NETWORK DESIGN VIA ITERATIVE ROUNDING OF SETPAIR RELAXATIONS

MAY, 2001 REVISED: NOVEMBER 2, 2001

JOSEPH CHERIYAN, SANTOSH VEMPALA, AND ADRIAN VETTA

ABSTRACT. A typical problem in *network design* is to find a minimum-cost sub-network H of a given network G such that H satisfies some prespecified connectivity requirements. Our focus is on approximation algorithms for designing networks that satisfy *vertex connectivity* requirements. Our main tool is a linear programming relaxation of the following *setpair formulation* due to Frank and Jordan: a *setpair* consists of two subsets of vertices (of the given network G); each setpair has an integer requirement, and the goal is to find a minimum-cost subset of the edges of G such that each setpair is covered by at least as many edges as its requirement. We introduce the notion of *skew bisupermodular* functions and use it to prove that the basic solutions of the linear program are characterized by "non-crossing families" of setpairs. This allows us to apply Jain's *iterative rounding method* to find approximately optimal integer solutions. We give two applications. (1) In the *k-vertex connectivity problem* we are given a (directed or undirected) graph $G = (V, E)$ with nonnegative edge costs, and the task is to find a minimum-cost spanning subgraph H such that H is k-vertex connected. Let $n = |V|$, and let $\epsilon < 1$ be a positive number such that $k \leq (1-\epsilon)n$. We give an $O(\sqrt{n/\epsilon})$-approximation algorithm for both problems (directed or undirected), improving on the previous best approximation guarantees for k in the range $\Omega(\sqrt{n}) < k < n - \Omega(1)$. (2) We give a 2-approximation algorithm for the *element connectivity* problem, matching the previous best approximation guarantee due to Fleischer, Jain and Williamson.

1. INTRODUCTION

A typical problem in network design is to find a minimum-cost sub-network H of a given network G such that H satisfies some prespecified connectivity requirements; examples are the *minimum spanning tree* (MST) problem and the *traveling salesman problem* (TSP). By a *network* we mean a graph (either undirected or directed) together with non-negative costs for the edges. Problems in network design have a central position in combinatorial optimization and theoretical computer science, and moreover, they arise in many practical settings (e.g., telephone networks). Many of the problems in network design are NP-hard. Over the past decade, there has been significant research

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.

on approximation algorithms for network design, and there have been some notable successes in the design of networks that satisfy various types of "edge connectivity" requirements, e.g., Goemans and Williamson [9], Jain [11], etc. Fewer results are known on the design of networks subject to "vertex connectivity" requirements. There are several recent results for restricted edge costs, such as uniform costs or metric costs, but in this paper we discuss only the general case of non-negative costs. Our focus is on approximation algorithms for designing networks that satisfy vertex connectivity requirements. Our results are based on the iterative rounding method (due to Jain [11]) and on an integer programming formulation called the *setpair formulation* (due to Frank and Jordan [5]).

Let $G = (V, E)$ be the graph of the given network, and let n denote the number of vertices. Most of the previous research in this area is based upon an integer programming formulation called the *cut covering* model. A non-negative, integer requirement $f(S, V \setminus S)$ is assigned to each vertex partition $(S, V \setminus S)$, and the goal is to find a minimum-cost subgraph H such that each cut $(S, V \setminus S)$ has at least $f(S, V \setminus S)$ edges of H. (For example, in the MST problem, $f(S, V \setminus S) = 1$ for each nonempty vertex set S, $S \neq V$.) Although this model captures many "edge connectivity" problems in network design, it seems less suited for "vertex connectivity" problems.

The *iterative rounding* method works as follows. Formulate the problem as an integer program, and then solve the LP (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 LP and the integer program, since the variable x_{e^*} is implicitly fixed at value 1. In detail, decrease by 1 the r.h.s. of every constraint where the variable x_{e^*} occurs, and then remove this variable from the LP. The resulting LP is the same as the LP 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. Iteratively solve the reduced problem. Jain [11] applied this method to the cut covering model, 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 proof is based on a key property of the LP: 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 the LP is characterized by a laminar family of "tight sets."

The setpair formulation [5] is a generalization of the cut covering model. A setpair (W_t, W_h) consists of two disjoint vertex sets W_t and W_h, i.e., $W_t \subseteq V$, $W_h \subseteq V$, $W_t \cap W_h = \emptyset$. Each setpair (W_t, W_h) is assigned an non-negative, integer requirement $f(W_t, W_h)$. The goal is to find a minimum-cost subgraph H that satisfies the requirement of every setpair, i.e., for each setpair (W_t, W_h), H should have at least $f(W_t, W_h)$ edges that have one end-vertex in W_t and the other end-vertex in W_h. Note that the cut covering model is a special case of the setpair formulation in which every setpair (W_t, W_h) with a positive requirement has $W_t = V \setminus W_h$ (i.e., a pair of complementary sets).

We introduce a special class of requirement functions f, namely, *skew bisupermodular* functions (see Section 2). These functions are a common generalization of weakly supermodular (symmetric) functions ([3], [8]) and crossing bisupermodular functions [5]. Skew bisupermodular requirement functions are useful because the extreme points of the LP relaxation are characterized by "non-crossing families" of setpairs (see Section 3 and Theorem 3.3). Note the correspondence with the cut covering model, where the extreme points of the LP relaxation may be characterized by a laminar family of sets. Based on this, we obtain lower bounds on the maximum weight of an edge for the basic solutions of the LP relaxation. Also, this LP relaxation has the self-reducibility property needed for the iterative rounding method. Hence, these structural results immediately give approximation algorithms.

We give applications to two specific problems. In the *k-vertex connectivity problem* we are given a graph $G = (V, E)$ (either directed or undirected) with nonnegative edge costs $c : E \to \mathbb{R}_+$, and the task is to find a minimum-cost spanning subgraph H such that H is k-vertex connected. A graph is called *k-vertex connected* if it has at least $k + 1$ vertices, and the deletion of any set of $k - 1$ vertices leaves a connected graph. We give an approximation algorithm for the undirected problem, and our method extends to the directed problem. For both the directed and the undirected problems, our results improve on the previous best approximation guarantees provided that k is in the range $\Omega(\sqrt{n}) < k < n - \Omega(1)$. The previous best approximation guarantee for the directed problem was $\Omega(k)$ [6], and the previous best approximation guarantee for the undirected problem was $O(\log k)$ for $k \le \sqrt{\frac{n}{6}}$ [2], and $\Omega(k)$ for $k > \sqrt{\frac{n}{6}}$ [13]. (An $O(\log k)$ approximation guarantee for the undirected problem was claimed earlier in [14], but subsequently an error has been found and that claim has been withdrawn.)

In the *element connectivity* problem [12], there is a set of *terminal* vertices $T \subseteq V$ and between each pair of distinct terminals i and j there is a connectivity requirement r_{ij} (each r_{ij} is a non-negative integer). Terminals are reliable and do not fail. Edges and non-terminal vertices are unreliable and are subject to failure. The edges and non-terminals are called *elements*. The problem is to find a minimum-cost subgraph H that contains r_{ij} element disjoint paths between each pair of terminals i and j. Alternatively, for every pair of terminals i and j, H should contain a path between i and j despite the failure of up to $r_{ij} - 1$ elements. The problem is NP-hard. Recently, Fleischer, Jain, and Williamson [7] gave a 2-approximation algorithm for this problem, improving on the previous best approximation guarantee of $2\ln(\max\{r_{ij}\})$ due to Jain et al [12]. We give a different proof of the same 2-approximation guarantee. The element connectivity problem is a generalization of the Steiner network problem, see Goemans et al [8] and Jain [11]. Even for rather special cases of this problem, the best approximation guarantee known is 2, and moreover, the LP relaxation has an integrality ratio of at least $2 - \frac{2}{n}$.

