Compacting cuts: a new linear formulation for minimum cut

Robert D. Carr* Goran Konjevod† Greg Little‡ Venkatesh Natarajan§
Ojas Parekh¶

Abstract

For a graph (V, E), existing compact linear formulations for the minimum cut problem require $\Theta(|V||E|)$ variables and constraints and can be interpreted as a composition of $|V|-1$ polyhedra for minimum s-t cuts in much the same way as early approaches to finding globally minimum cuts relied on $|V|-1$ calls to a minimum s-t cut algorithm. We present the first formulation to beat this bound, one that uses $O(|V|^2)$ variables and $O(|V|^3)$ constraints. An immediate consequence of our result is a compact linear relaxation with $O(|V|^2)$ constraints and $O(|V|^3)$ variables for enforcing global connectivity constraints. This relaxation is as strong as standard cut-based relaxations and has applications in solving traveling salesman problems by integer programming as well as finding approximate solutions for survivable network design problems using Jain's iterative rounding method. Another application is a polynomial-time verifiable certificate of size n for for the NP-complete problem of l_1-embeddability of a rational metric on an n-set (as opposed to a certificate of size n^2 known previously).

1 Introduction

Mathematical programming has enjoyed a burgeoning presence in theoretical computer science, both as a framework for developing algorithms and, increasingly, as a bona fide model of computation whose limits are expressed in terms of sizes of formulations and integrality gaps of formulations [3, 1, 21].

The impact of mathematical programming is also evident in algorithms for combinatorial optimization problems. In fact, some of the most celebrated algorithmic results of the last decade have been achieved by the reduction of the problem in question to a linear or semidefinite programming model [17, 19]. In many cases, the algorithms are directly based on such mathematical programming models and no combinatorial algorithms are known for such problems. Even those algorithms that do not rely directly on a mathematical program solver often have designs that were explicitly governed by insights gained from a mathematical programming model (for example, primal-dual algorithms, see [29]). It is not difficult to see why linear formulations are an appealing model of computation: both optimization and decision problems fit naturally into the framework, and both theoretically tractable and efficient practical algorithms exist for solving linear programs.

*Discrete Math. Group, Sandia National Labs, Albuquerque, NM 87185. Email: `bobcarr@cs.sandia.gov`.
†Dept. of Comp. Sci. and Eng., Arizona State University, Tempe, AZ 85287-5406. Email: `goran@asu.edu`. This author's work was supported in part by the NSF grant CCR-0209138.
‡CSAIL, Massachusetts Institute of Technology, Cambridge, MA 02139. Email: `glittle@gmail.com`.
§Dept. of Math. Sciences, Carnegie Mellon University, Pittsburgh, PA 15213. Email: `vn@andrew.cmu.edu`.
¶Math. and Comp. Sci. Dept., Emory University, Atlanta, GA 30322. Email: `ojas@mathcs.emory.edu`.

Smaller formulations for linear relaxations of NP-hard problems are important in the design of efficient approximation algorithms. Perhaps more surprising is the fact that smaller formulations for problems already known to be in P can have an impact on both the exact and approximation solution of NP-hard problems. For instance, state of the art approaches to exactly solving large scale instances of many NP-hard problems rely on integer programming approaches that require the repeated solution of integer programs representing problems in P [8]. Polynomial-sized exact linear formulations for these problems can be used in place of the integer programs to potentially improve performance [8]. In the case of the traveling salesman problem [2], one of the most famous combinatorial optimization problems, minimum cut is one such subroutine, and smaller exact formulations for it are likely to yield faster exact algorithms for large instances traveling salesman problem.

From a theoretical point of view, linear formulation complexity offers a different perspective than that offered by other prominent models of computation. For instance even though algorithmically, spanning tree is considered a relatively easy to solve member of P, the smallest known linear formulation requires $\Theta(|V||E|)$ constraints and variables [24]. In fact, before the results we present in this paper, the same was true of minimum cut [13, 28]. The situation of the matching problem is a most peculiar one, since this problem can be solved in polynomial time, yet has resisted all attempts to find polynomial-size linear formulations. Another highly interesting fact about this problem is that the matching problem can be solved in polynomial time by solving an exponential-size linear program. The foundation of this result is the equivalence between separation and optimization problems in linear programming (see also Section 2).

Some of these results can be placed in a better context by observing that Turing machine complexity is perhaps not the ideal model for comparison since the size of a linear program seems to intuitively depend upon an efficient encoding of feasible solutions. Yannakakis observed this and forged a connection between linear formulation complexity and nondeterministic communication complexity [30]. He also showed that any symmetric linear programming formulation of matching or of the traveling salesman problem must have exponential size. These results represent one of the earliest results on linear formulation complexity.

The results of Yannakakis are not unconditional—they give lower bounds only for linear programming formulations that satisfy a symmetry condition. In fact, Yannakakis even says "it is not clear what can be gained by treating one node differently than another." Our result can be seen as an example where additional asymmetry seems to allow a more compact linear formulation (however, we do not have a proof of this, such as would follow from a tight lower bound on the size of a symmetric formulation).

Martin [24] studied the relationship between optimization and separation problems (see also Section 2), and observed that his results imply a linear formulation for the minimum spanning tree problem with $\Theta(|V|^3)$ variables and $\Theta(|V|^3)$ constraints. This is no more compact than the standard relaxation [23], and Martin asked whether there exists a linear formulation with $O(|V|^2)$ variables or constraints. While this remains an open problem, in this paper we answer the analogous question positively for the related minimum cut problem.

1.1 Minimum cut

The *connectivity* of a weighted graph (V, E) is the minimum total capacity of a set of edges whose removal disconnects the graph. The problem of finding the connectivity of a (weighted) graph is called the *(global) minimum cut*, or *min cut*, problem.

Traditional linear programming formulations for the minimum cut problem [28] require $\Theta(|V||E|)$ variables and constraints and can be interepted as a composition of $(|V|-1)$ minimum s-t cut polyhedra (see [28] and Proposition 2 in [13]) in much the same way as the early algorithms for globally minimum cuts rely on $|V|-1$ calls to a minimum s-t cut algorithm. Better algorithms have been found, for example the $O(|V||E|)$-time algorithm by Nagamochi and Ibaraki [25], simplified by Stoer and Wagner [27], and the almost linear randomized algorithm by Karger [20], but no linear programming formulation has been known that uses $o(|V||E|)$ variables or constraints.

1.2 Our results

We show that the global minimum cut problem has linear formulation complexity $o(|V||E|)$. In particular, we show this problem can indeed be modeled in a way that takes advantage of global connectivity, as opposed to simply composing $|V|-1$ $(s-t)$-cut formulations.

