Iterative Rounding 2-Approximation Algorithms for Minimum-Cost Vertex Connectivity Problems *

Lisa Fleischer
GSIA
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
lkf@andrew.cmu.edu

Kamal Jain
Microsoft Research
One Microsoft Way
Redmond, WA 98052
kamalj@microsoft.com

David P. Williamson
IBM Almaden Research Center
650 Harry Rd.
San Jose, CA 95120
dpw@almaden.ibm.com

December 1, 2003

Abstract

The survivable network design problem (SNDP) is the following problem: given an undirected graph and values r_{ij} for each pair of vertices i and j, find a minimum-cost subgraph such that there are at least r_{ij} disjoint paths between vertices i and j. In the edge connected version of this problem (EC-SNDP), these paths must be edge-disjoint. In the vertex connected version of the problem (VC-SNDP), the paths must be vertex disjoint. The *element connectivity problem* (ELC-SNDP, or ELC) is a problem of intermediate difficulty. In this problem, the set of vertices is partitioned into terminals and nonterminals. The edges and nonterminals of the graph are called *elements*. The values r_{ij} are only specified for pairs of terminals i, j, and the paths from i to j must be element disjoint. Thus if $r_{ij} - 1$ elements fail, terminals i and j are still connected by a path in the network.

These variants of SNDP are all known to be NP-hard. The best known approximation algorithm for the EC-SNDP has performance guarantee of 2

*This paper is the union of two previously published extended abstracts. The extended abstract [6] presents negative examples for $\{0, 1, \ldots, k\}$-vertex connectivity, and the 2-approximation algorithm for $\{0, 1, 2\}$-vertex connectivity. This result is generalized to include element connectivity in [8].

and iteratively rounds solutions to a linear programming relaxation of the problem. ELC has a primal-dual $O(\log k)$-approximation algorithm, where $k = \max_{i,j} r_{ij}$. VC-SNDP is not known to have a non-trivial approximation algorithm.

In this paper we investigate applying iterative rounding to ELC and VC-SNDP. We show that iterative rounding will not yield a constant factor approximation algorithm for general VC-SNDP. On the other hand, we show how to extend the analysis of iterative rounding applied to EC-SNDP to yield 2-approximation algorithms for both general ELC, and for the case of VC-SNDP when $r_{ij} \in \{0, 1, 2\}$. The latter result improves on an existing 3-approximation algorithm. The former is the first constant factor approximation algorithm for a general survivable network design problem that allows node failures.

1 Introduction

The survivable network design problem (SNDP) is the following problem: given an undirected graph and values r_{ij} for each pair of vertices i and j, find a minimum-cost subgraph such that there are at least r_{ij} disjoint paths between vertices i and j. In the edge connected version of this problem (EC-SNDP), these paths must be edge disjoint. In the vertex connected version of the problem (VC-SNDP), the paths must be vertex disjoint. Jain et al. [14] propose a version of the problem intermediate in difficulty to these two, called the *element connectivity problem* (ELC-SNDP, or ELC). In this problem, the set of vertices is partitioned into terminals and nonterminals. The edges and nonterminals of the graph are called *elements*. The values r_{ij} are only specified for pairs of terminals i, j, and the paths from i to j must be element disjoint. Thus if $r_{ij} - 1$ elements fail, terminals i and j are still connected by a path in the network.

The motivation for studying element connectivity is the following: in real networks, both edges (links) and vertices (routers) fail. However, typically network terminals (end hosts) are more robust and located at the fringes of the network. Thus the failure of end hosts is uncommon, and less vital to the connectivity of the network as a whole. Additionally, vertex connectivity problems are much less well understood than edge connectivity problems. Thus, trying to capture node failures by using a vertex connectivity model makes the problem much more difficult. Element connectivity allows the modeling of node failures, while, as we will show in this paper, it shares some of the nice structure that edge connectivity problems have.

The three variants of the survivable network design problem are all NP-hard, since they all include the Steiner tree problem as a special case. Hence we consider *approximation algorithms* for these problems. We say we have a ρ-approximation algorithm for a problem if we have a polynomial-time algorithm which produces a solution of value no more than ρ times the value of an optimal solution.

1.1 Related Work

The best approximation algorithm for EC-SNDP known is a 2-approximation algorithm due to Jain [13]. This algorithm improved upon a primal-dual $2H_k$-approximation algorithm for EC-SNDP of Goemans et al. [11], where $k = \max_{i,j} r_{ij}$ and $H_k = 1 + \frac{1}{2} + \cdots + \frac{1}{k}$. Jain's algorithm is in fact somewhat more general, and gives an algorithm with performance guarantee 2 for selecting a minimum-cost set of edges such that at least $g(S)$ edges are selected from every cut $\delta(S) = \{(u,v) \in E : u \in S, v \notin S\}$, when g is a weakly supermodular function [11]. The algorithm runs in polynomial time given the existence of a polynomial-time separation algorithm. The EC-SNDP problem corresponds to a particular weakly supermodular function, and the polynomial-time separation algorithm exists in this case. Jain considers a linear programming relaxation of the problem which has a variable $x(e)$ for each edge e of the graph. The central result of the paper is a theorem showing that any basic solution to the LP will contain a variable $x(e)$ of value at least 1/2. His algorithm builds up a solution by solving the linear programming relaxation, adding to the solution all edges e with $x(e) \geq 1/2$, then iterating on the remaining subproblem. Rounding up each $x(e)$ from 1/2 to 1 gives the performance guarantee of 2 for the algorithm.

No non-trivial approximation algorithm for VC-SNDP is currently known. However, for special cases, there are some known approximation algorithms. In the case $r_{ij} = k$ for all i, j, there is a k-approximation algorithm due to Kortsarz and Nutov [17];[1] furthermore, when edge costs obey the triangle inequality, Khuller and Ragavachari [15] give a constant-factor approximation algorithm. Cheriyan, Vempala, and Vetta [3] give a $6H_k$-approximation algorithm for this problem for graphs that contain at least $6k^2$ vertices. When $r_{ij} \in \{0, 1, 2\}$, Ravi and Williamson [18] give a primal-dual 3-approximation algorithm.

Very recently, Kortsarz, Krauthgamer, and Lee [16] have shed some light on the difficulty of VC-SNDP by showing that it has no polynomial-time algorithm with approximation ratio better than $2^{\log^{1-\epsilon} n}$ for any $\epsilon > 0$ unless $NP \subseteq DTIME(n^{\text{polylog}(n)})$.

1.2 Our Contribution

We generalize Jain's theorem to a new class of two-set functions we call *weakly two-supermodular*. This allows us to give a 2-approximation algorithm for the

[1] Ravi and Williamson [18] had claimed a $2H_k$-approximation algorithm for the case of general edge costs, but there is an error in their paper; see [19] for details.

problem of selecting a minimum-cost set of edges such that there are at least $f(S,S')$ edges from $\delta(S,S') = \{(u,v) \in E : u \in S, v \in S'\}$ when f is a weakly two-supermodular function, and a polynomial-time separation algorithm exists. This result specializes to Jain's result exactly in the case that $f(S,S')$ is non-zero only when $S' = V - S$. In this case, $g(S) = f(S, V - S)$ is a weakly supermodular function. The weakly two-supermodular functions are related to bisupermodular functions, which are the negative of bisubmodular functions. Bisubmodular functions appear as increasing rank functions in [20], and in more general form in [9].

As an application of our theorem, we give a 2-approximation algorithm for the element connectivity problem. This improves on a previously known primal-dual $2H_k$-approximation algorithm for ELC due to Jain et al. [14] (a $2H_k$-approximation algorithm for ELC is also obtained as a special case of an algorithm by Zhao, Nagamochi, and Ibaraki [22]). Our algorithm gives the first constant approximation algorithm for a general survivable network design problem which allows node failures. To achieve this result, we introduce a new integer programming formulation for the element connectivity problem, derived from a formulation of VC-SNDP due to Stoer [21]. One consequence of our 2-approximation is a d-approximation algorithm for the minimum-cost hyperedge connectivity problem [22], where d is the maximum size of a hyperedge.

The connectivity requirement functions for general vertex connectivity are not weakly two-supermodular. Thus our theorem does not apply to these problems. In fact, such a theorem is not possible for general vertex connectivity. We show that there is a family of vertex connectivity problem instances with $|E| = |V|^2/4$ that have a basic solution with $x \leq 1/\sqrt{|E|} = 2/|V| = 1/k$. This implies that one cannot hope to get a polynomial-time algorithm with approximation ratio better than k by using the algorithmic framework of Jain [13].

Subsequent to this example first appearing in [6], Kortsarz, Krauthgamer, and Lee [16] showed that the vertex connectivity problem admits no efficient $2^{\log^{1-\epsilon} n}$ ratio approximation for any fixed $\epsilon > 0$ unless $NP \subseteq DTIME(n^{\mathrm{polylog}(n)})$.

Even when $r_{ij} \in \{0,1,2\}$, the connectivity requirement function for vertex connectivity is not weakly two-supermodular. We extend the definition of weakly two-supermodular further to show that all basic solutions for vertex connectivity problems with $r_{ij} \in \{0,1,2\}$ have an edge e with $x(e) \geq 1/2$.

In related work, Cheriyan and Vempala [2] have also considered problems of selecting a minimum-cost set of edges from pairs of sets, but in the case of directed graphs. In their problems, one must select $f(S,S')$ edges of those edges directed from a vertex in S to a vertex in S'. They consider crossing bisupermodular functions (a generalization of bisupermodular functions), and show any basic solution to the corresponding LP relaxation contains an edge e such that

$x(e) \geq \Omega(1/\sqrt{|E|})$. This model includes uniform vertex connectivity as a special case, but does not include general vertex connectivity. They also show that this is the best possible result for their general model, in the sense that there exists a family of functions f and problem instances for which there is a basic solution with $x \leq O(1/\sqrt{|E|})$.

Our paper is structured as follows. In Section 2, we introduce some notation that we will be using and review Jain's theorem. In Section 3, we give the integer and linear programming formulations for ELC and VC-SNDP, state our main theorems, and show that these theorems give 2-approximation algorithms for ELC and VC-SNDP, respectively. Section 4 contains the proof of the main theorem for ELC. Section 5 contains the proof of the main theorem for VC-SNDP. Section 6 discusses some implementation issues for the 2-approximation algorithms. Section 7 describes a family of examples for general VC-SNDP for which iterative rounding will not yield a constant factor approximation.

2 Preliminaries

Let $G = (V, E)$ be an undirected graph. Let $c(e)$ be a nonnegative cost for each $e \in E$. Let $\delta(S) = \{(u, v) \in E : u \in S, v \notin S\}$. Let $\delta(S, S') = \{(u, v) \in E : u \in S, v \in S'\}$. In this paper, we will only refer to $\delta(S, S')$ when $S \cap S' = \emptyset$. For a set of edges $F \subseteq E$, let $\delta_F(S) = \delta(S) \cap F$ and $\delta_F(S, S') = \delta(S, S') \cap F$. For $x \in \mathbb{R}^{|E|}$, *support* of x is the subset of edges $e \in E$ with $x(e) > 0$. The *characteristic vector* of the support of x is the vector $\chi_x \in \{0,1\}^{|E|}$ with $\chi_x(e) = 1$ if and only if $x(e) > 0$. We let $x(S) = \sum_{e \in \delta(S)} x(e)$ and $x(S, S') = \sum_{e \in \delta(S, S')} x(e)$. Similarly, $x_F(S) = \sum_{e \in \delta_F(S)} x(e)$ and $x_F(S, S') = \sum_{e \in \delta_F(S,S')} x(e)$.

Let $g : 2^V \to \mathbb{N}$. Consider the following integer program:

$$\begin{array}{llcll} \min & \sum_{e \in E} c(e)x(e) & & & \\ \text{s.t.} & x(S) & \geq & g(S), & \forall S \subset V \\ & x(e) & \in & \{0,1\}, & \forall e \in E. \end{array} \qquad (\mathbf{EC\text{-}SNDP})$$

If $x(e) = 1$, e is in the solution; if $x(e) = 0$, it is not. Thus an optimal solution to this integer program finds a minimum-cost set of edges such that there are at least $g(S)$ edges selected from $\delta(S)$ for each $S \subseteq V$. When $g(S) = \max_{i \in S, j \notin S} r_{ij}$, an optimal solution to this integer program gives the solution to the EC-SNDP problem. This is not hard to see, since for any $i, j \in V$, a feasible solution must have at least r_{ij} edges selected from each cut $\delta(S)$ separating i from j; thus, there are at least r_{ij} edge-disjoint paths from i to j.

We say that g is a *weakly supermodular* function if for any sets $S, T \subseteq V$,

$$\begin{aligned} & g(S) + g(T) \\ & \quad \leq \quad \max\{g(S \cup T) + g(S \cap T), g(S - T) + g(T - S)\}. \end{aligned}$$

It is not hard to prove that the function $g(S) = \max_{i \in S, j \notin S} r_{ij}$ is weakly supermodular [11].

