a 2-subcoloring for G' as follows: For every edge (v', v) in G use the color of v on vertices u and v of all connectors F added during removed crossovers. Such a coloring can be extended to the coloring of G'.

In the opposite direction, suppose that G' has a 2-subcoloring. Then, due to the properties of connector F (u and v are colored the same) and the "crossover" graph (opposite vertices maintain the same color), the color restriction on the original vertices is a proper 2-subcoloring of G.

2.3. 3-subcolorability of planar graphs. The 3-SUBCOLORABILITY of planar graphs is also *NP*-complete. We show a simple reduction from PLANAR GRAPH 3-COLORABILITY. Assume that G is a planar graph whose proper 3-coloring is questioned. We replace each edge (u, v) with a graph, depicted in Figure 4, composed of four copies of path P_{11} and six additional vertices (see [7]).

By a case study, it is easy to check that, under any 3-subcoloring of this replacement graph, vertices u, u', and u'' have the same color, distinct from the color used on v'', v', and v. Hence the result for the reduction is straightforward.

COROLLARY 5. *The planar* 3-SUBCOLORABILITY *is NP-complete on planar graphs.*

Recall that every planar graph is 4-subcolorable and that every outerplanar graph is 3-subcolorable.

2.4. The hardness of k- subcoloring for $k \geq 3$. In this section we show that, for each fixed $k \geq 2$, the k-SUBCOLORABILITY is an *NP*-complete problem on graphs with maximum degree at most k^2.

LEMMA 6. *If φ is a k-subcoloring of a graph G and H is an induced subgraph of G, then the restriction of φ on H is a k-subcoloring of H.*

Note that Lemma 6 does not hold for subgraphs in general.

LEMMA 7. *For every $k \geq 2$, the complete k-partite graph $K_{k,\ldots,k,k+1}$ consisting of $k-1$ (small) partitions with k vertices and one (big) partition of $k+1$ vertices has exactly one (up to permutation) k-subcoloring; this k-subcoloring is also its unique k-coloring. The graph $K_{k,\ldots,k,k+1}$ cannot be subcolored with less than k colors.*

The reduction from $(k-1)$-SUBCOLORABILITY to k-SUBCOLORABILITY goes as follows: Let G be the graph for which the existence of a $(k-1)$-subcoloring is questioned and let $V(G) = \{v_1, \ldots, v_n\}$. Then the graph G', the instance for k-SUBCOLORABILITY, is constructed as follows:

- Take n copies $H_1, \ldots, H_n$ of the $K_{k,\ldots,k,k+1}$;
- in each H_i, label four distinct vertices of the big class with $x_1^i, x_2^i, x_3^i, x_4^i$, and label one vertex in each of the $k-1$ small classes with y_j^i, $1 \leq j \leq k-1$;
- add edges $(x_1^i, y_j^{i-1}), (x_2^i, y_j^{i-1})$ for all $1 \leq j \leq k-1, 2 \leq i \leq n$;
- add edges $(x_3^i, v_i), (x_4^i, v_i)$ for all $1 \leq i \leq n$.

We claim by (1) and (2) below that G is $(k-1)$-subcolorable if and only if G' is k-subcolorable:

(1) Suppose that G can be subcolored with $k-1$ colors. Then subcolor, in each H_i, the $k-1$ small classes with these $k-1$ colors (each class gets one color), and take one new color for the big class. This is a k-subcoloring for G'.

(2) Consider a k-subcoloring of G'. Then, by Lemma 6, the restriction of this subcoloring on each H_i is a k-subcoloring. By Lemma 7, each H_i gets all k colors and each class in H_i is monochromatic. Moreover, the big classes of all H_i's have the same color.

We show that, for $1 \leq i < n$, x_1^i and x_1^{i+1} have the same color. Assume the contrary; then the color of x_1^{i+1} must occur in a small class of H_i; say y_1^i has this color. But then $x_1^{i+1}, x_2^{i+1}, y_1^i$ induce a monochromatic path P_3 in G', contradicting the definition of k-subcoloring.

No vertex in G can have the color occurring in the big classes of the H_i's. Therefore, the restriction of the k-subcoloring of G' on G is a $(k-1)$-subcoloring of G.

LEMMA 8. *For $k \geq 3$, $\Delta(G') = \max\{\Delta(G)+2, k^2\}$.*

Proof. Observe that $\Delta(G') = \max\{\Delta(G)+2, d_{G'}(x_1^2), d_{G'}(y_1^2)\}$. By the construction, $d_{G'}(x_1^2) = (k-1)k + (k-1) = k^2 - 1$ and also $d_{G'}(y_1^2) = (k-2)k + (k+1) + 2 = k^2 - k + 3$; hence, for $k \geq 3$, the lemma follows. □

Proof of Theorem 2. The case $k = 2$ is proven by Theorem 1. The statement for $k \geq 3$ follows from the construction and Lemma 8 and by noting that if G has maximum degree at most $(k-1)^2$, then the graph G' constructed from G as above has maximum degree at most k^2. □

3. Polynomially solvable cases—algorithms. In this section we show that the k-SUBCOLORABILITY problem allows a polynomial time algorithm on restricted classes of input graphs, in particular on graphs with bounded treewidth, cographs, and graphs with bounded cliquewidth.

For fixed k, k-SUBCOLORABILITY can be expressed in monadic second order logic, which is a language used to describe graph properties, using the following constructions: quantifications over vertices, edges, sets of edges, sets of vertices, membership tests, adjacency tests, and logic operations.

By the results of Courcelle it is known that each problem that can be stated in monadic second order logic can be solved in linear time on graphs with bounded treewidth [12] or graphs with bounded cliquewidth [13]. Unfortunately, in both cases the multiplicative constant grows very fast; essentially it is a tower of 2's whose height is the number of quantifier-alternations of the monadic second order logic formula. In our case the height is, at worst, linear in k.

We develop our algorithm on an underlying tree structure for a given graph, and with the use of dynamic programming we perform the feasibility test for all vertices of the tree. Here we would like to introduce notions that will be common for all three forthcoming subsections.

The tree is denoted by T, and its *nodes* (to distinguish them from *vertices* of G) are denoted by $X_1, \ldots, X_m$. The tree is rooted, and hence the parent-child relation $\succ$ is well defined. Moreover, each node has at most two descendants, and the size of T is always linear in the size of G. Each node X_i is of a certain type (this type is sometimes specified by its *label*) and corresponds to a subgraph of G, denoted by G_i.

For each node X_i we build a table $\mathtt{Tab}_i$ of constant or polynomial size whose entries describe necessary properties of a feasible k-subcoloring ϕ_i of the graph G_i. The situation in which the table $\mathtt{Tab}_m$ is nonempty for the root node X_m corresponds to the fact that a proper k-subcoloring of the entire graph G exists.

3.1. Graphs with bounded treewidth. The notion of treewidth was introduced by Robertson and Seymour in [26] via tree decompositions.

Let $G = (V, E)$ be a graph. The *tree decomposition* of G is a tree T whose nodes X_i are subsets of V. The following are satisfied:

1. For each edge $(u, v) \in E$ there exists a node $X_i \in V(T)$ such that $u, v \in X_i$.
2. For any $v \in V(G)$ the sets X_i containing v induce a nonempty connected subtree of T.

The width of a tree decomposition T is $\max_{X_i \in V(T)}\{|X_i|\}-1$ and the treewidth of G is the minimum width among all possible tree decompositions. We denote treewidth by $tw(G)$. If the treewidth of G is bounded by a constant c, a tree decomposition of width at most c of G can be constructed in linear time $O(|V| + |E|)$ (see [6]).

We present a decision algorithm that, for fixed k, tests whether the subchromatic number $\chi_s(G) \leq k$ for graphs G with bounded treewidth. For simplicity we restrict our attention to a nice tree decomposition.

A *nice tree decomposition* of G [22] is a tree decomposition such that T is a rooted binary tree and, for each $i \in V(T)$, at least one of the following cases applies:

- X_i is a leaf and $|X_i| = 1$; then X_i is a *leaf node*.
- X_i has one child j and $X_i = X_j \cup \{v\}$ for some $v \in V(G) \setminus X_j$; then we call X_i an *introduce node*.
- X_i has one child j and $X_i = X_j \setminus \{v\}$, where $v \in X_j$; then X_i is a *forget node*.
- X_i has two children j, j', and $X_i = X_j = X_{j'}$; then we call X_i a *join node*.

Any tree decomposition of width bounded by a constant can be transformed in linear time into a nice tree decomposition of the same width [22].

Denote by G_i the subgraph of G induced by vertices of X_i and by G'_i the subgraph of G induced by vertices of $\bigcup_{j \succ i} X_j$, where $j \succ i$ means that j is a descendant of i; i.e., i lies on the path from j to the root of T.

Let ϕ_i be a k-subcoloring of G_i; then the *color clique* of ϕ_i is any inclusion-maximal set $K \subseteq V(G_i) = X_i$ such that all vertices of K have the same color under ϕ_i and are mutually adjacent in G_i. In other words, a color clique is any clique that belongs to a color class of ϕ_i.

The entries of $\mathtt{Tab}_i$ consist of several pairs (ϕ_i, g_{ϕ_i}), where ϕ_i is a feasible k-subcoloring of G_i that might be extended to a subcoloring of G'_i, and g_{ϕ_i} is a function assigning to each color clique K of ϕ_i a Boolean variable, which helps us to properly define a new k-subcoloring in the inductive step. Note that a single ϕ_i may occur in some entry of $\mathtt{Tab}_i$ several times with different functions g_{ϕ_i}. However, as G has bounded treewidth, the number of all pairs (ϕ_i, g_{ϕ_i}) is bounded by a constant. The evaluation of $\mathtt{Tab}_i$ goes as follows:

1. If $X_i = \{v\}$ is a *leaf node*, then $\mathtt{Tab}_i$ contains all k possible k-subcolorings ϕ_i of $G_i = (\{v\}, \emptyset)$. For the only color clique $K = \{v\}$ and all ϕ_i set $g_{\phi_i}(K) = \mathsf{true}$.
2. Let X_i be a *forget node* with the child X_j. $\mathtt{Tab}_j$ already has been computed. Then let $\mathtt{Tab}_i$ contain all entries from $\mathtt{Tab}_j$, restricted to set X_i. Take $(\phi_j, g_{\phi_j}) \in \mathtt{Tab}_j$, and let ϕ_i be the restricted k-subcoloring. For the color clique K of ϕ_j containing the vertex $v = X_j \setminus X_i$ set $g_{\phi_i}(K \setminus \{v\}) = \mathsf{false}$ (if $K \setminus \{v\}$ is nonempty). For all other color cliques L of ϕ_i let $g\phi_i(L) = g\phi_j(L)$. Remove duplicated entries in $\mathtt{Tab}_i$, if any exist.
3. Let X_i be an *introduce node* with the child X_j, $v \in V(G)$ as the added vertex, and $\mathtt{Tab}_j$ already known. Then for every pair $(\phi_j, g_{\phi_j}) \in \mathtt{Tab}_j$ and every k-subcoloring ϕ_i of G_i, such that ϕ_i restricted onto X_j is equal to ϕ_j, find a

color clique K of ϕ_i containing v. If $K = \{v\}$ or $g_{\phi_j}(K \setminus \{v\}) = \texttt{true}$, then add into $\texttt{Tab}_i$ entry (ϕ_i, g_{ϕ_i}), where $g_{\phi_i}(K) = \texttt{true}$, and set $g_{\phi_i}(L) = g_{\phi_j}(L)$ for all other color cliques $L \neq K$ of ϕ_i.

4. Let X_i be a *join node* with children X_j and $X_{j'}$ and ϕ_i be a k-subcoloring of G_i. Then for all possible combinations of $(\phi_j, g_{\phi_j}) \in \texttt{Tab}_j$ and $(\phi_{j'}, g_{\phi_{j'}}) \in \texttt{Tab}_{j'}$ add the entry $(\phi_i, g_{\phi_j} \wedge g_{\phi_{j'}})$ into $\texttt{Tab}_i$ if and only if $\phi_i = \phi_j = \phi_{j'}$. Thus for each color clique K of ϕ_i the value of $g_{\phi_j}(K) \vee g_{\phi_{j'}}(K)$ is $\texttt{true}$. Again, if some entries are present more times, store only one.
5. Compute the values of $\texttt{Tab}_i$ for all nodes X_i in the tree, as described in steps 1–4. The graph G allows a k-subcoloring if and only if the table entry $\texttt{Tab}_m$ is nonempty for the root X_m.

To show that the algorithm is correct we further explain steps 2, 3, and 4.

In step 2 we remember that in the function g a certain color clique K has already lost a vertex v, and future extension of K by v' would cause the color class to contain an induced P_3, since the edge (v, v') does not belong to G. Therefore in step 3 we try extending only those color cliques which might be extended. The same argument is used in step 4, since it is impossible to identify two color cliques when both of them have already forgotten a vertex. Note that various functions g_{ϕ_i} for a single k-subcoloring ϕ_i may appear during steps 2 and 4.

This discussion concludes the proof of the first part of Theorem 3. For a graph G with tree decomposition of width bounded by a constant c the decision of k-SUBCOLORABILITY can be performed as fast as the evaluation of the table $(\texttt{Tab}_i)_{i \in V(T)}$, that is, in time $O(n2^c k^{c+1})$. This expression is linear in n.

Note that finding the minimum k such that G allows a k-subcoloring can be done in time $O(n^{c+2})$ by running at most n tests for all $k < n$.

3.2. Cographs.

Cographs, defined below, belong to the class of graphs with bounded cliquewidth. In this section we present an $O(nk^3)$ algorithm to decide k-SUBCOLORABILITY and propose an $O(n^4)$ algorithm to compute the subchromatic number of cographs (graphs of cliquewidth bounded by 2). In particular, k-SUBCOLORABILITY (k arbitrary) is efficiently solvable for cographs.

Cographs are inductively defined as follows [9]:

- Every single vertex graph is a cograph.
- If G_j and $G_{j'}$ are two cographs, then the disjoint union $G_j \cup G_{j'}$ is a cograph.
- Similarly the join graph $G_j + G_{j'}$ of two cographs is a cograph. The join graph $G_j + G_{j'}$ is obtained from the disjoint union $G_j \cup G_{j'}$ by adding all edges between vertices of G_j and $G_{j'}$.

With each cograph $G = (V, E)$ we associate a cotree T. Each leaf node X_i of the cotree T represents a vertex $v \in V$, and in this case $G_i = (v_i, \emptyset)$. Note that each vertex $v \in V$ is represented exactly once by a leaf node in T.

Internal nodes of T have either label $\cup$ or $+$. If a parent X_i of nodes X_j and X'_j carries label $\cup$, then $G_i = G_j \cup G_{j'}$, and similarly, if it is labeled by $+$, then $G_i = G_j + G_{j'}$.

There is a linear time algorithm for recognizing whether a given graph G is a cograph and, if so, for constructing a cotree T of G (see [10]).

Our algorithm for determining the subchromatic number of a cograph relies on the following notion. A subcoloring ϕ_i of a graph G_i is of *type* (α, β) if ϕ_i has α color classes, each of which induces a clique (called *small classes*), and β remaining classes (called *big classes*). If ϕ_i is of type (α, β), we also call ϕ_i an (α, β)*-subcoloring.*

We write $(\alpha, \beta) \preceq (\gamma, \delta)$ and say that (α, β) *minorizes* (γ, δ) if it simultaneously holds that $\beta \le \delta$ and $\alpha + \beta \le \gamma + \delta$. It is clear that from a subcoloring of type (α, β) any subcoloring of type (γ, δ) with $(\alpha, \beta) \preceq (\gamma, \delta)$ can be derived by adding extra colors or claiming some small color classes as big.

Consider an (α_i, β_i)-subcoloring ϕ_i of a graph G_i which arose by disjoint union or by the join of two graphs G_j and $G_{j'}$. In the following, for each $t = j, j'$ we denote by ϕ_t the restriction of ϕ_i on G_t and assume that ϕ_t is of type (α_t, β_t).

LEMMA 9. *If $G_i = G_j + G_{j'}$, then any (α_i, β_i)-subcoloring ϕ_i of G_i satisfies $\alpha_i \ge \max\{\alpha_j, \alpha_{j'}\}$ and $\beta_i = \beta_j + \beta_{j'}$.*

Proof. The first equality follows from the fact that any small class of ϕ_i may consist of at most two small color classes, one in ϕ_j and one in $\phi_{j'}$. The second equality expresses the fact that a big color class of ϕ_i is big in exactly one of ϕ_j or $\phi_{j'}$. □

LEMMA 10. *If ϕ_j and $\phi_{j'}$ are subcolorings of type (α_j, β_j) and $(\alpha_{j'}, \beta_{j'})$, respectively, then a $(\max\{\alpha_j, \alpha_{j'}\}, \beta_j + \beta_{j'})$-subcoloring of $G_i = G_j + G'_j$ can be obtained from ϕ_j and $\phi_{j'}$.*

Proof. A $(\max\{\alpha_j, \alpha_{j'}\}, \beta_j + \beta_{j'})$-subcoloring of G_i can be obtained by combining $\min\{\alpha_j, \alpha_{j'}\}$ small classes of G_j and $G_{j'}$ into the same color class of G_i and leaving the other color classes disjoint. □

LEMMA 11. *If $G_i = G_j \cup G_{j'}$, then $\beta_i \ge \max\{\beta_j, \beta_{j'}\}$, $\alpha_i + \beta_i \ge \alpha_t + \beta_t$ $(t = j, j')$, and $\alpha_i + 2\beta_i \ge \alpha_j + \beta_j + \alpha_{j'} + \beta_{j'}$.*

Proof. Let

- r denote the number of the big color classes C of ϕ_i such that, for each $t = j, j'$, $C \cap G_t$ is a big class in ϕ_t;
- r_t denote the number of big classes C of ϕ_i such that, for $t \ne t'$, $C \cap G_t$ is a small class in ϕ_t but $C \cap G_{t'}$ is a big class in $\phi_{t'}$;
- q_t denote the number of big classes of ϕ_i belonging only to ϕ_t;
- s denote the remaining big classes of ϕ_i, which are small in both G_j and $G_{j'}$;
- l_t denote the number of the small classes of ϕ_i belonging to ϕ_t.