Our formulation requires only $O(|V|^2)$ variables and $O(|V|^3)$ constraints, thus reducing the number of variables by a factor of $|V|$ on dense graphs. We give two proofs of our result. The first is based on the powerful graph-theoretic notion of splitting-off [16, 22] and uses induction on the size of a minimal counterexample. Our formulation can be seen (in a dual sense) as a sequence of splitting-off operations that preserve (local) connectivity while reducing the graph. The second proof is somewhat longer, but it is both elementary and constructive, in that it shows how to efficiently reconstruct an integral solution from any optimal fractional solution. This allows us to recover a convex combination of cuts dominated by any given feasible fractional solution to the linear program.

Also of note is that using our formulation one can obtain an $O(|V|^2) \times O(|V|^3)$ relaxation for the k-edge-connected spanning subgraph problem, which is that of finding a minimum cost k-edge-connected subgragph in a weighted graph. The relaxation obtained is equivalent in strength to the standard cut-based relaxation but has implications for algorithms that explicitly rely on solving LP relaxations, such as Jain's iterative rounding procedure [19].

Another consequence of our work is a smaller polynomially-verifiable certificate for ℓ_1-embeddability of metrics, consisting of $\Theta(n)$ instead of $\Theta(n^2)$ rational values (see Section 5).

Practical applications of our work are mostly to solving large-scale integer programs, where parts of the formulation may impose global connectivity constraints, and our formulation will help reduce the size of the instance.

1.3 Preliminaries

We use mostly standard notation that is consistent with Schrijver's text [26]. We let $V_n = \{1, \ldots, n\}$ and by E_n we mean the edge set $\{\{i,j\} \mid i \in V_n,\, j \in V_n\}$. We abbreviate $\{i,j\}$ as ij, or equivalently ji; when an edge ij is the index of a component of a vector, we write i,j instead. For a graph $G = (V,E)$, if we are a given a set $S \subseteq V$, the we define the edge sets $\delta(S) = \{i,j \in E \mid |\{i,j\} \cap S| = 1\}$ and $E(S) = \{i,j \in E \mid |\{i,j\} \cap S| = 2\}$; we abbreviate $\delta(\{v\})$ as $\delta(v)$. The graph $G - v$, for some $v \in V$, is obtained from G by deleting v and all edges in $\delta(v)$. We let $con(G,c)$ refer to the value of a minimal cut in G with respect to the cost vector c; if the value of c is clear from context we simply use $con(G)$.

Given sets S and I, we use S^I to refer to an $|I|$-dimensional vector space in which the components of each vector in S^I are in a correspondence with the elements of I. For a vector $x \in S^I$, and a set $J \subseteq I$, by $x(J)$ we mean $\sum_{j \in J} x_j$; the vector $x|_J \in S^J$ is obtained from the vector $x \in S^I$ by dropping all components in $I \setminus J$.

We refer the reader to Yannakakis [30] for a more detailed discussion of linear formulations of optimization problems; here, by an $M \times N$ formulation we mean a linear program with M constraints, excluding nonnegativity constraints, and N variables. By a *compact formulation* we mean one in which M and N are polynomial in the representation size of the optimization problem instance.

2 Compact relaxations for STSP and network design relaxations

The motivation for our compact minimum cut formulation has roots in compact linear relaxations for the symmetric traveling salesman problem (STSP), which given a complete graph $G = (V = V_n, E = E_n)$ with edge costs c, consists in finding a minimum cost Hamilton cycle in G. The classic subtour elimination polytope was defined by the linear relaxation for STSP given by Dantzig et. al [14]:

$(SEP(n))$ Minimize $\sum_{ij \in E} c_{i,j} x_{i,j}$

subject to

(1) $x(\delta(v)) = 2 \quad \forall v \in V$

(2) $x(\delta(S)) \geq 2 \quad \forall S \subseteq V : \emptyset \neq S \neq V$

(3) $1 \geq x_{i,j} \geq 0 \quad \forall ij \in E.$

For a more thorough exposition consult, for instance, Schrijver [26], Chapter 58. The LP $(SEP(n))$ has an exponential number of constraints, (2), however, an optimal solution to can be obtained in polynomial time via the ellipsoid algorithm since the separation problem for (2) is precisely minimum cut. We refer the reader to Chapter 6 of Grötschel, Lovász and Schrijver [18] for an in-depth discussion of the relation between separation and optimization. In fact, by linear programming duality, we have the following:

Lemma 1 ([24, 11])**.** *From an $M \times N$ compact formulation of the separation problem associated with an optimization problem, one can generate an $(N+1) \times M$ compact formulation for the optimization problem.*

Applying this lemma to a compact linear formulation for minimum cut yields a compact set of inequalities which are as strong as and may replace (2) in $(SEP(n))$. Thus our results give a compact formulation equivalent to $(SEP(n))$ with $O(n^3)$ variables and $O(n^2)$ constraints. Previously the smallest known equivalent formulations had size $\Theta(n^3) \times \Theta(n^3)$. Both Arthanari [4, 5] and Carr [10, 9] had proposed such relaxations. The formulation derived from our work, which we present below, can be seen as an extension of Carr's cycle-shrink relaxation [10]; however, we note that Arthanari's multistage-insertion relaxation [5], is equivalent in strength [6].

For any positive integer r, the set of linear inequalities we derive can be used in any linear relaxation in place of the constraints, $x(\delta(S)) \geq r$ for all $S \subseteq V : \emptyset \neq S \neq V$. Let $V_1 = V \setminus \{1\}$. We introduce $n-1$ vectors of variables, $y^k \in \mathbb{R}^{E_k}$ for $k \in V_1$, where one may think of each y^k as representing the edges in the complete graph (V_k, E_k):

$$\sum_{k \in V_1} y_e^k = x_e \quad \forall e \in E \tag{4}$$

$$y^k(\delta(k) \cap E_k) \geq r \quad \forall k \in V_1 \tag{5}$$

$$y^k(\delta(i) \cap E_k) \geq 0 \quad \forall k \in V_1,\ i \in V : i < k \tag{6}$$

$$y_{i,k}^k \geq 0 \quad \forall k \in V_1,\ i \in V : i < k \tag{7}$$

$$y_{i,j}^k \leq 0 \quad \forall k \in V_1,\ ij \in E_{k-1}. \tag{8}$$

We observe that there are $O(n^3)$ variables and $O(n^2)$ constraints, excluding the nonnegativity and nonpositivity constraints (7) and (8), respectively. If the variables were integral, we could inductively think of each y^k as adding a new vertex k to the r-connected subgraph of E_{k-1} defined by $y^2 + \cdots + y^{k-1}$ in a way that preserves r-connectivity. The inequalities insist that r edges among $\delta(k) \cap E_k$ are added, which is modeled by (5) and (7). Additionally some edges in E_{k-1} which are selected in $y^2 + \cdots + y^{k-1}$ are allowed to be deleted in a way that preserves r-connectivity; this is modeled by (6) and (8). We do not prove this explicitly here, but note that this follows from the results of the next section.

