Prize-Collecting Steiner Network Problems

2

MOHAMMADTAGHI HAJIAGHAYI, AT&T Labs – Research and University of Maryland
ROHIT KHANDEKAR, IBM T. J. Watson Research Center
GUY KORTSARZ, Rutgers University, Camden
ZEEV NUTOV, The Open University of Israel

In the Steiner Network problem, we are given a graph G with edge-costs and connectivity requirements r_{uv} between node pairs u, v. The goal is to find a minimum-cost subgraph H of G that contains r_{uv} edge-disjoint paths for all $u, v \in V$. In Prize-Collecting Steiner Network problems, we do not need to satisfy all requirements, but are given a *penalty function* for violating the connectivity requirements, and the goal is to find a subgraph H that minimizes the cost plus the penalty. The case when $r_{uv} \in \{0, 1\}$ is the classic Prize-Collecting Steiner Forest problem.

In this article, we present a novel linear programming relaxation for the Prize-Collecting Steiner Network problem, and by rounding it, obtain the first constant-factor approximation algorithm for submodular and monotone nondecreasing penalty functions. In particular, our setting includes all-or-nothing penalty functions, which charge the penalty even if the connectivity requirement is slightly violated; this resolves an open question posed by Nagarajan et al. [2008]. We further generalize our results for element-connectivity and node-connectivity.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Steiner network problem, prize-collecting, edge-connectivity, vertex-connectivity, all-or-nothing flows

ACM Reference Format:
Hajiaghayi, M., Khandekar, R., Kortsarz, G., and Nutov, Z. 2012. Prize-collecting steiner network problems. ACM Trans. Algor. 9, 1, Article 2 (December 2012), 13 pages.
DOI = 10.1145/2390176.2390178 http://doi.acm.org/10.1145/2390176.2390178

1. INTRODUCTION

Prize-collecting Steiner problems are well-known network design problems with several applications in expanding telecommunications networks (see, e.g., Johnson et al. [2000] and Salman et al. [2000]), cost sharing, and Lagrangian relaxation techniques

---

Part of this work was done while the authors were meeting at DIMACS. We would like to thank DIMACS for their hospitality.
A version of this article appeared in *Proceedings of the 14th Conference on Integer Programming and Combinatorial Optimization,* 71–84.
This work was supported by the National Science Foundation, under grant 0829959.
Authors' addresses: M. Hajiaghayi, AT&T Labs – Research & Computer Science Department, University of Maryland at College Park, MD 20742; email: hajiagha@cs.umd.edu; R. Khandekar, IBM T. J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598; email: rohitk@us.ibm.com; G. Kortsarz, Computer Science Department, Rutgers University, 303 Cooper Street, Camden, NJ 08102; email: guyk@camden.rutgers.edu; Z. Nutov, Department of Computer Science, The Open University of Israel, Raanana, Israel; email: nutov@openu.ac.il.

DOI 10.1145/2390176.2390178 http://doi.acm.org/10.1145/2390176.2390178

(see, e.g., Jain and Vazirani [2001] and Chudak et al. [2001]). A general form of these problems is the Prize-Collecting Steiner Forest problem[1]: given a network (graph) $G = (V, E)$, a set of source-sink pairs $\mathcal{P} = \{\{s_1, t_1\}, \{s_2, t_2\}, \ldots, \{s_k, t_k\}\}$, a non-negative cost function $c : E \to \Re_+$, and a non-negative penalty function $\pi : \mathcal{P} \to \Re_+$, our goal is a minimum-cost way of installing (buying) a set of links (edges) and paying the penalty for those pairs which are not connected via the installed links. When all penalties are ∞, the problem is the classic APX-hard Steiner Forest problem, for which the best known approximation ratio is $2 - \frac{2}{n}$ (n is the number of nodes of the graph) due to Agrawal et al. [1995] (see also Goemans and Williamson [1995] for a more general result and a simpler analysis). The case of Prize-Collecting Steiner Forest problem when all sinks are identical is the classic Prize-Collecting Steiner Tree problem. Bienstock et al. [1993] were the first to consider this problem (based on a problem earlier proposed by Balas [1989]) and gave a 3-approximation algorithm for it. The current best ratio for this problem is 1.992 by Archer et al. [2009], improving upon a primal-dual $(2-(1/(n-1)))$-approximation algorithm of Goemans and Williamson [1995]. When in addition all penalties are ∞, the problem is the classic Steiner Tree problem, which is known to be APX-hard [Bern and Plassmann 1989] and for which the best approximation ratio is 1.55 [Robins and Zelikovsky 2005]. Recently, Byrka et al. [2010] have announced an improved approximation algorithm for the Steiner tree problem.

The general form of the Prize-Collecting Steiner Forest problem has been first formulated by Hajiaghayi and Jain [2006]. They showed how by a primal-dual algorithm to a novel integer programming formulation of the problem with doubly exponential variables, we can obtain a 3-approximation algorithm for the problem. In addition, they show that the factor 3 in the analysis of their algorithm is tight. However, they show how a direct randomized LP-rounding algorithm with approximation factor 2.54 can be obtained for this problem. Their approach has been generalized by Sharma et al. [2007] for network design problems where violating arbitrary 0-1 connectivity constraints are allowed in exchange for a general penalty function. The work of Hajiaghayi and Jain has also motivated a game-theoretic version of the problem considered by Gupta et al. [2007].

In this article, we consider a more general, high-connectivity version of Prize-Collecting Steiner Forest, called Prize-Collecting Steiner Network, in which we are also given connectivity requirements r_{uv} for pairs of nodes u and v and a penalty function in case we do not satisfy all r_{uv}. Our goal is to find a minimum-cost way of constructing a network (graph) in which, for each pair u and v, either we connect it with at least r_{uv} edge-disjoint paths or pay a penalty for violating its connectivity requirement. This problem can arise in real-world network design, in which a typical client not only might want to connect to the network but also might want to connect via a few disjoint paths (e.g., to have a higher bandwidth or redundant connections in case of edge failures) and a penalty might be charged if we cannot satisfy its connectivity requirement. When all penalties are ∞, the problem is the classic Steiner Network problem. Improving on a long line of earlier research that applied primal-dual methods, Jain [2001] obtained a 2-approximation algorithm for Steiner Network using the iterative rounding method. This algorithm was generalized to so-called "element-connectivity" problem by Fleischer et al. [2001] and by Cheriyan et al. [2006]. Recently, some results were obtained for the node-connectivity version; the currently best-known ratios for the node-connectivity case are $O(R^3 \log n)$ for general requirements [Chuzhoy and Khanna 2009] and $O(R \log R)$ for rooted requirements [Nutov 2010], where $R = \max_{u,v \in V} r_{uv}$

[1]In the literature, this problem is also called "prize-collecting generalized Steiner tree".

is the maximum requirement. See also the survey by Kortsarz and Nutov [2007] for various min-cost connectivity problems.

