# Prize-collecting Survivable Network Design in Node-weighted Graphs *

Chandra Chekuri, Alina Ene, and Ali Vakilian

Dept. of Computer Science, University of Illinois, Urbana, IL 61801, USA
{chekuri,ene1,vakilia2}@illinois.edu

**Abstract.** We consider node-weighted network design problems, in particular the survivable network design problem (SNDP) and its prize-collecting version (PC-SNDP). The input consists of a node-weighted undirected graph $G = (V, E)$ and integral connectivity requirements $r(st)$ for each pair of nodes $st$. The goal is to find a minimum node-weighted subgraph $H$ of $G$ such that, for each pair $st$, $H$ contains $r(st)$ edge-disjoint paths between $s$ and $t$. PC-SNDP is a generalization in which the input also includes a penalty $\pi(st)$ for each pair, and the goal is to find a subgraph $H$ to minimize the sum of the weight of $H$ and the sum of the penalties for all pairs whose connectivity requirements are not fully satisfied by $H$. Let $k = \max_{st} r(st)$ be the maximum requirement. There has been no non-trivial approximation for node-weighted PC-SNDP for $k > 1$, the main reason being the lack of an LP relaxation based approach for node-weighted SNDP. In this paper we describe multiroute-flow based relaxations for the two problems and obtain approximation algorithms for PC-SNDP through them. The approximation ratios we obtain for PC-SNDP are similar to those that were previously known for SNDP via combinatorial algorithms. Specifically, we obtain an $O(k^2 \log n)$-approximation in general graphs and an $O(k^2)$-approximation in graphs that exclude a fixed minor. The approximation ratios can be improved by a factor of $k$ but the running times of the algorithms depend polynomially on $n^k$.

## 1 Introduction

In this paper we consider the survivable network design problem (SNDP) and its prize-collecting version (PC-SNDP). In SNDP the input consists of an undirected graph $G = (V, E)$ and a connectivity requirement function specified in terms of an integer $r(st)$ for each unordered pair of nodes $st$. The goal is to find a minimum-weight subgraph $H$ of $G$ that contains $r(st)$ disjoint paths for each pair $st$. We use EC-SNDP and VC-SNDP to refer to the versions of SNDP depending on whether the desired paths are edge or node disjoint. In this paper we focus on EC-SNDP; for notational convenience we use SNDP when we mean EC-SNDP. A parameter of interest is the maximum requirement $k = \max_{st} r(st)$. Special cases of SNDP include well-studied problems such as the Steiner tree and Steiner forest problems (here $k = 1$). The weight of the chosen subgraph $H$ can depend both on the edges and nodes in $H$. In the edge-weighted version, each edge has a weight $w(e)$ and the weight of $H$ is the sum of the weights of the edges

* Partially supported by NSF grant CCF-1016684. A longer version containing the omitted proofs will be made available on the authors' webpages.

in $H$; Jain [14] obtained a 2-approximation for this problem via the influential iterated rounding technique that he introduced. The focus of this paper is the more general *node-weighted* case where each node $v$ has a weight $w(v)$; the weight of $H$ is the sum of the weights of the nodes in it[1]. The node-weighted version is provably harder to approximate. In contrast to the constant factor approximation for edge-weighted SNDP, the node-weighted Steiner tree problem is already $\Omega(\log n)$-hard to approximate via a simple reduction from the Set Cover problem [17].

Klein and Ravi [17] were the first to study node-weighted network design from an approximation point of view. They showed the hardness result mentioned above and described algorithms that achieved an $O(\log n)$-approximation for the Steiner tree and Steiner forest problems. Their algorithms are based on finding a structure called *spider*. Nutov examined the approximability of node-weighted SNDP [19] and obtained an $O(k \log n)$-approximation via the augmentation framework of Williamson et al. [21] (the connectivity requirements are met in $k$ stages with each stage increasing the connectivity of every unsatisfied pair by 1). His algorithm is based on a non-trivial structural result on spiders for covering an arbitrary 0-1 uncrossable requirement function[2]. Further, Nutov gave evidence, via a reduction from the $k$-densest subgraph problem, that a dependence on $k$ is necessary in the approximation ratio when $k$ is large. The algorithms of Klein and Ravi [17] and that of Nutov [19] are combinatorial. Mathematical programming relaxation based algorithms are powerful and flexible and it is natural to ask about their efficacy for node-weighted network design, and in particular for SNDP. Guha et al. [9] considered a natural LP relaxation for node-weighted Steiner tree and forest and showed that its integrality gap is $O(\log n)$, matching the bound obtained via the combinatorial algorithm; in fact, their proof uses a nice dual-fitting argument via spiders. In more recent work Demaine, Hajiaghayi, and Klein [8] demonstrated the advantage of the LP relaxation by describing a primal-dual algorithm that achieves an $O(1)$-approximation for node-weighted Steiner tree and forest when the underlying graph is planar.

In recent work [5] we generalized the work of Demaine et al. [8] and described an $O(k)$-approximation for node-weighted SNDP in planar graphs. A technical point of interest is that the algorithm is not based on a single LP relaxation. It uses the augmentation framework in which the connectivity requirements are incrementally satisfied in $k$ phases; a separate LP relaxation (**Augment-LP**) for each stage (that depends on the solution for the previous stages) is used[3].

This paper is motivated by two questions. Is there a natural LP relaxation for node-weighted SNDP? Is there a non-trivial approximation for node-weighted PC-SNDP?

