

Approximation Algorithms for Relative Survivable Network Design Problems

by

JJ Nan

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics & Optimization

Waterloo, Ontario, Canada, 2025

© JJ Nan 2025

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The research in this thesis was completed in collaboration with Chaitanya Swamy and Nikhil Kumar. The thesis is adapted from a co-authored paper which will appear in the proceedings of the European Symposium on Algorithms (ESA) [KNS25].

Abstract

The *Survivable Network Design* (SND) problem is a classical and well-studied graph connectivity problem. Given a set of source-sink pairs and demands between them, SND asks one to compute a subgraph such that the number of paths between each pair meets their demand. SND is primarily interesting in modeling fault-tolerance; we can see the problem as requiring certain nodes to be connected even if some edges "fail". It is well known that a 2-approximation algorithm for SND exists, using the method of iterative rounding.

In 2022, Dinitz et al. introduced a problem that we refer to as *Path-Relative Survivable Network Design* (PRSND), a natural extension of SND that addresses cases where the underlying graph does not have the required connectivity; in this problem, we require that the connectivity of our subgraph is "as good as" it is in the original graph. Perhaps surprisingly, this variation makes PRSND much harder to approximate than standard SND, and outside of certain special cases no constant-factor approximations have been found.

In this thesis we introduce the *Cut-Relative Survivable Network Design* (CRSND) problem, another variant of SND that similarly aims to capture relative fault-tolerance. We show that this problem admits a 2-approximation algorithm, matching the best known approximation factor for SND, via a decomposition technique. We explore some properties of said approximation, as well as hardness and modeling properties of Cut-Relative Network Design problems.

Acknowledgements

I would like to thank my supervisor, Chaitanya Swamy, and our co-author Nikhil Kumar, for introducing me to and guiding me through my first experience in mathematical research. I would also like to thank my readers, Kanstantsin Pashkovich and Joseph Cheriyan.

I want to thank all of the friends I have met during my time in Waterloo working on this thesis.

A huge thanks to The Musical InterDudes and the UW A Cappella Club, who have welcomed me and stuck with me for my entire time in Waterloo, and where I have made some of my best friendships. I will refrain from naming every person from these groups whose friendship I want to acknowledge, for fear of affecting the page count of this thesis.

And of course, a special thank you to my partner Gale.

Table of Contents

Author's Declaration	ii
Statement of Contributions	iii
Abstract	iv
Acknowledgements	v
1 Introduction	1
1.1 Network Design Problems	1
1.1.1 Cut Requirement Functions and f -Network Design	2
1.1.2 Relative Network Design	3
1.2 Our Results	6
1.3 Related Work	7
1.4 Overview	8
2 Preliminaries	9
2.1 Notation	9
2.2 Problem Definitions	9
2.2.1 Properties of Cut-Requirement Functions	11
2.3 Approximation Algorithms	15
2.4 Ellipsoid Method	15
2.5 Iterative Rounding	16
2.5.1 Application to Cut-Relative Network Design	19
2.6 Modelling Power of CRND	21

3	A Decomposition Method To Approximate CRND	24
3.1	Algorithmic Results	32
4	Properties of Decomposition	34
4.1	Decomposition Tree	34
4.2	Key Structural Property of the Canonical Decomposition Tree	35
4.3	Efficient Computation of the Canonical Decomposition	38
4.4	A Self-Contained Description of the Canonical Decomposition	39
5	Hardness Results	42
6	Path-Relative SND	44
6.1	Cut-Based Formulation	44
6.1.1	LP Relaxation	45
6.2	Connected f -ND	45
6.3	Further Steps	47
6.3.1	Decomposition	47
6.3.2	Augmentation	47
7	Conclusion	49
	References	50

Chapter 1

Introduction

1.1 Network Design Problems

In the most general terms, a network design problem takes as input a graph $G = (V, E)$ with edge costs $c : E \rightarrow \mathbb{R}_{\geq 0}$ and some form of *connectivity requirement*. The goal is to find a minimum-cost subgraph H that satisfies the connectivity requirement. Throughout this thesis, we will overload notation to use H as both the set of edges in a solution and the underlying subgraph induced by that edge set, as well as to treat c as a vector $c \in \mathbb{R}_{\geq 0}^E$ where for $e \in E$, $c_e := c(e)$.

The specific connectivity requirement given defines the problem and its complexity. Some simple examples of requirements include the following.

- Requiring a path in H between any two vertices – this gives the *minimum spanning tree* problem.
- Requiring k edge-disjoint paths in H between any two vertices – this gives the *k -edge connected spanning subgraph* (k -ECSS) problem.
- Requiring a path in H between any two vertices in some set $S \subseteq V$ - this gives the *Steiner Tree* problem.

This thesis focuses on the following general and classical network design problem. For a pair of vertices $s, t \in V$, we call a vertex set $S \subseteq V$ an " s - t cut" if $|S \cap \{s, t\}| = 1$; that is, if S contains s but not t , or vice versa.

Definition. The *Survivable Network Design* (SND) problem (sometimes called the Generalized Steiner Network problem) takes as input a graph $G = (V, E)$, costs $c : E \rightarrow \mathbb{R}_{\geq 0}$, and a set of demands $\{(s_i, t_i, k_i)\}_{i \in [\ell]}$, where $s_i, t_i \in V$ and $k_i \in \mathbb{Z}_{\geq 0}$.

The goal is to find a minimum-cost subgraph H where for all $i \in [\ell]$, there are at least k_i edge-disjoint paths connecting s_i and t_i .

Equivalently, for all $i \in [\ell]$ any s_i - t_i cut $S \subseteq V$ has at least k_i edges on its boundary in H (i.e. $|\delta_H(S)| \geq k_i$).

It is straightforward to see via Menger's Theorem that these "Path" and "Cut" statements (respectively) of the problem are equivalent.

SND, and indeed many connectivity-based problems similar to it, is motivated by the idea of *fault-tolerance*. The high-level idea is that if some specific number of edges "fail", or are in some way removed from the graph, that we maintain desired connectivity. For instance, in this lens the definition of k -connectivity can be viewed as saying that "if any $k-1$ edges are removed from the graph, then there is still a path between any two vertices". This is the goal attained by k -ECSS. SND allows for *non-uniform* requirements between specific pairs of vertices, but the overall idea is the same: for every demand i , if less than k_i edges are removed from the graph, there is still a path between s_i and t_i . We can also see that SND can model problems with uniform demands, such as in our previous examples, which makes it a powerful model for network design problems.

SND is a classical and extensively studied problem. As an NP-hard problem, it is most likely impossible to efficiently compute an optimal solution to SND. As such, it is of particular interest to find *approximation algorithms* to find solutions of cost within some factor of the optimal. Most pertinently, Jain [Jai01] gave a 2-approximation for SND, introducing the iterative rounding method for network design problems; the method is described in some detail in Section 2.5.

1.1.1 Cut Requirement Functions and f -Network Design

It is well known that Survivable Network Design, and indeed many other network design problems, can be modelled using what is known as a *cut requirement function*: a function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ that assigns to each set $S \subseteq V$ some "cut requirement", or specification of a minimum number of edges from $\delta_G(S)$ that we require in a feasible solution. We can use this to state a general network design problem as follows:

Definition. The *f -Network Design* problem ($\text{ND}(f, G)$) takes as input a graph $G = (V, E)$, costs $c : E \rightarrow \mathbb{R}_{\geq 0}$, and a function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$. The goal is to find a minimum-

cost subgraph H where any set $S \subseteq V$ has at least $f(S)$ edges on its boundary in H (i.e. $|\delta_H(S)| \geq f(S)$).

This bears similarity to the "cut formulation" of SND; we can see that SND is modelled by the function

$$f^{SND}(S) := \max_{i: S \text{ is an } s_i-t_i \text{ cut}} k_i$$

that is, for any S and each i such that S is an s_i-t_i cut, we require k_i edges from the boundary $\delta_G(S)$. Taking the maximum over every such i guarantees all of these requirements are satisfied.

A key requirement of Jain's iterative rounding method is that the cut requirement function satisfies a property known as *weak supermodularity* (definition given in Section 2.2.1). The function f^{SND} satisfies this property. [Jai01]

1.1.2 Relative Network Design

In 2022, Dinitz, Koranteng, and Kortsarz [DKK22] presented a variant of Survivable Network Design, motivated by a significant shortcoming in the classical problem's definition: We require that the input graph G itself has at least k_i edges between s_i and t_i , otherwise the problem is infeasible. We give an example that demonstrates this shortcoming.

Example 1. See the graph in Figure 1, with the single demand $(s, t, 3)$, which we may also call " s - t 3-connectivity"; we want 3 edge-disjoint paths between s and t . Clearly, though, this cannot happen since there is only one edge with t as an endpoint, so there can be at most one edge-disjoint s - t path in any solution. The problem is therefore infeasible.

This is a somewhat disappointing observation. Specifically, one can see that even if s_i and t_i are not k_i -connected in G , there are *some* sets of edges of size less than k_i that we can remove without disconnecting s_i and t_i , even if this does not hold for *all* sets. Going back to Example 1, we see that there are many edge sets of size at most 2 that we can remove without disconnecting s and t in G , such as $\{su, sw\}$, $\{uw, vw\}$, and many others. We see that only the edge wt creates a "problem" for SND feasibility, since its removal alone disconnects s and t . It is desirable, then, to have a model that treats these problematic edges differently from others.

Accordingly, Dinitz et al. introduce the concept of *relative* fault-tolerance. At a high level, this encompasses statements of the form "If some edges are removed, our solution H performs *as well as* G in terms of connectivity".

Let $G = (V, E)$, and $F \subseteq E$ be some set of edges of G . Overloading notation, we denote $G - F$ as the subgraph of G found by removing edges in F , or $G - F := (V, E - F)$. Dinitz et al. introduce the following problem.

Definition. In the *Path-Relative Survivable Network Design* (PRSND) problem, we are given an SND instance and wish to find a minimum-cost subgraph H where for all $i \in [\ell]$ and any *fault set* $F \subseteq E$ with $|F| < k_i$, if there is an s_i - t_i path in $G - F$, there is an s_i - t_i path in $H - F$.

In this thesis, we introduce the following problem, which is clearly related to PRSND.

Definition. In the *Cut-Relative Survivable Network Design* (CRSND) problem, we are given an SND instance and wish to find a minimum-cost subgraph H where for all $i \in [\ell]$, s_i - t_i cut S , and $F \subseteq E$ with $|F| < k_i$,

$$\delta_{G-F}(S) \neq \emptyset \implies \delta_{H-F}(S) \neq \emptyset$$

that is to say, if the boundary of S in G is nonempty after faults, then the boundary of S in H must be as well.

Perhaps surprisingly, these two problems are *not equivalent* generally. Indeed, though it is clear to see that a feasible solution to CRSND is also feasible to PRSND, the converse is not necessarily true. We illustrate this with a simple example, again using the same graph in Figure 1.

Example 2. Consider the graph shown in Figure 1, with the single demand $(s, t, 2)$. That is to say, if we remove a fault set F with $|F| = 1$ (i.e. a single edge), we want our respective path/cut conditions to hold.

A feasible solution to PRSND is given by the edge-set $H^* = \{su, uw, sw, wt\}$, which has cost 3 (This is in fact an optimal solution).

However, H^* is *not feasible* for CRSND. This is because we require $|\delta_H(\{s, u, w\})| \geq 2$ in any feasible CRSND solution H , otherwise we may let $F = \delta_H(\{s, u, w\})$ to show H is infeasible. This is not satisfied by H^* , as $|\delta_{H^*}(\{s, u, w\})| = 1$. In fact, one can infer that any feasible CRSND solution must include all but one of the edges from $\{su, uw, sw, sv, vw\}$. This is because for any $E' \subseteq \{su, uw, sw, sv, vw\}$ with $|E'| = 2$, one can verify that there is an s - t cut S such that $|\delta_G(S)| = 3$ and $E' \subseteq \delta_G(S)$; so $|\delta_{G-E'}(S)| \leq 1$, which implies that there is no feasible solution that excludes F . A feasible CRSND solution must also include the edge wt , so the optimal value for CRSND on this graph is 4.

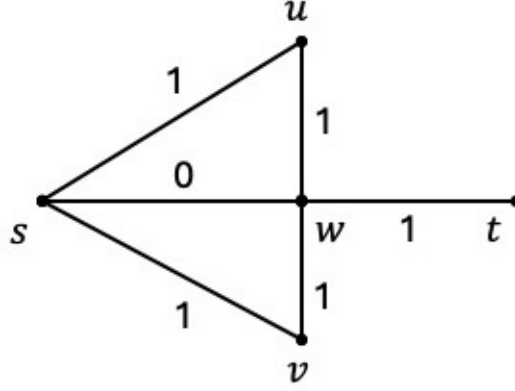


Figure 1: An example instance of SND, which illustrates both the motivation for relative problems (described in Example 1) and non-equivalence of Cut-Relative and Path-Relative SND (Example 2). Note the single source-sink pair s, t and edge costs given beside each edge.

Dinitz et al. introduced Path-Relative SND (referring to it simply as Relative Survivable Network Design), with this natural motivation as an extension of SND. Cut-Relative SND is introduced in this thesis as a somewhat natural variation of PRSND; even though these problems are not equivalent, general cut-path duality leads us to believe intuitively that solving one should give some insight into the other. While the motivation is less clear-cut for CRSND, we give some modelling applications in Section 2.6.

Though it may seem a minor change from SND, it is surprisingly hard to approximate PRSND, and in fact outside certain special cases (given in Section 1.3), no constant-factor approximation is known. As such, it is desirable to find possible techniques to approximate PRSND – and one can see that CRSND may be related enough to give some insight into this goal.

Similarly to how f -Network Design is defined in order to generalize SND, we can use cut requirement functions to define a generalized Cut-Relative Network Design problem.

Definition. The general *Cut-Relative f -Network Design* (CRND(f, G)) problem takes as input a graph $G = (V, E)$, costs $c : E \rightarrow \mathbb{R}_{\geq 0}$, and a function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$. The goal is to find a minimum-cost subgraph H where for any set $S \subseteq V$ and $F \subseteq E$ with $|F| < f(S)$,

$$\delta_{G-F}(S) \neq \emptyset \implies \delta_{H-F}(S) \neq \emptyset$$

Equivalently, any set $S \subseteq V$ has at least $\min\{f(S), |\delta_G(S)|\}$ edges on its boundary in H

(i.e. $|\delta_H(S)| \geq \min\{f(S), |\delta_G(S)|\}$).

It is straightforward to arrive at this latter equivalent statement, which is used extensively in this thesis. Consider a set S ; then for a solution H , if $|\delta_H(S)| < \min\{f(S), |\delta_G(S)|\}$, then we can take $F = \delta_H(S)$, note that $|F| < f(S)$ and $F \subset \delta_G(S)$, and see that H is infeasible. Conversely, if H is infeasible there exists some F with $|F| < f(S)$ where $\delta_{G-F}(S)$ is nonempty but $\delta_{H-F}(S)$ is empty; it must be, then, that $|\delta_H(S)| < f(S)$, and $\delta_H(S) \subset \delta_G(S)$, and so $|\delta_H(S)| < |\delta_G(S)|$.

We can see via this latter statement that Cut-Relative f -Network Design may be stated as standard f -Network Design with the cut requirement function $\min\{f(S), |\delta_G(S)|\}$. We refer to the input function f as the base cut requirement function. We show in Section 2.2 that CRSND is modelled as CRND with the base cut requirement function f^{SND} .

Recall that iterative rounding requires the cut requirement function to be weakly supermodular. A primary difficulty in finding approximate solutions to these problems is that even if the base cut requirement function f is weakly supermodular, the function $\min\{f(S), |\delta_G(S)|\}$ need not be.

1.2 Our Results

We introduce the Cut-Relative Survivable Network Design problem, and show that we can achieve an approximation guarantee of 2 for this problem – notably matching the best known approximation factor for standard SND.

Theorem 1.1. *There is a 2-approximation algorithm for Cut-Relative Survivable Network Design.*

We show as well that this extends to the general Cut-Relative f -Network Design problem based on a cut requirement function f whenever f is weakly supermodular (see definition in Section 2.2.1) and the LP relaxation of the problem can be solved efficiently. Technically, we show that Jain’s iterative rounding method (described in Section 2.5) can be applied to CRND using a decomposition technique, where we show that a CRND instance can be split up into multiple instances of standard f -Network Design. We proceed to give some notable properties of this decomposition, which may be of use in further work.

