ON APPROXIMABILITY AND LP FORMULATIONS FOR MULTICUT AND FEEDBACK SET PROBLEMS

BY

VIVEK MADAN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2018

Urbana, Illinois

Doctoral Committee:

Professor Chandra Chekuri, Chair
Professor Sariel Har-Peled
Assistant Professor Ruta Mehta
Assistant Professor Karthekeyan Chandrasekaran
Professor Anupam Gupta

Abstract

Graph cut algorithms are an important tool for solving optimization problems in a variety of areas in computer science. Of particular importance is the min s-t cut problem and an efficient (polynomial time) algorithm for it. Unfortunately, efficient algorithms are not known for several other cut problems. Furthermore, the theory of NP-completeness rules out the existence of efficient algorithms for these problems if the $P \neq NP$ conjecture is true. For this reason, much of the focus has shifted to the design of approximation algorithms. Over the past 30 years significant progress has been made in understanding the approximability of various graph cut problems. In this thesis we further advance our understanding by closing some of the gaps in the known approximability results. Our results comprise of new approximation algorithms as well as new hardness of approximation bounds. For both of these, new linear programming (LP) formulations based on a labeling viewpoint play a crucial role.

One of the problems we consider is a generalization of the min s-t cut problem, known as the multicut problem. In a multicut instance, we are given an undirected or directed weighted supply graph and a set of pairs of vertices which can be encoded as a demand graph. The goal is to remove a minimum weight set of edges from the supply graph such that all the demand pairs are disconnected. We study the effect of the structure of the demand graph on the approximability of multicut. We prove several algorithmic and hardness results which unify previous results and also yield new results. Our algorithmic result generalizes the constant factor approximations known for the undirected and directed multiway cut problems to a much larger class of demand graphs. Our hardness result proves the optimality of the hitting-set LP for directed graphs. In addition to the results on multicut, we also prove results for multiway cut and another special case of multicut, called linear-3-cut. Our results exhibit tight approximability bounds in some cases and improve upon the existing bound in other cases. As a consequence, we also obtain tight approximation results for related problems.

Another part of the thesis is focused on feedback set problems. In a subset feedback edge or vertex set instance, we are given an undirected edge or vertex weighted graph, and a set of terminals. The goal is to find a minimum weight set of edges or vertices which hit all of the cycles that contain some terminal vertex. There is a natural hitting-set LP which has an $\Omega(\log k)$ integrality gap for k terminals. Constant factor approximation algorithms have been developed using combinatorial techniques. However, the factors are not tight, and the algorithms are sometimes complicated. Since most of the related problems admit optimal approximation algorithms using LP relaxations, lack of good LP relaxations was seen as a fundamental roadblock towards resolving the approximability of these problems. In this thesis we address this by developing new LP relaxations with constant integrality gaps for subset feedback edge and vertex set problems.

To my parents, for their love and support.

Acknowledgments

First and foremost, I would like to express my deepest gratitude towards my advisor, Prof. Chandra Chekuri. I could not have hoped for a better advisor. In addition to being an excellent researcher, you are an amazing mentor. Your guidance and patience have helped me to navigate through some very difficult times. Your advice, whether on research and otherwise has been incredibly useful. It has made me grow as a researcher and as a person and I am sure it will continue to guide me in the future as well.

I have been very fortunate to have worked with many amazing people during my Ph.D.: Prof. Karthekeyan Chandrasekaran, Prof. Alexandra Kolla, Prof. Tamas Király, Prof. Ionnis Koutis, Kristof Bérczi, Shalmoli Gupta, Naman Agarwal, and Ali Kemal Sinop. I would like to thank my collaborators for all the work that we did together. Those long meetings trying to find a path out of hundreds of dead ends have been an incredible learning experience.

I would like to thank Prof. Ruta Mehta, Prof. Karthekeyan Chandrasekaran, Prof. Anupam Gupta, Prof. Sariel-Har-Paled, and Prof. Jeff Erickson for serving on my thesis committee and providing several helpful comments. I was honored to present my results to an excellent group of researchers.

I would like to thank the University of Illinois Urbana-Champaign and the Department of Computer Science for hosting me for Ph.D. It gave me a chance to learn from an exciting group of people: Hsien-Chih Chang, Chao Xu, Sahand Mozaffari, Xilin Yu, Alex Steiger, Shant Boodaghians, Charles Carlson, Mitchell Jones, Konstantinos Koiliaris, and Nirman Kumar. I am fortunate to have worked besides so many excellent colleagues. I would also like to thank the National Science Foundation for funding my research through various grants.

I would like to thank many of the friends who have been part of my life and shaped it in different ways. I am particularly thankful to Pritish Kamath for an exciting time in my undergraduate days. If it were not for the days and nights spent discussing ideas with Pritish, I might not have even considered pursuing a Ph.D. I am also espcially thankful to my amazing roommates and friends Ashish Khetan and Prasanna Giridhar. Some of the best memories of my Ph.D. years have been hanging out with Ashish and Prasanna playing games and arguing about stupid things.

Finally, I would like to thank my family for their unending love and support throughout my life. Nothing in my life would have been possible without the continous support given by my parents, my brother Ravi and my sister Bharti.

Table of Contents

Chapter 1

Introduction

Combinatorial optimization refers to the study of maximizing or minimizing a function over a finite domain subject to certain constraints. One mathematical structure which can model a large class of combinatorial optimization problems is a *graph*. Many intricate real-life problems have been formulated as optimization problems on graphs and several beautiful algorithms have been designed for solving these problems. One of the most famous such problems is the min s-t Cut problem:

s-t CUT

Input: Graph $G = (V, E)$ along with non-negative edge weights $w(e), e \in E$ and two terminal nodes $s, t \in V$.

Output: Minimum weight set of edges $E' \subset E$ such that there is no path from s to t in $G - E'$.

Here, $G - E'$ refers to the graph $(V, E \setminus E')$. Ford-Fulkerson [43] described an efficient algorithm for the min s-t CUT problem. Ever since then, numerous generalizations of s-t CUT have been considered. Unfortunately, many of these problems are difficult to solve. This is captured by the fact that these problems are NP-hard, and assuming the $P \neq NP$ conjecture, do not admit an efficient algorithm. We investigate the complexity of these NP-hard problems through the lens of approximability. We deisgn approximation algorithms as well as prove hardness of approximation results. Formal definitions of approximation and hardness of approximation are given in the next section.

One generalization of s-t CUT which has attracted substantial attention over the past 30 years is the Multicut problem.

MULTICUT

Input: Graph $G = (V, E)$ along with non-negative edge weights $w(e), e \in E$ and a set of terminal pairs $\{(s_i, t_i) \mid s_i, t_i \in V, i \in \{1, \ldots, k\}\}$.

Output: Minimum weight set of edges $E' \subset E$ such that there is no path from s_i to t_i in $G - E'$ for $i \in \{1, \ldots, k\}$.

If G is an undirected graph, we refer to MULTICUT as the UNDIR-MULTICUT problem, and if G is a directed graph, we refer to MULTICUT as the DIR-MULTICUT problem. MULTICUT has been useful in solving several real-life problems. For example, it has been successfully used as a sub-routine in problems such as communication cost minimization in parallel computations, efficient partitioning of files in a network, and VLSI design [66, 79]. Study of MULTICUT has also given rise to many new and interesting concepts which have been useful in solving related problems [66, 79].

Due to a variety of applications and interesting connections to other problems, MULTICUT has been extensively studied. This study has produced a large collection of results for several variants and special cases [1, 2, 3, 15, 16, 27, 29, 37, 46, 47, 48, 49, 52, 56, 62, 66, 70, 74, 79, 80]. However, there are critical gaps in our understanding of the approximability of the problem. In this thesis we close some of these gaps and also unify several results by taking a new viewpoint.

Another class of problems we study in this thesis are feedback set problems. In the general feedback set problem, we are given a directed or undirected graph and a collection $\mathcal{C}$ of cycles which may be implicitly defined. Our goal is to find a minimum weight set of vertices or edges that meet all cycles in $\mathcal{C}$. This problem arises in a variety of areas such as deadlock prevention, testing of circuits, Bayesian inference, genetics, and artificial intelligence [7, 76, 83]. In this thesis we focus on the feedback set problem when the graph is undirected, and $\mathcal{C}$ consists of cycles passing through a given subset of vertices.

Subset Feedback Vertex Set

Input: Undirected graph $G = (V, E)$ along with non-negative node weights $w(v), v \in V$, and a set $S = \{s_1, \ldots, s_k\} \subset V$ of terminals.

Output: Minimum weight set of nodes $V' \subset V$ such that there is no cycle containing any s_i in $G - V'$.

Subset Feedback Edge Set

Input: Undirected graph $G = (V, E)$ along with non-negative edge weights $w(e), e \in E$, and a set $S = \{s_1, \ldots, s_k\} \subset V$ of terminals.

Output: Minimum weight set of edges $E' \subset E$ such that there is no cycle containing any s_i in $G - E'$.

These problems have been studied for over three decades [5, 7, 8, 26, 35, 40, 45, 51, 57]. However, we still do not fully understand the approximability of these problems. In fact, not much progress had been made in improving the approximability bounds for more than 20 years. In this thesis we present a new approach which could be useful in obtaining improvements for both Subset Feedback Edge and Vertex Set problem.

Our focus in this thesis is on the design of approximation algorithms, formulating new linear programming relaxations, and proving hardness of approximation results for the problems mentioned above. We start by formally defining some of these concepts. Readers familiar with these definitions can skip the next section.

1.1 Background

In this thesis we focus on NP optimization problems (NPO).

Definition 1.1. *An NPO problem $\mathcal{P}$ is a tuple $(\Lambda, \mathcal{X}, val, goal)$ such that*

- *Λ is a set of strings and denotes the set of input instances of problem $\mathcal{P}$. The language defined by Λ is recognizable in polynomial time.*

- *For every $I \in \Lambda$, $\mathcal{X}_I$ is a set of strings and denotes the collection of feasible solutions for I. There exists a polynomial time algorithm, which given I and x, decides if x is in $\mathcal{X}_I$ or not in $\mathcal{X}_I$.*

- *For $I \in \Lambda, x \in \mathcal{X}_I$, $val(I,x)$ is the value of solution x on instance I. There exists a polynomial time algorithm to compute $val(I,x)$ for any $I \in \Lambda, x \in \mathcal{X}_I$.*

- *$goal \in \{\min, \max\}$ specifies whether we wish to find a feasible solution with minimum or maximum value.*

1.1.1 Approximation algorithms

An algorithm is practically useful if it is efficient. In this thesis we call an algorithm to be *efficient* if its running time on instance I is $p(|I|)$ for some fixed polynomial p where $|I|$ is the size of the instance. For example, s-t CUT admits an algorithm which runs in time polynomial in the size of the input graph and hence, is efficient. However, such polynomial time algorithms are not known for MULTICUT even for three terminal pairs ($k = 3$). In fact, efficient algorithms for many of the problems arising in practice are not known. The theory of NP-completeness has shown that many problems are NP-hard, and hence assuming the $P \neq NP$ conjecture, do not admit a polynomial time algorithm.

Since polynomial time algorithms are unlikely to exist for these problems, several heuristics are designed. One line of work is the area of approximation algorithms. It asks the following question: *Can we find solutions in polynomial time which are provably close to the optimum solution?*

For instance $I \in \Lambda$ of a minimization problem $\mathcal{P}$, let $OPT(I) = \min_{x \in \mathcal{X}_I} val(x)$ denote the value of an optimum solution.

Definition 1.2. *An algorithm $\mathcal{A}$ is an α-approximation algorithm for a minimization problem $\mathcal{P} = (\Lambda, \mathcal{X}, val, \min)$ if for all $I \in \Lambda$,*

$$\alpha \geq \frac{val(\mathcal{A}(I))}{OPT(I)} \tag{1.1}$$

Here, α is referred to as the approximation ratio of $\mathcal{A}$.

Note that α may or may not depend on I. Vertex Cover admits a 2-approximation algorithm and Set Cover admits a $(1 + \ln n)$-approximation algorithm where n is the size of the ground set.

1.1.2 Linear programming relaxations and rounding

A generic scheme for designing a large class of approximation algorithms is to map the original problem to an optimization problem in real space. Let $I \in \Lambda$ be an instance of a minimization problem $\mathcal{P}$. Often the nature of $\mathcal{P}$ ensures that the feasible solutions ($\mathcal{X}_I$) to I can be represented by a set S of integer vectors in a high-dimensional (say m) real space. Moreover, the value function val can represented by a linear function f over m variables. Consider the convex hull $\mathcal{T}$ of the point set S. Since the function f is linear, minimum of f over $\mathcal{T}$ is realized at one of the vertices of $\mathcal{T}$ which is in S. Hence we can solve the problem by minimizing the function f over $\mathcal{T}$. However, we may not know how to obtain a compact description of this convex hull $\mathcal{T}$ or minimize the function f efficiently if the problem $\mathcal{P}$ is NP-hard. Instead, we find a larger polyhedron $\mathcal{H}$ which is a super set of $\mathcal{T}$ such that the set of integer points in $\mathcal{H}$ are exactly the points in S and the minimum of f over $\mathcal{H}$ can be found efficiently. This is done by formulating the problem as an integer linear program (ILP) and then relaxing the integer constraints to get a linear programming (LP) relaxation whose solutions define $\mathcal{H}$. It does not fully solve the problem as the minimum of f over $\mathcal{H}$ may not be achieved at a point in S. Once we find the point x which minimize f over $\mathcal{H}$, we find a point y in S such that $f(y)$ is not much larger than $f(x)$.

An ILP is an optimization or feasibility program where the objective function and the constraints are linear, and the variables are restricted to be integers. In a 0-1 ILP, variables may only take value 0 or 1. An LP relaxation R for a problem $\mathcal{P}$ is obtained as follows: given a problem instance $I \in \Lambda$, we formulate it as a 0-1 ILP (say W_I) minimizing a linear function (say f) of a certain number of variables ($\overline{x} = x_1, \ldots, x_m$) subject to $\overline{x} \in \{0,1\}^m$ and a set of linear constraints $(L_1(\overline{x}), \ldots, L_k(\overline{x}))$. Let the LP with the objective function f and constraint set $L_1(\overline{x}), \ldots, L_k(\overline{x})$ and $0 \leq x_i \leq 1, i \in \{1, \ldots, m\}$ be R_I: LP relaxation R for instance I.

Let the optimum of R_I be achieved at x^* and $OPT_R(I) = f(x^*)$. Once we solve the LP R_I, we design an algorithm to convert the fractional vector x^* (values between 0 and 1) into an integer vector $\overline{y}$ (values 0 or 1) satisfying the linear constraints $(L_1(\overline{x}), \ldots, L_k(\overline{x}))$. This step is called the *rounding phase* of the algorithm. It involves proving that the cost of the integer vector is at most a small factor (say α) times the cost of the fractional vector. That is, $f(\overline{y}) \leq \alpha f(x^*) = \alpha OPT_R(I)$. Since value of the linear program R_I is at most the value of integer linear program W_I and the integer linear program exactly formulates the problem, we get $OPT_R(I) \leq OPT(I)$. Hence, $f(\overline{y}) \leq \alpha OPT(I)$ and the solution corresponding to y is an α-approximate solution for I.

There is a canonical ILP for graph cut problems, called the *hitting-set ILP*. It consists of constraints of the following form: *For any forbidden structure H, a feasible solution must contains at least one vertex or an edge from H.* Figure 1.1 shows such an ILP and its LP relaxation for MULTICUT. It consists of a variable x_e for each edge e and an exponential (in size of the input graph) set of constraints; the constraints specify that for any path p from s_i to t_i, sum of x_e along p is at least 1. Even though the LP contains an exponential number of constraints, it can be solved in polynomial-time using the Ellipsoid method [58]. One can also write a compact polynomial-sized formulation using distance variables. We can also write similar LP relaxations for SUBSET-FES and SUBSET-FVS as well.

This framework of writing LP relaxations and designing rounding schemes has been very useful in the design of approximation algorithms. However, there is a bottleneck in this framework, namely *integrality gap*. We cannot design an LP-based approximation algorithm with approximation ratio better

$$
\begin{array}{lll}
\min & \sum_{e\in E} w_e x_e & \\
 & \sum_{e\in p} x_e \geq 1 & p \in \mathcal{P}_{s_i t_i}, i \in [1,k] \\
 & x_e \in \{0,1\} & e \in E
\end{array}
$$

$$
\begin{array}{lll}
\min & \sum_{e\in E} w_e x_e & \\
 & \sum_{e\in p} x_e \geq 1 & p \in \mathcal{P}_{s_i t_i}, i \in [1,k] \\
 & 0 \leq x_e \leq 1 & e \in E
\end{array}
$$

Figure 1.1: Hitting set ILP on top and LP (DISTANCE-LP) at bottom for MULTICUT

than the worst-case gap between the value of the ILP and the LP.

Definition 1.3. *For a minimization problem $\mathcal{P} = (\Lambda, \mathcal{X}, val, goal)$, an LP relaxation R of $\mathcal{P}$ and an instance $I \in \Lambda$, let $OPT_R(I)$ be the value of R for instance I.*

$$
\textit{Integrality gap}(R) = \sup_{I\in\Lambda} \frac{OPT(I)}{OPT_R(I)} \tag{1.2}
$$

Novelty of an approximation algorithm lies in coming up with a good relaxation and designing a rounding scheme proving tightest possible bound on integrality gap. Integrality gap of the DISTANCE-LP is sometime referred to as the *flow-cut gap* since the dual of the LP is a maximum multicommodity flow LP.

Search for good relaxations have led to the development of generic strengthening procedures such as Lovász-Schrijver [67], Sherali-Adams [78], and Sum-of-Squares [64]. We start with a weak LP or SDP relaxation and add more constraints in a systematic way which are valid for the integer solutions.

1.1.3 Hardness of approximation

For every approximation algorithm that we design, a natural question arises: *Can we do better?* Can we design an algorithm with better approximation ratio? This has led to the study of the limits of approximability.

Definition 1.4. *A minimization problem $\mathcal{P}$ is α-hard if there is no polynomial time $(\alpha - \epsilon)$-approximation algorithm for $\mathcal{P}$ for any constant $\epsilon > 0$.*

As in approximation ratio, α may be constant or a function of $|I|$. Since we cannot yet rule out all problems in NPO being solvable in polynomial time, we cannot prove an unconditional hardness result with $\alpha > 1$ for any minimization problem $\mathcal{P} \in NPO$. Hence, hardness results are often proved under certain assumptions.

Definition 1.5. *A minimization problem $\mathcal{P}$ is APX-hard if there exists $c > 1$ such that assuming $P \neq NP$, $\mathcal{P}$ is c-hard.*

The famous PCP theorem by Arora *et al.* [4] showed a breakthrough result in this direction: for some constant $c < 1$, there is no c-approximation algorithm for MAX-3-SAT assuming $P \neq NP$. Subsequently, building upon this work, hardness results have been shown for problems such as Set Cover, Independent Set and Coloring [38,39,55]. In some cases, such as Set Cover, these results showed that a simple greedy heuristic achieves the best possible approximation ratio.

Even though we get tight approximation results for problems such as Set Cover, there are several fundamental problems such as Vertex Cover where the proven lower bounds do not match the known approximation ratios. The best-known approximation algorithm for Vertex Cover achieves a factor 2 approximation while the best lower bound rules out a c-approximation for some $c < 2$. Unique Games Conjecture (UGC) has come to the rescue for these problems [59]. It is an assumption on the inapproximability of a combinatorial optimization problem. Here, we present an equivalent formulation due to Khot, Kindler, Mossel and O'Donnell [60].

Γ-Max Lin (p): Input is a set of linear equations over integers of the form $x_i - x_j \equiv c_{ij} \mod p$. The goal is to find an integer assignment to x_i's such that maximum number of equations are satisfied.

Conjecture 1.1. *(**Unique Games Conjecture [60]**) Assuming $P \neq NP$, for any $\epsilon, \delta > 0$ there exists an integer p such that no polynomial time algorithm can distinguish between instances of Γ-Max Lin (p) where at least $(1-\epsilon)$ fraction of the equations are satisfiable and instances where at most δ fraction of the equations are satisfiable.*

Assuming UGC, inapproximability results have been obtained for several important problems. For example, assuming UGC, known approximation factors for Vertex Cover and Max-Cut are tight. Raghavendra [71] proved this phenomenon for a larger class of problems known as Max-CSP.

Though there is no consensus on the truthfulness of the conjecture, studying the implications of UGC has been very fruitful. It has exposed interesting new connections as well as the limits of linear and semidefinite programming for several problems [72].

1.2 Motivating Questions

As we already mentioned, both multicut and feedback set problems have been studied extensively over the last three decades [1,2,3,5,7,8,15,16,26,27,29,35,37,40,45,46,47,48,49,51,52,56,57,62,66,70,74,79,80]. In this section, we describe our motivations to reconsider these problems. We mention some concrete questions which addressed in this thesis.

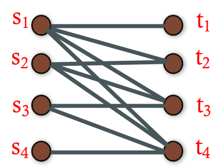


Figure 1.2: Demand graph for SKEW-MULTICUT

1.2.1 Role of demand graph in MULTICUT

UNDIR-MULTICUT admits an $O(\log k)$-approximation algorithm [47]. DIR-MULTICUT on the other hand admits $\min(k, \tilde{O}(n^{11/23}))$-approximation algorithm [2]; here $n = |V|$. Some special cases also admit improved approximation algorithms. For example, UNDIR-MULTICUT admits an $O(1)$-approximation if G is planar, $O(\log g)$-approximation if G is of genus g, $O(t)$-approximation algorithm if G has treewidth t [1, 37, 62]. UNDIR-MULTICUT is at least as hard as Vertex Cover even in trees. Hence, from known hardness result for Vertex Cover there exists $c > 1$ such that assuming $P \neq NP$, there is no $(c - \epsilon)$-approximation algorithm (APX-hardness). Under UGC, UNDIR-MULTICUT does not have a c-approximation for any fixed constant c [16]. DIR-MULTICUT is known to be a harder problem. Assuming $NP \neq ZPP$, it is hard to approximate DIR-MULTICUT to within a factor of $\Omega(2^{\log^{1-\epsilon} n})$ [27]; evidence is also presented in [27] that it could be hard to approximate to within an $\Omega(n^{\delta})$ factor for some fixed $\delta > 0$.

These results describe the approximability of MULTICUT when the demand pairs are arbitrary, with some improved results when G is restricted. In this thesis we are interested in the setting where the demand pairs are restricted and G is arbitrary. To capture the structure of demand pairs, consider an equivalent formulation of MULTICUT. *The input now consists of an edge-weighted* supply *graph* $G = (V, E)$ *and a* demand *graph* $H = (V, F)$. *The goal is to find a minimum weight set of edges* $E' \subseteq E$ *such that for each edge* $f = (s, t) \in F$, *there is no path from* s *to* t *in* $G - E'$. In other words the source-sink pairs are encoded in the form of the demand graph H. Either both G and H are directed in which case we refer to the problem as DIR-MULTICUT (directed MULTICUT) or both are undirected in which case we refer to the problem as UNDIR-MULTICUT (undirected MULTICUT). The question we ask in the thesis is the following:

Question 1.1. *What structural aspects of the demand graph determine the approximation ratio? Can we get improved approximation algorithms for restricted demand graphs?*

Before we describe a concrete application that motivated us to consider this question, we mention the well-known MULTIWAY CUT problem in undirected graphs (EDGE-WT-MWC) and directed graphs (DIR-MWC). In MULTIWAY CUT, H is the complete (or bi-directed complete) graph on a set of k terminals. This problem has been extensively studied over the years. EDGE-WT-MWC admits a 1.29-approximation [77] and DIR-MWC admits a 2-approximation [23, 70]. In a recent work [19], motivated by connections to the problem of understanding the information capacity of networks with delay constraints, a special case of MULTICUT namely SKEW-MULTICUT was considered. The demand graph H is a bipartite graph with k terminals $s_1, \ldots, s_k$ on one side and k terminals $t_1, \ldots, t_k$ on the other side: (s_i, t_j) is an edge in H iff

$i \le j$. See Figure 1.2 for an example with $k = 4$. If the edges are directed from s's to the t's we obtain the directed version. It was shown in [19] that the integrality gap (flow-cut gap) of the hitting-set LP (DISTANCE-LP 1.1) for DIR-SKEW-MULTICUT gives an upper bound on the capacity advantage of network coding with delays. Further, it was established that the flow-cut gap for DIR-SKEW-MULTICUT is $O(\log k)$ which is in contrast to the general setting where the gap can be as large as k. One can show APX-hardness and a constant factor lower bound on the flow-cut gap for UNDIR-SKEW-MULTICUT (and hence also for DIR-SKEW-MULTICUT) via a reduction from EDGE-WT-MWC problem. The following natural questions arose from this application.

Question 1.2. *What is the approximability of* UNDIR-SKEW-MULTICUT *and* DIR-SKEW-MULTICUT*? Is the flow-cut gap $O(1)$ for* UNDIR-SKEW-MULTICUT *and even* DIR-SKEW-MULTICUT*?*

Some previous work has also examined the role that demand graph plays in MULTICUT. One example is the original paper of Garg, Vazirani and Yannakakis [47] which showed that one can obtain an $O(\log h)$-approximation for UNDIR-MULTICUT where h is the vertex cover size of the demand graph. This was generalized by Steurer and Vishnoi [80] who showed that h can be chosen to be $\min_S \max_T |S \cap T|$ where S is a vertex cover in H and T is an independent set in H. Both these results are based on bounding the flow-cut gap. For UNDIR-SKEW-MULTICUT, h is equal to k. Hence, these results fail to improve upon the logarithmic bound on the flow-cut gap or the approximation ratio for UNDIR-SKEW-MULTICUT. Other than DIR-MWC, not much is known for DIR-MULTICUT exploiting the structure of demand graphs which could help improve the flow-cut gap or the approximation ratio for DIR-SKEW-MULTICUT.

