Edge Covers of Setpairs and the Iterative Rounding Method

Authors:

- Joseph Cheriyan (corresponding author)
  Department of Combinatorics and Optimization,
  University of Waterloo, 200 University Ave. West,
  Waterloo, ON, Canada, N2L 3G1.
  email: `jcheriyan@math.uwaterloo.ca`

- Santosh Vempala
  Department of Mathematics,
  M.I.T.,
  Cambridge, MA 02139, USA.
  email: `vempala@math.mit.edu`

Mailing address: Send all correspondence and e-mails to Joseph Cheriyan.

Abbreviated title: Iterative Rounding for Setpairs

Mathematics Subject Classification (2000):
Primary: 68W25, 90C35.
Secondary: 05C40, 68R10, 90C27, 90B10.

EDGE COVERS OF SETPAIRS AND THE ITERATIVE ROUNDING METHOD

September 3, 2001

JOSEPH CHERIYAN AND SANTOSH VEMPALA

Abstract. Given a digraph $G = (V, E)$, we study a linear programming relaxation of the problem of finding a minimum-cost edge cover of pairs of sets of vertices (called setpairs). Each setpair has a nonnegative integer-valued requirement, and the requirement function is crossing bisupermodular. Our results are as follows: (1) An extreme point of the LP is characterized by a noncrossing family of tight setpairs, $\mathcal{L}$ (where $|\mathcal{L}| \le |E|$). (2) In any extreme point $\boldsymbol{x}$, there exists an edge e with $x_e \ge \Theta(1)/\sqrt{|\mathcal{L}|}$, and there is an example showing that this lower bound is best possible. (3) The iterative rounding method applies to the LP and gives an integer solution of cost $O(\sqrt{|\mathcal{L}|}) = O(\sqrt{|E|})$ times the LP's optimal value. The proof of (2) relies on the fact that $\mathcal{L}$ can be represented by a special type of partially ordered set that we call diamond-free.

1. Introduction

Many NP-hard problems in network design including the Steiner tree problem and its generalizations are captured by the following formulation. We are given an (undirected) graph $G = (V, E)$ where each edge e has a nonnegative cost c_e, and each subset of vertices S has a nonnegative integer requirement $f(S)$. The problem is to find a minimum-cost subgraph H that satisfies all the requirements, i.e., H should have at least $f(S)$ edges in every cut $(S, V \setminus S)$. This can be modelled as an integer program.

$$\begin{aligned} (SIP) \quad \text{minimize} \; & \sum_e c_e x_e \\ \text{subject to} \; & \sum_{e \in (S, V\setminus S)} x_e \;\ge\; f(S), \qquad \forall S \subset V \\ & x_e \;\in\; \{0,1\}, \qquad \forall e \in E. \end{aligned}$$

Let (SLP) be the linear programming relaxation of (SIP). The requirement function $f(\cdot)$ should be such that (SIP) models some interesting problems in network design, (SLP) has a provably

Mathematics Subject Classification (2000): Primary: 68W25, 90C35. Secondary: 05C40, 68R10, 90C27, 90B10.

Key words and phrases. network design, vertex connectivity, setpair formulation, bisubmodular functions, linear programming relaxation, iterative rounding, approximation algorithms.

small integrality ratio, and (SLP) is solvable in polynomial time. Approximation algorithms based on (SIP) and (SLP) were designed and analyzed by Agrawal, Klein and Ravi [1], Goemans and Williamson [7], Williamson et al [12], and Goemans et al [6]. Then Jain [5] gave a 2-approximation algorithm for the case of weakly supermodular requirement functions via a technique called iterative rounding. A key discovery in [5] is that every non-zero extreme point $\boldsymbol{x}$ of (SLP) has $\max_{e \in E}\{x_e\} \geq \frac{1}{2}$. Subsequently, Melkonian and Tardos [9] studied the problem on directed graphs, and proved that if the requirement function is crossing supermodular, then every non-zero extreme point of their linear programming relaxation has an edge of value at least $\frac{1}{4}$.

There are several interesting problems in network design that elude the formulation of (SIP), such as the problem of finding a minimum-cost k-vertex connected spanning subgraph. Frank and Jordan [4] gave a more general formulation where pairs of vertex sets have requirements (also, see Schrijver [10] for earlier related results). In this formulation, we are given a digraph $G = (V, E)$ and each edge e has a nonnegative cost c_e. A *setpair* is an ordered pair of vertex sets $W = (W_t, W_h)$, where $W_t \subseteq V$ is called the *tail*, and $W_h \subseteq V$ is called the *head* (either W_t or W_h may be the empty set). Let $\mathcal{S}$ be the set of all setpairs. For a setpair W, $\delta(W)$ denotes the set of edges covering W, i.e., $\delta(W) = \{uv \in E | u \in W_t, v \in W_h\}$. Each setpair W has a nonnegative, integer requirement $f(W)$. The problem is to find a minimum-cost subgraph that satisfies all the requirements. (Note that the requirement function $f(\cdot)$ of (SIP) is the special case where every setpair with positive requirement is a partition of V and has the form $(S, V \setminus S)$ where $S \subseteq V$.)

$$
\begin{aligned}
(IP) \quad \text{minimize} \; & \sum_{e} c_e x_e \\
\text{subject to} \; & \sum_{e \in \delta(W)} x_e \;\geq\; f(W), \qquad \forall W \in \mathcal{S} \\
& \qquad\quad x_e \;\in\; \{0, 1\}, \qquad \forall e \in E.
\end{aligned}
$$

Throughout, we assume that the requirement function f of (IP) is *crossing bisupermodular* (this is defined in Section 2). Frank and Jordan [4] used this formulation to derive min-max results for special cost functions, and moreover, they showed that the linear programming relaxation is solvable in polynomial time. Fleiner [3] has related results. The problem of finding a minimum-cost k-vertex connected spanning subgraph of a digraph may be modeled by (IP) by taking the requirement function to be $f(W_t, W_h) = k - (|V| - |W_t \cup W_h|)$, where W_t, W_h are nonempty vertex subsets; this function is crossing bisupermodular.

We study the linear programming relaxation (LP) for arbitrary nonnegative cost functions. In Section 2 we show that for any extreme point of (LP), the space of incidence vectors of tight setpairs (setpairs whose requirement is satisfied exactly) is spanned by the incidence vectors of a *noncrossing* family of tight setpairs. A noncrossing family of setpairs is the analogue of a laminar family of sets. (Recall that two sets are *laminar* if they are either disjoint or one is contained in the other.) In Section 3, we study noncrossing families of setpairs by representing them as partially ordered sets (posets). It turns out that the Hasse diagram of such a poset has a special property — any two chains (dipaths) of the poset have at most one subchain in common. We refer to such posets as *diamond-free* posets. Based on this, we prove the following result. (Note the contrast with (SLP), see [5, 9].)

Theorem 1.1. *For any digraph $G = (V, E)$, any nonzero extreme point $\boldsymbol{x}$ of (LP) satisfies*

$$\max_{e \in E}\{x_e\} \geq \frac{\Theta(1)}{\sqrt{|E|}}.$$

In Section 4, we show that the bound in Theorem 1.1 is the best possible, up to a constant factor.