[1] The version where both edges and nodes have weights can be easily reduced to the node-weighted version by sub-dividing each edge $e$ and placing a weight of $w(e)$ on the new node.

[2] A 0-1 set function $f : 2^V \rightarrow \{0, 1\}$ is said to be uncrossable if $f(A) = f(B) = 1$ implies that $f(A \cap B) = f(A \cup B) = 1$ or $f(A - B) = f(B - A) = 1$.

[3] There is some subtlety to understanding the integrality gap of **Augment-LP** since it only applies to a certain restricted class of uncrossable functions that arise from proper functions; in particular, each uncrossable function is a residual function of a node-induced subgraph of the original graph. This is in contrast to the edge-weighted case where there is a natural cut relaxation for covering an arbitrary uncrossable function whose integrality gap is at most 2. We refer the reader to Subsection 2.1 and [5] for more details.

We now give some background on prize-collecting network design problems and then discuss our results.

**Prize-collecting SNDP (PC-SNDP):** In PC-SNDP the input, in addition to that for SNDP, consists of penalties $\pi(st)$ for each pair of nodes. The goal is to find a subgraph $H$ of $G$ to minimize the weight of $H$ plus the sum of the penalties for pairs whose connectivity requirement is not satisfied by $H$; a pair $st$ is not satisfied if the number of disjoint paths in $H$ between $s$ and $t$ is strictly less than $r(st)$; this is the all-or-nothing penalty model and is the most interesting one from a technical point of view. The prize-collecting version of Steiner tree and Steiner forest have been studied extensively and have several theoretical and practical applications [15, 11, 20, 10]. A simple technique, introduced by Bienstock et al. [2], shows how one can use an LP relaxation based $\rho$-approximation algorithm for Steiner tree (and Steiner forest) to obtain an $O(\rho)$ approximation algorithm for the prize-collecting version. PC-SNDP for higher connectivity has been recently studied [18, 13, 12]. In [12] a technique similar to that of Bienstock et al. is used for edge-weighted SNDP (and also for Elem-SNDP and VC-SNDP). However, [12] shows that a straightforward and natural LP relaxation has a large integrality gap, and introduce a stronger LP relaxation. In this paper we are concerned with node-weighted PC-SNDP. For node-weighted Steiner tree and Steiner forest there is a natural LP relaxation with $O(\log n)$ integrality gap (and $O(1)$ gap for planar graphs), and one can use this to obtain a corresponding approximation for the prize-collecting version. However, as we already remarked, the algorithms for node-weighted SNDP for $k > 1$ have not been based on a single LP relaxation.

**Our Contribution:** Our first contribution is to formulate an LP relaxation for node-weighted SNDP and PC-SNDP via *multi-route flows* [16, 1]. We give two relaxations, one for arbitrary $k$ and a different relaxation that is more suited for fixed $k$. The multi-route flow based relaxation easily allows us to apply the basic idea of Bienstock et al. [2] to reduce the PC-SNDP problem to the SNDP problem. Our second contribution is to analyze the integrality gap of these relaxations for node-weighted SNDP. We obtain an upper bound on the integrality gap by relating the optimum value of the relaxation to that of the **Augment-LP** relaxation [5] in each phase of the augmentation framework. For planar graphs we can use the result from [5] that showed that the integrality gap of the **Augment-LP** is $O(1)$. In this paper we show that **Augment-LP** has an integrality gap of $O(\log n)$ for general graphs. These ingredients give us the following theorem that summarizes our results.

**Theorem 1.** *There is an $O(k^2 \log n)$-approximation for node-weighted PC-SNDP in undirected graphs which improves to an $O(k^2)$-approximation for planar graphs. There is an algorithm with running time that is polynomial in $n^k$ that achieves an $O(k \log n)$ approximation for general graphs and an $O(k)$ approximation for planar graphs.*

**Discussion, Related Work and Extensions:** We start with the question as to why it is non-trivial to find a natural LP relaxation for the node-weighted SNDP problem. Consider the problem where the requirement is only for a single pair $st$; that is, we wish to find a minimum weight subgraph that has $k$ edge-disjoint paths from $s$ to $t$. If the weights are on the edges then this problem can be solved easily via min-cost flow. However, if the weights are on the nodes the edge-disjoint paths from $s$ to $t$ may use

a node $v$ multiple times, yet the weight of the node $v$ counts only once. (This is the same issue that is also present in the *capacitated* SNDP (Cap-SNDP) problem [3, 4].) The inability to solve the single pair problem exactly is at the heart of the difficulty of finding a relaxation for node-weighted SNDP. We write a multi-route flow based LP that we cannot solve in polynomial time because the separation oracle for the dual requires us to solve the single pair problem. However, this relaxation can be solved approximately within a factor of $k$. This is the reason that our approximation ratios depend on $k^2$, one factor of $k$ from approximating the relaxation, and another factor of $k$ from the augmentation framework. We write a different relaxation that can be solved in time that is polynomial in $n^k$. This relaxation is inspired by the formulation of the **Augment-LP** and allows us to improve the approximation when $k$ is a fixed constant. Multi-route flows and cuts are useful concepts when considering higher connectivity. Their applications and properties are not as widely known as they could be, and we hope our work helps highlight their usefulness.

