A 2-approximation for Minimum Cost $\{0, 1, 2\}$ Vertex Connectivity

Lisa Fleischer

Graduate School of Industrial Administration
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15217, USA
lkf@andrew.cmu.edu

Abstract. In survivable network design, each pair (i, j) of vertices is assigned a level of importance r_{ij}. The vertex connectivity problem is to design a minimum cost network such that between each pair of vertices with importance level r, there are r *vertex* disjoint paths. There is no approximation algorithm known for this general problem. In this paper, we give a 2-approximation for the problem when $r \in \{0, 1, 2\}^{V \times V}$, improving on a previous known 3-approximation. This matches the best known approximation for the easier problem that requires that the paths be only edge-disjoint.

Our algorithm extends an iterative rounding algorithm that gives a 2-approximation for the edge-connectivity problem, for arbitrary connectivity requirements r. (K. Jain, A factor 2 approximation for the generalized Steiner network problem.) This algorithm relies on well-known uncrossing lemma for tight edge cutsets. Our extension uses a new type of uncrossing lemma for tight cutsets that may include vertices as well as edges.

For $r \in \{1, k\}^{V \times V}$, $k \geq 3$, we show that a) uncrossing tight cutsets is not possible, and b) any analysis for iterative rounding that depends directly on the largest fractional value in the linear programming solution cannot provide approximation guarantees better than the maximum connectivity requirement.

1 Introduction

Let $G = (V, E)$ be an undirected graph on vertex set V and edge set E. Given $X \subset V$, define $\delta(X)$ as the set of edges with exactly one endpoint in X and $E(X)$ as the set of edges with both endpoints in X. Define $G - X$ as that graph obtained from G by removing all vertices in X and all edges in $\delta(X) \cup E(X)$. Given $F \subset E$, define $G - F$ to be that graph obtained from G by removing all edges in F. G is called *k-vertex connected* if $|V| > k$ and for every $X \subset V$ with $|X| < k$, $G - X$ is connected. G is called *k-edge connected* if $|V| > k$ and for every $F \subset E$ with $|F| < k$, $G - F$ is connected. A k-vertex connected graph is k-edge connected, but the converse does not typically hold. Given vertex connectivity requirements r_{ij} between any pair of vertices (i, j), G satisfies the connectivity requirements if for every subset $X \subseteq V - \{i, j\}$ with $|X| < r_{ij}$, i and j are in the same connected component of $G - X$. If the requirements are for edge connectivity instead, then G satisfies the connectivity requirements if for every subset $F \subseteq E$ with $|F| < r_{ij}$, i and j are in the same connected component of $G - F$.

Let c be a cost vector on the edges of G. The problem of finding the minimum cost subgraph of G so that G satisfies edge connectivity requirements $r \in \mathbf{Z}^{V \times V}$ is called the *minimum cost edge connectivity problem (MCEC)*. The equivalent problem for vertex connectivity is the *minimum cost vertex connectivity problem (MCVC)*. Both of these problems are Max-SNP hard since the Steiner tree problem is a special case of each.

In [10], Jain describes the first constant factor approximation for MCEC. He obtains this approximation by iteratively solving linear programs with the property that at least one variable in each program has solution value of at least 1/2. His main contribution is to prove this property holds for the iterative problems generated by his algorithm. The previous best approximation algorithm, by Goemans et al., gives a $O(\log k)$ approximation [5]. This is a primal-dual based approximation algorithm that does not rely on solving linear programs.

No nontrivial approximation algorithm is known for MCVC. For the problem where r is restricted to $\{0, 1, 2\}^{V \times V}$, Ravi and Williamson [16] describe a primal-dual 3-approximation algorithm. We call this problem $\{0, 1, 2\}$-MCVC. This problem arises in the design of survivable communications networks [8, 15].

In this paper we describe a 2-approximation algorithm for $\{0, 1, 2\}$-MCVC that iteratively rounds appropriately defined linear programs. This approximation guarantee now matches the best approximation guarantee for the corresponding edge-connectivity problem [10].

Our approximation algorithm extends the algorithm of Jain [10]. Jain considers basic solutions to a linear programming relaxation of an integer programming formulation of the problem. A *basic* solution is any solution corresponding to a vertex of the polytope defined by the inequalities describing the linear program. Any basic solution is uniquely defined by a set of $|E|$ inequalities that are satisfied at equality. Any inequality that is satisfied at equality is called *tight.* Jain shows that for any basic feasible solution to the LP, there exists a set of inequalities that define the solution that correspond to laminar subsets of vertices. Two sets S and T are called *laminar* if at least one of $S \cap T$, $S \backslash T$, $T \backslash S$, and $V \backslash (S \cup T)$ is empty. Otherwise S and T cross. A set of sets is laminar if every pair in the set is laminar. Using laminarity, Jain develops a charging scheme to bound from below the value of the highest solution coordinate.

For $\{0, 1, 2\}$ MCVC, we define an appropriate linear program whose set of integer solutions correspond to solutions to $\{0, 1, 2\}$ MCVC. We prove the existence of a laminar set of inequalities defining any basic feasible solution to this LP. The proof relies on a new uncrossing lemma. We can then use a very similar charging scheme to obtain the same lower bound on the value of the highest coordinate in the solution vector.

A natural question is if this can be extended to yield a constant factor approximation for $\{0, 1, \ldots, k\}$-vertex connectivity. We show the same proof technique will not work by exhibiting an infinite family of examples where the only sets of inequalities defining a basic solution to the appropriate linear program are highly non-laminar. In this example, the largest fraction after one rounding is $\frac{1}{k}$. This indicates that an approximation argument based simply on the size largest fraction in an iterative rounding scheme will not yield better than a k-approximation.