In this thesis we provide two results. First, we obtain a 2-approximation for UNDIR-SKEW-MULTICUT. Second, we show that under UGC, for any fixed constant k, the hardness of approximation for DIR-SKEW-MULTICUT coincides with its flow-cut gap. Our results for SKEW-MULTICUT are special cases of more general results that examine the role of the demand graph H in the approximability of MULTICUT. Our algorithmic results generalize the constant factor approximation algorithms known for undirected multiway cut (EDGE-WT-MWC) and directed multiway cut (DIR-MWC) problem. In addition to a 2-approximation for UNDIR-SKEW-MULTICUT, these results also improve the result of Steurer and Vishnoi [80] when h is constant. We get a factor 2 approximation for UNDIR-MULTICUT if $h = \min_S \max_T |S \cap T|$ is constant where S is a vertex cover in the demand graph and T is an independent set in the demand graph. Our hardness results significantly improve the inapproximability bounds for several important cases of DIR-MULTICUT. Of particular importance is the case when the demand graph is a directed matching of size k. Our hardness result implies that assuming UGC, DIR-MULTICUT with this demand graph is NP-hard to approximate within a factor $k - \epsilon$ for any fixed $\epsilon > 0$.

1.2.2 LP relaxations and improving approximation for subset feedback set problems

FVS, SUBSET-FVS and SUBSET-FES all admit constant factor approximation algorithms. In particular there are 2-approximations for FVS [5, 8] and SUBSET-FES [34], and an 8-approximation for SUBSET-

FVS [35]. However, hardness bounds known for SUBSET-FES and SUBSET-FVS do not match the known approximation ratio. In fact, the best known lower bounds are for the case of $|S| = 1$. SUBSET-FES and SUBSET-FVS with one terminal are equivalent to the edge-weighted multiway cut (EDGE-WT-MWC) and node weighted multiway cut (NODE-WT-MWC) problem respectively. Assuming UGC, EDGE-WT-MWC is hard to approximate within a factor $c - \epsilon$ for some $c < 2$ and NODE-WT-MWC is hard to approximate within a factor of $2 - \epsilon$ for any $\epsilon > 0$. Not much progress had been made in closing these gaps in the last 20 years.

We can write a hitting set LP relaxation for feedback set problems. For instance, consider FVS. The relaxation has a variable $z(v) \in [0, 1]$ for each $v \in V$, and for each cycle C, a constraint $\sum_{v \in C} z(v) \geq 1$. This LP relaxation has an $\Omega(\log n)$ integrality gap [34]. Constant factor approximation algorithms have mainly relied on combinatorial techniques at the high-level. The non-trivial 2-approximation algorithm for FVS from [5] has been later interpreted as a primal-dual algorithm by Chudak *et al.* [26], however, the underlying LP is not known to be solvable in polynomial-time and does not generalize to SUBSET-FES or SUBSET-FVS. The 2-approximation for SUBSET-FES [34] is simple and combinatorial but delicate to analyze. The 8-approximation for SUBSET-FVS [35] is very complicated to describe and analyze; the algorithm is combinatorial at the high-level but solves a sequence of relaxed multicommodity flow LPs to optimality. Recall that SUBSET-FVS captures the node-weighted undirected multiway cut problem (NODE-WT-MWC) as a special case and all the known constant factor approximations for NODE-WT-MWC are via LP relaxations. To some extent this explains why one needs LP-type techniques for SUBSET-FVS.

Question 1.3. *Does there exists polynomial time solvable LP relaxations for* SUBSET-FES *and* SUBSET-FVS *with constant integrality gaps?*

Even *et al.* [34] write that "finding a linear program for SUBSET-FES and SUBSET-FVS for which the integrality gap is constant is very challenging". One of the open problems in Vazirani's book on approximation [82] is also to find a simpler constant factor approximation algorithm for SUBSET-FVS with the eventual goal of finding an improved approximation ratio.

In this thesis we describe such LP relaxations for SUBSET-FES and SUBSET-FVS and derive constant factor approximations through them.

1.2.3 New Techniques

Most of the LP-based approximation algorithms known for the cut problems involve the hitting-set LP relaxation. Recall the basic idea for a hitting-set LP: For any subgraph $H \subset G$ such that it contains some structure which should not be present in the graph after deleting a feasible solution, we must remove at least one edge or vertex from H. In case of MULTICUT, we must remove at least one edge from a path from s to t where $(s, t) \in F$. In case of SUBSET-FES or SUBSET-FVS, we must remove at least one edge or vertex from a cycle passing through a terminal.

Better than hitting set LP: Hitting-set LP does not always yield optimal approximation algorithms.

For example, as discussed above, feedback set problems admit constant factor approximation but the hitting-set LPs for feedback set problems have a logarithmic integrality gap [34]. Similarly, a related problem known as the diamond hitting set has a constant factor approximation, but the hitting-set LP has a logarithmic integrality gap [41]. In fact, the more general problem of planar-F-deletion admits a constant factor approximation algorithm in the unweighted case but the hitting-set LP has large integrality gap [42]. So, one may ask the question:

Question 1.4. *Can we write LP relaxations for graph cut problems with integrality gap better than the hitting set LP?*

In this thesis we describe such LP relaxations for Multicut and Feedback Set problems. The basic idea of these LP relaxations is the following:

Suppose we know the optimum solution X to the problem. We consider the graph $G-X$ and define a labeling for all the vertices of graph based on the structure of the graph $G-X$. Then, we define linear constraints which must be satisfied by any such labeling and write a linear program comprising of all such constraints.

Labeling-based LPs has constant integrality gaps for SUBSET-FES and SUBSET-FVS and yield the above-mentioned result regarding constant factor approximations for SUBSET-FES and SUBSET-FVS through LP relaxations. It also yields the algorithmic result about UNDIR-MULTICUT generalizing the constant factor approximation algorithm for undirected multiway cut (EDGE-WT-MWC). For DIR-MULTICUT however, we show that the *labeling based LP relaxation is equivalent to the* DISTANCE-LP and hence, the integrality gap of labeling based LP relaxations is same as the flow-cut gap.

Integrality Gap vs Hardness of approximation: As we mentioned in the previous paragraph, labeling based LP does not help improve the integrality gap over the flow-cut gap for DIR-MULTICUT. So, a natural question to ask is if we can write another LP or other convex relaxation with better integrality gap? Or can we prove that there does not exists LP relaxations with better integrality gap? Even more ambitiously, can we prove that there does not exist algorithms with approximation ratio better than some function of flow-cut gap?

Question 1.5. *Is hardness of approximation a function of the flow-cut gap for* DIR-MULTICUT*?*

Surprisingly, such mysterious connections between the integrality gap and the hardness of approximation is known for several problems. For example, Set Cover has a natural LP relaxation with integrality gap at most $1+\ln n$ and is hard to approximate better than $\ln n$ assuming $P \neq NP$ [38]. Max-cut has an SDP relaxation with integrality gap at most 0.878 and is hard to approximate better than the same factor assuming UGC [60]. Raghavendra generalized this result and showed that assuming UGC, hardness of approximation matches the integrality gap of the canonical SDP for any Max-CSP problem [71]. Ene *et al.* showed a similar result for Min-CSP and Basic LP which states that assuming UGC, for any Min-CSP with a NAE_2-predicate, hardness of approximation matches the integrality gap of the basic LP [31].

In this thesis *we show a similar phenomenon for* DIR-MULTICUT *and* DISTANCE-LP *which proves that assuming UGC, hardness of* DIR-MULTICUT *with a fixed bi-partite demand graph matches the flow-cut gap.*

1.3 Results and Organization

In this section, we formally state the problems and the results proved for these problems in the subsequent chapters.

1.3.1 Role of demand graph in MULTICUT

In Chapter 2, we investigate the role of the demand graph in the approximability of MULTICUT. It consists of three meta results. First result (Theorem 1.1) is a 2-approximation algorithm for UNDIR-MULTICUT for a class of demand graphs which captures both undirected multiway cut (EDGE-WT-MWC) and UNDIR-SKEW-MULTICUT. Second result (Theorem 1.3) is an approximation algorithm for DIR-MULTICUT which improves the approximation ratio over the worst-case bound for a large class of demand graphs. Third result (Theorem 1.4) proves that the hardness of approximation is a function of flow-cut gap for DIR-MULTICUT assuming UGC is true.

(i) 2-approximation for tK_2-free demand graphs

We start with the UNDIR-MULTICUT problem in Chapter 2.

UNDIR-MULTICUT: The input is an undirected *supply* graph $G = (V_G, E)$ and an undirected *demand* graph $H = (V_H, F)$, $V_H \subseteq V_G$, along with non-negative edge weights $w(e), e \in E$. The goal is to find minimum weight set of edges $E' \subseteq E$ such that for each edge $f = (s, t) \in F$, there is no path from s to t in $G - E'$.

Our first result is for UNDIR-MULTICUT with tK_2-free demand graphs [15, 77]. This class is inspired by the observation that the UNDIR-SKEW-MULTICUT demand graph does not contain a matching with two edges as an *induced subgraph*.[1] A graph H is tK_2-free for an integer $t > 1$ if it does not contain a matching of size t as an induced subgraph.

Theorem 1.1. *There is a 2-approximation algorithm with running time poly$(n, k^{O(t)})$ on instances of* UNDIR-MULTICUT *with supply graph G and tK_2-free demand graph H. Here $n = |V_G|$ and $k = |V_H|$.*

It implies a 2-approximation for UNDIR-SKEW-MULTICUT.

UNDIR-SKEW-MULTICUT: The input is an undirected graph $G = (V, E)$ and a set of vertices $s_1, \ldots, s_k, t_1, \ldots, t_k \in V$ along with non-negative edge weights $w(e), e \in E$. The goal is to find minimum weight set of edges $E' \subset E$ such that for every $1 \le i \le j \le k$, there is no path from s_i to t_j in $G - E'$.

[1] G' is an induced subgraph of $G = (V, E)$ if $G' = G[V']$ for some $V' \subset V$.

Since demand graph in UNDIR-SKEW-MULTICUT instances are $2K_2$-free, Theorem 1.1 implies a 2-approximation for UNDIR-SKEW-MULTICUT which improves upon the previous $O(\log k)$-approximation achieved by bounding the flow-cut gap [19].

Corollary 1.1. UNDIR-SKEW-MULTICUT *admits a polynomial-time* 2*-approximation algorithm.*

Recall that if $h = \min_S \max_T |S \cap T|$ where S is a vertex cover and T is an independent set in the demand graph H, then UNDIR-MULTICUT admits an $O(\log h)$-approximation algorithm achieved by bounding flow-cut gap [47, 80]. Our result implies a factor 2-approximation algorithm if h is constant since such demand graphs are tK_2-free for $t = h + 1$.

Corollary 1.2. *Consider* UNDIR-MULTICUT *with supply graph G and demand graph H such that $n = |V_G|, k = |V_H|$. Let $h = \min_S \max_T |S \cap T|$ where S is a vertex cover and T is an independent set in H. Then, there exists a* 2*-approximation algorithm with running time poly$(n, k^{O(h)})$.*

Note that a graph containing $t-1$ parallel edges is tK_2-free. Via known lower bounds [47], DISTANCE-LP has integrality gap $\Omega(\log(t-1))$ on tK_2-free demand graphs. Thus, one cannot obtain a constant approximation for tK_2-free demand graphs for all fixed t using DISTANCE-LP. Our algorithm relies on a different relaxation and a reduction to uniform metric labeling [63]. Also, as we noted earlier, assuming UGC, UNDIR-MULTICUT is NP-hard to approximate within a constant factor [16]. Hence, assuming UGC, it is unlikely that one can obtain a fixed constant factor approximation for UNDIR-MULTICUT with tK_2-free demand graphs in $poly(n, k, t)$ time. In other words, exponential dependence on t in the running time poly$(n, k^{O(t)})$ is necessary if UGC is true.

(ii) $(k-1)$-approximation for k-matching extension free demand graphs

Next, we discuss the directed multiway cut (DIR-MWC) and directed multicut problem (DIR-MULTICUT).

DIR-MWC: The input is a directed graph $G = (V, E)$ and a set of vertices $s_1, \ldots, s_k \in V$ along with non-negative edge weights $w(e), e \in E$. The goal is to find minimum weight set of edges $E' \subseteq E$ such that for any $i \neq j \in [1, k]$, there is no path from s_i to s_j in $G - E'$.

DIR-MULTICUT: The input is a directed *supply* graph $G = (V_G, E)$ and a directed *demand* graph $H = (V_H, F)$, $V_H \subseteq V_G$ along with non-negative edge weights $w(e), e \in E$. The goal is to find minimum weight set of edges $E' \subseteq E$ such that for each edge $f = (s, t) \in F$, there is no path from s to t in $G - E'$.

Naor and Zosin obtained a 2-approximation for DIR-MWC in an elegant, surprising and somewhat mysterious fashion [70]. They write an LP relaxation called the *relaxed multiway flow* relaxation which is within a factor of 2 of the natural DISTANCE-LP, and show that an *optimum* solution to this new relaxation can be rounded without any loss in the approximation. This gives an indirect proof that DISTANCE-LP has an integrality gap (flow-cut gap) of at most 2 for DIR-MWC. The proof of correctness crucially relies on the complementary slackness properties of the optimum solution and is partly inspired by the ideas in [49]. Even though the result is beautiful in its own right, it is hard to generalize to DIR-MULTICUT. The relaxed multiway flow LP is specific to DIR-MWC and does not generalize to DIR-MULTICUT.

We simplify this result and give an algorithm directly proving that the flow-cut gap for DIR-MWC is at most 2.

Theorem 1.2. *There is a randomized algorithm that given a feasible solution* $\boldsymbol{x}$ *to* DISTANCE-LP *for* DIR-MWC*, returns a feasible integral solution of expected cost at most* $2\sum_e w_e x_e$*, and runs in* $O(m + n\log n)$ *time. The algorithm can be derandomized to yield a deterministic* 2*-approximation algorithm that runs in* $O(m\log n)$ *time. Here,* $m = |E(G)|, n = |V(G)|$.

This result implies that the flow-cut gap for DIR-MWC is at most 2 and there is a polynomial-time rounding algorithm that achieves this upper bound. In Chapter 2, we prove a similar result for DIR-MULTICUT with k-matching extension free demand graphs as well and bound the flow-cut gap by $k-1$. We say that a directed demand graph $H = (V_H, F)$ contains an induced k-matching-extension if there are two subsets of V_H, $S = \{s_1, s_2, \ldots, s_k\}$ and $T = \{t_1, t_2, \ldots, t_k\}$ satisfying the following properties: (i) for $1 \le i \le k$, $(s_i, t_i) \in F$ and (ii) for $i > j$, $(s_i, t_j) \notin F$. Note that $s_1, s_2, \ldots, s_k$ are distinct since S is a set and similarly $t_1, \ldots, t_k$ are distinct but some s_i may be the same as a t_j for $i \neq j$. Intuitively speaking, k-matching extension is a graph which lies between a matching and DIR-SKEW-MULTICUT demand pattern.

Theorem 1.3. *Consider* DIR-MULTICUT *where the demand graph does not contain an induced k-matching-extension. Then the flow-cut gap is at most* $k-1$ *and there is a polynomial-time rounding algorithm that achieves this upper bound. Thus, there exists a* $(k-1)$*-approximation algorithm for* DIR-MULTICUT *where the demand graph does not contain an induced k-matching extension.*

Demand graph of DIR-MWC does not contain an induced 3-matching-extension. Hence, this result can be seen as a generalization of 2-approximation algorithm for DIR-MWC to a larger class of demand graphs.

(iii) Hardness vs Flow-Cut gap

Next, we investigate the complexity of DIR-MULTICUT with *fixed demand graph*. To be formal we need to define H as a "pattern" since we need to specify the nodes of G to which the nodes of H are mapped. However, we avoid further notation and assume that $V_H \subset V_G$. For a *fixed* demand graph H, we define the problem DIR-MULTICUT-H as the special case of DIR-MULTICUT where G is arbitrary but the demand graph is constrained to be H. We define α_H to be the worst-case flow-cut gap over all instances with demand graph H. We conjecture the following general hardness of approximation result:

Conjecture 1.2. *For* any fixed *demand graph H and any fixed* $\epsilon > 0$*, unless* $P = NP$*, there is no polynomial-time* $(\alpha_H - \epsilon)$*- approximation for* DIR-MULTICUT-H.

In Chapter 2, we prove weaker forms of the conjecture, captured in the following two theorems:

Theorem 1.4. *Assuming UGC, for any fixed directed bipartite graph H, and for any fixed $\epsilon > 0$ there is no polynomial-time $(\alpha_H - \epsilon)$-approximation for* DIR-MULTICUT-H.

Theorem 1.5. *Assuming UGC, for any fixed directed graph H on k vertices and for any fixed $\epsilon > 0$, there is no polynomial-time $(\frac{\alpha_H}{2\lceil \log k \rceil} - \epsilon)$-approximation for* DIR-MULTICUT-H.

Recall the SKEW-MULTICUT problem discussed earlier with bi-partite demand graph.

DIR-SKEW-MULTICUT: The input is a directed graph $G = (V, E)$ and a set of vertices $s_1, \ldots, s_k, t_1, \ldots, t_k \in V$ along with non-negative edge weights $w(e), e \in E$. The goal is to find minimum weight set of edges $E' \subset E$ such that for every $1 \leq i \leq j \leq k$, there is no path from s_i to t_j in $G - E'$.

Via known flow-cut gap results [74], we obtain the following corollary.

Corollary 1.3. *Assuming UGC, for any fixed $\epsilon > 0$ the following hold.*

- *For any fixed k, if H is a collection of k disjoint directed edges then* DIR-MULTICUT *with demand graph H is hard to approximate within a factor of $k - \epsilon$.*
- *For any fixed k,* DIR-SKEW-MULTICUT*'s approximability coincides with its flow-cut gap.*

1.3.2 Linear-3-Cut

In Chapter 3, we consider a special case of DIR-MULTICUT, namely Linear-3-cut and design an approximation algorithm which is optimal assuming UGC. It is also a special case of the DIR-SKEW-MULTICUT problem discussed in Chapter 2.

(s, r, t)-EDGE-LIN-3-CUT: The input is a directed graph $G = (V, E)$ with specified nodes $s, r, t \in V$ along with non-negative edge weights $w(e), e \in E$, and the goal is to find a minimum weight edge set $E' \subseteq E$ such that $G - E'$ has no path from s to t, from s to r, and from r to t.

(s, r, t)-EDGE-LIN-3-CUT is equivalent to DIR-SKEW-MULTICUT with $k = 2$. To reduce (s, r, t)-EDGE-LIN-3-CUT to DIR-SKEW-MULTICUT, set $s_1 = s, s_2 = r, t_1 = r, t_2 = t$. To reduce DIR-SKEW-MULTICUT to (s, r, t)-EDGE-LIN-3-CUT, set $s = s_1, t = t_2$, add a node r and edges $(t_1, r), (r, s_2)$.

Both the approximation algorithm and the hardness result for (s, r, t)-EDGE-LIN-3-CUT are proved by showing tight bounds on the flow-cut gap.

Theorem 1.6. *Flow-cut gap for (s, r, t)-*EDGE-LIN-3-CUT *is at most $\sqrt{2}$ and there is a polynomial-time rounding algorithm that achieves this upper bound. For any constant $\delta > 0$, there exists a (s, r, t)-*EDGE-LIN-3-CUT *instance with flow-cut gap at least $\sqrt{2} - \delta$.*

Equivalence of (s, r, t)-EDGE-LIN-3-CUT and DIR-SKEW-MULTICUT with $k = 2$ and Theorem 1.4 implies the following corollary:

Corollary 1.4. *There is a polynomial-time $\sqrt{2}$-approximation for* (s,r,t)-EDGE-LIN-3-CUT. *Assuming UGC,* (s,r,t)-EDGE-LIN-3-CUT *has no polynomial-time* $(\sqrt{2}-\epsilon)$*-approximation algorithm for any constant* $\epsilon > 0$.

One of the motivations to consider (s,r,t)-EDGE-LIN-3-CUT arose from a closely related problem, abbreviated r-INOUT-NODE-BLOCKER. An *out-r-arborescence* (similarly, an *in-r-arborescence*) in a directed graph is a minimal subset of edges such that every node has a unique path from r (to r) in the subgraph induced by the edges. The smallest number of edges/nodes whose removal ensures that the graph has no arborescence holds the key to understanding reliability in networks. Computing this number is also a special case of the interdiction problem of covering bases of two *matroids* [11]. We recall that the problem of finding a minimum weight subset of edges or nodes whose deletion ensures that the remaining graph has no out-r-arborescence for a specified node r can be solved efficiently (by reducing to min $u \to v$ cut in directed graphs). We are interested in the case where remaining graph has no out-r-arborescence *and* no in-r-arborescence.

r-INOUT-NODE-BLOCKER: The input is a directed graph $G = (V, E)$ with a specified terminal node $r \in V$ along with non-negative node weights $w(v), v \in V$, and the goal is to find a minimum weight node set $V' \subseteq V$ such that $G - V'$ has no out-r-arborescence *and* no in-r-arborescence.

We show an approximation-preserving equivalence between r-INOUT-NODE-BLOCKER and (s,r,t)-EDGE-LIN-3-CUT which in turn, resolves the approximability of r-INOUT-NODE-BLOCKER.

Theorem 1.7. *There is a polynomial-time $\sqrt{2}$-approximation for* r-INOUT-NODE-BLOCKER. *Assuming UGC,* r-INOUT-NODE-BLOCKER *has no polynomial-time* $(\sqrt{2}-\epsilon)$*-approximation for any constant* $\epsilon > 0$.

Another motivation to consider (s,r,t)-EDGE-LIN-3-CUT arose from its connection to a global bi-cut problem, namely $\{s,*\}$-EDGE-BICUT.

$\{s,*\}$-EDGE-BICUT: The input is an edge-weighted directed graph with a specified node s, and the goal is to find a smallest weight subset of edges whose deletion ensures that the resulting graph has a node t such that s cannot reach t and t cannot reach s.

It is known that $\{s,*\}$-EDGE-BICUT is NP-hard, admits an efficient 2-approximation algorithm, and does not admit an efficient $(4/3-\epsilon)$-approximation for any constant $\epsilon > 0$ assuming UGC [9]. In Chapter 3, we show a reduction from (s,r,t)-EDGE-LIN-3-CUT to $\{s,*\}$-EDGE-BICUT, thus improving the hardness known for $\{s,*\}$-EDGE-BICUT.

Theorem 1.8. *Assuming UGC,* $\{s,*\}$-EDGE-BICUT *has no polynomial-time* $(\sqrt{2}-\epsilon)$*-approximation for any constant* $\epsilon > 0$.

1.3.3 Multiway Cut

As we mentioned earlier, we consider the directed multiway cut problem in Chapter 2 and show a polynomial time 2-approximation algorithm. In Chapter 4, we consider the undirected variant of the problem and a special case of the directed problem and improve integrality gap bounds of various LPs associated with these problems.

From the result of Zosin *et al.* [70] and Theorem 1.2, we know that the flow-cut gap for directed multiway cut (DIR-MWC) is 2. Hence, DIR-MWC admits a 2-approximation algorithm and does not admit a $(2-\epsilon)$-approximation algorithm for any $\epsilon > 0$ assuming UGC [61]. However, these results do not provide tight approximation ratio for DIR-MWC with fixed k (number of terminals), in particular for $k = 2$.

$\{s,t\}$-EDGE-BICUT: The input is a directed graph $G = (V, E)$ and two vertices $s, t \in V$ along with non-negative edge weights $w(e), e \in E$. The goal is to find minimum weight set of edges $E' \subseteq E$ such that there is no path from s to t *and* no path from t to s in $G - E'$.

In Chapter 4, we show that the flow-cut gap is 2 even for $\{s,t\}$-EDGE-BICUT.

Theorem 1.9. *Flow-cut gap for* $\{s,t\}$-EDGE-BICUT *is* 2 *even in directed planar graphs.*

Corollary 1.5. *Assuming UGC,* $\{s,t\}$-EDGE-BICUT *has no polynomial-time* $(2-\epsilon)$*-approximation algorithm for any constant* $\epsilon > 0$.

Next, we consider the undirected edge-weighted multiway cut (EDGE-WT-MWC).

EDGE-WT-MWC: The input is an undirected graph $G = (V, E)$ and a set of terminals $s_1, \ldots, s_k \in V$ along with non-negative edge weights $w(e), e \in E$. The goal is to find minimum weight set of edges $E' \subseteq E$ such that for $i \neq j \in [1, k]$, there is no path from s_i to s_j in $G - E'$.

Manokaran *et al.* [68] showed that the integrality gap of CKR relaxation gives a matching hardness of approximation bound for EDGE-WT-MWC assuming UGC. Călinescu, Karloff and Rabani introduced the relaxation and designed a rounding scheme which led to a $(3/2-1/k)$-approximation [15]. For k=3, Karger et al. [56] and Cunningham et al. [25] designed an alternative rounding scheme that led to a 12/11-approximation factor and also exhibited a matching integrality gap instance. After the results by Karger et al. and Cunningham et al., a rich variety of rounding techniques were developed to improve the approximation factor for $k \geq 4$ [12, 13, 77]. The current best approximation factor for EDGE-WT-MWC is 1.2965 due to Sharma and Vondrák [77].

From the result of Karger *et al.* and Cunningham et al., we know that the 12/11-factor integrality gap is tight for EDGE-WT-MWC with k=3. Freund and Karloff [44] constructed a class of instances showing a lower bound of $8/(7 + (1/(k-1)))$ on the integrality gap. Last year, Angelidakis, Makarychev and Manurangsi [3] gave a remarkably simple construction showing an integrality gap of 6/5. A technical challenge in improving the gap has been the lack of geometric tools to understand higher-dimensional simplices. In Chapter 4, we address this challenge by constructing a non-trivial 3-dimensional instance and improving the gap.

Theorem 1.10. *For every constant* $\delta > 0$*, there exists an instance of* EDGE-WT-MWC *such that the integrality gap of the CKR relaxation for that instance is at least* $1.20016 - \delta$.