One can also consider the node-weighted versions of element-connectivity SNDP (Elem-SNDP) and vertex-connectivity SNDP (VC-SNDP). Multi-route flow based relaxations can be written in the same fashion. The same difficulty present in EC-SNDP for solving the single-pair node-weighted problem extends to the Elem-SNDP problem. Interestingly, it is easy to write a multi-route LP relaxation for VC-SNDP and solve it in polynomial time! The reason for this is that in VC-SNDP the paths are required to be node-disjoint and hence the capacitated aspect goes away. However, the only non-trivial algorithmic technique for VC-SNDP at this point is via a (randomized) reduction from VC-SNDP to Elem-SNDP [7]. We believe that our algorithms and analysis will extend from EC-SNDP to Elem-SNDP as well and hence indirectly also to VC-SNDP.

The multi-route flow based LP relaxations can be solved in polynomial time for edge-weighted problems. For the prize-collecting version the relaxation is in fact equivalent to that in the work of Hajiaghayi et al. [12]; the multi-route flow view makes the cut-based relaxation in [12] easier to understand. Previous work on prize-collecting SNDP has considered submodular penalty functions [20, 12]; here the penalty for not connecting a set of pairs is a monotone submodular function of those pairs. It is easy to extend our algorithms and analysis to this more general case by simply replacing the linear penalty in the objective function of the relaxation by a Lovász-extension based convex penalty function; this is in the same fashion as in the work of Chudak and Nagano [6]. We omit the details in this version of the paper.

Finally, the advantage of having an LP relaxation based algorithm (for node-weighted SNDP) is the flexibility it affords in incorporating additional constraints and solving related problems. For instance, problems such as $k$-MST can be solved via relaxations for the Steiner tree. Guha et al. [9] studied an LP relaxation approach for node-weighted Steiner tree motivated by such considerations. Similar applications can now be derived for higher connectivity.

**Organization:** The rest of the paper is organized as follows. Section 2 discusses the multi-route flow based relaxations and relates their integrality gap to that of **Augment-LP**. In Section 3 we bound the integrality gap of **Augment-LP** by $O(\log n)$ for general graphs.

## 2 LP Relaxations for node-weighted PC-SNDP

Let $s$ and $t$ be two vertices of the graph and let $\ell$ be an integer. Consider a tuple $\overline{p} = (p_1, p_2, \cdots, p_\ell)$ such that each $p_i$ is a path from $s$ to $t$ and the paths in $\overline{p}$ are edge-disjoint; we refer to such a tuple $\overline{p}$ as an $\ell$-route tuple connecting $s$ to $t$. In the following, we ignore the order in which the paths appear in the tuple; more precisely, two tuples consisting of the same collection of paths are considered to be the same tuple. A vertex $v$ intersects $\overline{p}$ if there exists *some* path in $\overline{p}$ that contains $v$; we use $v \in \overline{p}$ to denote the fact that $v$ intersects $\overline{p}$. Similarly, an edge $e$ intersects $\overline{p}$ if there exists some path in $\overline{p}$ that contains $e$; we use $e \in \overline{p}$ to denote the fact that $e$ intersects $\overline{p}$.

Consider an instance of the node-weighted PC-SNDP problem. For each unordered pair $st$ of nodes, we let $\mathcal{P}_{st}^{r(st)}$ denote the collection of all $r(st)$-tuples that connect $s$ to $t$, where $r(st)$ is the requirement of the pair. We can write a relaxation for the problem as follows. We have a variable $x(v)$ for each vertex $v$ and a variable $z(st)$ for each pair $st$ of nodes with the interpretation that $x(v) = 1$ if $v$ is in the solution and $z(st) = 1$ if the requirement of $st$ is *not* satisfied by the solution. We also have variables $f(\overline{p})$, where $\overline{p} \in \mathcal{P}_{st}^{r(st)}$, with the interpretation that $f(\overline{p}) = 1$ if the paths connecting $s$ to $t$ are the paths of $\overline{p}$.

**PC-Multiroute-LP**

$$\min \sum_{v \in V} w(v)x(v) + \sum_{st \in V \times V} \pi(st)z(st)$$

$$\text{s.t.} \sum_{\overline{p} \in \mathcal{P}_{st}^{r(st)}} f(\overline{p}) = 1 - z(st) \quad \forall st$$

$$\sum_{\overline{p} \in \mathcal{P}_{st}^{r(st)},\, v \in \overline{p}} f(\overline{p}) \leq x(v) \quad \forall v, \forall st$$

$$0 \leq x(v) \leq 1 \quad \forall v$$

$$0 \leq z(st) \leq 1 \quad \forall st$$

$$f(\overline{p}) \geq 0 \quad \forall \overline{p}$$

**Multiroute-LP**

$$\min \sum_{v \in V} w(v)x(v)$$

$$\text{s.t.} \sum_{\overline{p} \in \mathcal{P}_{st}^{r(st)}} f(\overline{p}) = 1 \quad \forall st$$

$$\sum_{\overline{p} \in \mathcal{P}_{st}^{r(st)},\, v \in \overline{p}} f(\overline{p}) \leq x(v) \quad \forall v, \forall st$$

$$0 \leq x(v) \leq 1 \quad \forall v$$

$$f(\overline{p}) \geq 0 \quad \forall \overline{p}$$

**Proposition 1.** ***PC-Multiroute-LP*** *is a valid relaxation for the node-weighted PC-SNDP problem. Moreover if there is a single pair $st$ with non-zero requirement then the relaxation is exact.*

We summarize at a high-level our theorems about **PC-Multiroute-LP** and **Multiroute-LP** below.