Theorem 1.2. *Given any sufficiently large integer $|E|$, there exists a digraph $G = (V, E)$ such that (LP) has an extreme point $\boldsymbol{x}$ satisfying*

$$\max_{e \in E}\{x_e\} \leq \frac{\Theta(1)}{\sqrt{|E|}}.$$

The rest of the introduction discusses the iterative rounding method, and addresses some algorithmic questions that arise when this method is applied to (LP). To apply the *iterative rounding* method, we formulate the problem as an integer program, and then solve the linear programming relaxation to find a basic (extreme point) optimum solution $\boldsymbol{x}$. Pick an edge e^* of highest weight (i.e., $x_{e^*} \geq x_e$, $\forall e \in E$) and add it to the solution subgraph H (initially, $E(H)$ is empty). Then update the linear program and the integer program, since we implicitly fixed the variable x_{e^*} at value 1. In detail, we decrease by 1 the r.h.s. of every constraint where the variable x_{e^*} occurs, and then we remove this variable from the linear program. The resulting linear program is the same as the linear program for the "reduced" problem where the edge e^* is pre-selected for H. Under appropriate conditions on the requirement function f, the problem turns out to be "self reducible," i.e., the essential properties of the original problem are preserved in the reduced problem. We iteratively solve the reduced problem. Jain [5] applied this method to (SIP), and proved that it achieves an approximation guarantee of 2 provided that the requirement function f is weakly supermodular. (Such requirement functions capture several interesting problems, e.g., the Steiner network problem.) His analysis is based on a key property of (SLP): every non-zero extreme point has an edge of weight at least $\frac{1}{2}$. This result is based on an extension of a classic result that, under appropriate conditions on the requirement function f, every extreme point of (SLP) is characterized by a laminar family of "tight sets." Jain's analysis [5, Theorem 3.2] applies in a general setting: if the linear program has the self reducibility property, and for every nonzero basic solution $\boldsymbol{x}$ we have a lower bound of ϕ on $\max_{e\in E} x_e$, then the approximation guarantee is $\frac{1}{\phi}$.

The iterative rounding method applies to the setpairs formulation (IP), and gives an approximation algorithm that achieves a guarantee of $O(\sqrt{|E|})$. This follows from Theorem 1.1, and the fact that (LP) has the desired self reducibility property (since the crossing bisupermodular property of the requirement function is preserved on subtracting a bisubmodular function, see Section 2). Theorem 1.2 shows that the $O(\sqrt{|E|})$ approximation guarantee is tight. (LP) is solvable in polynomial time via the ellipsoid method, since a polynomial time separation subroutine is available (see [4, Lemma 7.2]). Although the approximation guarantee of the iterative rounding method hinges on a key property of basic solutions of the linear program, the method can be implemented efficiently via a polynomial time algorithm for finding an optimal solution (not necessarily basic). For this, we take each edge e in turn, and append to (LP) the constraint $x_e \geq \phi$, where ϕ is the lower bound in Theorem 1.1. One of these variants of (LP) has the same optimal value as (LP) (by Theorem 1.1), hence any optimal solution to that variant suffices for the iterative rounding method. There is another algorithmic issue worth noting. For many of the specific problems in network design that are captured by (IP), the relaxation (LP) can be written as a compact linear program via a "flow formulation," and an appropriate optimal solution (not necessarily basic) can be found in strongly polynomial time via Tardos' algorithm [11]. This is similar to the method used by Jain in [5, Section 9]. (For example, consider the problem of finding a minimum-cost k-vertex connected spanning subgraph of a digraph. For each vertex v, we split v into a pair of vertices v', v'', replace incoming edges to v by incoming edges to v', replace outgoing edges from v by outgoing edges from v'', and add the directed edge $v'v''$. The goal is to assign a non-negative real-valued capacity $x_e \leq 1$ to each edge e such that $\sum_e c_e x_e$ is minimized, and such that the max-flow for every ordered pair of vertices v'', w' is at least k, where each "new edge" $v'v''$ gets a capacity of 1.)

In the rest of the paper, an edge means a directed edge of the input digraph G.

2. Characterizing Extreme Points via Noncrossing Families

Two setpairs W, Y are *comparable* if either $W_t \supseteq Y_t$, $W_h \subseteq Y_h$, (denoted as $W \preceq Y$), or $W_t \subseteq Y_t$, $W_h \supseteq Y_h$, (denoted as $W \succeq Y$). Setpairs W, Y are *noncrossing* if either they are comparable, or their heads are disjoint ($W_h \cap Y_h = \emptyset$), or their tails are disjoint ($W_t \cap Y_t = \emptyset$); otherwise W, Y *cross*. A family of setpairs $\mathcal{L} \subseteq \mathcal{S}$ is called *noncrossing* if every two setpairs in $\mathcal{L}$ are noncrossing. For two crossing setpairs W, Y let $W \otimes Y$ denote the setpair $(W_t \cup Y_t, W_h \cap Y_h)$ and let $W \oplus Y$ denote the setpair $(W_t \cap Y_t, W_h \cup Y_h)$. Note that W (similarly, Y) is $\preceq W \oplus Y$ and is $\succeq W \otimes Y$. (If both W and Y are partitions of V, so $W = (V \setminus W_h, W_h)$, $Y = (V \setminus Y_h, Y_h)$, then note that $W \otimes Y$ is the partition of V with head $W_h \cap Y_h$, and $W \oplus Y$ is the partition of V with head $W_h \cup Y_h$.) A real-valued function f on $\mathcal{S}$, $f : \mathcal{S} \rightarrow \mathbb{R}$, is called *bisubmodular* if for any two setpairs W and Y we have

$$f(W) + f(Y) \geq f(W \otimes Y) + f(W \oplus Y).$$

A non-negative, integer-valued function f on $\mathcal{S}$, $f : \mathcal{S} \rightarrow \mathbb{Z}_+$, is called *crossing bisupermodular* if for any two crossing setpairs W and Y with $f(W) > 0$ and $f(Y) > 0$, we have

$$f(W) + f(Y) \leq f(W \otimes Y) + f(W \oplus Y).$$

Let χ_W denote the zero-one incidence vector of $\delta(W)$. For any two setpairs W and Y, note that if an edge is present in $\delta(W \oplus Y)$ or $\delta(W \otimes Y)$, then it is present in $\delta(W)$ or $\delta(Y)$, and if an edge is present in both $\delta(W \oplus Y)$ and $\delta(W \otimes Y)$, then it is present in both $\delta(W)$ and $\delta(Y)$. Hence, we have $\chi_{W \otimes Y} + \chi_{W \oplus Y} \leq \chi_W + \chi_Y$. Consequently, for any non-negative vector $\boldsymbol{x} : E \rightarrow \mathbb{R}_+$ on the edges, the corresponding function on setpairs, $x(\delta(W)) = \sum_{e \in \delta(W)} x_e$, is bisubmodular. (For any vector $\boldsymbol{x}$ on a groundset U and a subset Q of U, $x(Q)$ denotes $\sum_{i \in Q} x_i$.) Also, see Figure 1.

Given a feasible solution $\boldsymbol{x}$ of (LP), a setpair W is called *tight* (w.r.t. $\boldsymbol{x}$) if $x(\delta(W)) = f(W)$.

