NODE-WEIGHTED PRIZE-COLLECTING SURVIVABLE NETWORK DESIGN PROBLEMS

BY

ALI VAKILIAN

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Adviser:

Associate Professor Chandra Chekuri

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 $r(st)$ *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. We consider three types of connectivity requirements, *edge-connectivity (EC)*, *element-connectivity (ELC)* and *vertex-connectivity (VC)*. Let $k = \max_{st} r(st)$ be the maximum requirement. There has been no non-trivial approximation for node-weighted PC-SNDP for $k > 1$ even in edge-connectivity setup. We describe multiroute-flow based relaxations for PC-EC-SNDP and PC-ELC-SNDP and obtain approximation algorithms for PC-SNDP and PC-ELC-SNDP through them. The approximation ratios we obtain for PC-EC-SNDP are similar to those that were previously known for EC-SNDP via combinatorial algorithms. Specifically, for PC-EC-SNDP (and PC-ELC-SNDP) we obtain an $O(k \log n)$-approximation in general graphs and an $O(k)$-approximation in graphs that exclude a fixed minor. Moreover, based on the approximation algorithm of ELC-SNDP and the reduction method of Chuzhoy and Khanna [6] we obtain $O(k^4 \log^2 n)$-approximation for PC-VC-SNDP which improves to $O(k^4 \log n)$ on instances from a minor-closed families of graphs.

To Father and Mother.

Acknowledgment

First and foremost I would like thank my advisor, Chandra Chekuri, whose support has been invaluable during my studies at UIUC. He introduced me to the field of network design, and this thesis would not have been possible without his ideas, guidance and support. Chandra also helped me a lot with his useful suggestions on the presentation of the results and the writing style.

This thesis is based on a joint collaboration with Chandra Chekuri and Alina Ene. I would like to thank both for their contribution to this thesis.

I am also grateful of my other collaborators at UIUC, Marianne Winslett and Arash Termehchy. I am thankful for their support and guidance during my MS studies at UIUC.

I appericate the generous award and grants supporting my study and research in UIUC. I was honored to recieve Siebel Scholar Award for 2012-2013. In addition my research was supported by NSF grant CCF-1016684.

Last but definitely not least, I want to express my deepest gratitude to my beloved parents, Amir and Maryam, and my dearest siblings, Mohsen and Fatemeh.

Table of Contents

List of Figures . . . . . **vi**

List of Tables . . . . . **vii**

Chapter 1 Introduction **1**
1.1 Problems Statement and Previous Works 1
1.2 Results and Organization 3
1.3 Preliminaries and Definitions 5
1.3.1 Connectivity models 5
1.3.2 Biset 5
1.3.3 Witness Families 8

Chapter 2 PC-EC-SNDP via Multiroute Flows **11**
2.1 LP Relaxations for node-weighted PC-EC-SNDP 11
2.2 Approximate solution to `PC-Multiroute-LP` 13
2.3 Rounding a fractional solution to `PC-Multiroute-LP` 15
2.3.1 Integrality gap of `Multiroute-LP` via `Aug-LP` 17

Chapter 3 Improved Approach and Element-Connectivity **20**
3.1 Integrality gap of PC-ELC-SNDP via `PC-ELC-Aug-LP` 20
3.2 Approximation Algorithm for Node-weighted PC-VC-SNDP 30
3.2.1 LP Relaxations for node-weighted PC-VC-SNDP 31
3.3 Integrality gap of `ELC-Aug-LP` 34

References **46**

List of Figures

1.1 *Thus the contribution of an edge e to $|\delta_F(\hat{X})| + |\delta_F(\hat{Y})|$ is at least the contribution of e to $|\delta_F(\hat{X} \cap \hat{Y})| + |\delta_F(\hat{X} \cup \hat{Y})|$.* 8

3.1 *Terminals are either in A, B, C or D.* 22
3.2 Undirected graph G with two types of vertices: reliable vertices and non-reliable ones. Circles denote reliable vertices and squares denote non-reliable ones. 27
3.3 The corresponding G_x^{st} of the graph in Figure 3.2. 27
3.4 *A simple cut in G_x^{st}. White vertices denote the vertices in $V(H_{\ell-1})$.* 28
3.5 *This figure shows the case that integrality gap of `ELC-Aug-LP` is unbounded. Note that the inner part of the biset contains a non-zero weight vertex. Consider the function h that $h(\hat{S}) = 1$ and zero for other bisets.* 38

List of Tables

1.1 Approximation ratios for different versions of SNDP. The ratios with no citation are from this thesis and based on [4]. There is an $\Omega(\log n)$-hardness for all the node-weighted problems in the table for general graphs. 3

Chapter 1

Introduction

1.1 Problems Statement and Previous Works

In the *Survivable Network Design Problem (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 required to be edge-disjoint or vertex-disjoint. An intermediate connection model is element-connectivity in which the input graph has two types of vertices, *reliable* (R) and *non-reliable* ($V-R$), and the goal is to find a minimum weight subgraph H of G that contains $r(st)$ *element-disjoint* st-paths for each $s, t \in R$ where edges and non-reliable nodes are considered as elements. We use ELC-SNDP to refer to the version of SNDP that the desired paths are required to be element-disjoint. A parameter of interest is the maximum requirement $k = \max_{st} r(st)$. Several fundamental problems in combinatorial optimization are special cases SNDP. Among them there are some polynomially solvable such as *minimum spanning tree* and NP-complete problems such as *Steiner tree or forest* problem. Edge-weighted SNDP problems have been studied extensively and many different approaches have developed to give an approximation for them. 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 in H. By applying primal-dual approach in an augmentation framework, Williamson et al. [27] obtains $O(k)$-approximation for edge-weighted EC-SNDP. The ratio was improved to $O(\log k)$ by Goemans et al. via doing the augmentation in reverse [9]. Jain et al. [19] showed that primal-dual technique also gives $O(\log n)$-approximation for edge-weighted ELC-SNDP. Jain [18] obtained a 2-approximation for this problem via the influential iterated rounding technique that he introduced. Later, Fleischer et al. [8] extended the approach to the element-connectivity setup and obtained a 2-approximation for edge-weighted ELC-SNDP as well. However, the problem is much harder in vertex-connectivity setup. While EC-SNDP and ELC-SNDP both admits a 2-approximation in edge-weighted graphs, the only non-trivial approach toward edge-weighted VC-SNDP is due to Chuzhoy and Khanna which gives $O(k^3 \log n)$-approximation [6]. We refer the reader to the survey [12] for much more detailed information on the different approaches toward network design problems. In this thesis we focus on 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 EC-SNDP, the node-weighted Steiner tree problem is already $\Omega(\log n)$-hard to approximate via a simple reduction from the Set Cover problem [22].

Klein and Ravi [22] 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 $2H(n)$-approximation for the Steiner tree and Steiner forest problems where $H(n) = \sum_{i=1}^{n} 1/i = O(\log n)$. Guha and Khuller improved the ratio to $(1.35 + \varepsilon)H(n)$ [10]. Their algorithms are based on finding a structure called *spider*. Nutov examined the approximability of node-weighted SNDP [24] and obtained an $O(k \log n)$-approximation via the augmentation framework of Williamson et al. [27] (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. 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 [22] and that of Nutov [24] 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. [11] 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 [7] 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. Furthermore, Chekuri et al. [3] generalized the work of Demaine et al. [7] and described an $O(k)$-approximation for node-weighted EC-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 (`Aug-LP`) for each stage (that depends on the solution for the previous stages) is used[2].

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. The prize-collecting version of Steiner tree and Steiner forest have been studied extensively and have several theoretical and practical applications [13, 14, 20, 26]. Previous work on prize-collecting SNDP has considered submodular penalty

[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]There is some subtlety to understanding the integrality gap of `Aug-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 3.3 and [3] for more details.

	EC-SNDP	PC-EC-SNDP	Elem-SNDP	PC-ELC-SNDP	(PC-)VC-SNDP
Gen graphs, EW	2 [18]	2.54 [15]	2 [8]	2.54 [15]	$O(k^3 \log n)$ [6, 15]
Planar, EW	2 [18]	2.54 [15]	2 [8]	2.54 [15]	$O(k^3 \log n)$ [6, 15]
Gen graphs, NW	$O(k \log n)$ [24]	$O(k \log n)$	$O(k \log n)$ [24]	$O(k \log n)$	$O(k^4 \log^2 n)$ [4, 24]
Planar, NW	$O(k)$ [3]	$O(k)$	$O(k)$ [3]	$O(k)$	$O(k^4 \log n)$

Table 1.1: Approximation ratios for different versions of SNDP. The ratios with no citation are from this thesis and based on [4]. There is an $\Omega(\log n)$-hardness for all the node-weighted problems in the table for general graphs.

functions [15, 26]; 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 [5].

A simple scaling technique, introduced by Bienstock et al. [2], shows how one can use an LP relaxation based ρ-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 [15, 16, 23]. In [15] a technique similar to that of Bienstock et al. is used for edge-weighted EC-SNDP (and also for ELC-SNDP and VC-SNDP). However, [15] showed that a straightforward and natural LP relaxation has a large integrality gap, and introduced a stronger LP relaxation. 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.

1.2 Results and Organization

We formulate an LP relaxation for node-weighted EC-SNDP and PC-EC-SNDP in edge-connectivity setup via *multi-route flows* [1, 21]. The multi-route flow based relaxation easily allows us to apply the basic idea of Bienstock et al. [2] to reduce the PC-EC-SNDP problem to the EC-SNDP problem. Then, we analyze the integrality gap of this relaxation for node-weighted EC-SNDP. We obtain an upper bound on the integrality gap by relating the optimum value of the relaxation to that of the `Aug-LP` relaxation [3] in each phase of the augmentation framework. Chekuri et al. [3, 4] show that `Aug-LP` has an integrality gap of $O(\log n)$ for general graphs which improves to $O(1)$ in minor-closed families of graphs. Further we extend the approach to the element-connectivity and obtain the same approximation guarantees for prize-collecting versions. These ingredients give us the following theorem that summarizes our results.

Theorem 1.2.1. *There is an $O(k \log n)$-approximation for node-weighted PC-ELC-SNDP in undirected graphs. Moreover, let $\mathcal{G}$ be a minor-closed family of graphs. There is an $O(k)$-approximation for node-weighted PC-ELC-*

SNDP on instances in which the graph is in $\mathcal{G}$, where the constant only depends on the family $\mathcal{G}$.

Based on the $O(k \log n)$-approximation for `Aug-LP` in element-connectivity setup and the reduction method of Chuzhoy and Khanna for VC-SNDP [6], we obtain an $O(k^4 \log^2 n)$-approximation for node-weighted PC-VC-SNDP which improves to $O(k^4 \log n)$ for instances in minor-closed families of graphs.

Theorem 1.2.2. *There is an $O(k^4 \log^2 n)$-approximation for node-weighted PC-VC-SNDP in undirected graphs. Moreover, let $\mathcal{G}$ be a minor-closed family of graphs. There is an $O(k^4 \log n)$-approximation for node-weighted PC-VC-SNDP on instances in which the input graph is in $\mathcal{G}$, where the constant only depends on the family $\mathcal{G}$.*

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. The NP-hardness of the single pair problem[3] 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. In Chapter 2 we give an $O(k^2 \log n)$-approximation by solving the LP-relaxation approximately; this approach has appeared in [4]. One factor of k from approximating the relaxation, and another factor of k from the augmentation framework. In Chapter 3, we show that it is also possible to solve the prize-collecting instance directly in the augmentation framework; we use the augmentation framework and solve the problem in k phases. In each phase ℓ, we decide whether to increase the connectivity requirement of an unsatisfied pair or pay its penalty. A separate LP relaxation (`PC-Aug-LP`) is used in each stage. Via the scaling method of Bienstock et al. and the $O(\log n)$ integrality gap of `Aug-LP`, we show a similar integrality gap for `PC-Aug-LP`. This helps us to get around solving `Multiroute-LP` and save a factor of k that we missed to have an approximate feasible solution to `Multiroute-LP`. In Chapter 3 we consider this approach in a more general connectivity setup, element-connectivity, and give an $O(k \log n)$-approximation for node-weighted PC-ELC-SNDP. This ratio improves to $O(k)$ in minor-closed families of graphs.

[3] Via a reduction to the dense-k-subgraph, Nutov [24] gives an evidence that the single pair problem may not admit a polylogarithmic approximation.

1.3 Preliminaries and Definitions

1.3.1 Connectivity models

To define the problem first we need to know about the exact definition of different models of connections in a graph. In an undirected graph G, two nodes s and t are k-edge (k-vertex) connected if G contains k edge-disjoint (vertex-disjoint) paths from s to t. There is an intermediate notion of connectivity known as element-connectivity.

Definition 1.3.1. *Let $G = (V, E)$ be an undirected graph with two types of vertices: reliable and non-reliable where $R \subseteq V$ denotes reliable vertices. A pair of reliable nodes s, t is k-element connected iff there are k edge-disjoint st-paths in G such that each **non-reliable** node appears in at most one of them (there is no restriction for reliable nodes). In other words, we consider edges and non-reliables nodes as elements and we look for a set of paths that do not share any element.*

There is a close relation between the maximum connectivity of a pair of vertices and the minimum cut separating them which is characterized by Menger's theorem. Let $S \subset V(G)$ be a subset of vertices in G. We denote the set of edges crossing S by $\delta(S)$; $\delta(S) = \{uv \in E(G) | u \in S, v \notin S\}$. Moreover, we denote the set of all neighbors of S by $\Gamma(S)$; $\Gamma(S) = \{v | v \notin S, \exists u \in S \text{ s.t. } uv \in E(G)\}$. Menger's theorem plays an important role in the problems dealing with connectivity requirements. The statement of Menger's theorem in different connection models are as follows:

Theorem 1.3.2. Edge-connectivity version of Menger's theorem: *Let G be an undirected graph. Two vertices s and t are k-edge connected iff for each set $S \subset V$ such that $s \in S$ and $t \notin S$, $|\delta(S)| \geq k$.*

To state Menger's theorem for element-connectivity and vertex-connectivity we need to introduce some notation called *biset* (see Subsection 1.3.2).

Theorem 1.3.3. Element-connectivity version of Menger's theorem: *Let G be an undirected graph. Two vertices s and t are k-element connected iff for each bisets $\hat{S} \subset V \times V$ such that $s \in S$ and $t \notin S'$ and $S' - S$ only contains non-reliable vertices, $|\delta(\hat{S})| + |S' - S| \geq k$.*

Theorem 1.3.4. Vertex-connectivity version of Menger's theorem: *Let G be an undirected graph. Two vertices s and t are k-vertex connected iff for each biset $\hat{S} \subset V \times V$ such that $s \in S$ and $t \notin S'$, $|\delta(\hat{S})| + |S' - S| \geq k$.*

1.3.2 Biset

To deal with vertex connectivity and element connectivity problems we need to consider a pair of ordered sets. This will help us to establish Menger's theorem for these connection models. The definitions and notation that we use are

based on previous work [8, 17, 25]. We work with bisets $\hat{X} = (X, X') \in 2^V \times 2^V$ such that $X \subseteq X'$. The set X is the *inner part* of $\hat{X}$, X' is the *outer part* of $\hat{X}$, and $X' - X$ is the *boundary* of $\hat{X}$.

We define a relation $\preceq$ on the bisets as follows. Let $\hat{X} = (X, X')$ and $\hat{Y} = (Y, Y')$ be two bisets. We have $\hat{X} \preceq \hat{Y}$ iff $X \subseteq Y$ and $X' \subseteq Y'$. It is straightforward to verify that $\preceq$ is a partial order. We say that $\hat{Y}$ contains $\hat{X}$ if $\hat{X} \preceq \hat{Y}$.

We also define the following operations on the bisets. Let $\hat{X} = (X, X')$ and $\hat{Y} = (Y, Y')$ be two bisets. The union, intersection, and difference of $\hat{X}$ and $\hat{Y}$ are defined as $\hat{X} \cap \hat{Y} = (X \cap Y, X' \cap Y')$, $\hat{X} \cup \hat{Y} = (X \cup Y, X' \cup Y')$, and $\hat{X} - \hat{Y} = (X - Y', X' - Y)$.

Proposition 1. *Let $\hat{X} = (X, X')$ and $\hat{Y} = (Y, Y')$ be two bisets such that $X \subseteq X'$ and $Y \subseteq Y'$. We have $\hat{X} \cap \hat{Y} \preceq \hat{X}$ and $\hat{X} - \hat{Y} \preceq \hat{X}$.*

Definition 1.3.5. *Two bisets $\hat{X}$ and $\hat{Y}$ are **non-overlapping** iff **one** of the following holds:*

(i) $\hat{X} \preceq \hat{Y}$.

(ii) $\hat{Y} \preceq \hat{X}$.

(iii) The sets $X' \cap Y$ and $X \cap Y'$ are empty.

*If the bisets do not satisfy any of the conditions above, we say that they are **overlapping**.*

Similar to different types of set function, we can define submodular/supermodular functions in biset setup. Many network design (or more generally connectivity) problems in edge-connectivity model are captured as covering a set function via Menger's theorem (see Theorem 1.3.2). Similarly via Menger's theorem (see Theorem 1.3.4 and 1.3.3), we can model the same problem in vertex-connectivity (or element-connectivity) as covering a biset function.

Definition 1.3.6. ***Overlapping bifamily:*** *A biset family $\mathcal{F}$ is called overlapping if for any two overlapping bisets $\hat{X}$ and $\hat{Y}$, all bisets $\hat{X} \cap \hat{Y}$, $\hat{X} \cup \hat{Y}$, $\hat{X} - \hat{Y}$ and $\hat{Y} - \hat{X}$ are in $\mathcal{F}$.*

Definition 1.3.7. ***Bisubmodular function:*** *Let $f : \mathcal{P} \to \mathbb{Z}$ be a function defined on an overlapping bifamily $\mathcal{P}$. The function f is **bisubmodular** iff for any two bisets $\hat{X}$ and $\hat{Y}$ in $\mathcal{P}$, **both** of the following hold:*

$$f(\hat{X}) + f(\hat{Y}) \geq f(\hat{X} \cap \hat{Y}) + f(\hat{X} \cup \hat{Y}), \quad \forall \hat{X}, \hat{Y} \in \mathcal{P} \tag{1.1}$$

$$f(\hat{X}) + f(\hat{Y}) \geq f(\hat{X} - \hat{Y}) + f(\hat{Y} - \hat{X}), \quad \forall \hat{X}, \hat{Y} \in \mathcal{P} \tag{1.2}$$

*A function f is **bisupermodular** iff $-f$ is bisubmodular.*

Definition 1.3.8. ***Skew-bisupermodular function:*** *Let $f : \mathcal{P} \to \mathbb{Z}$ be a function defined on an overlapping bifamily $\mathcal{P}$. The function f is* ***skew-bisupermodular*** *iff for any two bisets $\hat{X}$ and $\hat{Y}$ in $\mathcal{P}$,* ***one*** *of the following holds:*

$$f(\hat{X} \cap \hat{Y}) + f(\hat{X} \cup \hat{Y}) \geq f(\hat{X}) + f(\hat{Y}), \quad \forall \hat{X}, \hat{Y} \in \mathcal{P} \tag{1.3}$$

$$f(\hat{X} - \hat{Y}) + f(\hat{Y} - \hat{X}) \geq f(\hat{X}) + f(\hat{Y}), \quad \forall \hat{X}, \hat{Y} \in \mathcal{P} \tag{1.4}$$

Definition 1.3.9. 0-1 ***Biuncrossable function:*** *Let $f : \mathcal{P} \to \mathbb{Z}$ be a function defined on a collection of bisets $\mathcal{P}$. The function f is* 0-1 ***biuncrossable*** *iff for any two bisets $\hat{X}$ and $\hat{Y}$ in $\mathcal{P}$ such that $f(\hat{X}) = f(\hat{Y}) = 1$,* ***one*** *of the following holds:*

$$\hat{X} \cap \hat{Y} \in \mathcal{P}, \hat{X} \cap \hat{Y} \in \mathcal{P}, f(\hat{X} \cap \hat{Y}) + f(\hat{X} \cup \hat{Y}) \geq f(\hat{X}) + f(\hat{Y}), \tag{1.5}$$

$$\hat{X} - \hat{Y} \in \mathcal{P}, \hat{Y} - \hat{X} \in \mathcal{P}, f(\hat{X} - \hat{Y}) + f(\hat{Y} - \hat{X}) \geq f(\hat{X}) + f(\hat{Y}), \tag{1.6}$$

Proposition 2. *Let f be a skew bisupermodular function and let g be a bisubmodular function on the same domain $\mathcal{P}$. Then $f - g$ is skew bisupermodular.*

Proof: Since f is a skew-bisupermodular function, for any two bisets $\hat{X}$ and $\hat{Y}$ in $\mathcal{P}$, either $f(\hat{X}) + f(\hat{Y}) \leq f(\hat{X} \cap \hat{Y}) + f(\hat{X} \cup \hat{Y})$ or $f(\hat{X}) + f(\hat{Y}) \leq f(\hat{X} - \hat{Y}) + f(\hat{Y} - \hat{Y})$. Suppose that the former case holds (the other case is similar). Since g is a bisubmodular function, $g(\hat{X}) + g(\hat{Y}) \geq g(\hat{X} \cap \hat{Y}) + g(\hat{X} \cup \hat{Y})$. Thus $f(\hat{X}) - g(\hat{X}) + f(\hat{Y}) - g(\hat{Y}) \leq f(\hat{X} \cap \hat{Y}) - g(\hat{X} \cap \hat{Y}) + f(\hat{X} \cup \hat{Y}) - g(\hat{X} \cup \hat{Y})$. □

For a set pair $\hat{X} = (X, X')$ such that $X \subseteq X'$ and any set F of edges, we let $\delta_F(\hat{X})$ be the set of all edges of F with one endpoint in X and the other in $V - X'$. Moreover, we let $\Gamma_F(\hat{X})$ be the set of all vertices with an incident edge in $\delta_F(\hat{X})$.