Theorem 2. *The constraints* (4)-(8) *are equivalent to* $x(\delta(S)) \geq r$ *for all* $S \subseteq V : \emptyset \neq S \neq V$*.*

With the results we prove in the next section, this theorem follows through some algebraic manipulation and the constructive proof of Lemma 1 [24, 11], which essentially follows from linear programming duality.

Our formulation also has applications for other network design problems. The inequalities (4)-(8) can be extended to model generalizations such as Steiner cuts, and we believe that they also extend to generalized Steiner cuts, although we have not verified the latter claim.

3 The formulation

We start by taking a complete graph $G = (V = V_n,\ E = E_n)$ with an edge-cost vector $c \in \mathbb{Q}_+^E$. Our formulation has a variable $x_{i,j}$ for each edge ij, and requires only $n-1$ additional variables and $O(n^3)$ constraints:

$$(P(n)) \qquad \text{Minimize} \sum_{ij \in E} c_{i,j} x_{i,j}$$

subject to

$$\sum_{2 \leq i \leq n} z_i = 1 \tag{9}$$

$$x_{i,k} + x_{j,k} \geq x_{i,j} + 2z_k \quad \forall ij \in E,\ k \in V : ij < k \tag{10}$$

$$x_{i,k} \geq z_k \quad \forall i \in V,\ k \in V : i < k \tag{11}$$

$$x_{i,j} \geq 0 \quad \forall ij \in E \tag{12}$$

$$z_i \geq 0 \quad \forall i \in V \setminus \{1\} \tag{13}$$

To establish the correctness of the above formulation, we begin by showing that the incidence vectors of cuts in G can indeed be extended to feasible solutions of $(P(n))$.

Lemma 3. *If x is the incidence vector of a cut $C \subseteq E$ then there exists a $z \in \{0,1\}^{V \setminus \{1\}}$ such that $(x, z) \in (P(n))$.*

Proof. Let $S \subset V$ be the set such that $C = \delta(S)$ and $1 \in S$. We construct a $z \in \{0,1\}^{V \setminus \{1\}}$ such that $(x, z) \in (P(n))$. If l is the least index of any vertex in $V \setminus S$, we set $z_l = 1$ and $z_i = 0$ for $i \neq l$. Observe that our choice of z satisfies (9), (13), and the members of (11) and (10) with $k \neq l$, since the latter simply reduce to standard triangle inequalities in this case. It remains to consider (11) and (10) when $k = l$.

By our choice of l, we have $l \in V \setminus S$ while $i \in S$ for all $i < l$, hence $x_{i,l} = 1$ for all $i < l$, and (x, z) satisfies (11). Similarly for a member of (10) with $k = l$ we have $i, j \in S$, hence $x_{i,k} = x_{j,k} = 1$, $x_{i,j} = 0$, and the inequality is satisfied. □

Next we show that the projections of integral solutions of $(P(n))$ onto the x variables are indeed related to cuts in G.

Lemma 4. *If $(x, z) \in (P(n))$ is integral then x dominates the incidence vector of a cut $C \subseteq E$.*

Proof. We assume $n \geq 2$ is the minimal value for which the statement does not hold. If $n = 2$, then (9) and (11) imply $x_{1,2} \geq 1$, so we must have $n > 2$. By (9) and (13) there exists an $l \in V \setminus \{1\}$ such that $z_l = 1$. If $l = n$, then by considering (11) for each $i < n$, we see that x dominates $\delta(\{n\})$, hence we must have $l \neq n$.

If we let $V' = V \setminus \{1, n\}$, $x' = x|_{E_{n-1}}$ and $z' = z|_{V'}$, then observe that $(x', z') \in (P(n-1))$. Thus by our initial assumption on the minimality of n, there exists a set $S \subset V'$ such that x' dominates $\delta(S)$; let $T = V' \setminus S$. It suffices for us to show that x dominates either $\delta(S \cup \{n\})$ or $\delta(T \cup \{n\})$. This does not hold only if there are $i \in S$ and $j \in T$ such that $x_{i,n} = x_{j,n} = 0$; however, since $x_{i,j} = 1$, the inequality of (10) with i and j as defined above and $k = n$ precludes this. □

Lemma 3 establishes that $(P(n))$ is a relaxation for the set of incidence vectors of minimum cuts: the optimal value of $(P(n))$ is at most $con(G)$. Our goal in this section is to show that this bound is tight, which by way of a convex decomposition lemma of Carr and Vempala [12] implies that $(P(n))$ has an integral optimum solution. In the next section we give an explicit procedure for obtaining such an integral convex decomposition of a feasible fractional solution.

Theorem 5. *The optimal value of $(P(n))$ is equal to $con(G)$.*

We will need the following well-known lemma due to Lovász.

Lemma 6 ([22], Exercise 6.51)**.** *Let G be an Eulerian multigraph, $x \in V(G)$, and suppose G is k-edge-connected between any two vertices $u, v \neq x$. Then we can find two neighbors y, z of x such that, if we remove two edges xy and xz but join y to z by a new edge, the resulting graph is still k-edge-connected between any two vertices u, $v \neq x$. Morever, if x has at least two distinct neighbors, then there exist ditinct y, z satisfying the above conditions.*

We refer to the operation outlined by this lemma as a *splitting-off* operation (x, y, z) at the vertex x. The distinctness condition is not always stated but can be guaranteed [16]. The key observation is that we may interpret each inequality (10) as corresponding to a splitting-off operation in G. We explore and make this idea precise in the proof of the lemma below.