Theorem 2.1. *Let the requirement function f of (LP) be crossing bisupermodular, and let $\boldsymbol{x}$ be an extreme point solution of (LP) such that $0 < x_e < 1$ for each edge $e \in E$. Then there exists a noncrossing family of tight setpairs $\mathcal{L}$ such that*

(i) *every setpair $W \in \mathcal{L}$ has $f(W) \geq 1$,*
(ii) $|\mathcal{L}| = |E|$,
(iii) *the vectors χ_W, $W \in \mathcal{L}$ are linearly independent, and*
(iv) *$\boldsymbol{x}$ is the unique solution to $\{x(\delta(W)) = f(W),\ \forall W \in \mathcal{L}\}$.*

The proof is based on the next two lemmas. The first of these lemmas "uncrosses" two tight setpairs that cross.

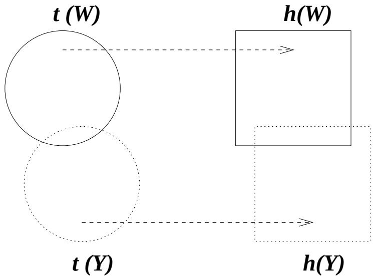

FIGURE 1. Illustration of crossing setpairs. The dashed edges contribute to $x(\delta(W)) + x(\delta(Y))$ but not to $x(\delta(W \otimes Y)) + x(\delta(W \oplus Y))$.

Lemma 2.2. *Let* $\boldsymbol{x} : E \rightarrow \mathbb{R}$ *be a feasible solution of (LP). If two setpairs* W, Y *with* $f(W) > 0, f(Y) > 0$ *are tight and crossing, then the setpairs* $W \otimes Y, W \oplus Y$ *are tight. Moreover, if* $x_e > 0$ *for each edge* $e \in E$*, then*

$$\chi_W + \chi_Y = \chi_{W \otimes Y} + \chi_{W \oplus Y}.$$

Proof. The requirement function $f(\cdot)$ is crossing bisupermodular, and the "edge supply" function $x(\delta(\cdot))$ satisfies the bisubmodular inequality $x(\delta(W)) + x(\delta(Y)) \geq x(\delta(W \otimes Y)) + x(\delta(W \oplus Y))$. Therefore, we have

$$f(W \otimes Y) + f(W \oplus Y) \leq x(\delta(W \otimes Y)) + x(\delta(W \oplus Y)) \leq$$
$$x(\delta(W)) + x(\delta(Y)) = f(W) + f(Y) \leq f(W \otimes Y) + f(W \oplus Y).$$

Hence, all the inequalities hold as equations, and so $W \otimes Y, W \oplus Y$ are tight.

The second statement in the lemma follows since we have $x(\delta(W \otimes Y)) + x(\delta(W \oplus Y)) = x(\delta(W)) + x(\delta(Y))$, and $x_e > 0$ for each edge $e \in E$. Hence, the inequality $\chi_W + \chi_Y \geq \chi_{W \otimes Y} + \chi_{W \oplus Y}$ holds as an equation. $\square$

Lemma 2.3. *Let* L *and* S *be two crossing setpairs. Let* $N = S \otimes L$ *(or, let* $N = S \oplus L$*). If another setpair* J *crosses* N*, then either* J *crosses* S *or* J *crosses* L*.*

Proof. We prove the lemma for the case $N = S \otimes L$; the other case is similar. The proof is by contradiction. Suppose the lemma fails. Then there is a setpair $J \in \mathcal{L}$ such that J, N cross (so $J_t \cap N_t \neq \emptyset$ and $J_h \cap N_h \neq \emptyset$), but both J, L and J, S are noncrossing.

We have four main cases, depending on whether J, L are head disjoint, tail disjoint, $J \succeq L$ or $J \preceq L$.

(i) J, L are head disjoint: Then J, N are head disjoint (by $N_h = S_h \cap L_h$) so J, N do not cross.

(ii) J, L are tail disjoint: We have three subcases, depending on the tails of J, S.

- J_t properly intersects S_t:

  Then J, S are head disjoint (since J, S are noncrossing) so J, N are also head disjoint, and do not cross.

- $J_t \subseteq S_t$:

  Since J, S are noncrossing, either J, S are head disjoint, in which case J, N are head disjoint and do not cross, or $J_h \supseteq S_h$, in which case $J_h \supseteq N_h$, so $J \succeq N$ and J, N do not cross.

- $J_t \supseteq S_t$:

  This is not possible, since J, L are tail disjoint, and S_t intersects L_t (since L, S cross).

(iii) $J \succeq L$: Then $J \succeq N$ since $J_h \supseteq L_h \supseteq N_h$ and $J_t \subseteq L_t \subseteq N_t$.

(iv) $J \preceq L$: As in case(ii), we have three subcases, depending on the tails of J, S.

- J_t properly intersects S_t:

  Similar to case(ii) above, first subcase.

- $J_t \subseteq S_t$:

  Similar to case(ii) above, second subcase.

- $J_t \supseteq S_t$:

  Since J, S do not cross, either J, S are head disjoint, in which case J, N are head disjoint and do not cross, or $J_h \subseteq S_h$, in which case $J \preceq N$ since $J_h \subseteq S_h \cap L_h = N_h$ and $J_t \supseteq S_t \cup L_t = N_t$ (note that $J_h \subseteq L_h$ and $J_t \supseteq L_t$).

This concludes the proof of the lemma. □

Proof. (of Theorem 2.1) Our proof is inspired by Jain's proof of [5, Lemma 4.2]. Since $\boldsymbol{x}$ is an extreme point solution (basic solution) with $0 < \boldsymbol{x} < 1$, there exists a set of $|E|$ tight setpairs such that the vectors χ_W corresponding to these setpairs W are linearly independent.

Let $\mathcal{L}$ be an (inclusionwise) maximal noncrossing family of tight setpairs. Let $span(\mathcal{L})$ denote the vector space spanned by the vectors χ_W, $W \in \mathcal{L}$. We will show that $span(\mathcal{L})$ equals the vector space spanned by the vectors χ_Y where Y is any tight setpair. The theorem then follows by taking a basis for $span(\mathcal{L})$ from the set $\{\chi_W \mid W \in \mathcal{L}\}$.

Suppose there is a tight setpair S such that $\chi_S \notin span(\mathcal{L})$. Choose such an S that crosses the minimum number of setpairs in $\mathcal{L}$ (this is a key point). Next, choose any setpair $L \in \mathcal{L}$ such that

S crosses L. By Lemma 2.2,

$$\chi_S = \chi_{S\otimes L} + \chi_{S\oplus L} - \chi_L.$$

Hence, either $\chi_{S\otimes L} \notin span(\mathcal{L})$ or $\chi_{S\oplus L} \notin span(\mathcal{L})$. Suppose the first case holds. (The argument is similar for the other case, and is omitted.) Let $N = S\otimes L = (S_t \cup L_t,\, S_h \cap L_h)$. The next claim follows from Lemma 2.3.

Claim. Any setpair $J \in \mathcal{L}$ that crosses N also crosses S (note that J, L do not cross since both are in $\mathcal{L}$).

Clearly, L does *not* cross N (since $L \succeq N$), but L crosses S. This contradicts our choice of S (since N is a tight setpair that crosses fewer setpairs in $\mathcal{L}$ and $\chi_N \notin span(\mathcal{L})$). □

3. An Edge of High Value in an Extreme Point

This section has the proof of Theorem 1.1. The theorem is proved by representing the noncrossing family $\mathcal{L}$ as a poset and examining the Hasse diagram.