For the more specialized connectivity problems of constructing a minimum cost uniformly k-connected graph, the best known approximation guarantee is roughly factor k [3, 13].[1] If c is a metric, then there are constant factor approximations for uniform k-connectivity [12, 13]. Last IPCO, Melkonian and Tardos [14] extend Jain's iterative rounding analysis to obtain a 4-approximation for uniform k-edge connectivity on directed graphs. They also describe a different approach that yields a 2-approximation. There are numerous approximation results for other special cases of vertex and edge connectivity. For surveys, see [6, 11].

We discuss our linear program formulation and the recursive algorithm in Section 2. We prove the existence of a variable with value at least $1/2$ in Section 3. In Section 4, we describe an infinite family of examples that show that these proof techniques do not extend when $r \in \{1, k\}^{V \times V}$.

2 A 2-approximation

Our 2-approximation relies on formulating the problem as an integer program, solving the LP relaxation of the integer program, and showing that there is at least one edge with fractional value greater than or equal to $1/2$. If we include this edge in our final solution, its contribution to the cost of our solution is no more than twice its contribution to the linear program solution, the latter being a lower bound on the cost of an optimal integer solution. We then show that the remaining problem is of the same general form as our original problem, and that we can use recursion to obtain a complete integer solution that has cost at most twice the optimal solution to the original linear program. Thus it is a 2-approximation to our problem.

This general outline was suggested by Jain in [10]. He uses the following linear programming relaxation of MCEC. There is a variable $x(e)$ for every edge $e \in E$, and an inequality for every subset of vertices that requires that the number of edges leaving the set be at least the maximum connectivity requirement over all pairs of vertices that have exactly one member of the pair in the set. Let $f(S)$ be defined to take this value for subset S. Let $\delta(S)$ denote the set of edges with

[1] In [16], a primal-dual algorithm is proposed, but the analysis has recently discovered to be flawed, and the algorithm does not provide the claimed $O(\log k)$-factor guarantee [17].

exactly one endpoint in S. The linear program is:

$$\begin{aligned} \min \quad & cx \\ \text{s.t. } \textstyle\sum_{e\in\delta(S)} x(e) \geq f(S), & \quad \forall S \subset V \\ 0 \leq x(e) \leq 1, & \quad \forall e \in E \end{aligned} \qquad \textbf{(MCEC)}$$

The key to the argument in [10] is establishing that any basic solution to (**MCEC**) contains at least one edge with fractional value at least one half. This is done by assuming that the ground set corresponds to the support of x. Then, if $x < \mathbf{1}$, he shows that any basic solution is defined by a set of tight inequalities that correspond to a set of laminar subsets of V. This is proven by demonstrating that f is weakly supermodular and using this to employ a well-known uncrossing lemma for tight edge cutsets. This is used in an innovative charging scheme that shows that laminarity implies the existence of an edge with sufficiently high value in the solution to (**MCEC**). The other part of the argument involves establishing that this technique may be invoked recursively. This is done using a general description of f as being *weakly supermodular.* A set function f is *weakly supermodular* if $f(S) + f(T) \leq \max\{f(S \cup T) + f(S \cap T), f(S\backslash T) + f(T\backslash S)\}$.

To extend these arguments to the vertex connectivity problem we 1) introduce an appropriate linear program relaxation of the MCVC 2) prove a new uncrossing lemma for $\{0, 1, 2\}$ MCVC 3) establish that this lemma may be invoked recursively, by extending the notion of weak supermodularity. The examples in Section 4 indicate that step 2) and step 3) do not hold for more general connectivity requirements. One problem is that the cutsets corresponding to inequalities of the linear program for MCVC consist of edges *and vertices.* The inclusion of vertices in cutsets makes uncrossing nontrivial, and when connectivity requirements are higher than 2, it is no longer possible.

We use a linear program description of the problem that contains a variable $x(e)$ for each edge that indicates whether or not the edge is selected in the final network. The formulation contains an exponential number of constraints. However, as long as we can find a constraint of the LP that is violated by a given vector $x \in \{0, 1\}^E$ in polynomial time, we can find an optimal, basic solution to the linear program in polynomial time [7]. We describe such a subroutine in Section 5.

Our results extend to the case where we are allowed to select an edge multiple times. In our case, this would be at most twice. In fact, the problem appears to be only harder when we are restricted to selecting an edge at most once.

We give the linear programming formulation below. The constraints are based on a theorem of Menger. (For multiple proofs and references, see [4]):

Theorem 1 (Menger). *Let $G = (V, E)$ be a graph, and $s, t \in V$ such that $(s, t) \notin E$. 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.*

For any subset $S \subset V$, and disjoint set of vertices $A \subset V\backslash S$, we express the connectivity required between S and $V\backslash(A \cup S)$ as $f(S, A)$. Similarly, we represent the x-value of edges with one endpoint in S and the other in $V\backslash(S \cup A)$

as $x(S,A)$. That is, $x(S,A) := \sum_{i\in S, j\in V\backslash(S\cup A)} x((i,j))$. Here, A is the subset of vertices in the cutset separating S from the rest of the graph. Since there may be at most one path from S to $V\backslash(S\cup A)$ through each vertex of A, the number of edges from S to $V\backslash(A\cup S)$ must therefore be at least $f(S,A)-|A|$. This yields the following formulation.

$$\begin{array}{lll} \min & cx & \\ \text{s.t.} & x(S,A) \geq f(S,A) - |A|, & \forall\ S, A \subset V, S\cap A = \emptyset \\ & 0 \leq x(e) \leq 1, & \forall\ e \in E \end{array} \qquad \textbf{(MCVC)}$$

The following lemma is a simple consequence of Menger's Theorem.

