Approximation Algorithms for Prize-Collecting Network Design Problems with General Connectivity Requirements

Chandrashekhar Nagarajan[1⋆], Yogeshwer Sharma[2⋆⋆], and David P. Williamson[1⋆⋆⋆]

[1] School of OR&IE, Cornell University, Ithaca, NY 14853.
[2] Department of Computer Science, Cornell University, Ithaca, NY 14853.

Abstract. In this paper, we introduce the study of prize-collecting network design problems having general connectivity requirements. Prior work considered only 0-1 or very limited connectivity requirements. We introduce general connectivity requirements in the prize-collecting generalized Steiner tree framework of Hajiaghayi and Jain [9], and consider penalty functions linear in the violation of the connectivity requirements. Using Jain's iterated rounding algorithm [11] as a black box, and ideas from Goemans [7] and Levi, Lodi, Sviridenko [14], we give a 2.54-factor approximation algorithm for the problem. We also generalize the 0-1 requirements of PCF problem introduced by Sharma, Swamy, and Williamson [15] to include general connectivity requirements. Here we assume that the monotone submodular penalty function of Sharma et al. is generalized to a multiset function that can be decomposed into functions in the same form as that of Sharma et al. Using ideas from Goemans and Berstimas [6], we give an $(\alpha \log K)$-approximation algorithm for the resulting problem, where K is the maximum connectivity requirement, and $\alpha = 2.54$.

1 Introduction

Over the past two decades, there has been a significant amount of work in the study of approximation algorithms for finding low-cost networks with specific connectivity requirements; see, for example, [6, 1, 8, 16, 11]. Kortsarz and Nutov [13] give a survey of some of this work. In *prize-collecting* network design problems, connectivity requirements become "soft" constraints; we may drop them if other considerations (such as cost) become more important. This is usually expressed through penalties on the connectivity requirements. We may drop the connectivity requirement if we are willing to instead pay the penalty.[3]

⋆ Email: `chandra@orie.cornell.edu`. Supported in part by NSF CCF-0514628.
⋆⋆ Email: `yogi@cs.cornell.edu`. Supported in part by NSF CCF-0514628.
⋆⋆⋆ Email: `dpw@cs.cornell.edu`. Supported in part by NSF CCF-0514628.

[3] One may well wonder why problems with penalties are called 'prize-collecting'. The answer is an historical accident. Balas [2] introduced the prize-collecting traveling

One of the first problems studied from this perspective was the *prize-collecting Steiner tree problem* (PCST). This problem is a variant of the Steiner tree problem in which we are given an undirected graph with nonnegative edge costs, a root vertex r, and a nonnegative penalty π_i for each vertex $i \in V$. The goal is to find a minimum-cost tree T connected to the root such that we minimize the cost of the tree plus the sum of the penalties of all vertices not spanned by the tree. The problem models one of a network provider deciding how to expand a network so as to maximize its profit [12]; here the penalty of a vertex represents the potential revenue that can be captured if the vertex is connected to the network. The objective function is that of maximizing the sum of the revenues generated by the vertices connected to the network minus the cost of the network. This objective gives the same optimal solution as that of the prize-collecting Steiner tree problem, but from an approximability standpoint the two problems are not the same; Feigenbaum, Papadimitriou, and Shenker [5] have shown the profit maximization problem cannot be approximated to within any factor. Bienstock, Goemans, Simchi-Levi, and Williamson [3] gave the first approximation algorithms for the PCST. Goemans and Williamson later gave a primal-dual 2-approximation algorithm [8].

Only recently have the generalizations of PCST been considered. Hajiagayi and Jain [9] consider the prize-collecting version of the generalized Steiner tree problem (abbreviated to PCGST for "prize-collecting generalized Steiner tree"). In the PCGST problem, we are given an undirected graph with nonnegative costs on the edges and a set of pairs of vertices s_i-t_i with penalties π_i for each i for not connecting that particular pair. The goal is to find a subset F of edges so as to minimize the cost of the selected edges plus the penalties of pairs not connected in the subgraph (V, F). Hajiaghayi and Jain give a primal-dual 3-approximation algorithm and an LP-rounding 2.54-approximation algorithm. Hayrapetyan, Swamy and Tardos [10] extend the PCST framework in a different direction, by considering the case when penalties are more expressive than just a simple sum of penalties of disconnected vertices. In their model, penalties are modelled by an arbitrary monotone submodular function of the set of disconnected vertices. They are able to extend the Goemans-Williamson primal-dual 2-approximation algorithm to this problem.

Sharma, Swamy, and Williamson [15] introduce prize-collecting forest problems (PCF for short) which generalize connectivity requirements still further and make penalties more expressive; this work generalizes both the results of Hajiaghayi and Jain and Hayrapetyan et al. In the PCF problem, the connectivity requirements are specified by an arbitrary function $f : 2^V \rightarrow \{0, 1\}$ which assigns a connectivity requirement to each subset of vertices; and the penalty is specified by a submodular function $\pi : 2^{2^V} \rightarrow \Re^+$ on *collections* of subsets of vertices. The goal is to find a subset F of edges such that the cost of selected

salesman problem, which had prizes that were collected by the salesman as he visited various nodes, and penalties for unvisited nodes. Bienstock et al. [3] dropped the prizes, but kept the penalties and the name. Most subsequent work has addressed the Bienstock et al. variant of the problem.

edges F plus the penalty function value on the collection of all *violated* subsets is minimized, where a subset S of vertices is said to be violated if $f(S) = 1$ but $\delta(S) \cap F = \emptyset$; $\delta(S)$ is the set of all edges with exactly one end point in S. Sharma et al. [15] give a primal-dual 3-approximation algorithm and an LP-rounding 2.54-approximation algorithm for the PCF problem when the penalty function obeys certain properties.

All of the work outlined above considers only problems where the network created is a tree or a forest. However, many fundamental network design questions involve more general connectivity requirements. In real world networks, a typical client might not just like to connect to the network, but might want to connect via a few (different) paths. There could be several reasons for this including needing higher bandwidth than a single connection can provide, or needing redundant connections in case of edge failures. For instance, in the survivable network design problem (called SNDP for short), the input is the same as that for generalized Steiner tree problem, except that now we are also given a connectivity requirement r_i for each pair s_i-t_i, and we need at least r_i edge-disjoint paths from s_i to t_i in the solution network. Jain [11] introduces the technique of iterated rounding and gives a 2-approximation algorithm for this problem. His technique extends to network design problems in which for every cut $\delta(S)$ in the network we must select at least $f(S)$ edges, where f is a weakly supermodular function.