- Given a feasible solution $(x, f, z)$ to **PC-Multiroute-LP** it is easy to obtain another feasible solution $(x', f', z')$, via the scaling trick of Bienstock et al. [2], such that $z'$ is integral and the cost of $(x', f', z')$ is at most 2 times the cost of $(x, f, z)$.
- The integrality gap of **Multiroute-LP** is $O(k \log n)$ for general graphs and $O(k)$ for graphs from a minor-closed family of graphs.

– **PC-Multiroute-LP** and **Multiroute-LP** are **NP**-hard to solve when $k$ is part of the input. However, one can find in polynomial time a feasible solution to them with cost at most $k$ times the optimum solution value. This is done by solving a compact relaxation. Combining the above three ingredients gives an $O(k^2 \log n)$ approximation for node-weighted PC-SNDP and the ratio improves to $O(k^2)$ for minor-closed families of graphs.
– There is a different relaxation that leads to an improvement in the approximation ratio for PC-SNDP to $O(k \log n)$ in general graphs and to $O(k)$ in minor-closed families of graphs respectively. The running time is, however, polynomial in $n^k$.

*Remark 1.* For edge-weighted problems the multi-route formulation will have a variable $x(e)$ for each edge and the total multi-route flow on each edge $e$ for any pair will be bounded by $x(e)$. This relaxation can be solved in polynomial time since the separation oracle for the dual is the min-cost flow problem. This relaxation for PC-SNDP is equivalent (in the sense of having the same optimal value for each instance) to the cut-based relaxation from [12].

We sketch the rounding step in the first item above that reduces the PC-SNDP problem to the SNDP problem, since it demonstrates the naturalness of the multi-route LP for higher connectivity. Let $(x, f, z)$ be a feasible fractional solution to **PC-Multiroute-LP**. Let $I = \{st \mid z(st) > 1/2\}$. Consider the SNDP instance that we get from the prize-collecting instance by setting the requirements of all the pairs in $I$ to zero. Let $J$ be the set of all pairs not in $I$. Let $x'$ and $f'$ be the following vectors. For each vertex $v \in V$, we set $x'(v) = \min\{1, 2x(v)\}$. For each pair $st \in J$ and each $\overline{p} \in \mathcal{P}_{st}^{r(st)}$, we set $f'(\overline{p}) = f(\overline{p})/(1 - z(st))$. (Note that, for each $st \in J$, $z(st) \leq 1/2$.) It is straightforward to show that $(x', f')$ is a feasible solution to **Multiroute-LP** for the pairs in $J$. Further, the penalty incurred for pairs in $I$ is at most twice the penalty that the fractional solution $(x, f, z)$ already paid for them. The factor of 2 loss here can be improved slightly via an idea of Goemans as was done in prior work, but we omit the improvement in this version.

In Subsection 2.1 we show an upper bound on the integrality gap of **Multiroute-LP** via the augmentation framework and **Augment-LP** from [5].

**A different relaxation.** Consider a solution $H$ that satisfies the requirement of the pair $st$. If we remove less than $r(st)$ of the edges of $H$ then there will be at least one path from $s$ to $t$ in the resulting graph. With this observation in mind, we can write an LP relaxation as follows. As before, we have a variable $x(v)$ for each vertex $v$ and a variable $z(st)$ for each pair $st$. We introduce the following constraints for each pair $st$ and each set $F \subseteq E$ such that $|F| < r(st)$. Consider the network $G_F = (V, E - F)$ with node capacities given by the values $x(v)$. We impose the valid constraint that the network $G_F$ supports at least $1 - z(st)$ units of flow from $s$ to $t$ subject to the node capacity constraints given by $x$. The resulting LP has $O(|E|^k)$ constraints and can be solved in time that is polynomial in $n^k$. When $k$ is a fixed constant, this relaxation leads to an improvement in the approximation ratio.

We refer to this LP as **PC-Cut-LP** and to its non-prize-collecting counterpart as **Cut-LP**. We note that **Multiroute-LP** is strictly stronger than **Cut-LP**; on instances of the problem in which there is a single requirement pair with requirement $k$, **Multiroute-LP**

is exact whereas **Cut-LP** has an $\Omega(k)$ integrality gap. Nevertheless, we can show that the integrality gap of **Cut-LP** is $O(k \log n)$ for general graphs and $O(k)$ for graphs from a minor-closed family. The approach for upper bounding the integrality gap of **PC-Cut-LP** and **Cut-LP** is very similar to the approach described in Subsection 2.1 for upper bounding the integrality gap of **PC-Multiroute-LP** and **Multiroute-LP**.

### 2.1 Integrality gap of Multiroute-LP via Augment-LP

In this section, we show that the integrality gap of **Multiroute-LP** is $O(k \log n)$ for general graphs and $O(k)$ for minor-closed families of graphs.

**Theorem 2.** *Let* OPT *be the value of the optimal fractional solution to* ***Multiroute-LP****. There is a polynomial time algorithm that constructs a subgraph $H$ of $G$ such that $H$ is a feasible solution for the node-weighted SNDP instance and the weight of $H$ is $O(k \log n) \cdot$ OPT.*

**Theorem 3.** *Let* OPT *be the value of the optimal fractional solution to* ***Multiroute-LP****. If the input graph $G$ belongs to a minor-closed family $\mathcal{G}$, there is a polynomial time algorithm that constructs a subgraph $H$ of $G$ such that $H$ is a feasible solution for the node-weighted SNDP instance and the weight of $H$ is $O(k) \cdot$ OPT, where the constant depends only on the family $\mathcal{G}$.*

In order to prove Theorem 2 and Theorem 3, we use the augmentation framework that was introduced by Williamson et al. [21] for the edge-weighted SNDP problem. Note that the theorems only upper bound the integrality gap of the relaxations; the algorithms for SNDP are not based on solving them. The relaxations need to be solved for PC-SNDP to identify the pairs to connect and reduce to SNDP.