The main theorem of Jain's paper [13] concerns basic solutions of the linear programming relaxation of the integer program ($\mathbf{EC\text{-}SNDP}$) in which the integrality constraints $x(e) \in \{0, 1\}$ are replaced by linear constraints $0 \leq x(e) \leq 1$. A *basic* solution to the linear relaxation of ($\mathbf{EC\text{-}SNDP}$) is any solution that satisfies at equality at least d linearly independent inequalities from the system $\{x(S) \geq g(S) \ \forall \ S \subset V; \ 0 \leq x \leq 1\}$, where d is the number of variables in the system; in this system $d = |E|$. [2]

Theorem 2.1 (Jain [13]) *When g is a weakly supermodular function, for every basic feasible solution to the linear program relaxation of* $\mathbf{EC\text{-}SNDP}$*, there is an edge e with $x(e) \geq 1/2$.*

Even though the linear program contains an exponential number of constraints, a basic, optimal solution to this linear program can be found in polynomial time as long as there exists a polynomial-time *separation oracle* [12]. Given any x, a separation oracle either verifies that x is a feasible solution to the linear program or returns a constraint of the LP violated by x.

Given a polynomial-time separation oracle, the 2-approximation algorithm of [13] works as follows. We start with an empty set of edges $F = \emptyset$. We solve the linear programming relaxation of the problem, and add to F all edges e such that $x(e) \geq 1/2$. We then iterate, now solving the linear programming relaxation with $g'(S) = g(S) - |\delta_F(S)|$ and with E replaced with $E - F$. It is not hard to show that if g is weakly supermodular, then so is g'. The algorithm terminates when F is a feasible solution to the problem. Essentially the proof of the performance guarantee compares the cost of the edges added to F in each iteration with the cost that these edges contribute to the cost of the optimal solution to the linear program. Since $x(e) \geq 1/2$, the cost that e contributes to the final solution when included in F is no more than twice its contribution to the linear programming relaxation. Furthermore, since the linear programming relaxation is a lower bound on the cost of an optimal integral solution, this implies that the cost of F is no more than twice optimal. It is easy to give a polynomial-time separation oracle for the function

[2] We refer the reader unfamiliar with the concept of a basic solution of a linear program to a standard textbook on linear programming, such as Chvátal [4].

$g(S)$ that defines EC-SNDP (and for functions g' that arise in later iterations); thus this algorithm is a 2-approximation algorithm for EC-SNDP.

3 Vertex and Element Connectivity

The problems ELC and VC-SNDP are not included in the weakly supermodular model used for EC-SNDP. In this section, we present a new model that includes these problems.

3.1 Formulations

We consider the following linear program, for a function $f : 2^{V \times V} \rightarrow \mathbb{N}$,

$$\begin{array}{lrcl} \min & \sum_{e \in E} c(e)x(e) & & \\ \text{s.t.} & x_E(S, S') & \geq & f(S, S'), \\ & & & \forall S, S' \subset V, S \cap S' = \emptyset, \\ & 0 \leq x(e) & \leq & 1, \qquad \forall e \in E. \end{array} \qquad \textbf{(WTS)}$$

The constraints of our linear program are based on a theorem of Menger (for multiple proofs and references, see [5]).

Theorem 3.1 (Menger) *Let s and t be vertices in a graph G with no edge between them. Then, the minimum number of vertices separating s from t in G is equal to the maximum number of vertex disjoint paths from s to t in G.*

In the case of the element connectivity problem, we let $R \subseteq V$ be the set of terminals and $Q = V - R$ be the set of nonterminals.

Corollary 3.2 *Let $G = (V, E)$ be a graph, with $V = R \cup Q$, $R \cap Q = \emptyset$ and $s, t \in R$. Then, the minimum number of elements in $E \cup Q$ separating s from t in G is equal to the maximum number of element disjoint paths from s to t in G.* ■

Let r_{ij} denote the vertex connectivity requirements between any pair of vertices $i, j \in V$. Menger's theorem says that G satisfies the connectivity requirements if for every subset $X \subseteq E \cup V - \{i, j\}$ with $|X| < r_{ij}$, i and j are in the same connected component of $G - X$. For any two disjoint subsets $S, S' \subset V$, we define the two-set function f_k by $f_k(S, S') := \max\{r_{ij} | i \in S, j \in S'\}$, where $r_{ij} \in \{0, 1, 2, \ldots, k\}$. We assume that $f_k(S, S')$ is undefined if S and S' are not disjoint; and that $f_k(S, S') = 0$ if either S or S' is the empty set and is otherwise well-defined.

Since there can be at most one vertex-disjoint path from S to S' through each vertex in $V - S - S'$, in order to have a feasible solution to the vertex connectivity problem, the number of edges from S to S' must therefore be at least $f_k(S, S') - |V - S - S'|$. Thus we define the two-set function g_k by $g_k(S, S') = f_k(S, S') - |V - S - S'|$. The following is a simple consequence of Menger's Theorem.

Lemma 3.3 *The set of integral solutions to the LP (*$\mathbf{WTS}$*) with the function* $f = g_k$ *equals the set of solutions to the corresponding vertex connectivity problem.* ■

Similarly, suppose r denotes the element connectivity requirements. Then Corollary 3.2 says that G satisfies the element connectivity requirements if for every subset $X \subseteq E \cup Q - \{i, j\}$ with $|X| < r_{ij}$, i and j are in the same connected component of $G - X$. For any two disjoint subsets $S, S' \subset V$, with all remaining vertices being nonterminals (that is, $V - S - S' \subseteq Q$), we define the two-set function f_{elt} by $f_{\mathsf{elt}}(S, S') := \max\{r_{ij} | i \in S \cap R, j \in S' \cap R\}$. We assume that $f_{\mathsf{elt}}(S, S')$ is undefined if S and S' are not disjoint or if $V - S - S' \not\subseteq Q$. We further assume that $r_{ij} \in \mathbb{Z}$ for all $i, j \in R$, and that $f_{\mathsf{elt}}(S, S') = 0$ if either S or S' is the empty set and is otherwise well-defined.

Since there can be at most one element-disjoint path from S to S' through each vertex in $V - S - S'$, in order to have a feasible solution to the element connectivity problem, the number of edges from S to S' must therefore be at least $f_{\mathsf{elt}}(S, S') - |Q - S - S'|$. Thus we define the two-set function g_{elt} by $g_{\mathsf{elt}}(S, S') = f_{\mathsf{elt}}(S, S') - |Q - S - S'|$. The following lemma is a simple consequence of Corollary 3.2.

Lemma 3.4 *The set of integral solutions to the LP (*$\mathbf{WTS}$*) with the function* $f = g_{\mathsf{elt}}$ *equals the set of solutions to the corresponding element connectivity problem.* ■

3.2 Weak Two-Supermodularity

In order to prove an analogous result to Theorem 2.1 for ($\mathbf{WTS}$), we need to define appropriate extensions of weakly supermodular functions that include f_{elt} and f_2, and yet have properties that allow us to prove that the linear program has nice properties.

The following definitions generalize the one-set function notions of submodularity, supermodularity, and weak supermodularity, and are related to the two-set notions of bisubmodularity and bisupermodularity. A two-set function f defined on the set of pairs of disjoint subsets of V that satisfies

$$
\begin{aligned}
& f(S,S') + f(T,T') \\
& \quad \geq \max\{ \quad f(S \cup T, S' \cap T') + f(S \cap T, S' \cup T'), \\
& \qquad\qquad\quad f(S \cap T', S' \cup T) + f(S' \cap T, S \cup T') \; \}
\end{aligned} \tag{1}
$$

will be called *two-submodular*. See Figure 1 for a pictoral representation of the sets involved in this definition. A two-set function f is called *bisubmodular* if it obeys only the "first part" of the inequality, namely,

$$f(S,S') + f(T,T') \geq f(S \cup T, S' \cap T') + f(S \cap T, S' \cup T').$$

If we let $S' = V - S$ and $T' = V - T$, then two-submodularity reduces to submodularity for symmetric one-set functions.

If $-f$ is two-submodular, then f is *two-supermodular*. This definition is equivalent to replacing $\geq$ with $\leq$ and $\max$ with $\min$ in the above definition. A two-set function f is *weakly two-supermodular* if

$$f(S,S') + f(T,T') \tag{2}$$

$$\leq \max\{ \quad f(S \cup T, S' \cap T') + f(S \cap T, S' \cup T'), \tag{3}$$

$$f(S \cap T', S' \cup T) + f(S' \cap T, S \cup T') \; \}. \tag{4}$$

That is, we simply reverse the inequality from our definition of two-submodular functions, without replacing the $\max$ by a $\min$. If we let $S' = V - S$ and $T' = V - T$, weak two-supermodularity reduces to weak supermodularity.

We prove the following theorem in Section 4.

Theorem 3.5 *For any weakly two-supermodular function f, any basic solution to ($\mathbf{WTS}$) has at least one variable e such that $x(e) \geq 1/2$.*

3.3 Approximation Algorithms

Given a polynomial-time separation oracle for ($\mathbf{WTS}$), we describe a 2-approximation algorithm for solving the integer program associated with ($\mathbf{WTS}$). The algorithm is the same as Jain's algorithm described in Section 2. First, we find an optimal, basic solution x^* to ($\mathbf{WTS}$). Let F be the set of all edges e with $x^*(e) \geq 1/2$. Consider the resulting residual LP:

$$
\begin{array}{rrcl}
\min & \sum_{e \in E-F} c(e)x(e) & & \\
\text{s.t.} & x_{E-F}(S,S') & \geq & f(S,S') - |\delta_F(S,S')|, \\
& & & \forall\, S, S' \subset V,\ S \cap S' = \emptyset, \\
& 0 \leq x(e) & \leq & 1, \qquad \forall e \in E - F
\end{array} \tag{\textbf{RES}}
$$

Let F_{res} be the set of edges returned by recursively applying the algorithm to the residual problem. The following theorem shows that the final set of edges returned by the algorithm is within a factor of 2 of an optimal solution. Let z^*_{res} be the optimal value of the residual LP and z^* be the optimal value of $(\mathbf{WTS})$.

Theorem 3.6 *If F_{res} is an integral solution to $(\mathbf{RES})$ with value at most $2z^*_{res}$, then $F_{res} \cup F$ is an integral solution to $(\mathbf{WTS})$ with value at most $2z^*$.*

Proof: (Sketch.) Follows from same arguments as in [13]. ■

In order to apply the algorithm recursively, we need to show the following lemmas.

Lemma 3.7 *The two-set functions $|\delta_F(S, S')|$ and $x(S, S')$ are two-submodular.*

Proof: The proofs in both cases follow from a simple counting argument that shows that any edge counted on the right-hand side of (1) also appears on the left-hand side. ■

Lemma 3.8 *If f is weakly two-supermodular and g is two-submodular, then $f - g$ is weakly two-supermodular.*

Proof: Whichever term of the definition of weak two-supermodularity (3) and (4) achieves the maximum, $-g(S, S') - g(T, T')$ satisfies the same inequality, by (1). Hence $(f - g)(S, S') + (f - g)(T, T')$ satisfies this inequality. ■

Corollary 3.9 *If f is weakly two-supermodular, then $f(S, S') - |\delta_F(S, S')|$ is also weakly two-supermodular for any set of edges $F \subseteq E$.*

The arguments above yield the following theorem.

Theorem 3.10 *Given a polynomial-time separation oracle for $(\mathbf{WTS})$ and $(\mathbf{RES})$, the algorithm above is a 2-approximation algorithm for finding the minimum-cost integer solution to $(\mathbf{WTS})$.*

3.4 Implications for Element and Vertex Connectivity

In order to apply Theorem 3.5 to vertex connectivity problems, we would need to show that f_k and g_k are weakly two-supermodular. For f_k this is not true, even when $k = 2$. Examples in Section 7 show that the modification of Theorem 3.5 to include general g_k does not hold. However, the connectivity functions for element connectivity are weakly two-supermodular, as the following theorem shows. We begin with some notation. Given (S, S'), there is a pair $i \in S \cap R$, $j \in S' \cap R$ that determines the value of $f_{\mathsf{elt}}(S, S')$. Let $i(S, S')$ denote one such i and $j(S, S')$ denote the corresponding j.

Lemma 3.11 *The two-set function f_{elt} is weakly two-supermodular.*

Proof: Before we can start the proof, we need to argue that the function value f_{elt} is well-defined on the arguments of f_{elt} used in the definition of weakly two-supermodular. We assume that S and S' are disjoint, that T and T' are disjoint, that $V - S - S' \subseteq Q$, and that $V - T - T' \subseteq Q$. We need to show (for example) that $f_{\mathsf{elt}}(S \cap T, S' \cup T')$ is well-defined. First, $S \cap T$ and $S' \cup T'$ are disjoint because any element in S is not in S' and any element in T is not in T', so that any element in $S \cap T$ cannot be in $S' \cup T'$. Second, we need to show that any element not in $(S \cap T) \cup (S' \cup T')$ must be in Q. This follows since any element not in these two sets must be in either $V - S - T - S' - T'$ (and thus is certainly in Q) or $S - T - T'$ (and thus in $V - T - T'$, which implies membership in Q) or $T - S - S'$ (and thus in $V - S - S'$, which implies membership in Q). The other cases are similar.