Hajiahayi and Nasri [2010] generalize the iterative rounding approach of Jain to Prize-Collecting Steiner Network. They consider a version where for each pair u, v with edge-connectivity r in the solution, one has to pay a penalty of $p(r)$ where p is a nonincreasing penalty function with an additional property that its marginal differences are nonincreasing, that is, $p(r+1) - p(r+2) \leq p(r) - p(r+1)$ for all $r \geq 0$. They obtain an iterative rounding 3-approximation algorithm for this case. For the special case when penalty functions are linear in the violation of the connectivity requirements, Nagarajan et al. [2008] using Jain's iterative rounding algorithm as a black box give a 2.54-approximation algorithm. They also generalize the 0-1 requirements of Prize-Collecting Steiner Forest problem introduced by Sharma et al. [2007] to include general connectivity requirements. Working with a slightly general form of the submodular and monotone non-decreasing penalty function of Sharma et al., they present an $O(\log R)$-approximation algorithm (recall that R is the maximum connectivity requirement). In this algorithm, they assume that we can use each edge possibly many times (without bound). They raise the question whether one can obtain a constant approximation ratio without all these assumptions, when penalty is a submodular multi-set function of the set of disconnected pairs? More importantly they pose as an open problem designing a good approximation algorithm for the all-or-nothing version of penalty functions, that is, penalty functions which charge the penalty even if the connectivity requirement is slightly violated. In this article, we answer affirmatively all these open problems by presenting the first constant-factor 2.54-approximation algorithm that is based on a novel LP formulation of the problem. We further generalize our results for element-connectivity and node-connectivity. In fact, for both types of connectivities, we prove a very general result (see Theorem 1.1) stating that if Steiner Network (the version without penalties) admits an LP-based ρ-approximation algorithm, then the corresponding prize-collecting version admits a $(\rho + 1)$-approximation algorithm.

1.1. Problems We Consider

In this section, we formally define the terms used in the article. For a subset S of nodes in a graph H, let $\lambda_H^S(u, v)$ denote the *S-connectivity* between u and v in H, namely, the maximum number of edge-disjoint uv-paths in H so that no two of them have a node in $S - \{u, v\}$ in common. In the Generalized Steiner-Network (GSN) problem, we are given a graph $G = (V, E)$ with edge-costs $c_e \geq 0$, a node subset $S \subseteq V$, a collection $\{u_1, v_1\}, \ldots, \{u_k, v_k\}$ of node pairs from V, and S-connectivity requirements $r_1, \ldots, r_k$. The goal is to find a minimum cost subgraph H of G so that $\lambda_H^S(u_i, v_i) \geq r_i$ for all i. Extensively studied particular cases of GSN are the Steiner Network problem, also called Edge-Connectivity GSN ($S = \emptyset$), Node-Connectivity GSN ($S = V$), and Element-Connectivity GSN ($S \cap \{u_i, v_i\} = \emptyset$ for all i). The case of *rooted requirements* is when there is a "root" s that belongs to all pairs $\{u_i, v_i\}$. We consider the following "prize-collecting" version of GSN.

All-or-Nothing Prize Collecting Generalized Steiner Network (PC-GSN):

Instance: A graph $G = (V, E)$ with edge-costs $c_e \geq 0$, $S \subseteq V$, a collection $\{u_1, v_1\}, \ldots, \{u_k, v_k\}$ of node pairs from V, S-connectivity requirements $r_1, \ldots, r_k > 0$, and a penalty function $\pi : 2^{\{1,\ldots,k\}} \rightarrow \Re_+$.

Objective: Find a subgraph H of G that minimizes the *value*

$$\mathtt{val}(H) = c(H) + \pi(\mathtt{unsat}(H))$$

of H, where $\mathtt{unsat}(H) = \{i \mid \lambda_H^S(u_i, v_i) < r_i\}$ is the set of requirements *not* (completely) satisfied by H.

We will assume that the penalty function π is given by a value oracle. We will also assume that π is *submodular*, namely, that π satisfies the decreasing marginal value property, $\pi(A) + \pi(B) \geq \pi(A \cap B) + \pi(A \cup B)$ for all A, B and that it is *monotone nondecreasing*, namely, $\pi(A) \leq \pi(B)$ for all A, B with $A \subseteq B$. As was mentioned, approximating the edge-connectivity variant of PC-GSN was posed as the main open problem by Nagarajan et al. [2008].

We next define the second problem we consider. We will in fact reduce this problem to the first one.

Generalized Steiner Network with Generalized Penalties (GSN-GP):

Instance: A graph $G = (V, E)$ with edge-costs $c_e \geq 0$, $S \subseteq V$, a collection $\{u_1, v_1\}, \ldots, \{u_k, v_k\}$ of node pairs from V, and non-increasing penalty functions $p_1, \ldots, p_k : \{0, 1, \ldots, n-1\} \to \Re_+$.

Objective: Find a subgraph H of G that minimizes the *value*

$$\texttt{val}'(H) = c(H) + \sum_{i=1}^{k} p_i\left(\lambda_H^S(u_i, v_i)\right).$$

This problem captures general penalty functions of the S-connectivity $\lambda^S(u_i, v_i)$ for given pairs $\{u_i, v_i\}$. It is natural to assume that the penalty functions are non-increasing, that is, we pay less in the objective function if the achieved connectivity is more. This problem was posed as an open question by Nagarajan et al. [2008]. In this article, we use the convention that $p_i(n) = 0$ for all i.