Lemma 1. *The set of integral solutions to the above LP equals the set of solutions to the corresponding vertex connectivity problem.* □

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

$$\begin{aligned} & f(S,A) + f(T,B) \geq \\ & \max\{\ f(S\cup T, (A\backslash T)\cup(B\backslash S)) + f(S\cap T, (A\cap T)\cup(B\cap S)\cup(A\cap B)), \\ & \quad f(S\backslash(T\cup B), (A\backslash T)\cup(B\cap S)) + f(T\backslash(S\cup A), (B\backslash S)\cup(A\cap T))\} \end{aligned}$$

is called *two-submodular*. For the case when $A = B = \emptyset$, this 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 *very weakly two-supermodular* if whenever $f(S,A) > 0$ and $f(T,B) > 0$ then

$$\begin{aligned} & f(S,A) + f(T,B) \leq \\ & \max\{\ f(S\cup T, (A\backslash T)\cup(B\backslash S)) + f(S\cap T, (A\cap T)\cup(B\cap S)\cup(A\cap B)), & (1) \\ & \quad f(S\backslash(T\cup B), (A\backslash T)\cup(B\cap S)) + f(T\backslash(S\cup A), (B\backslash S)\cup(A\cap T)), & (2) \\ & \quad \max\{\ f(S\cup T, (A\backslash T)\cup(B\backslash S)), f(S\cap T, (A\cap T)\cup(B\cap S)\cup(A\cap B)), & \\ & \quad f(S\backslash(T\cup B), (A\backslash T)\cup(B\cap S)), f(T\backslash(S\cup A), (B\backslash S)\cup(A\cap T))\ \}, & (3) \\ & \ \textbf{OR}\ \text{for some permutation of } S \text{ and } V\backslash(S\cup A),\ T \text{ and } V\backslash(T\cup B), & \\ & \quad f(S\cap T, (A\cap T)\cup(B\cap S)\cup(A\cap B)) + f(T\backslash(S\cup A), (B\backslash S)\cup(A\cap T))\ \} & \end{aligned}$$

While it may seem that (3) is included in (1) and (2), if f is allowed to take on negative values, this may not be the case.

Let $\delta_F(S,A)$ denote the set of edges in F that have exactly one endpoint in each of S and $V\backslash(S\cup A)$.

Lemma 2. *Both $x(S,A)$ and $|\delta_F(S,A)|$ are two-submodular.* □

Define f_k by $f_k(S,A) := \max\{r_{ij} | i \in S, j \in V \backslash (S \cup A)\}$, where $r_{ij} \in \{0,1,\ldots,k\}$ for all $i,j \in V$. Define g_k by $g_k(S,A) = f_k(S,A) - |A|$. In Section 3, we prove Lemmas 3 and 4 and Theorem 2.

Lemma 3. *The two-set function g_2 is very weakly two-supermodular.*

Lemma 4. *For any edge set F on V, $g_2(S,A) - |\delta_F(S,A)|$ is very weakly two-supermodular.*

Theorem 2. *For the function $f(S,A) := f_2(S,A) - |\delta_F(S,A)|$, any basic solution to (**MCVC**) has at least one component with value at least $\frac{1}{2}$.*

We now describe an algorithm that yields a two approximation to (**MCVC**). This algorithm mirrors the algorithm in [10] for MCEC.

Let x^* be an optimal basic solution to (**MCVC**). Let $E_{\frac{1}{2}+}$ be the set of edges which have x^*-value $\geq \frac{1}{2}$. Fix all values of edges in $E_{\frac{1}{2}+}$ to 1. Let $E_{res} = E - E_{\frac{1}{2}+}$, and consider the resulting *residual LP*:

$$\begin{array}{lll} \min & cx & \\ \text{s.t.} & x(S,A) \geq g_2(S,A) - |\delta_E(S,A) \cap E_{\frac{1}{2}+}|, & \forall\, S,A \subset V, S \cap A = \emptyset \\ & 0 \leq x(e) \leq 1, & \forall e \in E_{res} \end{array} \qquad (\mathbf{MCVC_2})$$

Let z^*_{res} be the optimal value of this LP and z^* be the optimal value of (**MCVC**). The following theorem follows from similar arguments presented by Jain for edge-connectivity [10].

Theorem 3. *If E_{res} is an integral solution to ($\mathbf{MCVC_2}$) with value at most $2z^*_{res}$, then $E_{res} \cup E_{\frac{1}{2}+}$ is an integral solution to (**MCVC**) with value at most $2z^*$.*

The 2-approximation algorithm: 1) Find an optimal basic solution x^* to (**MCVC**), 2) Include all edges e with $x^*(e) \geq 1/2$, in the final solution. 3) Delete all edges that were included in 2), and solve the residual problem on E_{res}.

3 Uncrossing Lemmas and Proof of Theorem

In this section, we prove an uncrossing lemma (Lemma 5) that we then use to establish laminarity of a set of spanning tight subsets (Corollary 2). The proof of this lemma relies on Lemmas 3 and 4. Laminarity of the tight subsets determining a basic solution in turn implies the main theorem.