We also show that even in a simple single-demand case (a single s, t pair), both Path-Relative and Cut-Relative SND can model the k -ECSS problem, and are thus *APX*-hard. Previously, even *NP*-hardness of these problems was not known.

We go on to discuss further work in finding such a strong approximation factor for Path-Relative Survivable Network Design, demonstrating the difficulties that arise in this setting in comparison to its Cut-Relative counterpart.

1.3 Related Work

As previously mentioned, Jain gave a 2-approximation for Survivable Network Design [Jai01], introducing the method of iterative rounding – we describe the relevant details of iterative rounding in Section 2.5; a detailed survey of the usage of iterative rounding and similar methods can be found in [LRS11].

In their introductory work on Path-Relative SND [DKK22], Dinitz et al. gave approximation results for some special cases of the problem. They achieved constant-factor approximations for two cases: a 2-approximation for cases where $k_i \leq 2$ for all i , and a $\frac{27}{4}$ -approximation for the case with a single demand of $k = 3$. They also examined path-relative k -ECSS (referred to as k -Edge-Fault-Tolerant Subgraph, or k -EFTS), giving a 2-approximation for the problem generally, and a $(1 + \frac{4}{k})$ -approximation for the unweighted case. They extend their results in a follow-up [DKKN23], giving a 2-approximation where $k_i \leq 3$ for all i , and a $2^{O(k^2)}$ -approximation for the general single-demand case (i.e. with the demand (s, t, k)).

In [DKK22] they use iterative rounding to solve k -EFTS by defining a property of "local weak supermodularity", observing that the weak supermodularity property need only hold for a certain collection of vertex sets. To solve single-demand cases, they partition the instance into a "chain" of instances with certain properties, where they may solve PRSND on simpler cases; this can be seen as a form of decomposition, not unlike our own results.

Non-iterative-rounding-derived methods have also been used to study network design problems. Goemens and Williamson [GW95, GW96] introduced a primal-dual method for solving network design problems, as well as popularizing the use of cut requirement functions to model network design problems. Their method gave 2-approximations to problems where f is a 0-1 function and is proper (Definition in Section 2.2.1).

Recently, Chekuri and Jain gave a polylogarithmic ratio approximation for Path-Relative SND [CJ24]. They do this by introducing a problem called Bulk-SND, which in fact generalizes PRSND further, and defining an augmentation problem: given a solution for the same problem with lower demands, add edges such that higher demands are satisfied. They then reduce this problem to the Hitting Set problem. Their approximation for PRSND has a factor of $O(k^2 \log^2 n)$, where k is the highest demand in the PRSND instance. This is the best known factor for general PRSND. However, one could seek better guarantees,

for instance ones that avoid dependence on either k or n (for instance, a guarantee that does not depend on n would yield a constant-factor approximation for constant k). Even a constant-factor approximation for PRSND is not ruled out.

1.4 Overview

In Chapter 2 we establish notation and tools that we will use later. We formally define the network design problems that we will examine in this thesis. We proceed to give some useful properties of cut requirement functions, in particular weak supermodularity. We give a brief overview of relevant concepts: approximation algorithms, the ellipsoid method and separation oracles, and iterative rounding. We finally give some examples of the modelling power of Cut-Relative f -Network Design.

In Chapter 3 we present our main result, a decomposition method leading to a proof of Theorem 1.1.

In Chapter 4 we elaborate on the results found in the previous chapter. We show that the decomposition of Chapter 3 can be described by a tree structure, and give some useful structural properties of the decomposition tree. These properties are used to give two explicit results: that one can efficiently compute the decomposition tree, and that one can give a self-contained definition of the decomposition, which was only described recursively in Chapter 3.

In Chapter 5 we present results on the hardness Cut-Relative and Path-Relative Survivable Network Design. We show that both problems are *APX*-hard by giving a reduction to k -ECSS.

In Chapter 6 we examine Path-Relative Survivable Network Design. We give a cut-based formulation and LP relaxation for the problem, and show some results about a variant of f -Network Design, *connected* f -Network Design, which we show is related to PRSND. We proceed to describe some lines of work that we have considered for finding improved approximations for PRSND.

Chapter 2

Preliminaries

2.1 Notation

Let $G = (V, E)$ be a simple, undirected graph. Let $S \subseteq V$. Let $\bar{S} := V - S$, the complement of S . Denote by $G[S]$ the subgraph of G induced by S . We denote by $\delta_G(S)$ the set of edges on the boundary of S in G (that is, edges with one endpoint in S and the other in $\bar{S}$). For $F \subseteq E$, we denote $\delta_F(S) := \delta_G(S) \cap F$. Let $x \in \mathbb{R}_{\geq 0}^E$. Let $x(F) := \sum_{e \in F} x_e$. Let $F(S)$ be the set of edges in F with both endpoints in S . Let $\chi^F \in \{0, 1\}^E$ denote the indicator vector of F ; that is, $\chi_e^F = 1$ iff $e \in F$. Let $S, T \subseteq V$ be disjoint sets, and let $\delta_G(S, T)$ be the set of edges with one endpoint in S and one endpoint in T . Let $s, t \in V$; we call S an **s - t cut** if $|S \cap \{s, t\}| = 1$; that is, exactly one of s or t is in S . Let $S, T \subseteq V$; we say that S and T **cross** if $S \cap T$, $S - T$, and $T - S$ are all nonempty. A family $\mathcal{F}$ of vertex sets is **laminar** if no two sets in $\mathcal{F}$ cross. For $\ell \in \mathbb{Z}_{>0}$, denote $[\ell] := \{1, \dots, \ell\}$.

2.2 Problem Definitions

Given a graph $G = (V, E)$, a **cut requirement function**, or CRF, is a function $f : 2^V \rightarrow \mathbb{R}$. We are concerned with the following problems, which each take as input a graph G :

- **Survivable Network Design** ($\text{SND}(G)$): given a set of 'demands' $\{(s_i, t_i, k_i)\}_{i \in [\ell]}$, where $s_i, t_i \in V$ and $k_i \in \mathbb{N}$ for all i , find a minimum-cost subgraph H such that for all $i \in [\ell]$, there are at least k_i edge-disjoint paths connecting s_i and t_i ; or equivalently, for any s_i - t_i cut $S \subseteq V$, $|\delta_H(S)| \geq k_i$.

- **Cut-Relative Survivable Network Design** (CRSND(G)): given demands as in SND, find a minimum-cost subgraph H such that for every $i \in [\ell]$, every $F \subseteq E$ with $|F| < k_i$, and every s_i - t_i cut S , if $\delta_{G-F}(S)$ is nonempty, then so is $\delta_{H-F}(S)$.
- **Path-Relative Survivable Network Design** (PRSND(G)): given demands as in SND, find a minimum-cost subgraph H such that for every $i \in [\ell]$ and every $F \subseteq E$ with $|F| < k_i$, if there is an s_i - t_i path in $G - F$, there is an s_i - t_i path in $H - F$.
- **f -Network Design** (ND(f, G)): given a cut requirement function f , find a minimum-cost subgraph H such that $|\delta_H(S)| \geq f(S)$ for all $S \subseteq V$.
- **Cut-Relative f -Network Design** (CRND(f, G)): given a cut requirement function f , find a minimum-cost subgraph H such that $|\delta_H(S)| \geq \min\{f(S), |\delta_G(S)|\}$ for all $S \subseteq V$.

As discussed previously, SND(G) can be modelled by ND(f^{SND}, G), where $f^{SND}(S) := \max_{i: |S \cap \{s_i, t_i\}|=1} \{k_i\}$. Here, we show that the same base CRF can model Cut-Relative SND; that is, CRSND(G) can be modelled by CRND(f^{SND}, G).

Proposition 2.1. CRSND(G) $\equiv$ CRND(f^{SND}, G)

Proof. Refer to CRSND(G) as (P1) and CRND(f^{SND}, G) as (P2). Suppose H is feasible for (P1) but not (P2), meaning there exists some $S \subseteq V$ where $|\delta_H(S)| < \min\{f^{SND}(S), |\delta_G(S)|\}$. Let $k = f^{SND}(S)$. Take $F = \delta_H(S)$. We thus have $|F| < k$, $\delta_{G-F}(S) \neq \emptyset$, but $\delta_{H-F}(S) = \emptyset$, giving a contradiction.

Conversely, suppose H is feasible for (P2). Then we may consider any $i \in [\ell]$, any $F \subseteq E$ with $|F| < k_i$, and any s_i - t_i cut S with $\delta_{G-F}(S) \neq \emptyset$, and obtain $|\delta_F(S)| < \min\{k_i, |\delta_G(S)|\} \leq \min\{f(S), |\delta_G(S)|\} \leq |\delta_H(S)|$; so $\delta_{H-F}(S) \neq \emptyset$ and thus H is feasible for (P1). $\square$

We show similarly in Chapter 5 that PRSND can be modelled by requiring the same $|\delta_H(S)| \geq \min\{f^{SND}(S), |\delta_G(S)|\}$ constraints, but in this case only for a certain collection of sets.

We will often refer instead to the linear program relaxations of these problems. We list here the LP relaxations of ND(f, G) and CRND(f, G), respectively:

$$\begin{aligned}
& \text{minimize:} && \sum_{e \in E} c_e x_e && (NDLP(f, G)) \\
& \text{subject to:} && x(\delta_G(S)) \geq f(S), && S \subseteq V \\
& && 0 \leq x_e \leq 1, && e \in E
\end{aligned}$$

$$\begin{aligned}
& \text{minimize:} && \sum_{e \in E} c_e x_e && (CRLP(f, G)) \\
& \text{subject to:} && x(\delta_G(S)) \geq \min\{f(S), |\delta_G(S)|\}, && S \subseteq V \\
& && 0 \leq x_e \leq 1, && e \in E
\end{aligned}$$

Note that integer solutions to $NDLP(f, G)$ and $CRLP(f, G)$ correspond precisely to feasible solutions to $ND(f, G)$ and $CRND(f, G)$ respectively. That is, for $H \subseteq E$, if $x = \chi^H$, the indicator vector of H , then x is feasible to $NDLP(f, G)$ (resp. $CRLP(f, G)$) iff H is feasible to $ND(f, G)$ (resp. $CRND(f, G)$).

2.2.1 Properties of Cut-Requirement Functions

We list here a number of properties for cut requirement functions.

Definition. Let $f : 2^V \rightarrow \mathbb{R}$. We say that f is:

- **weakly supermodular** if for all $A, B \subseteq V$,

$$f(A) + f(B) \leq \max\{f(A \cup B) + f(A \cap B), f(A - B) + f(B - A)\}$$

Note that equivalently, we can say that $f(A) + f(B) \leq f(A \cup B) + f(A \cap B)$ or $f(A) + f(B) \leq f(A - B) + f(B - A)$.

- **symmetric** if $f(S) = f(\bar{S})$
- **normalized** if $f(\emptyset) = f(V) = 0$
- **submodular** if $f(A) + f(B) \geq \max\{f(A \cup B) + f(A \cap B), f(A - B) + f(B - A)\}$

We will proceed by showing some basic properties of cut requirement functions.

Proposition 2.2. *For any symmetric CRF f and sets $A, B \subseteq V$ that do not cross, the weak supermodularity condition holds (that is to say, $f(A) + f(B) \leq \max\{f(A \cup B) + f(A \cap B), f(A - B) + f(B - A)\}$).*

Proof. Suppose either $A - B$ or $B - A$ is empty; this means either $A \subseteq B$ or $B \subseteq A$, and therefore $f(A) + f(B) = f(A \cup B) + f(A \cap B)$.

Now suppose $A \cap B$ is empty, then $\{A - B, B - A\} = \{A, B\}$ and so $f(A) + f(B) = f(A - B) + f(B - A)$. $\square$

We can cast any CRF f in a symmetrized and normalized form f^{sym} , defined as

$$f^{sym}(\emptyset) = f^{sym}(V) = 0$$

$$f^{sym}(S) = \max\{f(S), f(\bar{S})\} \text{ for } \emptyset \neq S \subset V$$

This allows us to cast any $\text{ND}(f, G)$ problem as an equivalent problem with the added nice properties of symmetry and normalization:

Lemma 2.3. *Let f^{sym} be defined as above.*

(a) *If f is weakly supermodular, then so is f^{sym}*

(b) *If f -ND has a feasible solution, then $\text{ND}(f, G)$ is equivalent to $\text{ND}(f^{sym}, G)$.*

Proof. Part (a): Let $A, B \subseteq V$. Since f^{sym} is symmetrical by definition, by Prop. 2.2 we may assume A and B cross.

Let $X \in \{A, V - A\}$ such that $f(X) = f^{sym}(A)$; similarly let $Y \in \{B, V - B\}$ where $f(Y) = f^{sym}(B)$. So

$$\begin{aligned} f^{sym}(A) + f^{sym}(B) &\leq \max\{f(X \cap Y) + f(X \cup Y), f(X - Y) + f(Y - X)\} \\ &\leq \max\{f^{sym}(X \cap Y) + f^{sym}(X \cup Y), f^{sym}(X - Y) + f^{sym}(Y - X)\}. \end{aligned}$$

We show that

$$f^{sym}(X \cap Y) + f^{sym}(X \cup Y) \in \{f^{sym}(A \cap B) + f^{sym}(A \cup B), f^{sym}(A - B) + f^{sym}(B - A)\},$$

and

$$f^{sym}(X - Y) + f^{sym}(Y - X) \in \{f^{sym}(A \cap B) + f^{sym}(A \cup B), f^{sym}(A - B) + f^{sym}(B - A)\}.$$

which will complete the proof. We proceed by cases:

- $X = A$ and $Y = B$: the condition is clearly satisfied.
- $X = A$, $Y = V - B$: We have $X \cap Y = A - B$, and $X \cup Y = V - (B - A)$, so $f^{sym}(X \cup Y) = f^{sym}(B - A)$. Also, $X - Y = A \cap B$, and $Y - X = V - (A \cup B)$, so $f^{sym}(Y - X) = f^{sym}(A \cup B)$.
- $X = V - A$, $Y = B$: This is symmetric to the above case.

- $X = V - A$, $Y = V - B$: We have $X \cap Y = V - (A \cup B)$, so $f^{sym}(X \cap Y) = f^{sym}(A \cup B)$, and $X \cup Y = V - (A \cap B)$, so $f^{sym}(X \cup Y) = f^{sym}(A \cap B)$. Also, $X - Y = B - A$ and $Y - X = A - B$.

Part (b): Let $S \subseteq V$. Note that if $S \in \{\emptyset, V\}$, then it must be that $f(S) \leq 0$; otherwise, $x(\delta_G(S)) = 0 < f(S)$ for any solution x , and therefore f -ND is infeasible. Therefore, we find that for any x , $x(\delta_G(S)) \geq f(S)$ and $x(\delta_G(S)) \geq f^{sym}(S)$.

Now suppose $S \notin \{\emptyset, V\}$. If x is feasible to f^{sym} -ND, then for all $S \subseteq V$, $x(\delta_G(S)) \geq \max\{f(S), f(\bar{S})\} \implies x(\delta_G(S)) \geq f(S)$.

If x is feasible to f -ND, then $x(\delta_G(S)) \geq f(S)$ and $x(\delta_G(S)) = x(\delta_G(\bar{S})) \geq f(\bar{S})$, and therefore $x(\delta_G(S)) \geq f^{sym}(S)$. $\square$

As a result of Lemma 2.3, for any f -ND or similar problem, we may assume f is symmetric and normalized whenever it is weakly supermodular.

We now give some operations that preserve weak supermodularity, which will prove useful for later proofs.