Let $\mathcal{L}$ be a noncrossing family of setpairs. We define the poset $\mathcal{P}$ representing $\mathcal{L}$ as follows. The elements of $\mathcal{P}$ are the setpairs in $\mathcal{L}$ and the relation between elements is the same as the relation between setpairs (for two setpairs W and Y, if $W_t \supseteq Y_t$, $W_h \subseteq Y_h$, then $W \preceq Y$; if $W_t \subseteq Y_t$, $W_h \supseteq Y_h$ then $Y \preceq W$; otherwise they are incomparable). The Hasse diagram of the poset, also denoted by $\mathcal{P}$, is a directed acyclic graph that has a node for each element in the poset, and for elements W, Z there is an arc (W, Z) if $W \preceq Z$ and there is no element Y such that $W \preceq Y \preceq Z$ (the Hasse diagram has no arcs that are implied by transitivity). In the Hasse diagram, an arc (W, Y) indicates that $W_h \subseteq Y_h$ and $W_t \supseteq Y_t$. Throughout, the term *node* refers to the poset $\mathcal{P}$, and the term *vertex* refers to the input digraph G. An *arc* means an arc of $\mathcal{P}$, whereas an *edge* means a directed edge of G. A node Z is called a *predecessor* (or *successor*) of a node W if the arc (Z, W) (or (W, Z)) is present. A directed path in $\mathcal{P}$ is called a *chain*. An *antichain* of $\mathcal{P}$ is a set of nodes that are pairwise incomparable. If C is a chain or an antichain of $\mathcal{P}$, then $|C|$ denotes the number of nodes in C; the number of nodes in $\mathcal{P}$ is denoted by $|\mathcal{P}|$. For an arbitrary poset, define a *diamond* to be a set of four (distinct) elements a, b, c, d such that b, c are incomparable, $a \succeq b \succeq d$ and $a \succeq c \succeq d$. A poset is called *diamond-free* if it contains no diamond. In other words, any two chains of such a poset have at most one subchain in common.

Lemma 3.1. *Let $\mathcal{L}$ be a noncrossing family of setpairs such that each setpair $W \in \mathcal{L}$ has both head and tail nonempty. Then the poset $\mathcal{P}$ representing $\mathcal{L}$ is diamond-free.*

Proof. Suppose that $\mathcal{L}$ has four setpairs W, X, Y, Z such that X, Y are incomparable, $W \succeq X \succeq Z$ and $W \succeq Y \succeq Z$. Since X, Y are incomparable, either they are head disjoint, or tail disjoint. Moreover, $X_h \supseteq Z_h$ since $X \succeq Z$, and $Y_h \supseteq Z_h$ since $Y \succeq Z$. Then X, Y are not head disjoint, since both heads contain the head of Z, which is nonempty. Similarly, it can be seen that X, Y are not tail disjoint, since both tails contain the tail of W, which is nonempty. This contradiction proves that $\mathcal{P}$ contains no diamond. □

We call a node W *unary* if the Hasse diagram has exactly one arc incoming to W and exactly one arc outgoing from W. Consider the maximum cardinality of an antichain in a diamond-free poset $\mathcal{P}$. This may be as small as one, since $\mathcal{P}$ may be a chain. The next result shows that this quantity cannot be so small if $\mathcal{P}$ has no unary nodes.

Proposition 3.2. *(1) If a diamond-free poset $\mathcal{P}$ has no unary nodes, then it has an antichain of cardinality at least $\sqrt{|\mathcal{P}|/2}$.*
(2) If $\mathcal{P}$ is a diamond-free poset such that neither the predecessor nor the successor of a unary node is another unary node, then it has an antichain of cardinality at least $\frac{1}{2}\sqrt{|\mathcal{P}|}$.

Proof. We prove part (1); the proof of part (2) is similar.

If $\mathcal{P}$ has an antichain of cardinality at least $\sqrt{|\mathcal{P}|/2}$, then we are done. Otherwise, by Dilworth's theorem (the minimum number of disjoint chains required to cover all the nodes of a poset equals the maximum cardinality of an antichain), $\mathcal{P}$ has a chain, call it C, with $|C| > |\mathcal{P}|/\sqrt{|\mathcal{P}|/2} = \sqrt{2|\mathcal{P}|}$. Let $C = W_1, W_2, \ldots, W_\ell$. Each of the internal nodes $W_2, \ldots, W_{\ell-1}$ is non-unary, so it has either two predecessors or two successors. Clearly, one of the two predecessors (or one of the two successors) is not in C. Let C_p be the set of nodes in $\mathcal{P} \setminus C$ that are predecessors of nodes in C, and similarly let C_s be the set of nodes in $\mathcal{P} \setminus C$ that are successors of nodes in C. Then either $|C_p| \geq (|C| - 2)/2$ or $|C_s| \geq (|C| - 2)/2$. Suppose that the first case holds (the argument is similar for the other case). Let us add W_1 (the first node of C) to C_p. Now, we claim that C_p is an antichain. Observe that part (1) follows from this claim, because $|C_p| \geq |C|/2 > \sqrt{|\mathcal{P}|/2}$.

To prove that C_p is an antichain, focus on any two (distinct) nodes $Y_i, Y_j \in C_p$. Let W_i and W_j be the nodes in C such that Y_i is the predecessor of W_i and Y_j is the predecessor of W_j. First, suppose that $W_i \neq W_j$, and (w.l.o.g.) assume that $W_i \preceq W_j$. We cannot have $Y_i \preceq Y_j$, otherwise, the nodes W_j, W_{j-1}, Y_j, Y_i will form a diamond, where W_{j-1} is the predecessor of W_j in C (note that the four nodes are distinct, and W_{j-1}, Y_j are incomparable, since both are predecessors of W_j). Also, we cannot have $Y_j \preceq Y_i$, otherwise, we have $Y_j \preceq Y_i \preceq W_i \preceq W_j$ and so the arc (Y_j, W_j)

is implied by transitivity. Hence, Y_i, Y_j are incomparable, if $W_i \neq W_j$. If $W_i = W_j$, then Y_i, Y_j are incomparable (by transitivity). $\square$

We restate Theorem 1.1 for convenience, and present our proof.

Theorem 1.1. *For any digraph $G = (V, E)$, any nonzero extreme point $\boldsymbol{x}$ of (LP) satisfies* $\max_{e \in E}\{x_e\} \geq \dfrac{\Theta(1)}{\sqrt{|E|}}$.

The proof is by contradiction. Let $\boldsymbol{x}$ be an extreme point of (LP), and let $F = \{e \in E \mid x_e > 0\}$. For convenience, assume that no edges e with $x_e = 0$ are present. Also, assume that each edge e has $x_e < 1$, otherwise the proof is done.

Let $\mathcal{L}$ be a noncrossing family of tight setpairs defining $\boldsymbol{x}$ and satisfying the conditions in Theorem 2.1, and let $\mathcal{P}$ be the poset representing $\mathcal{L}$. Note that $\mathcal{P}$ is diamond free (by Lemma 3.1, since each setpair $W \in \mathcal{L}$ has $f(W) \geq 1$, so both W_t, W_h are nonempty), and that $|\mathcal{P}| = |\mathcal{L}| = |F|$. Let U be the set of unary nodes of $\mathcal{P}$, and call a maximal chain of unary nodes a *U-chain*. Let $\mathcal{P}'$ be the "reduced" poset formed by replacing each U-chain by a single unary node. Note that $\mathcal{P}'$ is diamond-free, since $\mathcal{P}$ is diamond-free.