Given (S,A), there is a pair $i \in S$, $j \in V \backslash (S \cup A)$ that determines $f_2(S,A)$. Let $i(S,A)$ denote one such i and $j(S,A)$ denote the corresponding j.

Proof of Lemma 3. For $r \in \{0,1,2\}^{V \times V}$, the only values of $|A|$ that yield nontrivial inequalities for (**MCVC**) are $|A| = 0$ or 1. Since $f_2(S,A) = f_2(V \backslash (S \cup A), A)$, it suffices to show that weak two-supermodularity holds for S and T satisfying $S \cap B = T \cap A = \emptyset$. Hence $A \backslash T = A$ and $B \backslash S = B$, and it suffices to show that

$$g_2(S,A) + g_2(T,B) \le \max\{\ g_2(S \cup T, A \cup B) + g_2(S \cap T, \emptyset), \quad (4)$$

$$g_2(S \backslash T, A) + g_2(T \backslash S, B), \quad (5)$$

$$g_2(S \cap T, \emptyset), \quad (6)$$

OR by perhaps swapping (S, A) for (T, B),

$$g_2(S \cap T, \emptyset) + g_2(T \backslash S, B)\ \}. \quad (7)$$

If $|A| = |B| = 0$, then 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 [5, 10] implies that either (4) or (5) hold.

If $|A| = |B| = 1$, then $f_2(S, A) = f_2(T, B) = 2$. In this case, either
1) $i(S, A) \in S \backslash T$ and $i(T, B) \in T \backslash S$, or
2) $\{i(S, A), i(T, B)\} \cap (S \cap T)$ is nonempty.
In the first case, (5) holds. In the second case, (6) holds.

If $|A| = 1$, $B = \emptyset$, (the case $|B| = 1$, $A = \emptyset$ may be treated symmetrically) then T may have connectivity requirement 1 or 2. If $i(S, A) \in S \backslash T$ and $j(S, A) \in T \backslash S$ then $f_2(T, \emptyset) = 2$ and (5) holds. If $i(S, A) \in S \backslash T$ and there is no corresponding $j(S, A)$ in $T \backslash S$ then $f_2(T, \emptyset) = 1$. In this case, if $i(T, B) \in T \backslash S$, then (5) holds. Otherwise, (4) holds.

The remaining case has $i(S, A) \in S \cap T$. If $j(S, A) \in V \backslash (S \cup T)$, then $f_2(T, \emptyset) = 2$ as well, and (4) holds. Otherwise, $j(S, A) \in T \backslash S$. We consider the possible values of $f_2(T, \emptyset)$. If $f_2(T, \emptyset) = 1$, then (6) holds. Otherwise, $f_2(T, \emptyset) = 2$. If $j(T, B) \in S \backslash T$, then (5) holds. If $j(T, B) \in V \backslash (S \cup A \cup T)$, then (4) holds. Otherwise, $A = \{j(T, B)\}$. In this case only, none of (4)-(6) hold, and (7) holds. □

The proof of Lemma 3 demonstrates why and when we require (7) in the description of g_2. We summarize this in the following corollary so that we may easily refer to it.

Corollary 1. *If $T \cap A = \emptyset = S \cap B$ and $g_2(S, A) + g_2(T, B)$ is strictly greater than the maximum of (4)-(6), then $|A| + |B| = 1$, and assuming $|A| = 1$ (the other case is symmetric), then $A = \{j(T, B)\}$, $i(S, A) \in S \cap T$, $j(S, A) \in T \backslash S$, and $f_2(T, B) = f_2(T, \emptyset) = 2$.* □

The example in Section 4 shows that the corresponding g_k for $r \in \{1, k\}^{V \times V}$ is in general not very weakly two-supermodular for any $k \ge 3$. For $k = 3$, this example also demonstrates that the very weak two-supermodularity inequalities do not hold in this case even for the simplification $A \cap T = B \cap S = \emptyset$ used in the above proof.

Proof of Lemma 4. Since the proof of the lemma is independent of choice of F, and the context is clear, we use δ for δ_F. Suppose $g_2(S, A) > 0$ and $g_2(T, B) > 0$. If $g_2(S, A) + g_2(T, B)$ satisfies any of (4)-(6), then by the two-submodularity of $|\delta|$, we have $(g_2 - |\delta|)(S, A) + (g_2 - |\delta|)(T, B)$ satisfies the same inequality. If $g_2(S, A) + g_2(T, B)$ does not satisfy (4)-(6), then it satisfies (7), for some permutation of S, $V\backslash(S \cup A)$, T, and $V\backslash(T \cup B)$. If $|\delta(S, A)| + |\delta(T, B)| \geq |\delta(S \cap T, \emptyset)| + |\delta(T\backslash S, B)|$, then $g_2 - |\delta|$ also satisfies (7), and we are done. Otherwise, there is an edge from $S \cap T$ to $T\backslash S$ in F, since this is the only type of edge that contributes more to the right hand side of (7) than the left. Thus $|\delta(S, A)| \geq 1$. Since $g_2(S, A) + g_2(T, B)$ does not satisfy (4)-(6), by Corollary 1, we must have that $|A| + |B| = 1$. By swapping S, A for T, B we may assume $|A| = 1$. Then $g_2(S, A) = 1$, and thus $g_2(S, A) - |\delta(S, A)| \leq 0$. Hence, none of (4)-(7) need apply to (S, A) to establish weak two-supermodularity of $g_2 - |\delta|$. □

Let x be a basic solution to $(\mathbf{MCVC_2})$ with the property that $x(e) < 1$ for all $e \in E_{res}$, and let E_x be the set of edges with nonzero x-value, and let F be the set of edges already included in the final solution. A pair (S, A) is *tight* if it satisfies

$$x(S, A) \geq f_2(S, A) - |A| - |\delta_F(S, A)| \tag{8}$$

at equality. Given x, define $\chi_x(S, A)$ to be the characteristic vector of the support of $x(S, A)$. If $x(S, A) = 0$, we say that (S, A) is *empty*. If $x(S, A) > 0$, then (S, A) is *non-empty*. If (S, A) is empty, then $\chi_x(S, A) = \mathbf{0}$.