In this paper, we initiate the investigation of prize-collecting network design problems with general connectivity requirements. There has been some previous work investigating prize-collecting network design problems with connectivity requirements greater than 1 (see below for discussion), but to the best of our knowledge, there has been no previous work on prize-collecting problems with general connectivity requirements. In this investigation, we consider prize-collecting generalized Steiner tree problem (PCGST) and prize-collecting forest problem (PCF) and introduce general connectivity versions for these problems. We also design and analyze approximation algorithms for these problems.

As mentioned above, there has been some previous work on prize-collecting network design problems with connectivity requirements greater than 1. Based on a previous problem of Balas [2], Bienstock et al. [3] introduced the the prize-collecting travelling salesman problem (PCTSP). In PCTSP problem, we are given an undirected edge-weighted graph $G = (V, E, c)$ (edge weights satisfy triangle inequality), a root $r \in V$, and penalties π_i for all $i \in V$. The goal is to find a tour which includes root r, but excludes some (possibly empty) set of vertices such that the sum of the cost of the tour and aggregate penalty of excluded vertices is minimized. Goemans and Williamson [8] give a 2-approximation for this problem. Chimani, Kandyba, and Mutzel [4] consider the 2-root-connected prize-collecting Steiner network problem. In this problem, each node $v \in V$ has a connectivity requirement $r_v \in \{0, 1, 2\}$, which indicates how many *node-disjoint* paths it requires to the root. There is also a penalty π_v for not having r_v node disjoint paths from v to the root. The goal is then to select a subset E' of edges such that the cost of E' plus the sum of penalties of nodes whose connectivity requirement is not satisfied is minimized. Chimani, Kandyba, and Mutzel give an integer linear programming approach based on directed cuts to solve this problem to optimality.

The issue of how to define penalties immediately arises while considering prize-collecting problems with general connectivity. If we need to select $f(S)$ edges from $\delta(S)$, how much penalty should be charged if there are only $f(S)-1$? There are two obvious variants: (1) satisfaction of requirement is all-or-nothing, (as in Chimani, Kandyba, Mutzel [4]); (2) satisfaction of requirements is gradual in the sense that each violation carries an additional penalty. In this paper, we model the gradual version of the penalty with the following restriction: the earlier violation of the connectivity requirement of a set S costs at most as much as the later violations; that is, if the connectivity requirement is K, then reducing connectivity from K to $K-1$ carries at most as much additional penalty as reducing connectivity from $K-1$ to $K-2$, which in turn carries as most as much additional penalty as reducing it from $K-2$ to $K-3$ and so on. Designing algorithms for all-or-nothing version of the penalty is an important open problem.

We first consider the prize-collecting version of the survivable network design problem (called PCSNDP for short) with linear penalties. The input to this problem is the same as the SNDP problem, except that now there is penalty π_i associated with each pair s_i-t_i. The goal is to select a subset E' of edges so as to minimize the cost of the edges selected plus the sum over all pairs of the product of π_i and number of times its requirement is violated; the penalty is linear in this sense, as each additional violation costs the same. We use Jain's algorithm for the survivable network design problem as a black-box and use some variations of rounding techniques of Levi, Lodi, and Sviridenko [14] and of Goemans [7] to give a 2.54-approximation algorithm for this case.

Then we consider the PCF problem with integral connectivity requirements (called PCF-$\mathbb{Z}$ for short). The connectivity requirement function is any function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$. The penalty function is a function $\pi : (K+1)^{2^V} \rightarrow \mathbb{R}_{\geq 0}$ where K is the maximum value of the connectivity function f. The penalty function π is assumed to be decomposable in certain form, which reflects the fact that the multiset function $\pi(\cdot)$ satisfies a variant of the submodularity property. Each decomposed function of π is assumed to satisfy the conditions mentioned in earlier work of [15], in particular, it is a monotone submodular function. Without going into the details of the decomposition, it is worth pointing out that the decomposition is motivated by the fact that users have decreasing marginal utility for increased connectivity. We show an example of a SNDP problem with penalties convex in the number of missing connections that is expressible in our framework. Borrowing ideas from Goemans and Berstimas [6], we give an $(\alpha \cdot \log K)$-approximation algorithm for this problem, where $\alpha \approx 2.54$ is the approximation factor of the LP-rounding algorithm in [15] and K is the maximum connectivity value. For this result, we allow ourselves to purchase as many copies of an edge as needed; in the previous result for PCSNDP we can restrict the number of copies of each edge in the solution.

The rest of the paper is organized as follows. In Section 2, we introduce the prize collecting variant of the survivable network design problem, and give a 2.54-approximation algorithm. The general connectivity version of prize-collecting forest problem is introduced in Section 3, where we also give an algorithm for the problem. The algorithm is analyzed in Section 4. Section 5 concludes with open problems and future work.

2 The prize-collecting survivable network design problem

2.1 The problem definition

In the PCSNDP, we are given an edge weighted graph $G = (V, E, c : E \to \mathbb{R}_{\geq 0})$, a set I of k source sink pairs s_i-t_i for $i \in I$, a requirement r_i for each pair, and penalty π_i for each i for violating each connectivity requirement for that pair. The goal is to find a subset E' of edges such that the cost of the edges in E' and penalties of all disconnections (described below) is minimized. If pair i has q_i edge disjoint paths in E', then the disconnection penalty of pair i is π_i times $\max(0, r_i - q_i)$. The total disconnection penalty is the sum of disconnection penalties over all pairs, that is $\sum_{i \in I} \pi_i \cdot \max(0, r_i - q_i)$. There is also a requirement that each edge $e \in E$ can be used at most $a_e \in \mathbb{Z}_{\geq 0}$ number of times.