Let C be a maximum-cardinality antichain of $\mathcal{P}$. By Proposition 3.2(2), $|C| \geq \frac{1}{2}\sqrt{|\mathcal{P}'|}$. We may assume that each unary node of C (if any) is a bottom node of a U-chain. By an *upper* U-chain we mean one that has all nodes $\succeq$ some node in C, and by a *lower* U-chain we mean one that has all nodes $\preceq$ some node in C. Let U_0 be the set of bottom nodes of all the upper U-chains together with the set of top nodes of all the lower U-chains. Let U_* be the set of nodes $W \in U \setminus U_0$ in upper U-chains such that the predecessor Y of W has $f(W) = f(Y)$, together with the set of nodes $W \in U \setminus U_0$ in lower U-chains such that the successor Z of W has $f(W) = f(Z)$. Let U_1 be the set of nodes $W \in U \setminus U_0$ in upper U-chains such that the predecessor Y of W has $f(W) > f(Y)$, together with the set of nodes $W \in U \setminus U_0$ in lower U-chains such that the successor Z of W has $f(W) < f(Z)$. Similarly, let U_2 be the set of nodes $W \in U \setminus U_0$ in upper U-chains such that the predecessor Y of W has $f(W) < f(Y)$, together with the set of nodes $W \in U \setminus U_0$ in lower U-chains such that the successor Z of W has $f(W) > f(Z)$.

Clearly,

$$U \;=\; U_0 \,\cup\, U_1 \,\cup\, U_2 \,\cup\, U_* \quad \text{and} \quad |\mathcal{P}| - |\mathcal{P}'| \;=\; |U_1| + |U_2| + |U_*|.$$

Claim. If α is a number such that $x_e < 1/\alpha$, $\forall e \in E$, then

$$|F| \;>\; \alpha \cdot \max\{|C|,\, |U_1|,\, |U_2|\} \;+\; |U_*|.$$

We defer the proof of the claim, and complete the proof of the theorem. Let $\alpha = 4\sqrt{|\mathcal{P}|}$. Suppose that $x_e < 1/\alpha$ for each edge $e \in F$. Then, by the claim,

$$\begin{aligned}
|F| \;&>\; 4\sqrt{|\mathcal{P}|}\cdot \max\{|C|,\, |U_1|,\, |U_2|\} \;+\; |U_*| \\
&\geq\; 4\sqrt{|\mathcal{P}|}\cdot (\frac{1}{2}|C| + \frac{1}{4}|U_1| + \frac{1}{4}|U_2|) \;+\; |U_*| \\
&\geq\; 4\sqrt{|\mathcal{P}|}\cdot (\frac{1}{4}\sqrt{|\mathcal{P}'|} + \frac{1}{4}|U_1| + \frac{1}{4}|U_2|) \;+\; |U_*| \\
&\geq\; |\mathcal{P}'| + |U_1| + |U_2| + |U_*| \\
&\geq\; |\mathcal{P}|.
\end{aligned}$$

This is a contradiction, since $|F| = |\mathcal{P}|$. Hence, there exists an edge e with $x_e \geq 1/\alpha = 1/(4\sqrt{|\mathcal{P}|})$. This proves the theorem.

Proof. (of the Claim) We need to prove the three inequalities separately. Consider the first inequality:

$$|F| > \alpha \cdot |C| \;+\; |U_*|.$$

Each setpair $W \in \mathcal{L}$ has $f(W) \geq 1$, so W is covered by $> \alpha$ edges (otherwise, $x(\delta(W)) < 1$). Hence, each node $W \in \mathcal{P}$ is covered by $> \alpha$ edges. We assign all of the edges covering a node $W \in C$ to W; note that no edge covers two distinct nodes of C. This assigns a total of $> \alpha|C|$ edges. Now, consider a node $W_i \in U_*$ that is in an upper U-chain $W_1, \ldots, W_\ell$, where $1 < i \leq \ell$. Since $f(W_{i-1}) = f(W)$ and $\chi_{W_{i-1}} \neq \chi_{W_i}$, there is an edge in $\delta(W_i) \setminus \delta(W_{i-1})$. We assign this edge to W_i. Similarly, for a node $W_i \in U_*$ in a lower U-chain $W_1, \ldots, W_\ell$, where $1 \leq i < \ell$, we assign to W_i an edge in $\delta(W_i) \setminus \delta(W_{i+1})$. It can be seen that no edge is assigned to two different nodes. Hence, the first inequality follows.

Consider the second inequality: $|F| > \alpha \cdot |U_1| \;+\; |U_*|$. Let $W_i \in U_1$ be any node in an upper U-chain $W_1, \ldots, W_\ell$, where $1 < i \leq \ell$. Since $f(W_i) \geq f(W_{i-1}) + 1$, there must be $> \alpha$ edges in $\delta(W_i) \setminus \delta(W_{i-1})$. We assign all these edges to W_i. Similarly, for a node $W_i \in U_1$ that is in a lower U-chain $W_1, \ldots, W_\ell$, where $1 \leq i < \ell$, we assign $> \alpha$ edges in $\delta(W_{i+1}) \setminus \delta(W_i)$ to W_i. Finally, for nodes $W_i \in U_*$, if W_i is in an upper U-chain $W_1, \ldots, W_\ell$, then we assign to W_i an edge in

$\delta(W_i) \setminus \delta(W_{i-1})$, and if W_i is in a lower U-chain $W_1, \ldots, W_\ell$, then we assign to W_i an edge in $\delta(W_{i+1}) \setminus \delta(W_i)$. The second inequality follows, since no edge is assigned to two different nodes.

The proof of the third inequality is similar to the proof of the second inequality. $\square$

4. A Tight Example

In this section, we present an example of an extreme point $\boldsymbol{x}$ of (LP) such that $0 < x_e \leq \Theta(1)/\sqrt{|E|}$ for all edges $e \in E$. Thus the lower bound in Theorem 1.1 is tight (up to a constant factor). An extreme point $\boldsymbol{x}$ of (LP) is defined by a system of $|E|$ tight constraints, where each is of the form $x(\delta(W)) = f(W)$, for some setpair W (we assume $0 < \boldsymbol{x} < 1$ so the constraints $x_e \geq 0$, $x_e \leq 1$ are redundant). Let $\mathcal{L}$ be the noncrossing family of tight setpairs defining $\boldsymbol{x}$ (see Theorem 2.1), and let $\mathcal{P}$ be the poset (and the Hasse diagram) representing $\mathcal{L}$. Recall that the term *node* refers to $\mathcal{P}$, and an *arc* means an arc of $\mathcal{P}$, whereas an *edge* means a directed edge of G. Each edge $e \in E$ corresponds to a path $p(e)$ in $\mathcal{P}$, where the nodes of $p(e)$ are the setpairs $W \in \mathcal{L}$ that are covered by e, that is, $p(e) = W_1, \ldots, W_\ell$, where $W_1 \preceq \cdots \preceq W_\ell$ and $e \in \delta(W_i)$ $(i = 1, \ldots, \ell)$. We refer to such paths $p(e)$ as *e-paths.*

Let $m = |E(G)|$. Our example is a poset $\mathcal{P}$ with m nodes and m e-paths (see Figure 2), so $|\mathcal{P}| = m$. Let $\mathbf{0}$ and $\mathbf{1}$ denote column vectors with all entries at 0 and 1, respectively, where the dimension of the vector will be clear from the context. Define the incidence matrix A to be an $m \times m$ matrix whose rows correspond to the nodes, and whose columns correspond to the e-paths, such that the entry for node W and e-path p, A_{Wp}, is 1 if W is in p and is 0 otherwise. We will prove that A has rank $m-1$ and the system $A\boldsymbol{x} = \mathbf{1}$ has a solution where each entry of $\boldsymbol{x}$ is $\Theta(1)/\sqrt{m}$. (Note that $\boldsymbol{x}$ assigns a real number to each of the e-paths, and it corresponds to a solution of the LP.)