Lemma 7. *If $G = K_n$ and c is an even integral vector, then for any vector $(x, z) \in (P(n))$ there exists a vector $d \in \mathbb{Z}_+^{V\setminus\{1\}}$ such that $c \cdot x \geq d \cdot z$ and $con(G, c) = \min_{2\leq i\leq n} d_i$.*

Proof. We proceed by induction on n. If $n = 2$, then the inequality (11) implies $c_{1,2}x_{1,2} \geq c_{1,2}z_2 = con(G, c) \cdot z_2$ for any feasible (x, z). If $n > 2$ we let $G' = (V, E')$ be the multigraph that contains exactly $c_{u,v}$ parallel edges between each pair of vertices u, v, so that

$$con(G, c) = con(G', \mathbb{1}). \tag{14}$$

Lemma 6 guarantees a sequence of splitting-off operations, $S = ((n, s_1, t_1), \ldots, (n, s_k, t_k))$ in G' at vertex n such that: (i) vertex n has a single neighbor $s \in V$ in the resulting multigraph H', and (ii) $con(G', \mathbb{1}) = \min\{|\delta_{G'}(n)|, con(H', \mathbb{1})\}$. Moreover, since no minimal cut in H' may separate n and s, we have

$$con(G', \mathbb{1}) = \min\{|\delta_{G'}(n)|, con(H' - n, \mathbb{1})\}. \tag{15}$$

For each pair $u, v \in V$, we let $h_{u,v}$ be the number of edges between u and v in $H' - n$, so that

$$con(H' - n, \mathbb{1}) = con(G - n, h). \tag{16}$$

If $(x, z) \in (P(n))$ then observe that the restriction $(x, z)|_{E_{n-1}\times V_{n-1}\setminus\{1\}}$ belongs to $(P(n - 1))$, hence by our inductive hypothesis, there exists a vector $d \in \mathbb{Z}_+^{V_{n-1}\setminus\{1\}}$ such that $h \cdot x|_{E_{n-1}} \geq d \cdot z|_{\{2,\ldots,n-1\}}$ and $con(G - n, h) = \min_{2\leq i\leq n-1} d_i$. Setting $d_n = |\delta_{G'}(n)|$, our goal is to use the above bound on $h \cdot x|_{E_{n-1}}$ to show that $c \cdot x \geq d \cdot z + d_n z_n$, which would imply the lemma since by (14), (15), and (16) we have already established $con(G, c) = \min\{|\delta_{G'}(n)|, con(G - n, h)\} = \min_{2\leq i\leq n} d_i$.

We derived h from c by the sequence of splitting-off operations S, and we observe that each triple in S corresponds to an equality of (10) whose coefficients model the corresponding operation (in this calculation, the addition and equality signs on the left-hand-side refer to addition of whole inequalities, not just the left-hand-sides):

$$\begin{aligned}
& \sum_{i,j\in E(K_{n-1})} h_{i,j}x_{i,j} \geq \sum_{2\leq i\leq n-1} d_i z_i \\
+ \quad & \sum_{1\leq i\leq k} (x_{s_i,n} + x_{t_i,n} - x_{s_i,t_i} \geq 2z_n) \\
+ \quad & |\delta_{H'}(n)| \cdot (x_{s,n} \geq z_n) \\
= \quad & \sum_{i,j\in E(K_n)} c_{i,j}x_{i,j} \geq d_n z_n + \sum_{2\leq i\leq n-1} d_i z_i
\end{aligned}$$

□

Proof of Theorem 5. Without loss of generality we may assume that c has been scaled so that it is an integral vector whose components are even. For any $d \in \mathbb{Z}_+^{V\setminus\{1\}}$ and feasible solution $(x, z) \in (P(n))$, (9) and (13) imply that $d \cdot z \geq \min_{2\leq i\leq n} d_i$. Thus by Lemma 7 we have $c \cdot x \geq con(K_n, c)$. □

4 Constructive decomposition

In this section, we give an alternative proof of Theorem 5. As opposed to the one we presented in Section 3, this is a constructive proof, and as such provides an algorithm to extract an integral solution, from an optimal solution to the linear program $(P(n))$.

It is easy to describe how to extract an integral solution. We first do this and then focus on proving the correctness of our procedure. As a corollary of our result, we find that any feasible solution to the linear program $(P(n))$ dominates a convex combination of cuts.

Assumptions. Let (x^*, z^*) be an optimal solution to the LP $(P(n))$. We assume that (x^*, z^*) is *minimal*, that is, for any $x < x^*$, the vector (x, z^*) is not feasible for the linear program. This assumption can be made without loss of generality, because the cost $c_{i,j}$ is nonnegative for every edge ij. To simplify the exposition, we'll sometimes refer to $x_{j,i}$ as $x_{i,j}$ for $i < j$, and assume for convenience that $x_{i,i} = 0$ for all i.

4.1 Extracting an integral solution

We use the following notation:

$$\begin{aligned} k^A &= \max\{i \mid z_i^* > 0\} \\ S &= \Big\{\{j,k\} \mid x_{j,k}^* > 0 \text{ and } (x_{k^A,j}^* = 0 \text{ or } x_{k^A,k}^* = 0)\Big\} \\ \lambda &= \min\left(\{z_{k^A}^*\} \cup \{x_{i,j}^* \mid \{i,j\} \in S\}\right) \end{aligned}$$

Now we can define the integral vector (x^A, z^A):

$$x_{i,j}^A = \begin{cases} 1, & \{i,j\} \in S \\ 0, & \{i,j\} \notin S \end{cases} \qquad z_i^A = \begin{cases} 1, & i = k^A \\ 0, & i \neq k^A \end{cases} \tag{17}$$

We will show that (x^A, z^A) is feasible for the linear program, and therefore, by Lemma 3, a cut.

Theorem 8. *$\{x^A, z^A\}$ is a valid solution to the LP.*

When we "peel away" A, what remains we call B:

$$x_{i,j}^B = \frac{1}{1-\lambda} \cdot \begin{cases} x_{i,j}^* - \lambda, & \{i,j\} \in S \\ x_{i,j}^*, & \{i,j\} \notin S \end{cases} \qquad z_i^B = \frac{1}{1-\lambda} \cdot \begin{cases} z_i^* - \lambda, & i = k^A \\ z_i^*, & i \neq k^A \end{cases} \tag{18}$$

Just like (x^A, z^A), this new pair of vectors forms a feasible solution:

Theorem 9. *$\{x^B, z^B\}$ is a valid solution to the LP.*

The main result of this section is summarized in Theorem 10.