Lemma 2.4. *Let f be weakly supermodular and g be submodular. Then $g_1(S) := f(S) - g(S)$ is weakly supermodular.*

Proof. Let $A, B \subseteq V$, and $g'(S) = -g(S)$, and note that $g'(A) + g'(B) \leq g'(A \cup B) + g'(A \cap B)$ and $g'(A) + g'(B) \leq g'(A - B) + g'(B - A)$. Now $g_1(S) = f(S) + g'(S)$. If $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$, then

$$\begin{aligned} g_1(A) + g_1(B) &= f(A) + f(B) + g'(A) + g'(B) \\ &\leq f(A \cup B) + f(A \cap B) + g'(A \cup B) + g'(A \cap B) \\ &\leq g_1(A \cup B) + g_1(A \cap B) \end{aligned}$$

And similarly if $f(A) + f(B) \geq f(A - B) + f(B - A)$, we use $g'(A) + g'(B) \leq g'(A - B) + g'(B - A)$ to see $g_1(A) + g_1(B) \leq g_1(A - B) + g_1(B - A)$. $\square$

Lemma 2.5. *Let $x \in \mathbb{R}_{\geq 0}^E$. Let $g_2(S) = f(S) - x(\delta_G(S))$ for $S \subseteq V$. Then if f is weakly supermodular, then so is g_2 , and if f is symmetric, then so is g_2 .*

Proof. Symmetry is clear, as the sum of two symmetric functions must also be symmetric. We gain weak supermodularity by Lemma 2.4 by seeing that $x(\delta_G(S))$ is submodular for any x . $\square$

For some $F \subseteq E$, considering $x = \chi^F$ gives us the following corollary:

Corollary 2.6. *Let f be weakly supermodular, and let $F \subseteq E$. Let $g_2(S) = f(S) - |\delta_F(S)|$ for $S \subseteq V$. Then g_2 is weakly supermodular.*

We give one more useful operation:

Definition. Let $X \subseteq V$, and let $\mathcal{C}$ be a family of subsets of $\overline{X}$ that is closed under set intersection, union, difference, and complements (with respect to $\overline{X}$). The **projection** of f onto X with respect to $\mathcal{C}$ is a function $g_3 : 2^X \rightarrow \mathbb{R}$ defined by $g_3(\emptyset) = g_3(X) = 0$ and $g_3(S) = \max_{T \in \mathcal{C}} f(S \cup T)$ for $\emptyset \neq S \subset X$.

We can think of this projection operator as follows: suppose f is a cut-requirement function, and we know that no edges are picked from $E(V - X) \cup \delta_G(X)$. Then, roughly speaking, $g_3(S)$ seeks to capture the constraints that arise on $S \subset X$ due to f and $\mathcal{C}$.

Lemma 2.7. *If f is weakly supermodular and symmetric, then g_3 is weakly supermodular and symmetric.*

Proof. We begin with symmetry. Note that if $S \in \{\emptyset, X\}$, then $g_3(S) = g_3(X - S)$ by definition. Suppose $S \notin \{\emptyset, X\}$, and let $T_S \in \mathcal{C}$ such that $g_3(S) = f(S \cup T_S)$. We obtain

$$\begin{aligned} f(S \cup T_S) &= f(V - (S \cup T_S)) \\ &= f((X - S) \cup (V - X - T_S)) \\ &\leq g_3(X - S) \end{aligned}$$

Since $(V - X - T_S) \in \mathcal{C}$. By a symmetric argument we obtain $g_3(X - S) \leq g_3(S)$ by the same type of reasoning, and so $g_3(S) = g_3(X - S)$.

We now show weak supermodularity. Let A, B , and by Proposition 2.2 we may assume A and B cross. Let $T_A, T_B \in \mathcal{C}$ where $g_3(A) = f(A \cup T_A)$ and $g_3(B) = f(B \cup T_B)$. Let $A' = A \cup T_A$ and $B' = B \cup T_B$.

Let $S \in \{A' \cap B', A' \cup B', A' - B', B' - A'\}$, and note that, respectively, $S \cap X \in \{A \cap B, A \cup B, A - B, B - A\}$. We can write $S = (S \cap X) \cup T$, for some $T \in \mathcal{C}$, due to closure of $\mathcal{C}$. Therefore, $g_3(S \cap X) \geq f(S)$.

So if $f(A') + f(B') \leq f(A' \cap B') + f(A' \cup B')$, we obtain that

$$g_3(A) + g_3(B) = f(A') + f(B') \leq f(A' \cap B') + f(A' \cup B') \leq g_3(A \cap B) + g_3(A \cup B)$$

Similarly if $f(A') + f(B') \leq f(A' - B') + f(B' - A')$, then $g_3(A) + g_3(B) \leq g_3(A - B) + g_3(B - A)$.

□

Definition. A set S is an (f, G) –**small cut** if $|\delta_G(S)| < f(S)$.

In situations without ambiguity, we may simply call an (f, G) -small cut a "small cut".

We call a set a "large cut" if it is not a small cut.

Definition. Let f be a cut requirement function, and $F \subseteq E$. Define the **deficiency** of S as $\text{def}_{f,F}(S) := f(S) - |\delta_F(S)|$.

In situations without ambiguity, we may denote $\text{def}(S) := \text{def}_{f,E}(S)$

Note that small cuts have positive deficiency. Observe as well that if f is weakly supermodular and symmetric then so is deficiency by Lemma 2.5.

We will find it convenient to define the set of all edges that are in small cuts:

$$\begin{aligned} E_{\text{small}} &:= \{e \in E : \exists S, S \text{ is a small cut}, e \in \delta_G(S)\} \\ &= \bigcup_{S \subseteq V : \text{def}(S) > 0} \{\delta_G(S)\} \end{aligned}$$

2.3 Approximation Algorithms

We briefly define the relevant terminology pertaining to approximation.

Definition. Let P be a minimization problem, OPT be the optimal solution value of P , and $\alpha > 1$. An α –**approximation algorithm** to P is an algorithm that returns a feasible solution to P with value at most $\alpha \cdot OPT$.

2.4 Ellipsoid Method

Since the LP relaxations of our problems (Section 2.2) involve an exponential number of constraints (one for each subset of V), we employ the ellipsoid method of solving linear programs in order to solve linear relaxations of our network design problems.

In the ellipsoid method, the key crucial element we need is a *separation oracle* for our specific problem: an algorithm that, given a problem and a solution, either determines that the solution is feasible, or gives a constraint that is violated by the infeasible solution. Our ability to use the ellipsoid method in polynomial time to solve a linear program depends on whether there exists a polynomial-time separation oracle.

We indeed have separation oracles for some of our relevant problems.

Fact 2.8. [GLS93] *There exists a polynomial-time separation oracle for the linear relaxation of Survivable Network Design.*

We obtain a separation oracle for CRSND by adapting some ideas from [DKK22], where they find one for path-relative k -ECSS.

Lemma 2.9. *There exists a polynomial-time separation oracle for the linear relaxation of Cut-Relative Survivable Network Design.*

Proof. Observe that the constraints of $\text{CRLP}(f, G)$ can be stated as follows: for a solution x , if we consider $\{x_e\}_{e \in E}$ to represent edge capacities, then x is feasible iff for any edge $uv \in E$, either $x_{uv} = 1$ or for all i the minimum s_i - t_i cut that also separates u and v has capacity at least k_i . The first case encompasses edges in E_{small} , the second covers others.

To detect if x is a feasible solution, we need only consider all uv with $x_{uv} < 1$ and all i , and find the minimum s_i - t_i cut that also separates u and v . So we compute both the minimum $\{s_i, u\}$ - $\{t_i, v\}$ cut and $\{s_i, v\}$ - $\{t_i, u\}$ cut and take the minimum-capacity of those.

If x is a feasible solution, we will find that all computed min cuts have capacity at least k_i . If not, we will find a cut with capacity less than k_i , which corresponds to a violated constraint. $\square$

2.5 Iterative Rounding

We will be making use of Jain's method of iterative rounding to solve f -ND in our main results here. As such, we describe here the primary details of this technique, a full description of which is found in [Jai01]. The key theorem that enables iterative rounding is the following:

Theorem 2.10 ([Jai01]). *Let f be a cut-requirement function such that f is weakly supermodular. Then in any extreme-point feasible solution x to $\text{NDLP}(f, G)$, there exists at least one edge e such that $x_e \geq \frac{1}{2}$.*

This allows one to take every such $x_e \geq \frac{1}{2}$ and fix it to 1, and then iteratively solve the problem with the added fixed constraint, increasing our solution value by a factor of at most 2 with each iteration. A more detailed description of the algorithm will be given later; we will here briefly discuss the crucial steps to showing this result.

Recall that two sets $S, T \subseteq V$ cross if $S \cap T$, $S - T$, and $T - S$ are nonempty, and that a family $\mathcal{F}$ of sets is laminar if no two sets in $\mathcal{F}$ cross. Denote by $\mathcal{A}_G(S)$ the row of the constraint matrix of $\text{NDLP}(f, G)$ corresponding to the constraint given by S (that is, $\mathcal{A}_G(S) : \{0, 1\}^{|E|}$, where the entry corresponding to $e \in E$ is 1 iff $e \in \delta_G(S)$). Finally, let x be some extreme-point solution to $\text{NDLP}(f, G)$. We assume x is entirely fractional, as if

there is an edge with value 1, then the theorem holds trivially, and if there is an edge with value zero, we may assume that that edge does not exist without changing the cost of our solutions.

Definition. Let x be a solution to $\text{NDLP}(f, G)$. Call a set $S \subseteq V$ **tight** if $x(\delta_G(S)) = f(S)$.

In order to show Theorem 2.10, Jain uses weak supermodularity of f to show that given an extreme-point solution x , there exists a laminar family of tight sets (with respect to x) that satisfy certain properties. From there, he uses a token-counting argument to show that there must exist an edge e with $x_e \geq \frac{1}{2}$. In looking at problems where we wish to apply iterative rounding, such as Cut-Relative f -Network Design, it is thusly desirable to show that LP solutions may be described via these nice laminar families of tight sets. Specifically, Jain uses the following:

Lemma 2.11 ([Jai01]). *For an extreme-point solution x to $\text{NDLP}(f, G)$, there exists a laminar family $\mathcal{B}$ of tight sets satisfying the following:*

- (i) *The vectors $\{\mathcal{A}_G(S)\}_{S \in \mathcal{B}}$ are linearly independent.*
- (ii) $|\mathcal{B}| = |E(G)|$
- (iii) *For every $S \in \mathcal{B}$, $f(S) \geq 1$.*

The crucial piece of this lemma is item (i), as the remaining items can be argued simply using this linear independence. Further, showing item (i) reveals why weak supermodularity of f is critical, as we can use it to *uncross* sets to form our laminar family. Specifically, Jain shows the following.

Lemma 2.12 ([Jai01]). *If two sets $A, B \subseteq V$ are tight, then one of the following must hold:*

1. $A \cup B$ and $A \cap B$ are also tight, and $\mathcal{A}_G(A) + \mathcal{A}_G(B) = \mathcal{A}_G(A \cup B) + \mathcal{A}_G(A \cap B)$.
2. $A - B$ and $B - A$ are also tight, and $\mathcal{A}_G(A) + \mathcal{A}_G(B) = \mathcal{A}_G(A - B) + \mathcal{A}_G(B - A)$.

Proof. Note first that if A and B do not cross, then the lemma is trivially true by plugging in whichever of the relevant sets is empty. We therefore assume that A and B cross. Denote $S_1 := A - B$, $S_2 := A \cap B$, $S_3 := B - A$, and $S_4 := V - (A \cup B)$. Since A and B are tight, we have

$$\begin{aligned} f(A) &= x(S_1, S_3) + x(S_1, S_4) + x(S_2, S_3) + x(S_2, S_4) \\ f(B) &= x(S_1, S_2) + x(S_1, S_3) + x(S_2, S_4) + x(S_3, S_4) \end{aligned}$$

And by feasibility we also obtain

$$\begin{aligned} f(S_2) &\leq x(S_2, S_1) + x(S_2, S_3) + x(S_2, S_4) \\ f(V - S_4) &\leq x(S_4, S_1) + x(S_4, S_2) + x(S_4, S_3) \end{aligned}$$

(where for the latter we use symmetry of x). Using weak supermodularity, suppose $f(A) + f(B) \leq f(A \cup B) + f(A \cap B) = f(V - S_4) + f(S_2)$. Then adding inequalities we obtain $2x(S_1, S_3) \leq 0$; so, $x(S_1, S_3) = 0$, and plugging this equality back into the inequalities yields that S_2 and S_4 (again by symmetry of x) are both tight. Since we assumed there are no edges with value 0, $x(S_1, S_3) = 0$ implies that there are no edges between S_1 and S_3 , and therefore $\mathcal{A}_G(A) + \mathcal{A}_G(B) = \mathcal{A}_G(S_2) + \mathcal{A}_G(S_4)$.

We can show the other item with a similar argument in the case where $f(A) + f(B) \leq f(A - B) + f(B - A) = f(S_1) + f(S_3)$. $\square$

We may now proceed to the algorithmic description. As previously described, in iterative rounding we take all edges where the optimal LP solution gives the edge a value at least $\frac{1}{2}$, and fix them to have value 1. Then we iteratively solve the problem again, where these edges have this fixed value.

More specifically, let $Z_1 \subseteq E$ be some set of edges that we have fixed to a value of 1. Let $E' = E - Z_1$ and let $G' := (V, E')$; we call G' the *residual graph* after fixing Z_1 . The following LP models f -ND on the residual graph, where we remove the requirement from edges that we fix to 1.

$$\begin{aligned} \text{minimize:} \quad & \sum_{e \in E} c_e x_e && (\text{NDLP}_{res}(f, G, Z_1)) \\ \text{subject to:} \quad & x(\delta_{G'}(S)) \geq f(S) - |\delta_{Z_1}(S)|, && S \subseteq V \\ & 0 \leq x_e \leq 1, && e \in E' \end{aligned}$$

(recalling $|\delta_{Z_1}(S)| = |\delta_G(S) \cap Z_1|$) By the Corollary to Lemma 2.5, the cut requirement function of NDLP_{res} is weakly supermodular, and thus NDLP_{res} is of the same form as NDLP (specifically, we can see this is $\text{NDLP}(f(S) - |\delta_{Z_1}(S)|, G')$). This preserved form leads to two helpful results. First, Theorem 2.10 applies to NDLP_{res} and we continue to find edges with value at least $\frac{1}{2}$. Secondly, Fact 2.8 gives us that there is a separation oracle for the residual problem.

The high level description of the algorithm, as given by Jain [Jai01] is as follows (see Algorithm 1 for a rigorous description):

1. Find an optimal extreme-point solution to $\text{NDLP}(f, G)$.

2. Add all edges with solution value at least $\frac{1}{2}$ to the set Z_1 .
3. Solve the residual problem, $\text{NDLP}_{\text{res}} = \text{NDLP}(f(S) - |\delta_{Z_1}(S)|, G')$.
4. Repeat steps 2 and 3 until a feasible solution is found.

Algorithm 1: f -Network Design 2-Approximation

Input : Graph $G = (V, E)$, costs $\{c_e\}_{e \in E}$, function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$

Output: A subgraph $H \subseteq E$ where $|\delta_H(S)| \geq f(S)$ for all $S \subseteq V$

Assume we have a separation oracle for $\text{NDLP}(f, G)$

Initialize $H \leftarrow \emptyset$, $G' = (V, E') \leftarrow G$, $f' \leftarrow f$;

while H is not a feasible solution (detected via separation oracle) **do**

Compute an extreme-point optimal solution $\hat{x}$ to $\text{NDLP}(f', G')$;

Let $Z_1 = \{e \in E' : \hat{x}_e \geq 1/2\}$, update $H \leftarrow H \cup Z_1$, $E' \leftarrow E' - Z_1$, and

$f'(S) = f'(S) - |\delta_{Z_1}(S)|$ for all $S \subseteq V$;

end

Return H

The following theorem, again proven by Jain, shows that this indeed gives a 2-approximation algorithm for f -ND.

Theorem 2.13 ([Jai01]). *Let OPT be the optimal value of $\text{NDLP}(f, G)$. Let x be an integer extreme-point solution to $\text{NDLP}_{\text{res}}(f, G, Z_1)$. Let $E_x := \{e \in E : x_e = 1\}$. Then $E_x \cup Z_1$ is an integer solution to $\text{NDLP}(f, G)$ with value at most $2 \cdot OPT$.*

2.5.1 Application to Cut-Relative Network Design

The main takeaway from the previous section is that the property that extreme-point solutions may be described via laminar families of linearly-independent tight sets is crucial for iterative rounding. Naturally, then, one may wish to find if such a property exists for "relative" problems.

Dinitz et al. [DKK22] used this approach for approximating the k -Edge-Fault-Tolerant Subgraph problem (which may be alternatively modelled as Cut-Relative k -ECSS). While the cut-requirement function for this problem was not weakly supermodular, they leveraged a property of the CRF they called *local* weak supermodularity, using the insight that in the proof of Lemma 2.12 that the weak supermodularity condition needed only hold for a

certain collection of sets. They were thus able to find a useful laminar family description and use iterative rounding to achieve a 2-approximation.