The poset $\mathcal{P}$ consists of several copies of the following *path structure* Q. Let t be a parameter (we will fix $t = \sqrt{m/12}$), and let there be $3t$ nodes $1, 2, \ldots, 3t$. Then Q consists of a path $[1, \ldots, 3t]$ on these nodes, together with $2t$ *local e-paths*, call them $p_1, \ldots, p_{2t}$, where each p_j is a subpath of the path $[1, \ldots, 3t]$. For odd j $(j = 1, 3, 5, \ldots, 2t-1)$, p_j consists of the first j nodes (so $p_j = [1, \ldots, j]$), and for even j $(j = 2, 4, 6, \ldots, 2t)$, p_j consists of all the nodes, except the first $j-2$ nodes (so $p_j = [j-1, j, \ldots, 3t]$). Call the nodes $1, 3, 5, \ldots, 2t-1$ the *black* nodes, the nodes $2, 4, 6, \ldots, 2t-2$ the *white* nodes, and the remaining nodes $2t, 2t+1, \ldots, 3t$ the *red* nodes. Note that each black node is incident to $t+1$ local e-paths, and each of the other nodes is incident to t local e-paths.

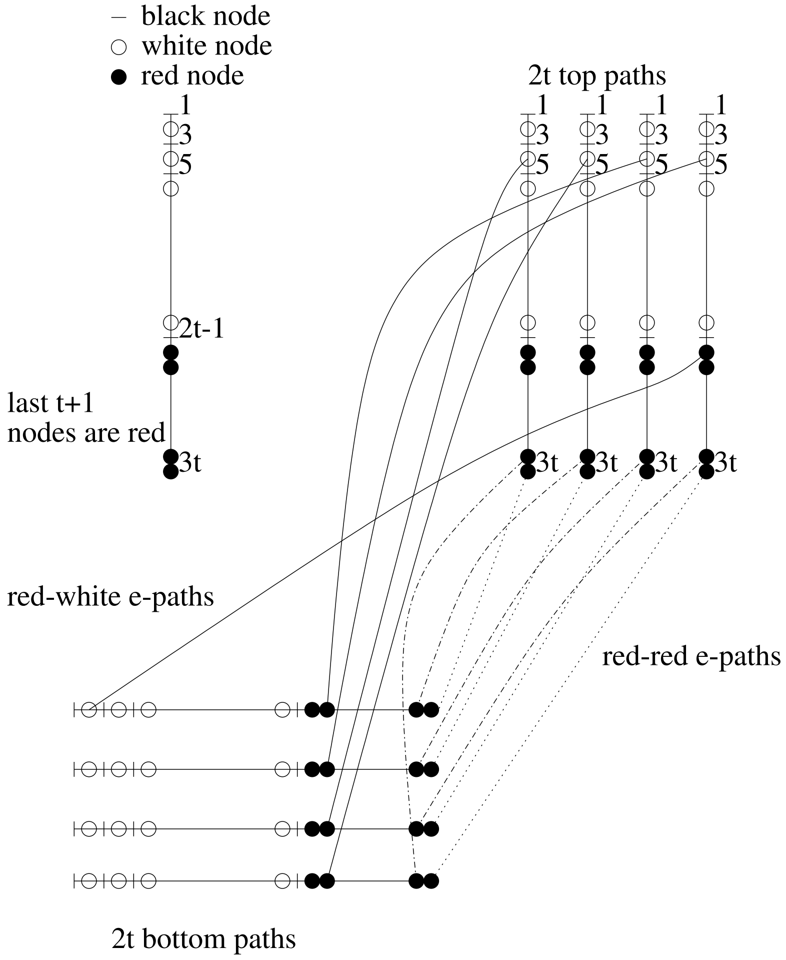

FIGURE 2. Illustration of the poset $\mathcal{P}$.

We take $4t$ copies of Q, and partition them into two sets, the *top* paths $T_1, \ldots, T_{2t}$, and the *bottom* paths $B_1, \ldots, B_{2t}$. (In fact, each T_i or B_i is a path structure consisting of a path and $2t$ local e-paths, but we call them paths for convenience.) Finally, we add another $4t^2$ *nonlocal e-paths* such that the following conditions hold:

- each node is incident to a total of $t+1$ e-paths;
- each nonlocal e-path is incident to exactly two nodes, one in a top path T_i and one in a bottom path B_j; moreover, for every T_i and every B_j, there is exactly one nonlocal e-path incident to both T_i and B_j;
- each nonlocal e-path is incident to either two red nodes, or one red node and one white node;
- each top/bottom path T_i or B_j is incident to exactly two red-red nonlocal e-paths, where (i) there is an e-path incident to the last node of B_i and the last node of T_i ($i = 1, \ldots, 2t$), and

(ii) there is an e-path incident to the 2nd last node of B_i and the 2nd last node of T_{i+1} $(i = 1, \ldots, 2t)$; the indexing is modulo $2t$, so $2t+1$ means 1; note that there is cyclic shift by 1 in the index of the top versus bottom paths;

- the red-white nonlocal e-paths are fixed according to the first two conditions, and are as follows: for $\ell = 1, 2, \ldots, t-1$, there is an e-path incident to the 2ℓth node of B_i and the $(2t-1+\ell)$th node of $T_{i+1+\ell}$ $(i = 1, \ldots, 2t)$, indexing modulo $2t$; note that there is a cyclic shift by $\ell+1$ in the index of the top versus bottom paths; similarly, for $\ell = 1, 2, \ldots, t-1$, there is an e-path incident to the 2ℓth node of T_i and the $(2t-1+\ell)$th node of $B_{i+1+\ell}$ $(i = 1, \ldots, 2t)$, indexing modulo $2t$.

Proposition 4.1. *Let t be a positive integer, and let $m = 12t^2$. Let A be the $m \times m$ incidence matrix of the poset $\mathcal{P}$ and the e-paths (constructed above). Then* $\operatorname{rank}(A) \geq m-1$ *and a solution to the system $A\boldsymbol{x} = \mathbf{1}$ is given by $\boldsymbol{x} = \frac{1}{t+1} \cdot \mathbf{1}$.*

Proof. A column vector of dimension ℓ with all entries at 0 (or, 1) is denoted by 0_ℓ (or, 1_ℓ). Let e_i denote the ith column of the $s \times s$ identity matrix I_s, where s is a positive integer. Let f_i denote $\sum_{j=1}^{i} e_j$; so f_i is a column vector with a 1 in entries $1, \ldots, i$ and a 0 in entries $i+1, \ldots, s$.