Theorem 10. *If (x^*, z^*) is a minimal optimal solution to the linear program $(P(n))$, then (x^A, z^A) is an optimal integral solution and its cost is equal to the cost of (x^*, z^*).*

4.2 Convex decomposition into cuts

Equation (17) and Theorem 10 show us how to round an optimal fractional solution of LP ($P(n)$) to an integral solution of equal cost. A stronger statement follows directly:

Theorem 11. *Any minimal feasible solution to the linear program* ($P(n)$) *is a convex combination of cuts.*

Proof. Let (x^*, z^*) be a minimal feasible solution, and define (x^A, z^A) and (x^B, z^B) by (17) and (18). If (x^B, z^B) is integral, we are done. Otherwise, we would like to treat (x^B, z^B) just like we did (x^*, z^*) and continue by extracting another integral solution. Such a process can go on for at most $n - 1 + \binom{n}{2}$ steps, because each time at least one additional variable is reduced to 0. The resulting cuts then give the claimed convex combination.

This will be possible, as long as (x^B, z^B) is guaranteed to be minimal. To see that this holds, suppose that $(x^C, z^C) < (x^B, z^B)$ were a feasible solution. Then $\lambda(x^A, z^A) + (1 - \lambda)(x^C, z^C)$ would be both feasible (as a convex combination of two feasible solutions) and strictly dominated by (x^*, z^*). Since (x^*, z^*) is minimal, it follows that (x^B, z^B) is minimal. □

4.3 Proofs

Lemma 12. *If $j < k$ and $x^*_{j,k} > z^*_k$, then there exists $i < k$, $i \neq j$ such that $x^*_{i,k} + x^*_{j,k} - 2z^*_k = x^*_{i,j}$.*

Proof. By minimality, $x^*_{j,k}$ cannot be reduced without violating some constraint. If $x^*_{j,k} > z^*_k$, the only active constraint for $x^*_{j,k}$ can be one of the constraints (10). □

Lemma 13. *(Triangle inequality.) If $i \neq j \neq k$ then $x^*_{i,j} + x^*_{i,k} \geq x^*_{j,k}$.*

Proof. Let m be the minimum possible value for which there exists a triple $\{i, j, k\}$ such that

$$x^*_{i,j} + x^*_{i,k} < x^*_{j,k} \tag{19}$$

and $m = \max\{i, j, k\}$. By symmetry, we may assume $j < k$.

In the case where $j < k < i$, the LP constraint $x^*_{j,i} + x^*_{k,i} - 2z^*_i \geq x^*_{j,k}$ contradicts the assumption (19).

In the case where $i < k$, we consider $x^*_{j,k}$. If $x^*_{j,k} = z^*_k$, then

$$\begin{aligned} x^*_{i,k} + x^*_{i,j} - x^*_{j,k} &\geq x^*_{i,k} - x^*_{i,j} - x^*_{j,k} \\ &= x^*_{i,k} + x^*_{j,k} - 2x^*_{j,k} - x^*_{i,j} \\ &= x^*_{i,k} + x^*_{j,k} - 2z^*_k - x^*_{i,j} \\ &\geq 0. \end{aligned}$$

If $x^*_{j,k} > z^*_k$, we have to do a little more work. First, by Lemma 12, there exists a p such that $p < k$, $p \neq j$ and

$$x^*_{j,k} + x^*_{p,k} - 2z^*_k = x^*_{p,j}. \tag{20}$$

If $p = i$, this becomes $x^*_{j,k} + x^*_{i,k} - 2z^*_k = x^*_{i,j}$. Since $x^*_{i,k} - z^*_k \geq 0$, we can subtract this quantity twice from the left side to obtain $x^*_{j,k} - x^*_{i,k} \leq x^*_{i,j}$, which again contradicts the assumption (19).

If $p \neq i$, then the fact that $i \neq j \neq p$, and $i, j, p < m$ and the choice of (minimal) m imply that $x^*_{p,i} + x^*_{i,j} \geq x^*_{p,j}$. We substitute the left-hand side for $x^*_{p,j}$ in (20) to get $x^*_{p,i} + x^*_{i,j} \geq x^*_{j,k} + x^*_{p,k} - 2z^*_k$. We also know from an LP constraint that $x^*_{i,k} + x^*_{p,k} - 2z^*_k \geq x^*_{p,i}$, which we can use to make the left side of our equation a little bigger: $x^*_{i,k} + x^*_{p,k} - 2z^*_k + x^*_{i,j} \geq x^*_{j,k} + x^*_{p,k} - 2z^*_k$. After some simple arithmetic, we arrive at $x^*_{i,k} + x^*_{i,j} \geq x^*_{j,k}$, which contradicts the assumption (19). □

In order to show that both (x^A, z^A) and (x^B, z^B) are feasible, we will examine closely the structure of the graph induced by x^A. The next few lemmas list some useful properties of its edge-set S.

Lemma 14. *If $\{j, k\} \in S$, then either $x^*_{k^A,j} = 0$ or $x^*_{k^A,k} = 0$, but not both.*

Proof. Let $\{j, k\} \in S$. By definition, then $x^*_{j,k} > 0$, and ($x^*_{k^A,j} = 0$ or $x^*_{k^A,k} = 0$). Not both $x^*_{k^A,j}$ and $x^*_{k^A,k}$ can vanish, however, because then the triangle inequality would dictate that $x^*_{j,k} = 0$. □

Lemma 15. *If $j < k^A$, then $\{j, k^A\} \in S$.*

Proof. We need only show that $x^*_{j,k^A} > 0$ and ($x^*_{k^A,k^A} = 0$ or $x^*_{k^A,j} = 0$). Recall that $z^*_{k^A} > 0$ and $x^*_{j,k^A} \geq z^*_{k^A}$, so $x^*_{j,k^A} > 0$. Also, $x^*_{k^A,k^A}$ is defined to be 0, so we satisfy the "or"-term. □

Lemma 16. *If $i, j < k^A$, then $\{i, j\} \notin S$.*

Proof. To show that $\{i, j\} \notin S$, it is enough to show that $x^*_{i,k^A} \neq 0$ and $x^*_{j,k^A} \neq 0$; we use Lemma 15 to get $\{i, k^A\} \in S$ and $\{j, k^A\} \in S$, from which it follows that $x^*_{i,k^A} > 0$ and $x^*_{j,k^A} > 0$. □

Lemma 17. *If $i \neq j \neq k$, then $\big\{\{i, j\}, \{i, k\}, \{j, k\}\big\} \not\subset S$.*

Proof. Assume the opposite and use Lemma 14 to get ($x^*_{k^A,i} = 0$ xor $x^*_{k^A,j} = 0$) and ($x^*_{k^A,i} = 0$ xor $x^*_{k^A,k} = 0$) and ($x^*_{k^A,j} = 0$ xor $x^*_{k^A,k} = 0$), which is false. □

Lemma 18. *If $i \neq j \neq k$ and $\{j, k\} \in S$, then $\{i, j\} \in S$ or $\{i, k\} \in S$.*