Lemma 5 (Uncrossing Lemma). *If (S, A) and (T, B) are tight and non-empty, then for the appropriate permutation of S and $V\backslash(S \cup A)$ and T and $V\backslash(T \cup B)$ so that $A \cap T = B \cap S = \emptyset$, one of the following holds.*

i. *$(S \cap T, \emptyset)$ is tight, $(S \cup T, A \cup B)$ is either empty or tight, and* $\chi_x(S, A) + \chi_x(T, B) = \chi_x(S \cap T, \emptyset) + \chi_x(S \cup T, A \cup B)$,
ii. *$(S - T, A)$ and $(T - S, B)$ are tight and* $\chi_x(S, A) + \chi_x(T, B) = \chi_x(S - T, A) + \chi_x(T - S, B)$,
iii. *After perhaps swapping (S, A) for (T, B), then $B = \emptyset$, and $(S \cap T, \emptyset)$ and $(T - S, \emptyset)$ are tight, and* $2\chi_x(S, A) + \chi_x(T, \emptyset) = \chi_x(S \cap T, \emptyset) + \chi_x(T - S, \emptyset)$.

Proof. For simplicity of notation, let $g' = g_2 - |\delta_F|$. Since x is a solution to $(\mathbf{MCVC_2})$, $g' - x \leq 0$. Since g' is very weakly two-supermodular by Lemma 4, if (S, A) and (T, B) are both nonempty, then for appropriate permutations of S and $V\backslash(S \cup A)$, T and $V\backslash(T \cup B)$, we have that $g'(S, A) + g'(T, B)$ must satisfy one of (4)-(7) with g_2 replaced by g'. If it satisfies any of (4)-(6), then since x is two-submodular, $(g' - x)(S, A) + (g' - x)(T, B)$ satisfies the same inequality. Thus if (S, A) and (T, B) are tight, then the left hand side of the corresponding inequality in (4)-(6) equals 0. Since $g' - x \leq 0$, this implies that each part of the corresponding right hand side equals 0. Thus, if (4) is satisfied, then i. holds with $(S \cup T, A \cup B)$ tight; if (5) is satisfied then, ii. holds; and if (6) is satisfied, then i. holds with $(S \cup T, A \cup B)$ empty.

If $g'(S, A) + g'(T, B)$ does not satisfy any of (4)-(6), then neither does g_2 and by Corollary 1), by perhaps swapping (S, A) for (T, B), we have that $|A| = 1$,

$B = \emptyset$, $g_2(S,A) = 1$, $g_2(T,\emptyset) = 2$, $i(S,A) \in S \cap T$ and $j(S,A) \in T\backslash S$. Thus, $g_2(S \cap T, \emptyset) = g_2(T\backslash S, \emptyset) = 2$. Since (S,A) and $(T,\emptyset)$ are tight, then in order to satisfy the inequalities in (**MCVC**) for $(S \cap T, \emptyset)$ and $(T\backslash S, \emptyset)$, we must have that all the edges crossing (S,A) must leave $S \cap T$ and enter $T\backslash S$. Then the weights of edges leaving T are evenly split among edges from $S \cap T$ to $S\backslash T$ and edges from $T\backslash S$ to $V\backslash(S \cup T)$. This means that $x(S \cap T, \emptyset) = 2$ and $x(T\backslash S, \emptyset) = 2$, so that $(S \cap T, \emptyset)$ and $(T - S, \emptyset)$ are tight and $2\chi_x(S,A) + \chi_x(T,\emptyset) = \chi_x(S \cap T, \emptyset) + \chi_x(T - S, \emptyset)$. Thus iii. holds. □

Let $\mathcal{T}$ be the set of tight set pairs for x. Set pairs (S,A) and (T,B) are called *pair-laminar* if T and S are laminar and if T or $T \cup B$ cross S or $S \cup A$ then $A = B$. Otherwise, they are said to cross. A subset $\mathcal{L} \subset \mathcal{T}$ is called pair-laminar if all the pairs of set pairs in $\mathcal{L}$ are pair-laminar. Before establishing that $\mathcal{T}$ is spanned by a collection of pair-laminar set pairs, we need the following technical lemma.

Lemma 6. *Suppose $A \cap T = B \cap S = \emptyset$, and (S,A) crosses (T,B). If (T',B') crosses at least one of $(S \cap T, \emptyset)$, $(S \cup T, A \cup B)$, $(S - T, A)$, $(T - S, B)$, and it does not cross (T,B), then it crosses (S,A).*

Proof. We use the following easy to see fact:

$$X \text{ crosses } Y \cap Z,\ Y \cup Z,\ Y - Z, \text{ or } Z - Y \text{ but not } Z \ \Rightarrow\ X \text{ crosses } Y. \quad (9)$$

If (T',B') crosses one of the four set pairs in the lemma, then either T' crosses one of the first sets in each set pair; or $T' \cup B'$ crosses one of the first sets or T' or $T' \cup B'$ cross one of the four unions of first and second sets, and B' does not equal the corresponding second set. We consider each case in turn, progressively assuming that the previous cases do not occur.