For many of the specific problems in network design that are captured by the setpairs formulation, the linear programming relaxation can be written as a compact linear program via a "flow formulation." Moreover, an appropriate optimal solution can be computed in strongly polynomial time via Tardos' algorithm [15]. The computed optimal solution may not be basic, but it will have an edge of sufficiently large weight. This applies to both the k-vertex connectivity problem and the element connectivity problem, and is similar to the method used by Jain in [11, Section 9].

Let us briefly comment on the relation of this work to [1, 7, 11]. The first two authors [1] analyzed the iterative rounding method [11] for setpair relaxations on directed graphs using crossing bisupermodular functions [5], during January–October 2000. As a follow-up to this, we came up with skew bisupermodular functions and our combinatorial characterization of the extreme points of the LP in October 2000. To the best of our knowledge, this was before Fleischer et al [7] started their collaboration. Our approximation guarantees for the two problems (k-vertex connectivity and element connectivity) were obtained during March–May 2001, and this was after Fleischer et al obtained their result for the element connectivity problem. We use the same linear programming relaxation for the element connectivity problem as Fleischer et al, but the two proofs are different. In particular, the definitions of "noncrossing setpairs" are different. In our setting, noncrossing setpairs may be tail disjoint, or head disjoint, or comparable (this follows [1, 5]), whereas [7] has a more restricted notion of noncrossing setpairs. We have included a proof of the 2-approximation guarantee for element connectivity, since the result of [7] does not imply that this approximation guarantee can be derived in our setting. Summarizing, our goal is to present general results on the iterative rounding of setpair relaxations, to model diverse problems in network design, and to show that despite the generality, this method gives the best approximation guarantees known for specific problems.

The rest of this paper is organized as follows. Section 2 has notation and definitions pertaining to the setpair formulation. Section 3 contains our main result on skew bisupermodular functions, a combinatorial characterization of the extreme points of the LP. Section 4 gives the approximation algorithm for the k-vertex connectivity problem. The concluding section, Section 5 describes a setpair formulation for the element connectivity problem, and gives our proof of the 2-approximation guarantee.

2. The Setpair Formulation and Skew Bisupermodular Functions

A *setpair* $W = (W_t, W_h)$ is an ordered pair of disjoint vertex sets; either W_t or W_h may be empty. We say that W_t is the *tail* of W, and W_h is the *head* of W. Let $\mathcal{S}$ denote the set of all setpairs. Denote by $\delta(W) = \delta(W_t, W_h)$ the set of edges with one end-vertex in W_t and the other in W_h; note that a *cut* $\delta(W_t, V \setminus W_t)$ is the special case of $\delta(W)$ where the head and tail of W form a complementary pair of vertex sets.

The setpair formulation (discussed in Section 1) may be written as the following integer program (IP).

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

Here, f is some non-negative integer function on the setpairs, and $c : E \to \mathbb{R}_+$ gives the edge costs. Let (LP) denote the linear programming relaxation obtained from (IP) by replacing the constraints $x_e \in \{0,1\}$ by $0 \leq x_e \leq 1$ for all edges $e \in E$..

2.1. Crossing Setpairs.

Given two setpairs $W = (W_t, W_h)$ and $Y = (Y_t, Y_h)$, 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 $(W_t \cap Y_t,\, W_h \cup Y_h)$.

Two setpairs W and Y are said to be *comparable* if

(i) $W_t \supseteq Y_t$ and $W_h \subseteq Y_h$, or (ii) $W_t \subseteq Y_t$ and $W_h \supseteq Y_h$.

The former case is denoted by $W \preceq Y$ and the latter case by $W \succeq Y$. Note that $W \preceq W \oplus Y$ and $W \succeq W \otimes Y$.

Two setpairs W and Y are said to be *non-crossing* if

(i) they are comparable, or (ii) their tails are disjoint, or (iii) their heads are disjoint.

Otherwise, W and Y are said to *cross*. Note that W and Y cross if and only if their tails intersect, their heads intersect and they are not comparable.

For a setpair $W = (W_t, W_h)$, let $\overline{W} = (W_h, W_t)$ be the *reverse setpair*. Observe that $\delta(W) = \delta(\overline{W})$ for an undirected graph. Also, note that $\overline{W \otimes Y} = \overline{W} \oplus \overline{Y}$.

Two setpairs W and Y are said to be *overlapping* if

(i) W and Y cross, or (ii) $\overline{W}$ and Y cross (which is the same as W and $\overline{Y}$ cross).

Otherwise, the setpairs are called *non-overlapping*. A family of setpairs $\mathcal{L} \subseteq \mathcal{S}$ is called non-crossing (non-overlapping) if no two setpairs in $\mathcal{L}$ cross (overlap).

2.2. **Bisubmodular Functions.** A real-valued function f on $\mathcal{S}$ 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 $\mathcal{S}$ 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 edge 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 $\boldsymbol{x} : E \to \mathbb{R}_+$ 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 $\sum_{i \in Q} x_i$.)

2.3. **Skew Bisupermodular Functions.** A non-negative, integer-valued function f on $\mathcal{S}$ is called *skew bisupermodular* if for any two overlapping setpairs W and Y with $f(W) > 0$ and $f(Y) > 0$, we have

(i) $f(W) + f(Y) \leq f(W \otimes Y) + f(W \oplus Y)$, or (ii) $f(W) + f(Y) \leq f(\overline{W} \otimes Y) + f(\overline{W} \oplus Y)$.

This is motivated by problems on undirected graphs. For these problems, it turns out that the relevant functions f are symmetric, i.e., $f(W) = f(\overline{W})$, $\forall W \in \mathcal{S}$.

Note that every crossing bisupermodular function is skew bisupermodular. Also, every weakly supermodular symmetric function is skew bisupermodular (see the discussion at the start of Section 5). Thus the skew bisupermodular property is a generalization of these earlier notions.

3. Characterizing a Basic Solution via a Non-Overlapping Family

The main result of this section, Theorem 3.3, characterizes a basic solution of (LP) via a non-overlapping family of setpairs, assuming that the requirement function f is skew bisupermodular, and the given graph G is undirected. This result is an extension of classic results to our setting. Our proof is similar to earlier proofs in [11, Lemma 4.2] and [1, Theorem 2.1]; however, when we add a new setpair W to our non-overlapping family, we have to ensure that neither W nor $\overline{W}$ crosses a setpair in the family.

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

Lemma 3.1. *Let the requirement function f of (LP) be skew bisupermodular, and let $\boldsymbol{x}$ be a feasible solution to (LP) such that $x_e > 0$ for all edges $e \in E$. Suppose that the setpairs W and Y have $f(W) > 0, f(Y) > 0$, and moreover, W and Y overlap, and are tight (also, note that $\overline{W}$ is tight, it overlaps Y, and $f(\overline{W}) > 0$). Then one of the following holds:*

(i) The setpairs $W \otimes Y$ and $W \oplus Y$ are tight, and $\chi_W + \chi_Y = \chi_{W \otimes Y} + \chi_{W \oplus Y}$.

(ii) The setpairs $\overline{W} \otimes Y$ and $\overline{W} \oplus Y$ are tight, and $\chi_W + \chi_Y = \chi_{\overline{W} \otimes Y} + \chi_{\overline{W} \oplus Y}$.