We analyze the gap of the instance by viewing it as a convex combination of 2-dimensional instances and a uniform 3-dimensional instance. We believe that this technique could be exploited further to construct instances with larger integrality gap. One of the byproducts from our proof technique is a generalization of a result on *Sperner admissible labelings* due to Mirzakhani and Vondrák [69] that might be of independent combinatorial interest. The above result in conjunction with the result of Manokaran et al. immediately implies the following corollary:

Corollary 1.6. *Assuming UGC,* EDGE-WT-MWC *has no polynomial-time* $(1.20016 - \epsilon)$*-approximation algorithm for any constant* $\epsilon > 0$.

1.3.4 LP formulations for Subset Feedback Set

In Chapter 5, we consider the subset feedback set problems.

SUBSET-FVS : Input is an undirected graph $G = (V, E)$ along with non-negative node weights $w(v), v \in V$, and a set $S = \{s_1, \ldots, s_k\} \subset V$ of terminals. A cycle is interesting if it contains a terminal. The goal is to find a minimum weight set of nodes $V' \subset V$ that intersect all interesting cycles.

SUBSET-FES : Input is an undirected graph $G = (V, E)$ along with a non-negative edge weights $w(e), e \in E$, and a set $S = \{s_1, \ldots, s_k\} \subset V$ of terminals. A cycle is interesting if it contains a terminal. The goal is to find a minimum weight set of edges $E' \subset E$ that intersect all interesting cycles.

As we discussed earlier in Section 1.2.2, constant factor approximation algorithms known for SUBSET-FES and SUBSET-FVS are largely combinatorial and the optimal algorithms known for related problems are based on LP formulations. And, this explains why one needs LP-type techniques for SUBSET-FES and SUBSET-FVS. In Chapter 5, we provide such LP relaxations for SUBSET-FES and SUBSET-FVS with constant integrality gap.

Theorem 1.11. *There are polynomial-sized integer programming formulations for* SUBSET-FES *and* SUBSET-FVS *whose linear programming relaxations have an integrality gap of at most* 13.

The approximation bound of 13 that we are able to establish is weaker than the existing approximation ratios for the problems. However, we do not know of an integrality gap worse than the hardness bounds 1.20016 and 2 known for SUBSET-FES and SUBSET-FVS respectfully. We believe that related formulations and ideas would lead to improved algorithms for SUBSET-FES and SUBSET-FVS.

1.4 Credits

Most of the results mentioned in this thesis have appeared in different published works.

- The 2-approximation algorithm for DIR-MWC in Section 2.3.1 and the flow-cut gap bound of 2 for $\{s, t\}$-EDGE-BICUT in Section 4.2 are based on a joint work with Chandra Chekuri [23].

- The 2-approximation algorithm for UNDIR-MULTICUT with tK_2-free demand graphs in Section 2.2, $(k-1)$-approximation algorithm for DIR-MULTICUT with k-matching-extension free demand graphs in Section 2.3.2, and the hardness results for DIR-MULTICUT in Section 2.4 are based on a joint work with Chandra Chekuri [24].
- Optimal algorithms for (s,r,t)-EDGE-LIN-3-CUT and related problems in Chapter 3 are based on a joint work with Kristóf Bérczi, Karthekeyan Chandrasekaran, and Tamás Király [10].
- Improvement in the integrality gap of CKR relaxation for EDGE-WT-MWC in Section 4.1 is based on a joint work with Kristóf Bérczi, Karthekeyan Chandrasekaran, and Tamás Király and is under submission.
- LP-based constant factor approximation algorithm for feedback set problems in Chapter 5 is based on a joint work with Chandra Chekuri [22].

Chapter 2

Influence of the demand graph on the approximability of Multicut

The minimum MULTICUT problem is a generalization of the classical s-t cut problem to multiple pairs. The input to the MULTICUT problem is an edge-weighted graph $G = (V, E)$ and k source-sink pairs $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$. The goal is to find a minimum weight subset of edges $E' \subseteq E$ such that all the given pairs are disconnected in $G - E'$; that is, for $1 \leq i \leq k$, there is no path from s_i to t_i in $G - E'$. In this chapter we consider an equivalent formulation that exposes, more directly, the structure that the source-sink pairs may have.

UNDIR-MULTICUT: The input is an undirected *supply* graph $G = (V_G, E)$ and an undirected *demand* graph $H = (V_H, F)$, $V_H \subseteq V_G$ along with non-negative edge weights $w(e), e \in E$. The goal is to find minimum weight set of edges $E' \subseteq E$ such that for each edge $f = (s, t) \in F$, there is no path from s to t in $G - E'$.

DIR-MULTICUT: The input is a directed *supply* graph $G = (V_G, E)$ and a directed *demand* graph $H = (V_H, F)$, $V_H \subseteq V_G$ along with non-negative edge weights $w(e), e \in E$. The goal is to find minimum weight set of edges $E' \subseteq E$ such that for each edge $f = (s, t) \in F$, there is no path from s to t in $G - E'$.

UNDIR-MULTICUT and DIR-MULTICUT are NP-Hard even in very restricted settings. For instance, UNDIR-MULTICUT is NP-Hard even when H has 3 edges and it generalizes VERTEX COVER even when G is a tree. DIR-MULTICUT is NP-Hard and APX-Hard even in the special case when H is a cycle of length 2 which is better understood as removing a minimum weight set of edges to disconnect s from t *and* t from s in a directed graph. We use k to denote the number of edges in the demand graph H. For UNDIR-MULTICUT there is an $O(\log k)$-approximation [47] which improves to an $O(r)$-approximation if the supply graph G excludes K_r as a minor [1, 37, 62] (in particular this yields a constant factor approximation in planar graphs). In terms of inapproximability, UNDIR-MULTICUT is at least as hard as VERTEX COVER even in trees and hence APX-Hard. Under the Unique Game Conjecture (UGC) it is known to be super-constant hard [16]. DIR-MULTICUT is known to be a harder problem. Assuming $NP \neq ZPP$ it is hard to approximate DIR-MULTICUT to within a factor of $\Omega(2^{\log^{1-\epsilon} n})$ [27]; evidence is also presented in [27] that it could be hard to approximate to within an $\Omega(n^{\delta})$ factor for some fixed $\delta > 0$. The best-known approximation is $\min\{k, \tilde{O}(n^{11/23})\}$ [2]; here $n = |V|$. Note that a k-approximation is trivial.

We note that all the preceding positive results for MULTICUT are based on bounding the integrality gap of a natural LP relaxation shown in Figure 2.1. This is the standard cut formulation with a variable

$$
\begin{array}{llll}
\min & \sum_{e \in E} w_e x_e & & \\
 & \sum_{e \in p} x_e & \geq 1 & p \in \mathcal{P}_{st}, st \in F \\
 & x_e & \geq 0 & e \in E
\end{array}
$$

$$
\begin{array}{llll}
\min & \sum_{e \in E} w_e x_e & & \\
 & d(s,t) & \geq 1 & st \in F \\
 & d(u,v) + d(v,w) - d(u,w) & \geq 0 & u,v,w \in V \\
 & d(u,v) - x_e & = 0 & e = uv \in E \\
 & d(u,v) & \geq 0 & u,v \in V
\end{array}
$$

Figure 2.1: DISTANCE-LP for Multicut written using exponential number of constraints (top) and in a compact fashion using additional variables (bottom)

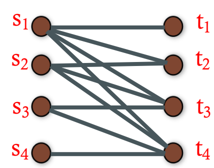


Figure 2.2: Demand graph for SKEW-MULTICUT

x_e for each edge e and an exponential set of constraints; the constraints specify that for each demand edge $st \in F$ the length of any path p from s to t is at least 1. One can solve this LP in polynomial-time using the Ellipsoid method. One can also write a compact polynomial-sized formulation using distance variables and it is shown in the same figure. The dual is a maximum multicommodity flow LP. We henceforth refer to the integrality gap of this LP as the *flow-cut gap* and the LP as DISTANCE-LP. Most multicut approximation algorithms are based on bounding the flow-cut gap.

Restricted demand graphs: Our preceding discussion has focused on the approximability of MULTICUT when H is arbitrary, with some improved results when G is restricted. Next, we mention results in the setting where H is restricted and G is arbitrary. A well-known such problem is the multiway cut problem (MULTIWAY CUT). Undirected edge-weighted multiway cut (EDGE-WT-MWC) admits a 1.29 approximation [77] and directed edge-weighted multiway cut (DIR-MWC) admits a 2-approximation [23, 70]. In a recent work [19], motivated by connections to the problem of understanding the information capacity of networks with delay constraints, the following special case of MULTICUT, called SKEW-MULTICUT was considered. The demand graph H is a bipartite graph with k terminals $s_1, \ldots, s_k$ on one side and k terminals $t_1, \ldots, t_k$ on the other side: (s_i, t_j) is an edge in H iff $i \leq j$. See Figure 2.2 for an example with $k = 4$. If the edges are directed from s's to the t's we obtain the directed version. It was established that the flow-cut gap for DIR-SKEW-MULTICUT is $O(\log k)$ [19] which is in contrast to the general setting where the gap can be as large as k. One can show APX-hardness and a constant factor lower bound on the flow-cut gap for UNDIR-SKEW-MULTICUT (and hence also for DIR-SKEW-MULTICUT) via a reduction from EDGE-WT-MWC problem.

Some other work has also examined the role that the demand graph plays in MULTICUT. Two examples are the original paper of Garg, Vazirani and Yannakakis [47] who showed that one can obtain an $O(\log h)$-approximation for UNDIR-MULTICUT where h is the vertex cover size of the demand graph. This was generalized by Steurer and Vishnoi [80] who showed that h can be chosen to be $\min_S \max_T |S \cap T|$ where S is a vertex cover in H and T is an independent set in H. Note that both these results are based on the flow-cut gap and yield only an $O(\log k)$ upper bound for UNDIR-SKEW-MULTICUT.

Algorithmic results

UNDIR-MULTICUT: We obtain a 2-approximation for a class of demand graphs. A graph is said to be tK_2-free for an integer $t > 1$ if it does not contain a matching of size t as an induced subgraph.

Theorem 2.1. *There is a* 2*-approximation algorithm with running time poly*$(n, k^{O(t)})$ *on instances of* UNDIR-MULTICUT *with supply graph G and tK_2-free demand graph H. Here $n = |V_G|$ and $k = |V_H|$.*

Since demand graph in UNDIR-SKEW-MULTICUT instances are $2K_2$-free we obtain the following corollary.

Corollary 2.1. UNDIR-SKEW-MULTICUT *admits a polynomial-time* 2*-approximation.*

Also, if $h = \min_S \max_T |S \cap T|$ where S is a vertex cover and T is an independent set in H, then H is tK_2-free for $t = h + 1$. So, we obtain the following corollary:

Corollary 2.2. *Consider* UNDIR-MULTICUT *with supply graph G and demand graph H such that $n = |V_G|, k = |V_H|$. Let $h = \min_S \max_T |S \cap T|$ where S is a vertex cover and T is an independent set in H. Then, there exists a* 2*-approximation algorithm with running time poly*$(n, k^{O(h)})$.

DIR-MULTICUT: Our first result simplifies the 2-approximation algorithm for DIR-MWC by Naor and Zosin [70]

Theorem 2.2. *There is a randomized algorithm that given a feasible solution $\boldsymbol{x}$ to* DISTANCE-LP *for* DIR-MWC*, returns a feasible integral solution of expected cost at most $2\sum_e w_e x_e$, and runs in $O(m + n \log n)$ time. The algorithm can be derandomized to yield a deterministic* 2*-approximation algorithm that runs in $O(m \log n)$ time. Here, $m = |E(G)|, n = |V(G)|$.*

Our second result bounds the flow-cut gap of DIR-MULTICUT with k-matching extension free demand graphs. We say that a directed demand graph $H = (V, F)$ contains an induced k-matching-extension if there are two subsets of V, $S = \{s_1, s_2, \ldots, s_k\}$ and $T = \{t_1, t_2, \ldots, t_k\}$ satisfying the following properties: (i) for $1 \le i \le k$, $(s_i, t_i) \in F$ and (ii) for $i > j$, $(s_i, t_j) \notin F$. Note that $s_1, s_2, \ldots, s_k$ are distinct since S is a set and similarly $t_1, \ldots, t_k$ are distinct but some s_i may be the same as a t_j for $i \neq j$. Intuitively speaking, k-matching extension is a graph which lies between a matching and DIR-SKEW-MULTICUT demand pattern.

Theorem 2.3. *Consider* DIR-MULTICUT *where the demand graph does not contain an induced k-matching-extension. Then the flow-cut gap is at most $k-1$ and there is a polynomial-time rounding algorithm that achieves this upper bound.*

Demand graph of DIR-MWC (complete bi-directed graph) does not contain an induced 3-matching-extension. Hence, 2-approximation for DIR-MWC can be seen as a special case of this result.

Hardness results

DIR-MULTICUT: In Section 2.4, we investigate the hardness for a fixed demand graph. To be formal we need to define H as a “pattern” since we need to specify the nodes of G to which the nodes of H are mapped. However, we avoid further notation and assume that $V_H \subset V_G$. For a *fixed* demand graph H, we define the problem DIR-MULTICUT-H as the special case of DIR-MULTICUT where G is arbitrary but the demand graph is constrained to be H. We define α_H to be the worst-case flow-cut gap over all instances with demand graph H. We conjecture the following general hardness of approximation result.

Conjecture 2.1. *For* any fixed *demand graph H and any fixed $\varepsilon > 0$, unless $P = NP$, there is no polynomial-time $(\alpha_H - \varepsilon)$- approximation for* DIR-MULTICUT-H.

In this thesis we prove weaker forms of the conjecture, captured in the following two theorems:

Theorem 2.4. *Assuming UGC, for any fixed directed bipartite graph H, and for any fixed $\varepsilon > 0$ there is no polynomial-time $(\alpha_H - \varepsilon)$ approximation for* DIR-MULTICUT-H.

Theorem 2.5. *Assuming UGC, for any fixed directed graph H on k vertices and for any fixed $\varepsilon > 0$, there is no polynomial-time $\frac{\alpha_H}{2\lceil \log k \rceil} - \varepsilon$ approximation for* DIR-MULTICUT-H.

Via known flow-cut gap results [74] and some standard reductions we obtain the following corollary.

Corollary 2.3. *Assuming UGC, for any fixed $\varepsilon > 0$ the following hold.*

- *For any fixed k, if H is a collection of k disjoint directed edges then* DIR-MULTICUT-H *is hard to approximate within a factor of $k - \varepsilon$.*
- *For any fixed k,* DIR-SKEW-MULTICUT*’s approximability coincides with its flow-cut gap.*

We will further see the use of Theorem 2.11 in Chapters 3 and 4 to derive other hardness results.

Organization: Section 2.2 describes the factor 2-approximation for UNDIR-MULTICUT with tK_2-free demand graph. Section 2.3 describes the 2-approximation for DIR-MWC and $(k-1)$ approximation for DIR-MULTICUT-H when H does not contain an induced k-matching extension. Section 2.4 describes the hardness of approximation results for DIR-MULTICUT-H.

2.1 Label LP

At a high-level, proof of Theorems 2.6, 2.11 and 2.5 are based on a *labeling* viewpoint for MULTICUT. Consider the supply graph $G = (V_G, E)$, demand graph $H = (V_H, F)$ and the optimum solution $X \subset E$. Let $V_H = \{s_1, \ldots, s_\ell\}$. We label the vertices of V_G by a label in $\{0,1\}^\ell$ based on its reachability from s_i's in $G - X$. For $u \in V_G, \sigma \in \{0,1\}^\ell$, let $x(u,\sigma)$ be a variable which is 1 if u is labeled σ and zero otherwise. Using these variables, we write the following constraints in a linear fashion: (i) If $(s_i, s_j) \in F$, then s_j is not reachable from s_i in $G - X$ (ii) For $(u,v) \in E, i \in [1,k]$, if u is reachable by s_i and v is not reachable by s_i in $G - X$, then $(u,v) \in X$. Relaxing the variables to be real valued and adding some non-negativity constraints gives us label-lp for MULTICUT.

For undirected graphs we show that this yields Theorem 2.6. Number of labels can be reduced to $poly(k^t)$ for tK_2-free demand graphs where $k = |V_H|$. This turns out to be very similar to the earthmover LP for uniform metric labeling [21]. Hence, we present a 2-approximation via a simple reduction to uniform metric labeling and using rounding scheme by Kleinberg and Tardos to round the solution of the earthmover LP [63]. In directed graphs we show that a labeling based LP is no more powerful than DISTANCE-LP which is stark contrast to the undirected graph setting. The labeling LP allows us to relate the hardness of DIR-MULTICUT-H to the hardness of a min constraint satisfaction problems (Min CSPs) via a standard labeling LP for CSPs called BASIC-LP. We crucially rely on a general hardness result for MIN-β-CSP due to Ene, Vondrak and Wu [31] that generalized prior work of Manokaran *et al.* [68].

2.2 Approximation algorithms for UNDIR-MULTICUT

In this section we obtain a 2-approximation for tK_2-free demand graphs and prove Theorem 2.6 which is restated below.

Theorem 2.6. *There is a* 2*-approximation algorithm with running time poly*$(n, k^{O(t)})$ *on instances of* UNDIR-MULTICUT *with supply graph G and tK_2-free demand graph H. Here $n = |V_G|$ and $k = |V_H|$.*

Before we prove the theorem, we consider the UNDIR-MULTICUT problem where the demand graph has k vertices. Given supply graph $G = (V,E)$ let $S = \{s_1, \ldots, s_k\} \subset V$ be the *terminals* participating in the demand edges specified by H. A feasible solution $E' \subset E_G$ of the UNDIR-MULTICUT instance will induce a partition over S such that if $s_i s_j$ is an edge in the demand graph H, then s_i and s_j belong to different components in $G - E'$. Note that two terminals that are not connected by a demand edge may be in the same connected component of $G - E'$. If k is a fixed constant we can "guess" the partition of the terminals induced by an optimum solution. With the guess in place it is easy to see that the problem reduces to an instance of EDGE-WT-MWC which admits a constant factor approximation. Thus, one can obtain a constant factor approximation for UNDIR-MULTICUT in $2^{O(k \log k)}\text{poly}(n)$ time by trying all possible partitions of the terminals.

To prove Theorem 2.6, we use this idea of enumerating feasible partitions. However, H is not necessarily of fixed size, and enumerating all possible partitions of the terminals is not feasible. Instead, we make use of the following theorem which bounds the number of *maximal* independent sets in a tK_2-free graph.

Theorem 2.7. *(Balas and Yu [6]) Any s-vertex tK_2-free graph has at most $s^{O(t)}$ maximal independent sets and these can be found in $s^{O(t)}$ time.*

We prove Theorem 2.6 by using the preceding theorem and reducing the UNDIR-MULTICUT problem to the UNIFORM-METRICLABELING problem. We now describe the general METRICLABELING problem.

METRICLABELING: The input consists of an undirected edge-weighted graph $G = (V, E)$, a set of labels $L = \{1, \dots, h\}$ and a metric $d(i, j), i, j \in L$ defined over the labels. In addition, for each vertex $u \in V$ and label $i \in L$ there is a non-negative assignment cost $c(u, i)$. Given an assignment $f : V \to L$ of vertices to labels we define its cost as $\sum_{u \in V} c(u, f(u)) + \sum_{uv \in E} w(uv) d(f(u), f(v))$. The goal is to find an assignment of minimum cost. The special case when the metric is uniform, that is $d(i, j) = 1$ for $i \neq j$, is referred to as UNIFORM-METRICLABELING.

Theorem 2.8. *(Kleinberg and Tardos [63]) There is a 2-approximation algorithm for* UNIFORM-METRIC-LABELING.

The algorithm for UNIFORM-METRICLABELING in [63] is based on an interactive rounding scheme for a solution to the earthmover LP relaxation of the problem.

Proof: [Theorem 2.6] Let the demand graph H of the UNDIR-MULTICUT instance be tK_2-free. Using Theorem 2.7, we can find all maximal independent sets in H. Let these independent sets be $I_1, \dots, I_r$ where $r \leq |V_H|^{O(t)}$. Note that the independent sets are considered only in the demand graph. Note that instance also need to specify the mapping of vertices of H to vertices of G. However, for ease of notation we will simply assume that $V_H \subset V_G$.

Consider the following instance of UNIFORM-METRICLABELING: The supply graph $G = (V, E)$ of the UNDIR-MULTICUT instance is the input graph to the UNIFORM-METRICLABELING instance. The label set $L = \{1, 2, \dots, r\}$, one for each maximal independent set in H. For each $u \in V_H$ let $c(u, i) = 0$ if $u \in I_i$ and $c(u, i) = \infty$ otherwise. And for each $u \in V \setminus V_H$, $c(u, i) = 0$ for all i.

We claim that the preceding reduction is approximation preserving. Assuming the claim, we can obtain the desired 2-approximation by solving the UNIFORM-METRICLABELING instance using Theorem 2.8.

The size of the UNIFORM-METRICLABELING instance that is generated from the given UNDIR-MULTICUT instance is poly($n, |V_H|^{O(t)}$) which explains the running time. We now prove the claim.

Let $f : V \to L$ be an assignment of labels to the nodes whose cost is finite (such an assignment always exists since each terminal is in some independent set). Let $E' \subset E$ be the set of edges "cut" by this assignment; that is, $uv \in E'$ iff $f(u) \neq f(v)$. The cost of this assignment is equal to the weight of E' since the metric is uniform, and the labeling costs are 0 or ∞. We argue that E' is a feasible solution for the UNDIR-MULTICUT instance. Suppose not. Then there are terminals u, v such that uv is an edge in the demand graph H and u, v belong to the same connected component of $G - E'$. The label $j = f(u)$ corresponds to a maximal independent set I_j in H which means that $v \notin I_j$. Thus $f(v) \neq j$ since $c(v, j) = \infty$. Therefore, u and v are assigned different labels and cannot be in the same connected component.

Conversely, let $E' \subset E$ be a feasible solution for UNDIR-MULTICUT instance and let $V_1, \ldots, V_\ell$ be vertex sets of the connected components of $G - E'$. Let T_j be the terminals in V_j. Since, all pairs of terminals connected by an edge in H are separated in $G - E'$, T_j must be an independent set in H. For each T_j, consider a maximal independent set in H containing all the vertices of T_j; pick arbitrary one if more than one exists. Let this independent set be I_{i_j}. We construct a labeling f by labeling all vertices of V_j by label i_j. It is easy to see that all terminals are assigned a label corresponding to an independent set in H containing that terminal. Hence, labeling cost is equal to zero. Also, all vertices corresponding to same connected component in $G - E'$ are assigned the same label. Hence, cost of the edges cut by the assignment f is at most the cost of the edges in E'.

2.3 Approximation algorithms for DIR-MULTICUT

2.3.1 2-approximation for DIR-MWC

Consider the DISTANCE-LP for DIR-MWC as shown in figure 4.1 and denoted as DIR-MWC-REL. The main result of the section is the following theorem.

Theorem 2.9. *There is a randomized algorithm that given a feasible solution* $\mathbf{x}$ *to* DIR-MWC-REL *returns a feasible integral solution of expected cost at most* $2\sum_e w_e x_e$*, and runs in* $O(m + n \log n)$ *time. The algorithm can be derandomized to yield a deterministic* 2*-approximation algorithm that runs in* $O(m \log n)$ *time. Here,* $m = |E(G)|, n = |V(G)|$.

We describe the simple randomized ball-cutting algorithm that achieves the properties claimed by the theorem. Let $\mathbf{x}$ be a feasible solution to DIR-MWC-REL. For any two nodes $u, v \in V$ we define $d_x(u, v)$ be the shortest path length from u to v using edge lengths given by $\mathbf{x}$. For notational simplicity we omit the subscript x since there is little chance of confusion. The algorithm adds new nodes $t_1, t_2, \ldots, t_k$ and adds the edge set $\{(t_i, s_j) \mid i \neq j\}$ and sets the x value of each of these new edges to 0. Note that, this is in effect a reduction of the DIR-MWC for the given instance to a DIR-MULTICUT instance which requires

$$
\begin{array}{llll}
 & \textbf{Dir-MWC-Rel} & & \\
\min & \sum_{e \in E} w_e x_e & & \\
 & \sum_{e \in p} x_e & \geq 1 & p \in \mathcal{P}_{ij}, i \neq j \\
 & x_e & \geq 0 & e \in E
\end{array}
$$

Figure 2.3: LP Relaxation for Dir-MWC

us to separate the pairs (t_i, s_i), $1 \leq i \leq k$. The solution $\mathbf{x}$ augmented with the extra nodes and edges leads to a feasible fractional solution for this Dir-Multicut instance. Our algorithm, formally described below, is very simple. We pick a random $\theta \in (0,1)$ and take the union of the cuts defined by balls of radius θ around each t_i. More formally let $B(v, r)$ be the set of all nodes at distance at most r from v. Then the algorithm simply outputs $\bigcup_{i=1}^{k} \delta^+(B(t_i, \theta))$ where $\delta^+(A)$ denote the set of outgoing edges from A.

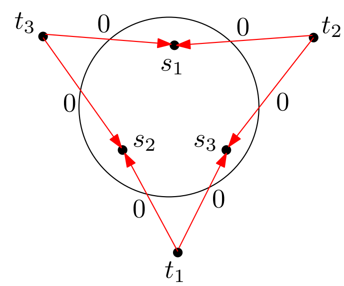


Figure 2.4: Addition of dummy vertices and edges

Algorithm 2.1 Rounding for Dir-MWC

1: Given a feasible solution $\mathbf{x}$ to Dir-MWC-Rel
2: Add new vertices $t_1, \ldots, t_k$, edges (t_i, s_j) for all $i \neq j$ and set $x(t_i, s_j) = 0$
3: Pick $\theta \in (0,1)$ uniformly at random
4: $C = \cup_{i=1}^{k} \delta^+(B(t_i, \theta))$
5: Return C

Note that C is a random set of edges that depends on the choice of θ. We denote by $C(\theta)$ the set of edges output by the algorithm for a given θ.