We now show that f_{elt} is weakly two-supermodular. For any pair (T, T'), note that since $V - T - T'$ contains only nonterminals, $i(S, S') \in (S \cap T) \cup (S \cap T')$ and $j(S, S') \in (S' \cap T) \cup (S' \cap T')$. These possible locations are depicted in Figure 2(a).

Call sets $S \cap T$ and $S' \cap T'$ complements, and sets $S \cap T'$ and $S' \cap T$ complements. The set $I = \{i(S, S'), i(T, T'), j(S, S'), j(T, T')\}$ intersects two, three, or all four of these sets. If only two, then these two sets are complements, $f_{\mathsf{elt}}(S, S') = f_{\mathsf{elt}}(T, T')$, and either (3) holds (Figure 2(b)) or (4) holds (Figure 2(c)), depending on the set of complements. If I intersects three sets, then two of these are complements with the property that a requirement vertex for (S, S') is contained in at least one set of the complementary pair, and a requirement vertex for (T, T') is contained in the complementary set. Thus the inequality of (3) or (4) corresponding to this complementary pair holds (Figures 2(d) and (e), respectively). Finally, we have the case when I intersects all four sets, as in Figure 2(f). In this case, select the complementary pair that contains a requirement vertex of pair (i, j) with $r_{ij} = \max\{f_{\mathsf{elt}}(S, S'), f_{\mathsf{elt}}(T, T')\}$. Then the inequality that corresponds to this complementary pair holds. ∎

Corollary 3.12 *The two-set function $g_{\mathsf{elt}}(S, S')$ is weakly two-supermodular.*

Proof: The function $h(S, S') := |V - S - S'|$ is two-submodular, and satisfies (1) with equality. Thus by Lemma 3.8, $f_{\mathsf{elt}}(S, S') - |V - S - S'|$ is weakly two-supermodular. ∎

As noted above, f_2 is not weakly two-supermodular. In Section 5, we prove the following theorem.

Theorem 3.13 *For $f(S, S') := g_2(S, S')$, any basic solution to (**RES**) has at least one edge $e \in E - F$ with $x(e) \geq \frac{1}{2}$.*

By the same arguments as those above, this theorem implies a 2-approximation algorithm for the VC-SNDP when $r_{ij} \in \{0, 1, 2\}$.

Theorem 3.14 *Given a polynomial-time separation oracle for (*$\mathbf{RES}$*) for the function g_2, we have a 2-approximation algorithm for VC-SNDP when $r_{ij} \in \{0, 1, 2\}$.*

In Section 6, we show that we have polynomial-time separation oracles for both ELC and VC-SNDP.

4 Proof of Theorem 3.5

We now turn to the proof of Theorem 3.5, restated here for convenience.

Theorem 3.5 *For any weakly two-supermodular function f, any basic solution to (*$\mathbf{WTS}$*) has at least one variable e such that $x(e) \geq 1/2$.*

Our proof outline follows that of Jain [13]. Before we can sketch the proof outline, we need to define some terms. We say that a pair of sets S, T *cross* if all three of $S \cap T$, $S - T$, and $T - S$ are nonempty. A collection of sets is *laminar* if no pair of sets in the collection cross. Given a feasible solution x to the LP ($\mathbf{EC\text{-}SNDP}$), we say that the constraint corresponding to the set $S \subset V$ is *tight* if $x(S) = f(S)$. The proof in [13] shows first if a pair of tight sets S and T cross, then either $S \cup T$ and $S \cap T$ are tight, or $S - T$ and $T - S$ are tight. Furthermore, there is a linear relationship between the edges with exactly one endpoint in these sets. This "uncrossing lemma" is used to prove that there exists a basic solution corresponding to a laminar collection of tight sets. This laminar collection defines a partial order on the sets of the collection via the subset relation. This partially ordered set (*poset*) has a forest structure, and the forest is used to prove the existence of an edge of value at least 1/2.

Here we prove analogs of the uncrossing lemma (Lemma 4.1) and the laminar basis lemma (Corollary 4.5). We then define an analogous poset of a laminar collection which has a forest structure (Lemmas 4.7, 4.8, 4.9). Given the forest, we invoke a lemma from [13] to prove the existence of the edge e with $x(e) \geq 1/2$. The central technical difficulty lies in deriving the appropriate analogs of the concepts of "cross", "laminar", and the poset when dealing with pairs of sets; and in defining the appropriate analog of "incidence" so that a charging argument similar to that in [13] works. Additionally, proofs that are quite simple for single sets become non-trivial for pairs of sets (e.g. Lemma 4.3).

We now begin the proof. Let x be a feasible solution to ($\mathbf{WTS}$) for a weakly two-supermodular function f. A set pair (S, S') is *tight* if $x(S, S') = f(S, S')$. In particular, we will be interested in the case when x is a basic solution to $(\mathbf{WTS})$ with the property that $x(e) < 1$ for all $e \in E$ (since otherwise Theorem 3.5 holds trivially). In this case, we define E_x to be the set of edges with nonzero x-value. From this point onward, x will always refer to a basic solution to $(\mathbf{WTS})$. Given x, define $\chi_x(S, S') \in \{0,1\}^{|E|}$ to be the characteristic vector of the support of $x(S, S')$. Recall that this implies $\chi_x(S, S')(e) = 1$ if and only if $x(S, S')(e) > 0$.

Lemma 4.1 (Uncrossing Lemma) *If (S, S') and (T, T') are tight for a weakly two supermodular function f with respect to x, then one of the following holds.*

i. *$(S \cap T, S' \cup T')$ and $(S \cup T, S' \cap T')$ are tight, and*
$\chi_x(S, S') + \chi_x(T, T') = \chi_x(S \cup T, S' \cap T') + \chi_x(S \cap T, S' \cup T')$,

ii. *$(S \cap T', S' \cup T)$ and $(S' \cap T, S \cup T')$ are tight and*
$\chi_x(S, S') + \chi_x(T, T') = \chi_x(S \cap T', S' \cup T) + \chi_x(S' \cap T, S \cup T')$.

Proof: Because f is weakly two-supermodular, either (3) or (4) holds for (S, S') and (T, T'); suppose (3) holds. That is, $f(S, S') + f(T, T') \leq f(S \cup T, S' \cap T') + f(S \cap T, S' \cup T')$. (The other case is similar.) We know that $x(S, S')$ is two-submodular by Lemma 3.7. Define $h(Z, Z') = f(Z, Z') - x(Z, Z')$. Thus $h(S, S') + h(T, T') \leq h(S \cup T, S' \cap T') + h(S \cap T, S' \cup T')$. By the feasibility of x, $x(X, X') \geq f(X, X')$, so that $h(X, X') \leq 0$ for all set pairs (X, X'). For (S, S') and (T, T') that are tight for f with respect to x, $h(S, S') = h(T, T') = 0$. Thus we know $h(S, S') + h(T, T') = 0$, $h(S \cup T, S' \cap T') \leq 0$, and $h(S \cap T, S' \cup T') \leq 0$, so it must be the case that $h(S \cup T, S' \cap T') = h(S \cap T, S' \cup T') = 0$, and $(S \cup T, S' \cap T')$ and $(S \cap T, S' \cup T')$ are tight. It is not hard to show that $\chi_x(S, S') + \chi_x(T, T') \geq \chi_x(S \cup T, S' \cap T') + \chi_x(S \cap T, S' \cap T')$ by showing that every edge that appears in the right-hand side of the inequality must appear in the left-hand side. We need to show that equality holds; suppose not — the inequality is strict. Since the x-value of an edge e is the same in all sets that contain e, the relation (i.e. $<$, $>$, or $=$) of $x(S, S') + x(T, T')$ to $x(S \cup T, S' \cap T') + x(S \cap T, S' \cap T')$ is the same as the relation of $\chi_x(S, S') + \chi_x(T, T')$ to $\chi_x(S \cup T, S' \cap T') + \chi_x(S \cap T, S' \cap T')$. Hence $\chi_x(S, S') + \chi_x(T, T') > \chi_x(S \cup T, S' \cap T') + \chi_x(S \cap T, S' \cap T')$ implies that $x(S, S') + x(T, T') > x(S \cup T, S' \cap T') + x(S \cap T, S' \cap T')$. But then $0 = h(S, S') + h(T, T') < h(S \cup T, S' \cap T') + h(S \cap T, S' \cup T') = 0$, a contradiction. ■

We define a relation $\leq$ on set pairs by $(S, S') \leq (T, T')$ if $S \subseteq T$ and $S' \supseteq T'$. It is easy to check that this relation is transitive, reflexive, and anti-symmetric, and hence defines a partial order.

Definition 4.2 *The pairs (S, S') and (T, T') pair-cross if they do not satisfy either of the following two conditions: (1) (S, S') and (T, T') are comparable in the* *partial order (that is, either $(S, S') \leq (T, T')$ or vice versa); (2) $S \subseteq T'$ and $T \subseteq S'$. A collection $\mathcal{L}$ of pairs (S, S') is called pair-laminar if no two pairs in $\mathcal{L}$ pair-cross.*

It is important to note that this definition is not symmetric: It is possible that (S, S') and (T, T') do not pair-cross while (S', S) and (T', T) do pair-cross. In this case, the uncrossing lemma applied to (S', S) and (T', T) will return (S, S') and (T, T'). This asymmetry is necessary in the following sense: if we use a perhaps more natural definition of crossing that defines two set pairs to cross if any two of their defining sets cross, then it may be the case that we have crossing set pairs, but that after applying the uncrossing lemma, we remain with the same two set pairs. For instance, consider the case when $T \subset S$, but S and T' cross. This happens when $S \not\subseteq T \cup T'$. In this case, condition (ii) of the uncrossing lemma does not hold, as $S' \cap T = \emptyset$; but condition (i) of the uncrossing lemma does hold. However, $S \cup T = S$ and $S \cap T = T$, so that the same two set pairs are returned after applying the uncrossing lemma. Thus we must relax this definition so that every application of the uncrossing lemma yields two set pairs which do not pair-cross. We cannot relax it too much, however, since we require specific properties of pair-laminar set pairs in order to obtain our result. In particular, a pair-laminar collection of set pairs must define a poset which has a forest structure. We prove that this is the case in Lemma 4.7. The second condition of pair-cross is also used critically in Lemma 4.8.

The following technical lemma is central to the validity of our main theorem. The proof contains many cases because the definition of pair-cross is not symmetric.

Lemma 4.3 *Let (S, S') and (T, T') be pairs that pair-cross. If (X, X') does not pair-cross either (S, S') or (T, T'), then it does not pair-cross any of the pairs:*
a*: $(S \cap T, S' \cup T')$,*
b*: $(S \cup T, S' \cap T')$,*
c*: $(S \cap T', S' \cup T)$,*
d*: $(S' \cap T, S \cup T')$.*

Proof: We note that if $(S, S') \leq (X, X')$ then it follows immediately that $(S \cap T, S' \cup T') \leq (S, S') \leq (X, X')$ and that $(S \cap T', S' \cup T) \leq (S, S') \leq (X, X')$; that is, (X, X') does not pair-cross **a** or **c**. Similarly, if $(S, S') \geq (X, X')$ then $(S \cup T, S' \cap T') \geq (S, S') \geq (X, X')$, and (X, X') does not pair-cross $(S' \cap T, S \cup T')$ since $S' \cap T \subseteq S' \subseteq X'$ and $X \subseteq S \subseteq S \cup T'$, and thus (X, X') does not pair-cross **b** or **d**. Similarly, if $(T, T') \leq (X, X')$ then $(S \cap T, S' \cup T') \leq (X, X')$ and $(S' \cap T, S \cup T') \leq (X, X')$, so that (X, X') does not pair-cross **a** or **d**. If

$(T,T') \geq (X,X')$ then $(S \cup T, S' \cap T') \geq (X,X')$ and (X,X') does not pair-cross $(S \cap T', S' \cup T)$ since $X \subseteq S' \cup T$ and $S \cap T' \subseteq X'$, so that (X,X') does not pair-cross **b** or **c**.

We now exhaustively enumerate cases. If both $(S,S'),(T,T') \leq (X,X')$, then by the above (X,X') does not pair-cross **a**, **c**, or **d**. Furthermore, it does not pair-cross **b** since we know $S,T \subseteq X$ and $S',T' \supseteq X'$, and therefore $(S \cup T, S' \cap T') \leq (X,X')$.

If both $(S,S'),(T,T') \geq (X,X')$, then by the above (X,X') does not pair-cross **b**, **c**, or **d**. It does not pair-cross **a** since we know $S,T \supseteq X$ and $S',T' \subseteq X'$ and therefore $(S \cap T, S' \cup T') \geq (X,X')$.

The case $(S,S') \leq (X,X') \leq (T,T')$ or the reverse cannot occur because this implies that (S,S') and (T,T') do not pair-cross, contradicting the hypothesis.