Proof. Assume that $f(W) + f(Y) \leq f(W \otimes Y) + f(W \oplus Y)$. The other case is similar. Then

$$\begin{aligned} f(W \otimes Y) + f(W \oplus Y) &\leq x(\delta(W \otimes Y)) + x(\delta(W \oplus Y)) \\ &\leq x(\delta(W)) + x(\delta(Y)) \\ &= f(W) + f(Y) \\ &\leq f(W \otimes Y) + f(W \oplus Y) \end{aligned}$$

Hence, all the inequalities hold as equations, therefore, $W \otimes Y$ and $W \oplus Y$ are both tight. This proves the first part of (i). The second part of (i) follows from the following three facts: $\chi_W + \chi_Y \geq \chi_{W \otimes Y} + \chi_{W \oplus Y}$; $x_e > 0$ for all edges e; $x(\delta(W \otimes Y)) + x(\delta(W \oplus Y)) = x(\delta(W)) + x(\delta(Y))$. □

The next result is in [1, Lemma 2.3] and is used to prove the main result of this section.

Lemma 3.2. *Let W, Y and Z be setpairs. If Z crosses $W \otimes Y$ (or $W \oplus Y$) then either Z crosses W or Z crosses Y.* □

Theorem 3.3. *Let the requirement function f of (LP) be skew bisupermodular, and let $\boldsymbol{x}$ be a basic solution to (LP) such that $0 < x_e < 1$ for all edges $e \in E$. Then there exists a non-overlapping family $\mathcal{L}$ of tight setpairs 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.*

(iv) $\boldsymbol{x}$ *is the unique solution to* $\{x(\delta(W)) = f(W), \forall W \in \mathcal{L}\}$.

Proof. Since $\boldsymbol{x}$ is a basic solution of (LP), and none of the constraints $0 \le x_e \le 1$ $(e \in E)$ holds with equality, there exist $|E|$ tight setpair constraints $x(\delta(W)) = f(W) > 0$ such that the edge incidence vectors χ_W corresponding to these constraints are linearly independent. Let $\mathcal{L}$ be a maximal, non-overlapping family of tight setpairs and let $span(\mathcal{L})$ denote the vector space spanned by the vectors χ_W, where $W \in \mathcal{L}$. The theorem holds because $span(\mathcal{L})$ equals $span(\mathcal{F})$, where $\mathcal{F}$ is the family of all tight setpairs. We prove this by contradiction. Suppose that there is a tight setpair Y such that $\chi_Y \notin span(\mathcal{L})$. Take such a Y that overlaps the minimum number of setpairs in $\mathcal{L}$ (this is a key step). Choose any setpair $W \in \mathcal{L}$ that overlaps Y; there exists such a W by the maximality of $\mathcal{L}$.

Either part (i) or part (ii) in the statement of Lemma 3.1 holds for W and Y. Assume that part (ii) holds. The other case is similar. Thus, the setpairs $\overline{W} \otimes Y$ and $\overline{W} \oplus Y$ are tight. In addition, $\chi_W + \chi_Y = \chi_{\overline{W} \otimes Y} + \chi_{\overline{W} \oplus Y}$. It follows that either $\chi_{\overline{W} \otimes Y} \notin span(\mathcal{L})$ or $\chi_{\overline{W} \oplus Y} \notin span(\mathcal{L})$. Consider the former case. Again, the other case is similar. Let $Q = \overline{W} \otimes Y$ and take any setpair $Z \in \mathcal{L}$. Suppose that Z overlaps Q. Then either Z crosses Q or Z crosses $\overline{Q} = \overline{\overline{W} \otimes Y} = W \oplus \overline{Y}$. Suppose that Z crosses Q. We apply Lemma 3.2 to Z and Q, noting that Z does not cross $\overline{W}$, to see that Z crosses Y. Similarly, if Z crosses $\overline{Q}$, then by Lemma 3.2, Z crosses $\overline{Y}$. Hence, Z overlaps Y. So every setpair in $\mathcal{L}$ that overlaps Q also overlaps Y.

It follows that Q overlaps fewer setpairs in $\mathcal{L}$ than Y. To see this, note that (1) W overlaps Y, by our choice of W, and (2) W does not overlap Q, because $Q \preceq \overline{W}$ (so W does not cross $\overline{Q}$) and W and Q are head disjoint (so W does not cross Q). Thus we get a contradiction to the choice of Y, since Q is a tight setpair that overlaps fewer setpairs in $\mathcal{L}$ than Y, and $\chi_Q \notin span(\mathcal{L})$. □

Remarks. The results in this section apply to undirected graphs but not to directed graphs, since we use the fact that a setpair W with $f(W) > 0$ is tight iff the reverse setpair $\overline{W}$ has $f(\overline{W}) > 0$ and is tight. In the theorem, if we replace a setpair $W \in \mathcal{L}$ by its reverse $\overline{W}$, then this preserves the properties (i)–(iv).

4. An Approximation Algorithm for the k-Vertex Connectivity Problem

The main result of this section is as follows.

Theorem 4.1. *Let k and n be positive integers, and let $\epsilon < 1$ be a positive number such that k is at most $(1-\epsilon)n$. There is a polynomial-time algorithm that, given an n-vertex (directed or undirected) graph, finds a solution to the k-vertex connectivity problem of cost at most $O(\sqrt{n/\epsilon})$ times the optimal cost.* □

The k-vertex connectivity problem is modeled by the following integer program.

$$\begin{array}{llrcll} \text{(IP-VC)} & \text{minimize} & \sum_{e \in E} c_e \, x_e & & & \\ & \text{subject to} & \sum_{e \in \delta(W)} x_e & \geq & f(W) & \forall\, W \in \mathcal{S} \\ & & x_e & \in & \{0,1\} & \forall\, e \in E \end{array}$$

Here, f is the following non-negative integer function defined on the setpairs.

$$f(W) = \begin{cases} \max\{0,\ k - |V \setminus (W_h \cup W_t)|\}, & \text{if } W_t \neq \emptyset \text{ and } W_h \neq \emptyset \\ 0, & \text{otherwise} \end{cases}$$

We use (LP-VC) to denote the linear programming relaxation.

The k-vertex connectivity requirement function f is crossing bisupermodular [5], hence, f is also skew bisupermodular. This can be seen as follows. For each setpair W, let $p(W)$ denote $|V \setminus (W_t \cup W_h)|$. For any two setpairs $W = (W_t, W_h)$ and $Y = (Y_t, Y_h)$, we have $p(W) + p(Y) = p(W \otimes Y) + p(W \oplus Y)$, because

$$|(W_t \cup Y_t) \cup (W_h \cap Y_h)| + |(W_t \cap Y_t) \cup (W_h \cup Y_h)| = |W_t| + |W_h| + |Y_t| + |Y_h|.$$

Hence, the above function $f(\cdot)$ is crossing bisupermodular (if either $W \otimes Y$ or $W \oplus Y$ has head or tail empty, then note that W, Y do not cross). The results in [5] imply that a basic optimal solution to (LP-VC) (if it exists) can be found in polynomial time, see [5], [10, Theorem 6.4.9]. Moreover, the requirement function f and (LP-VC) have the following "self-reducibility" property: for any edge e the "reduced" requirement function f' is also crossing bisupermodular (and skew bisupermodular), where f' is given by $f'(W) = f(W) - 1$ if the setpair W has $e \in \delta(W)$ and $f'(W) = f(W)$ otherwise (since the zero-one incidence function of e on $\mathcal{S}$ is bisubmodular).

Theorem 4.1 follows directly from the next result.