Lemma 1.3.10. *For any set F of edges, the function $|\delta_F(\cdot)|$ is bisubmodular.*

Proof: We show that $|\delta_F(\hat{X})| + |\delta_F(\hat{Y})| \geq |\delta_F(\hat{X} \cap \hat{Y})| + |\delta_F(\hat{X} \cup \hat{Y})|$ and the other case can be shown similarly. Consider an edge $e \in \delta_F \hat{X} \cap \hat{Y}$. It has one endpoint in $X \cap Y$ and its other endpoint is either in $X' - Y'$ (or $Y' - X'$) or $V - (X' \cup Y')$ (consider e_3 and e_4 in Figure 1.1). In the former case, $e \in \delta_F(\hat{X})$ and in the latter case e is in both $\delta_F(\hat{X})$ and $\delta_F(\hat{Y})$.

Next, consider an edge $e \in \delta_F(\hat{X} \cap \hat{Y})$. It has one endpoint in $V - (X' \cup Y')$ and its other endpoint is either in X (or Y) or $X \cap Y$) (consider e_1 and e_4 in Figure 1.1). In the first case, $e \in \delta_F(\hat{X})$ and in the second case e is in both $\delta_F(\hat{X})$ and $\delta_F(\hat{Y})$. Thus the contribution of an edge e to the left hand side is at least the contribution of e to the right hand side and the equation holds. □

Lemma 1.3.11. *For any two bisets $\hat{X}$ and $\hat{Y}$, we have*

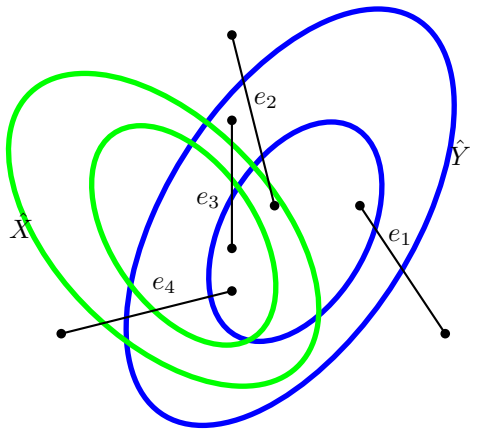

Figure 1.1: *Thus the contribution of an edge e to $|\delta_F(\hat{X})|+|\delta_F(\hat{Y})|$ is at least the contribution of e to $|\delta_F(\hat{X}\cap\hat{Y})|+|\delta_F(\hat{X}\cup\hat{Y})|$.*

- $|X' - X| + |Y' - Y| = |(X' \cap Y') - (X \cap Y)| + |(X' \cup Y') - (X \cup Y)|$, *and*
- $|X' - X| + |Y' - Y| = |(X' - Y) - (X - Y')| + |(Y' - X) - (Y - X)| + 2|(X' \cap Y') - (X \cap Y)|$.

1.3.3 Witness Families

A natural approach in covering a biset function is to consider the family of bisets that are required to be covered and exploit their structural properties. It essentially makes sense when the underlying biset function is a 0-1 function. In the following we define some useful concepts in this regards.

Definition 1.3.12. Feasible cover: *Let $f : \mathcal{P} \to \mathbb{Z}$ be a function defined on a collection $\mathcal{P}$ of bisets on vertex set of a graph $G = (V, E)$. A subgraph[4] H of G is a **cover** of f iff, for each biset $\hat{X} \in \mathcal{P}$, we have $|\delta_H(\hat{X})| \geq f(\hat{X})$.*

Definition 1.3.13. Minimal violated bisets: *Let $f : \mathcal{P} \to \mathbb{Z}$ be a function defined on a collection $\mathcal{P}$ of bisets. A biset $\hat{X}$ is a **violated biset** with respect to a graph H iff $|\delta_H(\hat{X})| \leq f(\hat{X})$. A biset $\hat{X}$ is a **minimal violated biset** of f iff $\hat{X}$ is violated with respect to H and there does not exist a violated biset $\hat{Y}$ with respect to H such that $\hat{Y} \preceq \hat{X}$.*

To analyze the performance of an algorithm for covering a biset function (or solving its corresponding network design problem), a useful concept is witness sets and families. Picking an inclusion-wise minimal feasible cover M of a set function f, S_e is a witness set for e if e is the only edge in M covering S_e; e is crucial to cover S_e. In this part, we extend the notion of witness sets and families to the biset setting. We use the witness biset and bifamilies in Section 3.3 to bound the integrality gap of `ELC-Aug-LP`.

Definition 1.3.14. *Let F be a set of edges. A set pair $\hat{S}_e = (S_e, S'_e)$ is an F**-witness pair** for an edge e iff $h(\hat{S}_e) = 1$ and $\delta_F(\hat{S}_e) = \{e\}$.*

[4]We sometimes abuse notation and we refer to a set of edges as a feasible cover.

A useful observation that we will need later is that minimal violated bisets of a biuncrossable function do not overlap with other (not necessarily minimal) violated bisets.

Definition 1.3.15. Bilaminar family: *Let $\mathcal{F}$ be a family of bisets. The family is **bilaminar** iff, for any two bisets $\hat{X}$ and $\hat{Y}$ in $\mathcal{F}$, $\hat{X}$ and $\hat{Y}$ are non-overlapping.*

As shown in the following lemma, we can construct a bilaminar witness family for "non-redundant" edges.

Lemma 1.3.16. *Let F be a feasible cover for h. Let $M \subseteq F$ be a subset of F such that, for each edge $e \in M$, $F - e$ is not a feasible cover for h. There is a bilaminar family $\mathcal{L} = \{\hat{S}_e \mid e \in M\}$ that contains an F-witness set pair $\hat{S}_e = (S_e, S'_e)$ for each edge $e \in M$.*

In the rest of this section we prove Lemma 1.3.16.

Lemma 1.3.17. *Let F be a feasible cover for biuncrossable function h. Let $M \subseteq F$ be a subset of F such that M is a feasible cover for h and for each edge $e \in M$, $M - e$ is not a feasible cover for h. There is a family $\mathcal{F} = \{\hat{S}_e \mid e \in M\}$ that contains an M-witness biset $\hat{S}_e = (S_e, S'_e)$ for each edge $e \in M$.*

Proof: Let e be an edge of M. Since $M - e$ is not a feasible cover for h, there is a biset $\hat{S}_e = (S_e, S'_e)$ such that $h(\hat{S}_e) = 1$ and $\delta_M(\hat{S}_e) = \{e\}$. $\hat{S}_e$ is an F-witness biset for e. Then $\mathcal{F} = \{\hat{S}_e \mid e \in M\}$ is the desired family. □

In order to get a laminar family of M-witness bisets, we start with a family $\mathcal{F}$ of witness biset that is guaranteed by Lemma 1.3.17 and we replace overlapping bisets with non-overlapping ones. The following lemma shows that, if we have two witness bisets $\hat{S}_{e_1}$ and $\hat{S}_{e_2}$ that overlap, we can replace them by two bisets that are also witness bisets; by Proposition 3, the latter bisets do not overlap.

Proposition 3. *Let $\hat{X}$ and $\hat{Y}$ be two bisets. The bisets $\hat{X} \cap \hat{Y}$ and $\hat{X} \cup \hat{Y}$ do not overlap. Additionally, the bisets $\hat{X} - \hat{Y}$ and $\hat{Y} - \hat{X}$ do not overlap.*

Lemma 1.3.18. *Let $\hat{S}_{e_1} = (S_{e_1}, S'_{e_1})$ and $\hat{S}_{e_2} = (S_{e_2}, S'_{e_2})$ be two M-witness bisets for the edges e_1 and e_2, respectively. Then one of the following must hold:*

(i) The bisets $\hat{S}_{e_1} \cap \hat{S}_{e_2}$ and $\hat{S}_{e_1} \cup \hat{S}_{e_2}$ are M-witness bisets for distinct edges in the set $\{e_1, e_2\}$.

(ii) The bisets $\hat{S}_{e_1} - \hat{S}_{e_2}$ and $\hat{S}_{e_2} - \hat{S}_{e_1}$ are M-witness bisets for distinct edges in the set $\{e_1, e_2\}$.

Proof Sketch: Both $\hat{S}_{e_1}$ and $\hat{S}_{e_2}$ are M-witness bisets. This implies that $h(\hat{S}_{e_1}) = h(\hat{S}_{e_2}) = 1$. Since h is a biuncrossable function, either $h(\hat{S}_{e_1} \cap \hat{S}_{e_2}) = h(\hat{S}_{e_1} \cup \hat{S}_{e_2}) = 1$ or $h(\hat{S}_{e_1} - \hat{S}_{e_2}) = h(\hat{S}_{e_2} - \hat{S}_{e_1}) = 1$.

Suppose that $h(\hat{S}_{e_1} \cap \hat{S}_{e_2}) = h(\hat{S}_{e_1} \cup \hat{S}_{e_2}) = 1$. Since M is a feasible cover for h, we have $|\delta_M(\hat{S}_{e_1} \cap \hat{S}_{e_2})| \geq 1$ and $|\delta_M(\hat{S}_{e_1} \cup \hat{S}_{e_2})| \geq 1$. Since $|\delta_M(\cdot)|$ is bisubmodular, it follows from the inequality (1.1) that $|\delta_M(\hat{S}_{e_1} \cap \hat{S}_{e_2})| +$

$|\delta_M(\hat{S}_{e_1} \cup \hat{S}_{e_2})| \leq 2$. Therefore $|\delta_M(\hat{S}_{e_1} \cup \hat{S}_{e_2})| = |\delta_M(\hat{S}_{e_1} \cap \hat{S}_{e_2})| = 1$. It is straightforward to verify that, since $\hat{S}_{e_1}$ and $\hat{S}_{e_2}$ are witness bisets for e_1 and e_2, $\hat{S}_{e_1} \cap \hat{S}_{e_2}$ and $\hat{S}_{e_1} \cup \hat{S}_{e_2}$ are witness bisets for distinct edges of $\{e_1, e_2\}$.

The case $h(\hat{S}_{e_1} - \hat{S}_{e_2}) = h(\hat{S}_{e_2} - \hat{S}_{e_1}) = 1$ is similar and we omit it. (Here we use the fact that $|\delta_M(\cdot)|$ satisfies inequality (1.2)). □

Using the following lemma, we can show that, if we replace two overlapping witness set pairs by the witness bisets guaranteed by Lemma 1.3.18, the number of bisets that are overlapping decreases and thus we are making progress towards a laminar witness bifamily.

Lemma 1.3.19 (Fleischer et al. [8])**.** *Let $\hat{X} = (X, X')$, $\hat{Y} = (Y, Y')$ and $\hat{Z} = (Z, Z')$ be bisets. Suppose that $\hat{X}$ and $\hat{Y}$ overlap. Then the number of pairs of $\{(\hat{X}, \hat{Z}), (\hat{Y}, \hat{Z})\}$ that overlap is at least the number of pairs of $\{(\hat{X} \cap \hat{Y}, \hat{Z}), (\hat{X} \cup \hat{Y}, \hat{Z})\}$ that overlap. Additionally, the number of pairs of $\{(\hat{X}, \hat{Z}), (\hat{Y}, \hat{Z})\}$ that overlap is at least the number of pairs of $\{(\hat{X} - \hat{Y}, \hat{Z}), (\hat{Y} - \hat{X}, \hat{Z})\}$ that overlap.*

Proof of Lemma 1.3.16: Let $\mathcal{F}$ be the witness bifamily guaranteed by Lemma 1.3.17. If no two bisets in $\mathcal{F}$ overlap, $\mathcal{F}$ is the desired bifamily. Otherwise, let $\hat{S}_{e_1}$ and $\hat{S}_{e_2}$ be two bisets that overlap. By Lemma 1.3.18, we can replace $\hat{S}_{e_1}$ and $\hat{S}_{e_2}$ by either $\hat{S}_{e_1} \cap \hat{S}_{e_2}$ and $\hat{S}_{e_1} \cup \hat{S}_{e_2}$ or by $\hat{S}_{e_1} - \hat{S}_{e_2}$ and $\hat{S}_{e_2} - \hat{S}_{e_1}$. Lemma 1.3.19 implies that the resulting bifamily is a witness bifamily for M that has fewer overlapping bisets. Thus we can repeat this process until we get a witness bifamily in which no two bisets overlap. □

Chapter 2

PC-EC-SNDP via Multiroute Flows

In this chapter, we consider node-weighted PC-SNDP in edge-connectivity setup (PC-EC-SNDP) and obtain an $O(k^2 \log n)$-approximation. In Chapter 3, we improve our method and give an $O(k \log n)$-approximation for node-weighted prize-collecting SNDP in a more general connectivity setup, element-connectivity.

2.1 LP Relaxations for node-weighted PC-EC-SNDP

Let s and t be two vertices of the graph and let ℓ 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 ℓ-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-EC-SNDP. 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 4. *PC-Multiroute-LP is a valid relaxation for the node-weighted PC-EC-SNDP. Moreover if there is a single pair st with non-zero requirement then the relaxation is exact.*

Proof: Let H be a feasible solution for the node-weighted PC-EC-SNDP on graph G with requirement function $r(st)$ and penalty function $\pi(st)$ for each pair of vertices st. Construct a solution (x, z) to PC-Multiroute-LP as follows. Let $x(v) = 1$ if $v \in V(H)$ and 0 otherwise. For each pair s and t which is not connected via $r(st)$ edge-disjoint paths in H, let $z(st) = 1$; otherwise, let $z(st) = 0$. For each pair of vertices st with $z(st) = 0$, consider a set of $r(st)$ edge-disjoint st-paths $\overline{p}_{st}$ and let $f(\overline{p}_{st}) = 1$; let $f(\overline{p}) = 0$ for any $\overline{p} \in \mathcal{P}_{st}^{r(st)} - \overline{p}_{st}$. Since for each pair of terminals st with $z(st) = 0$ we have picked an $r(st)$-flow, $\overline{p}_{st}$, (x, z) satisfy the first set of constraints. Moreover, $f(\overline{p}) = 1$ if all vertices of $\overline{p}$ are in H. Since for each $v \in V(H)$, $x(v) = 1$ and for each pair st one tuple has non-zero f-value, (x, z) satisfy the second set of constraints too. Hence, PC-Multiroute-LP is a valid relaxation for PC-EC-SNDP.

Now, consider the case that only one pair has non-zero requirement. Let s and t be the pair with non-zero requirement. For each $\overline{p} \in \mathcal{P}_{st}^{r(st)}$, let $w(\overline{p}) = \sum_{v \in \overline{p}} w(v)$. For each feasible solution of PC-Multiroute-LP, the value of the solution is a linear combination of $\pi(st)$ and $\{w(\overline{p}) | \overline{p} \in \mathcal{P}_{st}^{r(st)}\}$ which is less than $\min(\pi(st), \min_{\overline{p} \in \mathcal{P}_{st}^{r(st)}} w(\overline{p}))$. Since $\pi(st)$ and $w(\overline{p})$ correspond to integral solutions, the relaxation is exact in this case. □

Corollary 2.1.1. *Multiroute-LP is a valid relaxation for the node-weighted EC-SNDP. 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 method 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-EC-SNDP and the ratio improves to $O(k^2)$ for minor-closed families of graphs.

Remark 2.1.2. *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-EC-SNDP is equivalent (in the sense of having the same optimal value for each instance) to the cut-based relaxation from [15].*

2.2 Approximate solution to `PC-Multiroute-LP`

Consider the following special case of the node-weighted SNDP problem in which we are given a single pair (s, t) and an integer k and the goal is to find a minimum weight subgraph H such that s and t are k-edge-connected in H. We will refer to this special case as the node-weighted *single-pair connectivity problem*.

`Multiroute-LP` and `PC-Multiroute-LP` are exact on instances of the node-weighted single pair connectivity problem. The edge-weighted single pair connectivity problem is solvable in polynomial time but the node-weighted problem is known to be **NP**-hard when k is part of the input via a reduction from the DENSEST SUBGRAPH problem [24]. Therefore we cannot solve the `Multiroute-LP` or the `PC-Multiroute-LP` relaxations exactly when k is large. In the following, we show that we can solve the `PC-Multiroute-LP` relaxation approximately.

We can write a compact formulation of `PC-Multiroute-LP` as follows. We replace each edge uv by two directed edges $u \to v$ and $v \to u$, one in each direction; we refer to $u \to v$ and $v \to u$ as the arcs corresponding to the edge uv. Let A denote the set of all resulting arcs. As before, we have a variable $x(v)$ for each vertex v and a variable $z(st)$ for each pair st. We have a variable $f(a, st)$ for each directed edge a and each pair st with the interpretation that $f(a, st) = 1$ iff the solution satisfies the requirement of st and a is on the $r(st)$-tuple $\overline{p}_{st} \in \mathcal{P}_{st}^{r(st)}$ chosen for st.

For each pair st, we impose the constraint that the values $f(a, st)$ define a flow that sends $(1 - z(st))r(st)$ units of flow from s to t subject to the following capacity constraints. For a pair st, each vertex v has a capacity of $r(st)x(v)$. (As before, different pairs do not share the capacity.)

In the following, we use $\delta^-(v)$ to denote the set of all incoming arcs of v, that is, arcs $u \to v$. We use $\delta^+(v)$ to denote the set of all outgoing arcs of v, that is, arcs $v \to u$.

Compact-PC-Multiroute-LP

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

$$\text{s.t.} \sum_{a \in \delta^+(s)} f(a, st) - \sum_{a \in \delta^-(s)} f(a, st) = (1 - z(st))r(st) \quad \forall st$$

$$\sum_{a \in \delta^-(v)} f(a, st) = \sum_{a \in \delta^+(v)} f(a, st) \quad \forall st, \forall v \notin \{s, t\}$$

$$f(a, st) \leq 1 - z(st) \quad \forall a, \forall st$$

$$\sum_{a \in \delta^-(v)} f(a, st) \leq r(st)x(v) \quad \forall st, \forall v$$

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

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

$$f(a, st) \geq 0 \quad \forall a, \forall st$$

Proposition 5. *Compact-PC-Multiroute-LP is a valid relaxation for the node-weighted PC-SNDP.*

Proof: Let H be a feasible solution for PC-SNDP. For each $v \in V$, let $x(v) = 1$ if $v \in V(H)$ and zero otherwise. For each pair of terminals st, let $z(st) = 1$ if H contains less than $r(st)$ edge-disjoint st-paths and zero otherwise. Moreover, for each pair st with $z(st) = 0$ pick a set of $r(st)$ edge-disjoint st-paths and orient them from s to t. Let $f(a, st) = 1$ if a is in the st-flow of value $r(st)$.