We start by introducing some notation. A set $S$ *separates* a pair $st$ iff $S$ contains exactly one of $s, t$. Let $r : 2^V \to \mathbb{Z}_+$ be the function such that $r(S)$ is the maximum requirement of a pair separated by $S$. Let $r_\ell : 2^V \to \mathbb{Z}_+$ be the function such that $r_\ell(S) = \min\{r(S), \ell\}$ for all sets $S \subseteq V$. Let $\delta_H(S)$ be the set of all edges of $H$ with an endpoint in $S$ and the other in $V - S$ (note that $H$ may not contain all the vertices of $S$). A graph $H$ covers $r$ iff $|\delta_H(S)| \geq r(S)$ for all sets $S$. By Menger's theorem, a graph $H$ is a feasible solution to the SNDP instance iff $H$ covers $r$.

The algorithm selects a cover $H$ of $r$ in $k$ phases. The algorithm maintains the invariant that the first $\ell$ phases have selected a graph $H_\ell$ that covers $r_\ell$. During phase $\ell$, the algorithm adds a new set of nodes to $H_{\ell-1}$ in order to get a graph $H_\ell$ that covers $r_\ell$. More precisely, in phase $\ell$, we solve the following augmentation problem. It is convenient to assume that all the nodes in $H_{\ell-1}$ have weight zero; since we have already paid for the nodes, we can set their weight to zero at the beginning of phase $\ell$. Let $h_\ell : 2^V \to \{0, 1\}$ be the function such that $h_\ell(S) = 1$ iff $|\delta_{H_{\ell-1}}(S)| = \ell - 1$ and $r(S) \geq \ell$. Let $G'_\ell = (V, E - E(H_{\ell-1}))$. The goal is to select a minimum weight subgraph $K_\ell$ of $G'_\ell$ that covers $h_\ell$; once we have $K_\ell$, we let $H_\ell$ be the subgraph of $G$ induced by $V(H_{\ell-1}) \cup V(K_\ell)$.

In the following, we show that, in each phase $\ell$, we can select a subgraph $K_\ell$ that covers $h_\ell$ such that the node weight of $K_\ell$ is at most $O(\log n) \cdot$ OPT for general graphs

and $O(1) \cdot$ OPT for minor-closed families of graphs, where OPT is the value of the optimal solution to **Multiroute-LP**. It will then follow that the algorithm described above constructs a subgraph $H$ such that $H$ covers $r$ and the weight of $H$ is $O(k \log n) \cdot$ OPT for general graphs and $O(k) \cdot$ OPT for minor-closed families of graphs.

Consider a phase $\ell$. Recall that the goal is to cover $h_\ell$ using a subgraph of $G'_\ell$. Let $\Gamma_{G'_\ell}(S)$ be the vertex neighborhood of $S$; that is, the set of vertices $v \in V - S$ such that there is an edge $uv \in E(G'_\ell)$, where $u \in S$. We have the following relaxation for the augmentation problem of phase $\ell$.

**Augment-LP**$(G'_\ell, h_\ell)$

$$\min \sum_{v \in V} w(v)x(v)$$

$$\text{s.t.} \sum_{v \in \Gamma_{G'_\ell}(S)} x(v) \geq h_\ell(S) \quad \forall S \subseteq V$$

$$x(v) \geq 0 \quad \forall v \in V$$

As shown in Lemma 1, for each phase of the algorithm, the optimal value of **Augment-LP** is at most the optimal value of **Multiroute-LP**.

**Lemma 1.** *Let $(x, f)$ be a feasible solution to **Multiroute-LP**. For any phase $\ell$, $x$ is a feasible solution to **Augment-LP**$(G'_\ell, h_\ell)$.*

**Corollary 1.** *Let $\rho$ be such that, for each phase $\ell$, the integrality gap of **Augment-LP**$(G'_\ell, h_\ell)$ is at most $\rho$. Then the integrality gap of **Multiroute-LP** is at most $k\rho$.*

Therefore it suffices to upper bound the integrality gap of **Augment-LP**. We prove Theorem 4 in Section 3. Theorem 5 was shown in [5].

**Theorem 4.** *For each $\ell$, the integrality gap of **Augment-LP**$(G'_\ell, h_\ell)$ is $O(\log n)$. Moreover, there is a polynomial time algorithm that selects a subgraph $K_\ell$ of $G'_\ell$ such that $K_\ell$ covers $h_\ell$ and the weight of $K_\ell$ is at most $O(\log n)$ times the weight of the optimal fractional solution to **Augment-LP**$(G'_\ell, h_\ell)$.*

**Theorem 5 ([5]).** *Suppose that $G$ belongs to a minor-closed family $\mathcal{G}$. For each $\ell$, the integrality gap of **Augment-LP**$(G'_\ell, h_\ell)$ is a constant that depends only on the family $\mathcal{G}$. Moreover, there is a polynomial time algorithm that selects a subgraph $K_\ell$ of $G'_\ell$ such that $K_\ell$ covers $h_\ell$ and the weight of $K_\ell$ is at most $O(1)$ times the weight of the optimal fractional solution to **Augment-LP**$(G'_\ell, h_\ell)$.*

*Remark 2.* The integrality gap of **Augment-LP** is unbounded when the function $h_\ell$ is an arbitrary uncrossable function. However, the functions $h_\ell$ that arise from instances of the node-weighted SNDP problem via the augmentation framework have additional properties that are exploited by Theorem 2 and Theorem 3. We refer the reader to [5] for more details.