Theorem 4.2. *Let $\epsilon < 1$ be a positive number, and suppose that $k \leq (1-\epsilon)n$. Then any nonzero basic solution of (LP-VC) has an edge of weight $\Omega(\sqrt{\frac{\epsilon}{n}})$.*

We give the proof for undirected graphs, using results on skew bisupermodular functions. After that, we sketch the proof for directed graphs via crossing bisupermodular functions. The next result is from [1, Theorem 1.1].

Theorem 4.3. *Suppose that the requirement function f for the linear program (LP-VC) is crossing (or, skew) bisupermodular. Let $\boldsymbol{x}$ be a nonzero basic solution of (LP-VC), and let $\mathcal{L}$ be a noncrossing family of setpairs characterizing $\boldsymbol{x}$. Then there exists an edge e with $x_e \geq \frac{1}{\Omega(\sqrt{|\mathcal{L}|})}$.* □

Focus on a nonzero basic solution $\boldsymbol{x}$ of (LP-VC). By Theorem 3.3, $\boldsymbol{x}$ corresponds to a noncrossing family of setpairs $\mathcal{L}$ such that each setpair has a nonempty head and a nonempty tail. We may assume that each setpair $W \in \mathcal{L}$ has $|W_h| \geq |W_t|$, else we may replace W by $\overline{W}$ and this preserves all the properties of $\mathcal{L}$ in Theorem 3.3. Let $\mathcal{P}$ denote the poset (and the Hasse diagram) representing $\mathcal{L}$, and recall that $\mathcal{P}$ has a node W for each setpair W in $\mathcal{L}$ and it has an arc (W, Y), where $Y \in \mathcal{L}$, if $W \preceq Y$ and there is no other $Z \in \mathcal{L}$ such that $W \preceq Z \preceq Y$ (we omit arcs implied by transitivity from $\mathcal{P}$); we call W a *child* of Y, and call Y a *parent* of W. (Note that the terms "node" and "arc" refer to the poset $\mathcal{P}$, and the terms "vertex" and "edge" refer to the input graph or digraph G.) A *diamond* of a poset is a set of four nodes W, X, Y, Z such that X and Y are incomparable, W is a common descendant of X, Y, and Z is a common ancestor of X, Y. A poset is called *diamond-free* if it contains no diamond. The poset $\mathcal{P}$ representing $\mathcal{L}$ is diamond-free, see [1, Lemma 3.1].

Let m denote $|\mathcal{L}|$. If $m \leq 4n/\epsilon$, then by the above theorem, there is an edge e with $x_e \geq \sqrt{\frac{\epsilon}{\Theta(n)}}$. Hence, in this case, Theorem 4.2 holds. The rest of the proof focuses on the other case ($m > 4n/\epsilon$) and shows that there exists an edge e with either $x_e \geq \frac{1}{\Theta(1)}$ or $x_e \geq \Omega(\epsilon) \geq \Omega(\sqrt{\frac{\epsilon}{n}})$. (Note that $k \leq n-1$, hence $\epsilon \geq \frac{1}{n}$, and so $\epsilon \geq \sqrt{\frac{\epsilon}{n}}$.)

Lemma 4.4. *Suppose that $k \leq (1-\epsilon)n$. Then there are at most $\frac{2}{\epsilon}$ pairwise incomparable setpairs in $\mathcal{L}$ such that all their tails contain a common vertex.*

Proof. Let the setpairs $Y^1, Y^2, \ldots, Y^q \in \mathcal{L}$ be pairwise incomparable, and let v be a vertex that occurs in the tail of each of these setpairs. Since the Y^i are incomparable and their tails intersect, their heads are disjoint. Each head Y^i_h has cardinality at least $\frac{1}{2}(n-k) \geq \frac{1}{2}\epsilon n$; this follows from the properties of $\mathcal{L}$, and the fact that each requirement $f(Y^i)$ is at least one. Hence, there are at most $\frac{2}{\epsilon}$ such setpairs. □

We partition the setpairs in $\mathcal{L}$ into several sets.

- W is a type I setpair if it has no parents in $\mathcal{P}$.

- W is a type II setpair if it has at least two parents in $\mathcal{P}$.
- W is a type III setpair if it has exactly one parent Y in $\mathcal{P}$, and $W_t = Y_t$.
- W is a type IV setpair if it has exactly one parent Y in $\mathcal{P}$, and $W_t \neq Y_t$.

Lemma 4.5. *(a) There are at most $\frac{2n}{\epsilon}$ setpairs of type IV. (b) Each node (setpair) $W \in \mathcal{P}$ has at most $\frac{2n}{\epsilon}$ children.*

Proof. We prove (a) first, then sketch the proof of (b). To each type IV setpair W with parent Y we assign a vertex v in $W_t \setminus Y_t$. We claim that a vertex can be assigned to at most $\frac{2}{\epsilon}$ setpairs of type IV. This will imply the claim. Take a vertex v that is assigned to the type IV setpairs $W^1, W^2, \ldots, W^q$. Let their parents be $Y^1, Y^2, \ldots, Y^q$, respectively. Now $v \in W_t^1 \cap W_t^2 \cap \cdots \cap W_t^q$, but $v \notin Y_t^1 \cup Y_t^2 \cup \cdots \cup Y_t^q$. The setpairs $W^1, W^2, \ldots, W^q$ associated with v must be pairwise incomparable (since v is in the tail of every W^i, but for every proper ancestor of a W^i note that v is not in the tail). Then by Lemma 4.4 applied to $W^1, W^2, \ldots, W^q$, there are at most $\frac{2}{\epsilon}$ such setpairs whose tails contain v.

For (b), consider all the children $Y^1, Y^2, \ldots, Y^q$ of a node W. Then $Y^1, Y^2, \ldots, Y^q$ are pairwise incomparable (otherwise, one of the arcs (Y^i, W) in $\mathcal{P}$ is redundant), and moreover, each of the tails Y_t^i, $i = 1, \ldots, q$, contains W_t which is nonempty. Then by Lemma 4.4 applied to $Y^1, Y^2, \ldots, Y^q$, we have $q \leq \frac{2}{\epsilon}$. □

We have $|\mathcal{L}| = m \geq \frac{4n}{\epsilon}$, hence, by Lemma 4.5, $\mathcal{L}$ has at least $\frac{m}{6}$ setpairs of one of the types I, II, or III.

(Type I) Suppose that there are at least $\frac{m}{6}$ type I setpairs. The type I setpairs are pairwise incomparable. Hence the edge sets $\delta(W)$, where W is a type I setpair, are *quasidisjoint*, i.e., for each edge uv, there are at most two setpairs W, Y (among the type I setpairs) such that $uv \in \delta(W)$ (say $u \in W_t$ and $v \in W_h$) and $uv \in \delta(Y)$ (say $v \in Y_t$ and $u \in Y_h$). Hence, there is a type I setpair $W \in \mathcal{L}$ such that $\delta(W)$ has size at most $2 \cdot 6$, and so there is an edge $e \in \delta(W)$ with $x_e \geq \frac{1}{2 \cdot 6}$.