(x, z) satisfy the first set of constraints since for each st either $z(st) = 1$ and the right hand side is zero or there exists an st flow of value $r(st)$. The second constraint holds since we have a valid st-flow. Consider the third set of constraints. For a pair of terminals with $z(st) = 1$ we don not add any st-paths and it holds. In the case that $z(st) = 0$, the right hand side is 1 and since for each a, $f(a, st) \leq 1$ the constraint holds trivially. Moreover, (x, z) satisfy the last set of constraints because for each pair of terminals st we have picked at most $r(st)$ st-paths and thus in-degree of each vertex $v \in H$ is at most $r(st) = r(st)x(v)$. □

In the following, we show that, given an optimal solution to Compact-PC-Multiroute-LP, we can construct a k-approximate solution to PC-Multiroute-LP in polynomial time. More precisely, we prove the following theorem.

Theorem 2.2.1. *We can find in polynomial time a fractional solution (x, f, z) to PC-Multiroute-LP whose cost $\sum_v x(v)w(v) + \sum_{st} z(st)\pi(st)$ is at most $k \cdot$ OPT, where* OPT *is the cost of the optimal fractional solution to PC-Multiroute-LP.*

The following lemma allow us to decompose an edge-based flow into a multi-route flow. We refer the reader to Lemma 2 in [1] for a proof.

Lemma 2.2.2 ([1, 21]). *Let $G = (V, A)$ be a directed network and let s and t be two vertices of V. Let k be an integer and let ν be a non-negative real number. Let $f : A \to \mathbb{R}_+$ be an st flow such that the value of f is $k\nu$ and $f(a) \leq \nu$ for each arc $a \in A$. Let $\mathcal{P}_{st}^k$ be the set of all k-route tuples of G that connect s to t. There is a k-route flow $g : \mathcal{P}_{st}^k \to \mathbb{R}_+$ of value ν such that $g(a) \leq f(a)$ for all $a \in A$, where $g(a) = \sum_{\overline{p} \in \mathcal{P}_{st}^k, a \in \overline{p}} g(\overline{p})$. Moreover, we can construct such a flow g in polynomial time.*

Proposition 6. *Given a feasible solution (x, f, z) to `Compact-PC-Multiroute-LP`, we can find in polynomial time a feasible solution (x', z', f') to `PC-Multiroute-LP` such that*

$$\sum_v x'(v)w(v) + \sum_{st} z'(st)\pi(st) \leq k \left(\sum_v x(v)w(v) + \sum_{st} z(st)\pi(st) \right).$$

Proof: Let (x, f, z) be a feasible solution to `Compact-PC-Multiroute-LP`. Let (x', z', f') be the following solution. For each vertex v, we set $x'(v) = \min\{1, kx(v)\}$. For each pair st, we set $z'(st) = z(st)$. Finally, for each pair st, we apply Lemma 2.2.2 to $f(\cdot\,, st)$ in order to get an $r(st)$-route flow $g_{st} : \mathcal{P}_{st}^{r(st)} \to \mathbb{R}_+$ of value $1 - z(st)$, and we set $f'(\overline{p}) = g_{st}(\overline{p})$ for all pairs st and all tuples $\overline{p} \in \mathcal{P}_{st}^{r(st)}$. It is straightforward to verify that (x', z', f') is a feasible solution to `PC-Multiroute-LP`. □

Proposition 7. *Let (x, f, z) be a feasible solution to `PC-Multiroute-LP`. There exists a feasible solution (x', z', f') to `Compact-PC-Multiroute-LP` such that*

$$\sum_v x'(v)w(v) + \sum_{st} z'(st)\pi(st) \leq \sum_v x(v)w(v) + \sum_{st} z(st)\pi(st).$$

Proof: We set $x' = x$ and $z' = z$. For each pair st, we orient paths in $\mathcal{P}_{st}^{r(st)}$ from s to t. Then, we set $f'(a, st) = \sum_{\overline{p} \in \mathcal{P}_{st}^{r(st)}, a \in \overline{p}} f(\overline{p})$. It is straightforward to verify that (x', z', f') is feasible for `Compact-PC-Multiroute-LP`. □

The `Compact-PC-Multiroute-LP` relaxation has polynomial size and therefore we can solve it in polynomial time. Proposition 6 and Proposition 7 imply that we can find a k-approximate solution to `PC-Multiroute-LP` in polynomial time via solving the corresponding instance of `Compact-PC-Multiroute-LP` (see Theorem 2.2.1).

2.3 Rounding a fractional solution to `PC-Multiroute-LP`

We use the approach of Bienstock et al. [2] to relate the integrality gap of `PC-Multiroute-LP` to the integrality gap of `Multiroute-LP`.

Lemma 2.3.1. *Consider an instance $\langle G, w, \pi \rangle$ of the node-weighted PC-EC-SNDP where the input graph G belongs to a family $\mathcal{G}$ of graphs. Let (x, f, z) be a feasible fractional solution to* `PC-Multiroute-LP` *for this instance. There is a feasible integral solution H whose cost is at most*

$$2\gamma \left(\sum_v x(v)w(v) \right) + 2 \left(\sum_{st} z(st)\pi(st) \right) \leq 2\gamma \left(\sum_v x(v)w(v) + \sum_{st} z(st)\pi(st) \right),$$

where γ is an upper bound on the integrality gap of `Multiroute-LP` *on instances of the node-weighted SNDP problem in which the input graph G is in $\mathcal{G}$. Moreover, if there is a polynomial-time algorithm that rounds a fractional solution to* `Multiroute-LP` *to an integral solution of value at most ρ times the value of the fractional solution, then there is a polynomial time algorithm that rounds a fractional solution to* `PC-Multiroute-LP` *to an integral solution of cost at most 2ρ times the cost of the fractional solution.*

Proof: Let $I = \{st \mid z(st) > 1/2\}$. Consider the `Multiroute-LP` 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 $st \in J$, $z(st) \leq 1/2$.)

We can show that (x', f') is a feasible solution to the `Multiroute-LP` instance as follows. The solution clearly satisfies the first and third set of constraints. Therefore it suffices to verify that it also satisfies the second set of constraints. Consider a vertex $v \in V$ and a pair $st \in J$. Suppose that $x'(v) = 1$. We have

$$\sum_{\overline{p} \in \mathcal{P}_{st}^{r(st)}, v \in \overline{p}} f'(\overline{p}) \leq \sum_{\overline{p} \in \mathcal{P}_{st}^{r(st)}} f'(\overline{p}) = 1.$$

Therefore we may assume that $x'(v) = 2x(v)$. We have

$$\sum_{\overline{p} \in \mathcal{P}_{st}^{r(st)}, v \in \overline{p}} f'(\overline{p}) \leq 2 \sum_{\overline{p} \in \mathcal{P}_{st}^{r(st)}, v \in \overline{p}} f(\overline{p}) \leq 2x(v) = x'(v).$$

The first inequality follows from the fact that $z(st) \leq 1/2$ and the second inequality follows from the fact that (x, f, z) is a feasible solution to `PC-Multiroute-LP`.

Therefore (x', f') is a feasible solution to `Multiroute-LP`. Since the integrality gap of `Multiroute-LP` is γ, there is a subgraph H of weight at most $\gamma \sum_v w(v)x'(v)$ that satisfies the requirements of all the pairs in J. Since H satisfies the requirements of all pairs in J, the cost of H is at most

$$\gamma \sum_v x'(v)w(v) + \sum_{st \in I} \pi(st) \leq 2\gamma \sum_v x(v)w(v) + 2 \sum_{st} z(st)\pi(st) \leq 2\gamma \left(\sum_v x(v)w(v) + \sum_{st} z(st)\pi(st) \right).$$

□

In Theorem 2.3.8 we show that the integrality gap of Multiroute-LP is $O(k \log n)$. Following corollaries follow from the $O(k \log n)$ integrality gap of Multiroute-LP and the scaling method we describe in this section (see Theorem 2.3.1).

Corollary 2.3.2. *The integrality gap of PC-Multiroute-LP is $O(k \log n)$.*

Corollary 2.3.3. *There is an algorithm that takes as input a fractional solution (x, f, z) to PC-Multiroute-LP and in polynomial time it constructs a feasible integral solution whose cost is at most $O(k \log n)$ times the cost of the fractional solution.*

Corollary 2.3.4. *The integrality gap of PC-Multiroute-LP is $O(k)$ in minor-closed graph families, where the constant depends only on the family.*

Corollary 2.3.5. *Consider an instance of the node-weighted PC-EC-SNDP for which the input graph belongs to a minor-closed family $\mathcal{G}$. There is a polynomial time algorithm that takes as input a fractional solution (x, f, z) to PC-Multiroute-LP and it constructs a feasible integral solution whose cost is at most $O(k)$ times the cost of the fractional solution, where the constant depends only on the family $\mathcal{G}$.*

Now we turn our attention to the problem of approximating the node-weighted PC-EC-SNDP. We start by constructing an $O(k)$-approximate fractional solution to PC-Multiroute-LP; by Theorem 2.2.1, we can construct such a solution in polynomial-time. Once we have a fractional solution, we can use the algorithm guaranteed by Corollary 2.3.3 or Corollary 2.3.5 to construct a feasible integral solution. Therefore we have the following approximation guarantees for the problem.

Theorem 2.3.6. *There is an $O(k^2 \log n)$-approximation for node-weighted PC-EC-SNDP, where k is the maximum requirement between a pair of vertices.*

Theorem 2.3.7. *There is an $O(k^2)$-approximation for node-weighted PC-SNDP on minor-closed families of graphs, where k is the maximum requirement between a pair of vertices.*

2.3.1 Integrality gap of Multiroute-LP via Aug-LP

In this section, we show that the integrality gap of Multiroute-LP is $O(k)$ times the integrality gap of Aug-LP. In Section 3.3 we show that the integrality gap of ELC-Aug-LP which captures Aug-LP is $O(\log n)$.

Theorem 2.3.8. *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 EC-SNDP instance and the weight of H is $O(k \log n) \cdot$ OPT.*

By the fact that the integrality gap of ELC-Aug-LP is $O(\log n)$ we can similarly show that the integrality gap of ELC-Multiroute-LP is $O(k \log n)$.

Theorem 2.3.9. *Let* OPT *be the value of the optimal fractional solution to ELC-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 ELC-SNDP instance and the weight of H is $O(k \log n)\cdot$ OPT.*

Via primal-dual approach, Chekuri et al. [3] show that the integrality gap of Aug-LP that arise from instances of EC-SNDP for which the input graph belongs to a minor-closed family $\mathcal{G}$ is $O(1)$. Following theorems follow from the $O(1)$ integrality gap of Aug-LP in minor-closed families of graphs.

Theorem 2.3.10. *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 $\mathcal{G}$.*

Theorem 2.3.11. *Let* OPT *be the value of the optimal fractional solution to ELC-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 ELC-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 2.3.8 and Theorem 2.3.10, we use the augmentation framework that was introduced by Williamson et al. [27] for the edge-weighted EC-SNDP problem. Note that the theorems only upper bound the integrality gap of the relaxations; the algorithms for EC-SNDP are not based on solving them. The relaxations need to be solved for PC-EC-SNDP to identify the pairs to connect and reduce to EC-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 (see Theorem 1.3.2).

The algorithm selects a cover H of r in k phases. The algorithm maintains the invariant that the first ℓ phases have selected a graph H_ℓ that covers r_ℓ. During phase ℓ, the algorithm adds a new set of nodes to $H_{\ell-1}$ in order to get a graph H_ℓ that covers r_ℓ. More precisely, in phase ℓ, 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 ℓ. 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_ℓ of G'_ℓ that covers h_ℓ; once we have K_ℓ, we let H_ℓ be the subgraph of G induced by $V(H_{\ell-1}) \cup V(K_\ell)$.

Consider a phase ℓ. Recall that the goal is to cover h_ℓ using a subgraph of G'_ℓ. 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 ℓ.

$$\begin{aligned}
&\texttt{Aug-LP}(G'_\ell, h_\ell) \\
&\min \sum_{v \in V} w(v)x(v) \\
&\text{s.t.} \sum_{v \in \Gamma_{G'_\ell}(S)} x(v) \geq h_\ell(S) \quad \forall S \subseteq V \\
&x(v) \geq 0 \quad \forall v \in V
\end{aligned}$$

As shown in Lemma 2.3.12, for each phase of the algorithm, the optimal value of `Aug-LP` is at most the optimal value of `Multiroute-LP`.

Lemma 2.3.12. *Let (x, f) be a feasible solution to `Multiroute-LP`. For any phase ℓ, x is a feasible solution to `Aug-LP`(G'_ℓ, h_ℓ).*

Proof: Consider a set $S \subseteq V$ such that $h_\ell(S) = 1$. It follows from the definition of h_ℓ that S separates a pair st with requirement at least ℓ and $|\delta_{H_{\ell-1}}(S)| = \ell - 1$. Consider a tuple $\overline{p} \in \mathcal{P}_{st}^{r(st)}$. Since $|\delta_{H_{\ell-1}}(S)| = \ell - 1$ and $\overline{p}$ contains at least ℓ edge-disjoint s-t paths, there is at least one edge $e \in \overline{p}$ such that $e \in \delta_{G'_\ell}(S)$. Therefore, for any tuple $\overline{p} \in \mathcal{P}_{st}^{r(st)}$, there exists a vertex $v \in \Gamma_{G'_\ell}(S)$ such that $v \in \overline{p}$. It follows that

$$\sum_{v \in \Gamma_{G'_\ell}(S)} x(v) \geq \sum_{v \in \Gamma_{G'_\ell}(S)} \sum_{\overline{p} \in \mathcal{P}_{st}^{r(st)}, v \in \overline{p}} f(\overline{p}) \geq \sum_{\overline{p} \in \mathcal{P}_{st}^{r(st)}} f(\overline{p}) \geq 1$$

where the first and last inequalities follow from the fact that (x, f) is feasible for `Multiroute-LP` and the second inequality follows from the fact that each tuple $\overline{p} \in \mathcal{P}_{st}^{r(st)}$ contains a vertex of $\Gamma_{G'_\ell}(v)$. □

Corollary 2.3.13. *Suppose that for each phase ℓ, the integrality gap of `Aug-LP`(G'_ℓ, h_ℓ) is at most ρ. Then the integrality gap of `Multiroute-LP` is at most $k\rho$.*

The $O(\log n)$ integrality gap of `Aug-LP` (see Theorem 3.1.6) together with Corollary 2.3.13 imply an $O(k \log n)$-approximation for EC-SNDP (see Theorem 2.3.8). Similarly, the $O(k)$-approximation for EC-SNDP in minor-closed families of graphs (Theorem 2.3.10) follows from the $O(1)$ integrality gap of `Aug-LP` in minor-closed families of graphs (see Theorem 3.1.7) and Corollary 2.3.13.

Chapter 3

Improved Approach and Element-Connectivity

In this chapter, we consider node-weighted PC-SNDP in element-connectivity setup, PC-ELC-SNDP. We improve the approach introduced in Chapter 2 and give an $O(k \log n)$-approximation for PC-ELC-SNDP. Node-weighted ELC-SNDP is defined as follows. The input is an undirected node-weighted graph $G = (V, E)$ (weight of node v is denoted by $w(v)$) and a requirement function $r(st)$ for each pair of nodes. The vertices of G are partitioned into two sets, R and $V - R$; we refer to the vertices of R as **reliable** and to the vertices of $V - R$ as **non-reliable**. A vertex $s \in V$ is **terminal** if there is some vertex $t \in V$ such that $r(st) > 0$. Moreover, all terminals are in R. We assume that no two reliable vertices are adjacent in G; this assumption is without loss of generality, since we can subdivide each such edge using an element of weight zero. The goal is to find a minimum weight subgraph H of G such that H contains $r(st)$ element-disjoint paths between each pair s and t. We use k to denote the maximum requirement. Similar to PC-EC-SNDP, in prize-collecting ELC-SNDP the input in addition to that of ELC-SNDP, consists of penalties $\pi(st)$ for each pair of nodes s and t. 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 element disjoint paths in H between s and t is strictly less than $r(st)$ [1].

In Chapter 2, we described how to relate the integrality gap of `Multiroute-LP` to its augmentation counterpart, `Aug-LP`. Here we follow the same approach for the prize-collecting variant of `Aug-LP` in the element-connectivity setup which is called `PC-ELC-Aug-LP`. In this way, we are not required to solve the multiroute flow based relaxation of PC-ELC-SNDP and we save a factor of k in the approximation ratio.

3.1 Integrality gap of PC-ELC-SNDP via `PC-ELC-Aug-LP`

In this section, we show that there exists an $O(k \log n)$-approximation algorithm for node-weighted PC-ELC-SNDP. The ratio improves to $O(k)$ in minor-closed families of graphs.

Theorem 3.1.1. *There is an $O(k \log n)$-approximation algorithm for node-weighted PC-ELC-SNDP in undirected graphs which improves to an $O(k)$-approximation in minor-closed families of graphs.*

[1] We consider the all-or-nothing penalty model; one should pay the penalty even if the connectivity requirement is slightly violated.

In order to prove Theorem 3.1.1, we use the augmentation framework that was introduced by Williamson et al. [27] for the edge-weighted SNDP. The algorithm constructs a feasible solution to PC-ELC-SNDP in k phases. It selects a subgraph H of G and a set of pairs Z such that for each pair $st \notin Z$, H contains $r(st)$ element-disjoint st-paths. The set Z denotes the set of pairs whose connectivity requirement is not satisfied by H and we choose to pay their penalties instead.

Let $r_\ell : V \times V \to \mathbb{Z}_+$ be the requirement function of pairs in phase ℓ; for each pair st, $r_\ell(st) = \min\{\ell, r(st)\}$. The algorithm maintains the invariant that the first $\ell - 1$ phases have selected a subgraph $H_{\ell-1}$ and a set of pairs $Z_{\ell-1}$ such that for each pair $st \notin Z_{\ell-1}$, $H_{\ell-1}$ contains $r_\ell(st)$ element-disjoint st-paths. Let H_0 and Z_0 be empty sets. In the first $\ell - 1$ phases we have paid the cost of all vertices in $H_{\ell-1}$ and the penalty cost of all pairs in Z_ℓ.

Let $\mathcal{R}_\ell$ be the set of pairs with connectivity requirement at least ℓ; $\mathcal{R}_\ell = \{st | r(st) \geq \ell\}$. In phase ℓ, a pair st is called "unsatisfied" iff $st \in \mathcal{R}_\ell - Z_{\ell-1}$ and $H_{\ell-1}$ contains exactly $\ell - 1$ element-disjoint st-paths. In phase ℓ, we augment the connectivity of a set of unsatisfied pairs by at least one and pay the penalty of the rest of unsatisfied pairs. Since we have already paid for all vertices in $H_{\ell-1}$ and all pairs in $Z_{\ell-1}$, it is convenient to assume that for each $v \in H_{\ell-1}$, $w(v) = 0$ and for each $st \in Z_{\ell-1}$, $\pi(st) = 0$. Let $G'_\ell = (V, E(G) - E(H_{\ell-1}))$. In phase ℓ, the problem is to select a subgraph K_ℓ of G'_ℓ and a set of pairs Z'_ℓ such that for each pair $st \notin Z_{\ell-1} \cup Z'_\ell$, $K_\ell \cup H_{\ell-1}$ contains at least $r_\ell(st)$ element-disjoint st-paths and the weight of K_ℓ plus the total penalty of pairs in Z'_ℓ is minimized. Let $h_\ell : \mathcal{B}_{\text{ELC}} \to \{0, 1\}$ be the function that $h_\ell(\hat{S}) = 1$ iff $\hat{S}$ separates a pair $st \in \mathcal{R}_\ell - Z'_{\ell-1}$ and $|A' - A| + |\delta_{H_{\ell-1}}(\hat{S})| = \ell - 1$. Let $Z_\ell = Z_{\ell-1} \cup Z'_\ell$. By Menger's theorem on element-connectivity, the goal is to select K_ℓ and Z'_ℓ such that for each pair $st \in \mathcal{R}_\ell - Z_\ell$ and each biset $\hat{S} \in \mathcal{B}_{\text{ELC}}$ that separates st, $|\Gamma_{K_\ell}(\hat{S})| \geq h_\ell(\hat{S})$. At the end of phase ℓ, we pay the cost of the vertices in K_ℓ and the penalty cost of the pairs in Z'_ℓ. Moreover, we set $H_\ell = H_{\ell-1} \cup K_\ell$ and $Z_\ell = Z_{\ell-1} \cup Z'_\ell$.