Lemma 2.1. *If $\mathbf{x}$ is a feasible fractional solution to* Dir-MWC-Rel, *$C(\theta)$ is a feasible multiway cut for $\{s_1, \ldots, s_k\}$ for any $\theta \in (0,1)$. Thus, Algorithm 2.1 always returns a feasible integral solution given a feasible $\mathbf{x}$.*

Proof: Fix any $i \in \{1, \ldots, k\}$ and $\theta \in (0,1)$. Since $d(t_i, s_j) = 0$ for all $j \neq i$, we have that $s_j \in B(t_i, \theta)$ for all $j \neq i$. Moreover, by feasibility of $\mathbf{x}$, we have $d(t_i, s_i) \geq 1$ for otherwise there will be a path of length less than 1 from some s_j to s_i where $j \neq i$. Therefore $s_i \notin B(t_i, \theta)$ because $\theta < 1$. Therefore, $G - \delta^+(B(t_i, \theta))$ has no path from s_j to s_i for any $j \neq i$. Since $C(\theta) = \bigcup_i \delta^+(B(t_i, \theta))$, it follows that there is no path in $G - C(\theta)$ from s_j to s_i for any $j \neq i$.

We now bound the probability that any fixed edge e is cut by the algorithm, that is, $\Pr[e \in C]$. Note that e may be *simultaneously* cut by several t_i for the same value of θ but we are only interested in the probability that it is included in C.

Lemma 2.2. *For any edge $e \in E$, $\Pr[e \in C] \leq 2x_e$.*

Proof: Let $e = (u, v)$. Rename the terminals such that $d(s_1, u) \leq d(s_2, u) \leq \cdots \leq d(s_k, u)$. This implies that

$$d(t_1, u) = d(s_2, u) \tag{2.1}$$

and

$$d(t_2, u) = d(t_3, u) = \ldots = d(t_k, u) = d(s_1, u). \tag{2.2}$$

Edge $e \in \delta^+(B(t_i, \theta))$ if and only if $\theta \in [d(t_i, u), d(t_i, v))$; we have that $d(t_i, v) \leq d(t_i, u) + x_e$. Defining the interval I_i as $[d(t_i, u), d(t_i, u) + x_e)$, we see that $e \in \delta^+(B(t_i, \theta))$ only if $\theta \in I_i$. However, from the property that $d(t_2, u) = d(t_3, u) \ldots = d(t_k, u)$, $I_2 = I_3 = \ldots = I_k$. Thus, $e \in C$ only if $\theta \in I_1$ or $\theta \in I_2$ and since $|I_1|$ and $|I_2|$ are both at most x_e long and θ is chosen uniformly at random from $(0, 1)$,

$$\Pr[e \in C] \leq \Pr[\theta \in I_1] + \Pr[\theta \in I_2] \leq 2x_e. \tag{2.3}$$

Corollary 2.4. *$\mathbb{E}[C]$, the expected cost of C, is at most $2\sum_e w_e x_e$.*

Running time analysis and derandomization: A natural implementation of Algorithm 2.1 would first choose θ and then compute $\delta^+(B(t_i, \theta))$ for each i. This can be easily accomplished via k executions of Dijkstra's single-source shortest path algorithm, one for each t_i, leading to a running time of $O(k(m + n \log n))$ where $m = |E|$ and $n = |V|$. However, by taking advantage of our analysis in Lemma 2.2, we can obtain a run time that is equivalent to a single execution of Dijkstra's algorithm.

Consider a slight variation of Algorithm 2.1. For each edge $e = (u, v)$, define two intervals $I_1(e) = [d(s_1, u), d(s_1, u) + x_e)$ and $I_2(e) = [d(s_2, u), d(s_2, u) + x_e)$, where s_1, s_2 are the two terminals from which u is the closest in terms of distance. We pick $\theta \in (0, 1)$ uniformly at random and include e in C iff $\theta \in I_1(e)$ or $\theta \in I_2(e)$. The analysis in Lemmas 2.1 and 2.2 shows that even this modified algorithm outputs a feasible cut whose expected cost is at most $2\sum_e w_e x_e$. Note that the edges cut by this modified algorithm may be a strict superset of the edges cut by Algorithm 2.1. The advantage of the modified algorithm is that we only need to calculate $I_1(e)$ and $I_2(e)$ for each edge $e \in E$. To do this, for each node u, we need to find the two terminals from which u is the closest and their corresponding distances. More formally, consider the following h-nearest-terminal problem.

Pr Given a directed graph $G = (V, E)$ with non-negative edge-lengths, a set $S \subseteq V(G)$ of k terminals, and an integer $h \leq k$, for each vertex v, find the h terminals from which v is the closest among the terminals and their corresponding distances. In other words for each v find the h smallest values in $d(s_1, v), d(s_2, v), \ldots, d(s_k, v)$ where $S = \{s_1, \ldots, s_k\}$.

The above problem can be solved via a randomized algorithm using hashing that runs in expected time $O(h(m + n \log n))$, which corresponds to h executions of Dijkstra's algorithm. It can also be solved in $O(hm \log h + hn \log n)$ time via a deterministic algorithm. See [54] who refers to this as the h-nearest-neighbors problem.

Using the algorithm for the h-nearest-terminal problem with $h = 2$, we can calculate $I_1(e)$ and $I_2(e)$ for each $e \in E$ in $O(m + n \log n)$ time[1]. We then choose θ uniformly at random from $(0, 1)$ and cut e if θ lies in one of the range $I_1(e)$ or $I_2(e)$. This gives us a 2-approximate randomized algorithm with running time $O(m + n \log n)$.

We can derandomize the algorithm by computing the cheapest cut among all $\theta \in (0, 1)$ as follows. Once $I_1(e)$ and $I_2(e)$ are computed for each e we sort the $4m$ end points of these $2m$ intervals; let them be $\theta_1 \leq \theta_2 \leq \ldots \leq \theta_{4m}$. We observe that it suffices to evaluate the cut value at each of these values of θ. A simple scan of these $4m$ points while updating the cut-value at each end point can be accomplished in $O(m)$ time. Sorting the end points takes $O(m \log n)$ time. This leads to a deterministic 2-approximation algorithm with running time $O(m \log n)$.

Remark 2.1. *We can also write an equivalent version of algorithm 2.1 which does not require the addition of new vertices $t_1, \ldots, t_k$. This viewpoint is also useful in generalizing the* 2*-approximation for* DIR-MWC *to* DIR-MULTICUT *with k-matching extension free demand graphs.*

Algorithm 2.2 Equivalent rounding scheme for DIR-MWC

1: Given a feasible solution $\mathbf{x}$ to DIR-MWC-REL.
2: For all $u, v \in V$, $d(u, v)$= shortest path length from u to v according to lengths x_e
3: For all $u \in V, i \in [1, k]$, $d_1(u, s_i) = \max(0, 1 - \min_{j \in [1,k] \setminus \{i\}} d(s_j, u))$
4: Pick $\theta \in (0, 1)$ uniformly at random
5: $B_{s_i} = \{u \in V \mid d_1(u, s_i) \leq \theta\}$
6: $E' = \cup_{i \in [1,k]} \delta^-(B_{s_i})$
7: Return E'

---

[1] One can easily derive the $h = 2$ case from first principles also.

2.3.2 $(k-1)$-approximation for k-matching extension free demand graphs

In this section, we prove Theorem 2.10 which improves the approximation ratio for DIR-MULTICUT with restricted class of demand graphs. Recall that a directed demand graph $H = (V, F)$ contains an induced k-matching extension if there are two subsets of V, $S = \{s_1, \ldots, s_k\}$ and $T = \{t_1, \ldots, t_k\}$ satisfying the following properties: (i) for $1 \le i \le k, (s_i, t_i) \in F$ and (ii) for $i > j$, $(s_i, t_j) \notin F$.

Theorem 2.10. *Consider* DIR-MULTICUT *where the demand graph does not contain an induced k-matching-extension. Then the flow-cut gap is at most $k-1$ and there is a polynomial-time rounding algorithm that achieves this upper bound.*

Let $G = (V, E)$ and $H = (V, F)$ be the supply and demand graph for a DIR-MULTICUT instance. We provide a *generic* randomized rounding algorithm that given a fractional solution $\mathbf{x}$ to LP 2.1 for an instance (G, H) of DIR-MULTICUT returns a feasible solution; the rounding does not depend on H. We can prove that the returned solution is a $(k-1)$-approximation with respect to the fractional solution $\mathbf{x}$ or show that H contains an induced k-matching extension. This algorithm is inspired by our recent rounding scheme for DIR-MWC [23].

Let $\mathbf{x}$ be a feasible solution to LP 2.1. For $u, v \in V$, define $d(u, v)$ to be the shortest path length in G from vertex u to vertex v using lengths x_e. We also define another parameter $d_1(u, v)$ for each pair of vertices $u, v \in V$. $d_1(u, v)$ is the minimum non-negative number such that if we add an edge uv in G with $x_{uv} = d_1(u, v)$ then u is still separated from all the vertices it has to be separated from. Formally, for $u, v \in V$, $d_1(u, v) := \max(0, 1 - \min_{v' \in V, (u,v') \in F} d(v, v'))$. If for some vertex u, there is no demand edge leaving u in F then we define $d_1(u, v) = 0$ for all $v \in V$. The following properties of d_1 are easy to verify.

Lemma 2.3. $d_1(u, v)$ *satisfies the following properties:*

- $\forall u \in V, d_1(u, u) = 0$
- $\forall (u, v) \in F, v' \in V,\ d_1(u, v') + d(v', v) \ge 1$. *Hence,* $\forall (u, v) \in F, d_1(u, v) \ge 1$.
- *If* $d_1(u, v') \ne 0$, *then there exists* $(u, v) \in F$ *such that* $d_1(u, v') + d(v', v) = 1$
- $\forall u \in V, (a, b) \in E, d_1(u, b) - d_1(u, a) \le x_{ab}$

Algorithm 2.3 Rounding for DIR-MULTICUT

1: Given a feasible solution $\mathbf{x}$ to LP 2.1
2: For all $u, v \in V$, compute $d(u, v)$= shortest path length from u to v according to lengths x_e
3: For all $u, v \in V$, compute $d_1(u, v) = \max(0, 1 - \min_{v' \in V, uv' \in F} d(v, v'))$
4: Pick $\theta \in (0, 1)$ uniformly at random
5: $B_u = \{v \in V \mid d_1(u, v) \le \theta\}$
6: $E' = \cup_{u \in V} \delta^+(B_u)$
7: Return E'

Algorithm is a simple ball cut rounding around all the vertices as per $d_1(u,v)$. We pick a number $\theta \in (0,1)$ uniformly at random. For all $u \in V$, we consider θ radius ball around u for all $u \in V$; $B_u = \{v \in V \mid d_1(u,v) \leq \theta\}$. And then cut all the edges leaving the set B_u; $\delta^+(B_u) = \{(v,v') \in E_G \mid v \in B_u, v' \notin B_u\}$. Note that it is crucial that the same θ is used for all u.

Proving that E' is a feasible solution is easy. However, to bound the expected cost of the solution, we need the following lemma which shows that for any vertex v, number of u_i with different non-zero values of $d_1(u_i,v)$ is at most $k-1$.

Lemma 2.4. *If for some $v \in V$ there exists $u_1,\dots,u_k$ such that $0 \neq d_1(u_i,v) \neq d_1(u_j,v)$ for all $i \neq j$, then the demand graph H contains an induced k-matching extension.*

Proof: Rename the vertices $u_1,\dots,u_k$ such that $d_1(u_1,v) > \cdots > d_1(u_k,v) > 0$. By Lemma 2.3, there exists $v'_1,\dots,v'_k$ such that $u_iv'_i \in F$ and $d_1(u_i,v) + d(v,v'_i) = 1$. Consider the subgraph of H induced by the vertices $s_1,\dots,s_k,t_1,\dots,t_k$ where $s_i = u_i, t_i = v'_i$. Edge $(s_i,t_i) \in F$ as $(u_i,v'_i) \in F$. By construction $s_1,\dots,s_k$ are distinct. We also argue that $t_1,\dots,t_k$ are distinct. Suppose $t_i = t_j$, that is $v'_j = v'_i$ for $i < j$. Then, $d_1(u_j,v) + d(v,v'_j) < d_1(u_i,v) + d(v,v'_j) = d_1(u_i,v) + d(v,v'_i) = 1$. Since $u_jv'_j \in F$, by Lemma 2.3 $d_1(u_j,v) + d(v,v'_j) \geq 1$ which contradicts the inequality above.

For $i > j$, $d_1(s_i,v) + d(v,t_j) = d_1(u_i,v) + 1 - d_1(u_j,v) < 1$. By lemma 2.3, $(s_i,t_j) \notin F$. Thus, we have shown that $(s_i,t_i) \in F$ for $i \in [1,k]$ and $(s_i,t_j) \notin F$ for $i \geq j$. Thus H contains an induced k-matching extension.

Proof: [Theorem 2.10] We start by solving LP 2.1 and then perform the rounding scheme as per Algorithm 2.3. By Lemma 2.3, for all $(u,v) \in F, d_1(u,v) \geq 1$ and since $\theta < 1$, we have $u \in B_u, v \notin B_u$. We remove all the edges going out of the set B_u and hence, cut all the paths from u to v. As argued above, for all $uv \in E_H$, $u \in B_u, v \notin B_u$ and we cut the edges going out of B_u. Hence, there is no path from u to v in $G - E'$ and E' is a feasible Dir-Multicut solution.

We claim that $\Pr[e \in E'] \leq (k-1)x_e$ for all $e \in E_G$. Once we have this property, by linearity of expectation, the expected cost of E' can be bounded by $(k-1)$ times the LP cost: $\mathbb{E}[\sum_{e \in E'} w_e] \leq (k-1)\sum_{e \in E_G} w_e x_e$.

Now we prove the preceding claim. Consider an edge $e = (a,b) \in E$. Edge $e \in E'$ only if for some $u \in V$, $e \in \delta^+(B_u)$ and this holds only if $\theta \in [d_1(u,a), d_1(u,b))$. By Lemma 2.3, $d_1(u,b) \leq d_1(u,a) + x_e$. Hence, $e \in \delta^+(B_u)$, if $\theta \in [d_1(u,b) - x_{ab}, d_1(u,b))$. Denote this interval by $I_u(e)$.

By Lemma 2.4, there are at most $k-1$ distinct elements in the set $\{d_1(u,b) \mid u \in V\}$. This implies that there are at most $k-1$ distinct intervals $I_u(e)$. In other words, there exists $u_1,\dots,u_r, r \leq k-1$ such that $\cup_{u \in V} I_u(e) = \cup_{i=1}^r I_{u_i}(e)$.

$$
\begin{aligned}
\Pr[(a,b)\in E'] &\le \Pr[\theta \in \cup_{u\in V} I_u(e)] && (2.4)\\
&= \Pr[\theta \in \cup_{i=1}^{r} I_{u_i}(e)] && (2.5)\\
&\le \sum_{i=1}^{r} \Pr[\theta \in I_{u_i}(e)] && (2.6)\\
&\le r\cdot x_e \le (k-1)x_e. && (2.7)
\end{aligned}
$$

Penultimate inequality follows from the fact that $I_{u_i}(e)$ has length x_e and θ is chosen uniformly at random from $[0,1)$.

2.4 UGC-hardness of DIR-MULTICUT

In this section we prove hardness of approximation for DIR-MULTICUT-H, in particular Theorem 2.11 relating the hardness of approximation to the flow-cut gap. Recall that α_H is the worst-case flow-cut gap (equivalently, the integrality gap of the DISTANCE-LP) for instances of DIR-MULTICUT-H.

Theorem 2.11. *Assuming UGC, for any fixed directed bipartite graph H, and for any fixed $\varepsilon > 0$ there is no polynomial-time $(\alpha_H - \varepsilon)$ approximation for* DIR-MULTICUT-H.

We prove the theorem via a reduction to MIN-β-CSP and the hardness result of Ene, Vondrák and Wu [31]. We note that the result is technical and involves several steps. This is partly due to the fact that the theorem is establishing a meta-result. The theorem of [31] is in a similar vein. In particular [31] establishes that the hardness of MIN-β-CSP depends on the integrality gap of a specific LP formulation BASIC-LP (defined later). Our proof is based on establishing a correspondence between DIR-MULTICUT-H and a specific constraint satisfaction problem MIN-β_H-CSP where β_H is constructed from H (this is the heart of the reduction) and proving the following properties:

(I) Establish approximation equivalence between DIR-MULTICUT-H and MIN-β_H-CSP. That is, prove that each of them reduces to the other in an approximation preserving fashion.

(II) Prove that if the flow-cut gap for DIR-MULTICUT-H (equivalently the integrality gap of DISTANCE-LP) is α_H then the integrality gap of BASIC-LP for MIN-β_H-CSP is also α_H.

From (I), we obtain that the hardness of approximation factor for DIR-MULTICUT-H and MIN-β_H-CSP coincide. From (II), we can apply the result in [31] which shows that, assuming UGC, the hardness of approximation for MIN-β_H-CSP is the same as the integrality gap of BASIC-LP. Putting together these two claims give us our desired result.

It is not straightforward to directly relate DISTANCE-LP for DIR-MULTICUT-H and BASIC-LP for MIN-β_H-CSP. BASIC-LP appears to be stronger on first glance. In order to relate them we show that a seemingly strong LP for DIR-MULTICUT that we call LABEL-LP is in fact no stronger than DISTANCE-LP. In fact this can be seen as the key technical fact unerlying the entire proof and is independently interesting since it is quite different from the undirected graph setting. It is much easier to relate LABEL-LP and BASIC-LP. The rest of this section is organized as follows. In Section 2.4.1 we describe LABEL-LP and prove its equivalence with DISTANCE-LP. In Section 2.4.2 we describe MIN-β-CSP and BASIC-LP and formally state the theorem of [31] that we rely on. We then subsequently describe our reduction from DIR-MULTICUT-H to MIN-β_H-CSP and complete the proof.

2.4.1 LABEL-LP and equivalence with DISTANCE-LP for DIR-MULTICUT

In Section 2.2, we saw that if demand graph H has size k, then there is a labeling LP for MULTICUT (the undirected problem) with size poly$(2^k, n)$ and integrality gap at most 2 which improves upon the integrality gap of DISTANCE-LP which can be $\Omega(\log k)$. Here we describe a natural labeling LP for DIR-MULTICUT (LABEL-LP), but in contrast to the undirected case, we show that it is not stronger than DISTANCE-LP. We show this equivalence on an instance by instance basis. That is, for any DIR-MULTICUT instance I, given a solution to DISTANCE-LP, we can find a solution to LABEL-LP with same cost and vice versa.

Let the demand graph be H with vertex set $V_H = \{s_1, \ldots, s_k\}$, and the supply graph be $G = (V_G, E)$ with n vertices. We will assume here, for ease of notation, that $V_H \subset V_G$. Define a labeling set $L = \{0,1\}^k$ which corresponds to all subsets of V_H. We interpret the labels in L as k-length bit-vectors; if $\sigma \in L$ we use $\sigma[i]$ to denote the i'th bit of σ. For two labels $\sigma_1, \sigma_2 \in L$ we say $\sigma_1 \le \sigma_2$ if $\forall i, \sigma_1[i] \le \sigma_2[i]$. To motivate the formulation consider any set of edges $E' \subseteq E$ that can be cut. In $G' = G - E'$ we consider, for each $v \in V$, the reachability information from each of the terminals $s_1, s_2, \ldots, s_k$. For each v this can be encoded by assigning a label $\sigma_v \in L$ where $\sigma_v[i] = 1$ iff v is reachable from s_i in G'. E' is a feasible solution if s_i cannot reach s_j whenever (s_i, s_j) is an edge of H. The goal of the formulation to assign labels to vertices and to ensure that demand pairs are separated. An edge $e = (u, v)$ is cut if there is some s_i such that s_i can reach u but s_i cannot reach v. We add several constraints to ensure that the label assignment is consistent. The basic variables are $z_{v,\sigma}$ for each $v \in V_G$ and $\sigma \in L$ which indicate whether v is assigned the label σ. We also a variable x_e for each edge $e = (u, v) \in E_G$ that is derived from the label assignment variables. We start with the basic constraints involving these variables and then add additional variables that ensure consistency of the assignment.

- Each vertex is labelled by exactly one label. For $v \in V_G$, $\sum_{\sigma \in L} z_{v,\sigma} = 1$.
- Vertex s_i is reachable from s_i. For $s_i \in V_H$ and any $\sigma \in L$ such that $\sigma[i] = 0$, $z_{s_i,\sigma} = 0$
- Demand edges are separated. That is, if $(s_i, s_j) \in E_H$, then s_j is not reachable from s_i. That is, $z_{s_j,\sigma} = 0$ for any σ where $\sigma[i] = 1$ and $(s_i, s_j) \in E_H$.

For each edge $e = (u, v)$ we have variables of the form $z_{e,\sigma_1\sigma_2}$ where the intention is that u is labeled σ_1

$$
\begin{array}{lll}
\multicolumn{3}{c}{\textbf{LABEL-LP}} \\
\min & \sum_{e \in E} w_e x_e & \\
\sum_{\sigma \in L} z_{v,\sigma} = 1 & & v \in V_G, \sigma \in L \\
z_{s_i,\sigma} & = 0 & s_i \in V_H, \sigma \in L, \sigma[i] = 0 \\
z_{s_j,\sigma} & = 0 & \sigma \in L, \sigma[i] = 1, (s_i, s_j) \in E_H \\
\sum_{\sigma_2 \in L} z_{e,\sigma_1\sigma_2} = z_{u,\sigma_1} & & e = (u,v) \in E_G, \sigma_1 \in L \\
\sum_{\sigma_1 \in L} z_{e,\sigma_1\sigma_2} = z_{v,\sigma_2} & & e = (u,v) \in E_G, \sigma_2 \in L \\
\sum_{\sigma_1,\sigma_2 \in L : \sigma_1 \not\leq \sigma_2} z_{e,\sigma_1\sigma_2} = x_e & & e \in E_G \\
z_{v,\sigma}, z_{e,\sigma_1\sigma_2} & \leq 1 & v \in V_G, e \in E_G, \sigma, \sigma_1, \sigma_2 \in L \\
z_{v,\sigma}, z_{e,\sigma_1\sigma_2} & \geq 0 & v \in V_G, e \in E_G, \sigma, \sigma_1, \sigma_2 \in L
\end{array}
$$

Figure 2.5: LABEL-LP for DIR-MULTICUT

and v is labeled σ_2. To enforce consistency between edge assignment variables and vertex assignment variables we add the following set of constraints.

- For $e = (u, v) \in E_G$, $z_{u,\sigma_1} = \sum_{\sigma_2 \in L} z_{e,\sigma_1\sigma_2}$ and $z_{v,\sigma_2} = \sum_{\sigma_1 \in L} z_{e,\sigma_1\sigma_2}$.

Finally, the auxiliary variable x_e indicates whether e is cut.

- For $e = (u, v) \in E_G$, $x_e = 1$ if for some i, u is reachable from s_i and v is not reachable from s_i. Then, $x_e = 1$ if $z_{e,\sigma_1,\sigma_2} = 1$ for $\sigma_1 \not\leq \sigma_2$. We thus set $x_e = \sum_{\sigma_1,\sigma_2 \in L : \sigma_1 \not\leq \sigma_2} z_{e,\sigma_1\sigma_2}$.

It is not hard to show that if one constraints all the variables to be binary then the resulting integer program is a valid formuation for DIR-MULTICUT. Note that the number of variables is exponential in $k = |V_H|$. Relaxing the integrality constraint on the variables, we get LABEL-LP as shown in Fig 2.5.

Theorem 2.12. *For any instance G, H of* DIR-MULTICUT-H, *the optimum solution values for the formulations* LABEL-LP *and* DISTANCE-LP *are the same both in the fractional and integral settings.*

LABEL-LP has similarities to the earth-mover LP for metric labeling considered in [20, 63] except that the "distance" between labels is not a metric. Define a cost function $c : L \times L \to \{0, 1\}$ as follows: $c(\sigma, \sigma') = 0$ if $\sigma \leq \sigma'$ and 1 otherwise. In fact, given the basic labeling variables $z_{v,\sigma}$, other variables are decided in a min-cost solution. We explain this formally.

Interpreting Variables $z_{e,\sigma_1\sigma_2}$ and x_e as flow: Let $e = (u, v)$ be an edge in G. Consider a directed complete bipartite digraph B_{uv} with vertex set $\Gamma_u = \{u_\sigma \mid \sigma \in L\}$ and $\Gamma_v = \{v_\sigma \mid \sigma \in L\}$. We assign cost $c(\sigma, \sigma')$ on the edge $(u_\sigma, v_{\sigma'})$. We assign a supply of $z_{u,\sigma}$ on the vertex u_σ and a demand of $z_{v,\sigma}$ on the vertex v_σ. The values $z_{e,\sigma_1\sigma_2}$ can be thought of as flow from u_{σ_1} to v_{σ_2} satisfying the following properties: (i) total flow out of u_{σ_1} must be equal to the supply z_{u,σ_1} ($z_{u,\sigma_1} = \sum_{\sigma_2 \in L} z_{e,\sigma_1\sigma_2}$) (ii) total flow into v_{σ_2} must be equal to z_{v,σ_2} ($z_{v,\sigma_2} = \sum_{\sigma_1 \in L} z_{e,\sigma_1\sigma_2}$) (iii) flow is non-negative ($z_{e,\sigma_1\sigma_2} \geq 0$). The cost of the flow according to c is precisely x_e ($= \sum_{\sigma_1 \not\leq \sigma_2} z_{e,\sigma_1\sigma_2}$). In particular, given the values of the labeling variables $z_{u,\sigma}$, $\sigma \in L$ and $z_{v,\sigma'}$, $\sigma' \in L$ which can be thought of as two distributions on the labels, the smallest value of x_e that can be achieved is basically the min-cost flow in B_{uv} with supplies and demands defined by the two distributions. In other words the other variables are completely determined by the distributions if one wants a minimum cost solution.

In the sequel we use $\overline{z}_u$ to denote the vector of assignment value $z_{u,\sigma}$, $\sigma \in L$ and refer to $\overline{z}_u$ as the distribution corresponding to u.