Note that because (S,S') and (T,T') pair-cross, it cannot be the case that $(S,S') \leq (X,X')$ and $X \subseteq T'$ and $T \subseteq X'$ since this implies that $S \subseteq T'$ and $T \subseteq S'$. Similarly, this case with (S,S') interchanged with (T,T') cannot occur. So we assume $(S,S') \geq (X,X')$ and that $X \subseteq T'$ and $T \subseteq X'$. From the above, we know that (X,X') does not pair-cross **b** or **d**. Furthermore, it does not pair-cross **a** since $X \subseteq S' \cup T'$ and $S \cap T \subseteq X'$. It does not pair-cross **c** since $(X,X') \leq (S \cap T', S' \cup T)$. The case with (S,S') and (T,T') interchanged is similar.

Finally, we assume that (X,X') does not pair-cross (S,S') because $X \subseteq S'$ and $S \subseteq X'$, and does not pair-cross (T,T') because $X \subseteq T'$ and $T \subseteq X'$. It follows easily that (X,X') does not pair-cross **a** because $X \subseteq S' \cup T'$ and $S \cap T \subseteq X'$. It does not pair-cross **b** because $X \subseteq S' \cap T'$ and $S \cup T \subseteq X'$. It does not pair-cross **c** because $X \subseteq S' \cup T$ and $S \cap T' \subseteq X'$, and it does not pair-cross **d** because $X \subseteq S \cup T'$ and $S' \cap T \subseteq X'$. ■

Lemma 4.4 *Given pairs* (S,S') *and* (T,T')*, neither* (S,S') *nor* (T,T') *pair-cross any of the four pairs* $(S \cap T, S' \cup T')$, $(S \cup T, S' \cap T')$, $(S \cap T', S' \cup T)$, $(S' \cap T, S \cup T')$.

Proof: Consider (S,S'); the proof for (T,T') is analogous. We see that $(S \cap T, S' \cup T') \leq (S,S')$, $(S,S') \leq (S \cup T, S' \cap T')$, $(S \cap T', S' \cup T) \leq (S,S')$, and $S \subseteq S \cup T'$ and $S' \cap T \subseteq S'$. Thus (S,S') does not cross any of these four sets. ■

Let $\mathcal{T}$ be the collection of all tight set pairs with respect to x. Given a collection of set pairs $\mathcal{A}$, we define $\mathrm{Span}(\mathcal{A})$ to be the vector space spanned by the characteristic vectors $\chi_x(S,S')$ of the set pairs $(S,S') \in \mathcal{A}$.

Lemma 4.5 *Any maximal pair-laminar collection* $\mathcal{L}$ *of tight set pairs satisfies* $\mathrm{Span}(\mathcal{L}) = \mathrm{Span}(\mathcal{T})$.

Proof: If $\mathrm{Span}(\mathcal{L}) \neq \mathrm{Span}(\mathcal{T})$, then $\mathrm{Span}(\mathcal{L}) \subset \mathrm{Span}(\mathcal{T})$ since $\mathcal{L} \subset \mathcal{T}$. Thus there exists a pair $(S, S') \in \mathcal{T}$, with $\chi_x(S, S') \notin \mathrm{Span}(\mathcal{L})$, such that (S, S') pair-crosses a minimum number of set pairs in $\mathcal{L}$. Let (T, T') be one of those pairs. Then by Lemma 4.1, we can rewrite $\chi_x(S, S')$ as a linear combination of characteristic vectors of pair-laminar tight set pairs. Suppose that condition (i) of Lemma 4.1 holds; thus we can rewrite $\chi_x(S, S')$ as $\chi_x(S \cup T, S' \cap T') + \chi_x(S \cap T, S' \cup T') - \chi_x(T, T')$ (the case in which condition (ii) holds is similar). Since $\chi_x(S, S') \notin \mathrm{Span}(\mathcal{L})$, at least one of $\chi_x(S \cup T, S' \cap T')$ and $\chi_x(S \cap T, S' \cup T')$ is also not in $\mathrm{Span}(\mathcal{L})$. By Lemma 4.3, any set pair $(X, X') \in \mathcal{L}$ pair-crossing either of $(S \cup T, S' \cap T')$ and $(S \cap T, S' \cup T')$ must also have pair-crossed (S, S'), since (X, X') does not pair-cross (T, T') by the laminarity of $\mathcal{L}$. Furthermore, by Lemma 4.4 the set pairs $(S \cup T, S' \cap T')$ and $(S \cap T, S' \cup T')$ do not pair-cross (T, T'). Since these set pairs do not pair-cross (T, T'), they have strictly fewer crossings with sets in $\mathcal{L}$ than (S, S') does, contradicting the choice of (S, S'). ■

Corollary 4.6 *For any basic solution x, there exists a pair-laminar set $\mathcal{B}$ of tight set pairs satisfying*

1. $|\mathcal{B}| = |E_x|$,
2. *the vectors $\chi_x(S, S')$ for $(S, S') \in \mathcal{B}$ are linearly independent,*
3. $f(S, S') \geq 1$ *for all* $(S, S') \in \mathcal{B}$. ■

Proof: Let $\mathcal{L}$ be a maximal pair-laminar collection of tight set pairs. If there is a set pair in this set that is linearly dependent on other members of the set, remove it to form $\mathcal{B}$. Note that $\mathcal{B}$ satisfies the second and third conditions; and since the span is unchanged, Span($\mathcal{B}$) = Span($\mathcal{L}$) = Span($\mathcal{T}$) by Lemma 4.5. Since x is basic, the dimension of Span($\mathcal{T}$) is $|E_x|$. Thus, $\mathcal{B}$ also satisfies the first condition. ■

Lemma 4.7 *If $\mathcal{B}$ is a collection of pair-laminar set pairs, then the poset defined by $\leq$ on the set pairs in $\mathcal{B}$ is described by a unique forest.*

Proof: Let $(S, S'),\ (X, X'),\ (Y, Y') \in \mathcal{B}$. To show that the partial order defined by $\leq$ gives a forest, we only need show that if $(S, S') \leq (X, X')$ and $(S, S') \leq (Y, Y')$, then (X, X') and (Y, Y') must be comparable in the partial order. Thus it suffices to establish that if

(1) $(S, S') < (X, X')$,
(2) $(S, S') < (Y, Y')$, and
(3) $(Y, Y') \not< (X, X')$,

then $(X, X') < (Y, Y')$. Since $\mathcal{B}$ is pair-laminar, (X, X') and (Y, Y') do not pair-cross. If $(X, X') \not< (Y, Y')$, then by laminarity, $X \subseteq Y'$ and $Y \subseteq X'$. Since

$S \subseteq X \subseteq Y'$, and since Y and Y' are disjoint, this implies $S \not\subseteq Y$, contradicting $(S, S') \leq (Y, Y')$. So it must be the case that $(X, X') \leq (Y, Y')$. ■

To prove Theorem 3.5, we use a proof by contradiction. We use Lemma 4.7 to construct a forest of tight set pairs in $\mathcal{B}$. We then define a new concept of incidence of edges with fractional value to nodes in this forest. Given that no edge has value at least $\frac{1}{2}$, we can then charge edges with fractional value to the nodes in this forest in a way that leads to a contradiction.

Start with a pair-laminar family $\mathcal{B}$ as given by Corollary 4.6. Form the rooted forest corresponding to the containment poset on $\mathcal{B}$ as indicated in Lemma 4.7: the node set is $\mathcal{B}$ and there is an arc from $(S, S') \in \mathcal{B}$ to $(T, T') \in \mathcal{B}$ if (T, T') is the smallest pair of the partial order such that $(S, S') \neq (T, T')$ and $(S, S') \leq (T, T')$. Note that we refer to *nodes* and *arcs* of the forest, while we use *vertices* and *edges* when referring to the original graph.

A *socket* (e, i) is a pairing of an edge $e \in E_x$ with one of its two endpoints. Each edge $e = (i, j) \in E_x$ is associated with two sockets: (e, i) and (e, j). We assign each socket to at most one node of the forest; we say that the socket is *incident* to that node. For an edge $e = (i, j) \in E_x$, the socket (e, i) is incident to node (S, S') if (S, S') is the lowest node in the tree among all nodes with either $i \in S$ or $\{i, j\} \cap S' = \emptyset$.

Lemma 4.8 *Incidence is well-defined.*

Proof: We need to show that for a given socket (e, i), the definition of incidence is such that the socket is assigned to at most one node. It suffices to show that any two set pairs $(S, S'), (T, T') \in \mathcal{B}$ for which one of the two conditions of incidence holds must be comparable in the partial order.

First, if $i \in S$ and $i \in T$ then since (S, S') and (T, T') don't pair-cross, it must be the case that either $(S, S') \leq (T, T')$ or $(T, T') \leq (S, S')$.

Now suppose that $i \in S$, $i \notin T$, and $\{i, j\} \cap T' = \emptyset$. We know that (S, S') and (T, T') do not pair-cross. Since $i \in S$ and $i \notin T$, we know $(S, S') \not\leq (T, T')$. Since $i \in S$ and $i \notin T'$, it is not the case that $S \subseteq T'$ and $T \subseteq S'$. Thus it must be the case that $(T, T') \leq (S, S')$.

Suppose that $i \notin S$, $i \notin T$, $\{i, j\} \cap S' = \emptyset$, and $\{i, j\} \cap T' = \emptyset$, and that (S, S') and (T, T') are incomparable in the partial order. Since (S, S') and (T, T') do not pair cross, it must be that $j \notin S \cup T$. Thus $i, j \in V - (S \cup T \cup S' \cup T')$. We claim there is no set pair $(Z, Z') \in \mathcal{B}$ with $|\{i, j\} \cap Z| = 1$ and $|\{i, j\} \cap Z'| = 1$, since such a set pair would pair-cross with (S, S'). But then the edge (i, j) is not in the support of $x(Z, Z')$ for any $(Z, Z') \in \mathcal{B}$. Thus, $(i, j) \notin E_x$. ■

We say that an edge *crosses* a node (S, S') if exactly one of its associated sockets is incident to any node in the subtree rooted at (S, S').

Lemma 4.9 *An edge (i,j) with fractional value crosses a node (S,S') if and only if* $|\{i,j\} \cap S| = |\{i,j\} \cap S'| = 1$.

Proof: ($\Rightarrow$): For edge $e = (i,j)$ crossing node (S,S'), assume that the socket (e,i) is incident to a node in the subtree rooted at (S,S') and (e,j) is not. Then $i \notin S'$ and $j \in S'$ by definition of incidence and the properties of the partial order. Since $j \in S'$, this in turn implies $i \in S$. ($\Leftarrow$): If $i \in S$ and $j \in S'$ and $\mathcal{B}$ is pair-laminar, then by Lemma 4.8, (e,i) is incident to a node (T,T') satisfying $(T,T') \leq (S,S')$. By Lemma 4.7, this implies that (e,i) is incident to a node in the subtree rooted at (S,S'). Since $j \in S'$, (e,j) is not incident to any $(T,T') \leq (S,S')$. ■

Proof of Theorem 3.5: The proof is by contradiction: suppose every edge takes value strictly less than $\frac{1}{2}$. We show that given this assumption, we can "charge" the sockets incident to any rooted subtree of the forest in such a way that each node gets charged at least two sockets and the root gets charged at least 3 sockets. This leads to a contradiction, since the number of sockets is twice the number of edges, and the number of edges equals the number of nodes in the forest.

This charging scheme is carried out inductively bottom up on the structure of the tree. Consider first a leaf element (S,S') of the forest. Since we know $f(S,S') \geq 1$ and each edge has value less than 1/2, it must be the case that $|\delta_{E_x}(S,S')| \geq 3$. By Lemma 4.9 it must be the case that each of these edges cross (S,S'), which implies that at least three sockets are incident on (S,S'). We invoke a lemma of Jain below in order to carry out the induction.

For $e \in E_x$, define $y(e) = \frac{1}{2} - x(e)$. Since $x(e) < 1/2$ by hypothesis, note that $y(e) > 0$. For any pair $(S,S') \in \mathcal{B}$, define its *co-requirement* as the sum of $y(e)$'s for all the edges crossing (S,S'). The co-requirement satisfies $y(S,S') = \frac{1}{2}|\delta_{E_x}(S,S')| - f(S,S')$. Since $f(S,S')$ is integral, the co-requirement of (S,S') is an integral multiple of $\frac{1}{2}$.