Proposition 3.1.2. *Consider phase ℓ of the algorithm. The function h_ℓ is a biuncrossable function.*

Proof: Let $\hat{X}$ and $\hat{Y}$ be bisets that $h_\ell(\hat{X}) = h_\ell(\hat{Y}) = 1$. This implies that $r_\ell(\hat{X}) \geq \ell$ and $r_\ell(\hat{Y}) \geq \ell$. Since $\hat{X}, \hat{Y} \in \mathcal{B}_{\text{ELC}}$ and terminals are reliable nodes, terminals are either in $X \cap Y$, $X - Y'$, $Y - X'$ or $V - (X \cup Y)$ (see Figure 3.1). Let s_X, t_X be the pair separated by $\hat{X}$ and let s_Y, t_Y be the pair separated by $\hat{Y}$. Then it is straightforward to verify that either one of $\hat{X} \cap \hat{Y}$ and $\hat{X} \cup \hat{Y}$ and the other separates s_Y, t_Y or one of $\hat{X} - \hat{Y}$ and $\hat{Y} - \hat{X}$ separates s_X, t_X and the other separates s_Y, t_Y. Suppose that the former case holds (the other case is similar). By bisubmodularity of $|S' - S| + |\delta_{H_{\ell-1}}(\cdot)|$ (see Lemma 1.3.10 and 1.3.11), $h_\ell(\hat{X}) + h_\ell(\hat{Y}) \leq h_\ell(\hat{X} \cup \hat{Y}) + h_\ell(\hat{X} \cap \hat{Y})$ which implies that $h_\ell(\hat{X} \cup \hat{Y}) = h_\ell(\hat{X} \cap \hat{Y}) = 1$. □

In Theorem 3.1.6 and 3.1.7 we show that in phase ℓ, we can select a feasible solution (K_ℓ, Z_ℓ) whose cost is at most $O(\log n) \cdot$ OPT, where OPT is the value of an optimal solution of PC-ELC-SNDP; this ratio improves to $O(1)$ on instances for which the input graphs is in a minor-closed family of graphs $\mathcal{G}$. Thus at the end of phase k

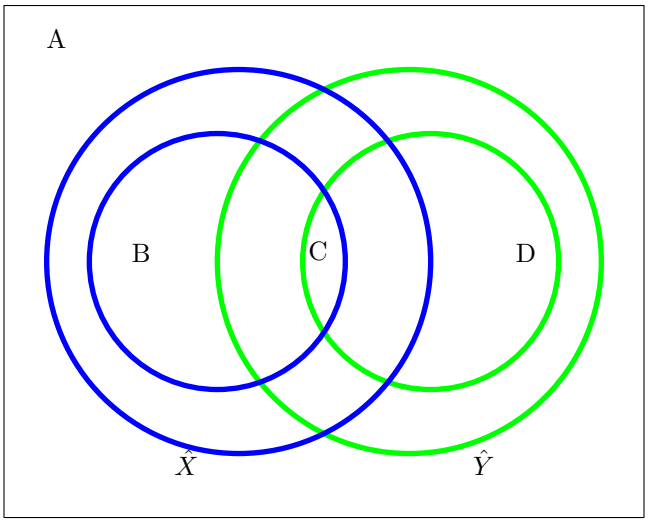

Figure 3.1: *Terminals are either in A, B, C or D.*

we have a feasible solution to PC-ELC-SNDP such that the weight of H_k plus the total penalty of all pairs in Z_k is $O(k \log n)\cdot$ OPT in general graphs and $O(k)\cdot$ OPT in minor-closed families of graphs.

Consider phase ℓ. Let $\Gamma_{G'_\ell}(\hat{S})$ be the vertex neighborhood of $\hat{S}$; $\Gamma_{G'_\ell}(\hat{S}) = \{v \in V - S' | \exists uv \in E(G'_\ell), u \in S\}$. The LP relaxation of the augmentation problem in phase ℓ is as follows.

$$\texttt{PC-ELC-Aug-LP}(G'_\ell, h_\ell)$$

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

$$\text{s.t.} \sum_{v \in \Gamma_{G'_\ell}(\hat{S})} x(v) \geq h_\ell(\hat{S})(1 - z(st)) \quad \forall st \in \mathcal{R}_\ell, \forall \hat{S} \in \mathcal{B}_{\texttt{ELC}}, s \in S, t \in V - S'$$

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

$$z(st) \geq 0 \quad \forall st \in V \times V$$

Lemma 3.1.3. *$\texttt{PC-ELC-Aug-LP}(G'_\ell, h_\ell)$ is a valid relaxation of the augmentation problem of PC-ELC-SNDP on input graph G in phase ℓ.*

Proof: Let (H, Z) be a feasible solution for the augmentation problem of PC-ELC-SNDP in phase ℓ. Define (x, z) such that $x(v) = 1$ if $v \in H$ and $x(v) = 0$ otherwise and $z(st) = 1$ if $(s, t) \in Z$ and zero otherwise. Consider a biset $\hat{S}$ that $h_\ell(\hat{S}) = 1$ and $\hat{S}$ separates a pair $st \in \mathcal{R}_\ell - Z$ (not that since we set $\pi(st) = 0$ for each $(s, t) \in Z_{\ell-1}$, $Z_{\ell-1} \subseteq Z$). Since $h_\ell(\hat{S}) = 1$, $r(st) \geq \ell$ and $|S' - S| + |\delta_{H_{\ell-1}}(\hat{S})| = \ell - 1$. Since (H, Z) is a feasible solution to the

augmentation problem in phase ℓ, H contains at least ℓ element-disjoint st-paths. This implies that for each $st \in \mathcal{R}_\ell$,

$$\sum_{x \in \Gamma_{G'_\ell}(\hat{S})} x(v) \geq \ell - |\delta_{H_{\ell-1}}(\hat{S})| \geq 1 \geq h_\ell(\hat{S})(1 - z(st)).$$

□

Following lemma shows that in each phase ℓ, the cost of an optimal solution to PC-ELC-Aug-LP(G'_ℓ, h_ℓ) is not more than the cost of an optimal solution of PC-ELC-SNDP.

Lemma 3.1.4. *Let subgraph H and set of pairs Z be a feasible solution of PC-ELC-SNDP. For each phase $\ell \leq k$, H along with Z is a feasible solution to* PC-ELC-Aug-LP(G'_ℓ, h_ℓ).

Proof: For each $v \in V$, let $x(v) = 1$ if $v \in V(H)$ and zero otherwise. Similarly, let $z(st) = 1$ if $st \in Z$ and zero otherwise. The pair (x, z) corresponds to the solution (H, Z). Clearly (x, z) satisfy the non-negativity constraints of PC-ELC-Aug-LP(G'_ℓ, h_ℓ). We only need to show that (x, z) satisfy the first set of constraints as well. Consider a biset $\hat{S} \in \mathcal{B}_{\text{ELC}}$ such that $h_\ell(\hat{S}) = 1$. Since $h_\ell(\hat{S}) = 1$, $\hat{S}$ separates a pair st with $r(st) \geq \ell$ and $|\delta_{H_{\ell-1}}(\hat{S})| + |S' - S| = \ell - 1$. If $z(st) = 1$ for st, the right-hand side of the corresponding constraint becomes 0 and it trivially holds. Consider the case that $z(st) = 0$. Note that $|\delta_{H_{\ell-1}}(\hat{S})| + |S' - S| = \ell - 1$ and $S' - S$ only contains non-reliable vertices. Since $st \in \mathcal{R}_\ell - Z$, H contains ℓ element-disjoint st paths. Thus there is at least one edge of $\delta_H(\hat{S})$ that is in $\delta_{G'_\ell}(\hat{S})$. Therefore, $|\Gamma_{G'_\ell}(\hat{S}) \cap V(H)|$ is nonempty and it follows that for each $\hat{S} \in \mathcal{B}_{\text{ELC}}$ separating a pair $st \in \mathcal{R}_\ell - Z$,

$$\sum_{v \in \Gamma_{G'_\ell}(\hat{S})} x(v) \geq 1.$$

Thus (x, z) (or equivalently H and Z) is a feasible solution to PC-ELC-Aug-LP(G'_ℓ, h_ℓ). □

Corollary 3.1.5. *Let G be a node-weighted graph from a family of graphs $\mathcal{G}$. Suppose that in each phase ℓ, there is an approximation algorithm that finds an integral ρ-approximate solution to* PC-ELC-Aug-LP(G'_ℓ, h_ℓ). *Then there is a $(k\rho)$-approximation for PC-ELC-SNDP on graphs from family $\mathcal{G}$.*

Thus it suffices to bound the integrality gap of PC-ELC-Aug-LP(G'_ℓ, h_ℓ). Following the scaling method of Bienstock et al. [2] and similar to Theorem 2.3.1, in Theorem 3.1.9 we show that a λ-approximation to ELC-Aug-LP gives a 2λ-approximation to PC-ELC-Aug-LP. Thus it is enough to upper bound the integrality gap of ELC-Aug-LP. We prove Theorem 3.1.6 in section 3.3.

Theorem 3.1.6. *In each phase ℓ, the integrality gap of* ELC-Aug-LP(G'_ℓ, h_ℓ) *is $O(\log n)$. Moreover, there is a polynomial time algorithm that selects a subgraph K_ℓ of G'_ℓ that covers h_ℓ and the weight of K_ℓ is at most $O(\log n)$ times the weight of the optimal solution to* ELC-Aug-LP(G'_ℓ, h_ℓ).

You can find the proof of Theorem 3.1.7 in the long version of [3].

Theorem 3.1.7 (Long version of [3])**.** *Suppose that G belongs to a minor-closed family $\mathcal{G}$. In each phase ℓ, the integrality gap of $\texttt{ELC-Aug-LP}(G'_\ell, h_\ell)$ is a constant that only depends on the family $\mathcal{G}$. Moreover, there is a polynomial time algorithm that selects a subgraph K_ℓ of G'_ℓ covers h_ℓ and the weight of K_ℓ is at most $O(1)$ times the weight of the optimal solution to $\texttt{ELC-Aug-LP}(G'_\ell, h_\ell)$.*

An important observation about the constraints of ELC-Aug-LP instances derived from ELC-SNDP is that if $h_\ell(\hat{S}) = 1$ then $S' - S$ is a subset of zero-weight vertices, $S' - S \subseteq V(H_{\ell-1})$.

Lemma 3.1.8. *Consider $\texttt{ELC-Aug-LP}(G'_\ell, h_\ell)$ and let $\hat{S} = (S, S')$ be a biset such that $h(\hat{S}) = 1$. Then, $S' - S \subseteq V(H_{\ell-1})$.*

Proof: Since $h_\ell(\hat{S}) = 1$, $r(\hat{S}) \geq \ell$ and $|S' - S| + |\delta_{H_{\ell-1}}(\hat{S})| = \ell - 1$. Let $T = (S' - S) \cap (V(G) - V(H_{\ell-1}))$. Suppose for contradiction that T is a non-empty set and let $\hat{Y} = (S, S' - T)$. Since all terminals have zero weight, $\hat{Y}$ separates all terminal pairs separated by $\hat{S}$ and $r(\hat{Y}) = r(\hat{S}) \geq \ell$. Note that since the vertices of T have non-zero weight, $H_{\ell-1}$ does not have any edge incident to a vertex of T. Therefore $\delta_{H_{\ell-1}}(\hat{S}) = \delta_{H_{\ell-1}}(\hat{Y})$. Thus $|S' - S - T| + |\delta_{H_{\ell-1}}(\hat{Y})| = \ell - 1 - |T| < \ell - 1$, which contradicts the fact that at the end of phase $\ell - 1$, $|S' - S| + |\delta_{H_{\ell-1}}(\hat{S})| \geq \ell - 1$ for each $\hat{S} \in \mathcal{P}$ with $r(\hat{S}) \geq \ell$. □

Suppose that $\texttt{PC-ELC-Aug-LP}(G'_\ell, h_\ell)$ is solvable in polynomial time. Then by the scaling method of Bienstock et al. [2], the integrality gap of $\texttt{PC-ELC-Aug-LP}(G'_\ell, h_\ell)$ is within a constant factor of the integrality gap of $\texttt{ELC-Aug-LP}(G'_\ell, h_\ell)$ (see Lemma 3.1.9) and by Corollary 3.1.5 and Theorem 3.1.6, we have an $O(k \log n)$-approximation for node-weighted PC-ELC-SNDP. Similarly by Corollary 3.1.5 and Theorem 3.1.7, there is an $O(k)$-approximation algorithm for node-weighted PC-ELC-SNDP on minor-closed families of graphs. Theorem 3.1.17 shows that we can find an optimal fractional solution to $\texttt{PC-ELC-Aug-LP}(G'_\ell, h_\ell)$ in polynomial time.

Lemma 3.1.9. *Suppose that the integrality gap of $\texttt{ELC-Aug-LP}(G'_\ell, h_\ell)$ is α and $\texttt{PC-ELC-Aug-LP}(G'_\ell, h_\ell)$ is solvable in polynomial time. Then the integrality gap of $\texttt{PC-ELC-Aug-LP}(G'_\ell, h_\ell)$ is $O(\alpha)$. Furthermore if there is an algorithm that finds an integral ρ-approximate solution to $\texttt{ELC-Aug-LP}(G'_\ell, h_\ell)$ in polynomial time, we can find an integral solution to $\texttt{PC-ELC-Aug-LP}(G'_\ell, h_\ell)$ in polynomial time whose cost is at most $O(\rho)$.*OPT*, where* OPT *is the cost of the optimal solution to $\texttt{PC-ELC-Aug-LP}(G'_\ell, h_\ell)$.*

Proof: Let (x, z) be a feasible fractional solution to $\texttt{PC-ELC-Aug-LP}(G'_\ell, h_\ell)$. Let $I = \{st \mid z(st) > 1/2\}$. Consider the $\texttt{ELC-Aug-LP}(G'_\ell, h'_\ell)$ instance that we get from the prize-collecting instance by setting the requirement of the pairs in I to zero. Let J be the set of all pairs not in I. $h'_\ell(\hat{S}) = 1$ iff $h_\ell(\hat{S}) = 1$ and $\hat{S}$ separates a pair $st \in J$ and zero otherwise. Similar to h_ℓ, we can show that h'_ℓ is a biuncrossable function. Let x' be the following vector. For each vertex $v \in V$, we set $x'(v) = \min\{1, 2x(v)\}$.

We can show that x' is a feasible solution to ELC-Aug-LP(G'_ℓ, h'_ℓ). The solution clearly satisfies $x'(v) > 0$ for all $v \in V(G'_\ell)$. Therefore it suffices to verify that it covers h'_ℓ. Consider a biset $\hat{S} \in \mathcal{B}_{\text{ELC}}$ and an unsatisfied pair $st \in J$. Since (x, z) is a feasible solution for the prize-collecting instance,

$$\sum_{v \in \Gamma_{G'_\ell}(\hat{S})} x(v) \geq h_\ell(\hat{S})(1 - z(st)).$$

If for a node $v \in \Gamma_{G'_\ell}(\hat{S})$, $x'(v) = 1$ then,

$$\sum_{v \in \Gamma_{G'_\ell}(\hat{S})} x'(v) \geq 1 \geq h'_\ell(\hat{S}).$$

Therefore we may assume that $x'(v) = 2x(v)$ for all $v \in \Gamma_{G'_\ell}(\hat{S})$. We have

$$\sum_{v \in \Gamma_{G'_\ell}(\hat{S})} x'(v) = \sum_{v \in \Gamma_{G'_\ell}(\hat{S})} 2x(v) \geq 2h_\ell(\hat{S})(1 - z(st)) \geq h_\ell(\hat{S}) \geq h'_\ell(\hat{S}).$$

The first inequality follows from the fact that (x, z) is a feasible solution to PC-ELC-Aug-LP(G'_ℓ, h_ℓ) and the second inequality follows from the fact that $z(st) \leq 1/2$ for all $st \in J$. Therefore x' is a feasible solution to ELC-Aug-LP(G'_ℓ, h'_ℓ). This implies that the integrality gap of PC-ELC-Aug-LP(G'_ℓ, h_ℓ) is at most 2α. If there exists an algorithm that finds an integral ρ-approximate solution of ELC-Aug-LP(G'_ℓ, h'_ℓ), there is a subgraph H of weight at most $\rho \sum_v w(v)x'(v)$ that satisfies the connectivity requirements of all pairs in J. Thus the cost of H plus the penalty of the pairs in I is at most

$$\rho \sum_v x'(v)w(v) + \sum_{st} \pi(st) \leq 2\rho \sum_v x(v)w(v) + 2 \sum_{st} z(st)\pi(st) \leq 2\rho \left(\sum_v x(v)w(v) + \sum_{st} z(st)\pi(st) \right).$$

Thus the subgraph H along with the set I is an $O(\rho)$-approximation solution for PC-ELC-Aug-LP(G'_ℓ, h_ℓ). □

Next, we show that PC-ELC-Aug-LP is solvable in polynomial time. In general, PC-ELC-Aug-LP has exponentially many constraints and to solve it in polynomial time we need to design a separation oracle and apply ellipsoid method. First we describe a separation oracle for ELC-Aug-LP and then we show how to modify the oracle so that it works for PC-ELC-Aug-LP as well.

Theorem 3.1.10. *Consider phase ℓ of the augmentation framework. There is an algorithm that finds an optimal fractional solution to ELC-Aug-LP(G'_ℓ, h_ℓ) in polynomial time.*

$$\texttt{ELC-Aug-LP}(G'_\ell, h_\ell)$$

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

$$\text{s.t.} \sum_{v \in \Gamma_{G'_\ell}(\hat{S})} x(v) \geq h_\ell(\hat{S}) \quad \forall \hat{S} \in \mathcal{B}_{\texttt{ELC}}$$

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

Given a fractional solution x to $\texttt{ELC-Aug-LP}(G'_\ell, h_\ell)$, we define the function g_ℓ as follows. For each biset $\hat{S} \in \mathcal{B}_{\texttt{ELC}}$, let $g_\ell(\hat{S}, x) = |\delta_{H_{\ell-1}}(\hat{S})| + |S' - S| + \sum_{v \in \Gamma_{G'_\ell}(S)} x(v)$. By the constraint of $\texttt{ELC-Aug-LP}$, a biset $\hat{S}$ is violated *w.r.t.* x if $g_\ell(\hat{S}, x) < r_\ell(\hat{S})$. To decide whether x is a feasible solution to $\texttt{ELC-Aug-LP}(G', h_\ell)$ we need to check whether there exists a biset $\hat{S} \in \mathcal{B}_{\texttt{ELC}}$ such that $g_\ell(\hat{S}, x) < r_\ell(\hat{S})$. The idea is to construct a directed graph G^{st}_x for each pair st with $r(st) \geq \ell$ such that the value of st-min-cut in G^{st}_x is equal to $\min_{\hat{S} \in \mathcal{P}, s \in S, t \in V - S'} g_\ell(\hat{S}, x)$. This implies that there is a polynomial time algorithm to check whether x is a feasible fractional solution of $\texttt{PC-ELC-Aug-LP}(G'_\ell, h_\ell)$ and report a violated constraint if there exists any.