The linear programming relaxation of the problem is as follows. The x_e variables represent the decision of including edges and z_i variables represent the decision to pay penalties for pairs. We use $S \odot i$ as a binary predicate which is true if $|S \cap \{s_i, t_i\}| = 1$.

$$\text{Min} \quad \sum_{e \in E} c_e x_e + \sum_{i \in I} \pi_i z_i \qquad \text{subject to} \qquad \text{(PCSNDP)}$$

$$\sum_{e \in \delta(S)} x_e + z_i \geq r_i \quad \forall i, S : S \odot i; \quad 0 \leq x_e \leq a_e \quad \forall e \in E; \quad z_i \geq 0 \quad \forall i \in I.$$

2.2 Idea of the algorithm

We first solve the LP using ellipsoid method. For that, we need a separation oracle for the inequalities. A simple max flow computation for each pair in I tells us whether the current solution is feasible for the linear program or not.

Let (x^*, z^*) be the optimal solution to the linear program. Then we "round" the z^* part of the solution. Let us fix a real number $\alpha \in [0, 1]$ and define $\bar{z}$ solution as follows, where $\{r\}$ denotes the fractional part of the number r: $\bar{z}_i = \lfloor z_i \rfloor$ if $\{z_i\} \leq \alpha$ and $\lceil z_i \rceil$ otherwise. In this process, we increase the penalty of the solution if we round the z_i variable up, and decrease it otherwise.

Then we solve a modified problem without prize-collecting constraints. The new problem has same graph and pairs as its input, but the connectivity requirements are adjusted to reflect the $\bar{z}_i$ variables' values. Let us define new requirements $r'_i = r_i - \bar{z}_i$. We then solve the SNDP using Jain's iterative rounding technique [11]. The LP for the modified problem is given below.

$$\text{Min} \quad \sum_{e \in E} c_e x_e \quad \text{subject to} \qquad \text{(SNDP)}$$

$$\sum_{e \in \delta(S)} x_e \geq r'_i (= r_i - \bar{z}_i) \quad \text{for all } i, S : S \odot i; \quad 0 \leq x_e \leq a_e \quad \text{for all } e \in E.$$

Let the solution we get from Jain's algorithm be called $\bar{x}$. We return $(\bar{x}, \bar{z})$ as the final solution, which is a feasible solution to the original problem.

We are now ready to prove that the algorithm presented above is a 3-approximation algorithm. Towards the end of this section, we will also point out how we can change the algorithm to make it a 2.54-approximation algorithm.

2.3 Bounding the edge costs

First, we focus on the edge costs. Consider the solution $\bar{x}$ which is the output of Jain's algorithm. By the properties of Jain's LP-rounding 2-approximation algorithm, we know that $\sum_{e \in E} c_e \bar{x}_e \leq 2 \cdot \mathsf{OPT}(\text{SNDP})$, where $\mathsf{OPT}(\text{SNDP})$ is the cost of the optimal solution to the LP (SNDP). Our goal is to prove that the solution $\frac{1}{1-\alpha} x^*$ costs at least $\mathsf{OPT}(\text{SNDP})$ by proving that the solution $\frac{1}{1-\alpha} x^*$ is feasible for the linear program SNDP. This will prove the following:

$$\sum_{e \in E} c_e \bar{x}_e \leq 2 \cdot \mathsf{OPT}(\text{SNDP}) \leq 2 \cdot \frac{1}{1-\alpha} \sum_{e \in E} c_e x^*_e.$$

Although we would like to show that $\frac{1}{1-\alpha} x^*$ is a feasible solution to the modified connectivity requirements r'_i, this might potentially violate the constraint on the number of edges used. Instead, we define the solution x^{**} as $x^{**}_e = \min\{\frac{1}{1-\alpha} x^*_e, a_e\}$. Here is the idea: if the connectivity requirements r_i were to be reduced by exactly z^*_i, then the scaled solution $\frac{1}{1-\alpha} x^*$ will actually be feasible for modified connectivity requirements. But the requirements actually go down by only $\lfloor z^*_i \rfloor$, and we need to make sure in such cases the scaled solution (which is also truncated at a_e) is indeed feasible.

Lemma 1. *x^{**} is a feasible solution for modified connectivity requirements r'.*

Proof. We prove that x^{**} satisfies all constraints in the modified LP (SNDP) above. Let us fix an i and $S : S \odot i$, and assume that z_i has been rounded down (otherwise, the solution is trivially feasible). We have $x^*(\delta(S)) \geq r_i - \lfloor z^*_i \rfloor - \alpha$ and we need to prove that $x^{**}(\delta(S)) \geq r_i - \lfloor z^*_i \rfloor$. The fact that $x^*(\delta(S)) \geq r_i - \lfloor z^*_i \rfloor - \alpha$ can be restated as $x^*(\delta(S)) \geq (r_i - \lfloor z^*_i \rfloor - 1) + (1 - \alpha)$. The following claim, based on an idea from Levi, Lodi, and Sviridenko [14], will help us get the required bound.

Claim. Let $\sum_{e \in E} y_e \geq n + (1 - \alpha)$ and suppose there is a non-negative integer n_e for each edge e such that $y_e \leq n_e$. Then, $\sum_{e \in E} \min\{\frac{y_e}{1-\alpha}, n_e\} \geq n + 1$.

Proof. To prove the inequality, we define new y' and n'; proving the inequality for new values will prove it for original values too. Let

$$y'_e = y_e - \lfloor y_e \rfloor; \quad n'_e = n_e - \lfloor y_e \rfloor; \quad n' = n - \sum_{e \in E} \lfloor y_e \rfloor.$$

Now we have that $\sum_{e\in E} y'_e \geq n' + (1-\alpha)$ and we need to prove $\sum_{e\in E} \min\{\frac{y'_e}{1-\alpha}, n'_e\} \geq n'+1$. Note that proving this suffices to prove the claim. We will prove something stronger: $\sum_{e\in E} \min\{\frac{y'_e}{1-\alpha}, 1\} \geq n'+1$. Let $F = \{e \in E : y'_e \geq 1-\alpha\}$. If $|F| \geq n'+1$, then we are done (each edge in F contributes at least a unit to the sum). Else, $|F| \leq n'$. Then the sum on the left hand side can be written as the following, proving the claim.