Lemma 4.10 (Jain [13], Lemma 4.6) *For any rooted subtree of the forest, we can charge the sockets incident to it such that every node gets charged at least 2 sockets and the root gets charged at least 3. Moreover, the root gets charged exactly 3 sockets only if its co-requirement is half.*

Note that the statement of the lemma in [13] refers to endpoints in a way that is analogous to our use of the term sockets here. The proof of this lemma for our problem is exactly the same proof in [13] which consists of checking a series of cases. We omit it here due to the similarity to, and the length of, the original. ■

5 $\{$0,1,2$\}$ Vertex Connectivity

In order to show that the algorithm described in Section 3.3 is a 2-approximation algorithm for VC-SNDP when $r \in \{0, 1, 2\}$, it suffices to prove Theorem 3.13, restated here:

Theorem 3.13 *For $f(S, S') := g_2(S, S') - |\delta_F(S, S')|$ and $E = E - F$, any basic solution to (**RES**) has at least one edge $e \in E - F$ with $x(e) \geq \frac{1}{2}$.*

As noted in Section 3.4, the functions f_2 and g_2 are not weakly two-supermodular, so Theorem 3.5 does not apply directly. However, since their function values are bounded by 2, we can extend the definitions and techniques used to prove Theorem 3.5 to handle these functions. We start by extending the definition of weak two-supermodularity, and showing that f_2 and g_2 are covered by this extension. In Section 5.2, we show that our extension of weak two-supermodularity is sufficient to obtain an uncrossing lemma. Once the uncrossing lemma is established, the rest of the proof of Theorem 3.13 follows the same arguments as used in the proof of Theorem 3.5.

5.1 Very Weak Two-Supermodularity

A two-set function f is *very weakly two-supermodular* if whenever $f(S, S') > 0$ and $f(T, T') > 0$ then

$$
\begin{aligned}
& f(S,S') + f(T,T') \leq \\
& \quad \max\{ \quad f(S\cup T, S'\cap T') + f(S\cap T, S'\cup T'), && (5)\\
& \qquad\qquad f(S\cap T', S'\cup T) + f(S\cup T', S'\cap T), && (6)\\
& \qquad\qquad \max\{ f(S\cup T, S'\cap T'), f(S\cap T, S'\cup T'), f(S\cap T', S'\cup T), \\
& \qquad\qquad\qquad f(S\cup T', S'\cap T) \}, && (7)\\
& \quad \textbf{OR} \quad \text{for some permutation of } S \text{ and } S', T \text{ and } T', \\
& \qquad\qquad f(S\cap T, S'\cup T') + f(S\cup T', S'\cap T) \}
\end{aligned}
$$

While it may seem that (7) is included in (5) and (6), if f is allowed to take on negative values, this may not be the case.

We say that f is symmetric if $f(S, S') = f(S', S)$ for all $S \subseteq V$. If $f \leq \mathbf{2}$, then the only meaningful values of $|V - S - S'|$ for the definition of g_2 are 0 or 1. If in addition, f is symmetric, as is the case for f_2 and g_2, then the definition of very weakly two-supermodular can be simplified to be that for all (S, S') and

(T,T') with $S \subseteq T \cup T'$ and $T \subseteq S \cup S'$,

$$
\begin{aligned}
f(S,S') + f(T,T') \quad \leq \max\{ \quad & f(S\cup T, S'\cap T') + f(S\cap T, S'\cup T'), && (8)\\
& f(S\cup T', S'\cap T) + f(S\cap T', S'\cup T), && (9)\\
& f(S\cap T, S'\cup T'), && (10)\\
& \textbf{OR} \text{ by perhaps swapping } (S,S') \text{ for } (T,T'), \\
& f(S\cap T, S'\cup T') + f(S\cup T', S'\cap T)\ \}. && (11)
\end{aligned}
$$

To show that g_2 is very weakly two-supermodular, we first introduce some notation. Given (S,S'), there is a pair $i \in S$, $j \in S'$ that determines $f_2(S,S')$. Let $i(S,S')$ denote one such i and $j(S,S')$ denote the corresponding j. Call the ordered pair $(i(S,S'), j(S,S'))$ a *witness* for $f_2(S,S')$.

Lemma 5.1 *The two-set function g_2 is very weakly two-supermodular.*

Proof: Let S and T be subsets of V satisfying $S \subseteq T\cup T'$, $T \subseteq S\cup S'$. The proof is a case analysis on the location of the four vertices $i(S,S')$, $j(S,S')$, $i(T,T')$, $j(T,T')$. Note that, by definition of very weakly supermodular, we only need to satisfy one of (8)-(11) when both $g_2(S,S')$ and $g_2(T,T')$ are strictly positive. This immediately implies that both $f_2(S,S')$ and $f_2(T,T')$ are in $\{1,2\}$; and if $|V-S-S'| = 1$, this means that $f_2(S,S') = 2$ and $g_2(S,S') = 1$.

Case (i): $S\cup S' = T \cup T' = V$. By the weak supermodularity of the one-set function f' defined by $f'(S) := \max\{r_{ij} | i \in S, j \in V\backslash S\}$ used in [11, 13] implies that either (8) or (9) hold.

Case (ii): $|V-S-S'| = |V-T-T'| = 1$. Then we need only check the case when $f_2(S,S') = f_2(T,T') = 2$, implying $g_2(S,S') = g_2(T,T') = 1$. In this case, either

(a) $i(S,S') \in S\cap T'$ and $i(T,T') \in T \cap S'$ (Figure 3(a)),

(b) $j(S,S') \in T\cap S'$, and $j(T,T') \in S\cap T'$ (Figure 3(b)), or

(c) $\{i(S,S'), i(T,T')\} \cap (S\cap T) \neq \emptyset$ and at least one of $j(S,S'), j(T,T')$ is in $(S'\cap T') \cup (V-S-S') \cup (V-T-T') = V-S-T$.

If (a) holds, then $(i(S,S'), j(S,S'))$ is a witness for $f_2(S\cap T', S'\cup T)$ and $(i(T,T'), j(T,T'))$ is a witness for $f_2(S'\cap T, S\cup T')$. Here, $|V-(S\cap T') - (S'\cup T)| = |V-(S'\cap T) - (S\cup T')| = 1$ so (9) holds. If (b) holds, then $(j(T,T'), i(T,T'))$ is a witness for $f_2(S\cap T', S'\cup T)$ and $(j(S,S'), i(S,S'))$ is a witness for $f_2(S'\cap T, S\cup T')$, and again (9) holds.

If (a) and (b) do not hold and (c) holds, then least one of $(i(S,S'), j(S,S'))$, $(i(T,T'), j(T,T'))$ is a witness for $f_2(S\cap T, S'\cup T')$. If $V-S-S' = V-T-T'$, then $j(S,S') \neq V-T-T'$ and $j(T,T') \neq V-S-S'$ so that $\{j(S,S'), j(T,T')\} \cap (S'\cap T')$ is nonempty and one of $(i(S,S'), j(S,S'))$, $(i(T,T'), j(T,T'))$ is also a witness for $f_2(S \cup T, S' \cap T')$. (See Figure 3(c1).) Since in this case, $|V - (S \cap T) - (S' \cup T')| = |V - (S \cup T) - (S' \cap T')| = 1$, we have that (8) holds. Otherwise, as depicted in Figure 3(c2), $|V - (S \cap T) - (S' \cup T')| = \emptyset$, so that $g_2(S \cap T, S' \cup T') = 2$ and (10) holds.

Case (iiia): $|V - S - S'| = 1, T \cup T' = V, i(S, S') \in S \cap T'$. If either $j(S, S')$ or $i(T, T') \in T \cap S'$, then (9) holds. Otherwise, $(i(S, S'), j(S, S'))$ is a witness for $f_2(S \cup T, S' \cap T')$, $(i(T, T'), j(T, T'))$ is a witness for $f_2(S \cap T, S' \cup T')$, and (8) holds.

Case (iiib): $|V - S - S'| = 1, T \cup T' = V, i(S, S') \in S \cap T$. If $j(S, S') \in S' \cap T'$, then (8) holds. Otherwise, $j(S, S') \in T \cap S'$. This situation is depicted at the bottom of Figure 3. In this case, $(i(S, S'), j(S, S'))$ is a witness for $f_2(S \cap T, S' \cup T')$; $(j(S, S'), i(S, S'))$ is a witness for $f_2(S' \cap T, S \cup T')$; and thus (11) holds.

Case (iv): $|V - T - T'| = 1, S \cup S' = V$. Argument is equivalent to Case (iii) once (S, S') is switched with (T, T'). ■

The proof of Lemma 5.1 demonstrates why and when we require (11) in the description of g_2. We summarize this in the following corollary so that we may easily refer to it. It is also partially depicted at the bottom of Figure 3.

Corollary 5.2 *If $S \subseteq T \cup T'$, $T \subseteq S \cup S'$, $g_2(S, S')$ and $g_2(T, T')$ are strictly positive, and $g_2(S, S') + g_2(T, T')$ is strictly greater than the maximum of (8)-(10), then by possibly swapping (S, S') for (T, T'), we have that $g_2(S, S') + g_2(T, T') \leq g_2(S \cap T, S' \cup T') + g_2(S \cup T', S' \cap T)$, $|V - S - S'| = 1$, $T \cup T' = V$, $i(S, S') \in S \cap T$, and $j(S, S') \in T \cap S'$.* ■

The example in Section 7 shows that the corresponding g_k for $r \in \{1, k\}^{V \times V}$ is in general not very weakly two-supermodular for any $k \geq 3$.

Lemma 5.3 *For any edge set F on V, $g_2(S, S') - |\delta_F(S, S')|$ is very weakly two-supermodular.*

Proof: Since the proof of the lemma is independent of choice of F, and the context is clear, we use δ for δ_F. Suppose $g_2(S, S') > 0$ and $g_2(T, T') > 0$. If $g_2(S, S') + g_2(T, T')$ satisfies any of (8)-(10), then by the two-submodularity of $|\delta|$, we have $(g_2 - |\delta|)(S, S') + (g_2 - |\delta|)(T, T')$ satisfies the same inequality. If $g_2(S, S') + g_2(T, T')$ does not satisfy (8)-(10), then it satisfies (11) (after possibly swapping (S, S') and (T, T')). If $|\delta(S, S')| + |\delta(T, T')| \geq |\delta(S \cap T, S' \cup T')| + |\delta(S \cup T', S' \cap T)|$, then $g_2 - |\delta|$ also satisfies (11), and we are done. Otherwise, there is an edge from $S \cap T$ to $T \cap S'$ in F, since this is the only type of edge that contributes more to the right hand side of (11) than the left. Thus $|\delta(S, S')| \geq 1$.

Since $g_2(S,S') + g_2(T,T')$ does not satisfy (8)-(10), by Corollary 5.2, we have that $|V - S - S'| = 1$ and $g_2(S,S') = 1$, and thus $g_2(S,S') - |\delta(S,S')| \leq 0$. Hence, since (8)-(10) only need to be satisfied by a very weak two supermodular function h on arguments (S,S') with $h(S,S') > 0$, none of (8)-(11) need apply to (S,S') to establish very weak two-supermodularity of $g_2 - |\delta|$. ■

5.2 Proof of Theorem 3.13

This section is devoted to the the proof of Theorem 3.13. To prove this theorem, we will use the fact that $g_2 - |\delta_F|$ is very weakly two-supermodular to obtain an analog to the uncrossing lemma, Lemma 4.1. The uncrossing lemma is in turn useful to establish the existence of a laminar basis of tight sets, a key component in the proof of the theorem.

Let x be the extension of a feasible solution to ($\mathbf{RES}$) with $f = g_2$ such that $x(e) = 1$ for all $e \in F$. In particular we will be interested in the case when x is a basic solution to ($\mathbf{RES}$) with the property that $x(e) < 1$ for all $e \in E - F$. In this case, let E_x be the set of edges with nonzero x-value. A pair (S,S') is *tight* if it satisfies

$$x_{E-F}(S,S') \geq f_2(S,S') - |V - S - S'| - |\delta_F(S,S')| \tag{12}$$

at equality. (Equivalently, $x_E(S,S') = f_2(S,S') - |V - S - S'| = g_2(S,S')$.) Given x, define $\chi_x(S,S') \in \{0,1\}^{|E-F|}$ to be the characteristic vector of the support of $x_{E-F}(S,S')$. If $x_{E-F}(S,S') = 0$, we say that (S,S') is *empty*. If $x_{E-F}(S,S') > 0$, then (S,S') is *non-empty*. If (S,S') is empty, then $\chi_x(S,S') = \mathbf{0}$.

Lemma 5.4 (Uncrossing Lemma B) *If (S,S') and (T,T') are tight and non-empty, then for the appropriate permutation of S and S' and T and T' so that $S \subseteq T \cup T'$ and $T \subseteq S \cup S'$ one of the following holds.*

(i) $(S \cap T, S' \cup T')$ *is tight,* $(S \cup T, S' \cap T')$ *is either empty or tight, and*
$\chi_x(S,S') + \chi_x(T,T') = \chi_x(S \cap T, S' \cup T') + \chi_x(S \cup T, S' \cap T')$*,*

(ii) $(S \cap T', S' \cup T)$ *and* $(S \cup T', S' \cap T)$ *are tight and*
$\chi_x(S,S') + \chi_x(T,T') = \chi_x(S \cap T', S' \cup T) + \chi_x(S \cup T', S' \cap T)$*,*

(iii) *After perhaps swapping* (S,S') *for* (T,T')*, then* $T \cup T' = V$*, and* $(S \cap T, S' \cup T')$ *and* $(S \cup T', S' \cap T)$ *are tight, and*
$2\chi_x(S,S') + \chi_x(T,T') = \chi_x(S \cap T, S \cup T') + \chi_x(S \cup T', S' \cap T)$*.*

Proof: For simplicity of notation, let $g' = g_2 - |\delta_F|$. Since x_{E-F} is a solution to (**RES**), $g' - x_{E-F} \leq 0$. Since g' is very weakly two-supermodular by Corollary 3.12, if (S, S') and (T, T') are both nonempty, then for appropriate permutations of S and S', T and T', we have that $g'(S, S') + g'(T, T')$ must satisfy one of (8)-(11) with g_2 replaced by g'. If it satisfies any of (8)-(10), then since x_{E-F} is two-submodular, $(g' - x_{E-F})(S, S') + (g' - x_{E-F})(T, T')$ satisfies the same inequality. Thus if (S, S') and (T, T') are tight, then the left hand side of the corresponding inequality in (8)-(10) equals 0. Since $g' - x_{E-F} \leq 0$, this implies that each part of the corresponding right hand side equals 0. Thus, if (8) is satisfied, then (i) holds with $(S \cup T, S' \cap T')$ tight; if (9) is satisfied then, (ii) holds; and if (10) is satisfied, then (i) holds with $(S \cup T, S' \cap T')$ empty.