(Type II) Let $Y^1, \ldots, Y^q$ be the minimal nodes in $\mathcal{P}$ (those with no children); clearly, these nodes are pairwise incomparable. Each Y^i and its ancestors forms a (directed) tree (since $\mathcal{P}$ is diamond-free). Each node of $\mathcal{P}$ is in at most $\frac{2}{\epsilon}$ of these trees, by Lemma 4.4. Focus on one of the trees and note that the number of leaf nodes (maximal nodes of $\mathcal{P}$) of the tree is greater than or equal to the number of type II nodes in the tree. Summing over all the trees, we see that the total number of leaf nodes (maximal nodes of $\mathcal{P}$) is at least $\frac{\epsilon}{2}$ times the number of type II nodes in $\mathcal{P}$, hence, $\mathcal{P}$

has at least $\frac{m\epsilon}{2\cdot 6}$ maximal nodes; clearly, these nodes are pairwise incomparable. Then the edge sets $\delta(W)$, where W is a maximal node of $\mathcal{P}$, are quasidisjoint, so there is an edge e with $x_e \geq \frac{\epsilon}{2\cdot 6}$.

(Type III) Finally, suppose that there are at least $\frac{m}{6}$ type III setpairs. Consider all of the type III setpairs $W^1, W^2, \ldots W^q$ and their unique parents $Y^1, Y^2, \ldots Y^q$. Observe that $f(W^i)+1 \leq f(Y^i)$, since $|W_t \cup W_h| < |Y_t \cup Y_h|$. Moreover, if W^i and W^j (where $1 \leq i < j \leq q$) have distinct parents Y^i and Y^j, then the edge sets $\delta(Y^i) \setminus \delta(W^i)$ and $\delta(Y^j) \setminus \delta(W^j)$ are quasidisjoint. To see this, consider an edge $uv \in \delta(Y^i) \setminus \delta(W^i)$ with $u \in Y^i_t = W^i_t$ and $v \in Y^i_h - W^i_h$. Now, every proper ancestor Z of W^i has $v \in Z_h$, so if W^j is an ancestor of W^i, then we cannot have $uv \in \delta(Y^j) \setminus \delta(W^j)$, $u \in Y^j_t = W^j_t$, and $v \in Y^j_h - W^j_h$.

Consequently, by Lemma 4.5, any edge occurs in at most $\frac{4}{\epsilon}$ of the edge sets $\delta(Y^i) \setminus \delta(W^i)$, and moreover, $x(\delta(Y^i) \setminus \delta(W^i)) \geq 1$, for each $i = 1, \ldots, q$. Hence, there is an edge e with $x_e \geq \frac{\epsilon}{4\cdot 6}$.

This concludes the proof of Theorem 4.2 for undirected graphs: any nonzero basic solution of (LP-VC) has an edge of weight $\Omega(\sqrt{\epsilon/n})$.

The proof extends to directed graphs via Theorem 4.3. We partition the noncrossing family of setpairs $\mathcal{L}$ into two subfamilies $\mathcal{L}_1$ and $\mathcal{L}_2$, depending on whether or not a setpair $W \in \mathcal{L}$ has $|W_h| \geq |W_t|$. Suppose that $|\mathcal{L}_1| \geq |\mathcal{L}_2|$; otherwise, we use a symmetric argument. Then we apply the arguments for undirected graphs to $\mathcal{L}_1$ (partitioning it into types I, II, III, IV, etc.). We obtain similar lower bounds on $\max\{x_e\}$, the main difference being that we lose a factor of two in the lower bound (since we have $|\mathcal{L}_1| \geq \frac{m}{2}$ rather than $|\mathcal{L}| = m$). Hence, Theorem 4.2 holds for directed graphs. □

Theorem 4.1 follows from Theorem 4.2: for $k \leq (1-\epsilon)n$, iterative rounding gives an $O(\sqrt{n/\epsilon})$ approximation algorithm for the (directed or undirected) k-vertex connectivity problem.

Is it possible to improve substantially on our analysis of the iterative rounding method for the k-vertex connectivity problem? This is not clear at the moment, since our approximation guarantee holds for directed graphs for large values of k (say $\Omega(\sqrt{n}) < k < n - \Omega(1)$), and for this case, the only other approximation guarantee known is $\Omega(k)$, which is substantially worse than our approximation guarantee. Moreover, there is an example of the undirected problem such that there is an extreme point x of (LP-VC) such that each edge e has $x_e \geq \frac{1}{\Omega(\sqrt{k})}$. Hence, in Theorem 4.2, the lower bound on the maximum weight of an edge cannot be improved beyond $\frac{1}{\Omega(\sqrt{k})}$.

5. An Approximation Algorithm for the Element Connectivity Problem

This section sketches a different proof of the 2-approximation guarantee for the element connectivity problem, a result first obtained in [7].

Recall that $T \subseteq V$ is the set of terminal vertices, and for all $i, j \in T$, there is a requirement for r_{ij} element disjoint paths between i and j. Let $g(W) = \max_{i \in W_t, j \in W_h} \{r_{ij}\}$, $\forall W \in \mathcal{S}$. Define a requirement function f on the setpairs $W \in \mathcal{S}$ as follows:

$$f(W) = \begin{cases} \max\{0,\ g(W) \ - \ |V \setminus (W_t \cup W_h)|\}, & \text{if } T \subseteq W_t \cup W_h \\ 0, & \text{otherwise.} \end{cases}$$

Note that if a setpair W has a positive requirement, then all the terminals are in $W_t \cup W_h$. It is easily seen that the integer program (IP), with the above requirement function f, models the element connectivity problem (for more details see [12]).

The requirement function f for the element connectivity problem is skew bisupermodular. Let us sketch the proof. Recall from Section 4 that $p(W)$ denotes $|V \setminus (W_t \cup W_h)|$ for each setpair W, and for any two setpairs W, Y we have $p(W) + p(Y) = p(W \otimes Y) + p(W \oplus Y)$. We claim that the function g is skew bisupermodular. The claim directly implies that the function $f(\cdot)$ is skew bisupermodular. Here is the proof of the claim. Define the function g' as follows. The groundset for g' is the set of terminals T. Let $g'(S) = \max_{i \in S, j \notin S} \{r_{ij}\}$ for any subset S of T. In addition, $g'(T) = g'(\emptyset) = 0$. Note the following correspondence between g' and g: for any setpair W with $T \subseteq W_t \cup W_h$, we have $g(W) = g'(T \cap W_h)$. Clearly, g' is symmetric, that is, $g'(S) = g'(T \setminus S)$ for all subsets S of T. A symmetric (non-negative, integral) function s on the subsets of a groundset U is called *weakly supermodular* if $s(U) = 0$, and for every two subsets X and Z, we have either $s(X) + s(Z) \leq s(X \cap Z) + s(X \cup Z)$ or $s(X) + s(Z) \leq s(\bar{X} \cap Z) + s(\bar{X} \cup Z)$, where $\bar{X} = U \setminus X$. Frank [3, Proposition 5.4] (also Goemans et al [8]) showed that g' is weakly supermodular. Now observe that g' is weakly supermodular if and only if g is skew bisupermodular. This follows from the term by term correspondence between the skew bisupermodular inequalities for g and the weakly supermodular inequalities for g'. (To see this in detail, consider overlapping setpairs W and Y with $f(W) > 0, f(Y) > 0$. Clearly, for both W and Y, T is contained in the union of the head and the tail. Then both $W \otimes Y$ and $W \oplus Y$ also have this property. Let $X = W_h \cap T$ and let $Z = Y_h \cap T$. Suppose that one of the weakly supermodular inequalities holds for X and Z, say, $g'(X) + g'(Z) \leq g'(X \cap Z) + g'(X \cup Z)$. Note that $g'(X) = g(W)$, $g'(Z) = g(Y)$, $g'(X \cap Z) = g(W \otimes Y)$, and $g'(X \cup Z) = g(W \oplus Y)$. Hence, the skew bisupermodular inequality

$g(W) + g(Y) \leq g(W \otimes Y) + g(W \oplus Y)$ holds for W and Y. Similarly, it can be seen that if g is skew bisupermodular, then g' is weakly supermodular.) The claim follows.