Proof. Let $i \neq j \neq k$ such that $\{j, k\} \in S$, and assume $\{i, j\} \notin S$ and $\{i, k\} \notin S$. This gives

$$\begin{aligned} & x^*_{j,k} > 0 \text{ and } (x^*_{k^A,j} = 0 \text{ or } x^*_{k^A,k} = 0), \\ & x^*_{i,j} = 0 \text{ or } (x^*_{k^A,i} > 0 \text{ and } x^*_{k^A,j} > 0), \text{ and} \\ & x^*_{i,k} = 0 \text{ or } (x^*_{k^A,i} > 0 \text{ and } x^*_{k^A,k} > 0). \end{aligned} \tag{21}$$

Since $x^*_{j,k} > 0$, by triangle inequality we have that $x^*_{i,j} > 0$ or $x^*_{i,k} > 0$. By symmetry, assume $x^*_{i,j} > 0$, which means that ($x^*_{k^A,i} > 0$ and $x^*_{k^A,j} > 0$). Since $x^*_{k^A,j} > 0$, it follows from ($x^*_{k^A,j} = 0$ or $x^*_{k^A,k} = 0$) that $x^*_{k^A,k} = 0$. Using the triangle inequality on $x^*_{k^A,k} = 0$ and $x^*_{k^A,i} > 0$ gives us $x^*_{i,k} > 0$, which we can use to conclude that ($x^*_{k^A,i} > 0$ and $x^*_{k^A,k} > 0$). This contradicts the already derived equality $x^*_{k^A,k} = 0$. □

The final lemma that we need will be used to argue that the subtraction used to define x^A and x^B in (17) and (18) is valid.

Lemma 19. *If $i, j < k$ and $i \neq j$ and $\{i, k\} \in S$ and $\{j, k\} \in S$, then $x^*_{i,k} + x^*_{j,k} - 2\lambda \geq x^*_{i,j}$.*

Proof. $\{i,j,k\}$ satisfies the conditions of the lemma, but $x^*_{i,k}+x^*_{j,k}-2\lambda < x^*_{i,j}$. Take such a triple with minimum possible $m=\max\{i,j\}$. By symmetry, assume $m=j>i$.

First, suppose that $x^*_{k^A,k}\neq 0$. In this case we derive a contradiction, so assume that $x^*_{i,j}+x^*_{i,k}-2\lambda < x^*_{j,k}$. Since $\{i,k\}\in S$ and $\{j,k\}\in S$, from Lemma 14 we know ($x^*_{k^A,i}=0$ xor $x^*_{k^A,k}=0$), and ($x^*_{k^A,j}=0$ xor $x^*_{k^A,k}=0$). Since $x^*_{k^A,k}\neq 0$, it follows that $x^*_{k^A,i}=0$ and $x^*_{k^A,j}=0$ and by the triangle inequality $x^*_{i,j}=0$, which we can plug into our assumption to get $x^*_{i,k}+x^*_{j,k}-2\lambda<0$. But now (since $\{i,k\}\in S$ and $\{j,k\}\in S$,) we have $x^*_{i,k}\geq\lambda$ and $x^*_{j,k}\geq\lambda$, which contradicts the previous inequality.

If $x^*_{k^A,k}=0$, but $x^*_{i,j}=0$, then the logic from the previous paragraph still applies (skipping to the part where we conclude that $x^*_{i,j}=0$), so let us now consider the case where $x^*_{k^A,k}=0$ and $x^*_{i,j}>0$. First, since $\{i,k\}\in S$ and $\{j,k\}\in S$, Lemma 17 gives $\{i,j\}\notin S$, which implies that $j\neq k^A$ (or else Lemma 15 would dictate that $\{i,j\}\in S$). This gives us two further cases:

Case $j<k^A$: Here we start with the LP constraint $x^*_{i,k^A}+x^*_{j,k^A}-2z^*_{k^A}\geq x^*_{i,j}$, and since $z^*_{k^A}\geq\lambda$, we get $x^*_{i,k^A}+x^*_{j,k^A}-2\lambda\geq x^*_{i,j}$. Now if $k=k^A$ we're done (simply substitute k for k^A), so let $k\neq k^A$: first, since $\{j,k\}\in S$, Lemma 14 gives us $x^*_{j,k^A}=0$ xor $x^*_{k,k^A}=0$. We can argue that $x^*_{j,k^A}>0$ (since $\{j,k^A\}\in S$ by Lemma 15), which means $x^*_{k,k^A}=0$. A similar argument applies to $\{i,k\}\in S$. Now we can use the triangle inequality to show that $x^*_{j,k^A}=x^*_{j,k}$ and $x^*_{i,k^A}=x^*_{i,k}$, which we can substitute into our previous inequality to get $x^*_{i,k}+x^*_{j,k}-2\lambda\geq x^*_{i,j}$.

Case $j>k^A$: By the definition of k^A, we have $z^*_j=0$, and since $x^*_{i,j}>0$, we may conclude that $x^*_{i,j}>z^*_j$. Now by Lemma 12 there exists p such that $p<j$, $p\neq i$ and

$$x^*_{i,j}+x^*_{p,j}-2z^*_j=x^*_{i,p}. \tag{22}$$

Since $z^*_j=0$, we actually have $x^*_{i,j}+x^*_{p,j}=x^*_{i,p}$. At this point we have yet another two cases to consider, based on $x^*_{p,k}$.

Case $x^*_{p,k}>0$: We see that $\{p,k\}\in S$, since we already know $x^*_{k^A,k}=0$. This, along with the fact that $i,p<j=m$, and $i,p<k$, and $i\neq p$, allows us to use the inductive hypothesis (the maximality of $m=j$) to get $x^*_{i,k}+x^*_{p,k}-2\lambda\geq x^*_{i,p}$. Substituting for $x^*_{i,p}$ gives $x^*_{i,k}+x^*_{p,k}-2\lambda\geq x^*_{i,j}+x^*_{p,j}$. Now by the triangle inequality we have $x^*_{p,j}+x^*_{j,k}\geq x^*_{p,k}$, which we can use to increase the value of $x^*_{p,k}$ in our previous equation to get $x^*_{i,k}+x^*_{p,j}+x^*_{j,k}-2\lambda\geq x^*_{i,j}+x^*_{p.j}$. This can now be simplified to $x^*_{i,k}+x^*_{j,k}-2\lambda\geq x^*_{i,j}$.