We need some definitions to introduce our results. A pair $T = \{T', T''\}$ of subsets of V is called a *setpair* (of V) if $T' \cap T'' = \emptyset$. Let $K = \{1, \ldots, k\}$. Let $T = \{T', T''\}$ be a setpair of V. We denote by $\delta(T)$ the set of edges in E between T' and T''. For $i \in K$ we use $T \odot (i, S)$ to denote that $|T' \cap \{u_i, v_i\}| = 1$, $|T'' \cap \{u_i, v_i\}| = 1$ and $V \setminus (T' \cup T'') \subseteq S$. While in the case of edge-connectivity a "cut" consists of edges only, in the case of S-connectivity a cut that separates between u and v is "mixed," meaning it may contain both edges in the graph and nodes from S. Note that if $T \odot (i, S)$, then $\delta(T) \cup (V \setminus (T' \cup T''))$ is such a mixed cut that separates between u_i and v_i. Intuitively, Menger's Theorem for S-connectivity (see Kortsarz and Nutov [2007]) states that the S-connectivity between u_i and v_i equals the minimum size of such a mixed cut. Formally, for a node pair u_i, v_i of a graph $H = (V, E)$ and $S \subseteq V$, we have

$$\lambda_H^S(u_i, v_i) = \min_{T \odot (i,S)} (|\delta(T)| + |V \setminus (T' \cup T'')|) = \min_{T \odot (i,S)} (|\delta(T)| + |V| - (|T'| + |T''|)).$$

Hence, if $\lambda_H^S(u_i, v_i) \geq r_i$ for a graph $H = (V, E)$, then, for any setpair T with $T \odot (i, S)$, we must have $|\delta(T)| \geq r_i(T)$, where $r_i(T) = \max\{r_i + |T'| + |T''| - |V|, 0\}$. Consequently, a standard "cut-type" LP-relaxation of the GSN problem is as follows (see Kortsarz and Nutov [2007]):

$$\min \left\{ \sum_{e \in E} c_e x_e \mid \sum_{e \in \delta(T)} x_e \geq r_i(T)\ \forall T \odot (i, S),\ \forall i \in K;\ x_e \in [0, 1]\ \forall e \right\}. \tag{1}$$

1.2. Our Results

We introduce a novel LP relaxation of the problem that is shown to be better, in terms of the integrality gap, than a "natural" LP relaxation considered in Nagarajan et al. [2008]. Using our LP relaxation, we prove the following main result.

THEOREM 1.1. *Suppose that for an instance defined by any subset $K' \subseteq K$ of node pairs, there exists a polynomial-time algorithm that computes an integral solution to LP (1) of cost at most ρ times the fractional optimal value of LP (1) for K'. Then* PC-GSN, *defined on all pairs K, admits a $(1 - e^{-1/\rho})^{-1}$-approximation algorithm, provided that the penalty function π is submodular and monotone nondecreasing.*

Note that since $1 - (1/\rho) < e^{-(1/\rho)} < 1 - (1/(\rho + 1))$ holds for $\rho \geq 1$, we have $\rho < (1 - e^{-1/\rho})^{-1} < \rho + 1$.

Let $R = \max_i r_i$ denote the maximum requirement. The best known values of ρ are as follows: 2 for Edge-GSN [Jain 2001], 2 for Element-GSN [Fleischer et al. 2001; Cheriyan et al. 2006], $O(R^3 \log |V|)$ for Node-GSN [Chuzhoy and Khanna 2009], and $O(R \log R)$ for Node-GSN with rooted requirements [Nutov 2010]. Substituting these values in Theorem 1.1, we obtain:

COROLLARY 1.2. PC-GSN *problems admit the following approximation ratios provided that the penalty function π is submodular and monotone nondecreasing:* 2.54 *for edge- and element-connectivity, $O(R^3 \log |V|)$ for node-connectivity, and $O(R \log R)$ for node-connectivity with rooted requirements.*

Our results for GSN-GP follow from Corollary 1.2.

COROLLARY 1.3. GSN-GP *problems admit the following approximation ratios:* 2.54 *for edge- and element-connectivity, $O(R^3 \log |V|)$ for node-connectivity, and $O(R \log R)$ for node-connectivity with rooted requirements. Here $R = max_{1 \leq i \leq k}\ min\{\lambda \geq 0 \mid p_i(\lambda) = 0\}$.*

PROOF. We present an approximation ratio preserving reduction from the GSN-GP problem to the corresponding PC-GSN problem. Given an instance of the GSN-GP problem, we create an instance of the PC-GSN problem as follows. The PC-GSN instance inherits the graph G, its edge-costs, and the set S. Let (u_i, v_i) be a pair in GSN-GP and let $R_i = \min\{\lambda \geq 0 \mid p_i(\lambda) = 0\}$. We introduce R_i copies of this pair, $\{(u_i^1, v_i^1), \ldots, (u_i^{R_i}, v_i^{R_i})\}$, to the set of pairs in the PC-GSN instance. We set the edge-connectivity requirement of a pair (u_i^t, v_i^t) to be t for $1 \leq t \leq R_i$. We also set the penalty function for singleton sets as follows $\pi(\{(u_i^t, v_i^t)\}) = p_i(t-1) - p_i(t)$ for all $1 \leq t \leq R_i$. Finally, we extend this function π to a set of pairs P by *linearity*, that is, $\pi(P) = \sum_{p \in P} \pi(\{p\})$. Note that such a function π is clearly submodular and monotone non-decreasing.

It is sufficient to show that for any subgraph H of G, its value in the GSN-GP instance equals its value in the PC-GSN instance, that is, $\texttt{val}(H) = \texttt{val}'(H)$; then we can use the algorithm from Corollary 1.2 to complete the proof. Fix a pair (u_i, v_i) in the GSN-GP instance. Let $\lambda_H^S(u_i, v_i) = t_i$. Thus, the contribution of pair (u_i, v_i) to the objective function $\texttt{val}(H)$ of the GSN-GP instance is $p_i(t_i)$. On the other hand, since π is linear, the total contribution of pairs $\{(u_i^1, v_i^1), \ldots, (u_i^{R_i}, v_i^{R_i})\}$ to the objective function $\texttt{val}'(H)$ of the PC-GSN instance is $\sum_{t=t_i+1}^{R_i} \pi(\{(u_i^t, v_i^t)\}) = \sum_{t=t_i+1}^{R_i} (p_i(t-1) - p_i(t)) = p_i(t_i)$. Note that the pairs (u_i^t, v_i^t) for $1 \leq t \leq t_i$ do not incur any penalty. Summing up over all pairs, we conclude that $\texttt{val}(H) = \texttt{val}'(H)$, as claimed. □

2. A NEW LP RELAXATION

We use the following LP-relaxation for the PC-GSN problem. We introduce variables $x_e \in [0, 1]$ for $e \in E$, $f_{i,e} \in [0, 1]$ for $i \in K$ and $e \in E$, and $z_I \in [0, 1]$ for $I \subseteq K$. In the intended integral solution H, these variables are supposed to take the following values: $x_e = 1$ if $e \in H$, $f_{i,e} = 1$ if $i \notin \texttt{unsat}(H)$ and e appears on a chosen set of r_i

S-disjoint $\{u_i, v_i\}$-paths in H and $z_I = 1$ if $I = \mathtt{unsat}(H)$.