Let (LP-EC) denote the linear programming relaxation of this integer program. A basic optimal solution to (LP-EC) (if it exists) can be found in polynomial time. This follows because a strong separation oracle for (LP-EC) may be obtained using standard network flow techniques (for more details, see [12]).

To prove the key result for the 2-approximation guarantee for element connectivity, namely, there exists an edge of value at least $\frac{1}{2}$ in any non-zero basic solution of (LP-EC), we need to develop some preliminaries and a key lemma. Based on this, the rest of the proof becomes an easy extension of the counting arguments in Jain's proof of [11, Lemma 4.6].

Let $\boldsymbol{x}$ be a non-zero basic solution of (LP-EC). We may assume that each edge $e \in E$ has $x_e > 0$, since any other edges may be discarded. In addition, we may assume that each edge $e \in E$ has $x_e < 1$, otherwise we are done. Let $\mathcal{L}$ be the corresponding non-overlapping family of setpairs of Theorem 3.3. Since E is the support of a basic solution, $|E| = |\mathcal{L}|$. Let m denote $|E|$. Let τ be an arbitrarily chosen terminal vertex in T. For each setpair in $\mathcal{L}$, we assume that the head contains τ. Otherwise, if τ is in the tail of $W \in \mathcal{L}$, then we replace W by $\overline{W}$ and this preserves all the properties of $\mathcal{L}$ in Theorem 3.3. Then the tails of the setpairs in $\mathcal{L}$ form a laminar family. This holds because the setpairs in $\mathcal{L}$ are pairwise non-overlapping, and the heads of any two setpairs in $\mathcal{L}$ intersect (since τ is in every head). We may also view $\mathcal{L}$ as a forest of rooted trees. Call $W \in \mathcal{L}$ the *parent* of $Y \in \mathcal{L}$ (and Y a *child* of W) if $W_t \supseteq Y_t$ and any other setpair $Z \in \mathcal{L}$ with $Z_t \supseteq Y_t$ either has $Z_t \supset W_t$ or has $Z_t = W_t$ and $Z_h \subset W_h$. We may refer to the setpairs in $\mathcal{L}$ as *nodes* of the forest. We say that a setpair $Y \in \mathcal{L}$ is *smaller* than a setpair $W \in \mathcal{L}$ if $Y \succeq W$, i.e., $Y_t \subseteq W_t$ and Y is a descendant of W in the forest.

In the rest of this section, we take the edges in E to be bidirected. That is, we replace every undirected edge $\{p, q\}$ in E by a pair of directed arcs pq and qp. For a setpair $W = (W_t, W_h)$, let $\delta(W)$ denote the set of arcs pq with $p \in W_t$ and $q \in W_h$. (Though we use the same notation $\delta()$ for undirected edges and directed arcs, the context will resolve any ambiguity.) The other notation, $\boldsymbol{x}$, χ_W, etc. remains the same. An arc pq is called *good* if there is a setpair $W \in \mathcal{L}$ such that pq is in $\delta(W)$. Note that, in the undirected setting, every edge e is in $\delta(W)$ for some $W \in \mathcal{L}$, but in the directed setting, there may exist arcs that are not good.

The proof of the main result hinges on assigning *token arcs* to the setpairs in $\mathcal{L}$. Here is a brief sketch; the details are given in the proof of Theorem 5.2. We have $|\mathcal{L}| = |E| = m$ and thus a total of $2m$ arcs. If a basic solution $\boldsymbol{x}$ of (LP-EC) has $x_e < \frac{1}{2}$ for each edge e, then we will be able to distribute the token arcs among the setpairs in $\mathcal{L}$ such that each gets at least two token arcs and some setpair gets at least three token arcs. Thus, we end up with at least $2|\mathcal{L}| + 1 = 2m + 1$ arcs. This contradiction shows that our assumption on $\max_e \{x_e\}$ is false. We use the following two rules to distribute the token arcs such that no arc is assigned to two different setpairs.

Rule 1. If pq is a good arc, then we assign it to the smallest setpair $W \in \mathcal{L}$ such that $pq \in \delta(W)$.
Rule 2. If pq is not a good arc, then we assign it to the smallest setpair $W \in \mathcal{L}$ such that $q \in W_t$ and $p \notin W_h$ (note that neither pq nor qp is in $\delta(W)$).

(Rule 2 is essential in the sense that the analysis fails if we insist on assigning arcs pq only to those setpairs W such that $\{p, q\} \in \delta(W)$.)

Given two setpairs W and Y, let $\delta(W) \bigtriangledown \delta(Y)$ denote the symmetric difference of $\delta(W)$ and $\delta(Y)$. Given $S^1, S^2, \ldots, S^\ell$, a collection of mutually disjoint vertex sets, we denote by $\gamma(S^1, S^2, \ldots, S^\ell)$ the set of arcs pq such that p and q are in different sets S^i and S^j, where $1 \le i, j \le \ell$.

Lemma 5.1. *Let W be a setpair in the forest $\mathcal{L}$, and let its children be the setpairs $Y^1, Y^2, \ldots, Y^\ell$. Then for every arc pq in*

$$(\delta(W) \bigtriangledown (\delta(Y^1) \cup \cdots \cup \delta(Y^\ell))) \ \setminus \ \gamma(Y_t^1, \ldots, Y_t^\ell)$$

either pq or qp is assigned to W by Rule 1 or Rule 2.

Proof. Note that the lemma applies to all arcs in the symmetric difference of $\delta(W)$ and $\bigcup_{i=1}^{\ell} \delta(Y^i)$ except for arcs whose two end-vertices are in the tails of different children. Figure 1 illustrates the arguments that follow, for the particular case of $\ell = 2$. [Note, for example, that the arc a_4 is in $\gamma(Y_t^1, Y_t^2)$.]

We have two cases to deal with.
(a) $pq \in \delta(W) \ \setminus \ \bigcup_{i=1}^{\ell} \delta(Y^i)$.
Now $q \in W_h \subseteq Y_h^i$, for $i = 1, \ldots, \ell$. However, by assumption, $p \in W_t \setminus \cup_{i=1}^{\ell} Y_t^i$ and so W is the smallest setpair in $\mathcal{L}$ such that p is in the tail. Thus, W is the smallest setpair in $\mathcal{L}$ with $pq \in \delta(W)$, and hence pq is assigned to W by Rule 1. [The arc a_1 in Figure 1 is an example.]
(b) $pq \in \delta(Y^i) \ \setminus \ \delta(W)$, for some $i \in \{1, \ldots, \ell\}$, and $pq \notin \gamma(Y_t^1, \ldots, Y_t^\ell)$.
Note that $p \in Y_t^i \subseteq W_t$ and $q \notin W_h$. We have two subcases.
(i) $q \in W_t$.
First, we show that qp is not a good arc. Clearly, $q \notin Y_t^i$ as $pq \in \delta(Y^i)$. For any other child Y^j,