If T' crosses $S \cap T$, $S - T$, $T - S$, or $S \cup T$, then by setting $X = T'$, $Y = S$, and $Z = T$, (9) implies it crosses S.

Otherwise, if $T' \cup B'$ crosses $S \cap T$, $S - T$, $T - S$, or $S \cup T$, then by setting $X = T' \cup B'$, $Y = S$, and $Z = T$, (9) implies it crosses S. We need to now establish that if B' does not equal the corresponding second set, then $B' \neq A$. Note that if the second set contains more than one element, then the set pair is empty, so it is not included in a collection of tight set pairs as described in Lemma 5. Suppose $B' = A$. Then the only cases of interest are $T' \cup B'$ crosses $T - S$ and $B' \neq B$, or $T' \cup B'$ crosses $S \cap T$ since in the other non-empty set pair cases A is the second set. If $T' \cup B'$ crosses $S \cap T$ or $T - S$, since by assumption, $B' \cap T = A \cap T = \emptyset$, then either $T' \subset T - S$ or $T' \subset S \cap T$. But then $T' \cup B'$ crosses T and $B' \neq B$, which contradicts (T',B') and (T,B) pair-laminar.

Otherwise, if T' crosses one of the four unions of set pairs in the lemma, then setting $X = T'$, $Y = S \cup A$, $Z = T \cup B$, implies that either T' crosses $S \cup A$ or $T \cup B$. If T' crosses $T \cup B$ then $B' = B \neq \emptyset$, so $T \subset T'$, and the only possibility is that T' crosses $(S \cup A) - T$ with $B' = B \neq A \neq \emptyset$. If T' does not cross $S \cup A$, then $T' \subseteq S \cup A$. Since $B' \neq B \notin S \cup A$ and T' does not cross S, it must be that $T' \subseteq S$. But then $T' \cup B'$ crosses S, and thus (T',B') crosses (S,A). A symmetric argument for the case of T' crossing $S \cup A$ with $B' = A$ yields a contradiction to (T', B') and (T, B) being pair laminar.

Finally, if none of the above cases hold, and $T' \cup B'$ crosses one of the four unions of set pairs, then $T' \cup B'$ crosses either $T \cup B$ or $S \cup A$. If the former holds, then $B' = B \neq \emptyset$, and $T \cap T' = \emptyset$. Thus, the only possibility of the four are that $T' \cup B'$ crosses $(S \cup A) - T$ and $B' = B \neq A \neq \emptyset$. Since T' does not cross $S \cup A$, and $B \notin S \cup A$, we have that $T' \subseteq S \cup A$. But then $T' \cup B'$ crosses S and thus (T', B') crosses (S, A). □

Corollary 2. *For any maximal, pair-laminar family $\mathcal{L}$ of tight set pairs, the following holds:* $\text{Span}(\mathcal{L}) = \text{Span}(\mathcal{T})$.

Proof. If $\text{Span}(\mathcal{L}) \neq \text{Span}(\mathcal{T})$, then $\text{Span}(\mathcal{L}) \subset \text{Span}(\mathcal{T})$, and there exists a pair $(S, A) \in \mathcal{T}$, with $(S, A) \notin \mathcal{L}$, such that (S, A) crosses a minimum number of set pairs in $\mathcal{L}$. Let (T, B) be one of those pairs. Then by Lemma 5, we can rewrite $\chi_x(S, A)$ as a linear combination of characteristic vectors of pair-laminar tight set pairs. Note that the new set pairs do not cross (T, B). Since $(S, A) \notin \text{Span}(\mathcal{L})$, at least one of these new set pairs is also not in $\text{Span}(\mathcal{L})$. By Lemma 6, any set pair $L \in \text{Span}(\mathcal{L})$ crossing any of the new sets must also have crossed (S, A). Since the new sets do not cross (T, B), they have strictly fewer crossings with sets in $\mathcal{L}$ than S does, contradicting the choice of S. □

Corollary 3. *There exists a collection $\mathcal{B}$ of pair-laminar tight set pairs satisfying*

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

We define containment on set pairs by $(S, A) \subseteq (T, B)$ if $S \subseteq T$ and $A \subseteq T \cup B$. It is easy to see that the containment relation is transitive, reflexive, and anti-symmetric. Thus it defines a partially ordered set (poset).

Lemma 7. *If $\mathcal{B}$ is a collection of pair-laminar set pairs, then the poset defined by the containment relation on the set pairs in $\mathcal{B}$ is described by a unique forest.*

The following theorem implies Theorem 2:

Theorem 4. *There is a tight set pair (S, A) with $(g_2 - \delta_F)(S, A) > 0$ and at most 2 edges in $\delta_{E_{res}}(S, A)$. Hence, at least one of these edges has x-value at least $\frac{1}{2}$.*

Proof. Using the following concept of incidence, along with Lemma 7, the proof is very similar to the proof of the corresponding statement for MCEC in [10]. Each edge $e = (i, j)$ in E has two endpoints, i_j and j_i. An endpoint i_j of an edge (i, j) is *incident* to node (S, A) if (S, A) is the lowest node in the tree among all nodes with either $i \in S$ or $\{i, j\} \in S \cup A$. A vertex i may be the endpoint of several edges; and each such endpoint may be incident to a different node of the forest. An edge *crosses* a node (S, A) if exactly one of its endpoints is incident to any node in the subtree rooted at (S, A). This assignment ensures that an edge (i, j) crosses a node (S, A) if and only if $i \in S$ and $j \in V \backslash (S \cup A)$. □

4 Examples and Counterexamples

In [10], Jain gives an example to show that the analysis of this algorithm is tight for the edge connectivity problem with connectivity requirements in $\{0, 1\}$. Since in this case the edge and vertex connectivity problems are the same, the same example shows that the analysis is also tight for the vertex connectivity problem.