If $g'(S, S') + g'(T, T')$ does not satisfy any of (8)-(10), then neither does g_2 and by Corollary 5.2, by perhaps swapping (S, S') for (T, T'), we have that $|V - S - S'| = 1$, $T \cup T' = V$, $g_2(S, S') = 1$, $i(S, S') \in S \cap T$ and $j(S, S') \in S' \cap T$. See bottom of Figure 3. Since $|V - S - S'| = 1$, we have that $f_2(i(S, S'), j(S, S')) = 2$ and thus $f_2(S \cap T, S' \cup T') = 2 = f_2(S \cup T', S' \cap T)$. Since also $T \cup T' = V$, this implies that $g_2(S \cap T, S' \cup T') = g_2(S \cup T', S' \cap T) = 2$. Since (S, S') is tight and nonempty, we have that $x_{E-F}(S, S') = 1$ and $x_F(S, S') = 0$. Since $(i(S, S'), j(S, S'))$ is a witness for both $(S \cap T, S' \cup T')$ and $(S' \cap T, S \cup T')$, we have that $x_E(S \cap T, S' \cup T') \geq 2$ and $x_E(S' \cap T, S \cup T') \geq 2$. All edges that contribute to both of these expressions are in $\delta_E(S, S')$, and thus their contribution is at most 1. We have

$$\begin{aligned}
2 &\leq x_E(S' \cap T, S \cup T') \\
&= x_E(S' \cap T, S \cap T) + x_E(S' \cap T, S' \cap T') + x_E(S' \cap T, S \cap T') \\
&\leq x_E(S, S') + x_E(S' \cap T, S' \cap T') \qquad (13) \\
&= x_{E-F}(S, S') + x_E(S' \cap T, S' \cap T') \\
&= 1 + x_E(S' \cap T, S' \cap T').
\end{aligned}$$

This implies that $x_E(S' \cap T, S' \cap T') \geq 1$. Similarly, for $x_E(\delta(S \cap T))$, we have

$$\begin{aligned}
2 &\leq x_E(S \cap T, S' \cup T') \\
&= x_E(S \cap T, S' \cap T) + x_E(S \cap T, S \cap T') + x_E(S \cap T, S' \cap T') \\
&\leq x_E(S, S') + x_E(S \cap T, S \cap T') \qquad (14) \\
&= x_{E-F}(S, S') + x_E(S \cap T, S \cap T') \\
&= 1 + x_E(S \cap T, S \cap T'),
\end{aligned}$$

which implies that $x_E(S \cap T, S \cap T') \geq 1$. Since T is tight, we have that $2 \geq x_E(T, T') \geq x_E(S' \cap T, S' \cap T') + x_E(S \cap T, S \cap T') \geq 2$, so that $g_2(T, T') = 2$ and $x_E(S' \cap T, S' \cap T') = x_E(S \cap T, S \cap T') = 1$.

This last pair of equations has several implications. By (13), this implies that all edges with $x(e) > 0$ in $\delta_{E-F}(S, S')$ enter $S' \cap T$, and in particular, $x_E(S \cap T, S' \cap T') = 0$. In addition, with (14), we see that all edges in $\delta_{E-F}(S, S')$ with $x(e) > 0$ also enter $S \cap T$, thus no edge leaving S contributes to $x_E(T, T')$. This last observation with $x_E(S' \cap T, S' \cap T') = x_E(S \cap T, S \cap T') = 1$, this implies that $(S \cap T, S' \cup T')$ and $(S \cup T', S' \cap T)$ are tight, and $2\chi_x(S, S') + \chi_x(T, T') = \chi_x(S \cap T, S' \cup T') + \chi_x(S \cup T', S' \cap T)$. Thus (iii) holds. ■

With this uncrossing lemma, we can now use the framework described in Section 4. As before, we define a relation $\leq$ on set pairs by $(S, S') \leq (T, T')$ if $S \subseteq T$ and $S' \supseteq T'$. This relation defines a partial order. We say that the pairs (S, S') and (T, T') *pair-cross* if they do not satisfy either of the following two conditions: (1) (S, S') and (T, T') are comparable in the partial order (that is, either $(S, S') \leq (T, T')$ or vice versa); (2) $S \subseteq T'$ and $T \subseteq S'$. A collection $\mathcal{L}$ of pairs (S, S') is called *pair-laminar* if no two pairs in $\mathcal{L}$ pair-cross. Using Lemma 4.3 and Lemma 5.4 in place of Lemma 4.1, we have the following corollaries with proof identical to that of Lemma 4.5 and Corollary 4.6

Corollary 5.5 *Any maximal pair-laminar collection $\mathcal{L}$ of tight set pairs satisfies* $\mathrm{Span}(\mathcal{L}) = \mathrm{Span}(\mathcal{T})$. ■

Corollary 5.6 *There exists a collection $\mathcal{B}$ of pair-laminar tight set pairs satisfying*

1. $|\mathcal{B}| = |E_x|$,
2. *the vectors $\chi_x(S, S')$ for $(S, S') \in \mathcal{B}$ are linearly independent,*
3. $(g_2 - |\delta_F|)(S, S') \geq 1$ *for all* $(S, S') \in \mathcal{B}$. ■

Similarly, using the same definitions of incidence and crossing, the analogs of Lemmas 4.8 and 4.9 also hold for this problem, and the remainder of the proof of Theorem 3.13 is identical to the proof of Theorem 3.5.

6 Implementation Issues

To solve $(\mathbf{WTS})$ for ELC in polynomial time, we need a separation oracle for the connectivity constraints: an algorithm that finds a violated constraint of the LP $(\mathbf{WTS})$ with the function $g_{\mathsf{elt}}(S, S')$ or $(\mathbf{RES})$ with the function $g_{\mathsf{elt}}(S, S') - |\delta_F(S, S')|$. To do this, we interpret x-values as capacities and transform the graph induced by the current fractional solution x and the fixed edges F into a directed graph by replacing every edge by oppositely oriented edges with the same capacity as the original undirected edge. We then perform a standard procedure of splitting nonterminal vertices to model the fact that at most one path can pass through any nonterminal. Then, in the resulting graph, the maximum flow value between i and j is vertex connectivity between i and j. If this is less than r_{ij}, the minimum cut reveals a violated inequality.

Thus we have a polynomial-time separation oracle for ($\mathbf{WTS}$) and ($\mathbf{RES}$) for element connectivity. Similarly, we can obtain a separation oracle for $\{0,1,2\}$-vertex connectivity. Using ellipsoid algorithm, we can then obtain a basic solution in polynomial time [12].

If we can be satisfied with an ϵ-approximate solution (a solution that has value at most $1+\epsilon$ times the optimal solution), then we can avoid using the ellipsoid algorithm, and instead use a fully polynomial time approximation scheme to solve the LP [7, 10]. This will lead to a slight deterioration in the quality of the final solution, i.e. to $2+2\epsilon$ instead of 2.

7 Examples and Counterexamples

There is an example in [13] that shows that the analysis of the approximation guarantee obtained by the iterative rounding algorithm is tight for the edge connectivity problem with connectivity requirements in $\{0,1\}$. Since in this case, the edge, element, and vertex connectivity problems are the same, the same example shows that the analysis is also tight for the element and vertex connectivity problems.

A natural question is: Can we extend the arguments given here to give a constant-factor approximation algorithm for vertex connectivity problems with higher connectivity requirements? We answer this question negatively for general $r \in \{1,k\}^{V\times V}$ by describing basic solutions to an infinite family of instances of ($\mathbf{RES}$) for which

1) the tight set pairs spanning the basis are highly non-laminar, and

2) the largest fraction is bounded above by $\frac{1}{\max r_{ij}}$.

Specifically, we construct a family of vertex connectivity instances where each vertex i has a demand $r_i \in \{1,k\}$ for all $i \in V$, and $r_{ij} = \min\{r_i, r_j\}$. This family has the property that after solving the initial LP and fixing all edges e with $x_e = 1$, the residual LP has a basic solution with largest x-value equal to $\frac{1}{k}$. This example does not itself rule out a proof that iterative rounding might always yield a good approximation guarantee. What it demonstrates is that if this were true, it would have to be proved by completely different methods. Since the first appearance of this example, however, there is now a proof that no constant factor approximation is obtainable via any polynomial time algorithm given certain complexity assumptions [16].

We depict the family of instances in Figure 4: For each k construct a graph on $2k$ vertices. The first k vertices $V = \{v_1, \ldots, v_k\}$ have demand k, the second k vertices $U = \{u_1, \ldots, u_k\}$ have demand 1. The edge set consists of a clique of 0-cost edges on V, and a complete bipartite graph between V and U of cost 1 edges. There is an optimal LP solution such that every edge e in the clique has $x(e) = 1$ and every edge e in the bipartite graph has $x(e) = 1/k$. After setting $F = \{e \mid x(e) = 1\}$, there is an optimal solution to ($\mathbf{RES}$) with $f = g_k$ that will still have every edge in the bipartite graph at value $1/k$. We will show below that this a basic solution. To establish that this is an optimal solution, it suffices to produce a feasible solution to the dual linear program of equal value. In the dual, there is a variable $y(S, S')$ for every possible set pair (S, S') and a variable $z(e)$ for each edge $e \in E$. The quantity $y(S, S')$ represents the marginal cost of increasing the demand from S to S'. The quantity $z(e)$ is the marginal value of edge e to meeting the demand. The dual linear program is

$$\begin{array}{lrcll} \max & \sum_{S,S'} \left[f(S,S') - |\delta_F(S,S')|\right] y(S,S') - \sum_{e \in E-F} z(e) & & & \\ \text{s.t.} & \sum_{S,S': e \in \delta(S,S')} y(S,S') - z(e) & \leq & c(e), & \forall e \in E - F \\ & y(S,S'), z(e) & \geq & 0. & \end{array}$$

In the case of this instance, if $f(S, S') - |\delta_F(S, S')| > 0$ then it equals 1; and $c(e) = 1$ for all $e \in E - F$. Thus, the solution $\mathbf{z} = \mathbf{0}$, $y(S, S') = 1$ for $S = \{u_i\}$, $S' = V - S$, $1 \leq i \leq k$, and otherwise, $y(S, S') = 0$, is feasible, and has value k, equal to the value of the feasible primal solution. Hence both are optimal.

We now establish that this is a vertex of the polytope described by ($\mathbf{RES}$) with all cost 0 edges included in $F = \{e |\ c(e) = 0\}$. We do this by describing a set of k^2 tight inequalities (note that k^2 is the number of fractional edges and hence variables in the remaining problem), constructing a matrix of the support of these inequalities, constructing a second matrix and arguing that the two matrices are inverses of each other, hence each are linearly independent. Since the solution is then the intersection of k^2 linearly independent halfspaces in $\mathbb{R}^{k^2}$, it is a vertex of the linear programming polytope.

The set of k^2 tight inequalities is divided into k blocks of k inequalities. Block 0 includes the k inequalities with $S = \{u_i\}$, $1 \leq i \leq k$, and $S' = V - S$. Aside from these inequalities, the point is highly degenerate: there are more tight inequalities than are necessary to define the point. For ease of presentation, we have chosen to particular symmetric set for the the remaining $k - 1$ blocks of inequalities. For inequality $2 \leq q \leq k$, in block $1 \leq i \leq k - 1$, define $S_{i,q} = \{v_{i+1}, u_1, u_2, \ldots, u_{k-q+1}\}$, and $S'_{i,q} = \{v_i, u_{k-q+2}, \ldots u_k\}$. For $q = 1$, let $S_{i,1} = \{v_{i+1}, u_1, u_2, \ldots, u_k\}$ and $S'_{i,1} = \{v_i\}$. There is exactly one edge in F that has one endpoint in each of S and S' (edge (v_i, v_{i+1})); the $(g_2 - \delta_F)$-value of the inequality is 1; and $x(S_{i,q}, S'_{i,q})$ is determined by the $q - 1$ edges from U to v_{i+1}

and the $k-q+1$ edges from U to v_i, for a total value of $(q-1+k-q+1)\frac{1}{k}=1$. Thus, these cuts are tight. See Figure 4.