We assume that $0 \leq x(v) \leq 1$ for each $v \in V(G)$; otherwise, we can simply find a violated constraint. Since $w(v) = 0$ for each $v \in V(H_{\ell-1})$, we can assume that $x(v) = 1$ for each v in $H_{\ell-1}$. For each pair st with $r(st) \geq \ell$, we construct a directed graph G^{st}_x with edge capacities as follows. Each edge $uv \in E(G)$ is replaced with two *directed arcs* $u \to v$ and $v \to u$ both with unit capacity ($c(u \to v) = c(v \to u) = 1$). Moreover, we replace each vertex $v \in V - \{s, t\}$ with two nodes v_i and v_o such that v_i serves as the entrance gate and v_o serves as the exit gate of v (all incoming arcs of v are incident to v_i and all outgoing arcs of v are incident to v_o) and an arc $v_i \to v_o$ connects v_i to v_o. The capacity of each arc $v_i \to v_o$ is defined as follows.

$$c(v_i \to v_o) = \begin{cases} x(v) & \text{if } v \in V(G) - V(H_{\ell-1}) \\ \infty & \text{if } v \in V(H_{\ell-1}) \cap R \\ 1 & \text{if } v \in V(H_{\ell-1}) - R \end{cases}$$

There are two types of arc in G^{st}_x: the arcs representing nodes of G, $v_i \to v_o$, and the arcs representing edges of G, $v_o \to u_i$. We call the first type, "node" arcs and the other one, "edge" arcs. In the following, we use $\delta^-(S)$ to denote the set of all incoming arcs of S, that is, arcs $u \to v$ where $u \notin S, v \in S$. We use $\delta^+(S)$ to denote the set of all outgoing arcs of S, that is, arcs $v \to u$ where $u \notin S, v \in S$.

For a set $S \subset V(G^{st}_x)$, $c(S)$ denote the value of the cut induced by S in G^{st}_x which is equal to $\sum_{a \in \delta^+_{G^{st}_x}(S)} c(a)$.

Remark 3.1.11. *Note that we consider terminal s (and similarly t) as a node representing both entrance and exit gates of s; $s = s_i = s_o$.*

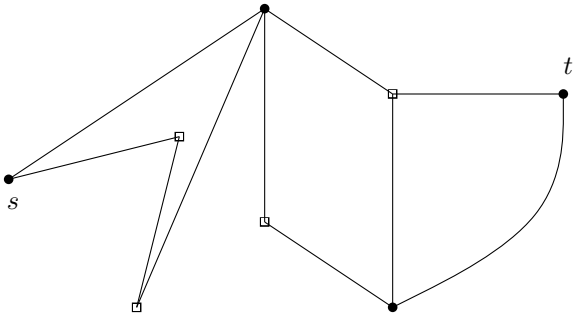

Figure 3.2: Undirected graph G with two types of vertices: reliable vertices and non-reliable ones. Circles denote reliable vertices and squares denote non-reliable ones.

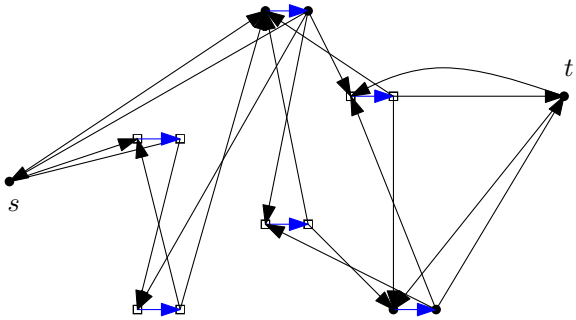

Figure 3.3: The corresponding G_x^{st} of the graph in Figure 3.2.

In following, we define *simple st-cut* in G_x^{st}. In addition, we show that for each violated biset $\hat{S} \in \mathcal{B}_{\texttt{ELC}}$ *w.r.t.* x that separates s and t there exists a simple st-cut S_x in G_x^{st} such that $c(S_x) = g_\ell(\hat{S}, x)$.

Definition 3.1.12. *In directed graph G_x^{st}, an st-cut S is a simple st-cut if:*

1. *If $v_i \to v_o$ crosses S and $v \in V(H_{\ell-1})$, then v is a non-reliable node.*
2. *If $v_o \in S$ then v_i is in S too.*
3. *If $v_o \to u_i$ crosses S, both u and v are zero-weight.*
4. *If $v_i \to v_o$ crosses S and $v \notin V(H_{\ell-1})$, there exists a node $u_o \in S$ such that $u_o v_i \in E(G_x^{st})$.*

In the following proposition we show that there exists an st-min-cut of G_x^{st} that is a simple st-cut.

Proposition 8. *Let G_x^{st} be the directed graph corresponding to a pair of nodes st and a fractional solution x to* `ELC-Aug-LP`(G, h). *There exists a simple st-cut S' that is an st-min-cut of G_x^{st}. Morover, we can find a simple st-min-cut in ploynomial time.*

Proof: We assume that G_x^{st} has a finite cut; otherwise, the proposition trivially holds. Consider an st-min-cut of G_x^{st}, S. We show that in polynomial time we can construct a simple st-cut S' such that $c(S') = c(S)$.

1. Since the capacity of a "node" arc representing a zero-weight reliable node is infinity, no such arc crosses S.
2. Suppose that there exists a node $v \in V(G)$ such that $v_o \in S$ and $v_i \notin S$. Consider $S - \{v_o\}$. Since the only incoming arc of v_o is $v_i v_o$ and $v_i \notin S$, $c(S - \{v_o\}) = c(S)$. Let $S_1 = S - \{v_o | v_o \in S, v_i \notin S\}$; $c(S_1) = c(S)$ and S_1 satisfies properties 1 and 2.
3. Then, we continue with S_1. Suppose that there exists an arc $v_o u_i$ crossing S_1 such that at least one of v and u is not in $V(H_{\ell-1})$. Assume that $v \notin V(H_{\ell-1})$; the other case is similar. Since the second property of simple

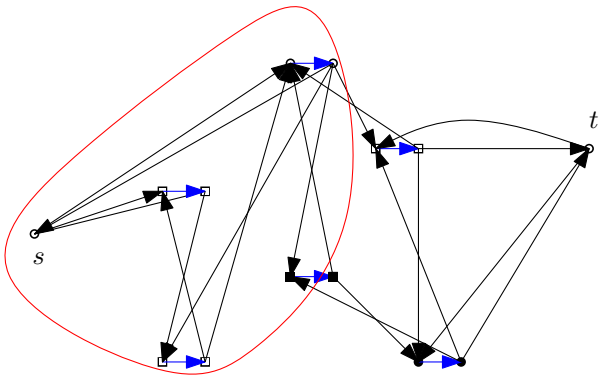

Figure 3.4: *A simple cut in G_x^{st}. White vertices denote the vertices in $V(H_{\ell-1})$.*

cuts holds for S_1, $v_i \in S_1$ too. Since the capacity of an "edge" arc is *one* and the capacity of a "node" arc corresponding to a vertex $v \notin V(H_{\ell-1})$ is $x(v) \leq 1$, $c(S_1 - \{v_o\}) \leq c(S_1)$ (in the case that $u \notin V(H_{\ell-1})$, we have $c(S_1 \cup \{u\}) \leq c(S_1)$). Since v_i is not a zero-weight vertex, this update does not violate properties 1 and 2. Note that each time we perform this step the number of "edge" arcs of the cut decreases; thus this update terminates in a polynomial time. We keep on performing this update until we reach a set S_2 such that $c(S_2) \leq c(S_1)$ and S_2 satisfies properties 1 to 3.

4. Now, we proceed with set S_2. Suppose that there exists an arc $v_i v_o$ crossing S_2 that $v \in V(G) - V(H_{\ell-1})$. If there exists no $u_o \in S_2$ such that $u_o v_i \in E(G_x^{st})$, $c(S_2 - \{v_i\}) \leq c(S_2)$; we can remove v_i from S_2 without increasing the cut value. We repeat this update until we obtain a set S_3 that satisfies properties 1 to 4 and $c(S_3) \leq c(S_2)$. Note that since v_i is removed from S_2 only if $v_o \notin S_2$ and $v_i \notin \delta_{G_x^{st}}(S_2)$, this update does not violate the properties 1 and 3. Moreover, since each time we perform this update the value of $|S_2|$ decreases, this step terminates in polynomial time.

The set $S' = S_3$ is a simple st-cut such that $c(S') \leq c(S)$. Our approach implies that we can find a simple st-min-cut in polynomial time. □

Now, we are ready to prove the main lemma that leads to a separation oracle for $\texttt{ELC-Aug-LP}(G', h_\ell)$. Lemma 3.1.13 and 3.1.14 shows that the value of an st-min-cut in G_x^{st} determines whether there exists a violated biset $\hat{S}$ *w.r.t.* x separating s and t.

Lemma 3.1.13. *Consider phase ℓ of the algorithm and let x be a fractional solution to $\texttt{ELC-Aug-LP}(G'_\ell, h_\ell)$. For each violated biset $\hat{S} = (S, S')$* w.r.t. *x, there exists a pair st and a simple st-min-cut S_x such that $c(S_x) = g_\ell(\hat{S}, x)$.*

Proof: Since $\hat{S}$ is a violated biset *w.r.t.* x, $h_\ell(\hat{S}) = 1$ and it separates a pair st with $r(st) \geq \ell$. We assume that $s \in S$ and construct S_x as follows. Let $S_x^1 = \{v_i, v_o | v \in S - \{s\}\}$, $S_x^2 = \{v_i | v \in S' - S\} \cup \{v_i | v \in \Gamma_{G'_\ell}(\hat{S})\}$. We define $S_x = S_x^1 \cup S_x^2 \cup \{s\}$.

First we show that S_x is a simple st-cut. An arc $v_i \to v_o$ crosses S_x if $v \in S' - S$ or $v \in \Gamma_{G'_\ell}(\hat{S})$. Since $\hat{S}$ is a violated biset *w.r.t.* x, $\Gamma_{G'_\ell}(\hat{S})$ contains no reliable node; moreover, $S' - S$ only contains non-reliable nodes ($\hat{S} \in \mathcal{P}$). Thus v is not a zero-weight reliable vertex and the first property holds. Our construction insures the second property; $v_i \in S_x$ if $v_o \in S_x$. Moreover, if $v_i \to v_o$ crosses S_x and v is not a zero-weight vertex, then v is either in $S' - S$ or $\Gamma_{G'_\ell}(\hat{S})$. Since $S' - S$ consists of zero-weight vertices (Lemma 3.1.8), $v \in \Gamma_{G'_\ell}(\hat{S})$. Thus there exists a node $u \in S$ such that $uv \in E(G'_\ell)$ and property 4 holds. Finally suppose that an arc $v_o \to u_i$ crosses S_x. This implies that $v \in S$, $u \in V - (S' \cup \Gamma_{G'_\ell}(\hat{S}))$. Since $v \in S$ and $u \in \Gamma_G(\hat{S}) - \Gamma_{G'_\ell}(\hat{S})$. Thus both u and v are zero-weight ($u \in \Gamma_{H_{\ell-1}}(\hat{S})$) and the third property holds too. Hence S_x is a simple st-cut.

Next, we show that $c(S_x) = g_\ell(\hat{S}, x)$. Since S_x is a simple st-cut, it only contains three types of arc:

$u_o \to v_i$ where both u and v have zero weight	$\iff$	$uv \in \Gamma_{H_{\ell-1}}(\hat{S})$
$v_i \to v_o$ where v is a zero-weight non-reliable vertex	$\iff$	$v \in S' - S$
$v_i \to v_o$ where $v \in \Gamma_{G'_\ell}(\hat{S})$ and v has non-zero weight	$\iff$	$v \in \Gamma_{G'_\ell}(\hat{S})$

For the first type, if $u_o \in S_x$ and $v_i \in V(G^{st}_x) - S_x$ we can conclude that $u \in S$ and $v \in V(G) - S'$ and since both have zero weight, $uv \in \delta_{H_{\ell-1}}(\hat{S})$. Furthermore, for each $uv \in \delta_{H_{\ell-1}}(\hat{S})$ the arc $u_o v_i$ crosses S_x. Since $u_o \to v_i$ is an "edge" arc and the capacity of "edge" arcs is one, $|\delta_{H_{\ell-1}}(\hat{S})| = \sum_{u_o \to v_i \in \delta_{G^{st}_x}(S_x)} c(u_o \to v_i)$.

For the second type, by Lemma 3.1.8, for each $v \in S' - S$, v is a zero-weight non-reliable node and $v_i \to v_o$ crosses S_x. Since $c(v_i \to v_o) = 1$ for all zero-weight non-reliable vertices, $|S' - S| = \sum_{\substack{v_i \to v_o \in \delta_{G^{st}_x}(S_x) \\ w(v)=0}} c(v_i \to v_o)$.

And for the last type, if $v_i \to v_o$ crosses S_x and it corresponds to a non zero-weight vertex v, $v \in \Gamma_{G'_\ell}(\hat{S})$. Moreover, for each $v \in \Gamma_{G'_\ell}(\hat{S})$, $v_i \in S_x$ and $v_o \in V(G_x) - S_x$. Since $\hat{S}$ is a violated biset *w.r.t.* x, v is a non-zero weight vertex. Thus $\sum_{v \in \Gamma_{G'}(\hat{S})} x(v) = \sum_{\substack{v_i v_o \in \delta_{G^{st}_x}(S_x) \\ w(v) \neq 0}} c(v_i v_o)$.

These all together imply that $c(S_x) = g_\ell(\hat{S}, x)$. □

Lemma 3.1.14. *Consider phase ℓ of the algorithm. For each finite simple st-cut S_x in G^{st}_x where $r(st) \geq \ell$, there exists a biset $\hat{S}$ such that $h_\ell(\hat{S}) = 1$ and $g_\ell(\hat{S}, x) = c(S_x)$.*

Proof: Consider a simple st-cut S_x where $r(st) \geq \ell$. Construct the biset $\hat{S} = (S, S')$ as follows. Let $S = \{v | v_i, v_o \in S_x\} \cup \{s\}$ and let $S_{bdy} = \{v | v_i \in S_x, v_o \notin S_x, w(v) = 0\}$. Define $S' = S \cup S_{bdy}$. Since S_x is a simple st-cut, the first property of simple cuts implies that S_{bdy} only contains zero-weight non-reliable vertices (see Definition 3.1.12). By a similar argument to the proof of Lemma 3.1.13, we can show that $|\delta_{H_{\ell-1}}(\hat{S})| = \sum_{v_o u_i \in \delta_{G^{st}_x(S_x)}} c(v_o u_i)$, $|S' - S| = \sum_{\substack{v_i v_o \in \delta_{G^{st}_x}(S_x) \\ w(v)=0}} c(v_i v_o)$ and $\sum_{v \in \Gamma_{G'_\ell}(\hat{S})} x(v) = \sum_{\substack{v_i v_o \in \delta_{G^{st}_x}(S_x) \\ w(v) \neq 0}} c(v_i v_o)$. Thus $c(S_x) = g_\ell(\hat{S}, x)$ □

Now, we are ready to present a polynomial time separation oracle for `ELC-Aug-LP` relaxation.

Lemma 3.1.15. *ELC-Aug-LP(G'_ℓ, h_ℓ) has a polynomial time separation oracle.*

Proof: We assume that $0 \leq x(v) \leq 1$ for each $v \in V$ in the given fractional solution x; otherwise we can simply find a violated constraint. Furthermore we can assume that $x(v) = 1$ for all zero-weight vertices in G'_ℓ.

By the definition of g_ℓ, if there exists a violated biset $\hat{S}$ *w.r.t.* x then $g_\ell(\hat{S}, x) < \ell$. For each pair st with $r(st) \geq \ell$ we construct G_x^{st}. If for each G_x^{st} the value of st-min-cut is not less than ℓ then x is a feasible solution (Lemma 3.1.13 and Proposition 8). Otherwise, if there exists a pair st such that the value of an st-min-cut in G_x^{st} is $\ell' < \ell$, by Proposition 8 we can find a simple st-cut whose value is ℓ'. Furthermore, Lemma 3.1.14 implies that a biset $\hat{S}$ such that $g_\ell(\hat{S}, x) = \ell' < \ell$. Consequently, the corresponding constraint of $\hat{S}$ is violated *w.r.t.* x. Since the number of st pairs is $O(n^2)$ and for each pair we run a polynomial time algorithm, the whole algorithm runs in a polynomial time. □

Thus we can solve ELC-Aug-LP(G'_ℓ, h_ℓ) via ellipsoid method by the separation oracle guaranteed in Lemma 3.1.15. Similarly, we are able to find an optimal fractional solution to PC-ELC-Aug-LP(G'_ℓ, h_ℓ) in polynomial time.

Lemma 3.1.16. *PC-ELC-Aug-LP(G'_ℓ, h_ℓ) has a polynomial time separation oracle.*

Proof: A fractional solution x is a feasible solution to PC-ELC-Aug-LP(G', h_ℓ) iff $0 \leq x(v) \leq 1$ for all $v \in V(G')$ and for each biset $\hat{S}$ where $h_\ell(\hat{S}) = 1$ and $\hat{S}$ separates s and t, $\sum_{v \in \Gamma_{G'}(\hat{S})} x(v) \geq 1 - z(st)$. The non-negativity constraints can be easily checked in polynomial time. Thus it is enough to check the constraints of type $\sum_{v \in \Gamma_{G'}(\hat{S})} x(v) \geq 1 - z(st)$ for each $\hat{S}$ with $h_\ell(\hat{S}) = 1$ that separates a terminal pair st with $r(st) \geq \ell$. Similar to Lemma 3.1.13, x is a feasible solution if for each pair st with $r(st) \geq \ell$, the value of st-min cut in G_x^{st} is at least $\ell - z(st)$. By Lemma 3.1.14 and similar to the proof of Lemma 3.1.15, we can output a violated biset $\hat{S}$ if $\hat{S}$ is not a feasible solution *w.r.t.* x. Since the number of st pairs is $O(n^2)$ and for each pair we run a polynomial time algorithm, the whole algorithm runs in a polynomial time. □

Theorem 3.1.17. *Consider phase ℓ of the augmentation framework. There is an algorithm that finds an optimal solution to PC-ELC-Aug-LP(G'_ℓ, h_ℓ).*

3.2 Approximation Algorithm for Node-weighted PC-VC-SNDP

In this section we show that there is an $O(k^4 \log^2 n)$-approximation for node-weighted PC-VC-SNDP which improves to $O(k^4 \log n)$ in minor-closed families of graphs. Chuzhoy and Khanna [6] show that in order to solve an instance of VC-SNDP, it is enough to solve $O(k^3 \log n)$ instances of ELC-SNDP obtained from the VC-SNDP instance. In this section, we combine the scaling method of Bienstock et al. [2] and Chuzhoy's and Khanna's reduction to give an $O(k^4 \log^2 n)$-approximation for node-weighted PC-VC-SNDP.

Similar to the initial step of the algorithms we proposed for PC-SNDP and PC-ELC-SNDP earlier in Section 2.3 and Section 3.1, first we reduce the given node-weighted PC-VC-SNDP instance to a non prize-collecting instance by solving an LP-relaxation of node-weighted PC-VC-SNDP. The relaxation we consider for the node-weighted PC-VC-SNDP (and VC-SNDP) is based on multiroute flows and we refer it as `PC-VC-Multiroute-LP` (and `VC-Multiroute-LP`). In Chapter 2, we proved that `Multiroute-LP` is NP-hard and we instead worked with a k-approximate feasible solution. However, in Lemma 3.2.2 we show that `PC-VC-Multiroute-LP` is solvable in polynomial time. Moreover, Corollary 3.2.3 implies that given a ρ-approximation to `VC-Multiroute-LP`, we can design an $O(\rho)$-approximation for `PC-VC-Multiroute-LP`. Following Chuzhoy's and Khanna's method, we decompose the given instance of node-weighted VC-SNDP into $O(k^3 \log n)$ instances of ELC-SNDP such that the union of their feasible solution is a feasible solution to the VC-SNDP instance. Since the integrality gaps of `ELC-Multiroute-LP` is $O(k \log n)$ (see Theorem 2.3.9) and a feasible solution to the `VC-Multiroute-LP` is a feasible solution to the `ELC-Multiroute-LP` instances arise from Chuzhoy's and Khanna's decomposition (Proposition 10), the integrality gap of `VC-Multiroute-LP` is $O(k^4 \log^2 n)$. Moreover since `VC-Multiroute-LP` is solvable in polynomial time, by Lemma 3.2.1, the integrality gap of `PC-VC-Multiroute-LP` is $O(k^4 \log^2 n)$ too (see Corollary 3.2.3).