Let the rows of A be ordered according to the nodes $1, \ldots, 3t$ of $T_1, \ldots, T_{2t}$, followed by the nodes $1, \ldots, 3t$ of $B_1, \ldots, B_{2t}$.

First, consider a bottom path B_i; top paths T_i are handled similarly, and this is sketched later.

Let M denote the incidence matrix of B_i versus all the e-paths. Then M is $3t \times m$ matrix, where the rows $1, \ldots, 3t$ correspond to the nodes $1, \ldots, 3t$ of B_i, and the columns of M are ordered as follows:

- the $2t$ local e-paths of B_i, $p_1, p_2, \ldots, p_{2t}$,
- the $t-1$ red-white e-paths whose red ends are in B_i (these are the e-paths incident to nodes $2t, 2t+1, \ldots, 3t-2$ of B_i),
- the two red-red e-paths incident to nodes $3t-1$ and $3t$ of B_i,
- the remaining e-paths (among these are $t-1$ red-white e-paths whose white ends are in B_i).

Let M^{beg} denote the submatrix of M formed by the first $2t$ columns, so M^{beg} is the incidence matrix of the nodes versus the local e-paths of B_i. Let M^{end} denote the submatrix of M formed by excluding the first $3t+1$ columns (keeping only the columns of the "remaining e-paths"). Then

$$M \;=\; \left[\begin{array}{c|c|cc|c} & 0_{2t-1}\ldots 0_{2t-1} & 0_{2t-1} & 0_{2t-1} & \\ M^{beg} & I_{t-1} & 0_{t-1} & 0_{t-1} & M^{end} \\ & 0\ldots 0 & 1 & 0 & \\ & 0\ldots 0 & 0 & 1 & \end{array}\right].$$

Note that the rows and columns of M^{end} may be reordered such that the submatrix in the first $t-1$ rows (make these the rows of the white nodes of B_i) and the first $t-1$ columns (make these the columns of the red-white e-paths incident to the white nodes of B_i) is the identity matrix I_{t-1}, and every other entry of the matrix is zero. Then

$$M^{beg} = [f_1,\ \mathbf{1},\ f_3,\ \mathbf{1}-f_2,\ f_5,\ \mathbf{1}-f_4,\ \ldots,\ f_{2t-1},\ \mathbf{1}-f_{2t-2}].$$

Using elementary column operations, we can rewrite this matrix as

$$[e_1, e_2, e_3, \ldots, e_{2t-1}, \mathbf{1}-f_{2t-1}] \;=\; \begin{bmatrix} I_{2t-1} & 0_{2t-1} \\ 0_{t+1}\ldots 0_{t+1} & 1_{t+1} \end{bmatrix}.$$

Then it is clear that the matrix $[M^{beg}\ M^{end}]$ may be rewritten using elementary column operations as

$$\begin{bmatrix} I_{2t-1} & 0_{2t-1} & \\ 0_{t+1}\ldots 0_{t+1} & 1_{t+1} & \mathbf{0}\ldots\mathbf{0} \end{bmatrix}.$$

Going back to M, observe that it may be rewritten using elementary column operations as

$$\left[\begin{array}{ccc|cc|c} I_{2t-1} & 0_{2t-1} & 0_{2t-1}\ldots 0_{2t-1} & 0_{2t-1} & 0_{2t-1} & \\ 0_{t-1}\ldots 0_{t-1} & 1_{t-1} & I_{t-1} & 0_{t-1} & 0_{t-1} & \mathbf{0}\ldots\mathbf{0} \\ 0\ldots 0 & 1 & 0\ldots 0 & 1 & 0 & \\ 0\ldots 0 & 1 & 0\ldots 0 & 0 & 1 & \end{array}\right],$$

or as

$$M^* \;=\; \left[\begin{array}{cc|ccc|c} I_{2t-1} & 0_{2t-1}\ldots 0_{2t-1} & 0_{2t-1} & 0_{2t-1} & 0_{2t-1} & \\ 0_{t-1}\ldots 0_{t-1} & I_{t-1} & 0_{t-1} & 0_{t-1} & 1_{t-1} & \mathbf{0}\ldots\mathbf{0} \\ 0\ldots 0 & 0\ldots 0 & 1 & 0 & 1 & \\ 0\ldots 0 & 0\ldots 0 & 0 & 1 & 1 & \end{array}\right].$$

Now, focus on the matrix A (the incidence matrix of the nodes of $\mathcal{P}$ versus all the e-paths), and its column vectors. Consider the two red-red e-paths incident to B_i and their column vectors in A. Let r_{3t-1} and r_{3t} denote the two red-red e-paths incident to the red nodes $3t-1$ and $3t$ (of B_i), respectively, and let the column vectors in A of these red-red e-paths also be denoted by the same symbols. Note that r_{3t} has two nonzero entries, namely, a 1 for node $3t$ of B_i and a 1 for node $3t$ of T_i. Similarly, r_{3t-1} has two nonzero entries, namely, a 1 for node $3t-1$ of B_i and a 1 for node $3t-1$ of T_{i+1} (indexing modulo $2t$). Keep the column r_{3t-1}, but use elementary column operations to replace r_{3t} by $r'_{3t} = r_{3t} + r_{3t-1} + M^*_{2t} + \cdots + M^*_{3t-2} - M^*_{3t+1}$, where M^*_j denotes the column vector of dimension m obtained from the jth column vector of M^* by padding with zeros (fixing entries of rows not in M^* at 0). Clearly, r'_{3t} has two nonzero entries, namely a 1 for node $3t$ of T_i and a 1 for node $3t-1$ of T_{i+1} (indexing modulo $2t$).

Let $M^{bot,i}$ be the $m \times m$ matrix obtained from M^* by replacing the $(3t-1)$th column by r_{3t-1}, deleting the $3t$th column, and padding with zeros (fixing at 0 all entries except those in M^* or in the $(3t-1)$th column).

The construction for a top path T_{i+1} ($i = 1, 2, \ldots, 2t$, indexing modulo $2t$) is similar, except for the handling of columns $3t-1$ and $3t$ of M^*. Let $M^{top,i+1}$ be the $m \times m$ matrix obtained from M^* by replacing the $3t$th column by r'_{3t}, deleting the $(3t-1)$th column, and padding with zeros.

Let A^* be the $m \times m$ matrix obtained from A by elementary column operations, where

$$A^* = \sum_{i=1}^{2t} M^{top,i} + \sum_{i=1}^{2t} M^{bot,i}.$$

Figure 3 illustrates the zero-nonzero pattern of A^*.

With the exception of one entry, A^* is an upper triangular matrix with every diagonal entry at 1; the exceptional entry is in row $6t$ (node $3t$ of T_{2t}) and column $3t-1$ (2nd last column of T_1). Then, deleting row $6t$ and column $6t$ of A^* we get an upper triangular matrix with determinant 1. This proves that $rank(A) \geq m-1$.