Case $x^*_{p,k}=0$: From the triangle inequalities we have $x^*_{i,k}=x^*_{i,p}$ and $x^*_{j,k}=x^*_{p,j}$. Substituting this into (22) gives $x^*_{i,j}+x^*_{j,k}=x^*_{i,k}$. Subtracting $x^*_{j,k}$ from both sides gives $x^*_{i,k}-x^*_{j,k}=x^*_{i,j}$. Next we'll note that $\{j,k\}\in S$, which means that $x^*_{j,k}\geq\lambda$. We then add $2(x^*_{j,k}-\lambda)$ to the left side of our previous equation to get $x^*_{i,k}+x^*_{j,k}-2\lambda\geq x^*_{i,j}$. □

Proof of Theorem 8. The constraint $\sum z^A_i=1$ is satisfied since only one component of z^A is positive (namely $z^A_{k^A}$), and it equals 1.

The constraint $x^A_{i,k}\geq z^A_k$ is only in question when $z^A_k>0$, that is when $k=k^A$, so let $i<k=k^A$. By Lemma 15, we have $\{i,k\}\in S$, which in turn means $x^A_{i,k}=1$. Thus the constraint ends up requiring $1\geq 1$, which is true.

The final LP constraint is $x^A_{i,k}+x^A_{j,k}-2z^A_k\geq x^A_{i,j}$ for all $i,j<k$, $i\neq j$.

Take such a triple $\{i,j,k\}$. Now if $k=k^A$, then we know $z^A_k=1$. Thus the constraint is equivalent to $x^A_{i,k^A}+x^A_{j,k^A}-2\geq x^A_{i,j}$. Since $\{i,k^A\}$, $\{j,k^A\}\in S$ (by Lemma 15), we know

$x^A_{i,k^A} = 1 = x^A_{j,k^A}$, so the constraint reduces to $1+1-2 \geq x^A_{i,j}$. It thus suffices to show that $x^A_{i,j} = 0$, which we can do because $\{i, k^A\}$, $\{j, k^A\} \in S$, and so (by Lemma 17), $\{i, j\} \notin S$.

Now if $k \neq k^A$, then $z^A_k = 0$ by definition, and the constraint becomes $x^A_{i,k} + x^A_{j,k} \geq x^A_{i,j}$. It is possible that none of $\{i, j\}, \{i, k\}$, and $\{j, k\}$ are in S, in which case the constraint becomes $0 + 0 \geq 0$. Otherwise Lemmas 18 and 17 tell us that exactly two of $\{i, j\}, \{i, k\}$, and $\{j, k\}$ are in S, and the constraint reduces to either $1 + 1 \geq 0$, or $1 + 0 \geq 1$. Both of these are true. $\square$

Proof of Theorem 9. Before looking at the constraints, it is worth mentioning that the implicit nonnegativity constraints are met, because we only subtract λ from variables which we've already proven to be at least λ.

Now, the first LP constraint requires that $\sum z^B_i = 1$. This is true because

$$\sum z^B_i = \frac{1}{1-\lambda}\Big(\sum_i z^*_i - \lambda\Big) = \frac{1}{1-\lambda}(1-\lambda) = 1.$$

The second LP constraint requires $x^B_{i,k} \geq z^B_k$ for $i < k$. Note that we won't bother talking about dividing by $1 - \lambda$, since we do this to both sides when obtaining this constraint from the $\{x^*, z^*\}$ version.

This constraint is not an issue if $k < k^A$, since in this case by Lemma 16 $\{i, k\} \notin S$, so we don't subtract from either side of the solution $\{x^*, z^*\}$. If $k = k^A$, then Lemma 15 gives $\{i, k^A\} \in S$, which means that we subtract from both sides of the $\{x^*, z^*\}$ version of this constraint, which is also fine. Finally, if $k > k^A$, then $z^B_k = z^*_k = 0$ (from the definition of k^A), turning the constraint into $x^B_{i,k} \geq 0$, which is satisfied since we already argued that all the variables are nonnegative.

The third LP constraint requires that $x^B_{i,k} + x^B_{j,k} - 2z^B_k \geq x^B_{i,j}$ for all $i, j < k$ where $i \neq j$. Again, we won't bother talking about dividing by $1 - \lambda$, since we do this to both sides. We'll first consider the case where none of $\{i, j\}, \{i, k\}$, and $\{j, k\}$ are in S. In this case, we know that $k \neq k^A$, or else Lemma 15 would dictate that $\{j, k\} \in S$. This means that our constraint remains the same as the $\{x^*, z^*\}$ version. Now, if at least one of $\{i, j\}, \{i, k\}$, and $\{j, k\}$ is in S, then Lemmas 18 and 17 tell us that exactly 2 of them are. This gives us three cases:

Case 1: $\{i, k\} \in S$ and $\{j, k\} \in S$. In this case we subtract 2λ from the left hand side of the $\{x^*, z^*\}$ version of this constraint. Now we know $k \geq k^A$ by Lemma 16, so if $k = k^A$, then we also end up adding 2λ to the left hand side of the constraint (in obtaining the $-2z^B_k$). If $k > k^A$, then the definition of k^A tells us that $z^B_k = 0$, and then Lemma 19 ensures that we can subtract 2λ from the left hand side of the $\{x^*, z^*\}$ version, and still meet the constraint.

Cases 2 and 3: $\{i, k\} \in S$ and $\{i, j\} \in S$ or $\{j, k\} \in S$ and $\{i, j\} \in S$. In both these cases $\{i, j\} \in S$, which means that $k \neq k^A$. both $\{i, j\} \in S$ and $\{i, k\} \in S$, if $\{i, j\} \in S$ by Lemma 17. Thus $z^B_k = z^*_k$, and we end up simply subtracting λ from both sides of the $\{x^*, z^*\}$ version of the constraint, which is fine. $\square$

Proof of Theorem 10. First, it is not difficult to see that $\lambda(x^A, z^A) + (1 - \lambda)(x^B, z^B) = (x^*, z^*)$. For example, for each $\{i, j\}$ it follows from the definitions that $\lambda x^A_{i,j} + (1 - \lambda)x^B_{i,j} = x^*_{i,j}$. Indeed, if $\{i, j\} \in S$ we have $\lambda + (x^*_{i,j} - \lambda) = x^*_{i,j}$, and if $\{i, j\} \notin S$ we have $0 + x^*_{i,j} = x^*_{i,j}$. (A similar argument gives $\lambda z^A + (1 - \lambda)z^B = z^*$.)