Theorem 2 and Theorem 3 follow from Corollary 1 and Theorem 4 and Theorem 5.

## 3 Integrality gap of Augment-LP

In this section, we prove Theorem 4 that upper bounds the integrality gap of **Augment-LP** in general graphs. We refer the reader to Subsection 2.1 for the relevant definitions and notation.

In order to simplify notation, we let $G' = G'_\ell$ and $h = h_\ell$; our goal is to select a minimum-weight subgraph $K$ of $G'$ that covers $h$. As we have already seen in Subsection 2.1, we have the following LP relaxation for this problem.

**Augment-LP**$(G', h)$

$$\min \sum_{v \in V} w(v)x(v)$$

$$\text{s.t.} \sum_{v \in \Gamma_{G'}(S)} x(v) \geq h(S) \quad \forall S \subseteq V$$

$$x(v) \geq 0 \quad \forall v \in V$$

Dual of **Augment-LP**$(G', h)$

$$\max \sum_{S \subseteq V} y(S)h(S)$$

$$\text{s.t.} \sum_{S: v \in \Gamma_{G'}(S)} y(S) \leq w(v) \quad \forall v \in V$$

$$y(S) \geq 0 \quad \forall S \subseteq V$$

Our proof uses the concept of a (generalized) *spider* that was introduced by Nutov [19] which we will define shortly. While Nutov uses a combinatorial algorithm to find a spider we find one via a primal-dual algorithm and relate its density to that of the LP relaxation. We start with some notation and some definitions that are based on [19, 21].

**Preliminaries.** Recall that we are working with a 0-1 uncrossable function $h : 2^V \to \{0, 1\}$. We can also view $h$ as a family consisting of all sets $S$ such that $h(S) = 1$. Following Nutov, we let $\mathcal{F} = \{S \mid h(S) = 1\}$ be the family corresponding to $h$. We refer to each set in $\mathcal{F}$ as a **violated set** and we refer to the inclusion-wise minimal sets of $\mathcal{F}$ as **min-cores**. Let $\mathcal{C}$ be the set of all min-cores of $\mathcal{F}$. The sets in $\mathcal{C}$ are disjoint and we can compute the collection $\mathcal{C}$ in polynomial time for the function $h$ that arises in SNDP [21]. Additionally, if $S$ is a violated set and $C$ is a min-core, either $C$ is contained in $S$ or $C$ and $S$ are disjoint.

A set $S \in \mathcal{F}$ is a **core** of $\mathcal{F}$ iff $S$ contains exactly one min-core $C$; we refer to a core $S$ that contains the min-core $C \in \mathcal{C}$ as a $C$-core. Let $\mathcal{A} \subseteq \mathcal{C}$ and let $u$ be a vertex. Let $\mathcal{S}(\mathcal{A}, u) \subseteq \mathcal{F}$ be the family consisting of all sets $S \in \mathcal{F}$ such that $S$ is an $A$-core for some $A \in \mathcal{A}$ and $u \notin S$. We refer to the family $\mathcal{S}(\mathcal{A}, u)$ as a **spider family**. We refer to the min-cores in $\mathcal{A}$ as the **feet** of $\mathcal{S}(\mathcal{A}, u)$ and we refer to $u$ as the **center** of $\mathcal{S}(\mathcal{A}, u)$. A set $F \subseteq E(G')$ of edges covers a family $\mathcal{F}'$ of sets iff, for each set $S \in \mathcal{F}'$, there is at least one edge of $F$ leaving $S$; more precisely, we have $|\delta_F(S)| \geq 1$ for each set $S \in \mathcal{F}'$. If $\mathcal{F}'$ is a spider family, we refer to $F$ as a **spider cover**. Nutov [19] introduced the notions of spider families and covers as a generalization to the concept of spiders that play an important role in the algorithm of Klein and Ravi [17] for the node-weighted Steiner tree problem; we refer the reader to [19] for more details. We remark that there are subtleties when thinking about spiders for uncrossable functions since a spider cover $F$ can be disconnected.

**The algorithm for covering $\mathcal{F}$.** Nutov extended the algorithm of Klein and Ravi to the problem of covering an uncrossable family $\mathcal{F}$ as follows. We find a spider family

$\mathcal{S}(\mathcal{A}, u)$ and a cover $F$ of $\mathcal{S}(\mathcal{A}, u)$. Let $\mathcal{F}' = \{S \mid S \in \mathcal{F}, \delta_F(S) = \emptyset\}$ be the subfamily of $\mathcal{F}$ that is not covered by $F$; the residual family $\mathcal{F}'$ is uncrossable as well. Let $G'' = (V, E(G') - F)$. We recursively construct a cover $F' \subseteq E(G'')$ for $\mathcal{F}'$ and we return $F \cup F'$ as our cover of $\mathcal{F}$.