By construction, each node of $\mathcal{P}$ is incident to exactly $t+1$ e-paths, hence fixing $x_e = 1/(t+1)$ for each e-path $p(e)$ gives a solution to the system $A\boldsymbol{x} = \mathbf{1}$. The proposition follows. □

Proposition 4.2. *Let P be the polytope $\{\boldsymbol{x} \in \mathbb{R}^m \mid \widetilde{A}\boldsymbol{x} \leq \widetilde{b},\ 0 \leq \boldsymbol{x} \leq 1\}$ and let F be the face $\{\boldsymbol{x} \in P \mid A\boldsymbol{x} = b\}$, where $A\boldsymbol{x} \leq b$ is a subsystem of $\widetilde{A}\boldsymbol{x} \leq \widetilde{b}$. If the matrix A has rank $m-1$ and there exists an $\boldsymbol{x} \in F$ such that each entry of $\boldsymbol{x}$ is $\leq \alpha$, then P has an extreme point $\widetilde{\boldsymbol{x}}$ such that each entry of $\widetilde{\boldsymbol{x}}$ is $\leq 2\alpha$.*

Proof. F is a line-segment and so it has two extreme points, call them $\boldsymbol{y}$ and $\boldsymbol{z}$. Note that $\boldsymbol{y}$ and $\boldsymbol{z}$ must be extreme points of P also. Hence, $\boldsymbol{x} = a \cdot \boldsymbol{y} + (1-a) \cdot \boldsymbol{z}$, where $0 \leq a \leq 1$. Suppose $a \geq 1/2$

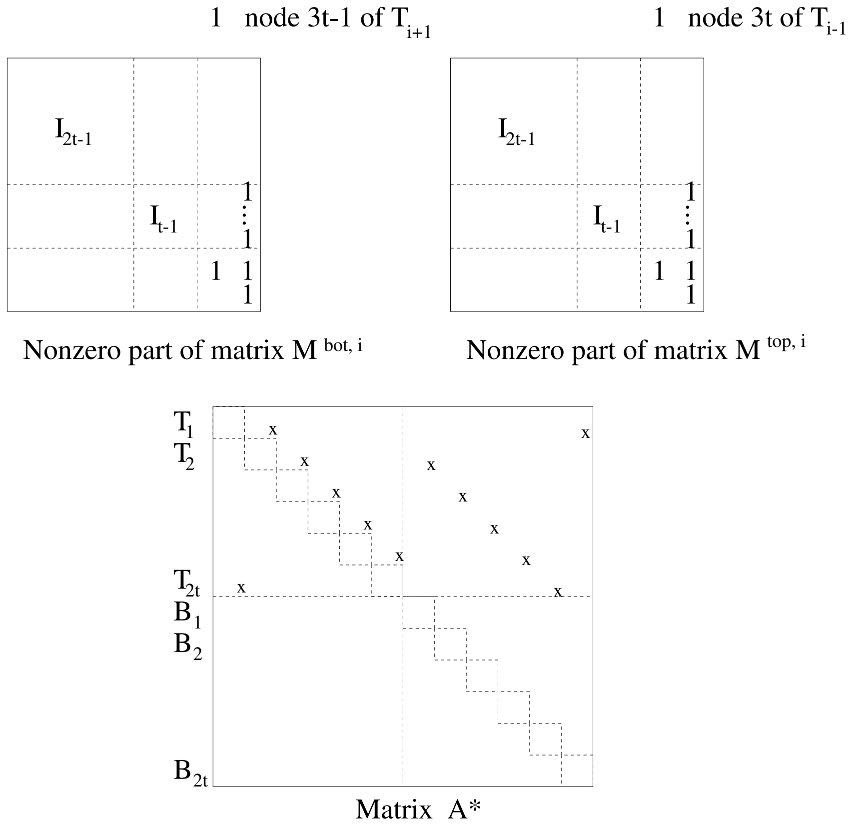

FIGURE 3. Illustration of matrices $M^{bot,i}$ and $M^{top,i}$, and the nonzero pattern of matrix A^*.

(the other case is similar). Then $\boldsymbol{y} \leq 2(\boldsymbol{x} - (1-a) \cdot \boldsymbol{z}) \leq 2\boldsymbol{x}$, since $\boldsymbol{z} \geq 0$, so each entry of $\boldsymbol{y}$ is $\leq 2\alpha$. □

Theorem 1.2, which is the main result of this section, follows from Propositions 4.1 and 4.2.

Theorem 1.2. *Given any sufficiently large integer $|E|$, there exists a digraph $G = (V, E)$ such that (LP) has an extreme point $\boldsymbol{x}$ satisfying* $\max_{e \in E}\{x_e\} \leq \dfrac{\Theta(1)}{\sqrt{|E|}}$.

5. Conclusion

In conclusion, we mention that our framework and the results in sections 2 and 3 lead to interesting approximation guarantees for specific problems in network design, see [2].

References

[1] A. Agrawal, P. N. Klein and R. Ravi, "When trees collide: An approximation algorithm for the generalized Steiner problem on networks," *SIAM J. Computing*, **24**, 1995. Preliminary version in *Proc. 23rd ACM STOC*, pp. 134–144, 1991.

[2] J. Cheriyan, S. Vempala, and A. Vetta, "Setpair relaxations for network design I, II," manuscript, submitted for journal publication, August 2001.

[3] T. Fleiner, "Note - Uncrossing a family of set-pairs," *Combinatorica* **21**(1), pp. 145–150, 2001.

[4] A. Frank and T. Jordan, "Minimal edge-coverings of pairs of sets," *J. Combinatorial Theory, Series B*, **65**, pp. 73–110, 1995.

[5] K. Jain, "A factor 2 approximation algorithm for the generalized Steiner network problem", *Combinatorica*, **21**(1), pp. 39–60, 2001. Preliminary version in *Proc. 39th IEEE Foundations of Computer Science*, 1998.

[6] M. Goemans, A. Goldberg, S. Plotkin, D. Shmoys, E. Tardos and D. Williamson, "Improved approximation algorithms for network design problems," in *Proc. ACM SIAM Symposium on Discrete Algorithms*, 1994, pp. 223–232.

[7] M. Goemans and D. Williamson, "A general approximation technique for constrained forest problems," *SIAM Journal on Computing*, **24**, pp. 296–317, 1995.

[8] M. Grötschel, L. Lovász and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, New York, 1988.

[9] V. Melkonian and E. Tardos, "Approximation algorithms for a directed network design problem," In the *Proceedings of the 7th International Integer Programming and Combinatorial Optimization Conference* (IPCO'99), Graz, Austria, 1999.

[10] A. Schrijver, "Matroids and linking systems," *J. Combinatorial Theory, Series B*, **26**, pp. 349–369, 1979.

[11] E. Tardos, "A strongly polynomial algorithm to solve combinatorial linear programs," *Operations Research*, **34**, pp. 250–256, 1986.

[12] D. Williamson, M. Goemans, M. Mihail, and V. Vazirani, "A primal-dual approximation algorithm for generalized Steiner network problems," *Combinatorica*, **15**, pp. 435–454, 1995.

Joseph Cheriyan, Dept. of Comb. and Opt., Univ. Waterloo, Waterloo, ON, Canada, N2L 3G1.
Supported in part by NSERC research grant OGP0138432.

E-mail address: `jcheriyan@math.uwaterloo.ca.`

Santosh Vempala, Department of Mathematics, M.I.T., Cambridge, MA 02139, USA.
Supported in part by NSF Career Award CCR-9875024.

E-mail address: `vempala@math.mit.edu`