Let matrix C be the support matrix of edges in each cutset above, with the rows of C corresponding to cutsets and the columns corresponding to edges. Thus, a "1" in the $(r,(j,l))$ place means that the edge from u_j to v_l crosses the set pair (S_r, S'_r). Here r corresponds to pairs (i,q). The rows of C are ordered first according to block, and then within each block, according to q. The first row block in C corresponds to the inequalities with $S=\{u_i\}$, i.e. it is the block 0 of the tight inequalities. The columns of C are ordered according to incidence to U, and then to V. See Figure 5. Figure 6 displays C and the inverse matrix B for $k=3$.

Let matrix B be a $k^2 \times k^2$ matrix with its columns and rows ordered into blocks of k. The first block of k columns (called column block 0) has a pattern that is slightly different from the rest. See Figure 7. The first column is 0 everywhere except in the last entry in the first row block, which is 1. The 2^{nd} through k^{th} columns have the following pattern: the first row block consists of $k-1$ entries of value $\frac{-1}{k}$ followed by a single entry of $\frac{k-1}{k}$. Then the q^{th} column has the q^{th} row block filled with $\frac{1}{k}$. All other entries are 0.

For the pattern of the i^{th} block of k columns, $i=1,\ldots,k-1$, see Figure 8. The first column of the first row block has $i-1$ entries of $\frac{-k+i}{k}$ followed by $k-i$ entries of $\frac{i}{k}$ followed by a single entry of $\frac{-k+i}{k}$. This column vector is denoted X_i. The first column of the last row block is the vector $-Y_i$ containing i entries of $\frac{k-i}{k}$ followed by $k-i$ entries of $\frac{-i}{k}$. The q^{th} column of the $k-q+1^{st}$ row block is $-Y_i$ for $1 \le q \le k$; and for $1 \le q \le k-1$, the q^{th} column of the $k-q+2^{nd}$ row block is the vector Y_i with i entries of $\frac{-k+i}{k}$ followed by $k-i$ entries of $\frac{i}{k}$. Note that $X_i = Y_i + \mathbf{e}_i^T - \mathbf{e}_k^T$, where $\mathbf{e}_i$ is the $1 \times k$ i^{th} elementary vector: $\mathbf{e}_i(i)=1,\ \mathbf{e}_i(j)=0$ for $j \neq i$.

The following lemma follows by inspection of B and C. We provide a proof for completeness.

Lemma 7.1 *For any k, matrices B and C are inverses.*

Proof: We show that $CB=I$ by examining each entry of CB, where we have $C=\begin{pmatrix} C_0 \\ C' \end{pmatrix}$ expressed as the row block 0, C_0, and the remaining rows C', which are partitioned into $k-1$ blocks of k rows each $C_1,\ldots C_{k-1}$; and $B=(B_0\ B_1\ \ldots\ B_{k-1})$ expressed as k blocks of k columns.

Block C_0 times B_0: The first row of C_0 will have product 1 with the first column of B_0 and product $(k-1)\frac{-1}{k}+\frac{k-1}{k}=0$ with the remaining $k-1$ columns B_0. The i^{th} row of C, for $2 \le i \le k$, will have product $k \cdot \frac{1}{k}=1$ with the i^{th} column of B and product 0 elsewhere.

Block C_i, $i \geq 1$, times B_0: Each row of C_i will have product $-1/k + 1/k = 0$ with each column of B_0.

C_0 times B_i, $i \geq 1$: Note that $\mathbf{1} \cdot Y_i = i(\frac{-k+i}{k}) + (k-i)\frac{i}{k} = 0$. Thus, $\mathbf{1} \cdot X_i = \mathbf{1} \cdot (Y_i + \mathbf{e}_i^T - \mathbf{e}_k^T) = 0$, and $\mathbf{1} \cdot -Y_i = 0$. So each row of C_0 times any column of B_i for $i \geq 1$ has product 0.

C_i times B_j, $i \geq 1$, $j \geq 1$: Note that the r^{th} entry of X_j is the same as the $(r+1)^{st}$ entry of Y_j for all $1 \leq r \leq k-1$. Thus, rows 2 through k of C_i have product 0 with the first column of B_j. Also note that the r^{th} entry of X_j differs from the r^{th} entry of Y_j precisely when $r = j$ or $r = k$. Thus, the only time a first row of C_i has nonzero product with a first row of B_j is when $i = j$; and in this case, the product is $\frac{i}{k} + \frac{k-i}{k} = 1$. Each remaining row $2 \leq l \leq k$ of C_i has a difference only between consecutive column blocks $k - l + 1$ and $k - l + 2$, where the first contains vector e_i and the second contains e_{i+1}. Thus, the product with column r of B_j will only be nonzero when $j = i$ and $r = l$ (for $2 \leq r \leq k$). ■

Acknowledgements

We thank the anonymous referees for helpful comments that contributed to an improved presentation of this paper. The first author acknowledges the partial support provided by NSF through grants EIA-0049084 and CCR-0049071, and IBM's T. J. Watson Research Center.

References

[1] *8th International Integer Programming and Combinatorial Optimization Conference*, 2001.

[2] J. Cheriyan and S. Vempala. Edge covers of setpairs and the iterative rounding method. In *8th International Integer Programming and Combinatorial Optimization Conference* [1], pages 30–44.

[3] J. Cheriyan, S. Vempala, and A. Vetta. Approximation algorithms for minimum-cost k-vertex connected subgraphs. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 306–312, 2002.

[4] V. Chvatal. *Linear programming*. W. H. Freeman, New York, 1983.

[5] R. Diestel. *Graph Theory*. Number 173 in Graduate Texts in Mathematics. Springer-Verlag, New York, 2nd edition, 2000.

[6] L. Fleischer. A 2-approximation for minimum cost $\{0, 1, 2\}$ vertex connectivity. In *8th International Integer Programming and Combinatorial Optimization Conference* [1], pages 115–129.

[7] L. Fleischer. A fast approximation scheme for fractional covering problems with variable upper bounds. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2004. To appear.

[8] L. Fleischer, K. Jain, and D. P. Williamson. An iterative rounding 2-approximation algorithm for the element connectivity problem. In *42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 339–347, 2001.

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

[10] N. Garg and R. Khandekar. Fast approximation algorithms for fractional steiner forest and related problems. In *43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002. To appear.

[11] M. X. Goemans, A. V. Goldberg, S. Plotkin, D. Shmoys, É. Tardos, and D. P. Williamson. Improved approximation algorithms for network design problems. In *Proc. 5th Annual ACM-SIAM Symp. on Discrete Algorithms*, pages 223–232, 1994.

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

[13] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[14] K. Jain, I. Mandoiu, V. V. Vazirani, and D. P. Williamson. A primal-dual schema based approximation algorithm for the element connectivity problem. In *Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 484–489, 1999.

[15] S. Khuller and B. Raghavachari. Improved approximation algorithms for uniform connectivity problems. *J. Algorithms*, 1996.

[16] G. Kortsarz, R. Krauthgamer, and J. R. Lee. Hardness of approximation for vertex-connectivity network-design problems. In *5th Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX)*, pages 185–199, 2002.

[17] G. Kortsarz and Z. Nutov. Approximating node connectivity problems via set covers. In *Approximation Algorithms for Combinatorial Optimization (Proc. of APPROX 2000)*, number 1913 in Lecture Notes in Comp. Sci., pages 194–205. Springer-Verlag, 2000.

[18] R. Ravi and D. P. Williamson. An approximation algorithm for minimum-cost vertex-connectivity problems. *Algorithmica*, 18(1):21–43, 1997.

[19] R. Ravi and D. P. Williamson. Erratum: an approximation algorithm for minimum-cost vertex-connectivity problems. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1000–1, 2002.

[20] A. Schrijver. Matroids and linking systems. *Journal of Combinatorial Theory B*, 26:349–369, 1979.

[21] M. Stoer. *Design of survivable networks*. Number 1531 in Lecture Notes in Mathematics. Springer-Verlag, Berlin, 1992.

[22] L. Zhao, H. Nagamochi, and T. Ibaraki. A primal-dual approximation algorithm for the survivable network design problem in hypergraphs. *Discrete Applied Mathematics*, 126(2-3):275–289, 2003.

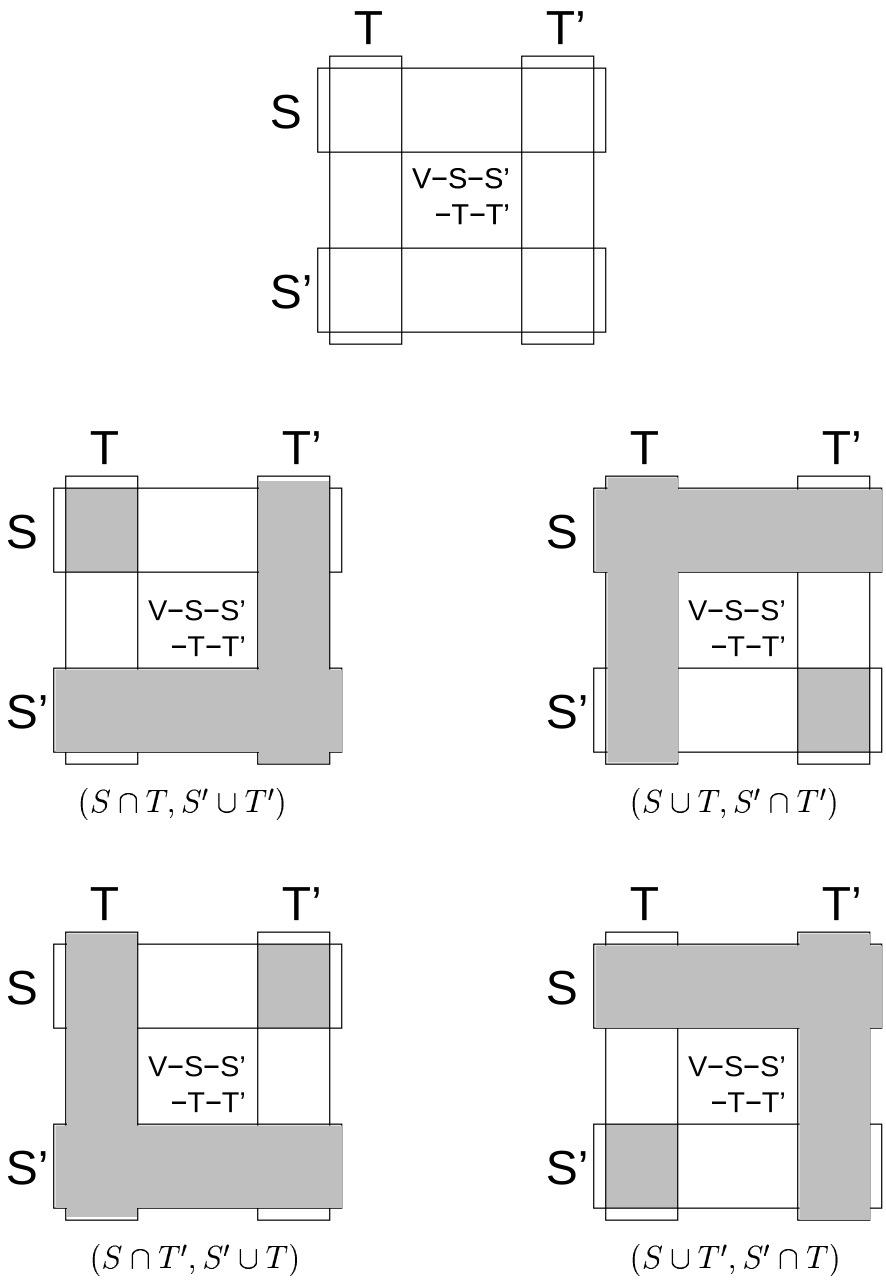

Figure 1: A pictoral representation of the set pairs involved in the definition of two-submodular and two-supermodular.