Nutov gave a polynomial time algorithm to find a spider cover whose weight (in terms of nodes) is "comparable" to the weight of the optimal *integral* solution; here the comparison is in the sense of density which is the weight divided by the number of min-cores that are removed by the addition of the cover. We show that we can find a spider cover whose weight is "comparable" to the weight of the optimal *fractional* solution for **Augment-LP**$(G', h)$. More precisely, we show the following theorem.

**Theorem 6.** *There is a spider family $\mathcal{S}(\mathcal{A}, u)$ of $\mathcal{F}$ and a cover $F$ of $\mathcal{S}(\mathcal{A}, u)$ with the following properties. Let $\mathcal{F}' = \{S \mid S \in \mathcal{F}, \delta_F(S) = \emptyset\}$ be the subfamily of $\mathcal{F}$ that is not covered by $F$, and let $\mathcal{C}'$ be the collection of all minimal sets of $\mathcal{F}'$. We have $|\mathcal{C}'| < |\mathcal{C}|$ and $w(V(F))$ (total weight of the nodes in $F$) is $O\big((|\mathcal{C}| - |\mathcal{C}'|)/|\mathcal{C}|\big)$ times the value of the optimal fractional solution to **Augment-LP**$(G', h)$. Moreover, we can find the feet $\mathcal{A}$, the center $u$, and the cover $F$ of $\mathcal{S}(\mathcal{A}, u)$ in polynomial time.*

Once we have Theorem 6, we can find a cover of $h$ using a greedy algorithm. If the collection $\mathcal{C}$ of all minimal violated components is empty, we return an empty cover. Otherwise, let $\mathcal{S}(A, u)$ and $F$ be the spider family and spider cover guaranteed by Theorem 6. Let $H'$ and $h'$ be as in the statement of Theorem 6, and let $G'' = (V, E - E(H'))$. We recursively find a cover $F'$ of $h'$ and we return $F \cup F'$.

It is straightforward to verify that the weight of the optimal fractional solution to **Augment-LP**$(G'', h')$ is at most the weight of the optimal fractional solution to **Augment-LP**$(G', h)$. This observation together with a standard set cover analysis gives us that the total weight of the cover constructed by the algorithm above is $O(\log |\mathcal{C}|)$ times the weight of the optimal fractional solution to **Augment-LP**$(G', h)$.

Therefore, in order to complete the proof of Theorem 4, it suffices to prove Theorem 6. In the following, we give the algorithm for constructing the spider family $\mathcal{S}(\mathcal{A}, u)$.

**Primal-dual algorithm for constructing the spider family.** Consider the dual of the **Augment-LP**$(G', h)$ (see above). The algorithm selects a set $X \subseteq V(G')$ of nodes as follows. The algorithm also maintains a solution $y$ that is feasible for the dual of **Augment-LP**$(G', h)$; the solution $y$ is implicitly initialized to zero.

We proceed in iterations. Consider iteration $i$ and let $X_{i-1}$ be the nodes selected in the first $i - 1$ iterations; $X_0$ is the set of all zero-weight nodes. A set $S$ is violated with respect to a set $Z$ of nodes iff $h(S) = 1$ and $\delta_{G'[Z]}(S)$ is empty. Recall that $\mathcal{C}$ is the collection of all minimal violated components of $h$; note that $\mathcal{C}$ is also the collection of all minimal sets that are violated with respect to $X_0$. Let $\mathcal{C}_{i-1}$ be the collection of minimal violated sets with respect to $X_{i-1}$. For each component $C \in \mathcal{C}_{i-1}$, we have $C \subseteq X_{i-1}$ [5]. Since the components of $\mathcal{C}_{i-1}$ are disjoint and two components $C \in \mathcal{C}$ and $C' \in \mathcal{C}_{i-1}$ do not properly intersect, we have $|\mathcal{C}_{i-1}| \leq |\mathcal{C}|$. If $|\mathcal{C}_{i-1}|$ is strictly less than $|\mathcal{C}|$, we return the set $X = X_{i-1}$ and the dual solution $y$, and we terminate the algorithm. In other words we stop the algorithm when at least two of the min-cores in $\mathcal{C}$ "merge" and are part of the same minimal violated set of $\mathcal{C}_{i-1}$. Otherwise, we

increase the dual variables $\{y(C)\}_{C \in \mathcal{C}_{i-1}}$ uniformly until a dual constraint for a vertex becomes tight. (Note that it is possible that the increase was zero if there was already a tight vertex at the beginning of the iteration; any vertex that was already tight is not in $X_{i-1}$.) Let $v$ be a vertex that became tight; if there are several such vertices, we pick one of them arbitrarily. We add $v$ to $X$ and we proceed to the next iteration (note that we have $X_i = X_{i-1} \cup \{v\}$).

Let $X$ be the set of nodes selected by the algorithm. Let $i^*$ denote the last iteration of the algorithm which adds a node. Let $\hat{\mathcal{C}} = \cup_{i \leq i^*} \mathcal{C}_{i-1}$ be the collection of all sets that were minimal violated sets throughout the history of the primal-dual algorithm before merging happens at the end of iteration $i^*$. Let $u$ be the node that was added to $X$ in iteration $i^*$. Intuitively, the addition of $u$ merged some of the cores. We formally identify the min-cores associated with the merged cores as follows. Let $\mathcal{A} = \{C \in \mathcal{C} \mid \text{there is } D \in \mathcal{C}_{i^*-1} \text{ such that } C \subseteq D \text{ and } u \in \Gamma_{G'}(D)\}$. The family $\mathcal{S}(\mathcal{A}, u)$ is the desired spider family.