3.2.1 LP Relaxations for node-weighted PC-VC-SNDP

Let s and t be two terminals and let ℓ 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 **vertex-disjoint**; we refer to such a tuple $\overline{p}$ as an ℓ-route tuple connecting s to t. A vertex v intersects $\overline{p}$ if there exists a path in $\overline{p}$ that contains v; we use $v \in \overline{p}$ to denote the fact that v intersects $\overline{p}$.

Consider an instance $\langle G, r, \pi \rangle$ of node-weighted PC-VC-SNDP. For each pair st, 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 connectivity 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-VC-Multiroute-LP

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

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

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

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

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

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

VC-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 9. *PC-VC-Multiroute-LP is a valid relaxation for node-weighted PC-VC-SNDP.*

Proof: Consider a feasible solution H of PC-VC-SNDP. Let I be the set of all pairs st such that there are less than $r(st)$ vertex-disjoint paths in H from s to t; differently said, I is the set of all pairs whose requirement is not satisfied by H. Note that the cost of H is $\sum_{v \in V(H)} w(v) + \sum_{st \in I} \pi(st)$.

We construct a solution (x, f, z) to PC-VC-Multiroute-LP as follows. For each vertex v, we set $x(v) = 1$ if v is in H and we set $x(v) = 0$ otherwise. For each pair st, we set $z(st) = 1$ if st is in I and we set $z(st) = 0$ otherwise. For each pair st *not* in I, we select a tuple $\overline{p} \in \mathcal{P}_{st}^{r(st)}$ such that all the vertices of $\overline{p}$ are in H, and we set $f(\overline{p}) = 1$. For each tuple $\overline{p}$ such that $f(\overline{p})$ is undefined, we set $f(\overline{p}) = 0$.

The solution (x, f, z) clearly satisfies the first set of constraints and the last three sets of constraints. Therefore it suffices to verify that the solution satisfies the second set of constraints. Consider a vertex v and a pair st. If v is not in H, there does not exist a tuple $\overline{p}$ such that $v \in \overline{p}$ and $f(\overline{p}) = 1$. Now suppose that v is in H. Note that $x(v) = 1$. Since we selected at most one tuple $\overline{p} \in \mathcal{P}_{st}^{st}$ for the pair st, there is at most one tuple $\overline{p}$ such that $v \in \overline{p}$ and $f(\overline{p}) = 1$.

□

We design an approximation algorithm for node-weighted PC-VC-SNDP as follows. Following the scaling technique of Bienstock et al., given a feasible solution (x, f, z) to PC-VC-Multiroute-LP it is easy to obtain another feasible solution (x', f', z') such that z' is integral and the cost of (x', f', z') is at most 2 times the cost of (x, f, z). The proof of the following lemma is similar to the proof of Lemma 2.3.1 and we skip it.

Lemma 3.2.1. *Consider an instance $\langle G, w, \pi \rangle$ of the node-weighted PC-VC-SNDP problem where the input graph G belongs to a family $\mathcal{G}$ of graphs. Let (x, f, z) be a feasible fractional solution to PC-VC-Multiroute-LP for this*

instance. There is a feasible integral solution H whose cost is at most

$$2\gamma\left(\sum_v x(v)w(v)\right) + 2\left(\sum_{st} z(st)\pi(st)\right) \leq 2\gamma\left(\sum_v x(v)w(v) + \sum_{st} z(st)\pi(st)\right),$$

where γ is an upper bound on the integrality gap of `VC-Multiroute-LP` *on instances of the node-weighted VC-SNDP in which the input graph G is in $\mathcal{G}$. Moreover, if there is a polynomial-time algorithm that finds an integral solution to* `VC-Multiroute-LP` *of value at most ρ times the value of the optimal solution, then there is a polynomial time algorithm that finds an integral solution to* `PC-VC-Multiroute-LP` *of cost at most 2ρ times the cost of its optimal solution.*

Lemma 3.2.2. *`PC-VC-Multiroute-LP`(G, r, π) is solvable in polynomial time.*

Proof: For each pair of terminals st, we need to check whether there exists $(1 - z(st))r(st)$ vertex-disjoint st-paths in G or not. It is equivalent to compute max st-flow in node-capacitated graph G' where the node capacity of each vertex v is $x(v)$. For each pair st, this can be computed in polynomial time via a max-flow algorithm. Thus `PC-VC-Multiroute-LP` has a separation oracle and we can find an optimal solution to `PC-VC-Multiroute-LP` in polynomial time via ellipsoid method. □

Lemma 3.2.1 together with Lemma 3.2.2 imply that there is an $O(\rho)$-approximation to `PC-VC-Multiroute-LP` if `VC-Multiroute-LP` admits a ρ-approximation.

Corollary 3.2.3. *Suppose that the integrality gap of* `VC-Multiroute-LP` *on instance of node-weighted VC-SNDP in which the input graph belongs to a family $\mathcal{G}$ of graphs is γ. Then the integrality gap of* `PC-VC-Multiroute-LP` *on instances of node-weighted PC-VC-SNDP in which the input graph belongs to $\mathcal{G}$ is $O(\gamma)$. Moreover, there is a polynomial time algorithm that finds an integral $O(\rho)$-approximation to* `PC-VC-Multiroute-LP` *if there exists an algorithm that finds an integral ρ-approximation to* `VC-Multiroute-LP` *in polynomial time.*

Following technical lemma follows form Chuzhoy's and Khanna's reduction [6].

Lemma 3.2.4. *Given an instance $\langle G, r\rangle$ of VC-SNDP, in polynomial time we can construct a family terminals $\mathcal{T} = \{T_1, T_2, \cdots, T_p\}$ of size $O(k^3 \log n)$ such that the union of the feasible solutions to ELC-SNDP instances on each T_i is a feasible solution to the VC-SNDP instance. For each T_i, the ELC-SNDP $\langle G, r_i\rangle$ is defined as follows. Vertices of T_i are considered as reliable and the rest are considered as non-reliable. Moreover, $r_i(st) = r(st)$ if $s, t \in T_i$ and zero otherwise.*

It is straightforward to verify the following proposition.

Proposition 10. *Consider an instance $\langle G, r\rangle$ of VC-SNDP. Let $\langle G, r_i\rangle$ be an instance of ELC-SNDP that arises from on terminal set T_i generated by Chuzhoy's and Khanna's reduction. A feasible solution H to* `VC-Multiroute-LP`(G, r)

is a feasible solution to `ELC-Multiroute-LP`(G, r_i).

Lemma 3.2.4 and Proposition 10 and the fact that the integrality gap of `ELC-Multiroute-LP` is $O(k \log n)$ in general graphs an $O(k)$ in minor-closed families of graphs (see Theorem 2.3.9 and Theorem 2.3.11) imply that the integrality gap of `VC-Multiroute-LP` is $O(k^4 \log^2 n)$.

Theorem 3.2.5. *There is an algorithm that constructs a feasible integral solution of* `VC-Multiroute-LP` *in polynomial time with cost at most* $O(k^4 \log^2 n)$OPT *where* OPT *is the cost of the optimal fractional solution of* `VC-Multiroute-LP`*. This ratio improves to* $O(k^4 \log n)$ *on instances for which the input graph belongs to a minor-closed family* $\mathcal{G}$ *where the constant only depends on the family* $\mathcal{G}$*.*

Corollary 3.2.3 and Theorem 3.2.5 show that there exists an $O(k^4 \log^2 n)$-approximation for node-weighted PC-VC-SNDP.

Theorem 3.2.6. *There is an algorithm that finds an* $O(k^4 \log^2 n)$*-approximate solution of PC-VC-SNDP in polynomial time. This ratio improves to* $O(k^4 \log n)$ *on instances for which the input graph belongs to a minor-closed family* $\mathcal{G}$ *where the constant only depends on the family* $\mathcal{G}$*.*

Chuzhoy and Khanna [6] also showed the following result that improves the approximation ratio for Rooted VC-SNDP in which in addition to the input graph G we are given a root vertex $r \in V(G)$ and $r(st) = 0$ if $s \neq r$.

Lemma 3.2.7. *Given an instance* $\langle G, r \rangle$ *of Rooted VC-SNDP with root* s*, in polynomial time we can construct a family terminals* $\mathcal{T} = \{T_1, T_2, \cdots, T_p\}$ *of size* $O(k^2 \log n)$ *such that the union of the feasible solutions to Rooted ELC-SNDP instances on each* T_i *is a feasible solution to the Rooted VC-SNDP instance. For each* T_i*, the Rooted ELC-SNDP instance* $\langle G, r_i \rangle$ *is defined as follows. Vertices of* T_i *are considered as reliable and the rest are considered as non-reliable. Moreover,* $r_i(st) = r(st)$ *if* $t \in T_i$ *and zero otherwise.*

Remark 3.2.8. *Similar to our approach we described for node-weighted PC-VC-SNDP and by Lemma 3.2.7, There is an* $O(k^3 \log^2 n)$*-approximation for node-weighted Rooted PC-VC-SNDP. The ratio improves to* $O(k^3 \log n)$ *on minor-closed families of graphs. By Corollary 3.2.3, we have an* $O(k^3 \log^2 n)$*-approximation for node-weighted Rooted PC-VC-SNDP which improves to* $O(k^3 \log n)$ *on instances in which the input graph belongs to a minor-closed families of graphs.*

3.3 Integrality gap of `ELC-Aug-LP`

In this section, we prove Theorem 3.1.6 that upper bounds the integrality gap of `ELC-Aug-LP`. 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 section 3.1, we have the following LP relaxation for this problem.

$\texttt{ELC-Aug-LP}(G', h)$

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

$$\text{s.t.} \sum_{v \in \Gamma_{G'}(\hat{S})} x(v) \geq h(\hat{S}) \quad \forall \hat{S} \in \mathcal{B}_{\texttt{ELC}}$$

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

Dual of $\texttt{ELC-Aug-LP}(G', h)$

$$\max \sum_{\hat{S} \in \mathcal{P}} y(\hat{S})h(\hat{S})$$

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

$$y(\hat{S}) \geq 0 \quad \forall \hat{S} \in \mathcal{B}_{\texttt{ELC}}$$

Our proof uses the concept of a (generalized) *spider* that was introduced by Nutov [24] 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 the solution of the LP relaxation. We start with some notation and some definitions that are based on [24,27].

Recall that we are working with a 0-1 biuncrossable function $h : \mathcal{B}_{\texttt{ELC}} \to \{0, 1\}$. We can also view h as a family consisting of all bisets $\hat{S}$ such that $h(\hat{S}) = 1$. Following Nutov, we let $\mathcal{F} = \{\hat{S} \mid h(\hat{S}) = 1\}$ be the bifamily corresponding to h. We refer to each biset in $\mathcal{F}$ as a **violated biset** and we refer to the inclusion-wise minimal bisets of $\mathcal{F}$ as **min-core**. Let $\mathcal{C}$ be the set of all min-core of $\mathcal{F}$. The min-cores in $\mathcal{C}$ are non-overlapping and we can compute the collection $\mathcal{C}$ in polynomial time for the function h that arises in ELC-SNDP [19]. Additionally, by Lemma 3.3.1 if $\hat{S}$ is a violated biset and $\hat{C}$ is a min-core, $\hat{C}$ and $\hat{S}$ are non-overlapping.

Lemma 3.3.1. *Consider a biuncrossable family $\mathcal{F}$. Let $\hat{S}$ be a biset in $\mathcal{F}$ and $\hat{C}$ be a min-core of $\mathcal{F}$. Then, $\hat{C}$ and $\hat{S}$ are non-overlapping. In particular, the inner part of the min-cores of $\mathcal{F}$ are pairwise disjoint.*

Proof: Suppose that $\hat{S}$ and $\hat{C}$ are overlapping. Since $\mathcal{F}$ is a biuncrossable family either $\hat{C} \cap \hat{S}$ or $\hat{C} - \hat{S}$ is in $\mathcal{F}$ which contradicts the minimality of $\hat{C}$. Thus $\hat{S}$ and $\hat{C}$ are non-overlapping. In particular, since no min-core contains a biset of $\mathcal{F}$, the inner part of the min-cores of $\mathcal{F}$ are pairwise disjoint. □

A biset $\hat{S} = (S, S') \in \mathcal{F}$ is a **core** of $\mathcal{F}$ iff $\hat{S}$ contains exactly one min-core of $\mathcal{F}$; we refer to a core $\hat{S}$ that contains the min-core $\hat{C} \in \mathcal{C}$ as a $\hat{C}$-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 bisets $\hat{S} \in \mathcal{F}$ such that $\hat{S}$ is an $\hat{A}$-core for some $\hat{A} \in \mathcal{A}$ and $u \notin S'$. We refer to the bifamily $\mathcal{S}(\mathcal{A}, u)$ as a **spider bifamily**. 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 bisets iff for each biset $\hat{S} \in \mathcal{F}'$, there is at least one edge of F leaving $\hat{S}$; more precisely, we have $|\delta_F(\hat{S})| \geq 1$ for each biset $\hat{S} \in \mathcal{F}'$. If $\mathcal{F}'$ is a spider bifamily, we refer to F as a **spider cover**. Nutov [24] 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 [22] for the node-weighted Steiner tree problem; we refer the reader to [24] for more details. We remark that there are subtleties when thinking about spiders for biuncrossable 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 bifamily $\mathcal{S}(\mathcal{A}, u)$ and a cover F of $\mathcal{S}(\mathcal{A}, u)$. Let $\mathcal{F}' = \{\hat{S} \mid \hat{S} \in \mathcal{F}, \delta_F(\hat{S}) = \emptyset\}$ be the subfamily of $\mathcal{F}$ that is not covered by F; the residual family $\mathcal{F}'$ is uncrossable as well (see Proposition 3.1.2). 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 $\texttt{ELC-Aug-LP}(G', h)$. More precisely, we show the following theorem.

Theorem 3.3.2. *There is a spider bifamily $\mathcal{S}(\mathcal{A}, u)$ of $\mathcal{F}$ and a cover F of $\mathcal{S}(\mathcal{A}, u)$ with the following properties. Let $\mathcal{F}' = \{\hat{S} \mid \hat{S} \in \mathcal{F}, \delta_F(\hat{S}) = \emptyset\}$ be the subfamily of $\mathcal{F}$ that is not covered by F, and let $\mathcal{C}'$ be the collection of all minimal bisets 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 $\texttt{ELC-Aug-LP}(G', h)$. 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 3.3.2, we can find a cover of h using a greedy algorithm. If the collection $\mathcal{C}$ of all minimal violated bisets is empty, we return an empty cover. Otherwise, let $\mathcal{S}(A, u)$ and F be the spider bifamily and spider cover guaranteed by Theorem 3.3.2. Let H' be the selected subgraph and h' be the 0-1 function corresponding to the residual family $\mathcal{F}'$ ($h'(\hat{S}) = 1$ if $\hat{S} \in \mathcal{F}'$ and zero otherwise), 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 $\texttt{ELC-Aug-LP}(G'', h')$ is at most the weight of the optimal fractional solution to $\texttt{ELC-Aug-LP}(G', h)$. 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 $\texttt{ELC-Aug-LP}(G', h)$.

Therefore, in order to complete the proof of Theorem 3.1.6, it suffices to prove Theorem 3.3.2. In the following, we give the algorithm for constructing the spider bifamily $\mathcal{S}(\mathcal{A}, u)$.

Primal-dual algorithm for constructing the spider bifamily. Consider the dual of the $\texttt{ELC-Aug-LP}(G', h)$. 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 $\texttt{ELC-Aug-LP}(G', h)$; 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 biset $\hat{S}$ is violated with respect to a set Z of nodes iff $h(\hat{S}) = 1$ and $\delta_{G'[Z]}(\hat{S})$

is empty. Recall that $\mathcal{C}$ is the collection of all minimal violated bisets of h; note that $\mathcal{C}$ is also the collection of all minimal bisets that are violated with respect to X_0. Let $\mathcal{C}_{i-1}$ be the collection of minimal violated bisets with respect to X_{i-1}. For each biset $\hat{C} \in \mathcal{C}_{i-1}$, we have $C' \subseteq X_{i-1}$; furthermore, $C' - C \subseteq X_0$ (see Lemma 3.1.8 and 3.3.3). Since the components of $\mathcal{C}_{i-1}$ are non-overlapping and two components $\hat{C} \in \mathcal{C}$ and $\hat{C}' \in \mathcal{C}_{i-1}$ do not overlap, 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 biset of $\mathcal{C}_{i-1}$. Otherwise, we increase the dual variables $\{y(\hat{C})\}_{\hat{C} \in \mathcal{C}_{i-1}}$ 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 $X_i = X_{i-1} \cup \{v\}$).

The primal-dual algorithm described above is not well-defined for an arbitrary biuncrossable function h. However, by Lemma 3.1.8 and the following lemma, primal-dual works on biuncrossable function arise in ELC-SNDP instances.

Lemma 3.3.3. *Let $\hat{C} = (C, C')$ be a biset in $\mathcal{C}_{i-1}$. Then C' is a subset of X_{i-1}.*

Proof: By Lemma 3.1.8, $C' - C$ is contained in X_0. Therefore it suffices to show that C is a subset of X_{i-1}. Let $S = C \cap X_{i-1}$. Suppose for contradiction that S is a proper subset of C. Note that S contains all of the terminals of C, since terminals have zero weight and let $\hat{Y} = (S, S \cup (C' - C))$. Since $C - S$ contains no terminal, we have $r(\hat{C}) = r(\hat{Y})$. Therefore $r(\hat{Y}) \geq \ell$. Note that $H_{\ell-1}$ does not have any edges incident to $C - S$, since the vertices in $C - S$ have non-zero weight. Therefore $\delta_{H_{\ell-1}}(\hat{Y}) = \delta_{H_{\ell-1}}(\hat{C})$. It follows that $\hat{Y}$ is violated, which contradicts the minimality of $\hat{C}$. □

Following lemma shows that primal-dual is well-defined for `ELC-Aug-LP` on biuncrossable functions arise in ELC-SNDP instances.

Lemma 3.3.4. *Consider iteration i in the phase ℓ of the augmentation problem of ELC-SNDP. The dual solution $\mathbf{y}$ constructed by the primal-dual algorithm satisfies the primal complementary slackness conditions. More precisely, for each vertex $v \in X$, we have $\sum_{\hat{S}: v \in \Gamma_{G'}(\hat{S})} y(\hat{S}) = w(v)$.*

Proof: We can prove the lemma by induction on the number of iterations of the primal-dual algorithm. Initially, $\mathbf{y}$ is zero and X_0 consists of all zero-weight vertices. Thus the complementary slackness conditions are satisfied at the beginning of the algorithm. Now consider iteration $i > 0$. By Lemma 3.3.3, none of the vertices in X_{i-1} are adjacent in G' to the minimal violated components of $\mathcal{C}_{i-1}$. Thus, at the end of iteration i, we have $\sum_{\hat{S}: v \in \Gamma_{G'}(\hat{S})} y(\hat{S}) = w(v)$ for each vertex $v \in X_{i-1}$. Additionally, the vertices in $X_i - X_{i-1}$ became tight in iteration i. Thus, at the end of iteration i, we have $\sum_{\hat{S}: v \in \Gamma_{G'}(\hat{S})} y(\hat{S}) = w(v)$ for each vertex $v \in X_i - X_{i-1}$ as well. □

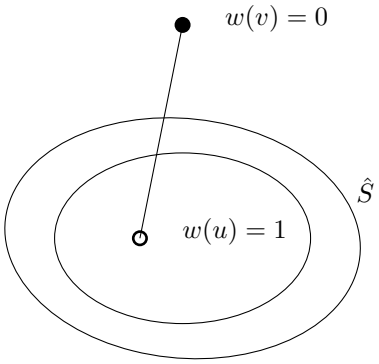

Figure 3.5: *This figure shows the case that integrality gap of* `ELC-Aug-LP` *is unbounded. Note that the inner part of the biset contains a non-zero weight vertex. Consider the function h that $h(\hat{S}) = 1$ and zero for other bisets.*

Remark 3.3.5. *The integrality gap of* `ELC-Aug-LP`(G'_ℓ, h_ℓ) *is unbounded when h_ℓ is an arbitrary* 0-1 *uncrossable function (Figure 3.5). However, the functions h_ℓ that arise from instances of the node-weighted ELC-SNDP in the augmentation framework have additional properties that guarantee a bounded integrality gap.*

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 bisets that were minimal violated bisets 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 $\mathcal{A} = \{\hat{C} \in \mathcal{C} \mid \text{there is } \hat{D} \in \mathcal{C}_{i^*-1} \text{ such that } \hat{C} \preceq \hat{D} \text{ and } u \in \Gamma_{G'}(\hat{D})\}$. The family $\mathcal{S}(\mathcal{A}, u)$ is the desired spider bifamily. In order to describe the reverse delete and analyze the cost of the selected nodes we need to establish some properties of bifamily $\hat{\mathcal{C}}$ formally. Following lemma implies that the collection $\hat{\mathcal{C}}$ is a non-overlapping union of non-empty chains $\{\mathcal{L}_{\hat{C}}\}_{\hat{C} \in \mathcal{C}}$, where each chain $\mathcal{L}_{\hat{C}}$ consists of $\hat{C}$-cores.