$$\sum_{e\in E} \min\left\{\frac{y'_e}{1-\alpha}, 1\right\} = \sum_{e\in F} \min\left\{\frac{y'_e}{1-\alpha}, 1\right\} + \sum_{e\notin F} \min\left\{\frac{y'_e}{1-\alpha}, 1\right\}$$

$$= |F| + \sum_{e\notin F} \frac{y'_e}{1-\alpha} \geq |F| + \frac{1}{1-\alpha}\left(\sum_{e\in E} y'_e - |F|\right)$$

$$\geq |F| + \frac{1}{1-\alpha}(n' + (1-\alpha) - |F|)$$

$$\geq |F| + (n' - |F|) + \frac{1-\alpha}{1-\alpha} \geq n'+1.$$

For a given S, we apply the claim by letting $y_e = x^*_e$ for all $e \in \delta(S)$, $n_e = a_e$, and $n = r_i - \lfloor z^*_i \rfloor - 1$. Then the claim shows that $\sum_{e\in\delta(S)} \min\{\frac{x^*_e}{1-\alpha}, a_e\} \geq (r_i - \lfloor z^*_i \rfloor - 1) + 1$, or that $x^{**}(\delta(S)) \geq r_i - \lfloor z^*_i \rfloor$. Hence the constraint in the modified linear program corresponding to $(i, S : S \odot i)$ is satisfied. Also note that $x^{**}_e \leq a_e$ is also satisfied by the definition of x^{**}. This finishes the proof of the lemma.

2.4 Penalty of the solution and its total cost

Now we can also bound the cost of the solution contributed by penalties. Note that when we round the z^*_i variables up, we scale them up by a factor of at most $\frac{1}{\alpha}$. This proves that $\sum_{i\in I} \pi_i \bar{z}_i \leq \frac{1}{\alpha} \sum_{i\in I} \pi_i z^*_i$. Hence the total cost of the solution is $\sum_{e\in E} c_e \bar{x}_e + \sum_{i\in I} \pi_i \bar{z}_i \leq \frac{2}{1-\alpha} \sum_{e\in E} c_e x^*_e + \frac{1}{\alpha} \sum_{i\in I} \pi_i z^*_i$. Taking $\alpha = \frac{1}{3}$ gives a 3-approximation.

The algorithm above can be improved to 2.54 approximation by choosing α uniformly at random from $[0, \beta]$ where $\beta = 1 - e^{-1/2}$. This is a standard technique introduced by Goemans [7]; see the paper of Hajiaghayi and Jain [9] or Sharma, Swamy, and Williamson [15] for details.

3 The prize-collecting forest problem with general connectivity requirements (PCF-$\mathbb{Z}$)

3.1 Definition and formulation of the problem

In the discussion below, we denote a family of subsets of V by uppercase scripted letters (like $\mathcal{S}$) and multisets of subsets of V by uppercase scripted letters with a bar over them (like $\bar{\mathcal{S}}$). The set of all families of subsets of V is denoted by 2^{2^V} and the set of all multisets of subsets of V by $\mathbb{N}^{2^V}$. Also, the set of all multisets with multiplicity of any subset at most K is denoted by $(K+1)^{2^V}$ or $[0\dots K]^{2^V}$. For a multiset $\bar{\mathcal{S}}$ and a subset S, we denote by $n_{\bar{\mathcal{S}}}(S)$ the number of copies of S that are contained in $\bar{\mathcal{S}}$. For two (multi)sets $\bar{\mathcal{S}}$ and $\bar{\mathcal{T}}$, $\bar{\mathcal{S}}+\bar{\mathcal{T}}$ (called *multiset addition*) is defined such that $n_{\bar{\mathcal{S}}+\bar{\mathcal{T}}}(S) = n_{\bar{\mathcal{S}}}(S) + n_{\bar{\mathcal{T}}}(S)$, for all S.

PCF-$\mathbb{Z}$ is a generalization of the prize-collecting forest problem (called PCF) in Sharma, Swamy, and Williamson (SSW) [15]. In the PCF problem, the input is an edge weighted graph $G = (V, E, c : E \to \mathbb{R}_{\geq 0})$ with a connectivity requirement function $f : 2^V \to \{0,1\}$ (each subset S requires us to select $f(S)$ from the cut $\delta(S)$). A submodular connectivity function $\pi : 2^{2^V} \to \mathbb{R}_{\geq 0}$ on the family of all subsets is also given, which is used to determine the penalty of the solution. The goal is to find a minimum cost subgraph $G' = (V, E')$ of G such that the sum of edges costs in E' plus the penalty function value on the family of violated subsets is minimized. The LP relaxation of the integer programming formulation of the problem is given below.

$$\text{Min} \quad \sum_{e\in E} c(e)x(e) + \sum_{\mathcal{S}\in 2^{2^V}} \pi(\mathcal{S})z(\mathcal{S}) \qquad \text{subject to} \qquad \text{(PCF-P)}$$

$$\sum_{e\in\delta(S)} x(e) + \sum_{\mathcal{S}:S\in\mathcal{S}} z(\mathcal{S}) \geq f(S) \quad \forall S\subseteq V; \quad x(e), z(\mathcal{S}) \geq 0 \quad \forall e\in E, \mathcal{S}\in 2^{2^V}.$$

SSW give a primal-dual 3-approximation algorithm and an LP-rounding 2.54-approximation algorithm for this problem. To achieve their result, they needed a few restrictions on the penalty functions, in particular the penalty function is assumed to be monotone and submodular. Please refer to [15] for details.

We generalize the PCF to the PCF-$\mathbb{Z}$ problem, in which general connectivity requirements are allowed. In the PCF-$\mathbb{Z}$ problem, the input is same as the PCF problem, save the following two changes: (1) The connectivity requirement function f is defined from 2^V to $\mathbb{Z}_{\geq 0}$. (2) The penalty function $\pi(\cdot)$ is defined on multisets of subsets of 2^V, since now a subset can be violated multiple times.

According to the problem definition, each subset S in 2^V has a requirement of $f(S)$ edges to cross it in a solution. We call a set S *violated in the subgraph* E' if this requirement is not satisfied for the set S; that is, $|\delta(S)\cap E'| < f(S)$. Our objective is to find a network E' such that we pay for the cost of the edges in the network N and pay penalties for the sets S for which there are less than $f(S)$ edges in the cut $\delta(S)$ in the solution E'.