A natural question is: Can we extend the arguments given here to give a constant factor approximation 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 ($\mathbf{MCVC_2}$) 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 with $r_{ij} = \min\{r_i, r_j\}$, and $r_i \in \{1, k\}$ for all $i \in V$. 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 $\frac{1}{k}$.

We depict the family of instances in Figure 1: 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. The optimal LP solution will choose every edge in the clique at value 1 and every edge in the bipartite graph at value $1/k$. After fixing all edges with $x_e = 1$, the remaining optimal LP solution will still have every edge in the bipartite graph at value $1/k$. It is not hard to establish that this is an optimal solution. For instance, consider the solution to the dual linear program that sets $y_{S,A} = 1$ for $S = \{u_i\}$ for $1 \leq i \leq k$ and $A = \emptyset$, and 0 otherwise. This is feasible, and has value equal to the feasible primal solution. Hence both are optimal.

We now establish that this is a vertex of the polytope described by ($\mathbf{MCVC_2}$) with all cost 0 edges included in $E_{\frac{1}{2}+}$. 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 $\mathbf{R}^{k^2}$, it is a vertex of the 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 $A = \emptyset$. Aside from these inequalities, the point is highly degenerate. The remaining $k - 1$ blocks of inequalities are described as follows. In block $i \in \{1, \ldots, k-1\}$, we have $v_{i+1} \in S$, $A^i = \{v_j | j \neq i, i+1\}$, and $v_i \in V \backslash (S \cup A^i)$. Thus there is exactly one cost-0 edge crossing this cut (edge (v_i, v_{i+1})), and the $(g_2 - \delta_F)$-value of the inequality is 1. Denote the set S for inequality $q \in \{1, \ldots, k\}$ in this block by S^i_q. We set $S^i_q := \{v_{i+1}, u_1, u_2, \ldots, u_{k-q+1}\}$. See Figure 1. Then $x(S^i_q, A^i)$ is

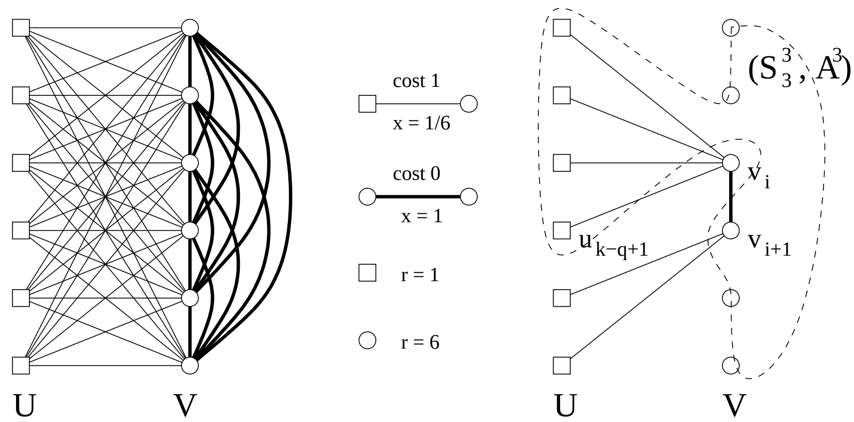

Fig. 1. On the left, a basic solution to $(\mathbf{MCVC_2})$ after fixing the cost 0 edges to 1. The largest fraction in the solution is $\frac{1}{k}$, here $k = 6$. On the right, an example of a set (S^i_q, A^i) for $i = 3$, $q = 3$ with $f_2(S^i_q, A^i) = k$. All the edges crossing the cut are included in the figure. They have total value 2. Together with the 4 vertices in the cutset, this satisfies (8) at equality. The collection of cuts $\{(S^i_q, A^i)\}_{1\leq i\leq k-1,\ 1\leq q\leq k}$ are highly crossing.

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.

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. 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. All the edges incident to u_i are in the i^{th} block. Within a block, the j^{th} edge is the edge incident to v_j. See Figure 2.

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 columns (called column block 0) has a pattern that is slightly different from the rest. See Figure 3. The first column is 0 everywhere except in the last entry in the first row block, which is 1. The second thru 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} column block, $i = 1, \ldots, k-1$, see Figure 4. 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 Z_i containing i entries of $\frac{k-i}{k}$ followed by $k-i$ entries of $\frac{-i}{k}$. The q^{th} column of the $k-q^{th}$ row block