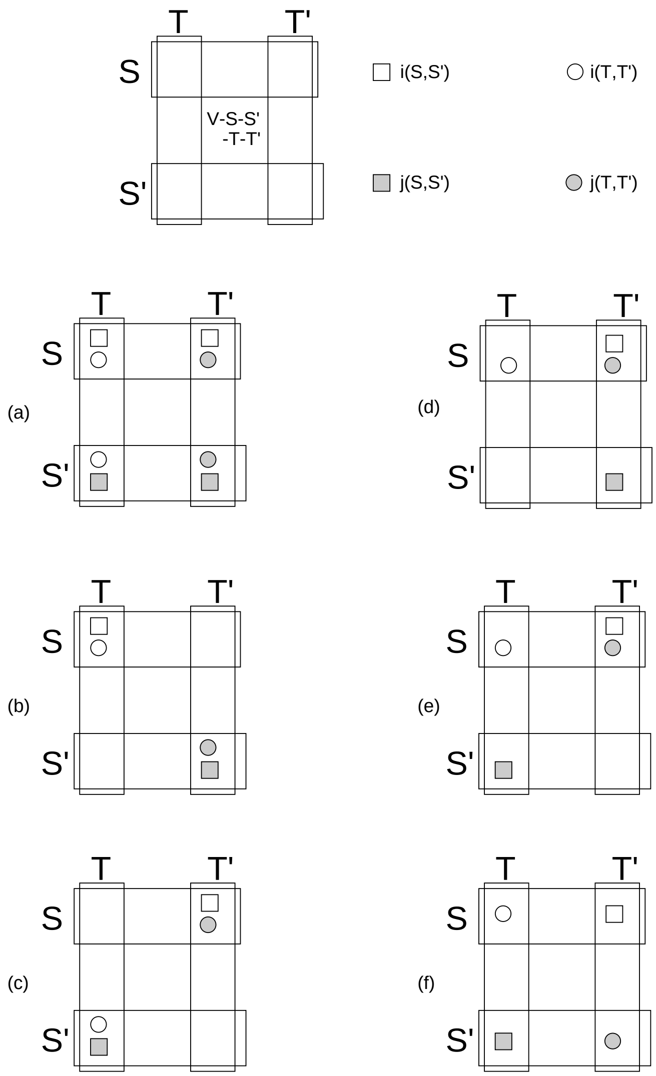

Figure 2: Cases in the proof of Lemma 3.11. Here $I = \{i(S,S'), i(T,T'), j(S,S'), j(T,T')\}$. (a) The possible locations of $i(S,S')$, $j(S,S')$, $i(T,T')$, and $j(T,T')$. (b) I intersects two sets and (3) holds. (c) I intersects two sets and (4) holds. (d) I intersects three sets; $j(S,S')$ and $j(T,T')$ are contained in complementary sets; and (4) holds. (e) I intersects three sets; $i(T,T')$ and $j(S,S')$ are contained in complementary sets; and (3) holds. (f) I intersects four sets. If $f_{\mathsf{elt}}(T,T') \geq f_{\mathsf{elt}}(S,S')$, then (3) holds; else (4) holds.

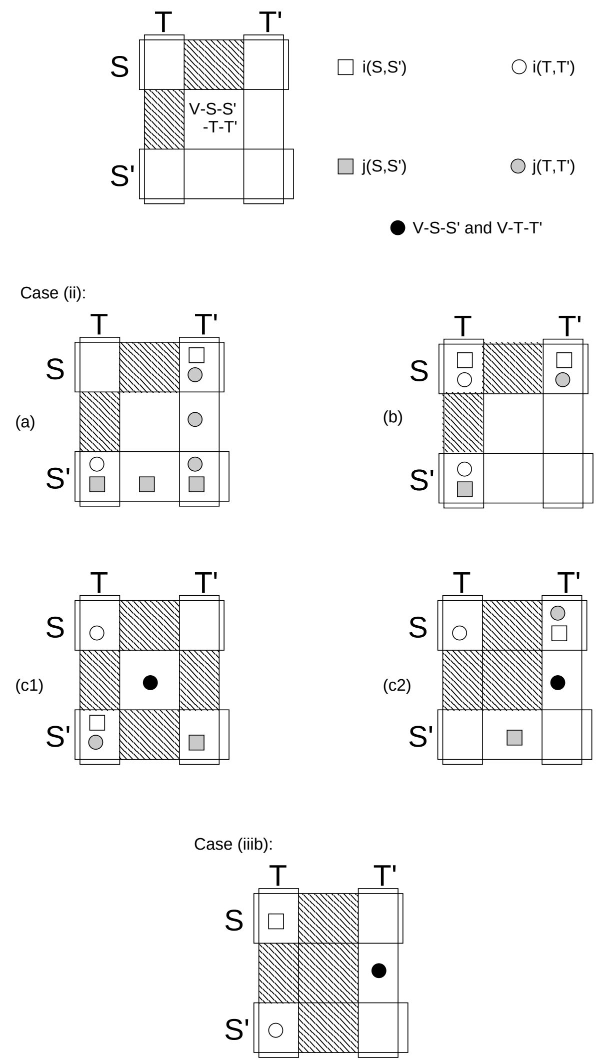

Figure 3: Representative cases in the proof of Lemma 5.1. The striped areas indicate empty regions. Case (ii): (a)-(b) Inequality (9) holds. (c1): $(i(S,S'),j(S,S'))$ is a witness for $f_2(S \cap T,\ S' \cup T')$; $(i(T,T'),j(T,T'))$ is a witness for $f_2(S \cup T,\ S' \cap T')$; and inequality (8) holds. (c2): Inequality (10) holds. Case (iiib): $(i(S,S'),j(S,S'))$ is a witness for $f_2(S \cap T,\ S' \cup T')$; $(j(S,S'),i(S,S'))$ is a witness for $f_2(S' \cap T,\ S \cup T')$; and inequality (11) holds.

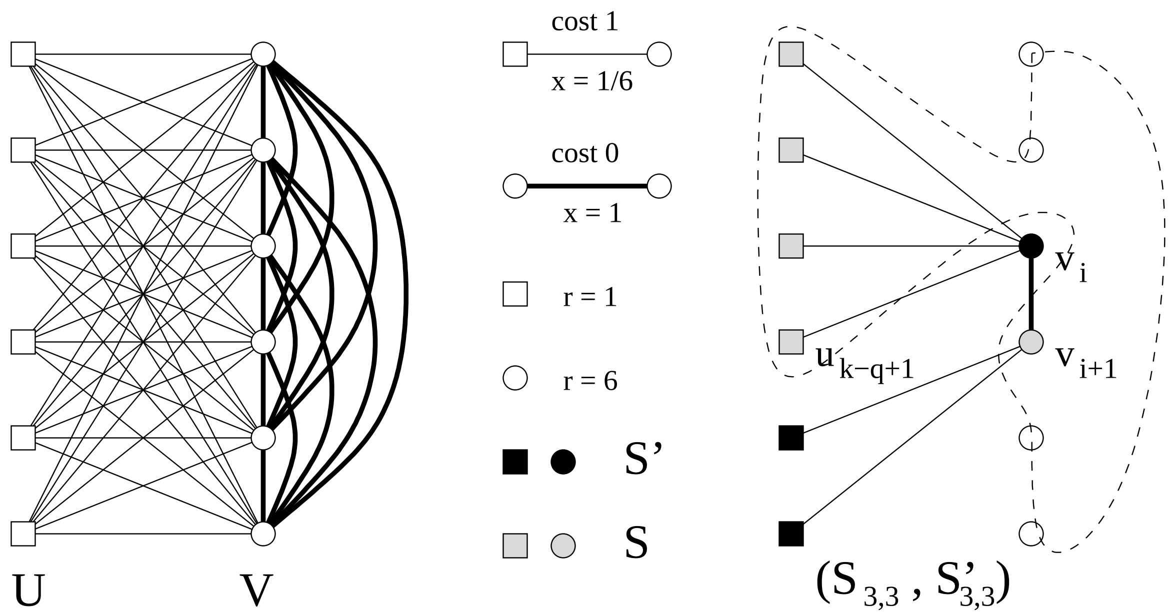

Figure 4: On the left, a basic solution to (**RES**) after fixing the 0-cost edges to 1. The largest fraction in the solution is $\frac{1}{k}$, here $k = 6$. On the right, an example of a set $(S_{i,q}, S'_{i,q})$ for $i = 3$, $q = 3$ with $f_2(S_{i,q}, S'_{i,q}) = k$. All the edges crossing the cut are included in the figure. They have total value 2. Together with the 4 vertices in $V - S - S'$, this satisfies (12) at equality. The collection of cuts $\{(S_{i,q}, S'_{i,q})\}_{1 \le i \le k-1,\ 1 \le q \le k}$ are highly crossing.

	u_1	u_2	$\cdots$	u_{k-1}	u_k
block 0	$\mathbf{1}$	$\mathbf{0}$		$\mathbf{0}$	$\mathbf{0}$
	$\mathbf{0}$	$\mathbf{1}$		$\mathbf{0}$	$\mathbf{0}$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	$\mathbf{0}$	$\mathbf{0}$		$\mathbf{1}$	$\mathbf{0}$
	$\mathbf{0}$	$\mathbf{0}$		$\mathbf{0}$	$\mathbf{1}$
block 1	$\mathbf{e}_1$	$\mathbf{e}_1$	$\cdots$	$\mathbf{e}_1$	$\mathbf{e}_1$
	$\mathbf{e}_1$	$\mathbf{e}_1$	$\cdots$	$\mathbf{e}_1$	$\mathbf{e}_2$
	$\mathbf{e}_1$	$\mathbf{e}_1$	$\cdots$	$\mathbf{e}_2$	$\mathbf{e}_2$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\mathbf{e}_1$	$\mathbf{e}_2$	$\cdots$	$\mathbf{e}_2$	$\mathbf{e}_2$
block 2	$\mathbf{e}_2$	$\mathbf{e}_2$	$\cdots$	$\mathbf{e}_2$	$\mathbf{e}_2$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\mathbf{e}_2$	$\mathbf{e}_3$	$\cdots$	$\mathbf{e}_3$	$\mathbf{e}_3$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
block k-1	$\mathbf{e}_{k-1}$	$\mathbf{e}_{k-1}$	$\cdots$	$\mathbf{e}_{k-1}$	$\mathbf{e}_{k-1}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$\mathbf{e}_{k-1}$	$\mathbf{e}_k$	$\cdots$	$\mathbf{e}_k$	$\mathbf{e}_k$

Figure 5: **Incidence Matrix C of k^2 Tight Set Pairs.** Here **1** denotes the $1 \times k$ row vector of all 1's, **0** denotes the $1 \times k$ row vector of all 0's, and $\mathbf{e}_i$ denotes the $1 \times k$ row vector that has an 1 in the i^{th} position, and 0's elsewhere.

$$C = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & -.333 & -.333 & .333 & 0 & .667 & -.333 & 0 & .333 \\ 0 & -.333 & -.333 & .333 & 0 & -.333 & .667 & 0 & .333 \\ 1 & .667 & .667 & -.667 & 0 & -.333 & -.333 & 0 & -.667 \\ 0 & .333 & 0 & 0 & .667 & -.667 & 0 & .333 & -.333 \\ 0 & .333 & 0 & 0 & -.333 & .333 & 0 & .333 & -.333 \\ 0 & .333 & 0 & 0 & -.333 & .333 & 0 & -.667 & .667 \\ 0 & 0 & .333 & .667 & -.667 & 0 & .333 & -.333 & 0 \\ 0 & 0 & .333 & -.333 & .333 & 0 & .333 & -.333 & 0 \\ 0 & 0 & .333 & -.333 & .333 & 0 & -.667 & .667 & 0 \end{pmatrix}$$

Figure 6: Matrices B and C for $k = 3$.

row	
block 1	$\frac{-1}{k}\mathbf{1} + \frac{1}{k}\mathbf{e}_1$
	$\vdots$
	$\frac{-1}{k}\mathbf{1} + \frac{1}{k}\mathbf{e}_1$
	$\frac{k-1}{k}\mathbf{1} + \frac{1}{k}\mathbf{e}_1$
block 2	$\frac{1}{k}\mathbf{e}_2$
	$\vdots$
	$\frac{1}{k}\mathbf{e}_2$
block 3	$\frac{1}{k}\mathbf{e}_3$
	$\vdots$
	$\frac{1}{k}\mathbf{e}_3$
	$\vdots$
block k	$\frac{1}{k}\mathbf{e}_k$
	$\vdots$
	$\frac{1}{k}\mathbf{e}_k$

Figure 7: **Column Block 0 of Matrix B:** The first k columns of matrix B. Here $\mathbf{e}_i$ is the $1 \times k$ i^{th} elementary row vector; and **1** is the $1 \times k$ vector of all 1's.

Col. Block i of B

$$\begin{matrix} X_i & \mathbf{0} & \cdots & \mathbf{0} & -Y_i \\ \mathbf{0} & \mathbf{0} & \cdots & -Y_i & Y_i \\ \mathbf{0} & \mathbf{0} & \cdots & Y_i & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & -Y_i & \cdots & \mathbf{0} & \mathbf{0} \\ -Y_i & Y_i & \cdots & \mathbf{0} & \mathbf{0} \end{matrix}$$

	X_i	Y_i
row 1	$\frac{-k+i}{k}$	$\frac{-k+i}{k}$
$\vdots$	$\vdots$	$\vdots$
row i-1	$\frac{-k+i}{k}$	$\frac{-k+i}{k}$
row i	$\frac{i}{k}$	$\frac{-k+i}{k}$
row i+1	$\frac{i}{k}$	$\frac{i}{k}$
$\vdots$	$\vdots$	$\vdots$
row k-1	$\frac{i}{k}$	$\frac{i}{k}$
row k	$\frac{-k+i}{k}$	$\frac{i}{k}$

Figure 8: On the left, the pattern of column block i of matrix B. On the right, the composition of the vectors X_i and Y_i that describe block i. Here **0** is the $k \times 1$ column vector of all 0's; and $X_i = Y_i + \mathbf{e}_i^T - \mathbf{e}_k^T$.