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{e\in E} c_e x_e + \sum_{I\subseteq K} \pi(I) z_I \\
\text{Subject to} \quad & \sum_{e\in\delta(T)} f_{i,e} \ge \left(1 - \sum_{I\ni i} z_I\right) r_i(T) \quad \forall i \ \ \forall T \odot (i, S) \\
& f_{i,e} \le 1 - \sum_{I\ni i} z_I \quad \forall i \ \ \forall e \\
& x_e \ge f_{i,e} \quad \forall i \ \ \forall e \\
& \sum_{I\subseteq K} z_I = 1 \\
& x_e, f_{i,e}, z_I \in [0, 1] \quad \forall i \ \ \forall e \ \ \forall I
\end{aligned} \tag{2}
$$

We first prove that (2) is a valid LP-relaxation of the PC-GSN problem.

LEMMA 2.1. *The optimal value of LP (2) is at most the value of any optimal solution to the* PC-GSN *problem. Moreover, if π is monotone nondecreasing, the optimum solution value to the* PC-GSN *problem is at most the value of the optimum integral solution of LP (2).*

PROOF. Given a feasible solution H to the PC-GSN problem define a feasible solution to LP (2) as follows. Let $x_e = 1$ if $e \in H$ and $x_e = 0$ otherwise. Let $z_I = 1$ if $I = \mathtt{unsat}(H)$ and $z_I = 0$ otherwise. For each $i \in \mathtt{unsat}(H)$ set $f_{i,e} = 0$ for all $e \in E$, while for $i \notin \mathtt{unsat}(H)$ the variables $f_{i,e}$ take values as follows: fix a set of r_i pairwise S-disjoint $\{u_i, v_i\}$-paths in H, and let $f_{i,e} = 1$ if e belongs to one of these paths and $f_{i,e} = 0$ otherwise. The defined solution is feasible for LP (2): the first set of constraints are satisfied by Menger's Theorem for S-connectivity, while the remaining constraints are satisfied by this definition of variables. It is also easy to see that this solution has value exactly $\mathtt{val}(H)$.

If π is monotone nondecreasing, we prove that for any *integral* solution $\{x_e, f_{i,e}, z_I\}$ to (2), the graph H with edge-set $\{e \in E \mid x_e = 1\}$ has $\mathtt{val}(H)$ at most the value of the solution $\{x_e, f_{i,e}, z_I\}$. To see this, first note that there is a unique set $I \subseteq K$ with $z_I = 1$ since the variables z_I are integral and $\sum_{I\subseteq K} z_I = 1$. Now consider an index $i \notin I$. Since $\sum_{I\ni i} z_I = 0$, we have $\sum_{e\in\delta(T)} x_e \ge \sum_{e\in\delta(T)} f_{i,e} \ge r_i(T)$ for all $T \odot (i, S)$. This implies that $i \notin \mathtt{unsat}(H)$, by Menger's Theorem for S-connectivity. Consequently, $\mathtt{unsat}(H) \subseteq I$, hence $\pi(\mathtt{unsat}(H)) \le \pi(I)$ by the monotonicity of π. Thus, $\mathtt{val}(H) = c(H) + \pi(\mathtt{unsat}(H)) \le \sum_{e\in E} c_e x_e + \sum_{I\subseteq K} \pi(I) z_I$ and the lemma follows. □

2.1. Why Does a "Natural" LP Relaxation not Work?

One may be tempted to consider a natural LP without using the flow variables $f_{i,e}$, namely, the LP obtained from LP (2) by replacing the the first three sets of constraints by the set of constraints

$$
\sum_{e\in\delta(T)} x_e \ge \left(1 - \sum_{I\ni i} z_I\right) r_i(T)
$$

for all i and $T \odot (i, S)$. Here is an example demonstrating that the integrality gap of this LP can be as large as $R = \max_i r_i$ even for edge-connectivity. Let G consist of $R - 1$ edge-disjoint paths between two nodes s and t. All the edges have cost 0. There is only one pair $\{u_1, v_1\} = \{s, t\}$ that has requirement $r_1 = R$ and penalty $\pi(\{1\}) = 1$. Let $\pi(\emptyset) = 0$. Clearly, π is submodular and monotone nondecreasing. We have $S = \emptyset$. No integral solution can satisfy the requirement r_1, hence an optimal integral solution pays the penalty $\pi(\{1\})$ and has value 1. A feasible fractional solution (without the flow variables) sets $x_e = 1$ for all e, and sets $z_{\{1\}} = 1/R$, $z_\emptyset = 1 - 1/R$. The new set of constraints is satisfied since $\sum_{e \in \delta(T)} x_e \geq (1 - 1/R) \cdot R = (1 - z_{\{1\}}) r_1(T)$ for any $\{s, t\}$-cut T. Thus, the optimal LP-value is at most $1/R$, giving a gap of at least R.

With flow variables, however, we have an upper bound $f_{1,e} \leq 1 - z_{\{1\}}$. Since there is an $\{s, t\}$-cut T with $|\delta(T)| = R - 1$, we cannot satisfy the constraints $\sum_{e \in \delta(T)} f_{1,e} \geq (1 - z_{\{1\}}) r_1(T)$ and $f_{1,e} \leq 1 - z_{\{1\}}$ simultaneously unless we set $z_{\{1\}} = 1$. Thus, in this case, our LP (2) with flow variables has the same value as that of the integral optimum.

2.2. Some Technical Results Regarding LP (2)

We will prove the following two statements that together imply Theorem 1.1.

LEMMA 2.2. *Any basic feasible solution to (2) has a polynomial number of non-zero variables. Furthermore, an optimal basic solution to (2) (the non-zero entries) can be computed in polynomial time.*

LEMMA 2.3. *Suppose that for an instance defined by any subset $K' \subseteq K$ of node pairs, there exists a polynomial-time algorithm that computes an integral solution to LP (1) of cost at most ρ times the optimal value of LP (1) for K'. Assume also that π is submodular and monotone nondecreasing. Then, there exists a polynomial-time algorithm that given a feasible solution $F^* = \{x^*, f^*, z^*\}$ to (2), computes as a solution to* PC-GSN, *a subgraph H of G so that $\texttt{val}(H) = c(H) + \pi(\texttt{unsat}(H))$ is at most $(1 - e^{-1/\rho})^{-1}$ times the value of F^*.*

Before proving these lemmas, we prove some useful results. The following statement can be deduced from a theorem of Edmonds for polymatroids (see Korte and Vygen [2002, Chapter 14.2]), as the dual LP $\mathrm{D}(\gamma)$ in the lemma seeks to optimize a linear function over a polymatroid. We provide a direct proof for completeness of exposition.