However, even this refinement breaks down for general cases. Indeed, one can construct simple examples of Cut-Relative f -ND instances where an extreme-point solution cannot be described by a suitable laminar family.

Example 3. Consider the graph G and edge costs shown in Figure 1. We define a CRSND instance on G with the single demand $(s, t, 2)$, or s - t 2-connectivity, which can be modelled with a cut requirement function f where $f(S) = 2$ if S is an s - t cut, and $f(S) = 0$ otherwise. For the LP relaxation of this instance, $\text{CRLP}(f, G)$, we give a feasible solution $\hat{x}$, shown in Figure 2. We show that $\hat{x}$ is an extreme-point solution that cannot be described by a laminar family of tight sets.

It is straightforward to confirm that $\hat{x}$ is feasible. We show that it is an extreme-point solution by showing it is the unique solution to a collection of tight constraints. Namely, consider the following equations, which are the constraints that are tight at $\hat{x}$:

$$x(\delta_G(S)) = 2 \quad \forall S \in \{\{s\}, \{s, u\}, \{s, u, w\}, \{s, v, w\}\} \quad (2.1)$$

$$x_e = 1 \quad \forall e \in \{sw, wt\} \quad (2.2)$$

We show that $\hat{x}$ is the unique solution to these constraints. If we set $x_{sw} = x_{wt} = 1$, then considering the constraint equations for pairs of sets in 2.1, we find that all edges in $\{su, uw, sv, vw\}$ must have the same x_e value:

$$\begin{aligned} x(\delta_G(\{s\})) = 2 = x(\delta_G(\{s, u\})) &\implies x_{sw} + x_{su} + x_{sv} = x_{sw} + x_{sv} + x_{uw} \implies x_{su} = x_{uw} \\ x(\delta_G(\{s\})) = 2 = x(\delta_G(\{s, u, w\})) &\implies x_{sw} + x_{su} + x_{sv} = x_{wt} + x_{sv} + x_{vw} \implies x_{su} = x_{vw} \\ x(\delta_G(\{s\})) = 2 = x(\delta_G(\{s, u, v\})) &\implies x_{sw} + x_{su} + x_{sv} = x_{sw} + x_{uw} + x_{vw} \implies x_{sv} = x_{vw} \end{aligned}$$

This fixes the values of all edges in $\{su, uw, sv, vw\}$ to 0.5. Thus, $\hat{x}$ is the unique solution to these 6 equations, and is therefore an extreme-point solution.

We now show $\hat{x}$ cannot be described by a laminar family of linearly independent tight cut constraints and tight edge constraints. Note that since no edges have $x_e = 0$, any tight set must be an s - t cut. The definition of laminarity implies then that any laminar family $\mathcal{L}$ is the union of two chains $\mathcal{L}_1 \cup \mathcal{L}_2$, where each chain is a nested family of sets; $\mathcal{L}_1$ comprises sets containing s and $\mathcal{L}_2$ comprises sets containing t . However, using symmetry of constraints and laminarity we see that $\mathcal{L}' := \mathcal{L}_1 \cup \{V - S : S \in \mathcal{L}_2\}$ is also a laminar family, where all sets in $\mathcal{L}'$ contain s and not t . Thus, we may assume that any laminar family is a chain of nested sets containing s and not t .

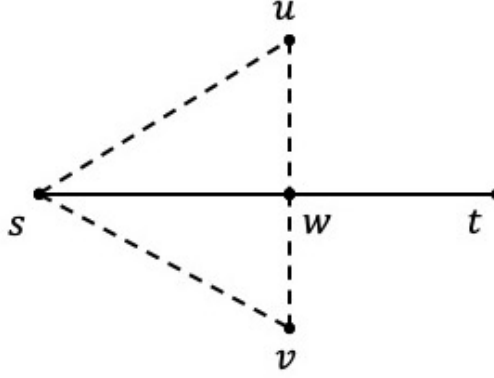


Figure 2: The $\text{CRND}(f, G)$ instance from Figure 1 and an extreme-point solution $\hat{x}$ that cannot be described by a laminar family (as described in Example 3). The solid edges have $\hat{x}_e = 1$, the dotted edges have $\hat{x}_e = 0.5$.

We now show no such chain, along with the tight edge constraints $x_{sw} = x_{wt} = 1$ (2.2), can uniquely define $\hat{x}$. Since $\hat{x}$ has 6 variables, the chain must consist of 4 tight cut constraints such that these constraints along with the 2 tight edge constraints $x_{sw} = x_{wt} = 1$ are linearly independent.

Observe that any set in this chain is of the form $\{s\} \cup T$, where $T \subseteq \{u, v, w\}$; hence, every possible chain of 4 distinct sets is obtained by ordering the elements u, v, w , and letting T range over all prefixes of this ordered sequence (including the empty prefix). Every such chain of 4 sets must contain the sets $\{s\}$ and $\{s, u, v, w\}$. However, since $x(\delta_G(\{s, u, v, w\})) = x_{wt} = 1$, the constraint given by $\{s, u, v, w\}$ is equivalent to one of our tight edge constraints. Therefore, any chain defines a system of only 5 constraints, and since $\hat{x}$ has 6 variables, no chain of tight cut constraints can uniquely define $\hat{x}$.

2.6 Modelling Power of CRND

While Path-Relative SND has a very natural and applicable motivation, this is somewhat less clear for Cut-Relative f -ND problems. One motivation for studying CRND is to gain insight into Path-Relative SND. However, we find that CRND has useful modelling power:

We introduce the following two problems here, and show that they can be modelled by CRND. Both of these problems are concerned with *graceful relative-fault-tolerance degradation*.

Definition. The Grace-Size-SND problem takes a graph G and a *non-increasing* function $\tau : \mathbb{Z}_{>0} \rightarrow \mathbb{Z}_{>0}$. The goal is to find a minimum-cost subgraph H such that for any fault set $F \subseteq E$, $G - F$ and $H - F$ have exactly the same connected components with at most $\tau(|F|)$ nodes.

Suppose G represents a communication network, where connectedness represents communication capability between two nodes; then this models the idea that after some number $|F|$ of faults, if a node can communicate with more than $\tau(|F|)$ nodes in G , then it can do the same in H , and if it can communicate with at most $\tau(|F|)$ nodes in G , then it can communicate with exactly the same nodes in H .

Recall that the k -Edge-Fault-Tolerant Subgraph problem, or k -EFTS, is the path-relative version of k -ECSS, or a PRSND problem with demands (u, v, k) for every $u, v \in V$. Note that for some k , if $\tau(k - 1) \geq n = |V|$, then H is feasible for k -EFTS.

We claim that Grace-Size-SND is equivalent to CRND with cut requirement function f^{grace} , where $f^{grace}(S) := \min \{\ell : \tau(\ell) < |S|\}$, for $S \neq \emptyset$, and $f^{grace}(\emptyset) = 0$.

Proposition 2.14. *H is feasible for $\text{CRND}(f^{grace}, G)$ iff for every fault set $F \subseteq E$ and S with $|S| \leq \tau(|F|)$, $G[S]$ is a component of $G - F$ iff it is a component of $H - F$.*

Proof. First, note that the function f^{grace} is constructed so that for any $S' \subseteq V$ and $E' \subseteq E$, we have $|S'| \leq \tau(|E'|)$ iff $f^{grace}(S') > |E'|$. So, if $|S| \leq \tau(|F|)$, $|\delta_H(S)| \geq f^{grace}(S)$ implies that $|S| > \tau(|\delta_H(S)|)$ and thus $|\delta_H(S)| < |F|$, which means that S is not a component of $H - F$.

Suppose H is such that $|\delta_H(S)| \geq \min\{f^{grace}(S), |\delta_G(S)|\}$ for all $S \subseteq V$. Consider any fault-set $F \subseteq E$ and any $S \subseteq V$ such that $|S| \leq \tau(|F|)$. By the above, if S is a component of $H - F$, then $|\delta_H(S)| < f^{grace}(S)$, and so S must also be a component of $G - F$. But this also implies that if S is a component of $G - F$, then it is a component of $H - F$; otherwise, there is some $A \subsetneq S$ that is a component of $H - F$ with $|A| < |S| \leq \tau(|F|)$, and so A must be a component of $G - F$, which is not the case.

Conversely, suppose that for every $F \subseteq E$, and $S \subseteq V$ with $|S| \leq \tau(|F|)$, we have that S is a component of $G - F$ iff it is a component of $H - F$. If $|\delta_H(S)| < \min\{f^{grace}(S), |\delta_G(S)|\}$ for some $S \subseteq V$, then take $F = \delta_H(S)$. Since $f^{grace}(S) > |F|$, we have $|S| \leq \tau(|F|)$. But S is a component of $H - F$ and not $G - F$, which yields a contradiction. $\square$

Lemma 2.15. *f^{grace} is weakly supermodular.*

Proof. The function f^{grace} is monotone decreasing (i.e. if $\emptyset \neq T \subseteq S$, then $f^{grace}(T) \geq f^{grace}(S)$). This follows from the definition, since τ is a non-increasing function.

In fact, we may note that $f^{grace}(S)$ is a function only of the size of S , and is monotone decreasing with respect to $|S|$. Observe that if $|T| \leq |S|$ then $f^{grace}(T) \geq f^{grace}(S)$, and $|T| = |S|$ iff $f^{grace}(T) \geq f^{grace}(S)$ (again assuming $\emptyset \neq T, S$).

We can thus get weak supermodularity: if $A - B, B - A \neq \emptyset$ then $f^{grace}(A) + f^{grace}(B) \leq f^{grace}(A - B) + f^{grace}(B - A)$, and otherwise $f^{grace}(A) + f^{grace}(B) = f^{grace}(A \cap B) + f^{grace}(A \cup B)$. $\square$

Note that f^{grace} is not itself symmetric nor normalized, but we may replace it with its symmetrized, normalized form by Lemma 2.3.

The following problem generalizes Grace-Size-SND, where the fault-tolerance degradation depends on monotone properties of sets other than their size. For example, suppose $\pi(S) = \max_{u,v \in S} \min_{u-v \text{ paths } P \text{ in } G} |P|$ is the weak-diameter of S . Setting $\tau(\ell) = n$ if $\ell < k$, and some $t < n$ for larger ℓ , we seek a k -EFTS solution H such that for $F \subseteq E$ with $|F| \geq k$, $G - F$ and $H - F$ have the same components of weak-diameter at most t .

Definition. The Grace-SND problem takes a graph G , a non-increasing function $\tau : \mathbb{Z}_{>0} \rightarrow \mathbb{Z}_{>0}$, and a *monotone* function $\pi : 2^V \rightarrow \mathbb{R}_{>0}$ (i.e. for $T \subseteq S$, $\pi(T) \leq \pi(S)$). The goal is to find a minimum-cost subgraph H such that for any fault set $F \subseteq E$ and $S \subseteq V$ with $\pi(S) \leq \tau(|F|)$, S is (the node-set of) a component of $G - F$ iff it is a component of $H - F$.

We once again show modelling via CRND, now with function f^{grace2} where $f^{grace2} := \min \{\ell : \tau(\ell) < \pi(S)\}$, for $S \neq \emptyset$.

Proposition 2.16. H is feasible for $\text{CRND}(f^{grace2}, G,)$ iff for every fault set $F \subseteq E$ and S with $\pi(S) \leq \tau(|F|)$, $G[S]$ is a component of $G - F$ iff it is a component of $H - F$.

Proof. We can see this by a very similar argument as the proof of Prop. 2.14, using monotonicity of π . $\square$

Lemma 2.17. f^{grace2} is weakly supermodular.

Proof. We need only observe that since π is monotone, f^{grace2} is monotone decreasing, and thus weakly supermodular using the same argument as in Lemma 2.15. $\square$

Chapter 3

A Decomposition Method To Approximate CRND

In this section, we prove Theorem 1.1, giving our decomposition technique to find a 2-approximation for Cut-Relative f -Network Design when f is weakly supermodular.

Recall that $\text{CRND}(f, G)$ denotes the problem on a graph G where we seek a min-cost solution containing at least $\min\{f(S), |\delta_G(S)|\}$ edges from $\delta_G(S)$ for each $S \subseteq V$, and that we refer to f as the base cut-requirement function (CRF). Throughout this chapter, we assume that f is weakly supermodular. As shown in Lemma 2.3, we may assume without loss of generality that f is symmetric and normalized. Recall that although the base CRF is weakly supermodular, this does not necessarily mean that the cut-requirement function $\min\{f(S), |\delta_G(S)|\}$ defining $\text{CRND}(f, G)$ is weakly supermodular.

At a high level, we partition G into subgraphs and define instances of standard f -ND on each subgraph such that the solutions of these f -ND instances combined with the edges on the boundaries of these subgraphs yield $\text{CRND}(f, G)$ solution. We use this correspondence to show that iterative rounding can be applied to CRND.

Given $x \in \mathbb{R}^E$ and a subset $S \subseteq V$, we use x^S to denote $\{x_e\}_{e \in E(S)}$, i.e., x restricted to the edges in $E(S)$.

Our result is formally stated here:

Theorem 3.1 (Decomposition Result). *There is a partition $V_1, \dots, V_p$ of V and weakly supermodular, symmetric, and normalized functions $f_i : 2^{V_i} \rightarrow \mathbb{Z}$ for all $i \in [p]$ (See Fig. 3) such that:*

- (a) x is a feasible solution to $\text{CRLP}(f, G) \iff$ for all i , x^{V_i} is a feasible solution to $\text{NDLP}(f_i, G[V_i])$ and, for all $e \in \delta_G(V_i)$, $x_e = 1$

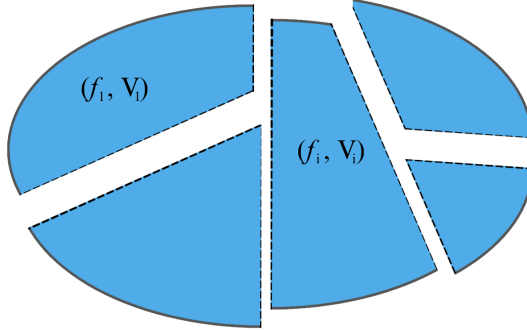


Figure 3: An illustration of the statement of Theorem 3.1. We partition the vertex-set V and define suitable cut requirement functions for each part such that solutions to standard $\text{ND}(f_i, G[V_i])$ on these subgraphs together with the boundaries of these parts yield a solution to $\text{CRND}(f, G)$.

- (b) x is an extreme-point solution to $\text{CRLP}(f, G) \iff$ for all i , x^{V_i} is an extreme-point solution to $\text{NDLP}(f_i, G[V_i])$ and, for all $e \in \delta_G(V_i)$, $x_e = 1$
- (c) H is a feasible solution to $\text{CRND}(f, G)$ on $G \iff$ for all i , $H[V_i]$ is a feasible solution to $\text{ND}(f_i, G[V_i])$, and $\delta_G(V_i) \subseteq H$

Before delving into the proof of Theorem 3.1, it is helpful to state the following corollary:

Corollary 3.2. *Let x be an extreme-point solution to $\text{CRLP}(f, G)$. Then there exists $e \in E$ such that $x_e \geq \frac{1}{2}$.*

Proof. By Theorem 3.1 (b), x induces a solution x^i to an f_i -ND instance on V_i for all i . By Theorem 2.10, each of these instances has an edge e with $x_e^i \geq \frac{1}{2}$. We see therefore that $x_e = x_e^i \geq \frac{1}{2}$. $\square$

Corollary 3.2 leads to a 2-approximation result: where $\text{CRLP}(f, G)$ can be solved efficiently, we may apply iterative rounding (noting that our cut requirement function remains in the same form after subtracting $|\delta_{Z_1}(S)|$) to find an integer solution; see Section 3.1.