Lemma 3.3.6. *The collection $\hat{\mathcal{C}}$ has the following properties:*

(a) $\hat{\mathcal{C}}$ is a laminar bifamily.

(b) For each min-core $\hat{C} \in \mathcal{C}$ and each iteration i of the primal-dual algorithm, there is a biset $\hat{D} \in \mathcal{C}_{i-1}$ such that $\hat{C} \preceq \hat{D}$.

(c) Each biset in $\hat{\mathcal{C}}$ is a core of h.

Consider the collection $\mathcal{L}_{\hat{C}}$ consisting of all bisets of $\hat{\mathcal{C}}$ that contain the min-core $\hat{C} \in \mathcal{C}$. By the theorem above, $\mathcal{L}_{\hat{C}}$ is a laminar bifamily. Since the bisets of $\mathcal{L}_{\hat{C}}$ are $\hat{C}$-cores, they form a chain. The lemma above follows from Proposition 12, Proposition 13, and Proposition 14.

Proposition 11. *Consider an iteration i of the primal-dual algorithm. The bisets of $\mathcal{C}_{i-1}$ are non-overlapping. Additionally, each biset of $\mathcal{C}_{i-1}$ contains a min-core of $\mathcal{C}$.*

Proof: The fact that the components of $\mathcal{C}_{i-1}$ are non-overlapping follows from Lemma 3.3.1. Moreover, each violated biset contains a min-core of $\mathcal{C}$ and therefore each biset of $\mathcal{C}_i$ contains a min-core of $\mathcal{C}$. □

Proposition 12. *The collection $\hat{\mathcal{C}}$ is a laminar bifamily.*

Proof: Consider two bisets $\hat{D}_i \in \mathcal{C}_i$ and $\hat{D}_j \in \mathcal{C}_j$. Without loss of generality, $i \leq j$. The set $\hat{D}_i$ is a minimal violated biset with respect to X_i and $\hat{D}_j$ is a violated biset with respect to $X_j \supseteq X_i$. Thus it follows from Lemma 3.3.1 that $\hat{D}_i$ and $\hat{D}_j$ do not overlap. □

Proposition 13. *Consider an iteration i of the primal-dual algorithm. For each min-core $\hat{C} \in \mathcal{C}$, there exists a biset $\hat{D} \in \mathcal{C}_{i-1}$ such that $\hat{C} \preceq \hat{D}$.*

Proof: Since $\mathcal{C}_0 = \mathcal{C}$, the claim holds for the first iteration. Therefore we may assume that $i \geq 2$. Suppose for contradiction that there is a min-core $\hat{C} \in \mathcal{C}$ such that there does not exist a biset $\hat{D} \in \mathcal{C}_{i-1}$ that contains $\hat{C}$. We claim that $|\mathcal{C}_{i-1}| < |\mathcal{C}|$. It follows from Proposition 11 that each biset of $\mathcal{C}_{i-1}$ contains at least one min-core of $\mathcal{C}$. Since the bisets of $\mathcal{C}_{i-1}$ are non-overlapping and $\hat{C}$ is not contained in any of the bisets of $\mathcal{C}_{i-1}$, it follows that $|\mathcal{C}_{i-1}| < |\mathcal{C}|$. But then the algorithm should have terminated at the end of iteration $i-1$ instead of i^*. □

Proposition 14. *Each biset $\hat{D} \in \hat{\mathcal{C}}$ is a core of h.*

Proof: Suppose for contradiction that there is a biset $\hat{D} \in \mathcal{C}_{i-1}$ that is not a core; thus $\hat{D}$ contains at least two min-cores of $\mathcal{C}$. By Proposition 11, the bisets of $\mathcal{C}_{i-1}$ are non-overlapping and each biset of $\mathcal{C}_{i-1}$ contains a min-core of $\mathcal{C}$. Therefore $|\mathcal{C}_{i-1}| \leq |\mathcal{C}|$. Since $\hat{D}$ contains at least two min-cores of $\mathcal{C}$, we have $|\mathcal{C}_{i-1}| < |\mathcal{C}|$. But then the algorithm should have terminated at the end of iteration $i-1$ instead of i^*. □

Each vertex $v \in X - (X_0 \cup \{u\})$ is added in some iteration of the primal-dual algorithm. Since none of the chains merged until u was added, we have the property that v is adjacent to only bisets in a single chain in $\hat{\mathcal{C}}$. Moreover, by definition u is adjacent to all of the chains whose min-cores are in $\mathcal{A}$. We let $Y_{\hat{C}}$ be the set of all the nodes in $X - (X_0 \cup \{u\})$ that are adjacent to some $\hat{C}$-core in $\hat{\mathcal{C}}$. The following lemma shows that $\{Y_{\hat{C}}\}_{\hat{C} \in \mathcal{C}}$ is a partition of $X - (X_0 \cup \{u\})$. Let $Y_{\hat{C},v}$ be the set of all vertices of $Y_{\hat{C}}$ that were added to X before v.

Lemma 3.3.7. *For each vertex $v \in X - (X_0 \cup \{u\})$, the collection $\{\hat{D} \mid \hat{D} \in \hat{\mathcal{C}}, v \in \Gamma_{G'}(\hat{D})\}$ is a subset of a single chain $\mathcal{L}_{\hat{C}}$ of $\hat{\mathcal{C}}$. Additionally, if $\hat{D}$ is a biset in $\hat{\mathcal{C}}$ such that $u \in \Gamma_{G'}(\hat{D})$, then $\hat{D}$ is an $\hat{A}$-core for some foot $\hat{A} \in \mathcal{A}$.*

The lemma follows from Proposition 17 and Proposition 18.

Proposition 15. *Consider a $\hat{C}$-core $\hat{D}$ such that $v \in \Gamma_{G'}(\hat{D})$. Suppose that v is added to X in iteration i. Then $\hat{D} \in \mathcal{C}_j$ for some $j \leq i-1$.*

Proof: Suppose for contradiction that $\hat{D} \in \mathcal{C}_j$ for some $j \geq i$ and thus $\hat{D}$ is violated with respect to X_j. By Lemma 3.3.3, we have $D \subseteq X_j$. Additionally, $v \in X_i \subseteq X_j$. Therefore the edge of G' connecting v to D is in $G'[X_j]$, which contradicts the fact that $\hat{D}$ is violated with respect to X_j. Therefore $j \leq i-1$. □

Proposition 16. *Consider a min-core of $\hat{C} \in \mathcal{C}$ and let v be a vertex in $Y_{\hat{C}}$ that is added to X in iteration j. There exists a $\hat{C}$-core $\hat{D} \in \mathcal{C}_j$ such that $v \in D$.*

Proof: Since $v \notin X_0$, v became tight in some iteration of the primal-dual algorithm. Let i denote the iteration in which v became tight; note that $i \leq j < i^*$ (we have $j < i^*$ because $v \neq u$). Since v became tight in iteration i and $v \in Y_{\hat{C}}$, Proposition 15 implies that there is a $\hat{C}$-core $\hat{S} \in \mathcal{C}_{j-1}$ such that $v \in \Gamma_{G'[X_{i-1} \cup \{v\}]}(\hat{S})$. Since $j < i^*$, there is a $\hat{C}$-core $\hat{T} \in \mathcal{C}_j$ (by Proposition 13). Additionally, $\hat{S} \preceq \hat{T}$, since $\hat{\mathcal{C}}$ is a laminar pair family (by Proposition 12). Since $\hat{T}$ is violated with respect to X_j and $v \in X_j$, we have $v \in T'$. Since $v \notin X_0$, Lemma 3.1.8 implies that $v \in T$. Therefore $\hat{D} = \hat{T}$ is the desired biset. □

Proposition 17. *Let v be a vertex in $X - (X_0 \cup \{u\})$ that is added to X in iteration i. The collection $\{\hat{D} \mid \hat{D} \in \hat{\mathcal{C}}, v \in \Gamma_{G'}(\hat{D})\}$ consists of cores containing the same min-core of $\mathcal{C}$.*

Proof: The claim is clearly true if there is at most one biset $\hat{D} \in \hat{\mathcal{C}}$ such that $v \in \Gamma_{G'}(\hat{D})$. Now consider two bisets $\hat{D}_1$ and $\hat{D}_2$ in $\hat{\mathcal{C}}$ such that $v \in \Gamma_{G'}(\hat{D}_1)$ and $v \in \Gamma_{G'}(\hat{D}_2)$. By Proposition 12, $\hat{D}_1$ and $\hat{D}_2$ are non-overlapping. Proposition 15 implies that $\hat{D}_1 \in \mathcal{C}_j$ for some $j \leq i-1$. Similarly, $\hat{D}_2 \in \mathcal{C}_\ell$ for some index $\ell \leq i-1$.

Suppose for contradiction that neither $\hat{D}_1 \preceq \hat{D}_2$ nor $\hat{D}_2 \preceq \hat{D}_1$; $\hat{D}$ is a $\hat{C}_1$-core and $\hat{D}_2$ is a $\hat{C}_2$ core where $\hat{C}_1$ and $\hat{C}_2$ are different min-cores of $\mathcal{C}$. Thus it follows from Proposition 16 that there are a $\hat{C}_1$-core $\hat{W}_1 \in \mathcal{C}_i$ and a $\hat{C}_2$-core $\hat{W}_2 \in \mathcal{C}_i$ such that $v \in W_1 \cap W_2$. Since $\hat{W}_1$ and $\hat{W}_2$ are cores of different min-cores, W_1 and W_2 are overlapping; however, by Proposition 12 the collection $\hat{\mathcal{C}}$ is a laminar bifamily and it is a contradiction. □

Proposition 18. *Consider a biset $\hat{D} \in \hat{\mathcal{C}}$ such that $u \in \Gamma_{G'}(\hat{D})$. Then $\hat{D}$ is an $\hat{A}$-core for some foot $\hat{A} \in \mathcal{A}$.*

Proof: Suppose for contradiction that $\hat{D}$ is a $\hat{B}$-core where $\hat{B} \notin \mathcal{A}$. Let $\hat{S} \in \hat{\mathcal{C}}$ be a maximal $\hat{B}$-core. Note that $\hat{S} \notin \mathcal{C}_{i^*-1}$, since otherwise $\hat{B}$ would be in $\mathcal{A}$. It follows that none of the bisets in $\mathcal{C}_{i^*-1}$ contain $\hat{B}$ But then $|\mathcal{C}_{i^*-1}| < |\mathcal{C}|$ (by Proposition 11). Therefore the algorithm should have terminated in iteration $i \leq i^* - 1$. □

The following lemma says that, during iteration i, the minimal violated bisets contain only vertices of X_0 and vertices of their chain that were added so far.

Lemma 3.3.8. *Let $\hat{D}$ be the unique $\hat{C}$-core in $\mathcal{C}_{i-1}$. Then $D \subseteq (Y_{\hat{C}} \cap X_{i-1}) \cup X_0$.*

Proof: By Lemma 3.3.3, we have $D \subseteq X_{i-1}$. Suppose for contradiction that there exists a vertex $v \in D$ such that $v \notin Y_{\hat{C}} \cup X_0$. Then $v \in Y_{\hat{C}'}$ for some min-core $\hat{C}' \in \mathcal{C}$ such that $\hat{C} \neq \hat{C}'$. Since $v \notin X_0$, it was added to X by a $\hat{C}'$-core. By Proposition 16, there is a $\hat{C}'$-core $\hat{D}' \in \hat{\mathcal{C}}$ that contains v in its inner part. But then $\hat{D}$ and $\hat{D}'$ are overlapping, which contradicts the fact that $\hat{\mathcal{C}}$ is a laminar bifamily (see Proposition 12). □

We consider each foot $\hat{A} \in \mathcal{A}$ separately. An important observation is that $G'[Y_{\hat{A}} \cup X_0 \cup \{u\}]$ covers $\mathcal{S}(\{\hat{A}\}, u)$. Following lemma proves this observation.

Lemma 3.3.9. *Consider a min-core $\hat{A} \in \mathcal{A}$, one of the feet. The graph $G'[Y_{\hat{A}} \cup X_0 \cup \{u\}]$ covers the spider bifamily $\mathcal{S}(\{\hat{A}\}, u)$.*

Proof: Let $\hat{D}_u$ be the maximal biset in $\mathcal{L}_{\hat{A}}$. Since $\hat{A} \in \mathcal{A}$, u is adjacent to $\hat{D}_u$, that is, $u \in \Gamma_{G'[X]}(\hat{D}_u)$. By Lemma 3.3.8, we have $D_u \subseteq (Y_{\hat{A}} \cap X_{i^*-1}) \cup X_0 = Y_{\hat{A}} \cup X_0$. Let $\hat{S}$ be a biset in $\mathcal{S}(\{\hat{A}\}, u)$ that is not covered by $Y_{\hat{A}} \cup X_0 \cup \{u\}$. Since $\hat{D}_u$ is the minimal violated pair with respect to $X - u$, $\hat{S}$ is not contained in $\hat{D}_u$. If $\hat{D}_u$ is contained in $\hat{S}$, we have $u \in \Gamma_{G'[Y_{\hat{A}} \cup X_0 \cup \{u\}]}(\hat{S})$. Therefore we may assume that $\hat{S}$ and $\hat{D}_u$ overlap. Since h is biuncrossable and $\hat{S}$ and $\hat{D}_u$ are both $\hat{A}$-cores, we have $h(\hat{S} \cap \hat{D}_u) = h(\hat{S} \cup \hat{D}_u) = 1$. The biset $\hat{D}_u \cap \hat{S}$ is not violated with respect to $X - u$ (for otherwise $\hat{D}_u$ would not be a minimal violated biset at that stage) and therefore there is an edge $e \in \delta_{G'[X-u]}(\hat{D}_u \cap \hat{S})$. Since $e \notin \delta_{G'[X-u]}(\hat{D}_u)$, e has one endpoint in $D_u \cap S$ and the other in $D'_u - S'$. By Lemma 3.3.3, $D'_u \subseteq Y_{\hat{A}} \cup X_0$ and this implies that both endpoints of e are in $Y_{\hat{A}} \cup X_0$. Since both endpoints of e are in $Y_{\hat{A}} \cup X_0$ and the edge e is leaving $\hat{S}$, $\hat{S}$ is covered by $G'[Y_{\hat{A}} \cup X_0]$. Therefore $G'[Y_{\hat{A}} \cup X_0 \cup \{u\}]$ covers $\mathcal{S}(\{\hat{A}\}, u)$. □

Finally, we perform the following *reverse-delete* step on the set X of nodes in order to identify a subset of nodes that cover $\mathcal{S}(\mathcal{A}, u)$. For each foot $\hat{A}$, we select a set $Z_{\hat{A}} \subseteq Y_{\hat{A}}$ such that $G'[Z_{\hat{A}} \cup X_0 \cup \{u\}]$ covers $\mathcal{S}(\{\hat{A}\}, u)$ as follows. We start with $Z_{\hat{A}} = Y_{\hat{A}}$. We consider the nodes of $Z_{\hat{A}}$ in the *reverse* of the order in which they were added to X. Let v be the current node. If the graph $G'[(Z_{\hat{A}} \cup X_0 \cup \{u\}) - \{v\}]$ covers the spider bifamily $\mathcal{S}(\{\hat{A}\}, u)$, we remove v from $Z_{\hat{A}}$. We set $Z = \cup_{\hat{A} \in \mathcal{A}} Z_{\hat{A}}$ and we output the family $\mathcal{S}(\mathcal{A}, u)$ and the cover $G'[Z \cup X_0 \cup \{u\}]$.

In the following we prove that the spider family $\mathcal{S}(\mathcal{A}, u)$ and the cover $G'[Z \cup X_0 \cup \{u\}]$ have the properties required by Theorem 3.3.2. As before we let $\mathcal{C} = \mathcal{C}_0$ be the set of all min-cores of h. Let $H' = G[V(H) \cup Z \cup X_0 \cup \{u\}]$. Let $\mathcal{C}'$ be the collection of all min-cores with respect to H'.

In the following, we relate the cost of the nodes in Z to the cost of $\texttt{ELC-Aug-LP}(G', h)$. Let Δ_i be the amount we increased the dual variables in iteration i of the primal-dual algorithm, and let $\Delta = \sum_{i=1}^{i^*} \Delta_i$.

Our first observation is that, since in each iteration the number of minimal violated bisets is $|\mathcal{C}|$ and we grew each biset by the same amount, the value of the dual solution is $\Delta|\mathcal{C}|$. Therefore the optimal cost of $\texttt{ELC-Aug-LP}(G', h)$ is at least $\Delta|\mathcal{C}|$. Our second observation — which is the main ingredient of the analysis — is that we can charge the vertices of Z to the growth of the chains; for each foot $\hat{A} \in \mathcal{A}$, we can charge the vertices in $Z_{\hat{A}}$ to the chain of $\hat{\mathcal{C}}$ whose min-core is $\hat{A}$. More precisely, we show in Lemma 3.3.11 that the total weight of Z is $O(|\mathcal{A}|\Delta)$. Finally, since $Z \cup X_0$ covers the spider family $\mathcal{S}(\mathcal{A}, u)$, after we add Z each new minimal violated biset that contains a foot also contains an additional min-core of $\mathcal{C}$. This observation gives us that the number of minimal violated bisets decreases by $\Omega(|\mathcal{A}|)$.

Proposition 19. *The dual solution y constructed by the primal-dual algorithm is a feasible solution such that*

$$\sum_{\hat{S}} y(\hat{S}) h(\hat{S}) = |\mathcal{C}|\Delta.$$

Proof: Note that the dual variable of each component in $\mathcal{C}_i$ increases by Δ_i in iteration i. Therefore the total increase of the dual variables in iteration i is $\Delta_i|\mathcal{C}_{i-1}|$. For each iteration $i \leq i^*$, we have $|\mathcal{C}_{i-1}| = |\mathcal{C}|$, and the proposition follows. □

By weak duality, the weight of the optimal fractional solution for $\texttt{ELC-Aug-LP}(G', h)$ is at least the value of the dual solution y constructed by the primal-dual algorithm. Therefore we have the following corollary.

Corollary 3.3.10. *The weight of the optimal fractional solution to* $\texttt{ELC-Aug-LP}(G', h)$ *is at least* $|\mathcal{C}|\Delta$.

The following lemma relates the weight of $Z \cup \{u\}$ to the total increase Δ, and it is the main component of our analysis. The main idea behind the proof of the lemma is that, for each foot A, each component of the chain $\mathcal{L}_{\hat{A}}$ has at most one neighbor in Z (see Lemma 3.3.12).