LEMMA 2.4. *Let $\gamma \in [0, 1]^k$ be a vector. Let π be a monotone nondecreasing and submodular function such that $\pi(\emptyset) = 0$. Consider a primal LP*

$$\mathrm{P}(\gamma) := \min \left\{ \sum_{I \subseteq K} \pi(I) z_I \mid \sum_{I \ni i} z_I \geq \gamma_i \ \forall i \in K, z_I \geq 0 \ \forall I \subseteq K \right\}$$

and its dual LP

$$\mathrm{D}(\gamma) := \max \left\{ \sum_{i \in K} \gamma_i y_i \mid \sum_{i \in I} y_i \leq \pi(I) \ \forall I \subseteq K, y_i \geq 0 \ \forall i \in K \right\}.$$

Let σ be a permutation of K such that $\gamma_{\sigma(1)} \leq \gamma_{\sigma(2)} \leq \cdots \leq \gamma_{\sigma(k)}$. Let us also use the notation that $\gamma_{\sigma(0)} = 0$. The optimum solutions to $\mathrm{P}(\gamma)$ and $\mathrm{D}(\gamma)$, respectively, are given by

$$z_I = \begin{cases} \gamma_{\sigma(i)} - \gamma_{\sigma(i-1)}, & \text{for } I = \{\sigma(i), \ldots, \sigma(k)\}, i \in K, \\ 0, & \text{otherwise}; \end{cases}$$

and

$$y_{\sigma(i)} = \pi(\{\sigma(i), \ldots, \sigma(k)\}) - \pi(\{\sigma(i+1), \ldots, \sigma(k)\}), \text{ for } i \in K.$$

PROOF. To simplify the notation, we assume without loss of generality that $\gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_k$, that is, that σ is the identity permutation. We also use the notation $\gamma_0 = 0$.

We argue that $\{z_I\}$ and $\{y_i\}$ form feasible solutions to the primal and dual LPs, respectively. Note that $z_I \geq 0$ for all I and $\sum_{I \ni i} z_I = \sum_{j=1}^{i} (\gamma_j - \gamma_{j-1}) = \gamma_i$ for all i. Since

π is monotone nondecreasing, the previously defined y_i satisfy $y_i \geq 0$ for all i. Now fix $I \subseteq K$. Let $I = \{i_1, \ldots, i_p\}$ where $i_1 < \cdots < i_p$. Therefore,

$$\begin{aligned}
\sum_{i \in I} y_i = \sum_{j=1}^{p} y_{i_j} &= \sum_{j=1}^{p} [\pi(\{i_j, i_j + 1, \ldots, k\}) - \pi(\{i_j + 1, \ldots, k\})] \\
&= \sum_{j=1}^{p} [\pi(\{i_j\} \cup \{i_j + 1, \ldots, k\}) - \pi(\{i_j + 1, \ldots, k\})] \\
&\leq \sum_{j=1}^{p} [\pi(\{i_j\} \cup \{i_{j+1}, \ldots, i_p\}) - \pi(\{i_{j+1}, \ldots, i_p\})] \\
&= \sum_{j=1}^{p} [\pi(\{i_j, i_{j+1}, \ldots, i_p\}) - \pi(\{i_{j+1}, \ldots, i_p\})] \\
&= \pi(\{i_1, \ldots, i_p\}) = \pi(I).
\end{aligned}$$

This inequality holds because of the submodularity of π. Next, observe that the solutions $\{z_I\}$ and $\{y_i\}$ satisfy

$$\begin{aligned}
\sum_{I} \pi(I) z_I &= \sum_{i=1}^{k} \pi(\{i, \ldots, k\}) \cdot (\gamma_i - \gamma_{i-1}) \\
&= \sum_{i=1}^{k} \gamma_i \cdot (\pi(\{i, \ldots, k\}) - \pi(\{i+1, \ldots, k\})) = \sum_{i=1}^{k} \gamma_i \cdot y_i.
\end{aligned}$$

Thus, from weak LP duality, they in fact form optimum solutions to primal and dual LPs, respectively. □

Recall that a vector $d \in \Re^k$ is a subgradient of a convex function $g : \Re^k \to \Re$ at a point $\gamma \in \Re^k$ if for any $\gamma' \in \Re^k$, we have $g(\gamma') - g(\gamma) \geq d \cdot (\gamma' - \gamma)$. For a differentiable convex function g, the subgradient corresponds to gradient ∇g. The function $\textsc{p}(\gamma)$ defined in Lemma 2.4 is essentially Lovasz's continuous extension of the submodular function π. The fact that $\textsc{p}$ is convex and its subgradient can be computed efficiently is given in Fujishige [2005]. We provide a full proof for completeness of exposition.

LEMMA 2.5. *The function* $\textsc{p}(\gamma)$ *in Lemma* 2.4 *is convex and given* $\gamma \in [0, 1]^k$*, both* $\textsc{p}(\gamma)$ *and its subgradient* $\nabla\textsc{p}(\gamma)$ *can be computed in polynomial time.*

PROOF. We first prove that $\textsc{p}$ is convex. Fix $\gamma_1, \gamma_2 \in [0, 1]^k$ and $\alpha \in [0, 1]$. To show that $\textsc{p}$ is convex, we will show $\textsc{p}(\alpha\gamma_1 + (1-\alpha)\gamma_2) \leq \alpha\textsc{p}(\gamma_1) + (1-\alpha)\textsc{p}(\gamma_2)$. Let $\{z_I^1\}$ and $\{z_I^2\}$ be the optimum solutions of the primal LP defining $\textsc{p}$ for γ_1 and γ_2, respectively. Note that the solution $\{\alpha z_I^1 + (1-\alpha) z_I^2\}$ is feasible for this LP for $\gamma = \alpha\gamma_1 + (1-\alpha)\gamma_2$. Thus, the optimum solution has value not greater than the value of this solution, which is $\alpha\textsc{p}(\gamma_1) + (1-\alpha)\textsc{p}(\gamma_2)$.

From Lemma 2.4, it is clear that given $\gamma \in [0, 1]^k$, the value $\textsc{p}(\gamma)$ can be computed in polynomial time. Lemma 2.4 also implies that the optimum dual solution $y^* = (y_1^*, \ldots, y_k^*) \in \Re_+^k$ can be computed in polynomial time. We now argue that y^* is a subgradient of $\textsc{p}$ at γ. Fix any $\gamma' \in \Re^k$. First note that, from LP duality, $\textsc{p}(\gamma) = y^* \cdot \gamma$. Thus, we have

$$\textsc{p}(\gamma) + y^* \cdot (\gamma' - \gamma) = y^* \cdot \gamma + y^* \cdot (\gamma' - \gamma) = y^* \cdot \gamma' \leq \textsc{p}(\gamma').$$

The last inequality holds from weak LP duality since y^* is a feasible solution for the dual LP $\text{D}(\gamma')$ as well. The lemma follows. □

3. PROOF OF LEMMA 2.3

We now describe how to round an LP (2) solution to obtain a $(\rho+1)$-approximation for PC-GSN. Later, we show how to improve the guarantee to $(1-e^{-1/\rho})^{-1}$. Let $\{x_e^*, f_{i,e}^*, z_I^*\}$ be a feasible solution to LP (2). Let $\alpha \in (0,1)$ be a parameter to be fixed later. We partition the requirements into two classes: we call a requirement $i \in K$ *good* if $\sum_{I \ni i} z_I^* \leq \alpha$ and *bad* otherwise. Let K_g denote the set of good requirements. The following statement shows how to satisfy the good requirements.