The remainder of this section is devoted to the proof of Theorem 3.1. We describe now a method of decomposing an original instance to find the partition and functions described in Theorem 3.1. Recall that a small cut is a set $S \subseteq V$ where $f(S) > |\delta_G(S)|$. The key insight is that if there are no small cuts, then $\text{CRND}(f, G)$ becomes equivalent to $\text{ND}(f, G)$, as then $\min\{f(S), |\delta_G(S)|\} = f(S)$, the base cut requirement function. Our

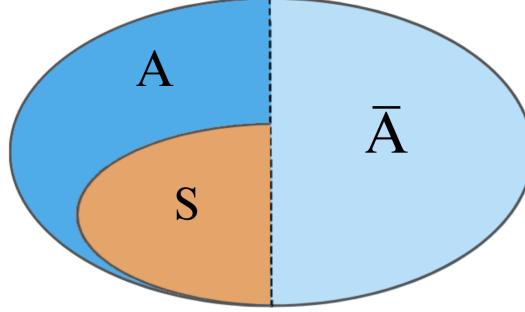


Figure 4: Illustrated motivation behind the definition of the restriction of f to A . For $S \subseteq A$, we wish to require in f_A at least as many edges between S and $A - S$ as we required for S by f minus the number of edges already taken on $\delta_Z(S)$. However, we also need to consider the requirement given by $S \cup \bar{A}$ (or equivalently, $A - S$) by f .

goal is to reduce to this case by splitting the graph along its small cuts, thereby eliminating such cuts. To this end, we introduce the following restriction operation.

Definition. Let $A \subset V$ and let $Z := \delta_G(A)$. We define the following function $f_A : 2^A \rightarrow \mathbb{Z}$, which we call the **restriction of f to A** , as

$$f_A(S) = \begin{cases} \max \begin{cases} f(S) - |\delta_Z(S)|, \\ f(S \cup \bar{A}) - |\delta_Z(S \cup \bar{A})| \end{cases} & \emptyset \neq S \subset A \\ 0 & \text{else} \end{cases}$$

Intuitively, $f_A(S)$ represents a lower bound on the remaining requirement of S supposing we take all edges of Z in our solution; see Figure 4. Note that by symmetry, we can replace $S \cup \bar{A}$ with $A - S$ in the above expression; we can therefore see the restriction also as a symmetrization of the requirement given by f on S .

Lemma 3.3. f_A is weakly supermodular, symmetric, and normalized.

Proof. Clearly, f_A is normalized by definition. Let $F = \delta_G(S)$ and define $f' : 2^A \rightarrow \mathbb{R}$ by $f'(T) = f(T) - |\delta_F(T)|$.

Note now that $f_A(S)$ is the projection of f' onto A with respect to $\mathcal{C} = \{\emptyset, \bar{A}\}$, and that $\mathcal{C}$ is closed under set intersection, union, difference, and complement. Therefore, by Lemma 2.7, f_A is weakly supermodular and symmetric.

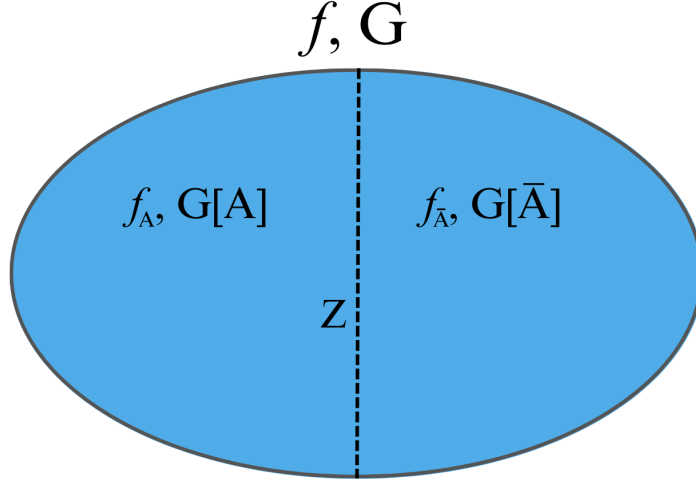


Figure 5: To decompose, we take a maximum-deficiency set A and define instances $\text{CRND}(f_A, G[A])$ and $\text{CRND}(f_{\bar{A}}, G[\bar{A}])$.

As a sidenote, we may also work with the $A - S$ definition, in which case we find that f_A is the symmetrization of f' (as in Lemma 2.3), and is therefore weakly supermodular and symmetric. $\square$

So, we split as follows: Take a small cut A of maximum deficiency, and define two further instances, $\text{CRND}(f_A, G[A])$ and $\text{CRND}(f_{\bar{A}}, G[\bar{A}])$. Lemma 3.3 shows that we can repeat this splitting operation on each instance, as the base CRFs (f_A and $f_{\bar{A}}$) of the new instances are themselves weakly supermodular. The lemma also gives that if we obtain an instance with no small cuts, we obtain an instance of standard f -Network Design with a weakly supermodular CRF.

In order to proceed with proving our theorem, we first establish a few helpful lemmas. Let $A \subseteq V$ be a small cut of maximum deficiency, and let $Z = \delta_G(A)$.

Lemma 3.4. *Recall that $E_{\text{small}} := \{e \in E : \exists S, S \text{ is a small cut}, e \in \delta_G(S)\}$.*

- (a) *If x is a feasible solution to $\text{CRLP}(f, G)$, then for any $e \in E_{\text{small}}$, $x_e = 1$.*
- (b) *If H is a feasible solution to $\text{CRND}(f, G)$, then $E_{\text{small}} \subseteq H$.*

Proof. Proving part (a) first, suppose there is $e \in E_{\text{small}}$ where $x_e < 1$. Letting S be a small cut such that $e \in \delta_G(S)$, we see that $x(\delta_G(S)) < |\delta_G(S)| < f(S)$ (since S is small),

meaning x is infeasible, giving a contradiction. Part (b) then follows by taking $x = \chi^H$, the indicator vector of H . $\square$

Lemma 3.5. *Let $\emptyset \neq S \subset A$, and $\delta_G(A) \subseteq F \subseteq E$. Then*

$$\text{def}_{f_A, F(A)}(S) = \max\{\text{def}_{f, F}(S), \text{def}_{f, F}(A - S)\}$$

Proof. We see this simply by plugging in:

$$\begin{aligned} \text{def}_{f_A, F(A)}(S) &= f_A(S) - |\delta_{F(A)}(S)| \\ &= \max\{f(S) - |\delta_Z(S)| - |\delta_{F(S)}(S)|, f(A - S) - |\delta_Z(A - S)| - |\delta_{F(S)}(A - S)|\} \end{aligned}$$

And noting that $|\delta_Z(S)| + |\delta_{F(S)}(S)| = |\delta_F(S)|$ and $|\delta_Z(A - S)| + |\delta_{F(S)}(A - S)| = |\delta_F(A - S)|$, we obtain our result. $\square$

We now show a key structural lemma, which shows that the solutions we obtain for a $\text{CRND}(f, G)$ instance correspond exactly with the solutions of the instances we define by our splitting operation.

Lemma 3.6. *Let A be a small cut in G of maximum deficiency.*

(a) x is feasible to $\text{CRLP}(f, G)$ if and only if:

- (i) For all $e \in \delta_G(A)$, $x_e = 1$
- (ii) x^A is feasible to $\text{CRLP}(f_A, G[A])$
- (iii) $x^{\bar{A}}$ is feasible to $\text{CRLP}(f_{\bar{A}}, G[\bar{A}])$

(b) H is feasible to $\text{CRND}(f, G)$ if and only if:

- (i) $\delta_G(A) \subseteq H$
- (ii) $H[A]$ is feasible to $\text{CRND}(f_A, G[A])$
- (iii) $H[A]$ is feasible to $\text{CRND}(f_{\bar{A}}, G[\bar{A}])$

Proof. We prove part (a) here. Note that part (b) follows by considering $x = \chi^H$. ($\implies$) direction: We prove each item separately.

- (i) Follows directly from Lemma 3.4(a), since A is a small cut.

- (ii) Suppose not; that is, x^A is not a feasible solution to $\text{CRLP}(f_A, G[A], \cdot)$. There must then be some $S \subseteq A$ where $x^A(\delta_{G[A]}(S)) < \min\{f_A(S), |\delta_{G[A]}(S)|\}$. We may assume that $\emptyset \neq S \subset A$, since $x^A(\cdot)$ must be nonnegative. Note that

$$\begin{aligned} x^A(\delta_{G[A]}(S)) &= x(\delta_G(S)) - x(\delta_Z(S)) \\ &= x(\delta_G(S)) - |\delta_Z(S)| \quad (\text{by item (i)}) \end{aligned}$$

And so we find that

$$\begin{aligned} x^A(\delta_{G[A]}(S)) &= x(\delta_G(S)) - |\delta_Z(S)| \\ &\geq \min\{f(S), |\delta_G(S)|\} - |\delta_Z(S)| \\ &\geq \min\{f(S) - |\delta_Z(S)|, |\delta_G(S)| - |\delta_Z(S)|\} \\ &= \min\{f(S) - |\delta_Z(S)|, |\delta_{G[A]}(S)|\} \end{aligned}$$

By symmetry of x , we also obtain:

$$x^A(\delta_{G[A]}(S)) = x^A(\delta_{G[A]}(A - S)) = x(\delta_G(A - S)) - |\delta_Z(A - S)|$$

And so similarly to the previous computation,

$$\begin{aligned} x^A(\delta_{G[A]}(S)) &= x(\delta_G(A - S)) - |\delta_Z(A - S)| \\ &\geq \min\{f(A - S), |\delta_G(A - S)|\} - |\delta_Z(A - S)| \\ &\geq \min\{f(A - S) - |\delta_Z(A - S)|, |\delta_G(A - S)| - |\delta_Z(A - S)|\} \\ &= \min\{f(A - S) - |\delta_Z(A - S)|, |\delta_{G[A]}(S)|\} \end{aligned}$$

combining these inequalities, since $f_A(S)$ equals either $f(S) - |\delta_Z(S)|$ or $f(A - S) - |\delta_Z(A - S)|$, we see that in either case,

$$x^A(\delta_{G[A]}(S)) \geq \min\{f_A(S), |\delta_{G[A]}(S)|\}$$

giving a contradiction.

- (iii) Shown with a symmetric argument as part (ii). □

($\Leftarrow$) direction: Assume items (i), (ii), and (iii) are all satisfied, and that x is not a feasible solution to $\text{CRLP}(f, G)$. There must be some set S such that $x(\delta_G(S)) < \min\{f(S), |\delta_G(S)|\}$.

We introduce a notion of *weighted* deficiency, along the lines of standard deficiency. For a cut-requirement function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ and weights $w : \mathbb{R}_{\geq 0}^{|E|}$, define weighted deficiency as $\text{wdef}_{f,w}(S) := f(S) - w(\delta_G(S))$.

We observe that weighted deficiency is symmetric, as well as weakly supermodular by Lemma 2.4, and similarly to Lemma 3.5, if $w_e = 1$ for all $e \in Z$, then $\text{wdef}_{f_A,w}(S) = \max\{\text{wdef}_{f,w}(S), \text{wdef}_{f,w}(A - S)\}$. Finally, observe that if $\text{wdef}_{f,x}(S) \leq 0$, then $x(\delta_G(S)) \geq f(S)$, and thus x satisfies the CRLP constraint at S .

It is intuitively easy to see if $S \subseteq A$ or $S \subseteq \bar{A}$, the constraint is indeed satisfied for S , as we simply take the LP constraint on S in the instance defined by A and add the remaining required edges from $Z := \delta_G(A)$. We show this for $S \subseteq A$: Observe

$$\begin{aligned} x(\delta_G(S)) &= x^A(\delta_{G[A]}(S)) + x(\delta_Z(S)) \\ &\geq \min\{f_A(S) + |\delta_Z(S)|, |\delta_{G[A]}(S)| + |\delta_Z(S)|\} \end{aligned}$$

By definition, we see that $f_A(S) \geq f(S) - |\delta_Z(S)|$, and thus

$$x(\delta_G(S)) \geq \min\{f(S), |\delta_G(S)|\}$$

Note specifically that if $S \subseteq A$ is an (f, G) -small cut, then this gives us that $x_e = 1$ for all $e \in \delta_G(S)$; if not, then $x(\delta_G(S)) \geq f(S)$. A symmetric argument holds if $S \subseteq \bar{A}$.

Now suppose that both $S \cap A$ and $S \cap \bar{A}$ are nonempty. By weak supermodularity and symmetry of wdef , we have

$$\text{wdef}_{f,x}(S) + \text{wdef}_{f,x}(A) \leq \begin{cases} \text{wdef}_{f,x}(S \cap A) + \text{wdef}_{f,x}(S \cup A) = \text{wdef}_{f,x}(S \cap A) + \text{wdef}_{f,x}(\bar{A} - S) \\ \text{or} \\ \text{wdef}_{f,x}(A - S) + \text{wdef}_{f,x}(S - A) \end{cases}$$

In either case, we have some $X \subseteq A$ and $Y \subseteq \bar{A}$ such that $\delta_G(S) \subseteq \delta_G(X) \cup \delta_G(Y) \cup Z$, and

$$\text{wdef}_{f,x}(S) + \text{wdef}_{f,x}(A) \leq \text{wdef}_{f,x}(X) + \text{wdef}_{f,x}(Y)$$

We also have $x(\delta_G(X)) \geq \min\{f(X), |\delta_G(X)|\}$ (and a similar inequality for Y) from our previous argument.

Note that by item (i), we have $\text{wdef}_{f,x}(A) = \text{def}_{f,G}(A)$ (since $x(\delta_G(S)) = |\delta_G(S)|$); further, we can show that $\text{wdef}_{f,x}(A) \geq \text{wdef}_{f,x}(X)$: if X is an (f, G) -small cut, then $x(\delta_G(X)) = |\delta_G(X)|$, and therefore $\text{wdef}_{f,x}(X) = \text{def}_{f,x}(X) \leq \text{def}_{f,G}(A) = \text{wdef}_{f,x}(A)$ since A is a maximum- (f, G) -deficiency set. We have argued above that if X is an (f, G) -large

cut, $x(\delta_G(S)) \geq f(S)$, and thus $\text{wdef}_{f,x}(X) \leq 0 \leq \text{def}_{f,G}(A) = \text{wdef}_{f,x}(A)$. A symmetric argument shows $\text{wdef}_{f,x}(A) \geq \text{wdef}_{f,x}(Y)$.

If both X and Y are small- (f, G) -cuts, then we have $x_e = 1$ for all $e \in \delta_G(X) \cup \delta_G(Y)$, and hence $x_e = 1$ for all $e \in \delta_G(S)$.

If neither of X and Y are small- (f, G) -cuts, then $\text{wdef}_{f,x}(X) \leq 0$, $\text{wdef}_{f,x}(Y) \leq 0$, so we have $\text{wdef}_{f,x}(S) \leq 0$.

So suppose, without loss of generality, that X is a small- (f, G) -cut, and Y is not a small- (f, G) -cut. Then $\text{wdef}_{f,x}(X) \leq \text{wdef}_{f,x}(A)$ as shown above, and so $\text{wdef}_{f,x}(S) \leq \text{wdef}_{f,x}(Y) \leq 0$. $\square$

We have thus described a method to decompose our graph into a partition, defining on each subset a weakly supermodular cut requirement function and a corresponding ND instance, and shown that we can at each step combine two solutions into a larger solution. We are now ready to prove Theorem 3.1.

Proof of Theorem 3.1. We begin with part (a). We induct on the number of vertices in our instance's graph.

For our base case, if we have no small cuts in our graph, then we simply let $p = 1$, $V_1 = V$ and $f_1 = f$. Otherwise, let A be a (f, G) -small cut of maximum deficiency, and define instances $\text{CRLP}(f_A, G[A])$ and $\text{CRLP}(f_{\bar{A}}, G[\bar{A}])$. By our induction hypothesis we obtain partitions of $G[A]$ and $G[\bar{A}]$, and by Lemma 3.6(a) we see that the partitions satisfy the conditions of the theorem.

For a part V_i contained in A , by the induction hypothesis, we obtain that $x_e^A = 1$ for all $e \in \delta_{G[A]}(V_i)$. Similarly, if $V_i \subseteq \bar{A}$, we have $x_e^{\bar{A}} = 1$ for all $e \in \delta_{G[\bar{A}]}(V_i)$. Combined with $x_e = 1$ for all $e \in \delta_G(A)$ this shows that $x_e = 1$ for all $e \in \delta_G(V_i)$.

To show part (c) we apply a similar argument considering $x = \chi^H$ and using Lemma 3.6(b).

We now show part (b); using part (a), we show that convex combinations of feasible solutions to $\text{CRLP}(f, G)$ correspond to convex combinations of solutions to the $\text{NDLP}(f_i, G)[V_i]$ instances.