In Theorems 8 and 9 we show that (x^A, z^A) and (x^B, z^B) are both valid solutions to the LP. Hence, the original solution (x^*, z^*) is a convex combination of an integral feasible solution (x^A, z^A) and another feasible solution. Since (x^*, z^*) is optimal so are both of the solutions that form the convex combination, and therefore the integral solution (x^A, z^A) is optimal for the linear program, which completes the proof. □

5 ℓ_1 embeddability

Every cut in a graph defines a (semi)metric d, where $d(v,w) = 1$ if v and w are on the opposite shores of the cut, and $d(v,w) = 0$ otherwise. Such *cut metrics* are closely related to ℓ_1 metrics, because every ℓ_1 metric on an n-element set V can be written as a nonnegative combination of cut metrics on V ([15], p. 40). Given a rational metric d, it is NP-complete to determine whether d is ℓ_1-embeddable [7]. The decomposition of d as $\sum_i \alpha_i \delta(S_i)$, where $\alpha_i > 0$ and $S_i \subseteq V$, is clearly a certificate for the ℓ_1-embeddability of d. The certificate consists of the vector α and the list of sets S_i. By Carathéodory's theorem, the number of cut metrics used need not be more than $\binom{n}{2}$, and so the certificate consists of $O(n^2)$ rational numbers and $O(n^2)$ n-bit vectors (to represent the sets).

Our linear programming formulation gives a more compact certificate. Take a decomposition of d into cut metrics as above and let $\lambda = \sum_i \alpha_i$. Then $d' = d/\lambda = \sum_i (\alpha_i/\lambda)\delta(S_i)$ is a convex combination of cut metrics. Let z be an $n-1$-vector and $d'_{v,w} = d(v,w)/\lambda$. If (d', z) is feasible for our LP, then we can use the procedure of Theorem 8 to extract cut metrics that define d' and thus verify that d' is a convex combination of cut metrics. The $(n-1)$-vector z thus together with λ forms a certificate for ℓ_1-decomposability of d.

The standard linear programming formulation [13, 28] also provides a certificate, but it consists of n rational vectors of length n each, that is, a total of n^2 values.

6 Concluding remarks and open questions

The $O(|V|^3) \times O(|V|^2)$ minimum cut formulation we present can be viewed (in a dual) sense as selecting a collection of splitting-off operations. Can we generalize this to other problems, for instance spanning tree, by forging a connection between an algorithmic procedure and a formulation? Finally, can the relaxation we obtain for k-edge-connected spanning subgraph be extended to generalized steiner network problems? The fact that splitting-off can be performed while preserving pairwise connectivity requirements indicates that this might be possible.

7 Acknowledgments

The four latter authors thank the first author for the insights outlined in Section 2. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

References

[1] M. Alekhnovich, S. Arora, and I. Tourlakis. Towards strong nonapproximability results in the lovasz-schrijver hierarchy. In *Proceedings of the 37th annual ACM symposium on Theory of computing*, pages 294–303, 2005.

[2] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. On the solution of traveling salesman problems. *Documenta Mathematica*, Extra Volume ICM 1998(3):645–656, 1998.

[3] S. Arora, B. Bollobás, and L. Lovász. Proving integrality gaps without knowing the linear program. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 313–322, 2002.

[4] T. S. Arthanari. On the traveling salesman problem. Presented at the XI Symposium on Mathematical Programming held at Bonn, West Germany, 1982.

[5] T. S. Arthanari and M. Usha. An alternate formulation of the symmetric traveling salesman problem and its properties. *Discrete Appl. Math.*, 98(3):173–190, 2000.

[6] T. S. Arthanari and M. Usha. On the equivalence of the multistage-insertion and cycle-shrink formulations of the symmetric traveling salesman problem. *Operations Res. Letters*, 29(3):129–139, 2001.

[7] D. Avis and M. Deza. The cut cone, l^1 embeddability, complexity and multicommodity flows. *Networks*, 21:595–617, 1991.

[8] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998.

[9] R. D. Carr. *Polynomial separation procedures and facet determination for inequalities of the traveling salesman polytope*. PhD thesis, Carnegie Mellon University, 1995.

[10] R. D. Carr. Separating over classes of TSP inequalities defined by 0 node-lifting in polynomial time. In *Proceedings of the 5th annual Integer Programming and Combinatorial Optimization conference*, pages 460–474, 1996.

[11] R. D. Carr and G. Lancia. Compact vs. exponential-size LP relaxations. *Operations Res. Letters*, 30:57–65, 2002.

[12] R. D. Carr and S. Vempala. Randomized metarounding. *Random Struct. Algorithms*, 20(3):343–352, 2002.

[13] M. Conforti, G. Rinaldi, and L. Wolsey. On the cut polyhedron. Technical Report 5, CORE, Université catholique de Louvain, 2000.

[14] G. Dantzig, D. Fulkerson, and S. Johnson. Solution of a large-scale traveling salesman problem. *J. of the Operations Res. Soc. of America*, pages 393–410, 1954.

[15] M. Deza and M. Laurent. *Geometry of cuts and metrics*. Springer-Verlag, 1997.

[16] A. Frank. On a theorem of mader. *Discrete Math.*, 101(1-3):49–57, 1992.

[17] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995.

[18] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, Berlin, 1988.

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

[20] D. R. Karger. Minimum cuts in near-linear time. *Journal of the ACM*, 47:46–76, 2000.

[21] S. Khot and N. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into ℓ_1. In *Proceedings of the 46th Symposium on Foundations of Computer Science*, 2005.

[22] L. Lovász. *Combinatorial Problems and Exercises*. North-Holland, Amsterdam, 1979.

[23] T. L. Magnanti and L. A. Wolsey. Optimal trees. In *Network models*, volume 7 of *Handbooks in Operations Research*, pages 503–615. Elsevier, 1995.

[24] R. K. Martin. Using separation algorithms to generate mixed integer model reformulations. *Operations Res. Letters*, 10:119–128, 1991.

[25] H. Nagamochi and T. Ibaraki. Computing edge-connectivity in multigraphs and capacitated graphs. *SIJDM: SIAM Journal on Discrete Mathematics*, 5:54–66, 1992.

[26] A. Schrijver. *Combinatorial optimization : Polyhedra and Efficiency.* Springer, 2003. Schrijver.

[27] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of the ACM*, 44(4):585–591, 1997.

[28] A. Tamir. Polynomial formulations of min-cut problems. Unpublished manuscript, 1994.

[29] V. V. Vazirani. *Approximation Algorithms.* Springer, 2001.

[30] M. Yannakakis. Expressing combinatorial optimization problems by linear programs. *J. Comput. Syst. Sci.*, 43(3):441–466, 1991.