LEMMA 3.1. *There exists a polynomial-time algorithm that computes a subgraph H of G of cost $c(H) \leq \frac{\rho}{1-\alpha} \cdot \sum_e c_e x_e^*$ that satisfies all good requirements.*

PROOF. Consider the LP-relaxation (1) of the GSN problem with good requirements only, with K replaced by K_g; namely, we seek a minimum cost subgraph H of G that satisfies the set K_g of good requirements. We claim that $x_e^{**} = \min\{1, x_e^*/(1-\alpha)\}$ for each $e \in E$ is a feasible solution to LP (1). Thus, the optimum value of LP (1) is at most $\sum_{e\in E} c_e x_e^{**}$. Consequently, using the algorithm that computes an integral solution to LP (1) of cost at most ρ times the optimal value of LP (1), we can construct a subgraph H that satisfies all good requirements and has cost at most $c(H) \leq \rho \sum_{e\in E} c_e x_e^{**} \leq \frac{\rho}{1-\alpha} \sum_e c_e x_e^*$, as desired.

We now show that $\{x_e^{**}\}$ is a feasible solution to LP (1), namely, that $\sum_{e\in\delta(T)} x_e^{**} \geq r_i(T)$ for any $i \in K_g$ and any $T \odot (i, S)$. Let $i \in K_g$ and let $\zeta_i = 1 - \sum_{I\ni i} z_I^*$. Note that $\zeta_i \geq 1-\alpha$, by the definition of K_g. By the second and the third sets of constraints in LP (2), for every $e \in E$ we have $\min\{\zeta_i, x_e^*\} \geq f_{i,e}^*$. Thus we obtain: $x_e^{**} = \min\{1, \frac{x_e^*}{1-\alpha}\} = \frac{1}{\zeta_i}\min\{\zeta_i, \frac{\zeta_i}{1-\alpha} x_e^*\} \geq \frac{1}{\zeta_i}\min\{\zeta_i, x_e^*\} \geq \frac{f_{i,e}^*}{\zeta_i} = \frac{f_{i,e}^*}{1-\sum_{I\ni i} z_I^*}$. Consequently, combining with the first set of constraints in LP (2), for any $T \odot (i, S)$ we obtain that $\sum_{e\in\delta(T)} x_e^{**} \geq \frac{\sum_{e\in\delta(T)} f_{i,e}^*}{1-\sum_{I\ni i} z_I^*} \geq r_i(T)$. □

Let H be as in Lemma 3.1, and recall that $\texttt{unsat}(H)$ denotes the set of requirements not satisfied by H. Clearly, each requirement $i \in \texttt{unsat}(H)$ is bad. The following lemma bounds the total penalty we pay for $\texttt{unsat}(H)$.

LEMMA 3.2. $\pi(\texttt{unsat}(H)) \leq \frac{1}{\alpha} \cdot \sum_I \pi(I) z_I^*$.

PROOF. Define $\gamma \in [0,1]^k$ as follows: $\gamma_i = 1$ if $i \in \texttt{unsat}(H)$ and 0 otherwise. Now consider LP $\text{P}(\gamma)$ defined in Lemma 2.4. Since each $i \in \texttt{unsat}(H)$ is bad, from the definition of bad requirements, it is clear that $\{z_I^*/\alpha\}$ is a feasible solution to LP $\text{P}(\gamma)$. Furthermore, from Lemma 2.4, the solution $\{z_I\}$ defined as $z_I = 1$ if $I = \texttt{unsat}(H)$ and 0 otherwise is the optimum solution to $\text{P}(\gamma)$. The cost of this solution, $\pi(\texttt{unsat}(H))$, is therefore at most the cost of the feasible solution $\{z_I^*/\alpha\}$, which is $\frac{1}{\alpha} \cdot \sum_I \pi(I) z_I^*$. The lemma thus follows. □

Combining Lemmas 3.1 and 3.2, we obtain $\max\{\frac{\rho}{1-\alpha}, \frac{1}{\alpha}\}$-approximation. If we substitute $\alpha = 1/(\rho+1)$, we obtain a $(\rho+1)$-approximation for PC-GSN.

3.1. Improving the Approximation to $(1 - e^{-1/\rho})^{-1}$

We pick α uniformly at random from the interval $(0, \beta]$ where $\beta = 1 - e^{-1/\rho}$. From Lemmas 3.1 and 3.2, the expected cost of the solution is at most

$$\mathbb{E}_\alpha\left[c(H) + \pi(\texttt{unsat}(H))\right] = \mathbb{E}_\alpha\left[\frac{\rho}{1-\alpha}\right] \cdot \sum_{e\in E} c_e x_e^* + \mathbb{E}_\alpha\left[\pi(\texttt{unsat}(H))\right]. \tag{3}$$

To complete the proof of $\frac{1}{\beta}$-approximation, we now argue that this expectation is at most $\frac{1}{\beta} \cdot \sum_{e\in E}(c_e x_e^* + \sum_I \pi(I) z_I^*)$.

Note that $\mathbb{E}_\alpha[\frac{\rho}{1-\alpha}] = \frac{1}{\beta}\int_0^\beta \frac{\rho}{1-\alpha} d\alpha = \frac{\rho}{\beta} \cdot (-\ln(1-\beta) + \ln 1) = \frac{\rho}{\beta} \cdot (-\frac{1}{\rho}) = \frac{1}{\beta}$, the first term in (3) is at most $\frac{1}{\beta} \cdot \sum_{e\in E} c_e x_e^*$. Since $\texttt{unsat}(H) \subseteq \{i \mid \sum_{I\ni i} z_I^* \geq \alpha\}$ and since π is monotone nondecreasing, the second term in (3) is at most $\mathbb{E}_\alpha[\pi(\{i \mid \sum_{I\ni i} z_I^* \geq \alpha\})]$. Lemma 3.3 bounds this quantity as follows. The ideas used here are also presented in Sharma et al. [2007].