The first statement of the lemma follows directly from

$$\begin{aligned} \alpha_i &= l_1 + l_2, & \beta_i &= r + r_1 + r_2 + q_1 + q_2 + s, \\ \alpha_j &= r_1 + l_1 + s, & \beta_j &= r + r_2 + q_1, \\ \alpha_{j'} &= r_2 + l_2 + s, & \beta_{j'} &= r + r_1 + q_2. \end{aligned}$$

Moreover, $\alpha_j + \beta_j = \alpha_i + \beta_i - l_2 - q_2$ and $\alpha_{j'} + \beta_{j'} = \alpha_i + \beta_i - l_1 - q_1$; hence the second statement holds. The third statement then follows from $\alpha_j + \beta_j + \alpha_{j'} + \beta_{j'} = 2(\alpha_i + \beta_i) - l_1 - l_2 - q_1 - q_2 = \alpha_i + 2\beta_i - q_1 - q_2$. □

In view of Lemmas 9 and 10, we are interested in (α_i, β_i)-subcolorings with a small number β_i of big classes. A way to obtain such a subcoloring of $G_i = G_j \cup G_{j'}$ from ϕ_j and $\phi_{j'}$ is as follows: We first merge $\min\{\beta_j, \beta_{j'}\}$ pairs of big classes of ϕ_j and $\phi_{j'}$ and then combine as many as possible of the $|\beta_j - \beta_{j'}|$ remaining big classes together with some small classes into a new color class of ϕ_i. The number of remaining small classes of ϕ_j is then $\kappa_j := \alpha_j - \min\{\alpha_j, \max\{0, \beta_{j'} - \beta_j\}\}$. Similarly, ϕ_2 contains $\kappa_{j'} := \alpha_{j'} - \min\{\alpha_{j'}, \max\{0, \beta_j - \beta_{j'}\}\}$ remaining small classes.

Note that $\kappa_j = \alpha_j$ (if $\beta_j \ge \beta_{j'}$) or $\kappa_{j'} = \alpha_{j'}$ otherwise. Finally, we combine κ, $0 \le \kappa \le \min\{\kappa_j, \kappa_{j'}\}$, small classes of ϕ_j with k small classes of $\phi_{j'}$ and get a $(\kappa_j + \kappa_{j'} - 2\kappa, \max\{\beta_j, \beta_{j'}\} + \kappa)$-subcoloring ϕ_i of $G_i = G_j \cup G_{j'}$.

This and Lemma 10 suggest the following algorithm for determining $\chi_s(G)$, assuming that the cotree T of a cograph G is given.

For each node X_i of T the algorithm stores in $\mathtt{Tab}_i$ the type (α_i, β_i) of all possible (α_i, β_i)-subcolorings ϕ_i of the graph G_i that are relevant for computing $\chi_s(G)$ as follows:

1. For each leaf node X_i of T, put $(1, 0)$ into $\mathtt{Tab}_i$.
2. If X_i has label $+$ and children $X_j, X_{j'}$, then for all combinations of entries $(\alpha_j, \beta_j) \in \mathtt{Tab}_j$ and $(\alpha_{j'}, \beta_{j'}) \in \mathtt{Tab}_{j'}$ put into $\mathtt{Tab}_i$ the entry (α_i, β_i), where $\alpha_i = \max\{\alpha_j, \alpha_{j'}\}$ and $\beta_i = \beta_j + \beta_{j'}$. Remove all minorized entries, if any exist.
3. If X_i has label $\cup$ and children $X_j, X_{j'}$, then for all combinations of entries $(\alpha_j, \beta_j) \in \mathtt{Tab}_j$ and $(\alpha_{j'}, \beta_{j'}) \in \mathtt{Tab}_{j'}$ perform the following computation:
   - 3.1. Set $\kappa_j := \alpha_j - \min\{\alpha_j, \max\{0, \beta_{j'} - \beta_j\}\}$ and $\kappa_{j'} := \alpha_{j'} - \min\{\alpha_{j'}, \max\{0, \beta_j - \beta_{j'}\}\}$.
   - 3.2. For each κ varying from 0 to $\min\{\kappa_j, \kappa_{j'}\}$ put into $\mathtt{Tab}_i$ the entry (α_i, β_i), where $\alpha_i = \kappa_j + \kappa_{j'} - 2\kappa$ and $\beta_i = \max\{\beta_j, \beta_{j'}\} + \kappa$.

   Remove all minorized entries, if any exist.
4. Return $\chi_s(G) = \min\{\alpha_m + \beta_m : (\alpha_m, \beta_m) \in \mathtt{Tab}_m\}$ for the root X_m of T.

Note that the number of all entries (α_i, β_i) stored in each $\mathtt{Tab}_i$ is bounded by k. Moreover, as discussed by Lemmas 9 and 10 and after Lemma 11, if $(\alpha_i, \beta_i) \in \mathtt{Tab}_i$, then there exists a subcoloring of G_i of type (α_i, β_i).

The following lemma shows the correctness of the algorithm.

LEMMA 12. *For every node X_i of T and every subcoloring ϕ_i of G_i of type (γ_i, δ_i), there exists a pair $(\alpha_i, \beta_i) \in \mathtt{Tab}_i$ such that $(\alpha_i, \beta_i) \preceq (\gamma_i, \delta_i)$.*

Proof. The proof is by induction on the level of X_i. The statement of the lemma is correct for leaves of the cotree. So, let X_i be an internal node of T, and let $X_j, X_{j'}$ be the two children of X_i. For $t = j, j'$ let ϕ_t denote the restriction of ϕ_i on G_t, and suppose that ϕ_t is of type (γ_t, δ_t). By induction there exists $(\alpha_t, \beta_t) \in \mathtt{Tab}_t$ such that

$$(\alpha_t, \beta_t) \preceq (\gamma_t, \delta_t). \tag{I}$$

We distinguish two cases.

Case 1. X_i is $+$ node.

Set $\alpha_i := \max\{\alpha_j, \alpha_{j'}\}$ and $\beta_i := \beta_j + \beta_{j'}$. Then, according to step 2 of the algorithm, some entry in $\mathtt{Tab}_i$ minorizes (α_i, β_i). We claim that $(\alpha_i, \beta_i) \preceq (\gamma_i, \delta_i)$: By the induction hypothesis (I) and Lemma 9, $\beta_i = \beta_j + \beta_{j'} \le \delta_j + \delta_{j'} = \delta_i$. To see the second condition in the definition of $\preceq$ we may assume without loss of generality that $\alpha_j \le \alpha_{j'}$. Then

$$\begin{aligned}\alpha_i + \beta_i &= \alpha_{j'} + \beta_j + \beta_{j'} \le \beta_j + \gamma_{j'} + \delta_{j'} \\ &\le \delta_j + \gamma_{j'} + \delta_{j'} = \gamma_{j'} + \delta_i \\ &= \max\{\gamma_j, \gamma_{j'}\} + \delta_i \le \gamma_i + \delta_i.\end{aligned}$$

Case 2. X_i has label $\cup$.

Let $\kappa_j, \kappa_{j'}$ be the integers computed from (α_j, β_j) and $(\alpha_{j'}, \beta_{j'})$ in step 3 of the algorithm. Note that by (I) and Lemma 11,

$$\max\{\beta_j, \beta_{j'}\} \le \max\{\delta_j, \delta_{j'}\} \le \delta_i,$$

and hence there exists some integer $\kappa \ge 0$ such that

$$\max\{\beta_j, \beta_{j'}\} + \kappa \le \delta_i \text{ and } \kappa \le \min\{\kappa_j, \kappa_{j'}\}.$$

Let κ be the maximum integer satisfying these properties. Note that by the maximality of κ,

$$\text{(II)} \qquad \text{either } \kappa = \min\{\kappa_j, \kappa_{j'}\} \text{ or } \kappa = \delta_i - \max\{\beta_j, \beta_{j'}\}.$$

Set $\alpha_i := \kappa_j + \kappa_{j'} - 2\kappa$ and $\beta_i := \max\{\beta_j, \beta_{j'}\} + \kappa$. Then according to step 3 of the algorithm, some entry in $\mathtt{Tab}_i$ minorizes (α_i, β_i). We claim that $(\alpha_i, \beta_i) \preceq (\gamma_i, \delta_i)$.

By the choices of κ and β_i, we have $\beta_i \leq \delta_i$. To see the second condition in the definition of $\preceq$, we may assume without loss of generality that $\beta_j \leq \beta_{j'}$. Then $\kappa_j = \alpha_j - \min\{\alpha_j, \beta_{j'} - \beta_j\}$ and $\kappa_{j'} = \alpha_{j'}$.

If $\kappa_j = 0$, then $\kappa \leq \min\{\kappa_j, \kappa_{j'}\} = 0$ and

$$\begin{aligned} \alpha_i + \beta_i &= \alpha_{j'} - 2\kappa + \beta_{j'} + \kappa \leq \alpha_{j'} + \beta_{j'} \\ &\leq \gamma_{j'} + \delta_{j'} \leq \gamma_i + \delta_i. \end{aligned}$$

If $\kappa_j = \alpha_j - (\beta_{j'} - \beta_j)$, then

$$\alpha_i + \beta_i = \alpha_j - (\beta_{j'} - \beta_j) + \alpha_{j'} - 2\kappa + \beta_{j'} + \kappa = \alpha_j + \beta_j + \alpha_{j'} - \kappa.$$

In this case, in consideration of (II) there are two possibilities for κ. If $\kappa = \delta_i - \max\{\beta_j, \beta_{j'}\} = \delta_i - \beta_{j'}$, we get

$$\begin{aligned} \alpha_i + \beta_i &= \alpha_j + \beta_j + \alpha_{j'} - (\delta_i - \beta_{j'}) \\ &\leq \gamma_j + \delta_j + \gamma_{j'} + \delta_{j'} - \delta_i \\ &\leq (\gamma_i + 2\delta_i) - \delta_i = \gamma_i + \delta_i, \end{aligned}$$

and if $\kappa = \min\{\kappa_j, \kappa_{j'}\}$, then

$$\begin{aligned} \alpha_i + \beta_i &= \alpha_j + \beta_j + \alpha_{j'} - \min\{\kappa_j, \kappa_{j'}\} \\ &= \alpha_j + \beta_j + \alpha_{j'} - \min\{\alpha_j - \beta_{j'} + \beta_j, \alpha_{j'}\} \\ &= \max\{\alpha_{j'} + \beta_{j'}, \alpha_j + \beta_j\} \\ &\leq \max\{\gamma_{j'} + \delta_{j'}, \gamma_j + \delta_j\} \leq \gamma_i + \delta_i. \end{aligned}$$

Thus, in any case, $\alpha_i + \beta_i \leq \gamma_i + \delta_i$. Hence $(\alpha_i, \beta_i) \preceq (\gamma_i, \delta_i)$, and the lemma is proved. □

Lemma 12 implies that the k-subcolorability can be decided for cographs in time $O(nk^3)$ using types of subcolorings with $\alpha + \beta \leq k$. This finishes the second part of Theorem 3. In addition we can compute $\chi_s(G)$ in time $O(n^4)$, since $\chi_s(G) \leq n$.

3.3. Graphs with bounded cliquewidth. Graphs of bounded cliquewidth generalize both the notion of cographs and graphs with bounded treewidth [14, 11].

We have already mentioned that all graph problems that are expressible in monadic second order logic can be solved in linear time on graphs with bounded cliquewidth [12, 13], given an expression defining the input graph. There are many problems not expressible in monadic second order logic (e.g., HAMILTONICITY, k-COLORABILITY with k arbitrary, etc.) but, nevertheless, are solvable in polynomial time on graphs with bounded cliquewidth [17, 23].

In this section, we extend this list by showing that k-SUBCOLORABILITY with k being a part of the instance can be solved in polynomial time on graphs with bounded cliquewidth. Our approach is different from that of [17] and [23]. SUBCOLORABILITY is, however, much more complicated than COLORABILITY (as one may see in the case of cographs) and it is not clear whether the schemes suggested in [17, 23] can be modified for our problem.

Let us now recall the notion of cliquewidth. Consider a construction tree T over a finite label set L, which recursively defines a graph G as follows:

- Every *leaf node* X_i with operation $t(v)$, which means creation of a one-vertex graph $G_i = (\{v\}, \emptyset)$ where v is labeled by $t \in L$.
- The *join node* X_i with operation $\eta_{s,t}$ and one child X_j inserts into the graph G_j all edges between vertices labeled s and t ($s, t \in L$). We require that labels s and t be distinct; however, some edges between vertices labeled s and t may already exist.
- The *relabel node* X_i with operation $\rho_{s \to t}$ ($s, t \in L$) and one child X_j changes all labels s in the graph G_j to t.
- Finally, the *union node* X_i with two children X_j and $X_{j'}$ corresponds to the graph $G_i = G_j \cup G_{j'}$, where all vertices maintain their labels from subgraph G_j and $G_{j'}$, respectively.

The *cliquewidth* of a graph G is the smallest cardinality of the label set L such that

(1) there exists a construction tree T that uses label set L, and

(2) G is isomorphic to the graph G_m corresponding to the root X_m of the tree T.

It is shown in [14] that every construction tree can be rearranged in polynomial time such that

- for each join operation $\eta_{s,t}$ we may assume that in this moment there are no edges between vertices labeled s and t.
- for each node X_i it is possible to compute in polynomial time a graph G'_i, the subgraph of G_m induced by $V(G_i)$ (i.e., it contains G_i and all edges that will be added later due to join operations on the path from X_i to the root X_m).
- for each X_i it is possible to compute also in polynomial time an auxiliary graph F_i, defined on the label set used on G_i, where two labels s and t are connected in F_i if on the path from X_i to the root there is a sequence of operations ρ changing label s to s' and t to t' (possibly in several iterations) followed by $\eta_{s',t'}$. In other words, edge (s, t) in F_i means that later there should be added an edge between vertices which in G_i have labels s and t.

Now assume that the size of the set L is a fixed constant c. Consider an arbitrary set of labels $K \subseteq L$ and define a (possibly empty) subgraph G_i^K of G'_i induced by all vertices of G_i whose label belongs to K.

Assume that ϕ_i is a k-subcoloring of a graph G'_i and V_a is its color class. The type of the color class V_a is a vector τ of length 2^c, where entries are indexed by sets $K \subseteq L$, and

$$\tau_K(V_a) = \begin{cases} 0 & \text{if } V_a \text{ induces an empty graph in } G_i^K, \\ 1 & \text{if } V_a \text{ induces in } G_i^K \text{ a single clique with at least one vertex,} \\ 2 & \text{if } V_a \text{ induces in } G_i^K \text{ a disjoint union of nonempty cliques.} \end{cases}$$

Observe that $\tau_K(V_a) \leq \tau_{K'}(V_a)$ whenever $K \subseteq K'$. Moreover, there exist at most $M = 3^{2^c}$ different types of color classes. Let Γ be the set of all possible color class types.

The following definition will help us to control color class types in the time of relabeling operation $\rho_{s \to t}$.

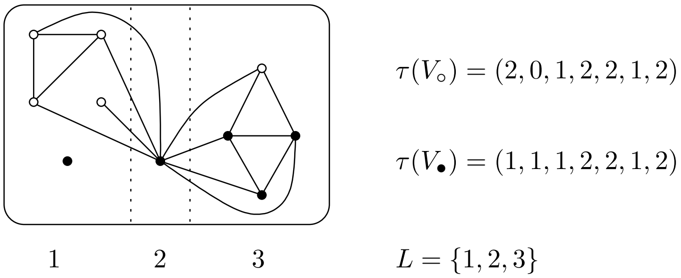

$\tau(V_\circ) = (2, 0, 1, 2, 2, 1, 2)$
$\tau(V_\bullet) = (1, 1, 1, 2, 2, 1, 2)$
1
2
3
$L = \{1, 2, 3\}$

FIG. 5. *Example of a* 2*-subcoloring of a graph of cliquewidth* 3.

We say that a color class of type ς transmutes into a color class of type τ via relabeling $s \to t$ if

- $\tau_K = \varsigma_K$ if $s, t \notin K$,
- $\tau_K = \varsigma_{K \cup \{s\}}$ if $t \in K, s \notin K$,
- $\tau_K = \tau_{K \setminus \{s\}}$ if $s \in K$.

Observe that for every type ς, the target type τ via transmutation $s \to t$ is unique. Note that such a test can be performed in constant time, as long as the length of the type is constant. To illustrate these notions, see a 2-subcoloring of a graph depicted in Figure 5. If we order subsets of L as $(1, 2, 3, 12, 13, 23, 123)$, then the types of the white and black classes are

$$\tau(V_\circ) = (2, 0, 1, 2, 2, 1, 2), \qquad \tau(V_\bullet) = (1, 1, 1, 2, 2, 1, 2).$$

Observe also that in this example the relabeling $\rho_{1 \to 3}$ transmutes the class $V_\bullet$ into a class of type $(0, 1, 2, 1, 2, 2, 2)$.