	u_1	u_2	$\cdots$	u_{k-1}	u_k
block 0	1 1 $\cdots$ 1 1				
		1 1 $\cdots$ 1			
			$\ddots$		
				1 1 $\cdots$ 1	
					1 1 $\cdots$ 1
block 1	1 0 $\cdots$ 0 0	1 0 $\cdots$ 0	$\cdots$	1 0 $\cdots$ 0	1 0 $\cdots$ 0
	1 0 $\cdots$ 0 0	1 0 $\cdots$ 0	$\cdots$	1 0 $\cdots$ 0	0 1 $\cdots$ 0
	1 0 $\cdots$ 0 0	1 0 $\cdots$ 0	$\cdots$	0 1 $\cdots$ 0	0 1 $\cdots$ 0
	$\vdots$	$\vdots$	$\vdots$		
	1 0 $\cdots$ 0 0	0 1 $\cdots$ 0	$\cdots$	0 1 $\cdots$ 0	0 1 $\cdots$ 0
block 2	0 1 $\cdots$ 0 0	0 1 $\cdots$ 0	$\cdots$	0 1 $\cdots$ 0	0 1 $\cdots$ 0
	$\vdots$	$\vdots$	$\ddots$		
	$\vdots$	$\vdots$	$\ddots$		
block k-1	$\vdots$	$\vdots$	$\vdots$		
	0 0 $\cdots$ 1 0	0 0 $\cdots$ 1	$\cdots$	0 0 $\cdots$ 1	0 0 $\cdots$ 1

Fig. 2. Incidence Matrix C of k^2 Tight Set Pairs

column	1	2	3	$\cdots$	k
row	0	$\frac{-1}{k}$	$\frac{-1}{k}$	$\cdots$	$\frac{-1}{k}$
block 1	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	0	$\frac{-1}{k}$	$\frac{-1}{k}$	$\cdots$	$\frac{-1}{k}$
	1	$\frac{k-1}{k}$	$\frac{k-1}{k}$	$\cdots$	$\frac{k-1}{k}$
	0	$\frac{1}{k}$	0	$\cdots$	0
block 2	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	0	$\frac{1}{k}$	0	$\cdots$	0
	0	0	$\frac{1}{k}$	$\cdots$	0
block 3	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	0	0	$\frac{1}{k}$	$\cdots$	0
	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$
	0	0	0	$\cdots$	$\frac{1}{k}$
block k	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
	0	0	0	$\cdots$	$\frac{1}{k}$

Fig. 3. Column Block 0 of Matrix B

Col. Block i of B

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

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

Fig. 4. On the left, the pattern of column block i of matrix B. On the right, the composition of the vectors X_i, Y_i and Z_i that describe block i.

is Z_i for $1 \leq q \leq k$, and the q^{th} column of the $k - q + 1^{st}$ row block is the vector Y_i with i entries of $\frac{-k+i}{k}$ followed by $k - i$ entries of $\frac{i}{k}$.

The following lemma follows by inspection of B and C.

Lemma 8. *For any k, matrices B and C are inverses.*

5 Algorithmic Details

To solve the LP in polynomial time, we need a separation algorithm for the connectivity constraints. We interpret x-values as capacities and transform the graph induced by the current fractional solution and the fixed edges 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 vertices to model the fact that at most one path can pass through any vertex. 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 algorithm for ($\mathbf{MCVC_2}$). Using ellipsoid algorithm, we can solve the LP in polynomial time. Once we have a solution, it may not be a basic solution. However, it can be transformed to a vertex solution in polynomial time, as described in [7, 10].

6 Acknowledgment

I thank Joseph Cheriyan for his suggestion and encouragement to work on minimum cost vertex connectivity problems and for interesting conversations. I thank Kirsten Wickelgren for her enthusiasm in learning about combinatorial optimization, which led me to revisit this problem. I am also grateful to the Fields Institute, Toronto, Ontario and Bell Labs, Lucent Technologies, Murray Hill, New

Jersey for hosting me during some parts of this work. Additional support was provided by NSF through grants EIA-9973858 and CCR-9985458.

References

1. A. Bouchet. Greedy algorithm and symmetric matroids. *Math. Programming*, 38:147–159, 1987.
2. R. Chandrasekaran and S. N. Kabadi. Pseudomatroids. *Discrete Math.*, 71:205–217, 1988.
3. J. Cheriyan, T. Jordán, and Z. Nutov. Approximating k-outconnected subgraph problems. In *Approximation algorithms for combinatorial optimization (Aarlborg)*, number 1444 in Lecture Notes in Comput. Sci., pages 77–88. Springer, Berlin, 1998.
4. R. Diestel. *Graph Theory.* Number 173 in Graduate Texts in Mathematics. Springer-Verlag, New York, 2nd edition, 2000.
5. 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.
6. M. X. Goemans and D. P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In Hochbaum [9], pages 144–191.
7. M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag, 1988.
8. M. Grötschel, C. L. Monma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low-connectivity constraints. *Operations Research*, 40(2):309–330, March-April 1992.
9. D. S. Hochbaum, editor. *Approximation Algorithms for NP-Hard Problems.* PWS Publishing Company, Boston, 1997.
10. K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. In *39th Annual IEEE Symposium on Foundations of Computer Science*, 1998.
11. S. Khuller. Approximation algorithms for finding highly connected subgraphs. In Hochbaum [9], pages 236–265.
12. S. Khuller and B. Raghavachari. Improved approximation algorithms for uniform connectivity problems. *J. Algorithms*, 1996.
13. 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.
14. V. Melkonian and E. Tardos. Approximation algorithms for a directed network design problem. In *7th International Integer Programming and Combinatorial Optimization Conference*, pages 345–360, 1999.
15. C. L. Monma and D. F. Shallcross. Methods for designing communications networks with certain two-connected survivability constraints. *Operations Research*, 1989.
16. R. Ravi and D. P. Williamson. An approximation algorithm for minimum-cost vertex-connectivity problems. *Algorithmica*, 18(1):21–43, 1997.
17. R. Ravi and D. P. Williamson, November 2000. Personal communication.

This research was sponsored in part by National Science Foundation (NSF) grant no. CCR-0122581.