LEMMA 3.3. *We have*

$$\mathbb{E}_\alpha\left[\pi\left(\left\{i \mid \sum_{I\ni i} z_I^* \geq \alpha\right\}\right)\right] \leq \frac{1}{\beta} \cdot \sum_I \pi(I) z_I^*. \tag{4}$$

PROOF. Let $\gamma_i = \sum_{I\ni i} z_I^*$ for all $i \in K$. Let us, without loss of generality, order the elements $i \in K$ such that $\gamma_1 \leq \gamma_2 \leq \cdots \leq \gamma_k$. We also use the notation $\gamma_0 = 0$. Note that $\{z_I^*\}$ forms a feasible solution to the primal LP $\mathrm{P}(\gamma)$ given in Lemma 2.4. Therefore, from Lemma 2.4, its objective value is at least that of the optimum solution:

$$\sum_I \pi(I) z_I^* \;\geq\; \sum_{i=1}^{k} \left[(\gamma_i - \gamma_{i-1}) \cdot \pi(\{i, \ldots, k\})\right]. \tag{5}$$

We now observe that the left-hand side of (4) can be expressed as follows. Since α is picked uniformly at random from $(0, \beta]$, we have that for all $1 \leq i \leq k$, with probability at most $\frac{\gamma_i - \gamma_{i-1}}{\beta}$, the random variable α lies in the interval $(\gamma_{i-1}, \gamma_i]$. When this event happens, we get that $\{i' \mid \sum_{I\ni i'} z_I^* \geq \alpha\} = \{i' \mid \gamma_{i'} \geq \alpha\} = \{i, \ldots, k\}$. Thus, the expectation in left-hand side of (4) is at most

$$\sum_{i=1}^{k} \left[\frac{\gamma_i - \gamma_{i-1}}{\beta} \cdot \pi(\{i, \ldots, k\})\right]. \tag{6}$$

From expressions (5) and (6), the lemma follows. □

Thus, the proof of $(1 - e^{-1/\rho})^{-1}$-approximation is complete. It is worth mentioning that, in this section, we obtain a solution with a bound on its expected cost. However, the choice of α can be simply derandomized by trying out all the breakpoints where a good demand pair becomes a bad one (plus 0 and β).

4. PROOF OF LEMMA 2.2

We next show that although LP (2) has exponential number of variables and constraints, the following lemma holds.

LEMMA 4.1. *Any basic feasible solution to LP (2) has a polynomial number of non-zero variables.*

PROOF. Fix a basic feasible solution $\{x_e^*, f_{i,e}^*, z_i^*\}$ to (2). For $i \in K$, let

$$\gamma_i = 1 - \frac{\min_{T:T\odot i} \sum_{e\in\delta(T)} f_{i,e}^*}{r_i} \qquad \text{and} \qquad \gamma_i' = 1 - \max_e f_{i,e}^*\,.$$

Now fix the values of variables $\{x_e, f_{i,e}\}$ to $\{x^*_e, f^*_{i,e}\}$ and project the LP (2) onto variables $\{z_I\}$ as follows.

$$\sum_{e\in E} c_e x^*_e + \min\left\{\sum_{I\subseteq K}\pi(I)z_I \,\middle|\, \sum_{I\subseteq K} z_I = 1, \gamma_i \le \sum_{I\ni i} z_I \le \gamma'_i \;\forall i\in K, z_I\ge 0\;\forall I\subseteq K\right\}. \quad (7)$$

Since $\{x^*_e, f^*_{i,e}, z^*_i\}$ is a basic feasible solution to (2), it cannot be written as a convex combination of two distinct feasible solutions to (2). Thus, we get that $\{z^*_I\}$ cannot be written as a convex combination of two distinct feasible solutions to (7), and hence it forms a basic feasible solution to (7). Since there are $1+2|K|$ nontrivial constraints in (7), at most $1+2|K|$ variables z_I can be non-zero in any basic feasible solution of (7). The lemma follows. □

We prove that LP (2) can be solved in polynomial time. Introduce variables $\gamma \in [0,1]^k$ and obtain the following program (the function P is as in Lemma 2.4):

$$\begin{array}{lrcll}
\text{Minimize} & \sum_{e\in E} c_e x_e & + & \text{P}(\gamma) & \\
\text{Subject to} & \sum_{e\in\delta(T)} f_{i,e} & \ge & (1-\gamma_i) r_i(T) & \forall i \;\; \forall T\odot(i,S) \\
& f_{i,e} & \le & 1-\gamma_i & \forall i \;\; \forall e \\
& x_e & \ge & f_{i,e} & \forall i \;\; \forall e \\
& x_e, f_{i,e}, \gamma_i & \in & [0,1] & \forall i \;\; \forall e.
\end{array} \quad (8)$$

It is clear that solving (8) is enough to solve (2). Now note that this is a convex program since P is a convex function. To solve (8), we convert its objective function into a constraint $\sum_{e\in E} c_e x_e + \text{P}(\gamma) \le \text{OPT}$ where OPT is the target objective value and thus reduce it to a feasibility problem using standard techniques. Now to find a feasible solution using an ellipsoid algorithm, we need to show a polynomial-time separation oracle.

The separation oracle for the constraints that are not in the first set of constraints is trivial. The separation oracle for the first set of constraints can be reduced to the problem of finding a minimum capacity mixed cut (containing both nodes and edges) that separates between two given nodes, in an undirected network with both edge and node capacities. The latter problem can be solved in polynomial time using standard network-flow algorithms. The reduction is as follows. Consider the constraints for a specific $i\in K$. Denoting $d_e = f_{i,e}/(1-\gamma_i)$, the constraints set for i can be rewriten as

$$\sum_{e\in\delta(T)} d_e \ge r_i(T) \quad \forall T\odot(i,S).$$

By the definition of $r_i(T)$, this is equivalent to

$$\sum_{e\in\delta(T)} d_e + |V\setminus(T'\cup T'')| \ge r_i \quad \forall T\odot(i,S). \quad (9)$$

View the input graph G as a network with source u_i, sink v_i, edge capacities $\{d_e : e\in E\}$, and node capacities $d_v = 1$ if $v\in S\setminus\{u_i, v_i\}$ and $d_v=\infty$ otherwise. Then, the following statement finishes the reduction.

LEMMA 4.2. *There exists a setpair $T=(T',T'')$ with $T\odot(i,s)$ for which (9) is violated if, and only if, in the network above there exists a mixed cut C of capacity strictly less than r_i that separates between u_i and v_i.*

Proof. Let $T = (T', T'')$ be a setpair with $T \odot (i, s)$ for which (9) is violated. Let $C = \delta(T) \cup (V \setminus (T' \cup T''))$. Then, the capacity of C equals the left-hand side of (9), and thus is strictly less than r_i, since (9) is violated for T.