For a k-subcoloring ϕ_i we define its characteristic vector $\mathbf{a}$ indexed by color class types $\tau \in \Gamma$ whose entry $\mathbf{a}_\tau$ equals the number of color classes of ϕ_i that are of type τ.

The following lemma gives us a tool to test whether the characteristic vector $\mathbf{a}$ of a k-subcoloring ϕ_i of G'_i can be composed from the characteristic vectors $\mathbf{b}$ and $\mathbf{c}$ of a k-subcoloring of G'_j and $G'_{j'}$, respectively, during the union operation $G_i = G_j \cup G_{j'}$. In such a case we say that $\mathbf{a}$ is *compatible* with a composition of $\mathbf{b}$ and $\mathbf{c}$ in the label graph F_i.

Before stating the lemma we would like to discuss in more detail one particular case in the composition of types τ and ω into ς (with respect to F_i). Consider $K \subseteq L$ such that $\tau_K = \omega_K = 1$. In this case we have to decide whether $\varsigma_K = 1$ or 2. To get the right answer we first find sets $I, J \subseteq K$ such that $\tau_I = \omega_J = 1$ and $\tau_{K \setminus I} = \omega_{K \setminus J} = 0$. Only the following two situations make a composition of τ and ω possible:

- There is no edge in F_i between any $u \in I$ and $v \in J$. In this case we set $\varsigma_K = 2$.
- Sets I and J are disjoint and F_i contains all edges between vertices from I and from J. In this case we set $\varsigma_K = 1$.

If only one of these two cases applies, we say that τ_K and ω_K *can be merged* into ς_K. Observe also that for a single K we may perform such a test in time $O(|K|^2)$.

In Figure 6 we present two examples for $K = \{1, 2, 3\}$. In the first one we consider $I = \{1, 2\}$ and $J = \{1\}$. These two types can be merged if and only if F_i does not contain the edge $(1, 2)$, and the resulting type is $\varsigma_K = 2$. In the next example assume $I = \{1, 2\}$ and $J = \{3\}$. Then these types can be merged if and only if none or both of the two edges $(1, 3)$ and $(2, 3)$ are in F_i. The existence of dotted edges is not

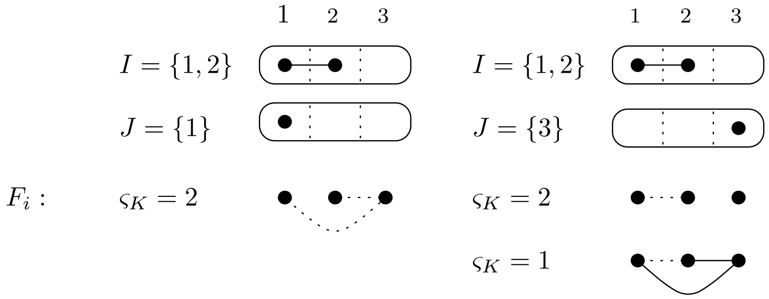


FIG. 6. *Merging of two color types $\tau_K = \omega_K = 1$.*

important here. In any other case of the composition of two types τ_K and ω_K, where at least one of them is not 1, we follow the majority principle, i.e., $\varsigma_K = \max\{\tau_K, \omega_K\}$.

LEMMA 13. *The type* $\mathbf{a}$ *is compatible with a composition of* $\mathbf{b}$ *and* $\mathbf{c}$ *in the label graph* F_i *if and only if the following system of linear inequalities over variables* $\mathbf{x}_{\varsigma,\tau,\omega}$ *has an integral solution:*

- $\mathbf{x}_{\varsigma,\tau,\omega} \geq 0$,
- $\mathbf{a}_\varsigma = \sum_{\tau,\omega\in\Gamma} \mathbf{x}_{\varsigma,\tau,\omega}$,
- $\mathbf{b}_\tau = \sum_{\varsigma,\omega\in\Gamma} \mathbf{x}_{\varsigma,\tau,\omega}$,
- $\mathbf{c}_\omega = \sum_{\varsigma\tau,\in\Gamma} \mathbf{x}_{\varsigma,\tau,\omega}$,
- $\mathbf{x}_{\varsigma,\tau,\omega} = 0$ *if there exists* $K \subseteq L$ *such that*
  - *either* $(\tau_K \neq 1 \vee \omega_K \neq 1)$ *and* $\varsigma_K \neq \max\{\tau_K, \omega_K\}$,
  - *or* $\tau_K = \omega_K = 1$ *and* τ_K, *and* ω_K *cannot be merged into* ς_K.

Since the dimension of this instance is bounded by a constant M^3, the corresponding integer linear program can be solved in time $O(M^9)$.

Now we are ready to present the decision algorithm. As in the previous cases we store in the table $\mathtt{Tab}_i$ all characteristic vectors of all proper k-subcolorings of the graph G'_i. Each entry in the characteristic vector of a k-subcoloring is bounded by k, so the number of records in $\mathtt{Tab}_i$ is bounded by k^M.

The recursive evaluation of $\mathtt{Tab}_i$ goes as follows:

1. For a leaf node X_i with $t(v)$ store in $\mathtt{Tab}_i$ the unique characteristic vector $\mathbf{a}$ of the k-subcoloring for which $\{v\} = V_1$ is its only color class and its type satisfies $\tau_K(V_1) = 1$ if $t \in K \subseteq L$, and $\tau_K(V_1) = 0$ otherwise. This operation requires $O(k^M)$ time.
2. All entries of a join node X_i with $\eta_{s,t}$ are taken from its only child X_j. Observe that in this case $G'_j = G_j$, including the vertex labeling, so no new restriction should be applied. We can use the same table for X_i as for X_j and this can be done in constant time.
3. For a recolor node X_i with label $\rho_{s\to t}$ and child X_j take every characteristic vector $\mathbf{b} \in \mathtt{Tab}_j$ and compute $\mathbf{a}_\tau$ as the sum of all $\mathbf{b}_\varsigma$, where the sum is taken over all types ς which transmute onto τ via relabeling $s \to t$. As shown above this can be done in time $O(k^M)$.
4. For the union node X_i with children $X_j, X_{j'}$ and for every possible type $\mathbf{a}$ perform the test whether or not it is compatible with some type $\mathbf{b} \in \mathtt{Tab}_j$ and some type $\mathbf{c} \in \mathtt{Tab}_{j'}$. If the test succeeds, put $\mathbf{a}$ into $\mathtt{Tab}_i$. By involving a linear program here the evaluation requires at most $O(k^{3M}M^9) = O(k^{3M})$ time units.

Remember that each step is followed by removing duplicates in table entries, if any appear.

Since the construction tree T has $O(cn)$ nodes and each evaluation of $\mathtt{Tab}_i$ can be done in time $O(k^{3^{2^c+1}})$, the entire time complexity of the algorithm is $O(nk^{3^{2^c+1}})$ in the worst case.

Note that computing $\chi_s(G)$ also can be determined in polynomial time: As, for constant cliquewidth c, the number of types of a color class is 3^{2^c}, the evaluation of the table $\mathtt{Tab}_{i\in V(T)}$ can be performed in $O(n^{3^{2^c+1}+1})$ time. Thus, after n tests, we can compute $\chi_s(G)$ in $O(n^{3^{2^c+1}+2})$ time.

4. Conclusion. In this paper we discussed both positive and negative results on the computational complexity of the k-SUBCOLORABILITY problem.

We showed a full complexity classification on planar graphs; in the case of 2-SUBCOLORABILITY we have even refined this classification on the degree condition.

Similarly, we have shown that the general k-SUBCOLORABILITY problem is *NP*-complete on graphs of degree at most k^2. Here we would like to point out that in view of degree constraints the PLANAR GRAPH 3-SUBCOLORABILITY and k-SUBCOLORABILITY are not fully classified and we are convinced that they deserve further research.

To motivate this study we would like to mention that we expect there is a possibility of constructing uniquely k-subcolorable graphs of degree $2k$. Here, we would like to propose a generalization of the complete computational complexity characterization for the case $k = 2$, as stated in the following conjecture.

CONJECTURE. *For every fixed $k \geq 2$, k-*SUBCOLORABILITY *is NP-complete for graphs with maximum degree $2k$.*

If true, this conjecture is best possible because every graph with maximum degree at most $2k - 1$ is k-subcolorable (cf. section 1).

Finally, we would like to remark that in all considered cases (bounded treewidth, cographs, bounded cliquewidth), an optimal subcoloring can be constructed with the same running time.

Acknowledgments. The third author thanks Hoàng-Oanh Le for her interesting comments. He also thanks the second author for inviting him to visit his research group in Kiel.

REFERENCES

[1] D. ACHLIOPTAS, *The complexity of G-free graph colourability*, Discrete Math., 165/166 (1997), pp. 31–38.

[2] M. O. ALBERTSON, R. E. JAMISON, S. T. HEDETNIEMI, AND S. C. LOCKE, *The subchromatic number of a graph*, Discrete Math., 74 (1989), pp. 33–49.

[3] S. ARNBORG, *Efficient algorithms for combinatorial problems on graphs with bounded decomposability—A survey*, BIT, 25 (1985), pp. 2–23.

[4] H. L. BODLAENDER, *A tourist guide through treewidth*, Acta Cybernet., 11 (1993), pp. 1–23.

[5] H. L. BODLAENDER, *A partial k-arboretum of graphs with bounded treewidth*, Theoret. Comput. Sci., 209 (1998), pp. 1–45.

[6] H. L. BODLAENDER, *A linear time algorithm for finding tree-decompositions of small treewidth*, SIAM J. Comput., 25 (1996), pp. 1305–1317.

[7] I. BROERE AND C. M. MYNHARDT, *Generalized colorings of outerplanar and planar graphs*, in Proceedings of Graph Theory with Applications to Algorithms and Computer Science, (Kalamazoo, MI, 1984), Wiley, New York, 1985, pp. 151–161.

[8] J. L. BROWN AND D. G. CORNEIL, *On generalized graph colorings*, J. Graph Theory, 11 (1987), pp. 87–99.

[9] D. G. Corneil, H. Lerchs, and L. Stewart Burlingham, *Complement reducible graphs*, Discrete Appl. Math., 3 (1981), pp. 163–174.

[10] D. G. Corneil, Y. Perl, and L. K. Stewart, *A linear recognition algorithm for cographs*, SIAM J. Comput., 14 (1985), pp. 926–934.

[11] D. G. Corneil and U. Rotics, *On the relationship between clique-width and treewidth*, in Graph-Theoretic Concepts in Computer Science—WG 2001, Lecture Notes in Comput. Sci. 2204, Springer-Verlag, Berlin, 2001, pp. 78–90.

[12] B. Courcelle, *Graph rewriting: An algebraic and logical approach*, in Handbook of Theoretical Computer Science, Vol. B, Elsevier, Amsterdam, 1990, pp. 192–242.

[13] B. Courcelle, *The monadic second-order logic of graphs* I: *Recognizable sets of finite graphs*, Inform. and Comput., 85 (1991), pp. 12–75.

[14] B. Courcelle and S. Olariu, *Uper bounds to the clique-width of graphs*, Discrete Appl. Math., 101 (2000), pp. 77–114.

[15] L. J. Cowen, R. H. Cowen, and D. R. Woodall, *Defective colorings of graphs in surfaces: Partitions into subgraphs of bounded valency*, J. Graph Theory, 10 (1986), pp. 187–195.

[16] P. Erdős, *Bipartite subgraphs of graphs*, Mat. Lapok (N.S.), 18 (1967), pp. 283–288.

[17] W. Espelage, F. Gurski, and E. Wanke, *How to solve NP-hard graph problems on clique-width bounded graphs in polynomial time*, in Graph-Theoretic Concepts in Computer Science—WG 2001, Lecture Notes in Comput. Sci. 2204, Springer-Verlag, Berlin, 2001, pp. 117–128.

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

[19] M. R. Garey, D. S. Johnson, and L. Stockmayer, *Some simplified NP-complete graph problems*, Theoret. Comput. Sci., 1 (1976), pp. 237–267.

[20] J. Gimbel, *Various Remarks on the Subchromatic Number of a Graph*, Technical Report 493, KAM-DIMATIA Series, Charles University, Prague, Czech Republic, 2000.

[21] H. A. Jung, *On a class of posets and the corresponding comparability graphs*, J. Combin. Theory Ser. B, 24 (1978), pp. 125–133.

[22] T. Kloks, *Treewidth—Computations and Approximations*, Lecture Notes in Comput. Sci. 842, Springer-Verlag, Berlin, 1994.

[23] D. Kobler and U. Rotics, *Polynomial algorithms for partitioning problems on graphs with fixed clique-width*, in Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2001, pp. 468–476.

[24] F. Maffray and M. Preissman, *On the NP-completeness of k-colorability problem of triangle-free graphs*, Discrete Math., 162 (1996), pp. 313–317.

[25] C. M. Mynhardt and I. Broere, *Generalized colorings of graphs*, in Proceedings of Graph Theory with Applications to Algorithms and Computer Science, (Kalamazoo, MI, 1984), Wiley, New York, 1985, pp. 583–594.

[26] N. Robertson and P. D. Seymour, *Graph minors.* II. *Algorithmic aspects of tree-width*, J. Algorithms, 7 (1986), pp. 309–322.

[27] T. J. Schaefer, *The complexity of the satisfiability problem*, in Proceedings of the 10th Annual ACM Symposium on Theory of Computing, ACM, New York, 1978, pp. 216–226.

SIAM J. Discrete Math.
Vol. 16, No. 4, pp. 651–662

COLORING POWERS OF PLANAR GRAPHS*

GEIR AGNARSSON† AND MAGNÚS M. HALLDÓRSSON‡

Abstract. We give nontrivial bounds for the *inductiveness* or *degeneracy* of power graphs G^k of a planar graph G. This implies bounds for the chromatic number as well, since the inductiveness naturally relates to a greedy algorithm for vertex-coloring the given graph. The inductiveness moreover yields bounds for the *choosability* of the graph. We show that the inductiveness of a square of a planar graph G is at most $\lceil 9\Delta/5 \rceil$, for the maximum degree Δ sufficiently large, and that it is sharp. In general, we show for a fixed integer $k \geq 1$ the inductiveness, the chromatic number, and the choosability of G^k to be $O(\Delta^{\lfloor k/2 \rfloor})$, which is tight.

Key words. distance-2 coloring, radio coloring

AMS subject classifications. 05C15, 05C85

DOI. 10.1137/S0895480100367950

1. Introduction.

The kth power G^k of a graph G is defined on the same set of vertices as G and has an edge between any pair of vertices of distance at most k in G. The topic of this paper is the coloring of power graphs or, equivalently, coloring the underlying graphs so that vertices of distance at most k receive different colors. We focus on the planar case, which has long been the center of attention for graph coloring.

We upper-bound the chromatic number and the *choosability* (see Definition 2.10) by the *inductiveness* of the graph G, which we denote here by $\mathrm{ind}(G)$. This measure of G, also known as the *degeneracy*, the *coloring number*, and the *Szekeres–Wilf* number, is defined to be $\max_{H \subseteq G}\{\min_{v \in H}(d_H(v))\}$, where H runs through all induced subgraphs of G. Inductiveness leads to an ordering of the vertices, $\{v_1, \ldots, v_n\}$, such that the number $d^+(v_i) = |\{v_j \in N_G(v_i) : j > i\}|$ of preneighbors v_j's of any v_i, with $j < i$, is at most $\mathrm{ind}(G)$.

The problem of coloring squares of graphs has applications to frequency allocation. Transceivers in a radio network communicate using channels at given radio frequencies. Graph coloring formalizes this problem well when the constraint is that nearby pairs of transceivers cannot use the same channel due to interference. However, if two transceivers are using the same channel and both are adjacent to a third station, a clashing of signals is experienced at that third station. This can be avoided by additionally requiring all neighbors of a node to be assigned different colors, i.e., that vertices of distance at most 2 receive different colors. This is equivalent to coloring the square of the underlying network. Another application of this problem, from a completely different direction, is that of approximating certain Hessian matrices [13]. Observe that neighbors of a node in a graph form a clique in the square of the graph. Thus, the minimum number of colors needed to color any square graph is at least $\Delta + 1$, where $\Delta = \Delta(G)$ is the maximum degree of the original graph. As a result,

*Received by the editors February 19, 2000; accepted for publication (in revised form) March 12, 2003; published electronically September 17, 2003. An earlier version of this paper appeared in *Proceedings of the* 11*th Annual ACM-SIAM Symposium on Discrete Algorithms*, SIAM, Philadelphia, 2000, pp. 654–662.

http://www.siam.org/journals/sidma/16-4/36795.html

†Department of Mathematical Sciences, George Mason University, MS 3F2, 4400 University Drive, Fairfax, VA 22030 (geir@math.gmu.edu).

‡Department of Computer Science, University of Iceland, IS-107 Reykjavik, Iceland (mmh@hi.is).

the number of colors used by our algorithms on power graphs will necessarily be a function of Δ. We are particularly interested in the asymptotic behavior as Δ grows.

Coloring squares of graphs, in particular planar graphs, has been studied in the literature from two perspectives: in graph theory, focusing on bounding the number of colors needed, and in computer science, focusing on complexity and approximate algorithms. We attempt here to contribute to both of these perspectives. We first review graph-theoretic results on planar graphs in chronological order.