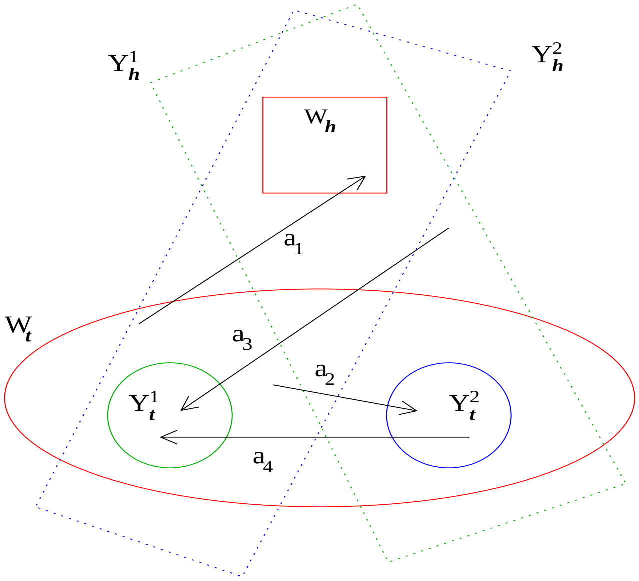

FIGURE 1. An illustration of the proof of Lemma 5.1.

observe that $q \notin Y_t^j$ as $pq \notin \gamma(Y_t^1, \ldots, Y_t^\ell)$. Hence, $q \in W_t \setminus \cup_{j=1}^{\ell} Y_t^j$. Then W is the smallest setpair in $\mathcal{L}$ whose tail contains q, and there is no setpair $Z \in \mathcal{L}$ such that $qp \in \delta(Z)$ (otherwise, $p \in Z_h \cap W_t$ and $q \in Z_t \subseteq W_t$, which is impossible). Thus qp is not a good arc. Moreover, $p \notin W_h$ (since $p \in W_t$), and so Rule 2 assigns qp to W. [The arc a_2 in Figure 1 is an example.]

(ii) $q \notin W_t$.

We first show that qp is not a good arc. Suppose $Z \in \mathcal{L}$ is a setpair such that $qp \in \delta(Z)$. We derive a contradiction by showing that W and Z overlap; in particular, W and $\overline{Z}$ cross. Note that W_t and Z_t are disjoint, because $p \in W_t \setminus Z_t$ and $q \in Z_t \setminus W_t$. Now Z_t and W_h have a terminal in common, because Z_t has at least one terminal and $W_t \cup W_h$ contains all the terminals. Then Z_t and W_h intersect properly, because (1) there is a terminal in $Z_t \cap W_h$, (2) the terminal τ is in $W_h \setminus Z_t$, and (3) we have $q \in Z_t \setminus W_h$. Finally, note that Z_h and W_t intersect since both contain the vertex p. Hence, $\overline{Z}$ and W cross. Therefore, qp is not a good arc and Rule 2 applies to qp. Rule 2 assigns qp to W since $p \in W_t$, $q \notin W_h$, and for any smaller setpair $X \in \mathcal{L}$ with $p \in X_t$ we have $q \in X_h$ (since $p \in Y_t^i$ and $q \in Y_h^i$). [The arc a_3 in Figure 1 is an example.] $\square$

We need an observation in order to apply parity arguments in the proof of Theorem 5.2. Let $Y = (Y_t, Y_h)$ and $Z = (Z_t, Z_h)$ be setpairs in $\mathcal{L}$. If Y and Z are tail disjoint, then $Y_t \subseteq Z_h$ and $Z_t \subseteq Y_h$. To see this, note that Y_t and Z_h have a terminal in common, since Y_t has at least one terminal, and $Z_t \cup Z_h$ contains all the terminals. Similarly, Z_t and Y_h have a terminal in common. Suppose that Y_t is not a subset of Z_h. Then it can be seen that Y_t and Z_h intersect properly.

Hence, $\overline{Z}$ crosses Y, and so Y and Z overlap. This contradiction shows that $Y_t \subseteq Z_h$. Similarly, we have $Z_t \subseteq Y_h$.

Theorem 5.2. *Any non-zero basic solution of (LP-EC) has an edge of weight at least $\frac{1}{2}$.*

The proof is by contradiction, and uses similar arguments to Jain's proof of [11, Lemma 4.6]. We include it in the appendix, for the sake of completeness.

Thus, we get a factor 2 approximation for the element connectivity problem by applying the iterative rounding method to (LP-EC). Note that if f is a (non-negative, integral) skew bisupermodular function on $\mathcal{S}$ and e is any edge, then the function f' is also skew bisupermodular, where $f'(W) = \max\{0,\ f(W)\ -\ 1\}$, if $e \in \delta(W)$, and $f'(W) = f(W)$, otherwise.

Theorem 5.3. *[Fleischer-Jain-Williamson [7]] There is a polynomial-time algorithm that given an instance of the element connectivity problem, finds a solution of cost at most twice the optimal cost.* □

Theorem 5.3 extends to element connectivity problems where the connectivity requirements are given by a weakly supermodular function g' on the set of terminals T (instead of the r_{ij} values). Then the requirement function f on the setpairs $W \in \mathcal{S}$ given by

$$f(W) = \begin{cases} \max\{0,\ g'(W_h \cap T)\ -\ |V \setminus (W_t \cup W_h)|\}, & \text{if } T \subseteq W_t \cup W_h \\ 0, & \text{otherwise.} \end{cases}$$

is skew bisupermodular. If an efficient separation oracle for the linear programming relaxation is available, then we get a 2-approximation algorithm. To see that this extension holds, note that the proof does not refer to the r_{ij} values.

References

[1] J. Cheriyan and S. Vempala, "Edge coverings of setpairs and the iterative rounding method", submitted for journal publication, September 2001. Preliminary version in the *Proceedings of the 8th International Integer Programming and Combinatorial Optimization Conference*, Utrecht, The Netherlands, 2001.

[2] J. Cheriyan, S. Vempala and A. Vetta, "An approximation algorithm for the minimum-cost k-vertex connected subgraph," submitted for journal publication, July 2001.

[3] A. Frank, "Augmenting graphs to meet edge-connectivity requirements", *SIAM J. Discrete Math.*, **5**, pp. 25-53, 1992. Preliminary version in *Proc. 31st IEEE FOCS*, pp. 708-718, 1990.

[4] A. Frank, "Connectivity augmentation problems in network design", in *Mathematical Programming: State of the Art 1994*, (Eds. J. R. Birge and K. G. Murty), pp. 34–63, The University of Michigan, Ann Arbor, MI, 1994.

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

[6] A. Frank and E. Tardos, "An application of submodular flows", *Linear Algebra and its Applications*, **114/115**, pp.329-348, 1989.

[7] L. Fleischer, K. Jain, and D. P. Williamson, "An iterative rounding 2-approximation algorithm for the element connectivity problem," to appear in *Proc. 42nd IEEE FOCS*, October 2001.

[8] M. Goemans, A. Goldberg, S. Plotkin, D. Shmoys, E. Tardos and D. Williamson, "Improved approximation algorithms for network design problems", *Proc. 5th ACM SIAM Symposium on Discrete Algorithms*, pp. 223-232, 1994.

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

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

[11] 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 FOCS*, 1998.

[12] K. Jain, I. Mandoiu, V. Vazirani and D. Williamson, "A primal-dual schema based approximation algorithm for the element connectivity problem", *Proc. 10th ACM SIAM Symposium on Discrete Algorithms*, pp. 484-489, 1999.

[13] G. Kortsarz and Z. Nutov, "Approximating node connectivity problems via set covers", *Approximation algorithms for combinatorial optimization*, Third international workshop, APPROX 2000, (K. Jansen, S. Khuller (eds.)), Springer, LNCS **1913**, pp.194-205, 2000.