Here we model the penalty function as $\bar{\pi}(\bar{\mathcal{S}})$ where $\bar{\mathcal{S}}$ denotes the multiset of sets for which the requirement is violated, i.e the number of edges selected from the cut is less than the associated requirement for that set. If the subgraph chosen is E', then the multiplicity of a set S in $\bar{\mathcal{S}}$ is $f(S) - |\delta(S)\cap E'|$, i.e. the number of times the set is violated. So we define the violated multiset of a network E' as $\bar{\mathcal{S}}(E') = \{S : \max(f(S) - \delta_{E'}(S), 0) \text{ times }\}$. Let $K = \max_S f(S)$, the maximum connectivity requirement of any subset. Without loss of generality, we assume that $f(S) = K$ for all S because if $f(S)$ were less than K for some S then we can set $f(S) = K$ for all subsets and modify the penalty function $\bar{\pi}$ into $\bar{\pi}'$ in such a way that it is equivalent to the original problem. This can be achieved by the defining for any $\bar{\mathcal{S}} \in [0\dots K]^{2^V}$, $\bar{\pi}'(\bar{\mathcal{S}}) := \bar{\pi}(\bar{\mathcal{T}})$ where $n_{\bar{\mathcal{T}}}(S) = \max\{n_{\bar{\mathcal{S}}}(S) - (K - f(S)), 0\}$ for all $S\subseteq V$.

This problem can be formulated as an integer problem, whose LP relaxation is shown below:

$$\text{Min} \quad \sum_{e\in E} c(e)x(e) + \sum_{\bar{\mathcal{S}}\in\mathbb{N}^{2^V}} \pi(\bar{\mathcal{S}})z(\bar{\mathcal{S}}) \qquad \text{subject to} \qquad \text{(PCF-}\mathbb{Z}\text{-P)}$$

$$\sum_{e\in\delta(S)} x(e) + \sum_{\bar{\mathcal{S}}:S\in\bar{\mathcal{S}}} z(\bar{\mathcal{S}})\cdot n_{\bar{\mathcal{S}}}(S) \geq f(S) \qquad \forall S\subset V$$

$$x(e)\geq 0 \qquad \forall e\in E$$

$$0\leq \sum_{\bar{\mathcal{S}}} z(\bar{\mathcal{S}}) \leq 1.$$

In the linear program above, the last constraint is valid for integer optimum solution since only one multiset needs to be set to 1.

3.2 Properties of the penalty function $\bar{\pi}(\bar{\mathcal{S}})$

In this section, we lay out and motivate the assumptions on the penalty function.

1. **The penalty function is decomposable**: We assume that the penalty function $\bar{\pi}$ can be decomposed into $\pi_1,\ldots,\pi_K : 2^{2^V} \to \mathbb{R}_{\geq 0}$ such that $\bar{\pi}(\bar{\mathcal{S}}) = \pi_1(\mathcal{S}_1)+\pi_2(\mathcal{S}_2)+\ldots+\pi_K(\mathcal{S}_K)$, where each of $\pi_1, \pi_2, \ldots, \pi_K$ satisfies penalty functions properties in [15] and $\text{split}^K(\bar{\mathcal{S}}) = (\mathcal{S}_1,\mathcal{S}_2,\ldots,\mathcal{S}_K)$. Here, $\mathcal{S}_r$ is the family of all subsets in $\bar{\mathcal{S}}$ which occur at least $K-r+1$ times; that is, $\mathcal{S}_r = \{S : n_{\bar{\mathcal{S}}}(S)\geq K-r+1\}$. To refer to the $\mathcal{S}_r$, we use the notation $\text{split}^K(\bar{\mathcal{S}}, r)$. Note that changing the penalty function as mentioned in the previous section (so that we can assume $f(S) = K$ for all S) does not affect whether π is decomposable in this fashion.

2. **The penalty function is monotonic**. We assume $\pi_1(\mathcal{S}) \geq \pi_2(\mathcal{S}) \geq \ldots \geq \pi_K(\mathcal{S})$ for all $\mathcal{S}$. Intuitively, π_1 charges penalties for subsets that are violated (at least) K times, π_2 charges penalties for subsets that are violated at least $K-1$ times, and so on. We assume that K-th violation costs at least as much as $(K-1)$-st violation, so $\pi_1(\mathcal{S}) \geq \pi_2(\mathcal{S})$ (and similarly for others).

3. **The penalty function is cross-submodular**. We assume that for $i<j$, $\pi_i(\mathcal{S}_i)+\pi_j(\mathcal{S}_j) \geq \pi_i(\mathcal{S}_i\cap\mathcal{S}_j)+\pi_j(\mathcal{S}_i\cup\mathcal{S}_j)$. This means that it is cheaper to have a larger set with larger indexed π function and smaller set with smaller indexed π function.

These restrictions on the $\bar{\pi}(\cdot)$ might seem restrictive, but we show below that an important variant of the prize-collecting survivable network design problem can be modelled in our framework.

A concrete problem The following problem can be cast in PCF-$\mathbb{Z}$ model considered above.

Let us consider an instance of PCSNDP problem in which each pair of vertices has a connectivity requirement of K. The profit function is defined as $\text{profit}(i) = \sqrt{iK}$ for $i = 0,1,\ldots,K$, which reflects how much profit the pair derives by getting i connections. Note that $\text{profit}(\cdot)$ is a concave function. The $\text{loss}(\cdot)$ is defined as negative of profit, but it is translated by K to make it a positive function. Thus, loss(i) $= K - \sqrt{(K-i)K}$. The loss function shows how much loss the pair i suffers if i of the K requirements are *not* satisfied. The penalty of the solution is defined to be aggregate loss of all pairs. Note that this imposes the natural condition that the difference in penalty for violating the very last unit of connectivity ($\text{loss}(1) - \text{loss}(0) = K - \sqrt{(K-1)K}$) is much less then violating the very first ($\text{loss}(K) - \text{loss}(K-1) = \sqrt{K}$).