Now suppose that a mixed cut C as in the lemma exists. Let $T_C = (T', T'')$ be a setpair defined as follows: T' is the connected component of $G \setminus C$ containing u_i and $T'' = V \setminus (C \cup T')$. Then, the left-hand side of (9) is at most the capacity of C. Consequently, the constraint (9) of the setpair T_C is violated, and $T_C \odot (i, S)$ holds, since the capacity of C is less than r_i and thus is finite. □

The separation oracle for the objective function is as follows. Given a point $(x, \gamma) = \{x_e, \gamma_i\}$ that satisfies $\sum_{e\in E} c_e x_e + \mathrm{P}(\gamma) > \mathrm{OPT}$, we compute a subgradient of the function $\sum_{e\in E} c_e x_e + \mathrm{P}(\gamma)$ with respect to variables $\{x_e, \gamma_i\}$. The subgradient of $\sum_{e\in E} c_e x_e$ with respect to x is simply the cost vector c. The subgradient of $\mathrm{P}(\gamma)$ with respect to γ is computed using Lemma 2.5; denote it by $y \in \Re^k_+$. From the definition of subgradient, we have that the subgradient (c, y) to the objective function at point (x, γ) satisfies

$$\left(\sum_{e\in E} c_e x'_e + \mathrm{P}(\gamma')\right) - \left(\sum_{e\in E} c_e x_e + \mathrm{P}(\gamma)\right) \geq (c, y) \cdot \big((x', \gamma') - (x, \gamma)\big).$$

Now fix any feasible solution (x^*, γ^*), that is, one that satisfies $\sum_{e\in E} c_e x^*_e + \mathrm{P}(\gamma^*) \leq \mathrm{OPT}$. Substituting $(x', \gamma') = (x^*, \gamma^*)$ in this equation we get,

$$\begin{aligned} 0 \;=\; \mathrm{OPT} - \mathrm{OPT} \;&>\; \left(\sum_{e\in E} c_e x^*_e + \mathrm{P}(\gamma^*)\right) - \left(\sum_{e\in E} c_e x_e + \mathrm{P}(\gamma)\right) \\ &\geq\; (c, y)\cdot(x^*, \gamma^*) - (c, y)\cdot(x, \gamma). \end{aligned}$$

Thus, (c, y) defines a separating hyperplane between the point (x, γ) and any point (x^*, γ^*) that satisfies $\sum_{e\in E} c_e x^*_e + \mathrm{P}(\gamma^*) \leq \mathrm{OPT}$. Hence, we have a polynomial-time separation oracle for the objective function as well.

Thus, we can solve (8) using the ellipsoid algorithm. The proof of Lemma 2.2 is hence complete. □

REFERENCES

Agrawal, A., Klein, P., and Ravi, R. 1995. When trees collide: An approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput. 24,* 3, 440–456.

Archer, A., Bateni, M., Hajiaghayi, M., and Karloff, H. 2009. Improved approximation algorithms for prize-collecting Steiner tree and TSP. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*. 427–436.

Balas, E. 1989. The prize collecting traveling salesman problem. *Networks 19,* 6, 621–636.

Bern, M. and Plassmann, P. 1989. The Steiner problem with edge lengths 1 and 2. *Inf. Proc. Lett. 32*, 171–176.

Bienstock, D., Goemans, M., Simchi-Levi, D., and Williamson, D. 1993. A note on the prize collecting traveling salesman problem. *Math. Prog. 59,* 3, Ser. A, 413–420.

Byrka, J., Grandoni, F., Rothvoss, T., and Sanita, L. 2010. An improved LP-based approximation for Steiner tree. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing*. 583–592.

Cheriyan, J., Vempala, S., and Vetta, A. 2006. Network design via iterative rounding of setpair relaxations. *Combinatorica 26,* 3, 255–275.

Chudak, F., Roughgarden, T., and Williamson, D. 2001. Approximate k-MSTs and k-Steiner trees via the primal-dual method and Lagrangean relaxation. In *Proceedings of the 8th International Conference on Integer Programming and Combinatorial Optimization*. 60–70.

Chuzhoy, J. and Khanna, S. 2009. An $O(k^3 \log n)$-approximation algorithms for vertex-connectivity network design. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*. 437–441.

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

Fujishige, S. 2005. *Submodular Functions and Optimization*. Elsevier.

Goemans, M. and Williamson, D. 1995. A general approximation technique for constrained forest problems. *SIAM J. Comput. 24,* 2, 296–317.

Gupta, A., Könemann, J., Leonardi, S., Ravi, R., and Schäfer, G. 2007. An efficient cost-sharing mechanism for the prize-collecting Steiner forest problem. In *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms*. 1153–1162.

Hajiaghayi, M. and Jain, K. 2006. The prize-collecting generalized Steiner tree problem via a new approach of primal-dual schema. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*. 631–640.

Hajiahayi, M. and Nasri, A. 2010. Prize-collecting Steiner networks via iterative rounding. In *Proceedings of LATIN Theoretical Informatics*. 515–526.

Jain, K. 2001. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica 21,* 1, 39–60.

Jain, K. and Vazirani, V. 2001. Approximation algorithms for metric facility location and *k*-median problems using the primal-dual schema and Lagrangian relaxation. *J. ACM 48,* 2, 274–296.

Johnson, D., Minkoff, M., and Phillips, S. 2000. The prize collecting Steiner tree problem: Theory and practice. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*. 760–769.

Korte, B. and Vygen, J. 2002. *Combinatorial Optimization: Theory and Algorithms*. Springer, Berlin.

Kortsarz, G. and Nutov, Z. 2007. Approximating minimum cost connectivity problems, In *Approximation Algorithms and Metahueristics*. T. F. Gonzales, Ed., CRC Press, Boca Raton, FL, Chap. 58.

Nagarajan, C., Sharma, Y., and Williamson, D. 2008. Approximation algorithms for prize-collecting network design problems with general connectivity requirements. In *Proceedings of the 6th International Workshop on Approximation and Online Algorithms*. 174–187.

Nutov, Z. 2010. Approximating minimum cost connectivity problems via uncrossable bifamilies. Manuscript.

Robins, G. and Zelikovsky, A. 2005. Tighter bounds for graph Steiner tree approximation. *SIAM J. Disc. Math. 19,* 1, 122–134.

Salman, F., Cheriyan, J., Ravi, R., and Subramanian, S. 2000. Approximating the single-sink link-installation problem in network design. *SIAM J. Optimiza. 11,* 3, 595–610.

Sharma, Y., Swamy, C., and Williamson, D. 2007. Approximation algorithms for prize collecting forest problems with submodular penalty functions. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*. 1275–1284.

Received July 2010; revised January 2011; accepted January 2011