Proof: [Theorem 2.12] **From LABEL-LP to DISTANCE-LP:** Let $(\mathbf{x}, \mathbf{z})$ be a feasible solution to LABEL-LP for the given instance of G, H. Consider a solution $\mathbf{x}'$ to DISTANCE-LP where we set $x'_e = x_e$. We claim that $\mathbf{x}'$ is a feasible solution to DISTANCE-LP for G, H. That is, for $(s_i, s_j) \in E_H$, and a path p from s_i to s_j, we have $\sum_{e \in p} x'_e \geq 1$.

Lemma 2.5. *For any edge $e = (u, v) \in E_G$ and $i \in \{1, \ldots, k\}$, $x_e \geq \sum_{\sigma \in L, \sigma[i]=1} z_{u,\sigma} - \sum_{\sigma \in L, \sigma[i]=1} z_{v,\sigma}$.*

Proof: Recall the interpretation of variables $z_{e,\sigma_1\sigma_2}$ as flow from set $\Gamma_u = \{u_\sigma \mid \sigma \in L\}$ to $\Gamma_v = \{v_\sigma \mid \sigma \in L\}$. Consider the following partition of Γ_u into $\Gamma_u^1 = \{u_\sigma \mid \sigma \in L, \sigma[i] = 1\}$ and $\Gamma_u^2 = \{u_\sigma \mid \sigma \in L, \sigma[i] = 0\}$. Similarly, consider the partition of Γ_v into Γ_v^1 and Γ_v^2. Amount of flow out of Γ_u^1 is equal to $\sum_{\sigma \in L, \sigma[i]=1} z_{u,\sigma}$ and amount of flow coming into Γ_v^1 is equal to $\sum_{\sigma \in L, \sigma[i]=1} z_{v,\sigma}$. Amount of flow from Γ_u^1 to Γ_v^1 is at most $\sum_{\sigma \in L, \sigma[i]=1} z_{v,\sigma}$. Hence, flow from Γ_u^1 to Γ_v^2 is at least $\sum_{\sigma \in L, \sigma[i]=1} z_{u,\sigma} - \sum_{\sigma \in L, \sigma[i]=1} z_{v,\sigma}$. For $u_{\sigma_1} \in \Gamma_u^1, v_{\sigma_2} \in \Gamma_v^2$, we have $\sigma_1 \not\leq \sigma_2$ and hence,

$$x'_e = x_e = \sum_{\sigma_1, \sigma_2 \in L : \sigma_1 \not\leq \sigma_2} z_{e,\sigma_1\sigma_2} \tag{2.8}$$

$$\geq \sum_{\sigma \in L : \sigma[i]=1} z_{u,\sigma} - \sum_{\sigma \in L, : \sigma[i]=1} z_{v,\sigma}. \tag{2.9}$$

Let $(s_i, s_j) \in E_H$. We prove that for any path p from s_i to s_j in G has $\sum_{e \in p} x'_e \geq 1$. Let the path p be $s_i, a_1, \ldots, a_\ell, s_j$. Then, by Lemma 2.5

$$x_{(s_i,a_1)} + \sum_{t=1}^{\ell-1} x_{(a_t,a_{t+1})} + x_{(a_\ell,s_j)} \geq \sum_{\sigma \in L : \sigma[i]=1} \Big[(z_{s_i,\sigma} - z_{a_1,\sigma}) \tag{2.10}$$

$$+ \sum_{t=1}^{\ell-1} (z_{a_t,\sigma} - z_{a_{t+1},\sigma}) + (z_{a_\ell,\sigma} - z_{s_j,\sigma}) \Big] \tag{2.11}$$

$$= \sum_{\sigma \in L, \sigma[i]=1} (z_{s_i,\sigma} - z_{s_j,\sigma}) \tag{2.12}$$

LABEL-LP ensures that $z_{s_i,\sigma} = 0$ if $\sigma[i] = 0$ and $z_{s_j,\sigma} = 0$ if $\sigma[i] = 1$. Hence, $\sum_{\sigma \in L:\sigma[i]=1} z_{s_i,\sigma} = 1$ and $\sum_{\sigma \in L:\sigma[i]=1} z_{s_j,\sigma} = 0$. Hence the right hand side in the preceding inequality is 1.

From DISTANCE-LP to LABEL-LP: Suppose $\mathbf{x}$ is a feasible solution to DISTANCE-LP for the given instance G, H. We construct a solution $(\mathbf{x}', \mathbf{z})$ for LABEL-LP such that $x'_e \le x_e$ for all $e \in E_G$. The edge lengths given by $\mathbf{x}$ induce shortest path distances in G and we use $d(u, v)$ to denote this distance from u to v. By adding dummy edges with zero cost as needed we can assume that $d(u, v) \le 1$ for each vertex pair (u, v). With this assumption in place we have that for any edge $e = (u, v)$ and any terminal s_i, $d(s_i, v) \le d(s_i, u) + x_e$; hence $x_e \ge \max_{1 \le i \le k}(d(s_i, v) - d(s_i, u))$. We will in fact prove that $x'_e \le \max_{1 \le i \le k}(d(s_i, v) - d(s_i, u))$.

We start by describing how to assign values to the variables $z_{v,\sigma}$. Recall that these induce values to the other variables if one is interested in a minimum cost solution. Let $d(u, v)$ denote the shortest distance from u to v in G as per lengths x_e.

For a vertex u, consider the permutation $\pi^u : \{1, \ldots, k\} \to \{1, \ldots, k\}$ such that $d(s_{\pi^u(1)}, u) \le \cdots \le d(s_{\pi^u(k)}, u)$. In other words π^u is an ordering of the terminals based on distance to u (breaking ties arbitrarily). Define $\sigma^u_0, \ldots, \sigma^u_k$ as follows:

$$\sigma^u_i[j] = \begin{cases} 1 & j \in \{\pi^u(1), \ldots, \pi^u(i)\} \\ 0 & j \notin \{\pi^u(1), \ldots, \pi^u(i)\} \end{cases} \tag{2.13}$$

In the assignment above it is useful to interpret σ^u_i as a set of indices of the terminals. Hence σ^u_0 corresponds to $\emptyset$ and σ^u_i to $\{\pi^u(1), \ldots, \pi^u(i)\}$. Thus, these sets form a chain with.

The assignment of values to the variables $z_{u,\sigma}$, $\sigma \in L$ is done as follows:

$$z_{u,\sigma} = \begin{cases} d(s_{\pi^u(1)}, u) & \sigma = \sigma^u_0 \\ d(s_{\pi^u(i+1)}, u) - d(s_{\pi^u(i)}, u) & \sigma = \sigma^u_i, i \in [1, k-1] \\ 1 - d(s_{\pi^u(k)}, u) & \sigma = \sigma^u_k \\ 0 & otherwise \end{cases} \tag{2.14}$$

Lemma 2.6. *$z_{u,\sigma}$ as defined above satisfy the following properties:*

- $\forall u \in V_G, \sigma \in L, z_{u,\sigma} \ge 0.$
- $\forall u \in V_G, \sum_{\sigma \in L} z_{u,\sigma} = 1.$
- *For $A \subseteq \{1, \ldots, k\}$, define $\sigma_A \in L$ as: $\sigma_A[i] = 1$ for $i \in A$ and 0 otherwise. Then,*

$$\sum_{\sigma \ge \sigma_A} z_{u,\sigma} = 1 - \max_{i \in A} d(s_i, u) \tag{2.15}$$

- *Terminals are labelled correctly. That is, for each s_j and $\sigma \in L$, $z_{s_j,\sigma} = 0$ if $\sigma[j] = 0$.*

- *If $(s_i, s_j) \in E_H$, then $z_{s_j,\sigma} = 0$ for $\sigma \in L$ such that $\sigma[i] = 1$.*

Proof: For $u \in V_G$, consider $\sigma_0^u, \sigma_1^u, \ldots, \sigma_k^u$ as defined above.

- $z_{u,\sigma} \geq 0$ is true by definition.

- By definition, $z_{u,\sigma} = 0$ if $\sigma \notin \{\sigma_0^u, \ldots, \sigma_k^u\}$. Hence,

$$\begin{aligned}
\sum_{\sigma \in L} z_{u,\sigma} &= \sum_{i=0}^{k} z_{u,\sigma_i^u} && (2.16)\\
&= d(s_{\pi^u(1)}, u) + 1 - d(s_{\pi(k)}, u) && (2.17)\\
&\quad + \sum_{i=1}^{k-1} d(s_{\pi^u(i+1)}, u) - d(s_{\pi^u(i)}, u) && (2.18)\\
&= 1 && (2.19)
\end{aligned}$$

- Let $j = \arg\max_{i:\pi^u(i) \in A} d(s_{\pi^u(i)}, u)$. Then, $\sigma_j^u, \ldots, \sigma_k^u \geq \sigma_A$ and $\sigma_0^u, \ldots, \sigma_{j-1}^u \ngeq \sigma_A$. Hence,

$$\begin{aligned}
\sum_{\sigma \geq \sigma_A} z_{u,\sigma} &= \sum_{i=j}^{k} z_{u,\sigma_i^u} && (2.20)\\
&= \sum_{i=j}^{k-1} d(s_{\pi^u(i+1)}, u) - d(s_{\pi^u(i)}, u) && (2.21)\\
&\quad +1 - d(s_{\pi^u(k)}, u) && (2.22)\\
&= 1 - d(s_{\pi^u(j)}, u) && (2.23)\\
&= 1 - \max_{i:\pi^u(i) \in A} d(s_{\pi^u(i)}, u) && (2.24)\\
&= 1 - \max_{i \in A} d(s_i, u) && (2.25)
\end{aligned}$$

- By definition of distance, $d(s_j, s_j) = 0$. Consider $A = \{j\}$. Applying the result from previous part, we get $\sum_{\sigma \geq \sigma_A} z_{s_j,\sigma} = 1 - 0 = 1$. Hence, $z_{s_j,\sigma} = 0$ if $\sigma \ngeq \sigma_A$. Equivalently speaking, $z_{s_j,\sigma} = 0$ if $\sigma[j] = 0$.

- Let $(s_i, s_j) \in E_H$. Then, for the solution $\mathbf{x}$ to be feasible, we must have $d(s_i, s_j) = 1$. Consider $A = \{i\}$. Then, using result from previous part, we get $\sum_{\sigma \geq \sigma_A} z_{s_j,\sigma} = 1 - 1 = 0$. Hence, $z_{s_j,\sigma} = 0$ if $\sigma \geq \sigma_A$. Equivalently speaking, $z_{s_j,\sigma} = 0$ if $\sigma[i] = 1$.

Consider an edge $e = (u, v)$. Recall that once the distributions of $\overline{z}_u$ and $\overline{z}_v$ are fixed, x'_e is simply the min-cost flow between these two distributions in the digraph B_{uv} with costs given by c. Our goal is to show that this cost is at most $\max\{0, \max_i(d(s_i, v) - d(s_i, u))\}$. Suppose we define a partial flow between $\overline{z}_u$ and $\overline{z}_v$ on zero-cost edges such that the total amount of this flow is γ where $\gamma \in [0,1]$. Then it is easy to see that we can complete this flow to achieve a cost of $(1-\gamma)$. This is because the graph is a complete bipartite graph and costs are either 0 or 1 and $\overline{z}_u$ and $\overline{z}_v$ are distributions that have a total of eactly one unit of mass on each side.

Next, we define a partial flow of zero cost between $\overline{z}_u$ and $\overline{z}_v$ by setting some variables $z_{e,\sigma_1\sigma_2}$ in a greedy fashion as follows. Initially all flow values are zero. We consider the vertex $u_{\sigma_i^u}$ with supply z_{u,σ_i^u} in order from $i = 0$ to k. Our goal is to send as much flow as possible from this vertex on zero-cost edges to demand vertices $v_{\sigma_j^v}$ which requires that $\sigma_i^u \le \sigma_j^v$. We maintain the invariants that we do not exceed supply or demand in this process. While trying to send flow out of $u_{\sigma_i^u}$ we again use a greedy process; if there are $j < j'$ such that σ_j^v and $\sigma_{j'}^v$ are both eligible to receive flow on zero-cost edges and have capacity left, we use j first; recall that σ_j^v corresponds to a subset of $\sigma_{j'}^v$. Let $z_{e,\sigma_1\sigma_2}$ be the partial flow created by the algorithm.

Lemma 2.7. *The total flow sent by the greedy algorithm described is at least* $1-\max\{0, \max_h(d(s_h, v) - d(s_h, u))\}$.

Assuming the lemma we are done because the zero-cost flow is at least $1-x_e$ and hence total cost of the flow is at most x_e, thus proving $x'_e \le x_e$ as desired. We now prove the lemma.

Consider the greedy flow. Let ℓ be the maximum integer such that $v_{\sigma_\ell^v}$ is not saturated by the flow. If no such ℓ exists then the greedy algorithm has sent a total flow of one unit on zero-cost edges and hence $x'_e = 0$. Thus, we can assume ℓ exists. Moreover, in this case we can also assume that $\ell < k$ for if $\ell = k$ the greedy algorithm can send more flow since $\sigma_i^u \le \sigma_k^v$ for all i. Let ℓ' be the maximum integer such that $\sigma_{\ell'}^u \le \sigma_\ell^v$. Such an ℓ' exists since $\ell' = 0$ is a candidate (corresponding to the empty set). Moreover, $\ell' < k$ since $\sigma_k^u \not\le \sigma_\ell^v$ since $\ell < k$. Let ℓ'' be the minimum integer such that $\sigma_{\ell'+1}^u \le \sigma_{\ell''}^v$. ℓ'' exists because k is a candidate for it.

Claim 2.1. $\pi_{\ell'+1}^u = \pi_{\ell''}^v$.

Proof: By choice of ℓ, ℓ', ℓ'' we have $\sigma_{\ell'}^u \le \sigma_\ell^v$ and $\sigma_{\ell'+1}^u \not\le \sigma_\ell^v$ while $\sigma_{\ell'+1}^u \le \sigma_{\ell''}^v$. Thus $\ell'' \ge \ell+1$ and $\sigma_{\ell'}^u \le \sigma_\ell^v \le \sigma_{\ell''-1}^v$. Moreover, since ℓ'' is chosen to smallest, $\sigma_{\ell'+1}^u \not\le \sigma_{\ell''-1}^v$. These facts imply the desired claim.

We now claim several properties of the partial flow and justify them.

- $\forall i \in [0,\ell']$, $j \in [\ell+1, k]$, $z_{e,\sigma_i^u\sigma_j^v} = 0$. This follows from the fact that the greedy algorithm did not saturate z_{v,σ_ℓ^v}.
- $\forall i \in [\ell'+1, k]$, $j \in [0, \ell''-1]$, $z_{e,\sigma_i^u\sigma_j^v} = 0$. From the definition of ℓ', ℓ'', this is not a zero cost edge.
- $\forall i \in [0,\ell']$, $\sum_{j=0}^{\ell} z_{e,\sigma_i^u\sigma_j^v} = z_{u,\sigma_i^u}$. From definition of ℓ', for each $i \le \ell'$, there is a zero-cost edge from $u_{\sigma_i^u}$ to $v_{\sigma_\ell^v}$. Since the greedy algorithm did not saturate $v_{\sigma_\ell^v}$, it means that $u_{\sigma_i^u}$ is saturated and sends flow only to $v_{\sigma_1^v}, \ldots v_{\sigma_\ell^v}$.

- $\forall j \in [\ell'', k]$, $\sum_{i=\ell'+1}^{k} z_{e,\sigma_i^u \sigma_j^v} = z_{v,\sigma_j^v}$. By definition of ℓ, for $j \geq \ell + 1$ we have the property that $v_{\sigma_j^v}$ is saturated. As we argued above, for $i \in [\ell'+1, k], j \in [0, \ell''-1]$ we have $z_{e,\sigma_i^u \sigma_j^v} = 0$. Hence, for $j \geq \ell'' \geq \ell + 1$, we have $\sum_{i=\ell'+1}^{k} z_{e,\sigma_i^u \sigma_j^v} = \sum_{i=0}^{k} z_{e,\sigma_i^u \sigma_j^v} = z_{v,\sigma_j^v}$.

From the preceding claim we see that the total value of the partial flow can be summed up as

$$\sum_{\sigma_1,\sigma_2 \in L} z_{e,\sigma_1 \sigma_2} = \sum_{i=0}^{\ell'} z_{u,\sigma_i^u} + \sum_{j=\ell''+1}^{k} z_{v,\sigma_j^v}. \tag{2.26}$$

Moreover, by construction of $\overline{z}_u$ and $\overline{z}_v$,

$$\sum_{i=0}^{\ell'} z_{u,\sigma_i^u} = d(s_{\pi^u(\ell'+1)}, u) \tag{2.27}$$

$$\text{and} \quad \sum_{j=\ell''+1}^{k} z_{v,\sigma_j^v} = 1 - d(s_{\pi^v(\ell'')}, v). \tag{2.28}$$

Letting $h = \pi^u_{\ell'+1} = \pi^v_{\ell''}$ we see that from the preceding equalities that the total flow routed on the zero-cost edges is

$$d(s_h, u) + 1 - d(s_h, v) = 1 - (d(s_h, v) - d(s_h, u)) \geq 1 - x_e. \tag{2.29}$$

This finishes the proof.

2.4.2 Min-CSP and Basic-LP

Min-CSP refers to a minimization version of constraint satisfaction problems. We set up the formalism borrowed from [31]. Let L denote a finite set of labels. A real-valued function $f : L^i \to \mathbb{R}$ has *arity* i. Let $\Gamma = \{\psi \mid \psi : L^i \to [0,1] \cup \{\infty\}, i \leq k\}$ be the set of functions defined on L with arity at most k and range $[0,1] \cup \{\infty\}$. Let $\beta \subset \Gamma$ be a finite subset of ψ. These functions are also referred to as *predicates*. k denotes the arity and L denotes the alphabet of β. Each β induces an optimization problem Min-β-CSP.

Definition 2.1. *An instance of Min-β-CSP consists of the following:*

- *A vertex set V and a set of tuples $T \subset \cup_{i=1}^{k} V^i$.*
- *A predicate $\psi_t \in \beta$ for each tuple $t \in T$ where cardinality of t matches the arity of ψ_t.*
- *A non-negative weight function over the set of tuples, $w : T \to \mathbb{R}^+$.*

$$
\begin{array}{|l|}
\hline
\textbf{Basic-LP} \\
\min \quad \sum_{t \in T} w_t \cdot \sum_{\alpha \in L^{|t|}} z_{t,\alpha} \cdot \psi_t(\alpha) \\
\quad \sum_{\sigma \in L} z_{v,\sigma} \quad = 1 \qquad v \in V \\
\sum_{\alpha \in L^{|t|} : \alpha[i] = \sigma} z_{t,\alpha} = z_{v,\sigma} \quad t \in T, v = t[i], \sigma \in L \\
z_{v,\sigma}, z_{t,\alpha} \quad \geq 0 \quad v \in V, \sigma \in L, t \in T, \alpha \in L^{|t|} \\
z_{v,\sigma}, z_{t,\alpha} \quad \leq 1 \quad v \in V, \sigma \in L, t \in T, \alpha \in L^{|t|} \\
\hline
\end{array}
$$

Figure 2.6: Basic LP for Min-β-CSP

The goal is to find a label assignment $\ell : V \rightarrow L$ to minimize $\sum_{t=(v_{i_1},\ldots,v_{i_j}) \in T} w_t \cdot \psi_t(\ell(v_{i_1}), \ldots, \ell(v_{i_j}))$.

Consider an integer programming formulation with following variables: for each vertex $v \in V$ and label $\sigma \in L$, we have a variable $z_{v,\sigma}$ which is 1 if v is assigned label σ. Also, for each tuple $t = (v_{i_1}, \ldots, v_{i_j}) \in T$ and $\alpha \in L^{|t|}$, we have a boolean variable $z_{t,\alpha}$ which is 1 if v_{i_p} is labelled $\alpha[p]$ for $p \in [1, j]$. These variables satisfy following constraints:

- Each vertex receives unique label: $\sum_{\sigma \in L} z_{v,\sigma} = 1$.
- Variables $z_{v,\sigma}$ and $z_{t,\alpha}$ are consistent. That is, if $v \in t$ is assigned label σ, then $z_{t,\alpha}$ must be zero if α does not assign label σ to v. For every tuple $t \in T, v = t[i], \sigma \in L$, we have: $z_{v,\sigma} = \sum_{\alpha \in L^{|t|} : \alpha[i] = \sigma} z_{t,\alpha}$.

The objective is to minimize $\sum_{t \in T} w_t \cdot \sum_{\alpha \in L^{|t|}} z_{t,\alpha} \cdot \psi_t(\alpha)$.

Basic-LP is the LP relaxation obtained by allowing the variables to take on values in $[0, 1]$ and is described in Fig 2.6. For instance, $\mathfrak{I}$, $LP(\mathfrak{I})$ and $OPT(\mathfrak{I})$ refer to the fractional and integral optimum values respectively.

A particular type of predicate termed NAE (for *not all equal*) is important in subsequent discussion.

Definition 2.2. *For $i \geq 2$, $NAE_i : L^i \rightarrow \{0, 1\}$ be a predicate such that $NAE_i(\sigma_1, \ldots, \sigma_i) = 0$ if $\sigma_1 = \sigma_2 = \cdots = \sigma_i$ and 1 otherwise.*

The following theorem shows that the hardness of Min-β_H-CSP coincides with the integrality gap of Basic-LP if NAE_2 is in β.

Theorem 2.13. *(Ene, Vondrak, Wu [31]) Suppose we have a* Min-β-CSP *instance $\mathfrak{I} = (V, T, \Psi, w)$ with fractional optimum (of Basic LP) $LP(\mathfrak{I}) = c$, integral optimum $OPT(\mathfrak{I}) = s$, and β contains the predicate NAE_2. Then, assuming UGC, for any ϵ, for some $\lambda > 0$, it is NP-hard to distinguish between instances of* Min-β-CSP *where the optimum value is at least $(s - \epsilon)\lambda$ and instances where the optimum value is less than $(c + \epsilon)\lambda$.*

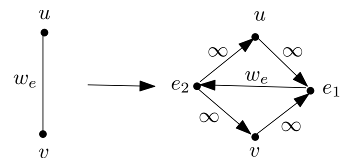


Figure 2.7: Gadget to convert undirected edge/NAE_2 predicate to a directed graph

2.4.3 DIR-MULTICUT-H and an equivalent MIN-β-CSP Problem

In this section, we show that given a bipartite directed graph $H = (S \cup T, E_H)$, we can construct a set of predicates β_H such that DIR-MULTICUT-H is equivalent to MIN-β_H-CSP. The notion of equivalence is as follows. We give a reduction from instances of DIR-MULTICUT-H to instances of MIN-β_H-CSP which preserves the cost of optimal integral solution and in addition, also preserves the cost of optimum fractional solution to LABEL-LP and BASIC-LP. Similarly, we give a reduction from MIN-β_H-CSP to DIR-MULTICUT-H which preserves the cost of both the integral and fractional solutions.

The basic idea behind the construction of β_H from H is to simulate the constraints of LABEL-LP via the predicates of β_H. In addition to setting up β_H correctly, we also need to preprocess the supply graph to prove the correctness of the reductions. Let the bipartite demand graph H be $(S \cup T, E_H)$ with $S = \{a_1, \dots, a_p\}$ and $T = \{b_1, \dots, b_q\}$ as the bipartition. For $u \in S$ let $N_H^+(u) = \{v \in T \mid (u, v) \in E_H\}$ be the neighbors of u in H. For $i \in [1, p]$, let $Y_i = \{j \in [1, p] \mid N_H^+(a_j) \subseteq N_H^+(a_i)\}$. That is, if $a_j \in Y_i$, the set of terminals that a_j needs to be separated from is a subset of the terminals that a_i needs to be separated from. For $j \in [1, q]$ let $Z_j = \{i \in [1, p] \mid a_i b_j \notin E_H\}$. That is, Z_j is the set of all terminals in S that do not need to be separated from b_j.

Assumptions on supply graph: We will assume that the supply graph G in the instances of DIR-MULTICUT-H satisfy the following properties.

- Assumption I: G may contain *undirected* edges. The meaning of this is that a path may include this edge in either direction. A simple and well-known gadget shown in Fig 2.7 shows that this is without loss of generality.
- Assumption II: For $1 \leq j \leq q$ and $i \in Z_j$, there is an infinite weight edge from a_i to b_j in G. Moreover b_j has no outgoing edge.
- Assumption III: For $1 \leq i \leq p$, and $i' \in Y_i$, there is an infinite weight edge from $a_{i'}$ to a_i in G. Moreover a_i has no other incoming edges.

The preceding assumptions are to make the construction of β_H and the subsequent proof of equivalence with DIR-MULTICUT-H somewhat more transparent and technically easier. Undirected edges allow us to use the NAE_2 predicate in β_H. Assumption II and III simplify the reachability information of terminals that needs to be kept track of and this allows for a simpler label set definition and easier proof of equivalence.

DISTANCE-LP easily generalizes to handle undirected edges; in examining paths from s_i to t_i for a demand pair we allow an undirected edge to be used in both directions. A more technical part is to generalize LABEL-LP to handle undirected edges in the supply graph. For a directed edge e recall that $x_e = \sum_{\sigma_1,\sigma_2 \in L:\sigma_1 \not\leq \sigma_2} z_{e,\sigma_1\sigma_2}$. For an undirected edge e we set $x_e = \sum_{\sigma_1,\sigma_2 \in L:\sigma_1 \neq \sigma_2} z_{e,\sigma_1\sigma_2}$.

The following two lemmas help establish that we can safely assume that the supply graph satisfies the assumptions I, II, and III. We omit the proof of the first lemma which involves tedious reworking of some of the details on equivalence of LABEL-LP and DISTANCE-LP.