Finally, we perform the following *reverse-delete* step on the set $X$ of nodes in order to identify a subset of nodes that cover $\mathcal{S}(\mathcal{A}, u)$. We let $Y_C$ be the set of all nodes in $X - (X_0 \cup \{u\})$ that are adjacent to some $C$-core in $\hat{\mathcal{C}}$. The sets $\{Y_C\}_{C \in \mathcal{C}}$ are disjoint and their union is $X - (X_0 \cup \{u\})$. We consider each foot $A \in \mathcal{A}$ separately. An important observation is that $G'[Y_A \cup X_0 \cup \{u\}]$ covers $\mathcal{S}(\{A\}, u)$. For each foot $A$, we select a set $Z_A \subseteq Y_A$ such that $G'[Z_A \cup X_0 \cup \{u\}]$ covers $\mathcal{S}(\{A\}, u)$ as follows. We start with $Z_A = Y_A$. We consider the nodes of $Z_A$ in the *reverse* of the order in which they were added to $X$. Let $v$ be the current node. If the graph $G'[(Z_A \cup X_0 \cup \{u\}) - \{v\}]$ covers the spider family $\mathcal{S}(\{A\}, u)$, we remove $v$ from $Z_A$. We set $Z = \cup_{A \in \mathcal{A}} Z_A$ and we output the family $\mathcal{S}(\mathcal{A}, u)$ and the cover $G'[Z \cup X_0 \cup \{u\}]$.

The spider family $\mathcal{S}(\mathcal{A}, u)$ and the cover $G'[Z \cup X_0 \cup \{u\}]$ have the properties required by Theorem 6; we defer the proof to a longer version of this paper.

## References


1. C.C. Aggarwal and J.B. Orlin. On multiroute maximum flows in networks. *Networks*, 39(1):43–52, 2002.
2. D. Bienstock, M.X. Goemans, D. Simchi-Levi, and D. Williamson. A note on the prize collecting traveling salesman problem. *Mathematical programming*, 59(1):413–420, 1993.
3. R.D. Carr, L.K. Fleischer, V.J. Leung, and C.A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. *Proc. of ACM-SIAM SODA*, pages 106–115, 2000.
4. D. Chakrabarty, C. Chekuri, S. Khanna, and N. Korula. Approximability of capacitated network design. *Proc. of IPCO*, pages 78–91, 2011.
5. C. Chekuri, A. Ene, and A. Vakilian. Node-weighted network design in planar and minor-closed families of graph. *Proc. of ICALP*, 2012.
6. F.A. Chudak and K. Nagano. Efficient solutions to relaxations of combinatorial problems with submodular penalties via the Lovász extension and non-smooth convex optimization. *Proc. of ACM-SIAM SODA*, pages 79–88, 2007.
7. J. Chuzhoy and S. Khanna. An $O(k^3 \log n)$-approximation algorithm for vertex-connectivity survivable network design. *Proc. of IEEE FOCS*, pages 437–441, 2009.
8. E. Demaine, M.T. Hajiaghayi, and P. Klein. Node-weighted Steiner tree and group Steiner tree in planar graphs. *Proc. of ICALP*, pages 328–340, 2009.

9. S. Guha, A. Moss, J.S. Naor, and B. Schieber. Efficient recovery from power outage. *Proc. of ACM STOC*, pages 574–582, 1999.
10. S. Gutner. Elementary approximation algorithms for prize collecting Steiner tree problems. *Information Processing Letters*, 107(1):39–44, 2008.
11. M.T. Hajiaghayi and K. Jain. The prize-collecting generalized Steiner tree problem via a new approach of primal-dual schema. *Proc. of ACM-SIAM SODA*, pages 631–640, 2006.
12. M.T. Hajiaghayi, R. Khandekar, G. Kortsarz, and Z. Nutov. Prize-collecting Steiner network problems. *Proc. of IPCO*, pages 71–84, 2010.
13. M.T. Hajiaghayi and A. Nasri. Prize-collecting Steiner networks via iterative rounding. *Proc. of LATIN*, pages 515–526, 2010.
14. K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001. Preliminary version in FOCS 1998.
15. D.S. Johnson, M. Minkoff, and S. Phillips. The prize collecting Steiner tree problem: theory and practice. *Proc. of ACM-SIAM SODA*, pages 760–769, 2000.
16. W. Kishimoto. A method for obtaining the maximum multiroute flows in a network. *Networks*, 27(4):279–291, 1996.
17. P. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted Steiner trees. *J. Algorithms*, 19(1):104–115, 1995. Preliminary version in IPCO 1993.
18. C. Nagarajan, Y. Sharma, and D.P. Williamson. Approximation algorithms for prize-collecting network design problems with general connectivity requirements. *Proc. of WAOA*, pages 174–187, 2008.
19. Z. Nutov. Approximating Steiner networks with node-weights. *SIAM Journal of Computing*, 39(7):3001–3022, 2010. Preliminary version in LATIN 2008.
20. Y. Sharma, C. Swamy, and D.P. Williamson. Approximation algorithms for prize collecting forest problems with submodular penalty functions. *Proc. of ACM-SIAM SODA*, pages 1275–1284, 2007.
21. D.P. Williamson, M.X. Goemans, M. Mihail, and V.V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica*, 15(3):435–454, 1995. Preliminary version in STOC 1993.