The first reference to appear on coloring squares of planar graphs was by Wegner [19], who gave bounds on the clique number of such graphs. In particular, he gave an instance for which the clique number is at least $\lfloor 3\Delta/2 \rfloor + 1$ (which is largest possible) and conjectured that this is an upper bound on the chromatic number. He conjectured that

$$\chi(G^2) \leq \begin{cases} \Delta + 5 & \text{if } 4 \leq \Delta \leq 7; \\ \lfloor 3\Delta/2 \rfloor & \text{if } \Delta \geq 8. \end{cases}$$

Some work has been done on the case $\Delta = 3$, as listed in [8, Problem 2.18]. Ramanathan and Lloyd [16, 12] showed that $\operatorname{ind}(G^2) \leq 9\Delta$, which is obtained by a minimum-degree greedy coloring algorithm. Krumke, Marathe, and Ravi [10] generalized the bound to other classes of graphs, obtaining that $\operatorname{ind}(G^2) \leq (2\operatorname{ind}(G) - 1)\Delta$.

Independent of the original version of this paper [1], there were at least two unrelated papers on bounding the chromatic number $\chi(G^2)$ of a square of a planar graph. van den Heuvel and McGuinness [6] showed that $\chi(G^2) \leq 2\Delta + 25$, using methods similar to those of the proof of the 4-color theorem. Also Jendrol' and Skupień [7] showed that $\chi(G^2) \leq 3\Delta + 9$, by bounding the inductiveness.

In the current paper, we show that for large values of Δ, squares of planar graphs are $\lceil 9\Delta/5 \rceil$-inductive, implying a $\lceil 9\Delta/5 \rceil$+1-coloring. We show that this is sharp for all large values of Δ by constructing graphs attaining this inductiveness. For larger powers of a planar graph G, we obtain that G^k is $O(\Delta^{\lfloor k/2 \rfloor})$-inductive for any $k \geq 1$. This gives an asymptotically tight algorithmic bound for the chromatic number of the power graph.

McCormick [13] showed that the problem of coloring the power of a graph is NP-complete, for any fixed power, and a later proof was given by Lin and Skiena [11]. McCormick gave a greedy algorithm with an $O(\sqrt{n})$-approximation for squares of general graphs. Heggernes and Telle [5] showed that determining if the square of a cubic graph can be colored with four colors or less is NP-complete, while determining if three colors suffice is easy.

Ramanathan and Lloyd [16, 12] showed the problem of coloring squares of planar graphs to be NP-complete. Their bound mentioned earlier gave an algorithm with a performance ratio of 9, which was the best result known previous to [1]. The result of Krumke, Marathe, and Ravi [10] yields in general a performance ratio of $2\operatorname{ind}(G) - 1$. They also gave a polynomial algorithm for graphs of both bounded treewidth and bounded degree and used that to give a 2-approximation for bounded-degree planar graphs. Sen and Huson [17] showed that coloring squares of unit-circle graphs is NP-complete, while a constant approximation algorithm was given in [18].

Zhou, Kanari, and Nishizeki [20] have, in independent work, given a polynomial algorithm for distance-d coloring partial-k trees, for any constants d and k. As indicated in section 4, this implies a 2-approximation for distance-d coloring planar graphs for any d. Their algorithm, however, has a large polynomial complexity.

Our contributions give several approximation results. Combining the bound for squares of large-degree planar graphs with previous results for bounded-degree graphs, we obtain a 2-approximation for coloring that holds for all values of Δ. By itself, our bound gives a 1.8 *asymptotic* approximate coloring, as the chromatic number of the square goes to infinity. For higher powers of planar graphs, we obtain the first constant factor approximation for coloring cubes of planar graphs. However, the real strength of the current bounds are in giving absolute bounds on the number of colors used by the algorithm, as opposed to relative approximations, and thus implicitly bounding the number of colors used by an optimal solution.

Note the fine distinction between coloring the power graph G^k and finding a *distance-k* coloring of G. The resulting coloring is naturally the same. However, in the latter case, the original graph is given. While it is easy to compute the power graph G^k from G, Motwani and Sudan [14] showed that it is NP-hard to compute the kth root G of a graph G^k. All of the algorithms presented in this paper work without knowledge of the underlying root graph.

The rest of the paper is organized as follows. We bound the inductiveness of squares of planar graphs in section 2 and general powers of planar graphs in section 3. We consider the implications of these bounds to approximate colorings of powers of planar graphs in section 4.

Notation. The degree of a vertex v within a graph G is denoted by $d_G(v)$, or simply by $d(v)$ when there is no danger of ambiguity. The maximum degree of G is denoted by $\Delta = \Delta(G)$. For a vertex v denote by $d_k(v)$ the degree of v in G^k. The distance between two vertices u and v in a graph is the number of edges on the shortest path from u to v and is denoted by $d_G(u, v)$. Let $G[W]$ denote the subgraph of G induced by vertex subset W. Let $N(v) = N_G(v)$ be the set of neighbors of v in G, and let $N[v] = N_G[v]$ be the closed neighborhood of v in G given by $N[v] = N(v) \cup \{v\}$. The common closed neighborhood of u and v in G, denoted $N[uv]$ or $N_G[uv]$, is given by $N[uv] = N[u] \cap N[v]$.

2. Squares of planar graphs.

We start with a look at the main technique we use to derive bounds on the inductiveness of a square graph (and more generally, power graphs). The argument that is used to show, e.g., that planar graphs are 5-inductive, is the following. Euler's formula states that in a planar graph G, $|E(G)| \leq 3|V(G)| - 2$ (see [4, p. 74]). Thus, G contains a vertex of degree at most 5. Place one such node first in the inductive ordering and remove it from the graph. Now the remaining graph is planar, so inductively we obtain a 5-inductive ordering.

The upper bound of 5 on the minimum degree of a planar graph also implies that squares of planar graphs are of minimum degree at most 5Δ. That would seem to imply a 5Δ-ordering of the square graph. However, when a vertex is deleted from the graph, its incident edges are deleted as well so that vertices originally distance 2 apart may become much further apart in the remaining graph. An example of this is shown in Figure 1. Namely, the problem is that an induced subgraph does not preserve the paths of length 2 between vertices within the subgraph. The upshot is that degrees in the remaining graph do not adequately characterize degrees in the remaining part of the square of the graph. Our solution is to replace the *deletion* of a vertex by the *contraction* of an incident edge.

The *contraction* of an edge uv in graph G is the operation of collapsing the vertices u and v into a new vertex, giving the simple graph G/uv defined by $V(G/uv) = V(G) \setminus \{v\}$ and $E(G) = \{ww' \in E(G) : w, w' \neq v\} \cup \{uw : vw \in E(G)\}$. Note that if G is planar, then G/uv is also planar. This is a property of various classes of graphs that are *closed under minor operations.* By the classic theorems of Kuratowski and Wagner (see [4, p. 85]), planar graphs are precisely those graphs for which repeated

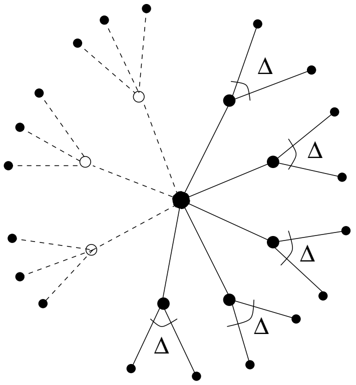


Fig. 1. *After the removal of nodes from a graph, a vertex can have vastly more of its original distance-2 neighbors remaining than its neighbors. After the deletion of the three white vertices, the center node has five neighbors but* $5\Delta + 9$ *of its remaining distance-2 neighbors.*

contractions do not yield supergraphs of K_5 or $K_{3,3}$. Minor-closedness holds for various other classes of graphs, e.g., partial-k trees, but not d-inductive graphs in general.

Since our bounds on the inductiveness are functions of Δ, it is imperative that the contraction operations do not increase the maximum degree.

Definition 2.1. *An edge* uv *is* mergeable *if* $|N[u] \cup N[v]| \leq \Delta + 2$.

The contraction of a mergeable uv in G yields a simple planar graph G/uv whose maximum degree stays at most Δ. Also, by the property of edge contractions, the new distance function is dominated by the one on G (i.e., distances in G/uv are at most those in G). Thus, to show that a square graph G^2 is $f(\Delta)$-inductive, we want to show the existence of a mergeable edge uv with $d_2(v) \leq f(\Delta)$. We state this as a general proposition.

Proposition 2.2. *Let* $\mathcal{G}$ *be a class of graphs closed under edge contractions, and let* f *be a nondecreasing function. Suppose every graph* G *in* $\mathcal{G}$ *contains a mergeable edge* uv *with* $d_2(v) \leq f(\Delta)$. *Then, the square of each* G *in* $\mathcal{G}$ *is* $f(\Delta)$*-inductive.*

2.1. Example applications of the contraction technique. We first illustrate the technique on simpler examples. Consider a minor-closed class of graphs that are 2-inductive (e.g., partial-2 trees or series-parallel graphs).

Theorem 2.3. *Squares of partial-*2 *trees are* 2Δ*-inductive.*

Proof. We inductively choose a vertex of degree at most 2 in the graph and contract one of its incident edges. In this case, either of its incident edges is mergeable, as the degree of each of its remaining neighbors does not increase. At most 2Δ vertices are within distance at most 2 of the selected vertex. Thus we obtain a 2Δ-inductive ordering of the square graph. □

Our second example yields a bound on the inductiveness of planar graphs of small degree that improves on the 9Δ-bound of [16] for 5-inductive graphs.

Theorem 2.4. *If* G *is a planar graph with* $\Delta(G) \geq 9$, *then* $\mathrm{ind}(G^2) \leq 4\Delta(G)+4$.

Proof. We consider a maximal supergraph G' of G and apply a theorem of Kotzig [9] (see also [7]). The theorem states that a maximal planar graph G' contains an edge uv such that $d_{G'}(u) + d_{G'}(v) \leq 13$ and, further, that $d_{G'}(u) + d_{G'}(v) \leq 11$ unless $d_{G'}(u) = 3$ or $d_{G'}(v) = 3$. We may assume $d_{G'}(u) \leq d_{G'}(v)$.

We claim that uv is mergeable when $\Delta \geq 9$ and that $d_2(u) \leq 4\Delta + 4$ (within

G). By Proposition 2.2, this yields the theorem. We show this by considering the following two cases. Observe first that since G' is maximal, u and v share two common neighbors a and b in G', and also that $N_G[w] \subseteq N_{G'}[w]$ for any node w.

Case when $d_{G'}(u) = 3$. In this case, $N_{G'}[u] = \{u, v, a, b\} \subseteq N_{G'}[v]$. Then, the union of the closed neighborhoods of u and v in G satisfies

$$N_G[u] \cup N_G[v] \subseteq N_{G'}[u] \cup N_{G'}[v] = N_{G'}[v].$$

Hence, $|N_G[u] \cup N_G[v]| \leq d_{G'}(v) + 1 \leq 11$. So, the edge uv is mergeable when $\Delta \geq 9$.

The number of distance-2 neighbors of u in G is at most the sum of the degrees of a, b, and v, not counting the possible edges from v to a and b, or is at most $2\Delta + 8$.

Case when $4 \leq d_{G'}(u) \leq 5$. Recall that the closed neighborhoods of u and v in G' share the four nodes a, b, u, and v. Thus,

$$|N_G[u] \cup N_G[v]| \leq |N_{G'}[u] \cup N_{G'}[v]| = |N_{G'}[u]| + |N_{G'}[v]| - 4 = d_{G'}(u) + d_{G'}(v) - 2 \leq 9.$$

Thus, uv is mergeable when $\Delta \geq 7$.

When counting the number of distance-2 neighbors of u in G, each of the neighbors of u other than v contributes at most Δ of them, while v contributes itself along with those of its neighbors not among $\{u, a, b\}$. Thus,

$$d_2(u) \leq (d(u) - 1)\Delta + [1 + (11 - d(u) - 3)] \leq 4\Delta + 4. \qquad \square$$

Jendrol' and Skupień [7] have recently given a refinement of Kotzig's result, obtaining a bound of $3\Delta + 8$ on the inductiveness of the square of a planar graph G with $\Delta(G) \geq 8$.

2.2. Sharp upper bound for large-degree graphs. We now turn to the main result of this section, which is that when G is planar and $\Delta = \Delta(G)$ is large enough, then G^2 is $\lceil 9\Delta/5 \rceil$-inductive. The following lemma is the key to this result.

LEMMA 2.5. *Let G be a simple planar graph of maximum degree $\Delta \geq 48$. Then there exists a mergeable edge vw in G with $d_2(v) \leq \max(\lceil 9\Delta/5 \rceil, \Delta + 600)$.*

Proof. We assume that we have a fixed planar embedding of G, i.e., that G is a plane graph. Let $V_h = \{v \in V(G) : d(v) \geq 26\}$ and $V_l = V(G) \setminus V_h$.

If there is a vertex $v \in V_l$ with at most one neighbor in V_h, then $d_2(v) \leq 1 \cdot \Delta + 24 \cdot 25 = \Delta + 600$. Select any incident edge vw to a low-degree neighbor w of v, and notice that the contracted edge would result in a node of degree at most $(25 - 1) + (25 - 1) = 48$. Since $\Delta \geq 48$, vw satisfies the claim of the lemma. Hence, for the rest of this proof, we assume the contrary, i.e., that every vertex in V_l has at least two neighbors in V_h.

Call a cycle of four vertices in G *forbidden* if exactly two opposite vertices of the cycle are in V_h and the enclosed region formed by the cycle in the plane properly contains at least one vertex in V_h.

If G contains a forbidden 4-cycle, then let G' be the subgraph of G induced by the region bounded by a minimal such 4-cycle. (Here, minimal means that no other 4-cycle is inside.) If G contains no such cycle, then let G' be G.

Consider now the multigraph H with vertex set $V(H) = V_h \cap V(G')$ and with colored edges defined as follows. For each edge uw in $E(G')$ with both $u, w \in V_h$, connect u and w with a red edge. For each vertex $v \in V_l$ adjacent to u and $w \in V_h$ in G' and to no other vertex in V_h, connect u and w in H with a green edge. Finally, for $v \in V_l$ adjacent to $u_1, u_2, \ldots, u_t \in V_h$ in G' in a clockwise order for $t \geq 3$, connect u_1

to u_2, u_2 to u_3,…,u_{t-1} to u_t, and u_t to u_1 with blue edges in H. A vertex in $V(G')$ is said to be *green* (*blue*) if the corresponding edge in H is.

Since G is planar, we note that H is also a planar multigraph. Hence, we can assume we have a drawing of H in the plane such that

- the vertices of H have the same configuration as in the plane graph G.
- for every pair $\{u, w\}$ of vertices of H connected by green or blue edges, their order with respect to u and w is the same as the order of the corresponding vertices of V_l.

By our assumption there is no vertex in V_l with at most one neighbor in V_h in G, and hence in G'. Therefore, the degree of a vertex in H is at least that in G'.

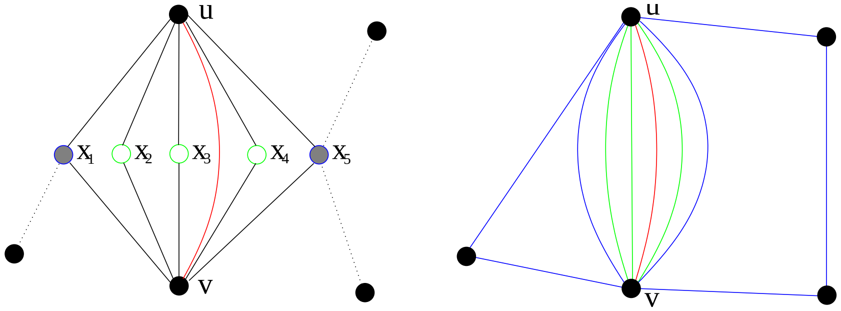


FIG. 2. *Example of a common neighborhood and the corresponding multigraph.*

For reference, we show in Figure 2 the common neighborhood in G of two vertices u and v, along with the the corresponding multigraph. Vertices in V_h are in black, blue vertices are grey, and green vertices are white. Here $N[uv]$ contains five nodes, in addition to u and v, corresponding to two blue and three green edges. Hence, in this figure we have in clockwise order w.r.t. the vertex v that x_1 is blue (grey in the figure) since it has three black neighbors, the vertices x_2, x_3, and x_4 are green (white in the figure) since each has two black neighbors u and v, and x_5 is blue (grey in the figure) since it has four black neighbors.

Let $v \in V(H)$ denote a vertex with at most five neighbors in H such that v is not on the 4-cycle defining G' (if G' was so defined). Euler's formula for planar graphs implies that there are at least three vertices of $V(H) = V_h \cap V(G')$ with at most five neighbors in H. Hence, there is such a vertex that is not on the 4-cycle defining G', as required. From now on, let v denote such a vertex.

CLAIM 2.6. *Let $x \in N_H(v)$. There are at most two vertices in $V_l \cap N_{G'}[vx]$ that have neighbors outside $N_{G'}[vx] \cup \{v, x\}$.*

Assume the contrary, i.e., that there are three vertices in $V_l \cap N_{G'}[vx]$ that have neighbors outside $N_{G'}[vx] \cup \{v, x\}$. Since G' is a plane graph, one of these three vertices, call it w, must be contained in the 4-cycle formed by v, x, and the other two vertices of those three. If w has a neighbor in $(V_h \cap V(G')) \setminus N_{G'}[vx]$, then we have a smaller forbidden 4-cycle, contradicting our assumption. If w has a neighbor in $(V_l \cap V(G')) \setminus N_{G'}[vx]$, then by our assumption, that neighbor must have at least two neighbors in $V_h \cap V(G')$ that cannot be the vertices $\{v, x\}$. That would again yield a smaller forbidden 4-cycle, a contradiction. Hence, we have the claim.