This problem cannot be modelled in the PCSNDP framework mentioned in the first part of the paper because each disconnection carries a different penalty. But this problem can indeed be modelled in the PCF-$\mathbb{Z}$ framework of this section (proof omitted). In fact, there was nothing special about the square-root function above, any concave profit function (and hence convex loss function) can be modelled in our framework.

3.3 Algorithm

In this section, we present the algorithm for the PCF-$\mathbb{Z}$ problem defined in last few sections. The algorithm uses the algorithm for PCF from SSW [15]. The call to the PCF algorithm is denoted by $\text{PCF}(G(V,E,c), f, \pi)$, where c is the cost function on edges, f is the connectivity requirement on subsets, and π is the penalty function.

1. Decompose $\bar{\pi} = \pi_1 + \pi_2 + \cdots + \pi_K$, and run $\text{PCF}(G(V,E,c), \mathbf{1}, \pi_k)$ for $k = 1, 2, \ldots, K$ to obtain forests $F_1, F_2, \ldots, F_K$ and violated families $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_K$.
2. Construct a network $E' = F_1 + F_2 + \cdots + F_K$ (multiset addition) and output E' as the set of selected edges. Output $\bar{\mathcal{S}} = \mathcal{S}_1 + \mathcal{S}_2 + \cdots + \mathcal{S}_K$ as the multiset of violated subsets, on which the penalty is paid.

Fig. 1. PCF-$\mathbb{Z}$ algorithm for $(G(V,E,c), K, \bar{\pi})$. K: maximum connectivity requirement.

4 Analysis for PCF-$\mathbb{Z}$ algorithm

In this section, we prove that the solution found by Algorithm in Figure 1 is good approximation to the minimum cost network for PCF-$\mathbb{Z}$. Let $\bar{\mathcal{S}}$ be the multiset of violated subsets in $F_1 + F_2 + \cdots + F_K$, that is $\bar{\mathcal{S}} = \mathcal{S}_1 + \mathcal{S}_2 + \cdots + \mathcal{S}_K$. It follows from property 3 of the penalty function that

$$\bar{\pi}(\bar{\mathcal{S}}) = \bar{\pi}(\mathcal{S}_1 + \mathcal{S}_2 + \cdots + \mathcal{S}_K) \le \pi_1(\mathcal{S}_1) + \pi_2(\mathcal{S}_2) + \ldots + \pi_K(\mathcal{S}_K). \quad (1)$$

4.1 The performance guarantee

We are now ready to prove the performance guarantee for the algorithm. We will need to consider three linear programs. The first one is the linear program in [15] for the 0-1 connectivity requirements and with penalty function π_r. Note there are K such linear programs, one for each $r = 1, 2, \ldots, K$.

$$\text{Min} \quad \sum_{e \in E} c(e)x(e) + \sum_{\mathcal{S} \in 2^{2^V}} \pi_r(\mathcal{S})z(\mathcal{S}) \quad \text{subject to} \qquad (\text{LP1}(r))$$

$$\sum_{e\in\delta(S)} x(e) + \sum_{\mathcal{S}:S\in\mathcal{S}} z(\mathcal{S}) \geq 1 \qquad \forall S \subset V$$

$$x(e), z(\mathcal{S}) \geq 0 \qquad \forall e \in E, \mathcal{S} \in 2^{2^V}.$$

We defer discussion of the second LP for a moment. The third LP is the LP for the original problem PCF-$\mathbb{Z}$ with each connectivity requirement equal to K. Recall that we argued previously that this is equivalent to the original problem by modifying the penalty function.

$$\text{Min} \quad \sum_{e\in E} c(e)x(e) + \sum_{\bar{\mathcal{S}}\in[0\ldots K]^{2^V}} \bar{\pi}(\bar{\mathcal{S}})z(\bar{\mathcal{S}}) \quad \text{subject to} \tag{LP3}$$

$$\sum_{e\in\delta(S)} x(e) + \sum_{\bar{\mathcal{S}}:S\in\bar{\mathcal{S}}} z(\bar{\mathcal{S}}) \cdot n_{\bar{\mathcal{S}}}(S) \geq K \qquad \forall S \subset V$$

$$x(e), z(\bar{\mathcal{S}}) \geq 0 \qquad \forall e \in E, \bar{\mathcal{S}} \in [K+1]^{2^V}$$

$$0 \leq \sum_{\bar{\mathcal{S}}} z(\bar{\mathcal{S}}) \leq 1.$$

Let $\mathsf{OPT}_{\text{LP1}(r)}$ and $(x^*_{\text{LP1}(r)}, z^*_{\text{LP1}(r)})$ be the optimal value and the optimal solution for the linear program LP1(r), and $\mathsf{OPT}_{\text{LP3}}$ and $(x^*_{\text{LP3}}, z^*_{\text{LP3}})$ be the optimal value and the optimal solution for the linear program LP3. We have the following inequalities, which prove the performance guarantee of $\alpha \cdot \ln(K)$, given the α-approximation algorithm for PCF from [15].

$$\text{cost}(F_1 + F_2 + \cdots + F_K) = \textstyle\sum_{e\in F_1+F_2+\cdots+F_K} c(e) + \bar{\pi}(\bar{\mathcal{S}})$$

$$\leq \sum_{e\in F_1} c(e) + \cdots + \sum_{e\in F_K} c(e) + \pi_1(S_1) + \cdots + \pi_K(S_K) \quad \text{(From (1))}$$

$$= \textstyle\sum_{r=1}^K \left(\sum_{e\in F_r} c(e) + \pi_r(\mathcal{S}_r)\right)$$

$$\leq \alpha \cdot \sum_{r=1}^K \left(\sum_{e\in E} c(e) \cdot x^*_{\text{LP1}(r)}(e) + \sum_{\mathcal{S}\in 2^{2^V}} \pi_r(\mathcal{S}) \cdot z^*_{\text{LP1}(r)}(\mathcal{S}) \right) \quad \text{(From [15])}$$

$$\leq \alpha \cdot \sum_{r=1}^K \left(\frac{\mathsf{OPT}_{\text{LP3}}}{r} \right) \leq \alpha \cdot \mathsf{OPT}_{\text{LP3}} \cdot \ln(K). \quad \text{(From (2))}$$

We need to prove the following theorem to finish the proof.