Suppose x is an extreme-point solution to $\text{CRLP}(f, G)$, and for some $i \in [p]$, x^{V_i} is not an extreme-point solution to $\text{NDLP}(f_i, G)[V_i]$. Then there exists $\text{NDLP}(f_i, G)[V_i]$ solutions y, z and $\lambda \in [0, 1]$ where $x^{V_i} = \lambda y + (1 - \lambda)z$, and $y \neq x^{V_i}$, $z \neq x^{V_i}$.

Now consider x' and x'' such that $x'_e = x''_e = x_e$ for all $e \in E - E(V_i)$, and $x'_e = y_e$, $x''_e = z_e$ for $e \in E(V_i)$. By part (a) we can see that x' and x'' are feasible solutions to $\text{CRLP}(f, G)$. However, we see that $\lambda x' + (1 - \lambda)x'' = x$, and therefore x is not an extreme-point solution, giving a contradiction.

Now suppose for all i , x^{V_i} is an extreme-point solution to $\text{NDLP}(f_i, G[V_i])$ and, for all $e \in \delta_G(V_i)$, $x_e = 1$, and that x is not an extreme-point solution to $\text{CRLP}(f, G)$. Let x', x'' be feasible solutions such that $\lambda x' + (1 - \lambda)x'' = x$. Then, for all $i \in [p]$ and all $e \in \delta_G(V_i)$, we have $x'_e = x''_e = 1$. So, by part (a), it must be that $(x')^{V_i}$ and $(x'')^{V_i}$ are feasible solutions to $\text{CRLP}(f, G)$ for all $i \in [p]$. Since $x' \neq x''$, there is some $i \neq p$ such that $(x')^{V_i} \neq (x'')^{V_i}$. But then $x^{V_i} = \lambda x'^{V_i} + (1 - \lambda)x''^{V_i}$; thus x^{V_i} is not an extreme-point solution to $\text{NDLP}(f_i, G[V_i])$, giving a contradiction. $\square$

3.1 Algorithmic Results

We now use the decomposition described in the previous section to obtain an explicit 2-approximation algorithm.

Using Corollary 3.2, we may use an iterative rounding algorithm very much in the style of Algorithm 1. We assume that we have a separation oracle for $\text{CRLP}(f, G)$, meaning that we can find an extreme-point solution to $\text{CRLP}(f, G)$ in polynomial time. Similarly to Algorithm 1, as we iterate we will fix certain x_e variables to 1; noting that doing so keeps the LP in the same form, our separation oracle can efficiently find an optimal solution in any residual LP. As a sidenote, observe that small cuts are preserved across iterations;

Algorithm 2: Cut-Relative Network Design 2-Approximation

Input : Graph $G = (V, E)$, costs $\{c_e\}_{e \in E}$, function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$
Output: A subgraph $H \subseteq E$ where $|\delta_H(S)| \geq \min\{f(S), |\delta_G(S)|\}$ for all $S \subseteq V$
 Assume we have a separation oracle for $\text{CRLP}(f, G)$

Initialize $H \leftarrow \emptyset$, $G' = (V, E') \leftarrow G$, $f' \leftarrow f$;

while H is not a feasible solution (detected via separation oracle) **do**

 Compute an extreme-point optimal solution $\hat{x}$ to $\text{CRLP}(f', G')$;

 Let $Z_1 = \{e \in E' : \hat{x}_e \geq 1/2\}$, update $H \leftarrow H \cup Z_1$, $E' \leftarrow E' - Z_1$, and

$f'(S) = f'(S) - |\delta_{Z_1}(S)|$ for all $S \subseteq V$;

end

Return H

that is to say, $S \subseteq V$ is an (f, G) -small cut iff it is an (f', G') -small cut in some iteration. This is because in every iteration we change $|\delta_{G'}S|$ and $f'(S)$ by the same amount, thereby keeping deficiency the same.

Theorem 3.7. *Let OPT be the optimal solution value of $\text{CRLP}(f, G)$. Then Algorithm 2 terminates, and returns a feasible $\text{CRLP}(f, G)$ solution H such that $c(H) \leq 2 \cdot OPT$.*

Proof. By the Corollary to Theorem 3.1, in each iteration we add at least one edge to H ; therefore, the algorithm must terminate in at most $|E|$ iterations.

To show the approximation factor, we induct on the number of iterations until termination. As a base case, if we terminate in 1 iteration, then at the first iteration the algorithm finds an integer extreme-point optimal solution to $\text{CRLP}(f, G)$, and therefore returns H where $c(H) = OPT$.

Now suppose at the end of a given iteration we compute $\hat{x}$, Z_1 , E' and f' , and let $H = Z_1 \cup H'$, where H' is the solution found by subsequent iterations for the instance $\text{CRND}(f', G')$. Let OPT' be the optimal solution value for $\text{CRLP}(f', G')$; then by our induction hypothesis we have $c(H') \leq 2 \cdot OPT'$. Since $(\hat{x}_e)_{e \in E'}$ is feasible to $\text{CRLP}(f', G')$, we have $OPT' \leq \sum_{e \in E'} c_e \hat{x}_e$. By definition we also have $c(Z_1) \leq 2 \sum_{e \in Z_1} c_e \hat{x}_e$. So,

$$c(F) = c(H') + c(Z_1) \leq 2 \sum_{e \in E} c_e \hat{x}_e = 2 \cdot OPT$$

□

By Lemma 2.9, there exists a separation oracle for CRSND . Therefore, we have shown Theorem 1.1, giving a 2-approximation algorithm for Cut-Relative Survivable Network Design.

Chapter 4

Properties of Decomposition

In this chapter, we show some properties of the decomposition described in Chapter 3. We show that the decomposition may be described in a binary tree structure, and establish some useful properties of this decomposition tree, which will lead to further interesting consequences.

4.1 Decomposition Tree

Theorem 3.1 does not in itself describe a specific decomposition, only that one exists which satisfies the given properties. For the purposes of this chapter, we will focus specifically on the decomposition given by the proof of Theorem 3.1, which we will call the *canonical decomposition*.

Recall that we obtain this decomposition by taking a maximum deficiency set $A \subseteq V$, creating the restrictions $f_A : 2^A \rightarrow \mathbb{Z}$ and $f_{\bar{A}} : 2^{\bar{A}} \rightarrow \mathbb{Z}$, and recursing on the instances $\text{CRND}(f_A, G[A])$ and $\text{CRND}(f_{\bar{A}}, G[\bar{A}])$.

We can describe this decomposition in a tree structure (see Fig. 6). Each node of the tree represents the instance $\text{CRND}(h, G[S])$, where $S \subseteq V$ is the node set on which the instance is defined and h is its base cut requirement function; we will label this node (h, S) , and will often simply refer to it as "node (h, S) ". The root of the tree is labelled (f, V) . For each node (h, S) , if there are no $(h, G[S])$ -small cuts, then (h, S) is a leaf node; otherwise, we find $A \subseteq S$ with maximum $(h, G[S])$ -deficiency, and we have two child nodes of (h, S) with labels (h_1, A) and $(h_2, S - A)$ (where h_1, h_2 are the restrictions of h to $A, S - A$ respectively). We call the described tree the *canonical decomposition tree*.

In keeping with Theorem 3.1, we refer to the leaves of the tree as (f_i, V_i) for $i \in [p]$.

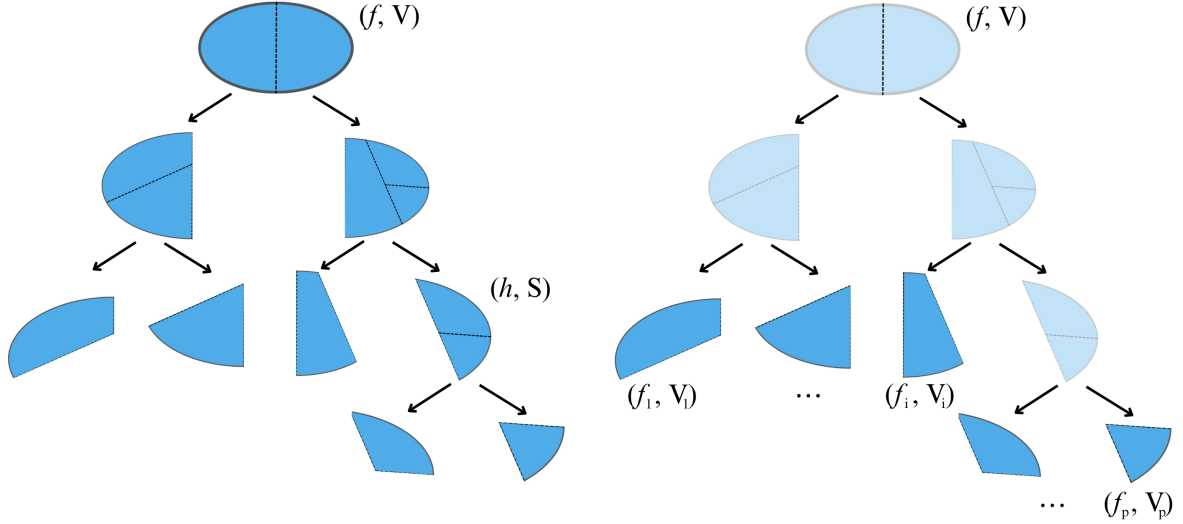


Figure 6: Left: Illustration of the Decomposition Tree. Dotted lines denote small cuts in each node, which are split on to create child nodes. Right: The leaves of the decomposition tree, corresponding to the partition given by Theorem 3.1.

These represent the base instances of f -Network Design (specifically, $\text{ND}(f_i, G[V_i])$) that we can solve to gain a solution to our original $\text{CRND}(f, G)$ problem. We refer to the V_i sets, making up the partition of Theorem 3.1, as "leaf sets".

The remainder of this chapter is dedicated to establishing useful properties of the canonical decomposition tree. In Section 4.2 we show that the canonical decomposition tree (and therefore the canonical decomposition itself) may be computed efficiently, assuming suitable algorithmic access to f . This may be of interest in finding alternative, possibly more efficient approximation algorithms. In Section 4.4 we show that the leaves of the decomposition tree, corresponding to the f -Network Design instances at the base of our recursion, may be described in a self-contained, non-recursive manner.

4.2 Key Structural Property of the Canonical Decomposition Tree

We give a key structural lemma that describes properties of the canonical decomposition, particularly in regard to deficiency with respect to each node. Part (a) unpacks the def-

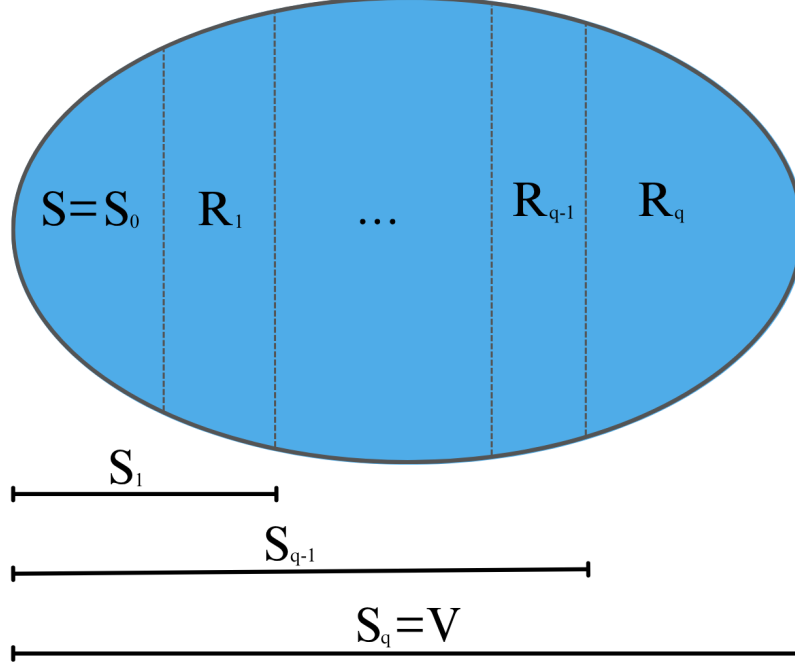


Figure 7: Illustration of the sets S_i and R_i corresponding to one path of the decomposition tree. Sets R_I are defined as unions of R_i sets.

inition of $(h, G[S])$ -deficiency, defining it in terms of the original base cut requirement function f . Part (b) uses this to prove a monotonicity property about deficiency. Finally, part (c) exploits this to argue that the deficiency of certain types of subsets of S can be defined in terms of the function f and the original graph G .

Lemma 4.1. *Let (h, S) be a node of the canonical decomposition tree. Denote the nodes on the path from (h, S) to the tree root as $(h_0, S_0) = (h, S), (h_1, S_1), \dots, (h_q, S_q) = (f, V)$. We call q the depth of (h, S) . For $I \subseteq [q]$, let $R_I := \bigcup_{i \in I} (S_i - S_{i-1})$; note that $R_\emptyset = \emptyset$. For $i \in [q]$ we will denote $R_{\{i\}}$ as R_i , so $R_i = S_i - S_{i-1}$ (See Figure 7). The following hold:*

- (a) *For any $\emptyset \neq T \subsetneq S$, we have $\text{def}_{h, G[S]}(T) = \max \{ \text{def}_{f, G}(T \cup R_I) : I \subseteq [q] \}$.*
- (b) *For any $i, j \in \{0, \dots, q\}$ with $i \leq j$, and any $T \subseteq S_i$, we have $\text{def}_{h_j, G[S_j]}(T) \leq \text{def}_{h_i, G[S_i]}(T)$.*

(c) Let $s, t \in S$. We have

$$\max\{\text{def}_{h,G[S]}(T) : T \subseteq S \text{ is an } s\text{-}t \text{ cut}\} = \max\{\text{def}_{f,G}(T) : T \subseteq V \text{ is an } s\text{-}t \text{ cut}\}$$

Moreover, if an s - t cut $T \subseteq V$ attains the maximum on the right-hand side, then $T \cap S$ (or $S - T$, by symmetry) attains the maximum on the left.

Proof. (a) We induct on q . The base case $q = 0$ is trivially true. Suppose the statement holds for depth at most $q - 1$. Recall that h is the restriction of h_1 to S . By Lemma 3.5 we have

$$\text{def}_{h,G[S]}(T) = \max\{\text{def}_{h_1,G[S_1]}(T), \text{def}_{h_1,G[S_1]}(T \cup R_1)\}$$

By our induction hypothesis, we have

$$\begin{aligned} \text{def}_{h_1,G[S_1]}(T) &= \max\{\text{def}_{f,G}(T \cup R_I) : I \subseteq \{2, \dots, q\}\} \\ \text{def}_{h_1,G[S_1]}(T \cup R_1) &= \max\{\text{def}_{f,G}(T \cup R_1 \cup R_I) : I \subseteq \{2, \dots, q\}\} \end{aligned}$$

Observing that $R_1 \cup R_I = R_{I \cup \{1\}}$, we find that $\text{def}_{h,G[S]}(T)$ attains the maximum over all $I \subseteq [q]$, obtaining our induction step.

(b) This follows from part (a); the sets $\{T \cup R_I : I \subseteq \{i + 1, \dots, \ell\}\}$ considered in the computation (as in part (a)) of $\text{def}_{h_i,G[S_i]}(T)$ is a superset of the sets $\{T \cup R_I : I \subseteq \{j + 1, \dots, \ell\}\}$ considered for $\text{def}_{h_j,G[S_j]}(T)$. Thus we must have that $\text{def}_{h_j,G[S_j]}(T) \leq \text{def}_{h_i,G[S_i]}(T)$.

(c) Let $\text{LHS} = \max\{\text{def}_{h,G[S]}(T) : T \subseteq S \text{ is an } s\text{-}t \text{ cut}\}$ and $\text{RHS} = \max\{\text{def}_{f,G}(T) : T \subseteq V \text{ is an } s\text{-}t \text{ cut}\}$. By part (a), for any s - t cut $T \subseteq S$, we can find some s - t cut $T_1 \subseteq V$ such that $\text{def}_{h,G[S]}(T) = \text{def}_{f,G}(T_1)$. So $\text{LHS} \leq \text{RHS}$.

We claim that $\text{def}_{f,G}(T') \leq \text{def}_{h,G[S]}(T' \cap S)$ for any s - t cut $T' \subseteq V$. Since $T' \cap S$ is an s - t cut contained in S , this implies that $\text{RHS} \leq \text{LHS}$.