From now on, let u be the node in $V(H)$ with the largest neighborhood $N_{G'}[uv]$ in common with v in G'. When breaking ties, we prefer nodes that are not adjacent to v with a red edge.

CLAIM 2.7. *There is a vertex $w \in N_{G'}[uv]$ such that vw is mergeable and $N_G[N_G[w]] \subseteq N_G[v] \cup N_G[v]$.*

Observe that the selection criteria for u also serve to maximize the multiplicity m_{uv} of edges uv in H. Since $d_G(v) \geq 26$ and $d_H(v) \leq 5$, we have that $m_{uv} \geq \lceil 26/5 \rceil = 6$. Among these at least six edges, there is at most one red edge, and (by Claim 2.6) at most two edges (blue or green) that correspond to vertices of $V_l \cap N_{G'}[uv]$ with neighbors outside $N_{G'}[uv] \cup \{u, v\}$. Let w', w, and w'' be nodes in V_l in this order that correspond to the first three of the remaining at least $m_{uv} - 3$ edges in a clockwise order viewed from v (i.e., the white nodes in Figure 2, from left to right). By the planarity of G', w must be properly enclosed in the cycle formed by $C = \{u, v, w', w''\}$. Hence, $N_G(w) = N_{G'}(w) \subseteq C$, and wv is mergeable. Further, since w' and w'' have no neighbors outside of $N[v] \cup N[u]$, all distance-2 neighbors of w are in $N[v] \cup N[u]$ as claimed.

To prove the lemma, it suffices to bound the distance-2 degree of either v or w. We split the argument into two cases, depending on whether there is a red edge incident to v in H.

Case I. There is no red edge incident to v. Then all of v's neighbors are in V_l. Recall that each of them must have at least two high-degree neighbors; thus each of them belongs to some $N_{G'}[vx]$ for some $x \in N_H(v)$. For each $x \in N_H(v)$, there are by Claim 2.6 at most two nodes in $N_{G'}[vx]$, excluding v and x, that have neighbors outside of $N_{G'}[vx]$. Since there are at most five nodes in $N_H(v)$, there are at most 10 neighbors of v that have neighbors outside of $N_{G'}[v] \cup N_H(v)$. Hence,

$$d_2(v) \leq \Delta + 10 \cdot 25 + 5 < \Delta + 600.$$

Case II. There is a red edge incident on v, say x_1v. Thus, $v \in N_{G'}[x_1v]$. Since each node in V_l is by assumption adjacent to at least two vertices in V_h, it holds that $\bigcup_{x \in N_H(v)} N_{G'}[xv] = N_{G'}[v]$. Then,

$$|N_G[uv]| = |N_{G'}[uv]| \geq |N_{G'}[v]|/|N_H(v)| \geq \lceil (d_{G'}(v) + 1)/5 \rceil.$$

Since $N_G[w] = N_{G'}[w] \subseteq N_{G'}[uv]$, and since $x \mapsto x - \lceil x/5 \rceil$ is an increasing function, we have

$$\begin{aligned} d_2(w) + 1 &= |N_G[u] \cup N_G[v]| \\ &\leq |N_G[u]| + |N_G[v]| - |N_G[uv]| \\ &\leq (\Delta + 1) + (d_{G'}(v) + 1) - \lceil (d_{G'}(v) + 1)/5 \rceil \\ &\leq 2(\Delta + 1) - \lceil (\Delta + 1)/5 \rceil \\ &= \lceil 9\Delta/5 \rceil + 1. \end{aligned}$$

Together, the two cases establish that for at least one of the nodes v, w, we have that the distance-2 degree is at most $\max(\lceil 9\Delta/5 \rceil, \Delta + 600)$. □

Our main result now follows from Lemma 2.5 and Proposition 2.2.

THEOREM 2.8. *If G is a planar graph with $\Delta = \Delta(G) \geq 750$, then G^2 is $\lceil 9\Delta/5 \rceil$-inductive.*

It turns out that $\lceil 9\Delta/5 \rceil$ is a *sharp* upper bound for the inductiveness for all values of $\Delta \geq 750$.

OBSERVATION 2.9. *For any $\Delta \geq 5$, there exists a planar graph G of maximum degree Δ such that G^2 is of minimum degree $\lceil 9\Delta/5 \rceil$.*

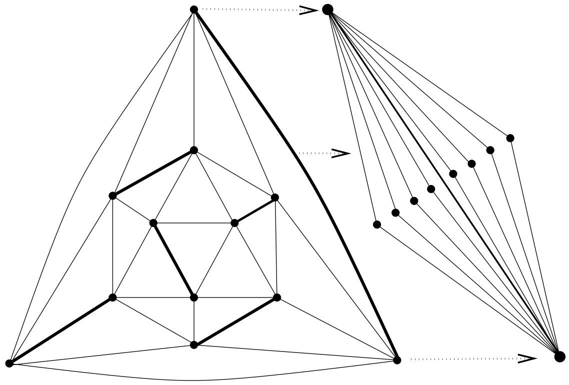


FIG. 3. *Icosahedron graph and split edges.*

Proof. Let $\Delta \geq 5$ and $q = \lfloor \Delta/5 \rfloor + 1$. Then $\Delta = 5q - i$, where $q \geq 2$ and $i \in \{1, 2, 3, 4, 5\}$. Let H be a five-regular planar icosahedron graph that can be partitioned into five perfect matchings (see Figure 3, where the edges of the first perfect matching are shown in bold). We construct from H a graph G as follows: To the first i perfect matchings we add $q - 2$ paths of length 2, and we replace the remaining $5 - i$ perfect matchings with q paths of length 2. Observe that there are two kinds of vertices in G; one kind has degree 2 and the other has degree Δ.

Consider a vertex w of degree 2 in G. If the neighbors of w of degree Δ are u and v, then there are precisely q vertices in $N[uv]$. Hence, the distance-2 degree of w is given by

$$\begin{aligned} d_2(w) + 1 &= |N[u]| + |N[v]| - |N[uv]| \\ &= 2(\Delta + 1) - (\lfloor \Delta/5 \rfloor + 1) \\ &= \lceil 9\Delta/5 \rceil + 1. \end{aligned}$$

However, a vertex v of degree Δ is connected to $i \geq 1$ other vertices of degree Δ. Call one of them u. Note that every vertex in $N[v] \cup N[u]$ is of distance 2 or less from v, and hence we have

$$d_2(v) + 1 \geq |N[v] \cup N[u]| = |N[u]| + |N[v]| - |N[uv]| = \lceil 9\Delta/5 \rceil + 1.$$

Therefore, the minimum degree of G^2 is precisely $\lceil 9\Delta/5 \rceil$, thereby completing our proof. □

Recall the following definition of choosability given in [4].

DEFINITION 2.10. *A graph G is k-*choosable *if for every collection $\{S_v : v \in V(G)\}$ of lists of colors, $S_v \subseteq \{1, 2, 3, \ldots\}$, where $|S_v| = k$ for every $v \in V(G)$, there is a color assignment*

$$c : V(G) \to \bigcup_{v \in V(G)} S_v$$

such that

- *$c(v) \in S_v$ for each $v \in V(G)$, and*
- *if $c(v) = c(u)$, then v and u are not neighbors in G.*

The minimum such k is called the choosability *of G and denoted by* $\mathrm{ch}(G)$*.*

We note that if a graph is k-choosable, then it is k-colorable. Also, by an easy induction, one can see that if a graph is k-inductive, then it is $(k+1)$-choosable. For any graph G we therefore have

$$\chi(G) \le \mathrm{ch}(G) \le \mathrm{ind}(G) + 1.$$

Hence, from Theorem 2.8 we have in particular the following corollary.

Corollary 2.11. *If G is a planar graph with $\Delta = \Delta(G) \ge 750$, then* $\mathrm{ch}(G^2) \le \lceil 9\Delta/5 \rceil + 1$*.*

3. General powers of planar graphs.

In this section we consider general powers G^k of planar graphs and establish tight asymptotic bounds on the inductiveness of $\mathrm{ind}(G^k)$. In fact we prove the following theorem, which in particular improves the bound of $\chi(G^k)$ given in [7], where it is shown that $\chi(G^k)$ is bounded from above by a polynomial in Δ of degree $k-1$.

Theorem 3.1. *Let G be a planar graph with maximum degree Δ. For any fixed $k \ge 1$, G^k is $O(\Delta^{\lfloor k/2 \rfloor})$-colorable. Also, there is a family of graphs that attains this bound. This bound is also asymptotically tight for the clique number, inductiveness, choosability, arboricity, and minimum degree of G^k.*

Let us first give a construction that matches the bound of the theorem. Given $k, \Delta \ge 1$, consider the tree T of height $\lfloor k/2 \rfloor$, where internal vertices have degree Δ. The number of vertices in T is

$$D_{\Delta,k} = 1 + \Delta + \Delta(\Delta-1) + \Delta(\Delta-1)^2 + \cdots + \Delta(\Delta-1)^{\lfloor k/2 \rfloor - 1} = \frac{\Delta(\Delta-1)^{\lfloor k/2 \rfloor} - 2}{\Delta - 2}.$$

Observe that T^k is a complete graph; thus $\chi(T^k) = D_{\Delta,k}$.

We now turn to proving the upper bound of the theorem. First we introduce some terminology.

Notation and arboricity. A *k-path* is a path of length exactly k. A *$(k, \le)$-path* is a path of length k or less. If u and v are vertices of a given graph, then a *walk* of length k from u to v is simply a sequence $(u_0, e_1, u_1, \ldots, u_{k-1}, e_k, u_k)$, where $u_0 = u$, $u_k = v$, and each e_i has end vertices u_{i-1} and u_i. Note that in a walk, both vertices and edges may be repeated.

Definition 3.2. *For a graph G, define its* arboricity*, denoted* $\mathrm{arb}(G)$*, as the minimum number of forests needed to cover all the edges of the graph G.*

Nash-Williams [15] proved that

$$\mathrm{arb}(G) = \max_{H \subseteq G} \left\lceil \frac{|E(H)|}{|V(H)| - 1} \right\rceil.$$

Arboricity is closely related to inductiveness.

Lemma 3.3. *For any graph G, we have* $\mathrm{arb}(G) \le \mathrm{ind}(G) \le 2\,\mathrm{arb}(G) - 1$*.*

Proof. Let q be $\mathrm{ind}(G)$. We first show that $E(G)$ can be partitioned into q forests. Given a linear arrangement of the vertices, such that each vertex v_i has at most q later neighbors, we arbitrarily color the edges from v_i to later vertices with at most q colors. In this way, each color class is acyclic, since two edges of the same color cannot have the same first-labeled endpoint, and thus is a forest. Therefore $\mathrm{arb}(G) \le q$, proving the first inequality.

For the second inequality, let $\mathrm{ind}(G) = q$. Let H be a subgraph of G such that $\min_v(d_H(v)) = q$. Since $2|E(H)| = \sum_{v \in V(H)} d_H(v) \ge q|V(H)|$, we have $\mathrm{arb}(G) >$

$|E(H)|/|V(H)| \geq q/2$. Since $\mathrm{arb}(G)$ is an integer, we have $q \leq 2\,\mathrm{arb}(G) - 1$, which completes our lemma. □

Note that if G is planar, we have that $\mathrm{arb}(G) \leq 3$ by Euler's formula and the Nash-Williams theorem. Also we have that $\mathrm{ind}(G) \leq 5$. Since there are planar graphs obtaining these values, the upper bound of Lemma 3.3 is tight for planar graphs.

From Theorem 2.4 and Lemma 3.3 we have in particular that $\mathrm{arb}(G^2) \leq 4\Delta + 4$ if $\Delta \geq 9$.

Arboricity of power graphs. We now want to find an upper bound of the arboricity of G^k in terms of Δ, where G is a planar graph. For a vertex set $U \subseteq V(G)$, let $E^k(U)$ be the edge set of the subgraph of G^k induced by U. Then, the arboricity of G^k is

$$\mathrm{arb}(G^k) = \max_{U \subseteq V(G)} \left\lceil \frac{|E^k(U)|}{|U| - 1} \right\rceil. \tag{3.1}$$

We will use this to bound $\mathrm{arb}(G^k)$, but first we note the following.

Lemma 3.4. *If G is a simple graph with* $\mathrm{arb}(G) = \alpha$, *then the edges of G can be directed in such a way that for each vertex $v \in V(G)$, at most α directed edges are pointing from v.*

Proof. Let $F_1, \dots, F_\alpha$ be the forests that cover the edges of G. For each subtree T of each F_i, direct its edges upward towards an arbitrarily chosen root r of T. In this way each F_i becomes a directed forest F_i^d in which every vertex, except the root, has outdegree 1, and the root has outdegree 0. Hence, as G is the disjoint union of the forests F_i, the outdegree of each vertex in G is at most α. □

Let G be a planar graph and let $U \subseteq V(G)$. Note that if two vertices of U are connected in G^k, then there is a $(k, \leq)$-path in G between them, and hence an i-walk between them, where $i \in \{k-1, k\}$.

Theorem 3.5. *For any graph G, we have* $\mathrm{arb}(G^k) \leq 2^{k+1}\alpha^{\lceil k/2 \rceil}\Delta^{\lfloor k/2 \rfloor}$, *where* $\alpha = \mathrm{arb}(G)$.

Remark. The main idea of the proof below, of counting the i-walks directly, is due to the anonymous referees.

Proof. By Lemma 3.4 we can direct the edges of G in such a way that for each vertex $v \in V(G)$ there are at most α directed edges pointing from v.

Let $U \subseteq V(G)$. If $uv \in E^k(U)$, then there is an i-walk in G, where $i \in \{k-1, k\}$, either from u to v, or from v to u, that walks against at most $\lfloor i/2 \rfloor$ of the given directions of the edges. Assume in this case there is such an i-walk $\vec{w}$ from u to v. There are $\sum_{j=0}^{\lfloor i/2 \rfloor} \binom{i}{j}$ possibilities of at most $\lfloor i/2 \rfloor$ edges in $\vec{w}$ pointing against the walk. Also, for each vertex on $\vec{w}$, there are at most α choices of directed edges pointing from the vertex, and at most $\Delta \geq \alpha$ choices of directed edges pointing to the vertex. Hence, the number of possible such i-walks $\vec{w}$ from u, with at most $\lfloor i/2 \rfloor$ edges pointing against the direction of the walk, is $\sum_{j=0}^{\lfloor i/2 \rfloor} \binom{i}{j} \alpha^{i-j}\Delta^j \leq (\sum_{j=0}^{\lfloor i/2 \rfloor} \binom{i}{j})\alpha^{\lceil k/2 \rceil}\Delta^{\lfloor k/2 \rfloor}$. Hence,

$$|E^k(U)| \leq 2 \left(\sum_{j=0}^{\lfloor (k-1)/2 \rfloor} \binom{k-1}{j} + \sum_{j=0}^{\lfloor k/2 \rfloor} \binom{k}{j} \right) \alpha^{\lceil k/2 \rceil}\Delta^{\lfloor k/2 \rfloor}|U| \leq 2^k \alpha^{\lceil k/2 \rceil}\Delta^{\lfloor k/2 \rfloor}|U|.$$

The theorem now follows from (3.1). □

Note that for a planar graph G we have $\mathrm{arb}(G) \leq 3$. Also note that for any set U of vertices in graph G, $2|E^k(U)| = \sum_{v \in U} d_{G[U]^k}(v)$, and hence, from the above proof, there is a vertex v with $d_{G[U]^k}(v) \leq 2^{k+1}\alpha^{\lceil k/2 \rceil}\Delta^{\lfloor k/2 \rfloor}$. With this in mind we have the following.

Corollary 3.6. *For a planar graph G with $\Delta \geq 3$ we have*

$$\operatorname{arb}(G^k), \operatorname{ind}(G^k) \leq 2^{k+1} 3^{\lceil k/2 \rceil} \Delta^{\lfloor k/2 \rfloor}.$$

By Lemma 3.3 and Theorem 3.5 we have that for any planar graph G, the chromatic number, clique number, choosability, and inductiveness are all at most $2\operatorname{arb}(G)$, which completes the proof of Theorem 3.1.

Remarks. Our original approach for proving Theorem 3.1, as found in our unrefereed report [1], was different. There, our argument was partly based on the following claimed "expansion property" for planar graphs: For a planar graph G and any subset $W \subseteq V(G)$ of vertices, there is a subset W' with $W \subseteq W' \subseteq V(G)$ and $|W'| \leq 10^{k-1}|W|$ such that if any two vertices in W are neighbors in G^k, then they are also neighbors in $G[W']^k$, the subgraphs of G^k induced by W'. Note that there are serious typos[1] in [1].

4. Approximation algorithms.

We can improve the best approximation factor known for coloring squares of planar graphs. Recall that since neighbors in G must be colored differently in G^2, $\chi(G^2) \geq \Delta + 1$. Thus, for $\Delta \geq 750$, Theorem 2.8 yields a 1.8-approximation. Hence, we obtain an *asymptotic* ratio of 1.8.

For constant values of Δ, we can use a result of Krumke, Marathe, and Ravi [10]. They stated a 3-approximation, but actually a 2-approximation easily follows from their approach, which is based on an often-used decomposition due to Baker [2]. The complexity of their approach is equivalent to the complexity of coloring a partial $O(\Delta)$-tree. Combining the results of [10] and [2] with our Theorem 2.8, we obtain a 2-approximation for any value of Δ.