Theorem 1. *For all $r = 1, 2, \ldots, K$,*

$$\sum_e c(e) \cdot x^*_{LP1(r)}(e) + \sum_{\mathcal{S}} \pi_r(\mathcal{S}) \cdot z^*_{LP1(r)}(\mathcal{S}) \leq \frac{\mathsf{OPT}_{LP3}}{r}. \tag{2}$$

4.2 Proof of Theorem 1

To prove this theorem we use a new linear program LP2(r) which requires a new truncated penalty function $\bar{\pi}^r$ which restricts the original penalty function $\bar{\pi}$ to the first r components in its decomposition (see property 1 of the penalty function). In other words for $\bar{\mathcal{S}} \in [0 \dots r]^{2^V}$, the new penalty function is defined as follows. Let $(\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_r) = \text{split}^r(\bar{\mathcal{S}})$ where $\text{split}^r(\cdot)$ is defined in the discussion of property 1 of the penalty function. Then $\bar{\pi}^r(\bar{\mathcal{S}}) = \pi_1(\mathcal{S}_1) + \pi_2(\mathcal{S}_2) + \ldots + \pi_r(\mathcal{S}_r)$. The linear program LP2(r) is the following; note that there is one linear program for each value of r.

$$\text{Min} \quad \sum_{e \in E} c(e)x(e) + \sum_{\bar{\mathcal{S}} \in [0 \dots r]^{2^V}} \bar{\pi}^r(\bar{\mathcal{S}}) z(\bar{\mathcal{S}}) \qquad \text{subject to} \qquad (\text{LP2}(r))$$

$$\sum_{e \in \delta(S)} x(e) + \sum_{\bar{\mathcal{S}} : S \in \bar{\mathcal{S}}} z(\bar{\mathcal{S}}) \cdot n_{\bar{\mathcal{S}}}(S) \geq r \qquad \forall S \subseteq V$$

$$x(e), z(\bar{\mathcal{S}}) \geq 0 \qquad \forall e \in E, \bar{\mathcal{S}} \in (r+1)^{2^V}$$

$$0 \leq \sum_{\bar{\mathcal{S}} \in (r+1)^{2^V}} z(\bar{\mathcal{S}}) \leq 1.$$

Here is the road-map of the proof. We first relate the optimum value of LP2(r) and LP3, and then of LP1(r) and LP2(r). Combining them will finish the proof.

Lemma 2. $\mathsf{OPT}_{LP2(r)} \leq \mathsf{OPT}_{LP3}$.

Lemma 3. $\mathsf{OPT}_{LP1(r)} \leq \mathsf{OPT}_{LP2(r)}/r$.

Proof. We construct a feasible solution $(x_{\text{LP1}(r)}, z_{\text{LP1}(r)})$ for LP1(r) from the optimal solution $(x^*_{\text{LP2}(r)}, z^*_{\text{LP2}(r)})$ of LP2(r) which costs no more than $\frac{\mathsf{OPT}_{\text{LP2}(r)}}{r}$.

The idea is to split the $z^*_{\text{LP2}(r)}(\bar{\mathcal{S}})$ value into r equal parts and give one part each to r families in $\text{split}^r(\bar{\mathcal{S}}) = (\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_r)$. Note that a family might end up getting z-contribution from many multisets, or even more than once from the same multiset. More formally, we define $(x_{\text{LP1}(r)}, z_{\text{LP1}(r)})$ as follows:

$$x_{\text{LP1}(r)}(e) = \frac{x^*_{\text{LP2}(r)}(e)}{r}, \qquad \text{and}$$

$$z_{\text{LP1}(r)}(\mathcal{S}) = \sum_{\bar{\mathcal{S}} \in (r+1)^{2^V} : \mathcal{S} \in \text{split}^r(\bar{\mathcal{S}})} \frac{z^*_{\text{LP2}(r)}(\bar{\mathcal{S}})}{r} \cdot n_{\text{split}^r(\bar{\mathcal{S}})}(\mathcal{S}).$$

Here $\text{split}^r(\bar{\mathcal{S}})$ is as defined in property (1) of the penalty function and $n_{\text{split}^r(\bar{\mathcal{S}})}(\mathcal{S})$ is the number of times family $\mathcal{S}$ occurs in (the ordered set) $\text{split}^r(\bar{\mathcal{S}})$.

Feasibility of constructed solution We first prove that the solution $(x_{\text{LP1}(r)}, z_{\text{LP1}(r)})$ constructed above is feasible for LP1(r).

The idea behind the proof is the following: we divide $z^*_{\text{LP2}(r)}(\bar{\mathcal{S}})$ into r parts and distribute it equally to r resulting families of $\text{split}^r(\bar{\mathcal{S}})$. In the original solution, $\bar{\mathcal{S}}$ was contributing $n_{\bar{\mathcal{S}}}(S) \cdot z^*_{\text{LP2}(r)}(\bar{\mathcal{S}})$ to the constraint of S, but in the new solution, $n_{\bar{\mathcal{S}}}(S)$ *different* families of subsets are contributing $z^*_{\text{LP2}(r)}(\bar{\mathcal{S}})/r$ to the constraint of S. Since the contribution to S from edges is also divided by r, the total contribution just gets divided by r. This proves the feasibility for an arbitrary subset S. A formal proof is omitted for space reasons.

Bounding the objective function The main idea in proving the bound on the objective function is that the z-value for a particular $\bar{\mathcal{S}}$ in the solution of LP2(r) is divided equally among r families (in $\text{split}^r(\bar{\mathcal{S}})$), whose penalties are evaluated by functions $\pi_1(\cdot)$, $\pi_2(\cdot)$, $\ldots$, $\pi_r(\cdot)$. Since $\pi_r(\cdot)$ is the least among them, if we evaluate all penalties at $\pi_r(\cdot)$, the cost only gets lower.