Lemma 3.3.11. *The total weight of the nodes in* Z *is at most* $|\mathcal{A}|\Delta$. *The weight of* u *is at most* $|\mathcal{A}|\Delta$.

By Lemma 3.3.7, if v is a vertex of $Z_{\hat{A}}$, all of the components $\hat{C} \in \hat{\mathcal{C}}$ that have v as a neighbor are in $\mathcal{S}(\{\hat{A}\}, u)$. Therefore we can consider each foot $\hat{A} \in \mathcal{A}$ separately. As shown in Lemma 3.3.12, each biset in the chain of $\hat{\mathcal{C}}$ containing the foot $\hat{A}$ has only a constant number of neighbors in $Z_{\hat{A}}$. The lemma allows us to charge the weight of the vertices of $Z_{\hat{A}}$ to the dual growth of the chain containing $\hat{A}$.

Lemma 3.3.12. *Consider a foot* $\hat{A} \in \mathcal{A}$ *and let* $\hat{D}$ *be an* $\hat{A}$*-core in* $\hat{\mathcal{C}}$. *We have* $|\Gamma_{G'}(\hat{D}) \cap Z_{\hat{A}}| \leq 1$.

Before proving Lemma 3.3.12, we show that it implies Lemma 3.3.11 via the standard primal-dual argument.

Proof of Lemma 3.3.11: Let $\mathcal{L}_{\hat{A}}$ be the collection of all bisets of $\hat{\mathcal{C}}$ that are $\hat{A}$-cores. Note that the bisets of $\mathcal{L}_{\hat{A}}$ form a chain, since $\hat{\mathcal{C}}$ is a laminar bifamily (by Proposition 12). Therefore we have

$$\sum_{\hat{C} \in \mathcal{L}_{\hat{A}}} y(\hat{C}) = \Delta.$$

Consider a vertex $v \in Z$. Since $v \notin X_0$, v became tight in some iteration of the primal-dual algorithm and therefore

$$w(v) = \sum_{\hat{S}:\, v \in \Gamma_{G'}(\hat{S})} y(\hat{S}).$$

The node v is in $Z_{\hat{A}}$ for some foot $\hat{A} \in \mathcal{A}$. We claim that, if $v \in \Gamma_{G'}(\hat{S})$ and $y(\hat{S})$ is non-zero, then $\hat{S}$ is in $\mathcal{L}_{\hat{A}}$. Note that the only bisets $\hat{S}$ that have a non-zero dual variable $y(\hat{S})$ are in $\hat{\mathcal{C}}$. By Lemma 3.3.7, $\hat{S}$ is an $\hat{A}$-core, and the claim follows. Therefore

$$w(v) = \sum_{\hat{S}:\, \hat{S} \in \mathcal{L}_{\hat{A}}, v \in \Gamma_{G'}(\hat{S})} y(\hat{S}).$$

Thus

$$w(Z_{\hat{A}}) = \sum_{v \in Z_{\hat{A}}} \sum_{\hat{S}:\ \hat{S} \in \mathcal{L}_{\hat{A}}, v \in \Gamma_{G'}(\hat{S})} y(\hat{S}) \leq \sum_{\hat{S} \in \mathcal{L}_{\hat{A}}} y(\hat{S}) \leq \Delta.$$

The second to last inequality follow from Lemma 3.3.12. It follows that $w(Z) \leq |\mathcal{A}|\Delta$.

Finally, consider the vertex u. Since u is not in X_0, we have

$$w(u) = \sum_{\hat{S}:\ \hat{S} \in \hat{\mathcal{C}}, u \in \Gamma_{G'}(\hat{S})} y(\hat{S}).$$

By Proposition 18,

$$w(u) \leq \sum_{\hat{A} \in \mathcal{A}} \sum_{\hat{S}:\ \hat{S} \in \mathcal{L}_{\hat{A}}, u \in \Gamma_{G'}(\hat{S})} y(\hat{S}) \leq |\mathcal{A}|\Delta.$$

□

We devote the rest of this section to the proof of Lemma 3.3.12. Let $q : 2^V \times 2^V \to \{0, 1\}$ be a biuncrossable function. Let F be a set of edges. A biset $\hat{W}_e$ is a *F-witness biset* for an edge e iff $q(\hat{W}_e) = 1$ and $\delta_F(\hat{W}_e) = \{e\}$ (see Subsection 1.3.3). Similar the result of [27] for sets system, we have the following lemma.

Lemma 3.3.13. *Let F be a feasible cover for q and $M \subseteq F$ be an inclusion-wise minimal cover for q. There is a laminar bifamily $\mathcal{L} = \{\hat{W}_e \mid e \in M\}$ such that $\hat{W}_e$ is an M-witness biset for e.*

We will need the following lemma to prove Lemma 3.3.12.

Lemma 3.3.14. *Let $\hat{A} \in \mathcal{A}$ and let $\hat{D} \in \mathcal{L}_{\hat{A}}$. Let $\mathcal{Q}_{\hat{D}}$ be the family consisting of all bisets $\hat{S} \in \mathcal{S}(\{\hat{A}\}, u)$ such that $\hat{D} \preceq \hat{S}$. For any two bisets $\hat{Q}_1$ and $\hat{Q}_2$ in $\mathcal{Q}_{\hat{D}}$, we have $\hat{Q}_1 \cap \hat{Q}_2 \in \mathcal{Q}_{\hat{D}}$ and $\hat{Q}_1 \cup \hat{Q}_2 \in \mathcal{Q}_{\hat{D}}$. Additionally, the graph $K = G'[D \cup Z_{\hat{A}} \cup X_0 \cup \{u\}]$ covers $\mathcal{Q}_{\hat{D}}$ and, for each vertex $v \in \Gamma_{G'}(\hat{D}) \cap Z_{\hat{A}}$, the graph $K - v$ does not cover $\mathcal{Q}_{\hat{D}}$.*

Proposition 20. *Consider a min-core $\hat{C} \in \mathcal{C}$ and let v be a vertex in $Y_{\hat{C}}$. Suppose that v was added to X in iteration j. There exists a $\hat{C}$-core $\hat{D}_v \in \mathcal{C}_{j-1}$ such that $v \in \Gamma_{G'[Y_{\hat{C},v} \cup X_0 \cup \{v\}]}(\hat{D}_v)$. Additionally, $D_v \subseteq Y_{\hat{C},v} \cup X_0$ and $\hat{D}_v$ is a minimal violated biset with respect to $Y_{\hat{C},v} \cup X_0$.*

Proof: Since $v \notin X_0$, v became tight in some iteration $i \leq j$ of the primal-dual algorithm. Therefore there exists a biset $\hat{D} \in \mathcal{C}_{i-1}$ such that $v \in \Gamma_{G'[X_{i-1} \cup \{v\}]}(\hat{D})$. By Proposition 13, there is a biset $\hat{S} \in \mathcal{C}_{j-1}$ that contains $\hat{C}$. Additionally, $\hat{D} \preceq \hat{S}$, since $\hat{\mathcal{C}}$ is a laminar bifamily (see Proposition 12). Therefore $v \in \Gamma_{G'[X_{j-1} \cup \{v\}]}(\hat{S})$. By Lemma 3.3.8, we have $S \subseteq (Y_{\hat{C}} \cap X_{j-1}) \cup X_0 = Y_{\hat{C},v} \cup X_0$. Therefore $\hat{D}_v = \hat{S}$ is the desired biset. □

Proof of Lemma 3.3.14: Let $\hat{Q}_1$ and $\hat{Q}_2$ be two bisets in $\mathcal{Q}_{\hat{D}}$. Note that $\hat{Q}_1 \cap \hat{Q}_2$ and $\hat{Q}_1 \cup \hat{Q}_2$ both contain $\hat{D}$ and they are in $\mathcal{S}(\{\hat{A}\}, u)$. Therefore $\hat{Q}_1 \cap \hat{Q}_2$ and $\hat{Q}_1 \cup \hat{Q}_2$ are in $\mathcal{Q}_{\hat{D}}$.

Consider a vertex $v \in \Gamma_{G'}(\hat{D}) \cap Z_{\hat{A}}$. Let $\hat{D}_v$ be the set guaranteed by Proposition 20. We claim that $\hat{D} \preceq \hat{D}_v$. Note that one of $\hat{D}$ and $\hat{D}_v$ is contained in the other, since $\hat{\mathcal{C}}$ is bilaminar (see Proposition 12). Suppose for contradiction that $\hat{D}_v$ is contained in $\hat{D}$. Let j denote the iteration in which v was added to X. Then $\hat{D}_v \in \mathcal{C}_{j-1}$. Additionally, by Proposition 16, there exists an $\hat{A}$-core $\hat{W} \in \mathcal{C}_j$ such that $v \in W$. Since $\hat{\mathcal{C}}$ is laminar, $\hat{W}$ contains both $\hat{D}_v$ and $\hat{D}$. Since W contains v and D does not contain v, it follows that $\hat{D} \in \mathcal{C}_i$ for some $i \leq j-1$, which is a contradiction. Therefore $\hat{D} \preceq \hat{D}_v$. Since $G'[Z_{\hat{A}} \cup X_0 \cup \{u\}]$ covers $\mathcal{S}(\{\hat{A}\}, u)$, K covers $\mathcal{Q}_{\hat{D}}$. Now consider a vertex $v \in \Gamma_{G'}(\hat{D}) \cap Z_{\hat{A}}$. Since we could not remove v in the reverse-delete step, there is a biset $\hat{Q}_v \in \mathcal{S}(\{\hat{A}\}, u)$ that is violated with respect to $(Z_{\hat{A}} - \{v\}) \cup Y_{\hat{A},v} \cup X_0 \cup \{u\}$. Since $D \subseteq D_v \subseteq Y_{A,v} \cup X_0$, $K - v$ does not cover $\hat{Q}_v$. Since $\hat{D}_v$ is a minimal violated biset with respect to $Y_{\hat{A},v} \cup X_0$, we have $\hat{D}_v \preceq \hat{Q}_v$ (see Lemma 3.3.1). Since $\hat{D} \preceq \hat{D}_v$, the biset $\hat{Q}_v$ is in $\mathcal{Q}_{\hat{D}}$. Therefore $K - v$ does not cover $\mathcal{Q}_{\hat{D}}$. □

Now we are ready to prove Lemma 3.3.12.

Proof of Lemma 3.3.12: Let $\mathcal{Q}_{\hat{D}}$ and K be as in the statement of Lemma 3.3.14. Let M be an edge-minimal subset of $E(K)$ that covers the bifamily $\mathcal{Q}_{\hat{D}}$. Note that, for each vertex $v \in \Gamma_{G'}(\hat{D}) \cap Z_{\hat{A}}$, there is at least one edge of M that is incident to v, since $K - v$ is not a feasible cover for $\mathcal{Q}_{\hat{D}}$. Let $\{\hat{W}_e \mid e \in M\}$ be the laminar witness bifamily guaranteed by Lemma 3.3.13. Note that the witness family $\{\hat{W}_e \mid e \in M\}$ is a chain. The set M contains at least one edge e_a such that $e_a \in \delta_M(\hat{D})$. Let a be the endpoint of e_a that is not in D'. Note that $a \in Z_{\hat{A}} \cup \{u\}$. Additionally, if $a = u$ then $\hat{D}$ has no neighbors in $Z_{\hat{A}}$ because e_a itself covers $\mathcal{Q}_{\hat{D}}$. Therefore we may assume that $a \neq u$.

Suppose for contradiction that there exists a vertex $b \in \Gamma_{G'}(\hat{D}) \cap Z_{\hat{A}}$ such that $b \neq a$. Since $K - b$ does not cover $\mathcal{Q}_{\hat{D}}$, there is at least one edge $e_b \in M$ that is incident to b; if there are several such edges, we choose e_b to be the edge with the maximal witness biset $\hat{W}_{e_b}$. Note that a is inside W'_{e_b}, since $\hat{D} \preceq \hat{W}_{e_b}$ and $\hat{W}_{e_b}$ is a witness biset for an edge $e_b \neq e_a$. Since $a \notin X_0$, by Lemma 3.1.8, $a \in W_{e_b}$. Additionally, each edge of M with an endpoint in a has both endpoints in W'_{e_b} unless that edge is ab. Since $K - a$ does not cover $\mathcal{Q}_{\hat{D}}$, there is a biset $\hat{Q} \in \mathcal{Q}_{\hat{D}}$ such that $\delta_{K-a}(\hat{Q}) = \emptyset$. We claim that b is in Q' for otherwise the edge of G' connecting D to b will cover $\hat{Q}$ in $K - a$ (since $\hat{D} \preceq \hat{Q}$). Since $b \notin X_0$, by Lemma 3.1.8, $b \in Q$. Additionally, all the edges of $M - \{ab\}$ that are incident to b have both endpoints in Q'. (We use ab to denote the edge ab; note that the edge may not exist in G'.) It follows that all the edges of M that have an endpoint in the set $\{a, b\}$ have both endpoints in $W'_{e_b} \cup Q'$. Note that $\hat{W}_{e_b} \cap \hat{Q}$ and $\hat{W}_{e_b} \cup \hat{Q}$ are both in $\mathcal{Q}_{\hat{D}}$. Therefore there is an edge $e \in M$ such that $e \in \delta_M(\hat{W}_{e_b} \cup \hat{Q})$. Since a is not an endpoint of e, $e \notin \delta_M(\hat{Q})$ and thus we must have $e \in \delta_M(\hat{W}_{e_b})$. But since b is not an endpoint of e, $e \neq e_b$, which contradicts the fact that $\hat{W}_{e_b}$ is a witness biset for e_b.

Therefore there is at most one vertex in $\Gamma_{G'}(\hat{D}) \cap Z_{\hat{A}}$. □

The following proposition follows essentially from [25].

Proposition 21. *We have* $|\mathcal{A}| \geq 1$ *and* $|\mathcal{C}| - |\mathcal{C}'| \geq |\mathcal{A}|/3$.

Proof: Note that, since $u \notin X_0$, there is at least one component $\hat{C} \in \mathcal{C}_{i^*-1}$ such that $u \in \Gamma_{G'}(\hat{C})$. Therefore $|\mathcal{A}| \geq 1$. Suppose that $|\mathcal{A}| = 1$. Since $|\mathcal{C}'| < |\mathcal{C}|$ and $|\mathcal{C}| - |\mathcal{C}'|$ is an integer, we have $|\mathcal{C}| - |\mathcal{C}'| \geq 1 = |\mathcal{A}|$. Therefore we may assume that $|\mathcal{A}| \geq 2$. In the following, we show that $|\mathcal{C}| - |\mathcal{C}'| \geq (|\mathcal{A}| - 1)/2$, which implies the proposition.

The components of $\mathcal{C}'$ are non-overlapping and each biset of $\mathcal{C}'$ contains at least one biset of $\mathcal{C}$. Moreover, since $G'[Z \cup X_0 \cup \{u\}]$ covers the spider bifamily $\mathcal{S}(\mathcal{A}, u)$, if $\hat{C} = (C, C') \in \mathcal{C}'$ contains a foot of $\mathcal{A}$ then either $u \in C'$ or $\hat{C}$ is not a core of a foot (contains at least two bisets of $\mathcal{C}$). Since $u \notin X_0$, $u \in C'$ implies that $u \in C$ and there is at most one biset $\hat{C} \in \mathcal{C}'$ that contains u in its inner part. Thus $|\mathcal{C}| - |\mathcal{C}'| \geq (|\mathcal{A}| - 1)/2$ and for $|\mathcal{A}| \geq 2$, we have $|\mathcal{A}| - 1 \geq 2|\mathcal{A}|/3$. □

Corollary 3.3.15. *The total weight of the nodes in $Z \cup \{u\}$ is $O\big((|\mathcal{C}| - |\mathcal{C}'|)/|\mathcal{C}|\big)$ times the weight of the optimal fractional solution to* `ELC-Aug-LP`(G', h).

Proof: By Proposition 21, we have $|\mathcal{C}| - |\mathcal{C}'| \geq |\mathcal{A}|/3$. Thus it follows from Lemma 3.3.11 that the weight of $Z \cup \{u\}$ is $O(|\mathcal{C}| - |\mathcal{C}'|) \cdot \Delta$. □

Theorem 3.3.2 follows by combining the preceding corollary with Corollary 3.3.10 and Proposition 21.

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. P. Williamson. A note on the prize collecting traveling salesman problem. *Mathematical programming*, 59(1-3):413–420, 1993.

[3] C. Chekuri, A. Ene, and A. Vakilian. Node-weighted network design in planar and minor-closed families of graphs. In *Automata, Languages, and Programming*, pages 206–217. Springer, 2012.

[4] C. Chekuri, A. Ene, and A. Vakilian. Prize-collecting survivable network design in node-weighted graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 98–109. Springer, 2012.

[5] 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. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 79–88, 2007.

[6] J. Chuzhoy and S. Khanna. An $O(k^3 \log n)$-approximation algorithm for vertex-connectivity survivable network design. *Theory of Computing*, 8:401–413, 2012.

[7] E. D. Demaine, M. Hajiaghayi, and P. N. Klein. Node-weighted Steiner tree and group Steiner tree in planar graphs. In *Automata, Languages and Programming*, pages 328–340. 2009.

[8] L. Fleischer, K. Jain, and D. P. Williamson. Iterative rounding 2-approximation algorithms for minimum-cost vertex connectivity problems. *Journal of Computer and System Sciences*, 72(5):838–867, 2006.

[9] M. X. Goemans, A. V. Goldberg, S. Plotkin, D. B. Shmoys, E. Tardos, and D. P. Williamson. Improved approximation algorithms for network design problems. In *Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 223–232, 1994.

[10] S. Guha and S. Khuller. Improved methods for approximating node weighted steiner trees and connected dominating sets. *Inf. Comput.*, 150(1):57–74, 1999.

[11] S. Guha, A. Moss, J. S. Naor, and B. Schieber. Efficient recovery from power outage. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 574–582, 1999.

[12] A. Gupta and J. Könemann. Approximation algorithms for network design: A survey. *Surveys in Operations Research and Management Science*, 16(1):3–20, 2011.

[13] S. Gutner. Elementary approximation algorithms for prize collecting Steiner tree problems. *Information Processing Letters*, 107(1):39–44, 2008.

[14] M. Hajiaghayi and K. Jain. The prize-collecting generalized Steiner tree problem via a new approach of primal-dual schema. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 631–640, 2006.

[15] M. Hajiaghayi, R. Khandekar, G. Kortsarz, and Z. Nutov. Prize-collecting steiner network problems. *ACM Transactions on Algorithms (TALG)*, 9(1):2, 2012.

[16] M. Hajiaghayi and A. A. Nasri. Prize-collecting Steiner networks via iterative rounding. In *LATIN 2010: Theoretical Informatics*, pages 515–526. Springer, 2010.

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

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

[19] K. Jain, I. I. Mandoiu, V. V. Vazirani, and D. P. Williamson. A primal-dual schema based approximation algorithm for the element connectivity problem. *J. Algorithms*, 45(1):1–15, 2002.

[20] D. S. Johnson, M. Minkoff, and S. Phillips. The prize collecting steiner tree problem: theory and practice. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pages 760–769, 2000.

[21] W. Kishimoto. A method for obtaining the maximum multiroute flows in a network. *Networks*, 27(4):279–291, 1996.

[22] P. N. Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted Steiner trees. *J. Algorithms*, 19(1):104–115, 1995.

[23] C. Nagarajan, Y. Sharma, and D. P. Williamson. Approximation algorithms for prize-collecting network design problems with general connectivity requirements. In *Approximation and Online Algorithms*, pages 174–187. 2009.

[24] Z. Nutov. Approximating Steiner networks with node-weights. *SIAM Journal on Computing*, 39(7):3001–3022, 2010.

[25] Z. Nutov. Approximating minimum-cost connectivity problems via uncrossable bifamilies. *ACM Transactions on Algorithms (TALG)*, 9(1):1, 2012.

[26] Y. Sharma, C. Swamy, and D. P. Williamson. Approximation algorithms for prize collecting forest problems with submodular penalty functions. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1275–1284, 2007.

[27] 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.