Theorem 4.1. *The problem of coloring squares of planar graphs has a* 2*-approximation.*

Theorem 3.1 also immediately gives an $O(1)$-approximation to coloring cubes of planar graphs. However, better factors are possible.

Zhou, Kanari, and Nishizeki [20] independently gave a polynomial algorithm for distance-d coloring partial k-trees for any constant d and k. The complexity of their algorithm is $O(n(\alpha+1)^{2^{2(k+1)(d+2)+1}} + n^3)$, where $\alpha = O(\min(\Delta^{d/2}, n))$ is the number of colors needed. Since it is not indicated in [20], we show here how this result yields a 2-approximation for coloring G^d, for any constant d, when combined with the decomposition of Baker.

The technique of Baker [2] partitions the vertex set V of a planar graph into subsets $V_1, V_2, \ldots$, referred to as *layers*, such that all edges are between adjacent layers or within the same layer; i.e., if $u \in V_i$ and $uv \in E$, then $v \in V_{i-1} \cup V_i \cup V_{i+1}$. Now, let $V' = \cup_{i \bmod 2d < d} V_i$, $V'' = V - V'$, and G', G'' be the subgraphs induced by V' and V''. Observe that both G' and G'' consist of a collection of disjoint subgraphs U_i, corresponding to $V_{di} \cup V_{di+1} \cup \cdots \cup V_{d(i+1)-1}$. Further, notice that the subgraphs induced by the U_i will also be disjoint in G'^d and G''^d, since the distance between any pair of nodes in different subgraphs U_i is at least $d+1$. Thus, G'^d can be computed by considering each U_i separately. Now, G^d restricted to U_i is a subgraph of the graph H_i^d, where $H_i = G[\cup_{j=di-(d-1)}^{d(i+1)-(d-2)} U_i]$. H_i is a $(3d-2)$-outerplanar graph, which means

[1] In [1, p. 659], the displayed inequality of Lemma 3.2 and the last line of its proof should have "$E(G)$" instead of "$E(F)$" on the right of that inequality. Likewise, in the first displayed inequality in the right column on that same page "$E(F_i)$" should be "$E(G[W_U])$." Finally, since $G[W_U]$ is planar, a factor of "3" should be in front of the rest of the remaining expressions in that display. This affects the rest of that article, in particular α_k in Lemma 3.3 and Corollary 3.1.

that it is a partial $(9d-8)$-tree by a result of Bodlaender [3]. Hence, we can compute the optimal coloring of each H_i in time $O(n^{2^{(9d-7)(d+2)+1}+1})$. Thus, we can solve G'^2 and G''^2 exactly and, in total, using at most twice the optimal number of colors.

Acknowledgments. We thank Madhav Marathe for introducing this problem to us, and Noga Alon and Jan Kratochvíl for advice. Most of this work was done while both authors were at the Science Institute, University of Iceland. In addition, Geir would like to thank the Department of Mathematics at Arizona State University for their hospitality, and Magnús would like to thank the Department of Communications Systems at Kyoto University for their hospitality. Last but not least we would like to sincerely thank the anonymous referees for their patience, helpful criticism, and for repeatedly providing valuable comments which, in particular, helped shorten and simplify our original proof of Theorem 3.1.

REFERENCES

[1] G. Agnarsson and M. M. Halldórsson, *Coloring powers of planar graphs*, in Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 2000, pp. 654–662.

[2] B. S. Baker, *Approximation algorithms for NP-complete problems on planar graphs*, J. ACM, 41 (1994), pp. 153–180.

[3] H. L. Bodlaender, *A partial k-arboretum of graphs with bounded treewidth*, in Theoret. Comput. Sci., 209 (1998), pp. 1–45.

[4] R. Diestel, *Graph Theory*, Graduate Texts in Math. 173, Springer-Verlag, Berlin, 1997.

[5] P. Heggernes and J. A. Telle, *Partitioning graphs into generalized dominating sets*, Nordic J. Comput., 5 (1998), pp. 128–143.

[6] J. van den Heuvel and S. McGuinness, *Colouring the Square of a Planar Graph*, Research Report LSE-CDAM-99-06, Centre for Discrete and Applicable Mathematics, London, UK, 1999.

[7] S. Jendrol' and Z. Skupień, *Local structures in plane maps and distance colourings*, Discrete Math., 236 (2001), pp. 167–177.

[8] T. R. Jensen and B. Toft, *Graph Coloring Problems*, Wiley Interscience, 1995. Archives available online at http://www.imada.sdu.dk/Research/Graphcol/.

[9] A. Kotzig, *Extremal polyhedral graphs*, Ann. New York Acad. Sci., 319 (1979), pp. 569–570.

[10] S. O. Krumke, M. V. Marathe, and S. S. Ravi, *Approximation algorithms for channel assignment in radio networks*, Wireless Networks, 7 (2001), pp. 575–584.

[11] Y.-L. Lin and S. S. Skiena, *Algorithms for square roots of graphs*, SIAM J. Discrete Math., 8 (1995), pp. 99–118.

[12] E. L. Lloyd and S. Ramanathan, *On the complexity of distance-2 coloring*, in Proceedings of the 4th International Conference on Computing and Information (ICCI '92, Toronto), IEEE Computer Society Press, Piscataway, NJ, 1992, pp. 71–74.

[13] S. T. McCormick, *Optimal approximation of sparse Hessians and its equivalence to a graph coloring problem*, Math. Programming, 26 (1983), pp. 153–171.

[14] R. Motwani and M. Sudan, *Computing roots of graphs is hard*, Discrete Appl. Math., 54 (1994), pp. 81–88.

[15] C. St. J. A. Nash-Williams, *Decomposition of finite graphs into forests*, J. London Math. Soc., 39 (1964), p. 12.

[16] S. Ramanathan and E. L. Lloyd, *Scheduling algorithms for multihop radio networks*, IEEE/ACM Trans. on Networking, 1 (1993), pp. 166–177.

[17] A. Sen and M. L. Huson, *A new model for scheduling packet radio networks*, Wireless Networks, 3 (1997), pp. 71–82.

[18] A. Sen and E. Malesinska, *On Approximation Algorithms for Radio Network Scheduling*, Technical Report 96–08, Department Computer Science and Engineering, Arizona State University, Tempe, AZ, 1996.

[19] G. Wegner, *Graphs with Given Diameter and a Coloring Problem*, Technical report, University of Dortmund, Dortmund, Germany, 1977.

[20] X. Zhou, Y. Kanari, and T. Nishizeki, *Generalized vertex-colorings of partial k-trees*, IEICE Trans. Fundamentals, E83-A (2000), pp. 671–678.

SIAM J. Discrete Math.
Vol. 16, No. 4, pp. 663–674

STATISTICAL INFERENCE FOR INTERNAL NETWORK RELIABILITY WITH SPATIAL DEPENDENCE*

I. H. DINWOODIE† AND EDWARD MOSTEIG‡

Abstract. Parameter identifiability and polynomial systems for maximum likelihood estimates are established for a statistical model of network reliability with spatial interaction.

Key words. Curie–Weiss model, Gröbner basis, maximum likelihood estimators, network tomography, parameter identifiability

AMS subject classifications. 62B05, 62F10, 13P10

DOI. 10.1137/S0895480101390813

1. Introduction.

In Cáceres et al. [1], maximum likelihood estimation was studied for a model of network reliability called the Bernoulli model. In this paper, we study a more general model called the interaction model which includes a parameter θ for spatial dependence. The model resembles the Curie–Weiss model in statistical mechanics (see Ellis [7]), because there is interaction among all pairs of edges. Parameter identifiability is established using algebraic methods in section 4. The technique involves some simple toric ideals different than those that arise in the Gröbner bases for the Markov Monte Carlo methods invented by Diaconis and Sturmfels in [5] (see Caffo and Booth [2], Fienberg, Makov, and Steele [8] and Pistone, Riccomagno, and Wynn [12] for further statistical applications of the Gröbner basis of a toric ideal). In section 5, computational methods based on algebraic equations are described for the maximum likelihood estimates and also for certain approximations to the maximum likelihood estimates which are based on a relaxation approach to maximizing the likelihood. The approximations can be obtained by extending existing estimation algorithms for the Bernoulli model of Cáceres et al. [1]. Simulation examples in section 6 confirm that the maximum likelihood estimates have less variability than the approximate estimators, but both work well. Large scale simulations would be interesting.

Let us recall the original problem and introduce some notation. A tree with vertices V and edge set E has root node $0 \in V$ and "leaf" nodes $R \subset V$ (R stands for receivers). The parent nodes will be denoted $V_P := V - R$. All vertices in V_P will be assumed to have at least two child nodes. The multicast statistical experiment is the following. One probe is sent from the root node 0 towards the receiver nodes R, and it copies itself at each vertex onto each subsequent edge on its trip towards the receiver nodes (this is the meaning of "multicast"). The probe is lost on an edge en route to the leaf nodes with a probability that depends on the edge, say $\beta_i \in [0, 1)$, for the edge that connects vertex i to its parent $f(i)$. (We will assume that $\beta_i > 0$ in order to get parameter identifiability.) The observed data is the vector $\mathbf{y} \in \{0, 1\}^R$, where component i indicates whether the multicast signal was lost ($y_i = 1$ means it

*Received by the editors June 13, 2001; accepted for publication (in revised form) April 1, 2003; published electronically September 17, 2003 DATE DATE. This research was supported by NSF grant DMS-0200888.

http://www.siam.org/journals/sidma/16-4/39081.html

†Department of Mathematics, Tulane University, New Orleans, LA 70118 (ihd@math.tulane.edu).

‡Department of Mathematics, Loyola Marymount University, Los Angeles, CA 90045 (emosteig@lmu.edu).

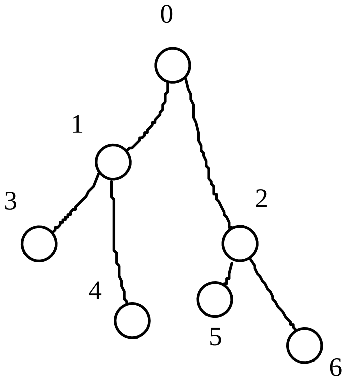


FIG. 1.

was lost) on the trip from 0 to the leaf node $i \in R$. The data $\mathbf{y}$ is the image under a many-to-one linear map A of a hidden outcome $\mathbf{x}$ indicating success or failure on each edge.

This experiment is repeated independently and identically $n \geq 1$ times, and observations $Y_1, \ldots, Y_n$ are a random sample of independently and identically distributed (i.i.d.) $\{0, 1\}$ vectors at the receiver nodes.

Our goal here is to generalize the original Bernoulli multicast model to one with a new interaction parameter $\theta \geq 0$ which adds dependence across all edges.

2. Bernoulli multicast model. In this section we describe the Bernoulli multicast model of Cáceres et al. [1], which we aim to generalize to include spatial dependence.

Let $\mathcal{T}$ be the tree with vertices V numbered $0, 1, \ldots, c$, and let $0 \in V$ be designated the root node. Define $V_0 := V - \{0\}$ to be the collection of nonroot nodes. Let $f : V_0 \to V$ be the function that gives the parent node of a nonroot node, that is, the vertex before i in the unique path from 0 out to $i \in V$. The leaf vertices $R \subset V$ are the ones in V_0 with no descendants or, in other words, those vertices $i \in V_0$ for which $f^{-1}(i) = \emptyset$. In Figure 1, $R = \{3, 4, 5, 6\}$. Let the descendants of node i (the set of nodes whose path back to 0 goes through i but not including i) be denoted $d(i)$. The siblings of i would be $f^{-1} \circ f(i)$.

The assumption that all parent vertices (V_P) have at least two child nodes means that for each $i \in V_P$ it holds that $f^{-1} \circ f(i) - \{i\} \neq \emptyset$. This property will play a role in the identifiability of parameters for this model (cf. Theorem 4.1).

Basic hidden outcomes are vectors of counts $\mathbf{x} = (x_i)_{i \in V_0} \in \{0, 1\}^{V_0}$, where x_i specifies how many probes were lost on the edge $\{f(i), i\}$ (the edges are labelled by the outer vertex). The multicast data $\mathbf{y}$ can be written as a many-to-one function of basic hidden outcomes $\mathbf{x}$ with the help of a routing matrix A. The matrix A will have $d = |R|$ rows, one for each leaf node, and $c = |V_0|$ columns indexed by edges. The row for leaf node v will have "1" in column j if j is on the path from 0 to leaf node i. For the binary tree in Figure 1, the matrix is given by

$$A = \begin{array}{c} \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}. \tag{2.1}$$

The experiment is repeated $n \geq 1$ times. Then the total observed loss vector $\mathbf{y}_k$ (for experiment k out of n) at leaf nodes is given by

$$\mathbf{y}_k = A\mathbf{x}_k.$$

It is convenient to have a $|V| \times |V_0|$ routing matrix B for all nodes in V, not just the leaf nodes. The row for vertex i would have "1" in each column for vertices on the path from 0 to i. $(B\mathbf{x})_i$ for a node $i \in V_0$ would give the total number of messages lost along the path from 0 out to i. The row for vertex 0 in B is identically 0.

Now let β_i be the probability that a probe from vertex $f(i)$ will fail to cross edge $\{f(i), i\}$ to reach vertex $i \in V_0$. We will use an odds-ratio parametrization of the failure probability β_i:

$$\beta_i = \frac{\lambda_i}{1+\lambda_i}, \quad \lambda_i \geq 0.$$

For the basic original "Bernoulli model" it is assumed that a probe fails to cross edge $\{f(i), i\}$ with probability $\lambda_i/(1+\lambda_i)$, and edges and probes all behave independently given failure count data on parent nodes (we generalize this below). The distribution μ_λ on $S_0 = \{\mathbf{x} = (x_i) \in Z_+^{V_0} : x_i \in \{0,1\}\}$ is

$$\begin{aligned} \mu_\lambda(\mathbf{x}) &= \prod_{i \in V_0} \binom{1-(B\mathbf{x})_{f(i)}}{x_i} \frac{\lambda_i^{x_i}}{(1+\lambda_i)^{1-(B\mathbf{x})_{f(i)}}} \\ &= \lambda^{\mathbf{x}} \prod_{i \in V_0} \binom{1-(B\mathbf{x})_{f(i)}}{x_i} \frac{1}{(1+\lambda_i)^{1-(B\mathbf{x})_{f(i)}}} \\ &= \lambda^{\mathbf{x}} \prod_{i \in V_0} \frac{\binom{1-(B\mathbf{x})_{f(i)}}{x_i}}{(1+\lambda_i)} \prod_{i \in V_0} (1+\lambda_i)^{(B\mathbf{x})_{f(i)}} \\ &= \lambda^{\mathbf{x}} \prod_{i \in V_0} \frac{\binom{1-(B\mathbf{x})_{f(i)}}{x_i}}{(1+\lambda_i)} \prod_{i \in V_0 - R} \prod_{j \in d(i)} (1+\lambda_j)^{x_i}. \end{aligned} \tag{2.2}$$

For $i \in V_0$, let $p_i = \prod_{j \in d(i)} (1+\lambda_j)$. Then (2.2) can be written

$$\begin{aligned} \mu_\lambda(\{\mathbf{y}\}) &= \sum_{\{\mathbf{x} \in Z_+^c : A\mathbf{x} = \mathbf{y}\}} h(\mathbf{x}) \frac{\lambda^{\mathbf{x}} \prod_{i \in V_0} p_i(\lambda)^{x_i}}{z_\lambda} \\ &= \sum_{\{\mathbf{x} \in Z_+^c : A\mathbf{x} = \mathbf{y}\}} h(\mathbf{x}) \frac{\lambda^{\mathbf{x}} \mathbf{p}(\lambda)^{\mathbf{x}}}{z_\lambda}, \end{aligned}$$

where

$$\begin{aligned} h(\mathbf{x}) &= \prod_{i \in V_0} \binom{1-(B\mathbf{x})_{f(i)}}{x_i} \\ z_\lambda &= \prod_{i \in V_0} (1+\lambda_i), \end{aligned} \tag{2.3}$$

and the notation $\mathbf{p}(\lambda)^{\mathbf{x}}$ is the usual representation of $\prod_{i=1}^c p_i(\lambda)^{x_i}$. The vector of parameters $(\lambda_i : i \in V_0)$ is identifiable (see Dinwoodie [6]), meaning that two different vectors give rise to two different distributions $\mu_\lambda(\{\mathbf{y}\})$ on the observed (incomplete) data $\mathbf{y}$. (In the Bernoulli model, identifiability holds, in fact, even if vertex v_0 has only one child, and the failure probabilities are allowed to be zero.) This gives consistent maximum likelihood estimates (the estimates converge to the true parameter values) as the sample size n increases.

3. Multicast model with spatial dependence: Interaction model. We have described the original model of Cáceres et al. [1]. Now we generalize the Bernoulli multicast model to one which includes spatial dependence among sibling edges, which we will call the interaction model. We will add an interaction parameter $\theta \geq 0$ which will affect the probability of multiple losses. The new model reduces to the Bernoulli model when $\theta = 1$. The range of the interaction is across all edges (as in the Curie–Weiss model), and values of θ greater than 1 mean that multiple losses are more likely than they would be under the Bernoulli model. Identifiability will be proved in section 4, assuming that the failure rates λ_i are positive (not just nonnegative) and that all parent vertices including the root vertex have at least two children.

Let $V_P = V - R$ be the collection of nonreceiver or parent vertices, which includes the root vertex "0."