Lemma 2.8. *For any instance G, H of* DIR-MULTICUT-H *where the supply graph has undirected edges, the optimum solution values for the formulations* LABEL-LP *and* DISTANCE-LP *are the same both in the fractional and integral settings.*

Assuming the preceding lemma, following lemma is easy to prove:

Lemma 2.9. *For bipartite H,* DIR-MULTICUT-H *with a general supply graph and* DIR-MULTICUT-H *restricted to supply graphs satisfying Assumptions I, II and III are equivalent in terms of approximability and in terms of the integrality gap of* DISTANCE-LP *(equal to integrality gap of* LABEL-LP*).*

Proof: We sketch the proof. Undirected edges can be handled by the gadget shown in Fig 2.7. It is easy to see that given any instance of DIR-MULTICUT-H with supply graph G and bipartite demand graph H we can first add dummy terminals to G and assume that each terminal a_i has only one outgoing infinite weight edge (to the original terminal) and each b_j has only one incoming infinite weight edge. With this in place adding edges to satisfy Assumptions II and III can be seen to not affect the integral or fractional solutions to DISTANCE-LP.

We will assume for simplicity that all weights (for edges and constraints) are either 1 or ∞. Generic weights can be easily simulated by copies and the proofs make no essential use of weights other than that some are finite, and others are infinite.

Constructing β_H from H: Next, we formally define β_H for a bipartite graph $H = (S \cup T, E_H)$ where $S = \{a_1, \ldots, a_p\}$ and $T = \{b_1, \ldots, b_q\}$. Recall the definitions of Y_i for $1 \leq i \leq p$ and Z_j for $1 \leq j \leq q$ based on E_H. Observe that no vertex other than b_j is reachable from b_j. And, since labels encode the reachability from terminals, we can ignore the reachability from b_j and define β_H with respect to terminal set S. For $\sigma \in \{0,1\}^p$, let $J_\sigma = \{i \in [1,p] \mid \sigma[i] = 1\}$

- Alphabet (Label Set) $L = \{0,1\}^p$. Labels encode the list of a_i's from which a vertex is reachable.
- For $i \in [1,p]$, a unary predicate ψ_{a_i} encode the correct label for a_i and is defined as follows: $\psi_{a_i}(\sigma) = 0$ if $J_\sigma = Y_i$, otherwise $\psi_{a_i}(\sigma) = \infty$.
- For $j \in [1,q]$, predicate ψ_{b_j} that encodes the correct label for b_j. $\psi_{b_j}(\sigma) = 0$ if $J_\sigma = Z_j$, otherwise $\psi_{b_j}(\sigma) = \infty$.

- A binary predicate $\mathcal{C}$ that encodes whether a directed edge is cut or not. If $\sigma_1 \leq \sigma_2$ $\mathcal{C}(\sigma_1, \sigma_2) = 0$, otherwise $\mathcal{C}(\sigma_1, \sigma_2) = 1$.
- A binary predicate NAE_2 that encodes whether an undirected edge is cut or not. If $\sigma_1 = \sigma_2$ $\text{NAE}_2(\sigma_1, \sigma_2) = 0$, otherwise $\text{NAE}_2(\sigma_1, \sigma_2) = 1$.

Thus $\beta_H = \{\mathcal{C}, \text{NAE}_2\} \cup \{\psi_{a_i} \mid i \in [1, p]\} \cup \{\psi_{b_j} \mid j \in [1, q]\}$. MIN-$\beta_H$-CSP has label set L, predicate set β_H and arity 2.

The main technical theorem we prove is the following. We remark that when we refer to DIR-MULTICUT-H we are referring to the problem where the supply graph satisfies the assumptions I, II, III that we outlined previously.

Theorem 2.14. *Let H be a directed bipartite graph. There is a polynomial time reduction that given a* DIR-MULTICUT-H *instance $I_M = (G = (V_G, E_G, w_G : E_G \to R^+), H = (S \cup T, E_H))$, outputs a* MIN-$\beta_H$-CSP *instance $I_C = (V_C, T_C, \psi_{T_C} : T_C \to \beta_H, w_{T_c} : T_C \to R^+)$ such that the following holds: given a solution $(\boldsymbol{x}, \boldsymbol{z})$ of the Label LP for I_M, we can construct a solution $\boldsymbol{z}'$ of Basic LP for I_C with cost at most that of $(\boldsymbol{x}, \boldsymbol{z})$ and vice versa. Moreover, if $(\boldsymbol{x}, \boldsymbol{z})$ is an integral solution, then $\boldsymbol{z}'$ is also an integral solution and vice versa. A similar reduction exists from* MIN-β_H-CSP *to* DIR-MULTICUT-H.

With the preceding theorem in place we can formally prove Theorem 2.11

Proof: [Theorem 2.11] Let I_M be some fixed instance of DIR-MULTICUT-H with flow-cut gap α_H. From Theorem 2.12 the integrality gap of LABEL-LP on I_m is also α_H. Let I_C be the MIN-β_H-CSP instance obtained via the reduction guaranteed by Theorem 2.14. I_M and I_C have the same integral cost. Fractional cost of LABEL-LP for I_M and BASIC-LP for I_C are also the same. Therefore, the integrality gap of BASIC-LP on I_C is also α_H. Via Theorem 2.13, assuming UGC, MIN-β_H-CSP is hard to approximate within a factor of $\alpha_H - \varepsilon$ for any fixed $\varepsilon > 0$.

Theorem 2.14 (the second part) implies that MIN-β_H-CSP reduces to DIR-MULTICUT-H in an approximation preserving fashion. Thus, DIR-MULTICUT-H is at least has hard to approximate as MIN-β_H-CSP which implies that assuming UGC, the hardness of DIR-MULTICUT-H is at least $\alpha_H - \varepsilon$.

Proof: [Theorem 2.14] BASIC-LP and LABEL-LP are almost identical except for the fact that LABEL-LP is defined with label set $\{0, 1\}^k$ where $k = p + q$ is the total number of terminals whereas BASIC-LP is defined with label set $\{0, 1\}^p$. However, since b_i's do not have any outgoing edge, reachability from b_i is trivial.

1. Reduction from MIN-β_H-CSP *to* DIR-MULTICUT-H Let the MIN-β-CSP instance be $I_C = (V_C, T_C, \psi_{T_C} : T_C \to \beta_H, w_{T_C} : T_C \to R^+)$. We refer to tuple $t = (u)$ with $\psi_{T_C}(t) = \psi_{a_i}$ as constraint $\psi_{a_i}(u)$, $t = (u), \psi_{T_C}(u) = \psi_{b_j}$ as constraint $\psi_{b_j}(u)$, $t = (u, v), \psi_{T_C}(t) = \mathcal{C}$ as constraint $\mathcal{C}(u, v)$ and $t = (u, v), \psi_{T_C}(u) = NAE_2$ as constraint $NAE_2(u, v)$. We assume that for every $i \in [1, p]$, there is a constraint $\psi_{a_i}(u_i)$ for some vertex $u_i \in V_C$, and similarly for every $j \in [1, q]$ there is a constraint $\psi_{b_j}(v_j)$ for some vertex $v_j \in V_C$; moreover, we will assume that $u_1, \ldots, u_p, v_1, \ldots, v_q$ are distinct vertices. One can ensure that this assumption holds by adding dummy vertices and dummy constraints with zero weight. We create an instance $I_M = (G = (V_G, E_G, w_G : E_G \to R^+), (S \cup T, E_H))$ of DIR-MULTICUT-H as follows.

- $V_G = V_C$, the vertex remains the same. Pick vertices $u_1, \ldots, u_p$ and $v_1, \ldots, v_q$ that are all distinct such that for $1 \le i \le p$ there is a constraint $\psi_{a_i}(u_i)$ in I_C and for $1 \le j \le q$ there is a constraint $\psi_{b_j}(v_j)$ in I_C. This holds by our assumption. For $i \in [1, p]$ associate the terminal $a_i \in V_H$ with u_i and for $j \in [1, q]$ associate the terminal $b_j \in V_H$ with v_j.
- E_G and w_G are defined as follows:
  - For each constraint $\psi_{a_i}(u)$ in I_C where $u \neq a_i$ add an undirected edge $e_t = a_i u$ to E_G with $w_G(e_t) = \infty$.
  - For each constraint $\psi_{b_j}(v)$ in I_C where $v \neq b_j$ add an undirected edge $e_t = b_j v$ to E_G with $w_G(e_t) = \infty$.
  - For each constraint $\mathcal{C}(u, v)$ in I_C add a directed edge $e_t = (u, v)$ in G with $w_G(e_t)$ equal to the weight of the constraint in I_C.
  - For each constraint $\text{NAE}_2(u, v)$, add an undirected edge $e_t = uv$ with $w_G(e_t)$ equal to the weight of the constraint in I_C.
  - For each $i \in [1, p]$ and for each $i' \in Y_i$, add a directed edge $e = (a_{i'}, a_i)$ with $w_G(e) = \infty$.
  - For each $j \in [1, q]$ and each $i \in Z_j$, add a directed edge $e = (a_i, b_j)$ with $w_G(e) = \infty$.

We now prove the equivalence of I_C and I_M from the point of view solutions to BASIC-LP and LABEL-LP respectively.

Given two labels σ and σ' which can be interpreted as binary strings, we use the notation $\sigma \cdot \sigma'$ to denote the label obtained by concatenating σ and σ'.

1.1. From LABEL-LP *to* BASIC-LP*:* Suppose $(\mathbf{x}, \mathbf{z})$ is a feasible solution to LABEL-LP for I_M. We construct a solution $\mathbf{z}'$ to BASIC-LP for I_C in the following way. $\mathbf{z}'$ is simply a projection of $\mathbf{z}$ from label set $\{0,1\}^{p+q}$ onto label set $\{0,1\}^p$. Recall that in the instance I_M the terminals $b_1, \ldots, b_q$ do not have any outgoing edges. Hence, in the solution $(\mathbf{x}, \mathbf{z})$ with label space $\{0,1\}^{p+q}$, which encodes reachability from both the a_iï£is and the b_j's the information on reachability from the b_jï£is does not play any essential role. We formalize this below.

- For $v \in V_C, \sigma \in \{0,1\}^p$, $z'_{v,\sigma} = \sum_{\sigma' \in \{0,1\}^q} z_{v,\sigma\cdot\sigma'}$.
- For unary constraint $t = (v) \in T_C$ and $\sigma \in \{0,1\}^p$, $z'_{t,\sigma} = z'_{v,\sigma}$.
- For binary constraint $t = (u, v) \in T_C$, for $\sigma_1, \sigma_2 \in \{0,1\}^p$,

$$z'_{t,\sigma_1\sigma_2} = \sum_{\sigma' \in \{0,1\}^q} \sum_{\sigma'' \in \{0,1\}^q} z_{e_t,\sigma_1\cdot\sigma'\sigma_2\cdot\sigma''} \tag{2.30}$$

Note that if $(\mathbf{x}, \mathbf{z})$ is an integral solution then $\mathbf{z}'$ as defined above is also an integral solution.

Feasibility of $\mathbf{z}'$ for BASIC-LP is an "easy" consequence of the projection operation but we prove it formally.

Lemma 2.10. *$\mathbf{z}'$ as defined above is a feasible solution to* BASIC-LP *for instance I_C.*

Proof: From the definition of $\mathbf{z}'$, for each vertex v,

$$\sum_{\sigma \in \{0,1\}^p} z'_{v,\sigma} = \sum_{\sigma \in \{0,1\}^p} \sum_{\sigma' \in \{0,1\}^q} z_{v,\sigma \cdot \sigma'} = 1 \tag{2.31}$$

which proves that one set of constraints holds.

Next, we prove that for $t \in T_C, v = t[i], \sigma \in L = \{0,1\}^p$, the constraint $z'_{v,\sigma} - \sum_{\alpha \in L^{|t|}:\alpha[i]=\sigma} z'_{t,\alpha} = 0$ holds. We consider unary and binary predicates separately.

- For $t = (v)$ such that $v = t[i], \sigma \in L = \{0,1\}^p$,

$$z'_{v,\sigma} - \sum_{\alpha \in L^{|t|}:\alpha[i]=\sigma} z'_{t,\alpha} = z'_{v,\sigma} - z'_{t,\sigma} = z'_{v,\sigma} - z'_{v,\sigma} = 0. \tag{2.32}$$

- For $t = (u, v) \in T_C, \sigma \in \{0,1\}^p$

$$\begin{aligned} & z'_{v,\sigma} - \sum_{\sigma_1 \in \{0,1\}^p} z'_{t,\sigma_1\sigma} && (2.33)\\ = \; & z'_{v,\sigma} - \sum_{\sigma_1 \in \{0,1\}^p} \sum_{\sigma',\sigma'' \in \{0,1\}^q} z_{e_t,\sigma_1 \cdot \sigma'' \sigma \cdot \sigma'} && (2.34)\\ = \; & z'_{v,\sigma} - \sum_{\sigma' \in \{0,1\}^q} \sum_{\sigma_1 \in \{0,1\}^p, \sigma'' \in \{0,1\}^q} z_{e_t,\sigma_1 \cdot \sigma'' \sigma \cdot \sigma'} && (2.35)\\ = \; & z'_{v,\sigma} - \sum_{\sigma' \in \{0,1\}^p} z_{v,\sigma \cdot \sigma'} = 0. && (2.36) \end{aligned}$$

  Similar argument holds for u as well.

Lemma 2.11. *The cost of $\mathbf{z}'$ is at most $\sum_{e \in E_G} w_e x_e$ which is the cost of $(\mathbf{x}, \mathbf{z})$ to I_M.*

Before we prove Lemma 2.11 we establish some properties satisfied by $(\mathbf{x}, \mathbf{z})$.

Lemma 2.12. *If the solution $(\mathbf{x}, \mathbf{z})$ to* LABEL-LP *has finite cost, then the following conditions hold:*

- *For directed edge $e = (u, v)$, and for $i \in [1, p]$ $x_e \geq \sum_{\sigma \in \{0,1\}^{p+q}:\sigma[i]=1} z_{u,\sigma} - \sum_{\sigma \in \{0,1\}^{p+q}:\sigma[i]=1} z_{v,\sigma}$. Hence, if edge e has infinite weight ($w_G(e) = \infty$), then $\sum_{\sigma \in \{0,1\}^{p+q}:\sigma[i]=1} z_{u,\sigma} \leq \sum_{\sigma \in \{0,1\}^{p+q}:\sigma[i]=1} z_{v,\sigma}$*

- *For $i \in [1,p], \sigma \in \{0,1\}^p, \sigma' \in \{0,1\}^q$ s.t. $J_\sigma \neq Y_i$, we have $z_{a_i,\sigma\cdot\sigma'} = 0$. Hence, for $\sigma \in \{0,1\}^p, z'_{a_i,\sigma} = 1$ if $J_\sigma = Y_i$ and 0 otherwise.*

- *For $j \in [1,q], \sigma \in \{0,1\}^p, \sigma' \in \{0,1\}^q$ s.t. $J_\sigma \neq Z_j$ we have $z_{b_j,\sigma\cdot\sigma'} = 0$. Hence, for $\sigma \in \{0,1\}^p, z'_{b_j,\sigma} = 1$ if $J_\sigma = Z_j$ and 0 otherwise.*

- *For an undirected edge $e = uv \in E_G$ with $w_G(e) = \infty$, and $\sigma_1, \sigma_2 \in \{0,1\}^{p+q}$, $z_{e,\sigma_1\sigma_2} = 0$ if $\sigma_1 \neq \sigma_2$. For $\sigma \in \{0,1\}^{p+q}, z_{u,\sigma} = z_{v,\sigma}$ and for $\sigma_1 \in \{0,1\}^p, z'_{u,\sigma_1} = z'_{v,\sigma_1}$. Hence, for $t = (u) \in T_C$ s.t. $\psi_{T_C}(t) = \psi_{a_i}, z'_{u,\sigma} = 1$ if $J_\sigma = Y_i$ and 0 otherwise.*

Proof: If $(\mathbf{x}, \mathbf{z})$ has finite cost, then for an edge e with infinite weight ($w_G(e) = \infty$), we must have $x_e = 0$.

- Let $e = (u, v)$ be a directed edge, and $i \in [1,p]$

$$
\begin{aligned}
x_e &= \sum_{\sigma_1,\sigma_2 \in \{0,1\}^{p+q}:\sigma_1 \not\leq \sigma_2} z_{e,\sigma_1\sigma_2} && (2.37)\\
&\geq \sum_{\sigma_1,\sigma_2 \in \{0,1\}^{p+q}:\sigma_1[i]=1,\sigma_2[i]=0} z_{e,\sigma_1\sigma_2} && (2.38)\\
&= \sum_{\sigma_1 \in \{0,1\}^{p+q}:\sigma_1[i]=1} z_{u,\sigma_1} && (2.39)\\
&\quad - \sum_{\sigma_1,\sigma_2 \in \{0,1\}^{p+q}:\sigma_1[i]=1,\sigma_2[i]=1} z_{e,\sigma_1\sigma_2} && (2.40)\\
&\geq \sum_{\sigma_1 \in \{0,1\}^{p+q}:\sigma_1[i]=1} z_{u,\sigma_1} && (2.41)\\
&\quad - \sum_{\sigma_2 \in \{0,1\}^{p+q}:\sigma_2[i]=1} z_{v,\sigma_2} && (2.42)\\
&= \sum_{\sigma \in \{0,1\}^{p+q}:\sigma[i]=1} z_{u,\sigma} - \sum_{\sigma \in \{0,1\}^{p+q}:\sigma[i]=1} z_{v,\sigma} && (2.43)
\end{aligned}
$$

  If edge e has infinite weight, then $x_e = 0$ and $\sum_{\sigma \in \{0,1\}^{p+q}:\sigma[i]=1} z_{u,\sigma} \leq \sum_{\sigma \in \{0,1\}^{p+q}:\sigma[i]=1} z_{v,\sigma}$

- We prove the following two statements which in turn imply that for $\sigma \in \{0,1\}^p, \sigma' \in \{0,1\}^q$, if $J_\sigma \neq Y_i$, then $z_{a_i,\sigma\cdot\sigma'} = 0$.

$$
\begin{aligned}
\forall j \in Y_i, \sum_{\sigma \in \{0,1\}^p \sigma' \in \{0,1\}^q:\sigma[j]=1} z_{a_i,\sigma\cdot\sigma'} &= 1 && (2.44)\\
\forall j \in [1,p] \setminus Y_i, \sum_{\sigma \in \{0,1\}^p \sigma' \in \{0,1\}^q:\sigma[j]=1} z_{a_i,\sigma\cdot\sigma'} &= 0 && (2.45)
\end{aligned}
$$

  Let $j \in Y_i$, then by construction of G, there exists an infinite weight edge from a_j to a_i. Using the result from previous part we get $\sum_{\sigma \in \{0,1\}^p \sigma' \in \{0,1\}^q:\sigma[j]=1} z_{a_i,\sigma\cdot\sigma'} \geq \sum_{\sigma \in \{0,1\}^p \sigma' \in \{0,1\}^q:\sigma[j]=1} z_{a_j,\sigma\cdot\sigma'}$.

LABEL-LP enforces that term on the right side is lower bounded by 1 (a_j reachable from itself). Hence, term on the left side is lower bounded by 1. Since, it is also upper bounded by 1, it must be equal to 1.

Let $j \in [1, p] \setminus Y_i$. By definition of Y_i, we have $N_H^+(a_j) \not\subseteq N_H^+(a_i)$. That is, there exists $j' \in [1, q]$ such that $a_j b_{j'} \in E_H$ and $a_i b_j \notin E_H$. Since $a_j b_{j'} \in E_H$, LABEL-LP enforces that

$$\sum_{\sigma \in \{0,1\}^p \sigma' \in \{0,1\}^q : \sigma[j]=1} z_{b_{j'}, \sigma \cdot \sigma'} = 0 \tag{2.46}$$

Also, we have $a_i b_{j'} \notin E_H$ and hence, there is an infinite weight edge from a_i to $b_{j'}$ in G. Applying the result from previous part, we get $\sum_{\sigma \in \{0,1\}^p \sigma' \in \{0,1\}^q : \sigma[j]=1} z_{a_i, \sigma \cdot \sigma'} \leq \sum_{\sigma \in \{0,1\}^p \sigma' \in \{0,1\}^q : \sigma[j]=1} z_{b_{j'}, \sigma \cdot \sigma'} = 0$

Next, to prove that $z'_{a_i, \sigma} = 1$ if $J_\sigma = Y_i$ and 0 otherwise, we argue as follows:

$$\begin{aligned}
1 &= \sum_{\sigma \in \{0,1\}^p, \sigma' \in \{0,1\}^q} z_{a_i, \sigma \cdot \sigma'} && (2.47)\\
&= \sum_{\sigma \in \{0,1\}^p : J_\sigma = Y_i} \sum_{\sigma' \in \{0,1\}^q} z_{a_i, \sigma \cdot \sigma'} && (2.48)\\
&\quad + \sum_{\sigma \in \{0,1\}^p : J_\sigma \neq Y_i} \sum_{\sigma' \in \{0,1\}^q} z_{a_i, \sigma \cdot \sigma'} && (2.49)\\
&= \sum_{\sigma \in \{0,1\}^p : J_\sigma = Y_i} z'_{a_i, \sigma} && (2.50)
\end{aligned}$$

- Again, we prove the following two statements which in turn implies that for $\sigma \in \{0,1\}^p, \sigma' \in \{0,1\}^q$ if $J_\sigma \neq Z_j$, then $z_{b_j, \sigma \cdot \sigma'} = 0$:

$$\begin{aligned}
\forall i \in Z_j, \quad \sum_{\sigma \in \{0,1\}^p \sigma' \in \{0,1\}^q : \sigma[i]=1} z_{b_j, \sigma \cdot \sigma'} &= 1 && (2.51)\\
\forall i \in [1, p] \setminus Z_j, \quad \sum_{\sigma \in \{0,1\}^p \sigma' \in \{0,1\}^q : \sigma[i]=1} z_{b_j, \sigma \cdot \sigma'} &= 0 && (2.52)
\end{aligned}$$

Let $i \in Z_j$. Hence, there is an infinite weight directed edge from a_i to b_j in G. Applying the result from first part, we get $\sum_{\sigma \in \{0,1\}^p \sigma' \in \{0,1\}^q : \sigma[i]=1} z_{b_j, \sigma \cdot \sigma'} \geq \sum_{\sigma \in \{0,1\}^p \sigma' \in \{0,1\}^q : \sigma[i]=1} z_{a_i, \sigma \cdot \sigma'}$

LABEL-LP enforces that right side is lower bounded by 1 (a_i reachable from itself). Hence, left side is lower bounded by 1. It is also upper bounded by 1 and hence, it must be equal to 1.

Let $i \in [1, p] \setminus Z_j$. Then, $a_i b_j \in E_H$ and hence, from the constraint in LABEL-LP

$$\sum_{\sigma \in \{0,1\}^p \sigma' \in \{0,1\}^q : \sigma[i]=1} z_{b_j, \sigma \cdot \sigma'} = 0 \tag{2.53}$$

Next, to prove that $z'_{b_j,\sigma} = 1$ if $J_\sigma = Z_j$ and 0 otherwise, we argue as follows:

$$
\begin{aligned}
1 &= \sum_{\sigma\in\{0,1\}^p,\sigma'\in\{0,1\}^q} z_{b_j,\sigma\cdot\sigma'} && (2.54)\\
&= \sum_{\sigma\in\{0,1\}^p:J_\sigma=Z_j}\ \sum_{\sigma'\in\{0,1\}^q} z_{b_j,\sigma\cdot\sigma'} && (2.55)\\
&\quad + \sum_{\sigma\in\{0,1\}^p:J_\sigma\neq Z_j}\ \sum_{\sigma'\in\{0,1\}^q} z_{b_j,\sigma\cdot\sigma'} && (2.56)\\
&= \sum_{\sigma\in\{0,1\}^p:J_\sigma=Z_j} z'_{b_j,\sigma} && (2.57)
\end{aligned}
$$

- For an undirected edge $e = uv$, $x_e = \sum_{\sigma_1,\sigma_2\in\{0,1\}^{p+q}:\sigma_1\neq\sigma_2} z_{e,\sigma_1\sigma_2}$. Since, weight of e is infinite, x_e must be 0. Hence, $z_{e,\sigma_1\sigma_2} = 0$ if $\sigma_1 \neq \sigma_2$. For $\sigma_1 \in \{0,1\}^{p+q}$

$$
\begin{aligned}
z_{u,\sigma_1} &= \sum_{\sigma_2\in\{0,1\}^{p+q}} z_{e,\sigma_1\sigma_2} = z_{e,\sigma_1\sigma_1} && (2.58)\\
&= \sum_{\sigma_2\in\{0,1\}^{p+q}} z_{e,\sigma_2\sigma_1} = z_{v,\sigma_1} && (2.59)
\end{aligned}
$$

Let $t = (u) \in T_C$ s.t. $\psi_{T_C}(t) = \psi_{a_i}$. If $u = a_i$, then we have already proved that $z'_{u,\sigma} = 1$ if $J_\sigma = Y_i$ and 0 otherwise. If $u \neq a_i$, then there is an infinite weight undirected edge between u and a_i in G. Hence, $z'_{u,\sigma} = z'_{a_i,\sigma}$ for all $\sigma \in \{0,1\}^p$ and the result follows.

Proof: [Lemma 2.11] Next, we argue about the cost of the solution $\mathbf{z}'$. We assume here that $(\mathbf{x},\mathbf{z})$ has finite cost. For a constraint $t \in T_C$, the cost according to $\mathbf{z}'$ is $w_{T_C}(t)\sum_{\alpha\in L^{|t|}} z'_{t,\alpha}\cdot\psi_t(\alpha)$. We consider four cases based on the type of t.