Let $a \in \{0, \dots, q\}$ be the minimum index such that $T' \subseteq S_a$. For all $i \in \{0, \dots, a\}$, let $X_i := T' \cap S_i$ (so $X_a = T'$ and $X_0 = T' \cap S$). We wish to show

$$\text{def}_{h_i,G[S_i]}(X_i) \leq \text{def}_{h_{i-1},G[S_{i-1}]}(X_{i-1}) \quad \forall i \in [a] \quad (\star)$$

This implies that $\text{def}_{h_a,G[S_a]}(X_a) \leq \text{def}_{h,G[S]}(X_0)$, or in other words, we have $\text{def}_{h_a,G[S_a]}(T') \leq \text{def}_{h,G[S]}(T' \cap S)$, completing the proof.

We now prove inequality $\star$. Let $i \in [a]$. By weak supermodularity of deficiency, we have

$$\text{def}_{h_i, G[S_i]}(X_i) + \text{def}_{h_i, G[S_i]}(S_{i-1}) \leq \text{def}_{h_i, G[S_i]}(A_{i-1}) + \text{def}_{h_i, G[S_i]}(B_{i-1}) \quad (4.1)$$

Where either $A_{i-1} = X_i \cap S_{i-1}$, $B_{i-1} = X_i \cup S_{i-1}$ or $A_{i-1} = S_{i-1} - X_i$, $B_{i-1} = X_i - S_{i-1}$.

Recalling that we split along maximum deficiency sets in the canonical decomposition, we have $\text{def}_{h_i, G[S_i]}(S_{i-1}) \geq \text{def}_{h_i, G[S_i]}(Y)$ for all $Y \subseteq S_i$. Observe that $B_{i-1} \subseteq S_i$, and thus we have $\text{def}_{h_i, G[S_i]}(S_{i-1}) \geq \text{def}_{h_i, G[S_i]}(B_{i-1})$, and so Inequality 4.1 implies that

$$\text{def}_{h_i, G[S_i]}(X_i) \leq \text{def}_{h_i, G[S_i]}(A_{i-1})$$

Observe that $A_{i-1} \subseteq S_{i-1}$, so by part (b), we also have $\text{def}_{h_i, G[S_i]}(A_{i-1}) \leq \text{def}_{h_{i-1}, G[S_{i-1}]}(A_{i-1})$.

Recall that either $A_{i-1} = X_i \cap S_{i-1} = X_{i-1}$ or $A_{i-1} = S_{i-1} - X_i = S_{i-1} - X_{i-1}$. By symmetry of deficiency, we therefore obtain

$$\text{def}_{h_{i-1}, G[S_{i-1}]}(A_{i-1}) = \text{def}_{h_{i-1}, G[S_{i-1}]}(X_{i-1})$$

And thus, combining the last two inequalities completes our proof. $\square$

4.3 Efficient Computation of the Canonical Decomposition

Making use of Lemma 4.1(c), we show that the canonical decomposition can be efficiently computed, assuming the ability to compute maximum-deficiency sets in (f, G) . Specifically, we consider the following problems:

Definition. In the **maximum-deficiency s - t cut** problem, we are given $s, t \in V$ and want to return an s - t cut $A \subseteq V$ that maximizes $\text{def}_{f, G}(S)$ over all s - t cuts $S \subseteq V$.

In the **weighted maximum-deficiency s - t cut** problem, we are additionally given weights $w \in \mathbb{R}_{\geq 0}^E$ and want to return an s - t cut $A \subseteq V$ that maximizes $\text{wdef}_{f, G}(S) := f(S) - w(\delta_G(S))$ over all s - t cuts $S \subseteq V$.

Theorem 4.2. *Given a polynomial-time algorithm for the max-deficiency s - t cut problem, one can compute the canonical decomposition tree in polynomial time.*

Proof. We need only show that given a node (h, S) of the current tree, we can find a maximum deficiency subset of S . Clearly, if we can efficiently compute $\max\{\text{def}_{h,G[S]}(T) : T \subseteq S \text{ is an } s\text{-}t \text{ cut}\}$ for all $t \in S, t \neq s$ and simply take the maximum of these cuts, we are done. By Lemma 4.1(c) we can find these cuts by finding maximum deficiency $s\text{-}t$ with respect to f, G with our algorithm, and taking intersections with S . Thus, we can extend a node of the tree with $|S|$ runs of our algorithm. $\square$

Corollary 4.3. *Given a polynomial-time algorithm for the weighted max-deficiency $s\text{-}t$ cut problem, one can give a separation oracle for $\text{NDLP}(f_i, G[S_i])$ for any leaf (f_i, S_i) of the canonical decomposition tree.*

Proof. Recall that given weights $w : \mathbb{R}_{\geq 0}^{|E|}$, we define weighted deficiency as $\text{wdef}_{f,w}(S) := f(S) - w(\delta_G(S))$. We can straightforwardly show an analogous property to Lemma 4.1(c) for weighted deficiency; that is, we have $\max\{\text{wdef}_{f_i,G[S]}(T) : T \subseteq S \text{ is an } s\text{-}t \text{ cut}\} = \max\{\text{wdef}_{f,G}(T) : T \subseteq V \text{ is an } s\text{-}t \text{ cut}\}$, and if T attains the maximum on the right, then $T \cap S$ attains the maximum on the left. Therefore, given a solution x we can compute $\max_{T \subseteq S}(f_i(T) - x(\delta_{G[S]}(T)))$ by performing $|S|$ weighted max- $s\text{-}t$ -deficient-cut computations with respect to (f, G) , using x_e as weights. If this value is positive, then we find a violated constraint; otherwise, we find that x is feasible. $\square$

Corollary 4.4. *For any instance of CRSND, we can compute the canonical decomposition tree in polynomial time. Furthermore, for any leaf (f_i, S_i) of the tree, we have a polynomial time separation oracle for $\text{NDLP}(f_i, G[S_i])$*

Proof. It suffices to show that finding weighted max-deficiency cuts can be done efficiently; we do this by showing that we can equivalently solve a min-cut problem. Specifically, to find $\max\{(f^{SND}(T) - x(\delta_{G[S]}(T)); T \subseteq V \text{ is an } s\text{-}t \text{ cut}\}$, we consider each $i \in [k]$, and find T_i that minimizes $x(\delta_G(T))$ over all sets $T \subseteq V$ such that $|T \cap \{s_i, t_i\}| = |T \cap \{s, t\}| = 1$, by solving a min-cut problem. We then return $\max_{i \in [k]}(k_i - x(\delta_G(T_i)))$. $\square$

4.4 A Self-Contained Description of the Canonical Decomposition

In this section, we show a self-contained definition of the f -ND instances at the leaves of the decomposition tree as described in the previous section, which have so far been described purely recursively.

Recall that $E_{\text{small}} := \{e \in E : \exists S, S \text{ is a small cut}, e \in \delta_G(S)\}$ is the set of all edges that

are in some small cut. Let $C_1, \dots, C_q$ be the node sets of the components of $G - E_{\text{small}}$. Note that a component C_i need not be a (f, G) -small cut, even though $\delta_G(C_i) \subseteq E_{\text{small}}$. For example, in a problem with a single demand for s, t , then every (f, G) -small cut must be an s - t cut; so, unless there are only two small cuts, there will be C_i sets that are not small cuts.

We call components C_i, C_j *non-separable* if for every (f, G) -small cut A , we have that $C_i \subseteq A$ iff $C_j \subseteq A$. Clearly, non-separability is an equivalence relation.

We show that the $V_1, \dots, V_k$ leaf-set partition given by the leaves of the decomposition tree is precisely the partition formed by the equivalence classes of the non-separability relation. This yields a simple and succinct description of the partition $V_1, \dots, V_p$ in Theorem 3.1. Later, we show that we can also succinctly specify the f_i CRFs.

Lemma 4.5. *The leaf sets $V_1, \dots, V_p$ of the canonical decomposition tree are the equivalence classes of the non-separability relation. That is, for each V_i , there exists a set $\text{EQ}_i \subseteq [q]$ of indices such that:*

- (i) $V_i = \bigcup_{j \in \text{EQ}_i} C_j$
- (ii) *for any two indices $j \in \text{EQ}_i, j' \in [q]$, components C_j and $C_{j'}$ are non-separable iff $j' \in \text{EQ}_i$.*

In particular, this means there are no edges of E_{small} that are internal to a leaf set.

Proof. We show first that the $C_1, \dots, C_q$ partition is a refinement of the $V_1, \dots, V_k$ partition, which is equivalent to item (i). Consider any node $(h, S) \neq (f, V)$ of the canonical decomposition tree, and let (h', S') be the parent of (h, S) . We claim that $\delta_{G[S']}(S) \subseteq E_{\text{small}}$. Applying this to every node of the canonical decomposition tree yields that $\delta_G(V_i) \subseteq E_{\text{small}}$ for every leaf-set V_i . To see the claim, applying Lemma 4.1(a) to (h', S') taking $T = S$, since $\text{def}_{h', G[S']}(S) > 0$, we obtain that there is some small- (f, G) -cut $A \subseteq V$ such that $S = A \cap S'$. Thus, $\delta_{G[S']}(S) \subseteq \delta_G(A) \subseteq E_{\text{small}}$.

To show item (ii), consider some leaf set V_i and any two components $C_j, C_{j'}$ where $C_j \subseteq V_i$. We must show that C_j and $C_{j'}$ are non-separable iff $C_{j'} \subseteq V_i$.

Suppose first that $C_{j'} \not\subseteq V_i$, and let $C_{j'} \subseteq V_{i'}$ for some $i' \neq i$. Let (h, S) be the least common ancestor of (f_i, V_i) and $(f_{i'}, V_{i'})$ (that is, the common ancestor of these nodes of maximum depth). Let $(h_1, A), (h_2, S - A)$ be the children of (h, S) . Clearly, A separates V_i and $V_{i'}$. By our splitting operation we have $0 < \text{def}_{h, G[S]}(A) = \max\{\text{def}_{h, G[S]}(X) : X \subseteq S\}$; so for any $s \in A, t \in S - A$, by Lemma 4.1(c) we have that $A = T \cap S$, where $T \subseteq V$ is such that $\text{def}_{f, G}(T) = \text{def}_{h, G[S]}(A)$. Thus, T is a (f, G) -small cut that separates V_i and $V_{i'}$, and therefore also separates C_j and $C_{j'}$.

Now suppose that $C_{j'} \subseteq V_i$ and suppose there exists a (f, G) -small cut T that separates C_j and $C_{j'}$. Then for any $s \in C_j$, $t \in C_{j'}$, by Lemma 4.1(c) there must be a $(f_i, G[V_i])$ -small cut $T' \subseteq V_i$ that separates s and t ; since (f_i, V_i) is a leaf of the canonical decomposition tree, this yields a contradiction. $\square$

Lemma 4.6. *Let $\{(f_i, V_i)\}_{i \in [p]}$ be the leaves of the canonical decomposition tree. For each $i \in [p]$ let $\mathcal{C}_i := \{\bigcup_{j \in I} V_j : I \subseteq [p], i \notin I\}$. Define*

$$\gamma_i(S) = \max \{f(S \cup X) - |\delta_{E_{\text{small}}}(S \cup X)| : X \in \mathcal{C}_i\}$$

Then γ_i coincides with f_i , the function obtained from the canonical decomposition.

Proof. As before, denote the nodes on the path from (f_i, V_i) to the tree root as $(h_0, S_0) = (f_i, V_i), (h_1, S_1), \dots, (h_q, S_q) = (f, V)$. As well recall for $I \subseteq [q]$, $R_I := \bigcup_{i \in I} (S_i - S_{i-1})$; and for $i \in [q]$, $R_{\{i\}}$ as R_i , so $R_i = S_i - S_{i-1}$. Consider any $\emptyset \neq T \subset V_i$.

We first show $\gamma_i(T) \geq f_i(T)$. By Lemma 4.1(a), we can see that f_i is defined similarly to γ_i , but by further restricting the sets $X \in \mathcal{C}_i$ over which the maximum is taken (just sets of the form R_I for $I \subseteq [q]$). Formally, we show equivalently that $\text{def}_{\gamma_i, G[V_i]}(T) \geq \text{def}_{f_i, G[V_i]}(T)$.

Observe that for $j \in [q]$, we have that R_j is a union of some leaf sets other than V_i , and therefore $R_j \in \mathcal{C}_i$, and therefore $R_I \in \mathcal{C}_i$ for all $I \subseteq [q]$. Thus,

$$\text{def}_{f_i, G[V_i]}(T) = \max \{\text{def}_{f, G}(T \cup R_I) : I \subseteq [q]\} \leq \max \{\text{def}_{f, G}(T \cup X) : X \in \mathcal{C}_i\} = \text{def}_{\gamma_i, G[V_i]}(T)$$

To show $\gamma_i(T) \leq f_i(T)$, let $X \in \mathcal{C}_i$ such that $\text{def}_{\gamma_i, G[V_i]}(T) = \text{def}_{f, G}(T \cup X)$, and let $T' = T \cup X$. In the proof of Lemma 4.1(c), we showed $\text{def}_{f, G}(T') \leq \text{def}_{f_i, G[V_i]}(T' \cap V_i)$; since $T' \cap V_i = T$, we obtain $\text{def}_{\gamma_i, G[V_i]}(T) \leq \text{def}_{f_i, G[V_i]}(T)$ $\square$

While we have these self-contained descriptions, it is worth noting that we have not yet found a proof that taking solutions of these $\text{ND}(\gamma_i, G[V_i])$ instances found at the leaves of the decomposition tree, along with the edges on the boundaries $\delta_G(V_i)$, yields a solution to the original $\text{CRND}(f, G)$ problem, that does not use properties of the canonical decomposition (namely, Lemma 4.1). That is to say, arguing directly that these sets and functions satisfy Theorem 3.1 is much harder without using the recursive view of the decomposition.

Chapter 5

Hardness Results

In this chapter, we show that both Cut-Relative and Path-Relative SND are *APX*-hard. We do this by showing a reduction from the *APX*-hard k -edge connected spanning subgraph (k -ECSS) problem to single-demand Cut- and Path-Relative SND.

The class *APX* comprises problems that have polynomial-time constant-factor approximation algorithms. Similarly to how *NP*-hardness is defined, roughly, a problem is *APX*-hard if there is a polynomial-time reduction from any problem in *APX* to it, with the added requirement that the reduction preserve the property that a problem admits a polynomial-time approximation scheme, a type of approximation algorithm that can give solutions with value arbitrarily close to the optimal value. [WS11]

Consider a k -ECSS instance with graph $G = (V, E)$ and costs $\{c_e\}_{e \in E}$. We may assume that G is k -edge connected, as otherwise the problem is infeasible. Define new nodes s, t and new edges sv and vt for all $v \in V$, all with cost 0 (see Fig. 8). Denote this new graph $G' = (V', E')$. We define a base SND instance on G' with the single demand $(s, t, k + n)$, where $n = |V|$; so, our CRF f is defined as $f(S) = k + n$ if $S \subseteq V'$ is an s - t cut, and $f(S) = 0$ otherwise.

We will show that using f as the base cut requirement function, Cut- and Path-Relative SND are in fact equivalent, and that solving these instances on G' are equivalent to solving k -ECSS on G .

Lemma 5.1. *For the above instance, CRSND and PRSND are equivalent.*

Proof. We show in Section 6.1 (Lemma 6.1) that Path-Relative SND is equivalent to imposing the CRSND constraints only on fully connected sets (also defined in Section 6.1; we call a set $S \subseteq V$ fully connected if both $G[S]$ and $G[V' - S]$ are connected graphs, where $V' \subseteq V$ is the vertex set of the component containing S).

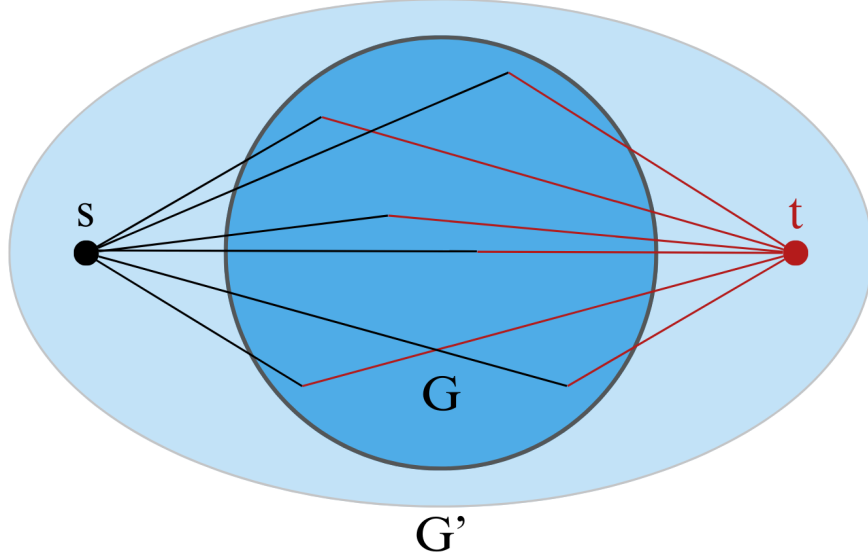


Figure 8: Relative SND instance that models k -ECSS.