For $\mathbf{x} \in \{0,1\}^{V_0}$, let $|\mathbf{x}| := \sum_{i \in V_0} x_i$ be the number of 1's in $\mathbf{x}$. Then the notation $[|\mathbf{x}| - 1]_+$ will give $|\mathbf{x}| - 1$ if there are two or more 1's in $\mathbf{x}$; otherwise, it will vanish. The new law $\nu_{\gamma,\theta}$ in parameters $\gamma_i > 0, i = 1, \ldots, c, \theta \geq 0$, is specified by

$$\begin{aligned} \nu_{\gamma,\theta}(\mathbf{x}) &:= h(\mathbf{x})\,\frac{\gamma^{\mathbf{x}}\theta^{[|\mathbf{x}|-1]_+}}{w_{\gamma,\theta}}, \\ w_{\gamma,\theta} &= \frac{\theta - 1 + z_{\lambda(\theta\gamma)}}{\theta}, \end{aligned} \tag{3.1}$$

where the formula for the normalizing constant $w_{\gamma,\theta}$ relates to z from the Bernoulli model as follows. Consider the one-to-one reparametrization from positive γ to positive λ with inverse given by

$$\gamma_i = \lambda_i p_i(\lambda), \quad i = 1, \ldots, c.$$

Then $\lambda(\theta\gamma)$ is the vector $(\lambda_1(\theta\gamma), \ldots, \lambda_c(\theta\gamma))$ that comes from finding the λ corresponding to $(\theta\gamma_1, \theta\gamma_2, \ldots, \theta\gamma_c)$. From the Bernoulli model, we know that

$$\prod_{i=1}^{c}(1 + \lambda_i(\gamma\theta)) = \sum_{\mathbf{x} \in Z_+^c} h(\mathbf{x})\gamma^{\mathbf{x}}\theta^{|\mathbf{x}|}$$

and separating the case $\mathbf{x} = \mathbf{0}$ gives the formula. In terms of the odds-ratio parameters λ_i, the law ν can be written

$$\nu_{\gamma(\lambda),\theta}(\mathbf{x}) := h(\mathbf{x})\,\frac{\lambda^{\mathbf{x}}\mathbf{p}(\lambda)^{\mathbf{x}}\theta^{[|\mathbf{x}|-1]_+}}{w_{\gamma(\lambda),\theta}}. \tag{3.2}$$

4. Identifiability of parameters. In the model (3.1), we have parameters $\gamma_1, \ldots \gamma_c, \theta$ (or equivalently $\lambda_1, \ldots, \lambda_c, \theta$) with c-dimensional outcomes in $\{0,1\}_+^{V_0}$ and d-dimensional observations (data) in $\{0,1\}_+^{R}$, where $c = |V_0|,\ d = |R|$ (the number of receiver nodes), and A is a $d \times c$ matrix of 0's and 1's.

The parameter pair (γ, θ) is called identifiable for positive γ and nonnegative θ if $\nu_{\gamma,\theta}(\{\mathbf{x} \geq 0 : A\mathbf{x} = \mathbf{y}\}) = \nu_{\gamma',\theta'}(\{\mathbf{x} \geq 0 : A\mathbf{x} = \mathbf{y}\})$ for all $\mathbf{y} \in Z_+^d$ implies that $\gamma = \gamma'$

and $\theta = \theta'$, where γ and γ' are assumed to be positive in each coordinate and θ and θ' are assumed to be nonnegative real numbers. If the parameters are identifiable, then different parameters will lead to different statistical patterns, and consistent estimation is possible under repeated, independent experiments. Otherwise, different parameter values may be statistically indistinguishable based on repeated experimental outcomes.

We define $\delta : V \to \{0, 1, 2, \ldots\}$ recursively on the set of vertices in $\mathcal{T}$. For $v \in R$, set $\delta(v) = 0$. Given a vertex $v \in V - R$, we define $\delta(v) = 1 + \min\{\delta(v') : v' \in f^{-1}(v)\}$. We call $\delta(v)$ the *height* of the vertex v. The *height* of the tree $\mathcal{T}$ is defined as $\delta(\mathcal{T}) := \max\{\delta(v) : v \in V\}$.

THEOREM 4.1. *Assume that all parent vertices V_P have at least two children. Then the parameters $\lambda_i > 0$ and $\theta \geq 0$ are identifiable.*

Proof. We prove the result in case $\delta(\mathcal{T}) \geq 2$. For $\delta(\mathcal{T}) = 1$, the proof is similar but much easier.

Let $\gamma > 0,\ \gamma' > 0,\ \theta \geq 0,\ \theta' \geq 0$ be two possible parameter values. Assume that $\nu_{\gamma,\theta}(\{\mathbf{y}\}) = \nu_{\gamma',\theta'}(\{\mathbf{y}\})$ for all vectors $\mathbf{y}$. We must show that the condition of containment on indeterminate differences implies that $\gamma = \gamma',\ \theta = \theta'$.

Consider the expressions (3.2) for the law $\nu_{\gamma,\theta}$ on the observed vector $\mathbf{y}$. By setting $\mathbf{y} = \mathbf{0}$ and observing that only $\mathbf{x} = \mathbf{0}$ hits $\mathbf{0}$ under A and $h(\mathbf{0}) = 1$, it follows that $w_{\gamma,\theta} = w_{\gamma',\theta'}$. Therefore it follows that for all $\mathbf{y} \in \mathbf{Z}_+^d$,

$$\sum_{A\mathbf{x}=\mathbf{y}} h(\mathbf{x})\,\gamma^{\mathbf{x}}\theta^{[|\mathbf{x}|-1]_+} = \sum_{A\mathbf{x}=\mathbf{y}} h(\mathbf{x})\,{\gamma'}^{\mathbf{x}}{\theta'}^{[|\mathbf{x}|-1]_+}.$$

Given $v \in R$, define $\mathbf{y}_v$ to be the vector in $\mathbf{Z}_+^R$ with "1" in entry for $v \in R$, and "0" elsewhere. For $v \in V - R$, define $\mathbf{y}_v = \sum_{w \in d(v) \cap R} \mathbf{y}_w$. Note that if $v \in R$, then $\sum_{A\mathbf{x}=\mathbf{y}_v} h(\mathbf{x})\gamma^{\mathbf{x}}\theta^{[|\mathbf{x}|-1]_+} = \gamma_v$ and $\sum_{A\mathbf{x}=\mathbf{y}_v} h(\mathbf{x})\gamma^{\mathbf{x}}{\theta'}^{[|\mathbf{x}|-1]_+} = \gamma'_v$, and so $\gamma_v = \gamma'_v$.

We now demonstrate that $\theta = \theta'$. Let v_1, v_2 be receiver nodes that are not siblings. We know such a pair exists since $\delta(\mathcal{T}) \geq 2$, and we have assumed that every parent vertex has at least two children. In this case, $\sum_{A\mathbf{x}=\mathbf{y}_{v_1}+\mathbf{y}_{v_2}} h(\mathbf{x})\gamma^{\mathbf{x}}\theta^{[|\mathbf{x}|-1]_+} = \gamma_{v_1}\gamma_{v_2}\theta$ and $\sum_{A\mathbf{x}=\mathbf{y}_{v_1}+\mathbf{y}_{v_2}} h(\mathbf{x})\gamma^{\mathbf{x}}{\theta'}^{[|\mathbf{x}|-1]_+} = \gamma'_{v_1}\gamma'_{v_2}\theta'$. Since $v_1, v_2 \in R$, we have $\gamma_{v_1} = \gamma'_{v_1}$ and $\gamma_{v_2} = \gamma'_{v_2}$, and so $\theta = \theta'$.

We now prove that $\gamma_v = \gamma'_v$ by induction on the height of the vertex $v \in V_0$. We have already shown that if $\delta(v) = 0$ (that is, if $v \in R$), then $\gamma_v = \gamma'_v$. Now suppose that $\gamma_v = \gamma'_v$ for all v such that $\delta(v) < n$, and let $v' \in V_0$ with $\delta(v') = n$. Since $\theta = \theta'$, we have

$$\sum_{A\mathbf{x}=\mathbf{y}_{v'}} h(\mathbf{x})\,\gamma^{\mathbf{x}} = \sum_{A\mathbf{x}=\mathbf{y}_{v'}} h(\mathbf{x})\,{\gamma'}^{\mathbf{x}}.$$

Note that $\sum_{A\mathbf{x}=\mathbf{y}_{v'}} h(\mathbf{x})\,\gamma^{\mathbf{x}}$ consists of a sum of the term $\gamma_{v'}$ together with terms of the form $\gamma^{\mathbf{x}}$, where $\mathbf{x}$ contains components that correspond to descendants of v'. Similarly, $\sum_{A\mathbf{x}=\mathbf{y}_{v'}} h(\mathbf{x})\,{\gamma'}^{\mathbf{x}}$ consists of a sum of the term $\gamma'_{v'}$ together with terms of the form ${\gamma'}^{\mathbf{x}}$, where $\mathbf{x}$ contains components that correspond to descendants of v'. By the induction hypothesis, whenever $\mathbf{x}$ consists of components that correspond to descendants of v', $\gamma^{\mathbf{x}} = {\gamma'}^{\mathbf{x}}$. Thus, $\gamma_{v'} = \gamma'_{v'}$. □

5. Estimation and inference. In this section, we consider numerical methods for finding estimates of the unknown parameter values. In particular, we seek maximum likelihood estimates and approximations that are based on a relaxation approach. With the "incomplete" data $\mathbf{y}_i = A\mathbf{x}_i$ as a many-to-one function of outcomes

$\mathbf{x}_i$, it seems that the EM-algorithm (see Dempster, Laird, and Rubin [4]) is appropriate. However, it is quite complicated and cumbersome for the interaction model, and here we pursue more direct ways to maximize the likelihood function.

Let the observations for an i.i.d. sample be $\mathbf{y}_1 = A\mathbf{x}_1, \ldots, \mathbf{y}_n = A\mathbf{x}_n$. Observe that the relationship between the Bernoulli model μ_λ and the interaction model $\nu_{\lambda,\theta}$ implies the following formula:

$$\nu_{\gamma,\theta}(\{\mathbf{y}\}) = \mu_{\lambda(\theta\gamma)}(\{\mathbf{y}\}) \left(\frac{\theta I_{\mathbf{0}}(\mathbf{y}) + I_{\neq \mathbf{0}}(\mathbf{y})}{\theta - 1 + z_{\lambda(\theta\gamma)}} \right) z_{\lambda(\theta\gamma)},$$

where $\theta\gamma := (\theta\gamma_1, \ldots, \theta\gamma_c)$.

Let $N_{\mathbf{0}}$ be the number of times the vector $\mathbf{0}$ appears in the sample. The objective function for maximum likelihood estimation is the log-likelihood function l in (γ, θ) given by

$$\begin{aligned} l(\gamma,\theta) &= \frac{1}{n} \sum_{i=1}^{n} \log \nu_{\gamma,\theta}(\{\mathbf{y}_i\}) \\ &= \frac{1}{n} \sum_{i=1}^{n} \log(\mu_{\lambda(\theta\gamma)}(\{\mathbf{y}_i\})) + \frac{N_{\mathbf{0}}}{n} \log(\theta) + \log\left(\frac{z_{\lambda(\theta\gamma)}}{\theta - 1 + z_{\lambda(\theta\gamma)}} \right). \end{aligned} \tag{5.1}$$

It follows that $l(\gamma,\theta) = l_0(\theta\gamma, \theta)$, where l_0 is defined by

$$l_0(\gamma',\theta) = \frac{1}{n} \sum_{i=1}^{n} \log \mu_{\lambda(\gamma')}(\{\mathbf{y}_i\}) + \frac{N_{\mathbf{0}}}{n} \log(\theta) + \log\left(\frac{z_{\lambda(\gamma')}}{\theta - 1 + z_{\lambda(\gamma')}} \right). \tag{5.2}$$

The objective function l_0 (5.2) can be simplified for more efficient numerical optimization. For each $\mathbf{y} \in Z_+^R$, let $V^{\mathbf{y}} \subset V_0$ be the collection of edge labels that are closest to the root whose failure could lead to observation $\mathbf{y}$. For example, in the binary tree in Figure 1, $V^{(1,1,1,1)} = \{1,2\}$, $V^{(1,1,0,1)} = \{1,6\}$. Define polynomials $q_v, \ v \in V_0$ in variables $s_v, \ v \in V_0$ recursively by

$$\begin{aligned} q_r &:= s_r, r \in R, \\ q_v &:= s_v + \prod_{w \in f^{-1}(v)} q_w. \end{aligned}$$

By the independence in the Bernoulli model,

$$\mu_{\lambda(\gamma)}(\{\mathbf{y}\}) = \sum_{A\mathbf{x}=\mathbf{y}} h(\mathbf{x}) \frac{\gamma^{\mathbf{x}}}{z_{\lambda(\gamma)}} = \frac{1}{z_{\lambda(\gamma)}} \prod_{v \in V^{\mathbf{y}}} q_v(\gamma),$$

which leads to the formula for the normalizing constant $z_{\lambda(\gamma)}$ in terms of the variables q_v:

$$z_{\lambda(\gamma)} = \sum_{\mathbf{y} \in \{0,1\}^R} \prod_{v \in V^{\mathbf{y}}} q_v(\gamma).$$

Let $N^v, v \in V_0$, be the number of observations in the sample $\mathbf{y}_1, \ldots, \mathbf{y}_n$ such that the corresponding collections $V^{\mathbf{y}_i}$ include v:

$$N^v := \#\{i : 1 \leq i \leq n, v \in V^{\mathbf{y}_i}\}.$$

Now we can represent the distribution μ_λ in a simplified form:

$$\begin{aligned}\prod_{i=1}^{n} \mu_{\lambda(\gamma)}(\{\mathbf{y}_i\}) &= \frac{1}{z^n_{\lambda(\gamma)}} \prod_{i=1}^{n} \prod_{v \in V^{\mathbf{y}_i}} q_v(\gamma)\\ &= \frac{1}{z^n_{\lambda(\gamma)}} \prod_{v \in V_0} \prod_{i : v \in V^{\mathbf{y}_i}} q_v(\gamma)\\ &= \frac{1}{z^n_{\lambda(\gamma)}} \prod_{v \in V_0} q_v(\gamma)^{N^v}.\end{aligned}$$

This leads to a simpler form of the objective function l_0:

$$l_0(\gamma, \theta) = \frac{1}{n} \sum_{v \in V_0} N^v \log q_v(\gamma) + \frac{N_{\mathbf{0}}}{n} \log(\theta) - \log(\theta - 1 + z_{\lambda(\gamma)}). \tag{5.3}$$

The procedure to maximize l over (γ, θ) is to maximize l_0 over γ', θ and transform back:

$$\begin{aligned}(\hat{\gamma}', \hat{\theta}) &:= \texttt{arg max}_{\gamma > 0, \theta \geq 0}\ l_0(\gamma', \theta),\\ (\hat{\gamma}, \hat{\theta}) &:= (\hat{\gamma}'/\hat{\theta}, \hat{\theta}).\end{aligned} \tag{5.4}$$

Observe that l_0 as it is written in (5.2) can be maximized approximately over $\gamma' > 0,\ \theta > 0$ by extending the methods for the Bernoulli model in what is called a "relaxation" approach. If $\hat{\gamma}'_B$ maximizes $\sum_{i=1}^n \log \mu_{\lambda(\gamma')}(\{\mathbf{y}_i\})$ (in other words, it is the maximum likelihood estimate for the Bernoulli model), then let $\hat{\theta}$ maximize the second part of the objective function, fixing γ'_B:

$$\hat{\theta} := \texttt{arg max}_{\theta > 0}\ \frac{N_{\mathbf{0}}}{n} \log(\theta) + \log\left(\frac{z_{\lambda(\gamma'_B)}}{\theta - 1 + z_{\lambda(\gamma'_B)}}\right).$$

It is a simple calculus exercise to see that $\hat{\theta}$ satisfies the equation

$$\frac{\hat{\theta}}{\hat{\theta} + z_{\lambda(\gamma'_B)} - 1} = \frac{N_{\mathbf{0}}}{n},$$

which can be solved explicitly for $\hat{\theta}$ whenever $N_{\mathbf{0}}/n \in (0, 1)$. Then the combination

$$\begin{aligned}\hat{\gamma}'_B &:= \texttt{arg max}_{\gamma' > \mathbf{0}} \sum_{i=1}^{n} \log \mu_{\lambda(\gamma')}(\{\mathbf{y}_i\}),\\ \hat{\theta}_{rel} &= \frac{z_{\lambda(\hat{\gamma}'_B)} - 1}{(n/N_{\mathbf{0}}) - 1},\\ \hat{\gamma}_{rel} &:= \frac{\hat{\gamma}'_B}{\hat{\theta}}\end{aligned} \tag{5.5}$$

gives an estimator $(\hat{\gamma}_{rel}, \hat{\theta}_{rel})$ for (γ, θ) that is approximately the maximizer of l when θ is near 1, and it is computationally faster than finding the exact maximum likelihood estimates $(\hat{\gamma}, \hat{\theta})$.