- t corresponds to constraint of the form $\psi_{a_i}(v)$. As argued in Lemma 2.12, then $z'_{t,\sigma} = z_{v,\sigma} = 0$ if $J_\sigma \neq Y_i$ and 1 if $J_\sigma = Y_i$. On the other hand, $\psi_{a_i}(\sigma) = 0$ if $J_\sigma = Y_i$ and ∞ if $J_\sigma \neq Y_i$. Hence, $z'_{t,\sigma}\psi_{a_i}(\sigma) = 0$ for all σ. Therefore, this constraint contributes zero to the cost.
- t corresponds to constraint of the form $\psi_{b_j}(v)$. From Lemma 2.12, $z'_{t,\sigma} = z'_{v,\sigma} = 0$ if $J_\sigma \neq Z_j$ and 1 if $J_\sigma = Z_j$. And $\psi_{b_j}(\sigma) = 0$ if $J_\sigma = Z_j$ and ∞ if $J_\sigma \neq Z_j$. Hence, $z'_{t,\sigma}\psi_{b_j}(\sigma) = 0$ for all σ. Therefore, the contribution of this constraint is zero.
- t corresponds to constraint $\mathcal{C}(u,v)$. This corresponds to a directed edge $e_t = (u,v)$ in G and the cost paid by $(\mathbf{x},\mathbf{z})$ is x_{e_t}. The cost for t in $\mathbf{z}'$ is given by:

$$\sum_{\sigma_1,\sigma_2\in\{0,1\}^p} z'_{t,\sigma_1\sigma_2}\cdot \mathcal{C}(\sigma_1,\sigma_2) \tag{2.60}$$
$$= \sum_{\sigma_1,\sigma_2\in\{0,1\}^p:\sigma_1\not\leq\sigma_2} z'_{t,\sigma_1\sigma_2} \tag{2.61}$$
$$= \sum_{\sigma_1,\sigma_2\in\{0,1\}^p:\sigma_1\not\leq\sigma_2}\ \sum_{\sigma',\sigma''\in\{0,1\}^q} z_{e_t,\sigma_1\cdot\sigma'\sigma_2\cdot\sigma''} \tag{2.62}$$
$$\leq \sum_{\sigma_1,\sigma_2\in\{0,1\}^p,\sigma',\sigma''\in\{0,1\}^q:\sigma_1\cdot\sigma'\not\leq\sigma_2\cdot\sigma''} z_{e_t,\sigma_1\cdot\sigma'\sigma_2\cdot\sigma''} \tag{2.63}$$
$$= x_{e_t}. \tag{2.64}$$

First equality follows from the fact that $C(\sigma_1,\sigma_2)=0$ if $\sigma_1\leq\sigma_2$ and 1 otherwise. Penultimate inequality follows because if $\sigma_1\not\leq\sigma_2$, then $\sigma_1\cdot\sigma'\not\leq\sigma_2\cdot\sigma''$ for any $\sigma',\sigma''\in\{0,1\}^q$.

- t corresponds to constraint $\mathrm{NAE}_2(u,v)$. This corresponds to an undirected edge $e_t=uv$ in G and the cost paid by $(\mathbf{x},\mathbf{z})$ is x_{e_t}. The cost for t in $\mathbf{z}'$ is given by:

$$\sum_{\sigma_1,\sigma_2\in\{0,1\}^p} z'_{t,\sigma_1\sigma_2}\cdot \mathrm{NAE}_2(\sigma_1,\sigma_2) \tag{2.65}$$
$$= \sum_{\sigma_1,\sigma_2\in\{0,1\}^p:\sigma_1\neq\sigma_2} z'_{t,\sigma_1\sigma_2} \tag{2.66}$$
$$= \sum_{\sigma_1,\sigma_2\in\{0,1\}^p:\sigma_1\neq\sigma_2}\ \sum_{\sigma',\sigma''\in\{0,1\}^q} z_{e_t,\sigma_1\cdot\sigma'\sigma_2\cdot\sigma''} \tag{2.67}$$
$$\leq \sum_{\sigma_1,\sigma_2\in\{0,1\}^p,\sigma',\sigma''\in\{0,1\}^q:\sigma_1\cdot\sigma'\neq\sigma_2\cdot\sigma''} z_{e_t,\sigma_1\cdot\sigma'\sigma_2\cdot\sigma''} \tag{2.68}$$
$$= x_{e_t}. \tag{2.69}$$

Combining the four cases, the total cost of the solution $\mathbf{z}'$ is equal to the cost of the binary constraints each of which corresponds to an edge in G with the same weight. From the above inequalities we see that the cost is at most $\sum_{e\in E_G} w_G(e)x_e$ which is the cost of $(\mathbf{x},\mathbf{z})$.

1.2. From BASIC-LP *to* LABEL-LP*:* Let $\mathbf{z}$ be a BASIC-LP solution to I_C. Let $\sigma_0=1^q$. We define a solution $(\mathbf{x}',\mathbf{z}')$ to LABEL-LP for I_M as follows:

- For $v\in V_C, \forall\sigma_1\in\{0,1\}^p,\sigma_2\in\{0,1\}^q$,

$$z'_{v,\sigma_1\cdot\sigma_2} = \begin{cases} z_{v,\sigma_1} & \sigma_2=\sigma_0 \\ 0 & otherwise \end{cases} \tag{2.70}$$

- For unary constraint $t = (u)$ s.t. $u \notin \{a_1, \dots, a_p, b_1, \dots, b_q\}$ and $\sigma_1, \sigma_2 \in \{0,1\}^p, \sigma_3, \sigma_4 \in \{0,1\}^q$,

$$z'_{e_t,\sigma_1\cdot\sigma_3\sigma_2\cdot\sigma_4} = \begin{cases} z_{u,\sigma_1} & \sigma_1 = \sigma_2, \sigma_3 = \sigma_4 = \sigma_0 \\ 0 & otherwise \end{cases} \tag{2.71}$$

- For binary constraint $t = (u,v) \in T_C$ such that $\psi_{T_C}(t) = \mathcal{C}$ or NAE_2, and $\sigma_1, \sigma_2 \in \{0,1\}^p, \sigma_3, \sigma_4 \in \{0,1\}^q$

$$z'_{e_t,\sigma_1\cdot\sigma_3\sigma_2\cdot\sigma_4} = \begin{cases} z_{t,\sigma_1\sigma_2} & \sigma_3 = \sigma_4 = \sigma_0 \\ 0 & otherwise \end{cases} \tag{2.72}$$

- The edge variables x'_e are induced by the z' variables. We explicitly write them down. For directed edge $e \in E_G$, $x'_e = \sum_{\sigma_1,\sigma_2\in\{0,1\}^p,\sigma_3,\sigma_4\in\{0,1\}^q:\sigma_1\cdot\sigma_3\not\leq\sigma_2\cdot\sigma_4} z'_{e,\sigma_1\cdot\sigma_3\sigma_2\cdot\sigma_4}$. For undirected edge $e \in E_G$, $x'_e = \sum_{\sigma_1,\sigma_2\in\{0,1\}^p,\sigma_3,\sigma_4\in\{0,1\}^q:\sigma_1\cdot\sigma_3\neq\sigma_2\sigma_4} z'_{e,\sigma_1\cdot\sigma_3\sigma_2\cdot\sigma_4}$

It is easy to check that $(\mathbf{x}', \mathbf{z}')$ is integral if $\mathbf{z}$ is integral.

Lemma 2.13. $(\boldsymbol{x}', \boldsymbol{z}')$ *is a feasible solution to* LABEL-LP *for* I_M.

Proof: It is easy to check that all the variables are non-negative and upper bounded by 1.

We show that the other constraints are satisfied one at a time. Recall that LABEL-LP considered here has a constraint for undirected edges in addition to the constraints showed in Fig 2.5. The label set for LABEL-LP is $\{0,1\}^{p+q}$ which we can write as $\{\sigma_1 \cdot \sigma_2 | \sigma_1 \in \{0,1\}^p, \sigma_2 \in \{0,1\}^q\}$.

Constraint 1: For each v, $\sum_{\sigma\in\{0,1\}^{p+q}} z'_{v,\sigma} = 1$

$$\sum_{\sigma_1\in\{0,1\}^p \sigma_2\in\{0,1\}^q} z'_{v,\sigma_1\cdot\sigma_2} = \sum_{\sigma_1\in\{0,1\}^p} z'_{v,\sigma_1\cdot\sigma_0} \tag{2.73}$$

$$= \sum_{\sigma_1\in\{0,1\}^p} z_{v,\sigma_1} = 1 \tag{2.74}$$

Constraint 2: For $\sigma_1 \in \{0,1\}^p, \sigma_2 \in \{0,1\}^q$, $z'_{a_i,\sigma_1\sigma_2} = 0$ if $\sigma_1[i] = 0$. And $z'_{b_j,\sigma_1\sigma_2} = 0$ if $\sigma_2[j] = 0$.

There is $t = (a_i) \in T_C$ such that $\psi_{T_C}(t) = \psi_{a_i}$. For $\mathbf{z}$ to be a finite valued solution, we must have $z_{t,\sigma_1} = z_{a_i,\sigma_1} = 0$ if $J_{\sigma_1} \neq Y_i$. Since, $i \in Y_i$, we have that $z_{a_i,\sigma_1} = 0$ if $\sigma_1[i] = 0$. And hence, $z'_{a_i,\sigma_1\sigma_2} = 0$ if $\sigma_1[i] = 0$.

For $v \in V_C$, $z'_{v,\sigma_1\sigma_2} = 0$ if $\sigma_2 \neq \sigma_0 = 1^q$. Hence, $z'_{v,\sigma_1\sigma_2} = 0$ if $\sigma_2[j] = 0$. In particular, $z'_{b_j,\sigma_1\sigma_2} = 0$ if $\sigma_2[j] = 0$.

Constraint 3: For $e = (u,v) \in E_G, \sigma_1 \in \{0,1\}^p, \sigma_3 \in \{0,1\}^q$, $z'_{u,\sigma_1\cdot\sigma_3} = \sum_{\sigma_2\in\{0,1\}^p \sigma_4\in\{0,1\}^q} z_{e,\sigma_1\cdot\sigma_3\sigma_2\cdot\sigma_4}$. If $\sigma_3 \neq \sigma_0$, then all the terms are zero and hence, the equality holds. Else, $\sigma_3 = \sigma_0$ and there are two types of edges:

- For $t = (u)$, $e = e_t$

$$\sum_{\sigma_2 \in \{0,1\}^p, \sigma_4 \in \{0,1\}^q} z'_{e,\sigma_1\cdot\sigma_0\sigma_2\cdot\sigma_4} \tag{2.75}$$

$$= \sum_{\sigma_2 \in \{0,1\}^p} z'_{e,\sigma_1\cdot\sigma_0\sigma_2\cdot\sigma_0} \tag{2.76}$$

$$= z_{u,\sigma_1} = z'_{u,\sigma_1\cdot\sigma_0} = z'_{u,\sigma_1\cdot\sigma_3} \tag{2.77}$$

- For $t = (u, v), e = e_t$,

$$\sum_{\sigma_2 \in \{0,1\}^p, \sigma_4 \in \{0,1\}^q} z'_{e,\sigma_1\cdot\sigma_0\sigma_2\cdot\sigma_4} \tag{2.78}$$

$$= \sum_{\sigma_2 \in \{0,1\}^q} z'_{e,\sigma_1\cdot\sigma_0\sigma_2\cdot\sigma_4} \tag{2.79}$$

$$= \sum_{\sigma_2 \in \{0,1\}^q} z_{e,\sigma_1\sigma_2} \tag{2.80}$$

$$= z_{u,\sigma_1} = z'_{u,\sigma_1\cdot\sigma_0} = z'_{u,\sigma_1\cdot\sigma_3} \tag{2.81}$$

Constraint 4: For $e = (u, v) \in E_G, \sigma_2 \in \{0,1\}^p, \sigma_4 \in \{0,1\}^q$, $z'_{v,\sigma_2\cdot\sigma_4} = \sum_{\sigma_1 \in \{0,1\}^p \sigma_3 \in \{0,1\}^q} z_{e,\sigma_1\cdot\sigma_3\sigma_2\cdot\sigma_4}$. Proof is similar to the previous part.

Constraint 5: For directed edge e, $x'_e - \sum_{\sigma_1,\sigma_3 \in \{0,1\}^p, \sigma_2\sigma_4 \in \{0,1\}^q : \sigma_1\cdot\sigma_3 \not\leq \sigma_2\cdot\sigma_4} z'_{e,\sigma_1\cdot\sigma_3\sigma_2\sigma_4} = 0$. This is true by definition of x'_e.

Constraint 6: For undirected edge e, $x'_e - \sum_{\sigma_1,\sigma_3 \in \{0,1\}^p, \sigma_2\sigma_4 \in \{0,1\}^q : \sigma_1\cdot\sigma_3 \neq \sigma_2\cdot\sigma_4} z'_{e,\sigma_1\cdot\sigma_3\sigma_2\sigma_4} = 0$. This is true as well from the definition of x'_e.

Lemma 2.14. *The cost $(\boldsymbol{x}', \boldsymbol{z}')$ is upper bounded by the cost of $\boldsymbol{z}$.*

Proof: Recall that $\sigma_0 = 1^q$. We consider three cases based on the type of edge e

- $e = e_t = (u, v)$ for constraint $\mathcal{C}(u, v)$. Then, $x'_{e_t} =$

$$\sum_{\sigma_1,\sigma_2 \in \{0,1\}^p, \sigma_3,\sigma_4 \in \{0,1\}^q : \sigma_1\cdot\sigma_3 \not\leq \sigma_2\cdot\sigma_4} z'_{e_t,\sigma_1\cdot\sigma_3\sigma_2\cdot\sigma_4} \tag{2.82}$$

$$= \sum_{\sigma_1,\sigma_2 \in \{0,1\}^p : \sigma_1\cdot\sigma_0 \not\leq \sigma_2\cdot\sigma_0} z'_{e_t,\sigma_1\cdot\sigma_0\sigma_2\cdot\sigma_0} \tag{2.83}$$

$$= \sum_{\sigma_1,\sigma_2 \in \{0,1\}^p : \sigma_1 \not\leq \sigma_2} z_{t,\sigma_1\sigma_2} \tag{2.84}$$

- $e = e_t = (u, v)$ for constraint $NAE_2(u, v)$. Then, $x'_{e_t} =$

$$\sum_{\sigma_1,\sigma_2\in\{0,1\}^p\sigma_3,\sigma_4\in\{0,1\}^q:\sigma_1\cdot\sigma_3\neq\sigma_2\cdot\sigma_4} z'_{e_t,\sigma_1\cdot\sigma_3\sigma_2\cdot\sigma_4} \tag{2.85}$$

$$= \sum_{\sigma_1,\sigma_2\in\{0,1\}^p:\sigma_1\cdot\sigma_0\neq\sigma_2\cdot\sigma_0} z'_{e_t,\sigma_1\cdot\sigma_0\sigma_2\cdot\sigma_0} \tag{2.86}$$

$$= \sum_{\sigma_1,\sigma_2\in\{0,1\}^p:\sigma_1\neq\sigma_2} z_{t,\sigma_1\sigma_2} \tag{2.87}$$

- $e = e_t = (u, a_i)$ or (u, b_j) for constraint $\psi_{a_i}(u)$ or $\psi_{b_j}(u)$. In such a case $z'_{e_t,\sigma_1\cdot\sigma_3\sigma_2\cdot\sigma_4}$ is non-zero only if $\sigma_1 = \sigma_2, \sigma_3 = \sigma_4 = \sigma_0$. Hence, $x'_{e_t} =$

$$\sum_{\sigma_1,\sigma_2\in\{0,1\}^p,\sigma_3,\sigma_4\in\{0,1\}^q:\sigma_1\cdot\sigma_3\neq\sigma_2\sigma_4} z'_{e_t,\sigma_1\cdot\sigma_3\sigma_2\cdot\sigma_4} = 0 \tag{2.88}$$

Combining the above three facts we get the following. First, infinite any infinite weight edge e in G has $x'_e = 0$. For any finite weight edge x'_e is the same as the fractional cost paid by the corresponding finite weight binary constraint in I_C. Hence, cost of $(\mathbf{x}', \mathbf{z}')$ is upper bounded by cost of $\mathbf{z}$.

2. Reduction from DIR-MULTICUT-H *to* MIN-β-CSP Let the DIR-MULTICUT-H instance be $I_M = (G = (V_G, E_G, w_G : E_G \to R^+), (S \cup T, E_H))$. Recall that the supply graph satisfies assumptions I, II, and III. We reduce it an equivalent MIN-β-CSP instance $I_C = (V_C, T_C, \psi_{T_C} : T_C \to \beta_H, w_{T_C} : T_C \to R^+)$ as follows.

- Vertex Set $V_C = V_G$.
- T_C, ψ_{T_C}, w_{T_C} are defined as follows:
  - For every $a_i \in S$, add a tuple $t = (a_i)$ in T_C with $\psi_{T_C}(t) = \psi_{a_i}$ and $w_{T_C}(t) = 1$.
  - For every $b_j \in T$, add a tuple $t = (b_j)$ in T_C with $\psi_{T_C}(t) = \psi_{b_j}$ and $w_{T_C}(t) = 1$.
  - For every directed edge $e = (u, v) \in E_G$, add a tuple $t = (u, v)$ in T_C with $\psi_{T_C}(t) = \mathcal{C}$ and $w_{T_C}(t) = w_G(e)$.
  - For every undirected edge $e = uv \in E_G$, add a tuple $t = (u, v)$ in T_C with $\psi_{T_C}(t) = \mathrm{NAE}_2$ and $w_{T_C}(t) = w_G(e)$.

The proof of equivalence between LABEL-LP for I_M and BASIC-LP for I_C is essentially identical to the proof for the reduction in the other direction and hence we omit it.

This finishes the proof of Theorem 2.14.

2.4.4 Hardness for Non-bipartite Demand graphs

Here we prove Theorem 2.5 on the hardness of approximation of DIR-MULTICUT-H when H is fixed and may not be bipartite. Let γ_H denote the hardness of approximation for DIR-MULTICUT-H. Recall that α_H is the worst-case flow-cut gap for DIR-MULTICUT-H.

Let the demand graph be H with 2^p vertices, $V_H = \{s_\sigma \mid \sigma \in \{0,1\}^p\}$. If number of vertices is not a power of two, then we can add dummy isolated vertices without changing the problem. We find $r = 2p$ subgraphs $H_1, \ldots, H_r$ such that $H = H_1 \cup \cdots \cup H_r$ and

- Each H_i is a directed bipartite graph.
- $\alpha_H \leq \sum_{i=1}^r \alpha_{H_i}$.
- For $1 \leq i \leq r$, there is an approximation preserving reduction from DIR-MULTICUT-H_i to DIR-MULTICUT-H. Hence, $\gamma_H \geq \gamma_{H_i}$.

Since, H_i is bipartite, Theorem 2.11 implies, under UGC, that $\gamma_{H_i} \geq \alpha_{H_i} - \varepsilon$. Since, $\gamma_H \geq \gamma_{H_i}$ for all $i \in [1, r]$, we have $\gamma_H \geq \frac{1}{r}\sum_{i=1}^r \gamma_{H_i}$. Therefore,

$$\gamma_H \geq \frac{1}{r}\sum_{i=1}^{r}\gamma_{H_i} \geq \frac{1}{r}\sum_{i=1}^{r}(\alpha_{H_i} - \varepsilon) \geq \frac{1}{r}\alpha_H - \varepsilon. \tag{2.89}$$

Since $r = 2\lceil \log k \rceil$ where $k = |V_H|$, we obtain the proof of Theorem 2.5.

Next, we show how to construct H_i which satisfy the properties above. For each number $j \in [1, p]$, define $A_j = \{s_\sigma \mid \sigma \in \{0,1\}^p, \sigma(j) = 0\}, B_j = \{s_\sigma \mid \sigma \in \{0,1\}^p, \sigma(j) = 1\}$. Let H_{2j-1} be the subgraph of H with vertex set V_H and edge set containing edges of H with head in B_j and tail in A_j. H_{2j} be the subgraph of H with vertex set V_H and edge set containing edges of H with head in A_j and tail in B_j.

$$V_{H_{2j-1}} = V_{H_{2j}} = V_H \tag{2.90}$$
$$E_{H_{2j-1}} = \{(s_{\sigma_1}, s_{\sigma_2}) \in E_H \mid s_{\sigma_1} \in A_j, s_{\sigma_2} \in B_j\} \tag{2.91}$$
$$E_{H_{2j}} = \{(s_{\sigma_1}, s_{\sigma_2}) \in E_H \mid s_{\sigma_1} \in B_j, s_{\sigma_2} \in A_j\} \tag{2.92}$$

By construction, it is clear that H_{2j-1}, H_{2j} are bi-partite.

Lemma 2.15. *H_i as defined above satisfy the following properties:*

- $E_H = \cup_{i=1}^r E_{H_i}$.
- $\alpha_H \leq \sum_{i=1}^r \alpha_{H_i}$.
- *For* $i \in [1, r], \gamma_H \geq \gamma_{H_i}$.

Proof:

- Let $e = (s_{\sigma_1}, s_{\sigma_2}) \in E_H$. Since, there are no self-loops in H, there exists $j \in [1, p]$ such that either $\sigma_1[j] = 0, \sigma_2[j] = 1$ or $\sigma_1[j] = 1, \sigma_2[j] = 0$. In the first case, $e \in E_{H_{2j-1}}$ and in the second case $e \in E_{H_{2j}}$.

- Given a DIR-MULTICUT-H instance (G, H), idea is to solve (G, H_i) for $i \in [1, p]$. Let $I = (G, H)$ be a DIR-MULTICUT-H instance. Let $\mathbf{x}$ be the optimal solution to DISTANCE-LP on I. Let $I_i = (G, H_i)$ be the instance with the same supply graph G but demand graph H_i. It is easy to see that $\mathbf{x}$ is a feasible fractional solution to I_i since H_i is a subgraph of H. Since the worst-case integrality gap for DIR-MULTICUT-H_i is α_{H_i}, there is a set $E'_i \subseteq E_G$ such that $w(E'_i) \le \alpha_{H_i} w(\mathbf{x})$ and $G - E'_i$ disconnects all demand pairs in H_i. Clearly $\cup_i E'_i$ is a feasible integral solution to (G, H) since $H = \cup_i H_i$. The cost of $\cup_i E'_i$ is at most $(\sum_i \alpha_{H_i}) w(\mathbf{x})$. Since (G, H) was an arbitrary instance of DIR-MULTICUT-H, this proves that $\alpha_H \le \sum_i \alpha_{H_i}$.

- We prove that there is an approximation preserving reduction from DIR-MULTICUT-H_i to DIR-MULTICUT-H which in turn proves that $\gamma_H \ge \gamma_{H_i}$. Assume that $i = 2j - 1$ (case when $i = 2j$ is similar). Let (G, H_i) be a DIR-MULTICUT-H_i instance. G' is defined as follows:

  - $V_{G'} = V_G \cup A'_j \cup B'_j$ where $A'_j = \{s'_\sigma \mid s_\sigma \in A_j\}, B'_j = \{s'_\sigma \mid s_\sigma \in B_j\}$.
  - G' contains all the edges of G and an infinite edge from s'_σ to s_σ for every $s_\sigma \in A_j$ and infinite weight edge from s_σ to s'_σ for every $s_\sigma \in B_j$.

  Let H' be a demand graph with vertex s_σ of H renamed as s'_σ. Then, (G', H') is a DIR-MULTICUT-H instance. Note that for $s_\sigma \in A_j$, s'_σ in G' has no incoming edge and for $s_\sigma \in B_j$, s'_σ in G' has no outgoing edge. Hence, for DIR-MULTICUT instance (G', H'), we only need to separate $(s'_{\sigma_1}, s'_{\sigma_2})$ if $s_{\sigma_1} \in A_j, s_{\sigma_2} \in B_j$. Hence, DIR-MULTICUT instances (G, H_i) and (G', H') are equivalent.

2.5 Concluding Remarks

Our hardness result show that assuming UGC, hardness matches the flow-cut gap for a fixed bi-partite demand graph. However, there is a logarithmic loss for non-bipartite demand graphs. Can we remove this logarithmic factor?

Question 2.1. *Assuming UGC, is* DIR-MULTICUT-H *hard to approximate better than the worst-case flow-cut gap with demand graph H for non-bipartite H?*

Question 1.2 asks for the approximability and flow cut gap for UNDIR-SKEW-MULTICUT and DIR-SKEW-MULTICUT. In this chapter, we proved that UNDIR-SKEW-MULTICUT admits a constant factor approximation.

However, the question of determining the flow-cut gap is still open. Also, if this flow-cut gap turns out to be constant, we could ask the question about tK_2-free demand graphs as well.

Question 2.2. *Does there exists a function $f : N \to N$ such that the flow-cut gap for* UNDIR-MULTICUT *with tK_2-free demand graph is at most $f(t)$? If yes, what is the tightest possible bound on $f(t)$?*

For DIR-SKEW-MULTICUT, we proved that flow-cut gap is a lower bound on hardness of approximation. However, determining the exact flow-cut gap for DIR-SKEW-MULTICUT is still an open question. If this flow-cut gap turns out to be constant, we could also ask about flow-cut gap when demand graph does not contain matching of size k as an induced subgraph.

Question 2.3. *Let H be a fixed demand graph that does not contain a matching of size k as an induced subgraph. Is there a k-approximation for* DIR-MULTICUT-H*?*

Hardness results in this chapter rely on UGC. Can we prove these results assuming $P \neq NP$; specifically when demand graph is a collection of k disjoint directed edges.

Question 2.4. *Assuming $P \neq NP$, can we prove that* DIR-MULTICUT-H *does not admit $k-\epsilon$ approximation algorithm if H is a collection of k disjoint edges?*

Chapter 3

Resolving the approximability of Linear-3-Cut

In this chapter, we investigate the approximability of a special case of DIR-MULTICUT, namely Linear-3-cut. We formally distinguish the node weighted and edge weighted variants below:

(s,r,t)**-NODE-LIN-3-CUT**: The input is a directed graph $D=(V,E)$ with specified nodes $s,r,t \in V$ and node weights $w \in \mathbb{R}_+^{V\setminus\{r,s,t\}}$, and the goal is to find a minimum weight node set $U \subseteq V \setminus \{r,s,t\}$ such that $D[V-U]$ has no path from s to t, from s to r and from r to t.

(s,r,t)**-EDGE-LIN-3-CUT**: The input is a directed graph $D=(V,E)$ with specified nodes $s,r,t \in V$ and edge weights $w \in \mathbb{R}_+^{E}$, and the goal is to find a minimum weight edge set $F \subseteq E$ such that $D-F$ has no path from s to t, from s to r and from r to t.