We show that any s - t cut S is fully connected, and therefore the constraints imposed by CRSND and PRSND are the same. Without loss of generality, suppose $s \in S$; then for any $u, v \in S$ the path $\{us, sv\}$ exists in $G[S]$, and for any $u', v' \in V - S$, the path $\{u't, tv'\}$ exists in $G[V - S]$. Therefore, S is fully connected. $\square$

Theorem 5.2. $H \subseteq E'$ is a feasible solution to the above Cut-/Path-Relative SND iff $\delta_{G'}(\{s\}) \cup \delta_{G'}(\{t\}) \subseteq H$ and $H(V)$ is feasible to k -ECSS on G .

Proof. First, observe that the only small cuts in G' are $\{s\}, \{t\}$, and their respective complements; for any other s - t cut $S' \subseteq V'$, taking $S = S' \cap V$, we have $|\delta_{G'}(S')| = |\delta_G(S)| + |S| + |V - S| \geq k + n$ (since $\emptyset \neq S \subset V$ and G is k -edge-connected).

($\Leftarrow$) direction: H as described covers both small cuts. For any other s - t cut $S' \subseteq V'$, taking $S = S' \cap V$, we have $|\delta_H(S')| = |\delta_{H(V)}(S)| + |S| + |V - S| \geq k + n$, since $H(V)$ is a feasible k -ECSS solution for G .

($\Rightarrow$) direction: Suppose H is feasible to CRSND. By Lemma 3.4, we must have $E_{\text{small}} = \delta_{G'}(\{s\}) \cup \delta_{G'}(\{t\}) \subseteq H$. Again, let $S' \subseteq V'$ be an s - t cut, and assume S' is not one of the small cuts. Let $S = S' \cap V$. Now, we have $|\delta_H(S')| \geq k + n$, and $|\delta_H(S') \cap \delta_{G'}(\{s\})| + |\delta_H(S') \cap \delta_{G'}(\{t\})| = n$, so $|\delta_H(S')| = |\delta_{H(V)}(S)| + |S| + |V - S| \geq k + n \implies |\delta_{H(V)}(S)| \geq k$. So, $H(V)$ is feasible to k -ECSS. $\square$

Chapter 6

Path-Relative SND

In this chapter, we discuss Path-Relative SND, describing some useful properties of the problem and outlining possible future work on finding improvements to the best known approximation algorithms for PRSND.

6.1 Cut-Based Formulation

It is desirable to find a cut-based formulation for Path-Relative SND, such that we may see how methods for Cut-Relative SND may be applied to PRSND. We give the following cut-based formulation, first described in [DKKN23], where we apply the CRSND constraints to specific subsets of V .

Recall that an instance of PRSND comprises a graph $G = (V, E)$, costs $c : E \rightarrow \mathbb{R}_{\geq 0}$, and demands $\{(s_i, t_i, k_i)\}_{i \in [\ell]}$. We may assume that for all i , s_i and t_i are connected in G , as if not, PRSND does not impose any connectivity requirement between s_i and t_i . We call a set $S \subseteq V$ **fully connected** if both $G[S]$ and $G[V' - S]$ are connected graphs, where $V' \subseteq V$ is the vertex set of the component containing S .

Lemma 6.1. *H is feasible for Path-Relative SND iff $|\delta_H(S)| \geq \min\{f^{SND}(S), |\delta_G(S)|\}$ for every fully connected set $S \subseteq V$.*

Proof. Suppose H is feasible for PRSND, and that there is some fully connected $S \subseteq V$ such that $|\delta_H(S)| < \min\{f^{SND}(S), |\delta_G(S)|\}$. Let $V' \subseteq V$ be the vertex set of the component containing S . Let $i \in [\ell]$ such that S is an s_i - t_i cut and $f^{SND}(S) = k_i$; without loss of generality suppose $s_i \in S$. Let $F = \delta_H(S)$; then $|F| < k_i$ and $\delta_{G-F}(S) \neq \emptyset$. Clearly there is no s_i - t_i path in $H - F$.

Let $e = uv \in \delta_{G-F}(S)$, where $u \in S$. Since S is fully connected, there are paths from s_i to u and v to t_i in $G[V'] - F$; so combining these paths and the edge uv gives an s_i - t_i path in $G - F$. Since there is no s_i - t_i path in $H - F$, H is not feasible for PRSND, giving a contradiction.

Conversely, suppose that $|\delta_H(S)| \geq \min\{f^{SND}(S), |\delta_G(S)|\}$ for all fully connected $S \subseteq V$. Consider any $i \in [\ell]$ and any $F \subseteq E$ with $|F| < k_i$ such that $G - F$ has an s_i - t_i path. Therefore, for any s_i - t_i cut $S \subseteq V$, $|\delta_G(S)| > |\delta_F(S)|$.

Suppose that $H - F$ does not have an s_i - t_i path. Let $V' \subseteq V$ be the vertex set of the component containing s_i ; note that $t_i \in V'$ as well. Let $A \subseteq V'$ be the set of vertices reachable from s_i in $H - F$, and let $B \subseteq V' - A$ be the set of vertices reachable from t_i in $G[V'] - \delta_G(A)$. Note that $\delta_G(B) \subseteq \delta_G(A)$ by construction. So every edge in $\delta_H(B)$ lies in $\delta_G(A) \cap H = \delta_H(A) \subseteq F$, and hence, $\delta_H(B) \subseteq \delta_F(B)$. Therefore, $|\delta_H(B)| \leq |\delta_F(B)| < \min\{k_i, |\delta_G(B)|\} \leq \min\{f^{SND}(S), |\delta_G(S)|\}$. Hence, if we can show that B is fully connected, we will obtain a contradiction.

By construction, $G[A]$ and $G[B]$ are connected, and further there are no edges between B and $V' - A - B$. If $G[V' - B]$ is not connected, then there is some $T \subseteq V' - A - B$ such that $\delta_{G[V' - B]}(T) = \emptyset$. However, we also have $\delta_{G[V' - A]}(T) = \emptyset$ from earlier, which means that $\delta_G(T) = \emptyset$, contradicting that V' is connected. $\square$

6.1.1 LP Relaxation

From the cut-based formulation, we obtain the following LP relaxation.

$$\begin{aligned}
&\text{minimize:} && \sum_{e \in E} c_e x_e && (PRLP(f, G)) \\
&\text{subject to:} && x(\delta_G(S)) \geq \min\{f(S), |\delta_G(S)|\}, && S \subseteq V : S \text{ is fully connected} \\
&&& 0 \leq x_e \leq 1, && e \in E
\end{aligned}$$

6.2 Connected f -ND

One can infer from the above that there are two sources of complication in dealing with PRSND: (1) the cut-constraints are of the form $\min\{f(S), |\delta_G(S)|\}$ (as in CRSND), and (2) these constraints are only imposed for fully-connected sets. CRSND can be viewed as dealing with complication (1), but not complication (2), and as we have seen admits a 2-approximation. One might ask if solving a problem that retains complication (2) but not complication (1), combined with our insights from CRSND, might shed light on solving

both: i.e., what if consider the simpler cut constraints $|\delta_H(S)| \geq f(S)$, as in $\text{ND}(f, G)$, but impose them only for fully-connected sets. To this end, we define the following problem and corresponding LP relaxation:

Connected f -Network Design ($\text{ConND}(f, G)$): find a minimum-cost subgraph H such that $|\delta_H(S)| \geq f(S)$ for all *fully connected* $S \subseteq V$.

$$\begin{aligned} \text{minimize:} \quad & \sum_{e \in E} c_e x_e && (\text{ConLP}(f, G)) \\ \text{subject to:} \quad & x(\delta_G(S)) \geq f(S), && S \subseteq V : S \text{ is fully connected} \\ & 0 \leq x_e \leq 1, && e \in E \end{aligned}$$

Since the 2-approximation of f -ND was instrumental in obtaining a 2-approximate CRND, it seems a helpful first step to see if and when Connected f -ND admits a constant-factor approximation. Lemma 6.2 gives a 2-approximation for Connected f -ND by showing an equivalence:

Definition. Let $f : 2^V \rightarrow \mathbb{Z}$ be a cut requirement function. We say that f is **component-subadditive** if for any two connected components $A, B \subseteq V$, $f(A \cup B) \leq f(A) + f(B)$.

Lemma 6.2. *If f is symmetric and component-subadditive, then $\text{ConLP}(f, G)$ is equivalent to $\text{NDLP}(f, G)$.*

Proof. Clearly, any feasible solution to $\text{NDLP}(f, G)$ is feasible to $\text{ConLP}(f, G)$; a solution that satisfies constraints for all sets necessarily satisfies constraints for fully connected sets.

We first show that if $x(\delta(S)) \geq f(S)$ for all fully connected sets, then it does so for all connected sets. Let S be connected, but not fully connected. Then $\bar{S}$ has some connected components $S_1, \dots, S_k$. Each of these components is fully connected. So,

$$x(\delta(S)) = \sum_i x(\delta(S_i)) \geq \sum_i f(S_i) \geq f(\bar{S}) = f(S)$$

So $x(\delta(S)) \geq f(S)$ for all connected S .

Now, let S be such that neither S nor $\bar{S}$ are connected (if $\bar{S}$ is connected, S satisfies the constraints by symmetry). Now, S has some connected components $S_1, \dots, S_k$. As these are connected,

$$x(\delta(S)) = \sum_i x(\delta(S_i)) \geq \sum_i f(S_i) \geq f(S)$$

So $x(\delta(S)) \geq f(S)$. □

6.3 Further Steps

6.3.1 Decomposition

For the remainder of this chapter, we denote $f(S) := f^{SND}(S)$.

We describe here an attempt at applying decomposition to PRSND. Similarly to the canonical decomposition, let $A \subseteq V$ be a maximum-deficiency small cut. Since f is component subadditive, one may assume that A is fully connected. Let $Z := \delta_G(A)$. We sort S into one of three types:

- (i) $\delta_G(S) \supseteq Z$, and thus $\delta_Z(A - S) = \emptyset$
- (ii) $\delta_G(A - S) \supseteq Z$, and thus $\delta_Z(S) = \emptyset$
- (iii) Both $\delta_Z(S)$ and $\delta_Z(A - S)$ are nonempty

We give a new definition of restriction $f_A : 2^A \rightarrow \mathbb{Z}$ as follows: $f_A(A) = f_A(\emptyset) = 0$, and for $\emptyset \neq S \subset A$:

$$f_A(S) = \begin{cases} f(\bar{A} \cup S) & S \text{ type (i)} \\ f(S) & S \text{ type (ii)} \\ \max \left\{ f(S) - |\delta_Z(S)|, f(\bar{A} \cup S) - |\delta_Z(\bar{A} \cup S)| \right\} & S \text{ type (iii)} \end{cases}$$

Observe that the restriction for type (iii) sets is the same as the original restriction function for CRSND. If $\emptyset \neq S \subseteq A$ is of type (i), then S is not fully connected in G and thus should not impose a constraint. Moreover, if S is fully connected in $G[A]$, then note that $\bar{A} \cup S$ is fully connected in G . So, we only include the constraint given by $\bar{A} \cup S$. A symmetric argument applies for type (ii) sets.

We wish to see whether a similar decomposition to Chapter 3 may be applied using this restriction, where we take a small cut $A \subseteq S$ of maximum deficiency and define new instances on A and $\bar{A}$. As in Chapter 3, thus, we must ensure that the restriction f_A is weakly supermodular, and that solutions to each sub-instance correspond to a solution to our original. We have not yet been able to show either of these points as yet.

6.3.2 Augmentation

Another approach to approximating PRSND is via *augmentation*. To simplify, we narrow our focus here to single-demand PRSND, where we have a single demand (s, t, k) . The

augmentation problem is roughly stated: given an existing solution H to SD-PRSND with demand $(s, t, k - 1)$, find a solution to the problem with demand (s, t, k) .

Intuitively, this should reduce to a minimal-cut-covering problem, in which we wish to select edges that cover cuts S where the original graph G has the required connectivity, but our solution H does not (that is to say, $\delta_G(S) \geq k$ and $\delta_H(S) < k$).

A similar approach was followed by Chekuri-Jain, leading to their $O(k^2 \log^2 n)$ for PRSND by reducing to a hitting-set problem. If we want to avoid a dependence on n using this approach, then one possibility would be to see if the $\{0, 1\}$ -cut requirement function encoding the augmentation problem admits some decomposition along the lines of that for CRNDP. This is a possibility we have yet to fully explore.

Chapter 7

Conclusion

The Path-Relative Survivable Network Design problem, introduced in 2022 by Dinitz, Koranteng, and Kortsarz, seeks to model fault tolerance in graphs that lack the required connectivity for standard network design. In pursuit of improved approximations for PRSND, this thesis introduces the Cut-Relative Survivable Network Design problem and gives a 2-approximation algorithm for this new problem. This latter problem captures some – but not all – of the difficulty of PRSND.

Using a decomposition technique, we gave a 2-approximation algorithm for CRSND. By splitting our graph into subgraphs and defining suitable subproblems, we show that solutions to CRSND correspond to a collection of solutions to the well-studied f -Network Design problem. Using this, we showed that one may use the method of iterative rounding to approximate CRSND – notably, this is a special case where iterative rounding may be applied to a problem without a weakly supermodular cut requirement function. We went on to give some structural properties of the decomposition, leading to some interesting results on efficient computation of the decomposition. Finally, we showed that PRSND and CRSND are both *APX*-hard.

Further work is outlined in Chapter 6. Since our 2-approximation for CRSND matches the best known approximation factor for standard SND, it is unlikely that improvements can be made to this problem itself; as such, the primary open question is whether improvements may be made to the best known approximations for PRSND.

References

- [CJ24] Chandra Chekuri and Rhea Jain. Approximation algorithms for network design in non-uniform fault models. *arXiv preprint arXiv:2403.15547*, 2024.
- [DKK22] Michael Dinitz, Ama Koranteng, and Guy Kortsarz. Relative survivable network design. In *Proceedings of the 25th APPROX*, pages 41:1–41:19, 2022.
- [DKKN23] Michael Dinitz, Ama Koranteng, Guy Kortsarz, and Zeev Nutov. Improved approximations for relative survivable network design. In *Proceedings of the 21st WAOA*, pages 190–204, 2023.
- [GGP⁺94] Michel X. Goemans, Andrew V. Goldberg, Serge A. Plotkin, David B. Shmoys, Éva Tardos, and David P. Williamson. Improved approximation algorithms for network design problems. In *Proceedings of the 5th SODA*, pages 223–232, 1994.
- [GLS93] Martin Grötschel, László Lovász, and Lex Schrijver. *Geometric algorithms and combinatorial optimization*. Algorithms and Combinatorics 2. Springer, 1993.
- [GW95] Michel X. Goemans and David P. Williamson. A general approximation technique for constrained forest problems. *SIAM J. on Computing*, 24:296–317, 1995.
- [GW96] Michel X. Goemans and David P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In Dorit Hochbaum, editor, *Approximation algorithms for NP-hard problems*, chapter 4. PWS Publishing Company, 1996.
- [Jai01] Kamal Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

- [KNS25] Nikhil Kumar, JJ Nan, and Chaitanya Swamy. Tight guarantees for cut-relative survivable network design via a decomposition technique. *arXiv preprint arXiv:2507.04473*, 2025.
- [LRS11] Lap Chi Lau, R. Ravi, and Mohit Singh. *Iterative methods in combinatorial optimization*. Cambridge University Press, 2011.
- [Wil95] Goemans Michel X. Mihail Milena Vazirani Vijay V. Williamson, David P. A primal-dual approximation algorithm for generalized steiner network problems. *Combinatorica*, 15:435—454, 1995.
- [WS11] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.