[14] R. Ravi and D. P. Williamson, "An approximation algorithm for minimum-cost vertex-connectivity problems," *Algorithmica*, **18**, 1997, pp. 21-43. (Preliminary version in *Proc. 6th Ann. ACM-SIAM Symposium on Discrete Algorithms*, 1995, pp. 332-341.)
Erratum, submitted to *Algorithmica*, 9 pages, July 2001.

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

6. Appendix: A proof of Theorem 5.2

Let us restate Theorem 5.2 for convenience: Any non-zero basic solution of (LP-EC) has an edge of weight at least $\frac{1}{2}$.

Proof. Let $\boldsymbol{x} \neq \boldsymbol{0}$ be a basic solution to (LP-EC) such that $x_e < \frac{1}{2}$ for each edge e. Then the following claim implies that we end up with at least $2|\mathcal{L}| + 1$ arcs. This is a contradiction. Hence, $\max_e \{x_e\} \geq \frac{1}{2}$.

Define the *corequirement* of an edge e, denoted $\alpha(e)$, to be $\frac{1}{2} - x_e > 0$, and the *corequirement* of a setpair W, denoted $\alpha(W)$, to be $\sum_{e \in \delta(W)} \alpha(e)$. Note that a tight setpair W has $\alpha(W) = \frac{1}{2}|\delta(W)| - f(W) > 0$. We will use the fact that if a setpair W has $\alpha(W) = \frac{1}{2}$, then $|\delta(W)|$ is odd.

Claim. *Suppose that $x_e < \frac{1}{2}$, $\forall e \in E$. Then the arcs may be redistributed to the nodes in the forest $\mathcal{L}$ such that for each rooted subtree of $\mathcal{L}$, each node gets at least two token arcs and the root gets at least three token arcs. Moreover, a root with corequirement $\neq \frac{1}{2}$ gets at least four token arcs.*

Again, we prove the claim by induction. For the base case take a leaf node $W \in \mathcal{L}$. Then $\delta(W)$ has at least 3 arcs. By Rule 1, all these arcs are assigned to W. If W gets exactly 3 token arcs, then note that $f(W) = 1$ and $|\delta(W)| = 3$, so we have $\alpha(W) = \frac{1}{2}$. Thus W satisfies the induction hypothesis.

For the induction step, consider a subtree rooted at a node $W \in \mathcal{L}$. We have four cases.
(a) W has at least four children.

By the induction hypothesis, each child has at least 3 token arcs. We reassign one token arc from each child to W. Hence, W gets at least 4 token arcs.

(b) W has three children.

Call the children X, Y and Z. If one of the children has at least 4 token arcs, then we can reassign the token arcs from X, Y and Z to W so that W has at least 4 token arcs and each of X, Y and Z is left with at least 2 token arcs.

Otherwise, by the induction hypothesis, each of X, Y and Z has 3 token arcs, so each has corequirement $\frac{1}{2}$ and each of $|\delta(X)|$, $|\delta(Y)|$ and $|\delta(Z)|$ is odd. We reassign one token arc from each of X, Y and Z to W. Now focus on the symmetric difference of $\delta(W)$ and $\delta(X) \cup \delta(Y) \cup \delta(Z)$. By the linear independence of χ_W, χ_X, χ_Y and χ_Z, there is at least one arc pq in the symmetric difference. If possible, choose $pq \notin \gamma(X_t, Y_t, Z_t)$. Then by Lemma 5.1 either pq or qp is assigned to W as a token arc. Hence, W ends up with at least 4 token arcs. Otherwise, every arc in the symmetric difference is in $\gamma(X_t, Y_t, Z_t)$. Therefore $\delta(W) \subset \delta(X) \cup \delta(Y) \cup \delta(Z)$, and for each arc pq in $(\delta(X) \cup \delta(Y) \cup \delta(Z)) \setminus \delta(W)$ the arc qp is also in the same set (by the observation stated after Lemma 5.1). Hence, $|\delta(W)|$ is odd, and so the corequirement $\alpha(W)$ is a *semi integer* (i.e., an odd integer multiple of $\frac{1}{2}$). Moreover, $\alpha(W) < \alpha(X) + \alpha(Y) + \alpha(Z) = \frac{3}{2}$. Hence, $\alpha(W) = \frac{1}{2}$, and so W satisfies the induction hypothesis.

(c) W has two children.

Call the children Y and Z. If we can assign 4 token arcs to W by reassigning token arcs from Y and Z, and by applying Lemma 5.1 to $(\delta(W) \triangledown (\delta(Y) \cup \delta(Z))) \setminus \gamma(Y_t, Z_t)$, then we are done. Otherwise, we are left with two subcases.

(i) Each of Y and Z has 3 token arcs and a corequirement of $\frac{1}{2}$, and there is one arc pq in $(\delta(W) \triangledown (\delta(Y) \cup \delta(Z))) \setminus \gamma(Y_t, Z_t)$. Then W gets 3 token arcs, one each by reassigning arcs from Y and Z and one by Lemma 5.1. Both $|\delta(Y)|$ and $|\delta(Z)|$ are odd (since $\alpha(Y) = \alpha(Z) = \frac{1}{2}$) and the arcs (if any) in $\gamma(Y_t, Z_t)$ occur in pairs uv and vu (by the observation stated after Lemma 5.1). Therefore, $|\delta(W)|$ is odd and $\alpha(W)$ is a semi integer. Moreover, $0 < \alpha(W) \leq \alpha(Y) + \alpha(Z) + \alpha(\{p, q\}) < \frac{3}{2}$. Hence, $\alpha(W) = \frac{1}{2}$, and so W satisfies the induction hypothesis.

(ii) At least one of the children, say Z, has 3 token arcs and a corequirement of $\frac{1}{2}$, and there are no arcs in $(\delta(W) \triangledown (\delta(Y) \cup \delta(Z))) \setminus \gamma(Y_t, Z_t)$. We will prove that this case cannot occur. First, note that $|\delta(W)|$ and $|\delta(Y)|$ have opposite parity (one is odd and the other is even), because $|\delta(Z)|$ is odd and the arcs in the symmetric difference of $\delta(W)$ and $\delta(Y) \cup \delta(Z)$ occur in pairs uv and vu. Furthermore, note that χ_W, χ_Y and χ_Z are linearly independent. It follows that there is an arc in $\gamma(Y_t, Z_t)$ and there is an arc in $\delta(Z) \cap \delta(W)$. Hence, $\alpha(W) < \alpha(Y) + \alpha(Z)$ (due to the arc in $\gamma(Y_t, Z_t)$) and $\alpha(Y) < \alpha(W) + \alpha(Z)$ (due to the arc in $\delta(Z) \cap \delta(W)$). Therefore, $\alpha(Y) - \frac{1}{2} < \alpha(W) < \alpha(Y) + \frac{1}{2}$, implying that $\alpha(Y) = \alpha(W)$ (since $\alpha(W)$ is an integer multiple of $\frac{1}{2}$). Then $|\delta(W)|$ and $|\delta(Y)|$ have the same parity. This contradiction shows that this case cannot occur.

(d) W has one child.

Call the child Y. If the symmetric difference of $\delta(W)$ and $\delta(Y)$ has exactly 2 arcs and $\alpha(Y) = \frac{1}{2}$, then W gets 3 token arcs (two by Lemma 5.1 and one reassigned from Y). In this case, note that $f(W) = f(Y)$ and $|\delta(W)| = |\delta(Y)|$. As a result, $\alpha(W) = \alpha(Y) = \frac{1}{2}$. Otherwise, W gets at least 4 token arcs. In either case, W satisfies the induction hypothesis. This completes the proof of the claim.

The theorem follows. □

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

E-mail address: {vempala, avetta}@math.mit.edu