The approximate procedure (5.5) can be viewed as one step in a relaxation method that starts with $\theta^0 = 1$ and alternates between maximizing over γ' and maximizing over θ. Observe that $\hat{\theta} = 1$ exactly when $N_{\mathbf{0}}/n = 1/z_{\lambda(\hat{\gamma}')}$, in other words when the observed fraction of $\mathbf{0}$ vectors is the probability of the $\mathbf{0}$ vector under the Bernoulli model with parameter $\hat{\gamma}'$. If the proportion of $\mathbf{0}$'s is higher, then $\hat{\theta} > 1$. In fact, the proportion of $\mathbf{0}$'s under the interaction model with parameters (γ, θ) is higher than the proportion under the Bernoulli model with parameters $\theta\gamma$ when $\theta > 1$:

$$\frac{1}{z_{\lambda(\theta\gamma)}} < \frac{\theta}{\theta - 1 + z_{\lambda(\theta\gamma)}} = \frac{1}{w_{\gamma,\theta}}.$$

We now describe further algebraic features of the quantities in (5.3). With indeterminates $q_c > q_{c-1} > \cdots > q_1$ for a ring $\mathbf{R}[q_c, \ldots, q_1]$ ordered so that the variables indexed by a node's children are greater than the one for the node itself, quantities above can be described precisely in algebraic terms. Let G be the Gröbner basis in $\mathbf{R}[q_c, \ldots, q_1]$ given by $G = \{\prod_{w \in f^{-1}(v)} q_w - q_v, v \in V_0 - R\}$. Then $V^{\mathbf{y}}$ is the collection of indeterminates present in the monomial $\mathtt{nf}(\mathbf{q}^{\mathbf{y}}, G)$, where $\mathtt{nf}$ denotes the normal form with respect to plex order of the monomial $\mathbf{q}^{\mathbf{y}} := \prod_{v \in R} q_v^{y_v}$. We will use the simpler notation $\mathbf{q}^{\mathbf{y}}$ for $\prod_{v \in R} q_v^{y_v}$. Define the polynomial $Z(q_1, \ldots, q_c)$ by

$$Z(\mathbf{q}) := \sum_{\mathbf{y} \in \{0,1\}^R} \prod_{v \in V^{\mathbf{y}}} q_v.$$

Then it can be shown that

$$\begin{aligned} z_{\lambda(\gamma)} &= Z(\mathbf{q}(\gamma)), \\ Z(\mathbf{q}) &= \mathtt{nf}\left(\prod_{v \in R} 1 + q_v, G\right) = \sum_{\mathbf{y} \in \{0,1\}^R} \mathtt{nf}(\mathbf{q}^{\mathbf{y}}, G). \end{aligned}$$

An interior stationary point for $l_0(\gamma, \theta)$ can be found as a positive solution to a system of polynomial equations in the variables q_v, θ by the chain rule for derivatives. Using the definition above for $Z(\mathbf{q})$ in terms of $q_v, v \in V_0$, consider l_0 in the variables $q_1, \ldots, q_c$:

$$l_0(\mathbf{q}, \theta) = \frac{1}{n} \sum_{v \in V_0} N^v \log q_v + \frac{N_{\mathbf{0}}}{n} \log(\theta) - \log(\theta - 1 + Z(\mathbf{q})). \tag{5.6}$$

Setting $\nabla l_0 = \mathbf{0}$ leads to $c + 1$ polynomial equations:

$$\begin{aligned} &\left(\frac{N^v}{n - N_{\mathbf{0}}}\right)(Z(\hat{\mathbf{q}}) - 1) = \hat{q}_v \frac{\partial Z(\hat{\mathbf{q}})}{\partial q_v} = \sum_{\mathbf{y}: v \in V^{\mathbf{y}}} \prod_{w \in V^{\mathbf{y}}} \hat{q}_w, \quad v \in V_0, \\ &\left(\frac{N_{\mathbf{0}}}{n - N_{\mathbf{0}}}\right)(Z(\hat{\mathbf{q}}) - 1) = \hat{\theta}, \end{aligned} \tag{5.7}$$

for which a solution $\hat{q}_v$ (in the range of $(q_1(\gamma), \ldots, q_c(\gamma))$ for positive $\gamma > 0$ and positive in $\hat{\theta}$) is sought. This can then be transformed back to positive $\hat{\gamma}'$ and used in (5.4). A search for monomials in the right side can be avoided. In fact,

$$\sum_{\mathbf{y}: v \in V^{\mathbf{y}}} \prod_{w \in V^{\mathbf{y}}} q_w = Z(\mathbf{q}) - \mathtt{nf}(Z(\mathbf{q}), \{q_v\}).$$

Thus the polynomial system (5.7) can be rewritten as follows.

PROPOSITION 5.1. *Suppose that $\hat{\gamma}, \hat{\theta}$ are strictly positive, and maximize $l_0(\gamma, \theta)$ over positive $\gamma \in \mathbf{R}^c$ and positive $\theta \in \mathbf{R}$. Then $\mathbf{q}(\gamma) = (q_v(\gamma))_{v \in V_0}$ satisfies the polynomial system of $c+1$ equations:*

$$\begin{aligned} \left(\frac{N^v}{n - N_{\mathbf{0}}}\right)(Z(\hat{\mathbf{q}}) - 1) &= Z(\hat{\mathbf{q}}) - \mathtt{nf}(Z(\hat{\mathbf{q}}), \{q_v\}), \\ \hat{\theta} &= \left(\frac{N_{\mathbf{0}}}{n - N_{\mathbf{0}}}\right)(Z(\hat{\mathbf{q}}) - 1). \end{aligned} \tag{5.8}$$

Proof. The variables $\mathbf{q}$ and γ are in one-to-one correspondence. The condition that $\nabla l_0(\mathbf{q}, \theta) = \mathbf{0}$ holds at an interior maximum $\hat{\mathbf{q}}$ $(= (q_1(\hat{\gamma}), \ldots, q_c(\hat{\gamma}))$ $), \hat{\theta}$ leads to the system above in the variables $\mathbf{q}$. □

Observe that the vector $(N^v, v \in V_0)$ is a multidegree of a monomial:

$$(N^v, v \in V_0) = \mathtt{multideg} \prod_{i=1}^{n} \mathtt{nf}(\mathbf{q}^{\mathbf{y}_i}, G).$$

This representation implies the following algebraic formula:

$$\begin{aligned} \prod_{i=1}^{n} \mu_{\lambda(\gamma)}(\{\mathbf{y}_i\}) &= \frac{1}{z^n_{\lambda(\gamma)}} \prod_{i=1}^{n} \prod_{v \in V^{\mathbf{y}_i}} q_v(\gamma) \\ &= \frac{1}{z^n_{\lambda(\gamma)}} \prod_{v \in V_0} q_v(\gamma)^{N^v} \\ &= \frac{1}{z^n_{\lambda(\gamma)}} \prod_{i=1}^{n} \mathtt{nf}(\mathbf{q}^{\mathbf{y}_i}, G)(\gamma). \end{aligned}$$

The main reason one may be interested in the approximate method (5.5) is because the approximate estimators are relatively easy to compute. Most of the work is in fitting a Bernoulli model (finding γ'_B). Just as a stationary point for (5.6) satisfied a polynomial system (5.7), a stationary point γ'_B for the Bernoulli part of (5.5) $\sum_{i=1}^{n} \log(\mu_\lambda(\{\mathbf{y}_i\})) = \sum_v N^v \log(q_v) - Z(\mathbf{q})$ in the variables $q_1, \ldots, q_c$ satisfies a polynomial system

$$\frac{N^v}{n} Z(\hat{\mathbf{q}}) = \hat{q}_v \frac{\partial Z(\hat{\mathbf{q}})}{\partial q_v} = \sum_{\mathbf{y}: v \in V^{\mathbf{y}}} \prod_{w \in V^{\mathbf{y}}} \hat{q}_w = Z(\hat{\mathbf{q}}) - \mathtt{nf}(Z(\hat{\mathbf{q}}), \{q_v\}).$$

There is also the recursive numerical method of Cáceres et al. [1].

Example 5.1. Consider the simplest example of a tree with root node 0 and two child nodes 1, 2, with data $\mathbf{y}_1 = (0,0)$, $\mathbf{y}_2 = (0,1)$, $\mathbf{y}_3 = (1,0)$, $\mathbf{y}_4 = (1,1)$. There are three parameters: $\gamma_1, \gamma_2, \theta$.

To use the approximate method (5.5), first fit the Bernoulli model with parameters γ'_B. The optimal values are $\hat{\gamma}'_B = (1,1)$. Then the value of $\hat{\theta}_{rel}$ becomes $(4-1)/(4/1-1) = 1$. Finally, $\hat{\gamma}_{rel}$ is found to be the correct maximizer $(1,1)$.

For the polynomial system (5.8), $N^1 = 2$, $N^2 = 2$, $N_{\mathbf{0}} = 1$, and the polynomial system becomes

$$\begin{aligned} &\frac{2}{3}(q_1 q_2 + q_1 + q_2) = q_1 q_2 + q_1, \\ &\frac{2}{3}(q_1 q_2 + q_1 + q_2) = q_1 q_2 + q_2, \\ &\frac{1}{3}(q_1 q_2 + q_1 + q_2) = \theta, \end{aligned}$$

which can be solved for $\hat{q}_1 = \hat{q}_2 = \hat{\theta} = 1$. This transforms to the solution $\hat{\gamma}'_1 = 1, \hat{\gamma}'_2 = 1, \hat{\theta} = 1$, which by (5.4) leads to the final estimates $\hat{\gamma}_1 = 1, \hat{\gamma}_2 = 1, \hat{\theta} = 1$.

Example 5.2. Consider the binary tree in Figure 1 with $\gamma_i = 1$, $\theta = 2$. Then $\lambda(\gamma) = (1/4, 1/4, 1, 1, 1, 1)$, $\lambda(\theta\gamma) = (2/9, 2/9, 2, 2, 2, 2)$, $w_{\gamma,\theta} = 61$. To find an interior stationary point for the function $l_0(\mathbf{q}, \theta)$, we solve the following system (5.8) for positive q_v, θ:

$$\begin{aligned}
&\frac{N^1}{n - N_{\mathbf{0}}}((1 + q_3 + q_4 + q_1)(1 + q_5 + q_6 + q_2) - 1) = q_1(1 + q_5 + q_6 + q_2),\\
&\frac{N^2}{n - N_{\mathbf{0}}}((1 + q_3 + q_4 + q_1)(1 + q_5 + q_6 + q_2) - 1) = q_2(1 + q_3 + q_4 + q_1),\\
&\frac{N^3}{n - N_{\mathbf{0}}}((1 + q_3 + q_4 + q_1)(1 + q_5 + q_6 + q_2) - 1) = q_3(1 + q_5 + q_6 + q_2),\\
&\frac{N^4}{n - N_{\mathbf{0}}}((1 + q_3 + q_4 + q_1)(1 + q_5 + q_6 + q_2) - 1) = q_4(1 + q_5 + q_6 + q_2),\\
&\frac{N^5}{n - N_{\mathbf{0}}}((1 + q_3 + q_4 + q_1)(1 + q_5 + q_6 + q_2) - 1) = q_5(1 + q_3 + q_4 + q_1),\\
&\frac{N^6}{n - N_{\mathbf{0}}}((1 + q_3 + q_4 + q_1)(1 + q_5 + q_6 + q_2) - 1) = q_6(1 + q_3 + q_4 + q_1),\\
&\frac{N_{\mathbf{0}}}{n - N_{\mathbf{0}}}((1 + q_3 + q_4 + q_1)(1 + q_5 + q_6 + q_2) - 1) = \theta.
\end{aligned}$$

The solution must be transformed to $\hat{\gamma}'$ and then again transformed to $\hat{\gamma}$ using (5.4).

6. Simulation. Simulating from the interaction model is useful for several applications. First, it can be used for bootstrap variance estimates of the parameter estimates. That is, after finding particular estimates $\hat{\gamma}, \hat{\theta}$, one can simulate the process with these parameters, recomputing estimates each time for both γ and θ. Then sample variances can be used to understand the variability. A second application is to assess the bias of the estimates.

Whereas the Bernoulli model is easy to simulate, simulation of the full interaction model requires some extensions. A Metropolis or Metropolis–Hastings algorithm (see Hastings [11] or Fishman [9]) is possible but complicated compared to importance sampling.

To find the expectation and the mean square error for estimators of γ, θ on sample size n, it is more efficient to simulate from the distribution $\mu_{\lambda(\gamma)}$ with the Bernoulli model and reweight the relevant random variables with the ratio of the distributions. This is "importance sampling." It is ideal to find bias and mean square error for estimates in the interaction model, because $\mu_{\lambda(\gamma)}$ is close the $\nu_{\gamma,\theta}$, and yet it is very easy to simulate from the Bernoulli distribution $\mu_{\lambda(\gamma)}$. The method can be easily justified. If $X_{k,1}, X_{k,2}, X_{k,3}, \ldots, X_{k,n}$, $k = 1, 2, 3 \ldots$, are random samples of size n from the distribution $\mu_{\lambda(\theta\gamma)}(\mathbf{x})$, let $\hat{\gamma}_1^k$ be the value of the estimator for γ_1 on the sample k (whose distribution will depend on n). Then as $m \to \infty$,

$$\begin{aligned}
&\frac{1}{m}\sum_{k=1}^{m} \hat{\gamma}_1^k \prod_{i=1}^{n} \frac{\nu_{\gamma,\theta}(X_{k,i})}{\mu_{\lambda(\theta\gamma)}(X_{k,i})} \to E_{\mu_{\lambda(\theta\gamma)}} \hat{\gamma}_1 \prod_{k=1}^{n} \frac{\nu_{\gamma,\theta}(X_k)}{\mu_{\lambda(\theta\gamma)}(X_k)} = E_{\nu_{\gamma,\theta}} \hat{\gamma}_1,\\
&\frac{1}{m}\sum_{k=1}^{m} \big(\hat{\gamma}_1^k - \gamma_1\big)^2 \prod_{i=1}^{n} \frac{\nu_{\gamma,\theta}(X_{k,i})}{\mu_{\lambda(\theta\gamma)}(X_{k,i})} = E_{\nu_{\gamma,\theta}} \big(\hat{\gamma}_1 - \gamma_1\big)^2.
\end{aligned} \tag{6.1}$$

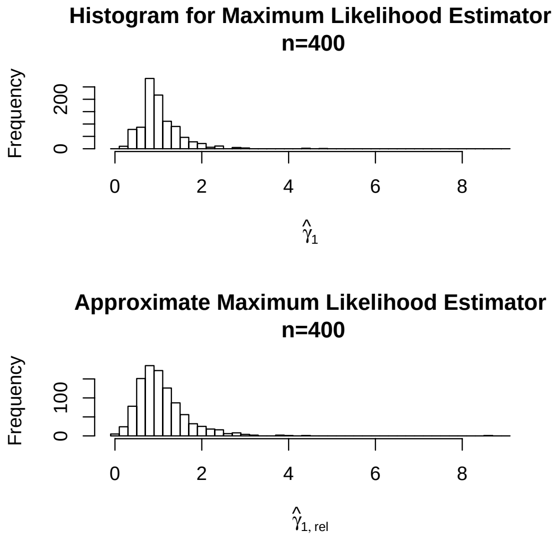


Fig. 2.

Example 6.1. With $n = 400$ and $\gamma = (1, 1, 1, 1, 1, 1)$, $\theta = 1.2$ on the binary tree in Figure 1, 1000 outcomes of both the maximum likelihood estimator $\hat{\gamma}_1$ and the approximate relaxation estimator $\hat{\gamma}_{1,rel}$ gave sample means of 1.034 and 1.077, with empirical mean square errors of .226 and .380. The corresponding histograms are shown in Figure 2. The greater value of θ results in more variability in the estimates for γ.

Problems for further research include finding efficient methods to numerically resolve the polynomial systems (5.7) and (5.8), such as characterizing a Gröbner basis in plex order or another elimination order. Also, variance estimates or bounds on the variance more practical than the asymptotic quantities from Fisher information could be useful for experimental design. Finally, extending the interaction model described here to one with more general interaction, including shorter range, could be of practical interest.

Acknowledgment. We used *R* from lib.stat.cmu.edu for simulations.

REFERENCES

[1] R. Cáceres, N. G. Duffield, J. Horowitz, D. Towsley, and T. Bu, *Multicast-based inference of network internal characteristics: Accuracy of packet loss estimation*, IEEE Trans. Inform. Theory, 45 (2000), pp. 2462–2480.

[2] B. Caffo and J. Booth, *A Markov chain Monte Carlo algorithm for approximating exact conditional tests*, J. Comput. Graph. Statist., 10 (2001), pp. 730–745.

[3] A. Capani, G. Niesi, and L. Robbiano, *Cocoa: A System for Doing Computations in Commutative Algebra*, 2000. Available via anonymous ftp from cocoa.dima.unige.it.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, J. Roy. Statist. Soc. Ser. B, 39 (1977), pp. 1–38.

[5] P. Diaconis and B. Sturmfels, *Algebraic algorithms for sampling from conditional distributions*, Ann. Statist., 26 (1998), pp. 363–397.

[6] I. H. Dinwoodie, *Polynomial statistical models*, Stat. Comput., 12 (2002), pp. 307–314.

[7] R. S. Ellis, *Entropy, Large Deviations, and Statistical Mechanics*, Springer, New York, 1985.

[8] S. E. Fienberg, U. E. Makov, and R. J. Steele, *Disclosure limitation using perturbation and related methods for categorical data*, Journal of Official Statistics, 14 (1998), pp. 485–511.

[9] G. Fishman, *Monte Carlo*, Springer, New York, 1996.

[10] E. L. Lehman, *Theory of Point Estimation*, Wiley, New York, 1983.

[11] W. K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.

[12] G. Pistone, E. Riccomagno and H. P. Wynn, *Algebraic Statistics: Computational Commutative Algebra in Statistics*, Chapman and Hall, London, 2001.

[13] B. Sturmfels, *Gröbner Bases and Convex Polytopes*, AMS, Providence, RI, 1996.