More formally, the objective function of the solution $(x_{\text{LP1}(r)}, z_{\text{LP1}(r)})$ for LP1(r) can be bounded in terms of the objective function value of optimal solution $(x^*_{\text{LP2}(r)}, z^*_{\text{LP2}(r)})$ of LP2(r) as follows:

$$\sum_{e\in E} c(e)x^*_{\text{LP2}(r)}(e) + \sum_{\bar{\mathcal{S}}\in[0\ldots K]^{2^V}} \bar{\pi}^r(\bar{\mathcal{S}})z^*_{\text{LP2}(r)}(\bar{\mathcal{S}})$$

$$= r\cdot\left[\sum_{e\in E} c(e)\frac{x^*_{\text{LP2}(r)}(e)}{r} + \sum_{\bar{\mathcal{S}}\in[0\ldots r]^{2^V}} (\pi_1(\mathcal{S}_1)+\cdots+\pi_r(\mathcal{S}_r))\frac{z^*_{\text{LP2}(r)}(\bar{\mathcal{S}})}{r}\right]$$

(here we denote $(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_r) = \text{split}^r(\bar{\mathcal{S}})$)

$$\geq r\cdot\left[\sum_{e\in E} c(e)\frac{x^*_{\text{LP2}(r)}(e)}{r} + \sum_{\bar{\mathcal{S}}\in[0\ldots r]^{2^V}} (\pi_r(\mathcal{S}_1)+\cdots+\pi_r(\mathcal{S}_r))\frac{z^*_{\text{LP2}(r)}(\bar{\mathcal{S}})}{r}\right]$$

$$= r\cdot\left[\sum_{e\in E} c(e)\frac{x^*_{\text{LP2}(r)}(e)}{r} + \sum_{\bar{\mathcal{S}}}\sum_{\mathcal{S}:\mathcal{S}\in\text{split}^r(\bar{\mathcal{S}})} \pi_r(\mathcal{S})\cdot n_{\text{split}^r(\bar{\mathcal{S}})}(\mathcal{S})\cdot\frac{z^*_{\text{LP2}(r)}(\bar{\mathcal{S}})}{r}\right]$$

$$= r\cdot\left[\sum_{e\in E} c(e)\frac{x^*_{\text{LP2}(r)}(e)}{r} + \sum_{\mathcal{S}}\pi_r(\mathcal{S})\sum_{\bar{\mathcal{S}}:\mathcal{S}\in\text{split}^r(\bar{\mathcal{S}})} n_{\text{split}^r(\bar{\mathcal{S}})}(\mathcal{S})\cdot\frac{z^*_{\text{LP2}(r)}(\bar{\mathcal{S}})}{r}\right]$$

(Changing the order of summation)

$$= r\cdot\left[\sum_{e\in E} c(e)x_{\text{LP1}(r)}(e) + \sum_{\mathcal{S}}\pi_r(\mathcal{S})z_{\text{LP1}(r)}(\mathcal{S})\right]$$

This shows that $\mathsf{OPT}_{\text{LP1}(r)} \leq \text{cost}_{\text{LP1}(r)}(x_{\text{LP1}(r)}, z_{\text{LP1}(r)}) \leq \frac{\mathsf{OPT}_{\text{LP2}(r)}}{r}$, proving the lemma.

5 Conclusions

One of the most important open problems is to design algorithms for the *all-or-nothing* version of penalty functions: penalty functions which charge the penalty even if the connectivity requirement is slightly violated. Other open problems include the following.

- Can we generalize the form of penalties as in the case of prize-collecting forest problem with general connectivity requirements? For example, penalties could be a submodular multi-set function of the set of disconnected pairs.
- We assume that the penalty function is decomposable in some simpler function which satisfy economy of scale conditions, but it would be nice to generalize it to submodular functions.

– Our algorithm for the prize-collecting forest problem needs to use each edge possibly many times (without bound). Can it be modified such that each edge is used a maximum number of times, which is a function of that edge?

References

[1] Ajit Agrawal, Philip N. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995.

[2] Egon Balas. The prize collecting traveling salesman problem. *Networks*, 19:621–636, 1989.

[3] Daniel Bienstock, Michel X. Goemans, David Simchi-Levi, and David P. Williamson. A note on the prize collecting traveling salesman problem. *Math. Programming*, 59:413–420, 1993.

[4] Markus Chimani, Maria Kandyba, and Petra Mutzel. A new ILP formulation for 2-root-connected prize-collecting Steiner networks. In Lars Arge, Michael Hoffmann, and Emo Welzl, editors, *ESA*, number 4698 in Lecture Notes in Computer Science, pages 681–692, 2007.

[5] J. Feigenbaum, C. H. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. *Journal of Computer and System Sciences*, 63:21–41, 2001.

[6] M. X. Goemans and D. Bertsimas. Survivable networks, linear programming relaxations and the parsimonious property. *Math. Program.*, 60:145–166, 1993.

[7] Michel Goemans, 1998. Personal communication.

[8] Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24(2):296–317, 1995.

[9] M.T. Hajiaghayi and K. Jain. The prize-collecting generalized Steiner tree problem via a new approach of primal-dual schema. In *SODA*, pages 631–640, 2006.

[10] Ara Hayrapetyan, Chaitanya Swamy, and Éva Tardos. Network design for information networks. In *SODA*, pages 933–942, 2005.

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

[12] David S. Johnson, Maria Minkoff, and Steven Phillips. The prize collecting Steiner tree problem: theory and practice. In *SODA*, pages 760–769, 2000.

[13] G. Kortsarz and Z. Nutov. Approximating minimum cost connectivity problems. In T. Gonzales, editor, *Handbook of Approximation Algorithms and Metaheuristics*. CRC Press, 2006.

[14] Retsef Levi, Andrea Lodi, and Maxim Sviridenko. Approximation algorithms for the multi-item capacitated lot-sizing problem via flow-cover inequalities. In Matteo Fischetti and David P. Williamson, editors, *Integer Programming and Combinatorial Optimization*, number 4513 in Lecture Notes in Computer Science, pages 454–468, 2007.

[15] Yogeshwer Sharma, Chaitanya Swamy, and David P. Williamson. Approximation algorithms for prize collecting forest problems with submodular penalty functions. In *SODA*, pages 1275–1284, 2007.

[16] David P. Williamson, Michel X. Goemans, Milena Mihail, and Vijay V. Vazirani. A primal-dual approximation algorithm for generalized steiner network problems. *Combinatorica*, 15(3):435–454, 1995.