We note that the unreachability requirements are determined by an ordering of the terminal nodes s, r, and t, and this is the origin for the terminology *linear* 3*-cut* [32]. By standard transformations, the edge weighted and the node weighted variants are equivalent. In this chapter, we focus on the node-weighted version as the results are easier to present for the node-weighted problem. It is also equivalent to DIR-SKEW-MULTICUT problem defined in Chapter 2 for $k=2$. DIR-SKEW-MULTICUT is a special case of DIR-MULTICUT where the demand graph H is a bipartite graph with k terminals $s_1,\ldots,s_k$ on one side and k terminals $t_1,\ldots,t_k$ on the other side: (s_i,t_j) is an edge in H iff $i \le j$. To reduce (s,r,t)-EDGE-LIN-3-CUT to DIR-SKEW-MULTICUT, set $s_1=s, s_2=r, t_1=r, t_2=t$. To reduce DIR-SKEW-MULTICUT to (s,r,t)-EDGE-LIN-3-CUT, set $s=s_1, t=t_2$, add a node r and edges $(t_1,r),(r,s_2)$.

If G is undirected, the (s,r,t)-NODE-LIN-3-CUT and (s,r,t)-EDGE-LIN-3-CUT reduces to node and edge weighted versions of multiway cut problem with three terminals. NODE-WT-MWC (node weighted, undirected multiway cut) with three terminals admits a 4/3-approximation [49] and does not admit a $(4/3-\epsilon)$-approximation for any constant $\epsilon > 0$ assuming the Unique Games Conjecture (UGC) [31]; EDGE-WT-MWC (edge weighted, undirected multiway cut) with three terminals admits a 12/11-approximation [25,56] and does not admit a $(12/11-\epsilon)$-approximation for any constant $\epsilon > 0$ assuming UGC [25,56,68]. In this chapter, we prove such tight approximability results for (s,r,t)-NODE-LIN-3-CUT.

On the hardness side, (s,r,t)-NODE-LIN-3-CUT is NP-hard and has no $(4/3-\epsilon)$-approximation assuming UGC by an approximation-preserving reduction from NODE-WT-MWC with three terminals. Bidirect all edges and add new nodes s,r,t with edges $s \to t_1, t_2 \to r \to t_2, t_3 \to t$. On the approximability side, (s,r,t)-NODE-LIN-3-CUT admits a simple combinatorial 2-approximation: Return union of min $\{s\}-\{r,t\}$

cut and min $\{r\} - \{t\}$ cut. Our next theorem closes this gap and prove that the optimal bound is $\sqrt{2}$.

Theorem 3.1. *There is a polynomial-time $\sqrt{2}$-approximation for (s,r,t)-NODE-LIN-3-CUT. Assuming UGC, (s,r,t)-NODE-LIN-3-CUT has no polynomial-time $(\sqrt{2}-\varepsilon)$-approximation for any constant $\varepsilon > 0$.*

Connections to other problems

Recall that an *out-r-arborescence* (similarly, an *in-r-arborescence*) in a directed graph is a minimal subset of edges such that every node has a unique path from r (to r) in the subgraph induced by the edges.

r-INOUT-NODE-BLOCKER: The input is a node-weighted directed graph with a specified terminal node r and the goal is to find a minimum weight set of non-terminal nodes whose removal ensures that the resulting graph has no out-r-arborescence *and* no in-r-arborescence.

We show an approximation-preserving equivalence between r-INOUT-NODE-BLOCKER and (s,r,t)-NODE-LIN-3-CUT which in turn, resolves the approximability of r-INOUT-NODE-BLOCKER.

Theorem 3.2. *There is a polynomial-time $\sqrt{2}$-approximation for r-INOUT-NODE-BLOCKER. Assuming UGC, r-INOUT-NODE-BLOCKER has no polynomial-time $(\sqrt{2}-\epsilon)$-approximation for any constant $\epsilon > 0$.*

In the $\{s,t\}$-EDGE-BICUT problem the input is an edge-weighted directed graph and two specified nodes s and t, and the goal is to find a smallest weight subset of edges whose deletion ensures that s cannot reach t *and* t cannot reach s in the resulting graph.

In this chapter, we consider a global variant where the node t is not fixed.

$\{s,*\}$-EDGE-BICUT: The input is an edge-weighted directed graph with a specified node s, and the goal is to find a smallest weight subset of edges whose deletion ensures that the resulting graph has a node t such that s cannot reach t and t cannot reach s.

It is known that $\{s,*\}$-EDGE-BICUT is NP-hard, admits an efficient 2-approximation, and does not admit an efficient $(4/3-\epsilon)$-approximation for any constant $\epsilon > 0$ assuming UGC [9]. We show a reduction from (s,r,t)-EDGE-LIN-3-CUT to $\{s,*\}$-EDGE-BICUT, thus improving the hardness known for $\{s,*\}$-EDGE-BICUT.

Theorem 3.3. *Assuming UGC, $\{s,*\}$-EDGE-BICUT has no polynomial-time $(\sqrt{2}-\epsilon)$-approximation for any constant $\epsilon > 0$.*

Organization: Section 3.1 gives an overview of the techniques in this chapter. We present the upper bound of Theorem 3.1 in Section 3.2 and the integrality gap instances leading to the lower bound of Theorem 3.1 in Section 3.3. We obtain tight approximation results for r-INOUT-NODE-BLOCKER (Theorem 3.2) and improved inapproximability results for $\{s,*\}$-EDGE-BICUT (Theorem 3.3) in Section 3.4.

3.1 Techniques

Both the algorithm and the hardness results are based on proving bounds on the integrality gap of DISTANCE-LP. We briefly remark on some of the salient features of our results.

Approximation: Our main contribution for the upper bound is an analysis exhibiting the tight approximation factor for a natural rounding scheme. A natural rounding scheme is to take the best of the following two alternatives: (i) first ensure that s and r cannot reach t by suitably rounding the LP-solution to obtain a node set K_1 to be removed, and then find a minimum $s \to r$ directed cut K_2 in the graph obtained after deleting K_1, and return $K_1 \cup K_2$; (ii) first ensure that s cannot reach r and t by suitably rounding the LP-solution to obtain a node set K_1 to be removed, and then find a minimum $r \to t$ directed cut K_2 in the graph obtained after deleting K_1, and return $K_1 \cup K_2$. We note that in both alternatives, the first step can be implemented by standard *deterministic ball-cut rounding schemes*[1] while the second step can be solved exactly in polynomial time. The main technical challenge lies in analyzing the approximation factor of such a *best of alternatives rounding scheme* where the second step in each alternative depends on the first. We overcome this challenge by showing that a single-step randomized ball-cut rounding scheme already achieves the desired expected value. The distribution underlying our single-step scheme turns out to be extremely non-trivial in nature (as it is not a simple distribution like uniform or geometric). In the proofs, we derive the distribution with the goal of obtaining the best approximation factor instead of stating the distribution upfront and bounding the approximation factor.

Inapproximability: It is known that the inapproximability factor under UGC for (s,r,t)-NODE-LIN-3-CUT is identical to the integrality gap of a natural distance-based LP (Theorem 2.11). We construct a sequence of instances such that the sequence of integrality gaps of the distance-based LP converges to $\sqrt{2}$. Our gap instances are also non-trivial and can be viewed as a weighted graph sequence converging to a kind of "graph limit structure" having irrational weights. While irrational gap instances for semi-definite programming relaxations of natural combinatorial optimization problems are known to exist (e.g., the max-cut problem [50, 60]), we are unaware of irrational gap instances for natural LP-relaxations of natural combinatorial optimization problems besides the one studied in this work.

3.2 $\sqrt{2}$-approximation algorithm

Let $D = (V, E)$ be an input directed graph with specified nodes $s, r, t \in V$, and node weights $w \in \mathbb{R}_+^{V\setminus\{s,r,t\}}$. The (s,r,t)-NODE-LIN-3-CUT problem asks for a minimum weight node set $U \subseteq V \setminus \{s,r,t\}$ such that $D[V-U]$ has no path from s to t, from s to r, and from r to t. The collection of feasible solutions remains the same if we add the arcs $t \to r$ and $r \to s$ to the directed graph. In the rest of this

[1] Pick $\theta \in (0,1)$ and set K_1 to be the set of nodes which have incoming (outgoing) arcs to nodes which are within a distance θ from the terminal(s) of interest. Since there are only polynomially many θ values of interest, the best solution can be obtained in polynomial time.

section, we assume that these arcs are present in D.

For a subset $U \subseteq V$, let us denote $w(U) := \sum_{u \in U} w_u$. For nodes $u, v \in V$, let $\mathcal{P}_{uv}$ denote the set of all directed paths from u to v in D. For $x \in \mathbb{R}_+^V$ and a path P in D, we define $x(P) := \sum_{v \in V(P)} x_v$. For $u, v \in V$, let $\mathsf{dist}_x(u, v) := \min\{x(P) : P \in \mathcal{P}_{uv}\}$. DISTANCE-LP for (s, r, t)-NODE-LIN-3-CUT is the following:

$$\begin{aligned} \min w^T x & \qquad \text{(DISTANCE-LP)} \\ x \in \mathbb{R}_+^V & \\ \mathsf{dist}_x(s,t),\ \mathsf{dist}_x(s,r),\ \mathsf{dist}_x(r,t) \geq 1 & \\ x_s = x_r = x_t = 0. & \end{aligned}$$

This LP is solvable in polynomial time since separation amounts to finding shortest paths. If x is a feasible solution to DISTANCE-LP, then there is a feasible solution x' to DISTANCE-LP such that $x'_v \leq x_v$ for every $v \in V$ and moreover:

$$\text{if } x'_v > 0, \text{ then } \mathsf{dist}_{x'}(r, v) + \mathsf{dist}_{x'}(v, r) \leq 1 + x'_v. \tag{3.1}$$

To achieve this property, we observe that if $x_v > 0$ and $\mathsf{dist}_x(r, v) + \mathsf{dist}_x(v, r) > 1 + x_v$, then $x(P) > 1$ for all $P \in \mathcal{P}_{st} \cup \mathcal{P}_{sr} \cup \mathcal{P}_{rt}$ that contains v. Indeed, for any such path P, there is a subset F of arcs from the set of arcs $\{t \to r, r \to s\}$ such that $F \cup P$ is the concatenation of $P_1 \in \mathcal{P}_{rv}$ and $P_2 \in \mathcal{P}_{vr}$; therefore, $x(P) = x(P_1) + x(P_2) - x_v > 1$. This means that we can decrease x_v until the property is satisfied.

Let x be a feasible solution to DISTANCE-LP that satisfies (3.1). We present an algorithm that, given x as input, constructs in polynomial time a feasible solution U to (s, r, t)-NODE-LIN-3-CUT that satisfies $w(U) \leq \sqrt{2} w^T x$. The algorithm itself is a simple and natural deterministic ball-cut scheme, described below. The main novelty is the proof of the approximation ratio, which is obtained by considering a weaker, randomized ball-cut algorithm.

For a node $u \in V$ and $0 < \theta \leq 1$, let

$$B^{out}(u, \theta) := \{v \in V : \mathsf{dist}_x(u, v) < \theta\}, \tag{3.2}$$
$$S^{out}(u, \theta) := \{v \in V : \theta \in (\mathsf{dist}_x(u, v) - x_v, \mathsf{dist}_x(u, v)]\}, \tag{3.3}$$
$$B^{in}(u, \theta) := \{v \in V : \mathsf{dist}_x(v, u) < \theta\}, \tag{3.4}$$
$$S^{in}(u, \theta) := \{v \in V : \theta \in (\mathsf{dist}_x(v, u) - x_v, \mathsf{dist}_x(v, u)]\}. \tag{3.5}$$

One can think of $B^{out}(u, \theta)$ as the open ball of radius θ around u with respect to the distances from u, and $S^{out}(u, \theta)$ can be thought of as the boundary of $B^{out}(u, \theta)$. The sets $B^{in}(u, \theta)$ and $S^{in}(u, \theta)$ are analogous, but with respect to distances *to* u (as opposed to distances *from* u). We note that $S^{out}(u, \theta)$ and $S^{in}(u, \theta)$ cannot contain nodes v with $x_v = 0$. Furthermore, the presence of edges $r \to s$ and $t \to r$

implies that $s \in B^{out}(r,\theta)$ and $t \in B^{in}(\theta)$.

Claim 3.1. *For any $\theta \in (0,1]$, there exists $\theta' \in (0,1]$ such that $S^{out}(r,\theta') = S^{out}(r,\theta)$, and $\theta' = \mathsf{dist}_x(r,v)$ or $\theta' = \mathsf{dist}_x(r,v) - x_v$ for some $v \in V$. A similar statement holds for $S^{in}(r,\theta)$.*

Proof: Let $\theta' = \min\{\gamma : \gamma \geq \theta,\ \gamma = \mathsf{dist}_x(r,v) \text{ or } \gamma = \mathsf{dist}_x(r,v) - x_v \text{ for some } v \in V\}$. The minimum is chosen from a non-empty set because $\mathsf{dist}_x(r,t) = 1$. Now it is easy to verify that $S^{out}(r,\theta) = S^{out}(r,\theta')$.

As a consequence, there are at most $2n$ distinct sets of the form $S^{out}(r,\theta)$ (where $\theta \in (0,1]$). The deterministic ball-cut scheme is based on enumerating these and is given in Figure 3.1.

Deterministic Ball-Cut Algorithm for (s,r,t)-NODE-LIN-3-CUT

Input: feasible solution x to DISTANCE-LP that satisfies (3.1)
Compute distances $\mathsf{dist}_x(r,v)$ and $\mathsf{dist}_x(v,r)$ for every $v \in V$
$U := V \setminus \{r,s,t\}$
for every $v \in V$ **do**
  for every $\theta \in (0,1] \cap \{\mathsf{dist}_x(r,v), \mathsf{dist}_x(r,v) - x_v\}$ **do**
    $K_1 := S^{out}(r,\theta)$
    Find minimum weight $s \to r$ cut K_2 in $D[V \setminus K_1]$
    if $w(K_1 \cup K_2) < w(U)$ **then** $U := K_1 \cup K_2$
  for every $\theta \in (0,1] \cap \{\mathsf{dist}_x(v,r), \mathsf{dist}_x(v,r) - x_v\}$ **do**
    $K_1 := S^{in}(r,\theta)$
    Find minimum weight $r \to t$ cut in $D[V \setminus K_1]$
    if $w(K_1 \cup K_2) < w(U)$ **then** $U := K_1 \cup K_2$
return U

Figure 3.1: Deterministic Ball-Cut Algorithm

The algorithm has the running time of $O(|V|)$ max flow computations. The following claim implies that the output is a feasible solution to (s,r,t)-NODE-LIN-3-CUT.

Claim 3.2. *If $\theta \in (0,1]$ and K is an $s \to r$ cut in $D[V \setminus S^{out}(r,\theta)]$, then $S^{out}(r,\theta) \cup K$ is a feasible solution to (s,r,t)-NODE-LIN-3-CUT. Similarly, if K is an $r \to t$ cut in $D[V \setminus S^{in}(r,\theta)]$, then $S^{in}(r,\theta) \cup K$ is a feasible solution.*

Proof: We prove the first part of the claim, the second part being similar. We observe that for every $u, v \in V$, every $P \in \mathcal{P}_{uv}$, and every two consecutive nodes w and w' in the direction of P, we have $\mathsf{dist}_x(u,w) \geq \mathsf{dist}_x(u,w') - x_{w'}$ and $\mathsf{dist}_x(w',v) \geq \mathsf{dist}_x(w,v) - x_w$.

We now show that every path $P \in \mathcal{P}_{rt}$ contains a node in $S^{out}(r,\theta)$. Let $P \in \mathcal{P}_{rt}$ with the nodes $w_0 := r, w_1, w_2, \ldots, w_k, w_{k+1} := t$ appearing in order. If $\mathsf{dist}_x(r,w_i) < \theta$ for every $i \in [k]$, then $\mathsf{dist}_x(r,w_k) < \theta \leq 1$ and hence, $\mathsf{dist}_x(r,t) < 1$, a contradiction. Hence, there exists a node w_i such that $\mathsf{dist}_x(r,w_i) \geq \theta$.

Pick the node w_i with the smallest index i such that $\mathsf{dist}_x(r, w_i) \geq \theta$. By the observation from the previous paragraph, we have $\mathsf{dist}_x(r, w_i) - x_{w_i} \leq \mathsf{dist}_x(r, w_{i-1}) < \theta$, where the second inequality is by the choice of the index i. Thus, $w_i \in S^{out}(r, \theta)$ and hence, the path P contains a node in $S^{out}(r, \theta)$.

Due to the presence of the edge $r \to s$ in the graph and the fact that $s, r \notin S^{out}(r, \theta)$, we also have that every path $P \in \mathcal{P}_{st}$ contains a node in $S^{out}(r, \theta)$. Now, let us consider a path $P \in \mathcal{P}_{sr}$ without any nodes in $S^{out}(r, \theta)$. Since K is an $s \to r$ cut in $D[V \setminus S^{out}(r, \theta)]$, P contains a node in K. This means that $S^{out}(r, \theta) \cup K$ is a feasible solution to (s, r, t)-NODE-LIN-3-CUT.

The difficulty in analyzing the approximation factor of the algorithm presented in Figure 3.1 is due to dependence of K_2 on the choice of K_1. We overcome this difficulty by abandoning the minimum weight cuts K_2 in favor of random ball cuts that are easier to analyze. To do this, we need to define two types of feasible solutions to (s, r, t)-NODE-LIN-3-CUT.

For $0 < \theta_1 \leq 1$ and $0 < \theta_2 \leq 1$, the **vertical T-shaped cut** $V(\theta_1, \theta_2)$ is defined as

$$V(\theta_1, \theta_2) := S^{out}(r, \theta_1) \cup (B^{out}(r, \theta_1) \cap S^{in}(r, \theta_2)), \tag{3.6}$$

while the **horizontal T-shaped cut** $H(\theta_1, \theta_2)$ is defined as

$$H(\theta_1, \theta_2) := S^{in}(r, \theta_2) \cup (B^{in}(r, \theta_2) \cap S^{out}(r, \theta_1)). \tag{3.7}$$

The name "T-shaped cut" comes from the observation that if each node v is represented in the plane by the square $(\mathsf{dist}_x(r, v) - x_v, \mathsf{dist}_x(r, v)] \times (\mathsf{dist}_x(v, r) - x_v, \mathsf{dist}_x(v, r)]$, then the cut consists of nodes whose square is intersected by two segments forming a rotated "T" shape (see Figure 3.2). We call a node set U to be **T-shaped** if $U = V(\theta_1, \theta_2)$ or $U = H(\theta_1, \theta_2)$ for some pair $0 < \theta_1, \theta_2 \leq 1$.

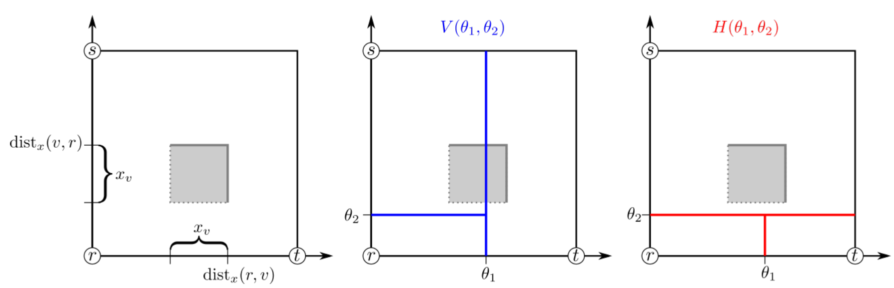


Figure 3.2: Representation of T-shaped cuts. Left: the square corresponding to node v. Center: v is in $V(\theta_1, \theta_2)$ because one of the blue lines intersects the square. Right: v is not in $H(\theta_1, \theta_2)$ because the red lines do not intersect the square.

Lemma 3.1. *The set $B^{out}(r, \theta_1) \cap S^{in}(r, \theta_2)$ is an $s \to r$ cut in $D[V \setminus S^{out}(r, \theta_1)]$, and $B^{in}(r, \theta_2) \cap S^{out}(r, \theta_1)$ is an $r \to t$ cut in $D[V \setminus S^{in}(r, \theta_2)]$.*

Proof: We prove only the first part of the claim, the proof of the second part being similar. Let us consider a path $P \in \mathcal{P}_{sr}$ without any nodes in $S^{out}(r, \theta_1)$. Let the nodes in P be $w_0 := s, w_1, w_2, \ldots, w_k, w_{k+1} := r$

appearing in order. We will show that $w_i \in B^{out}(r,\theta_1)$ for every $i \in \{1,\dots,k\}$ by induction on i. For the base case, owing to the presence of the edge $r \to s$ in the graph, we have that $\mathsf{dist}_x(r,w_0) - x_{w_0} = 0 < \theta_1$ and since $w_0 \notin S^{out}(r,\theta_1)$, it follows that $\mathsf{dist}_x(r,w_0) < \theta_1$. For the induction step, we have that $\mathsf{dist}_x(r,w_{i+1}) - x_{w_{i+1}} \le \mathsf{dist}_x(r,w_i) < \theta_1$, where the second inequality follows by induction hypothesis. Now, since $w_{i+1} \notin S^{out}(r,\theta_1)$, it follows that $\mathsf{dist}_x(r,w_{i+1}) < \theta_1$. Hence, all nodes of P are in $B^{out}(r,\theta_1)$.

We now show that at least one of the nodes in P should be in $S^{in}(r,\theta_2)$. If $\mathsf{dist}_x(w_i,r) < \theta_2$ for every $i \in [k]$, then $\mathsf{dist}_x(w_1,r) < \theta_2 \le 1$ and hence, $\mathsf{dist}_x(s,r) < 1$, a contradiction. Hence, there exists a node w_i such that $\mathsf{dist}_x(w_i,r) \ge \theta_2$. Pick the node w_i with the largest index i such that $\mathsf{dist}_x(w_i,r) \ge \theta_2$. We have $\mathsf{dist}_x(w_i,r) - x_{w_i} \le \mathsf{dist}_x(w_{i+1},r) < \theta_2$, where the second inequality is by the choice of the index i. Thus, $w_i \in S^{in}(r,\theta_2)$ and hence, the path P contains a node in $S^{in}(r,\theta_2)$. Consequently, P contains a node in $B^{out}(r,\theta_1) \cap S^{in}(r,\theta_2)$.

Corollary 3.1. *Every T-shaped cut is a feasible solution to (s,r,t)-*NODE-LIN-3-CUT*, and the weight of the cut found by the Deterministic Ball-Cut Algorithm is at most the minimum weight of a T-shaped cut.*

Proof: Feasibility follows directly from Lemma 3.1 and Claim 3.2. For the second statement, consider $V(\theta_1,\theta_2)$ and $H(\theta_1,\theta_2)$ for some $\theta_1,\theta_2 \in (0,1]$. By Claim 3.1, there exists $\theta' \in \{\mathsf{dist}_x(r,v'), \mathsf{dist}_x(r,v') - x_{v'}\}$ for some $v' \in V$ such that $S^{out}(r,\theta') = S^{out}(r,\theta_1)$, and there exists $\theta'' \in \{\mathsf{dist}_x(v'',r), \mathsf{dist}_x(v'',r) - x_{v''}\}$ for some $v'' \in V$ such that $S^{in}(r,\theta'') = S^{in}(r,\theta_2)$.

When the algorithm considers v' and θ', it finds a minimum weight $s \to r$ cut K_2 in $D[V \setminus S^{out}(r,\theta')]$. As $B^{out}(r,\theta_1) \cap S^{in}(r,\theta_2)$ is also an $s \to r$ cut in $D[V \setminus S^{out}(r,\theta')]$ by Lemma 3.1, $w(S^{out}(r,\theta') \cup K_2) \le w(S^{out}(r,\theta_1) \cup (B^{out}(r,\theta_1) \cap S^{in}(r,\theta_2))$.

When the algorithm considers v'' and θ'', it finds a minimum weight $r \to t$ cut K_1 in $D[V \setminus S^{in}(r,\theta'')]$. As $B^{in}(r,\theta_2) \cap S^{out}(r,\theta_1)$ is also an $r \to t$ cut in $D[V \setminus S^{in}(r,\theta'')]$ by Lemma 3.1, $w(S^{in}(r,\theta'') \cup K_1) \le w(S^{in}(r,\theta_2) \cup (B^{in}(r,\theta_2) \cap S^{out}(r,\theta_1))$.

We can bound the approximation factor of Deterministic Ball-Cut Algorithm by estimating the minimum weight of a T-shaped cut. We show that the latter differs from the cost of the DISTANCE-LP by a factor of at most $\sqrt{2}$.

Theorem 3.4. *There exists a T-shaped cut U such that $w(U) \le \sqrt{2} w^T x$.*

To prove Theorem 3.4, we will follow a probabilistic argument. We will exhibit a distribution over T-shaped cuts for which the expected weight satisfies the bound mentioned in Theorem 3.4. This distribution turns out to be non-trivial in nature (as it is not simply a uniform/geometric/exponential distribution). Instead of stating this distribution upfront and analyzing its approximation factor, we will derive the optimal distribution as a natural consequence of the following lemma, which provides a sufficient condition for achieving a certain approximation factor.

Lemma 3.2. *Let $\xi : [0,1]^2 \to \mathbb{R}_+$ be a function satisfying*

$$\forall a, b \in \mathbb{R}_+,\ a+b \le 1,$$
$$\int_0^1 (\xi(a,z)+\xi(b,z))\,\mathrm{d}z + \int_a^1 \xi(z,b)\,\mathrm{d}z + \int_b^1 \xi(z,a)\,\mathrm{d}z = 1. \quad (3.8)$$

*Let $\alpha := 2\int_0^1\int_0^1 \xi(z_1,z_2)\,\mathrm{d}z_1\,\mathrm{d}z_2$. Then, for any instance of (s,r,t)-*NODE-LIN-3-CUT*, there exists a T-shaped cut U such that*

$$w(U) \le \frac{1}{\alpha} w^T x. \quad (3.9)$$

Proof: We define a probability distribution on the set of T-shaped cuts by giving a weighing function $f : \{\mathrm{Ver}, \mathrm{Hor}\} \times [0,1]^2 \to \mathbb{R}_+$. For $(\theta_1,\theta_2) \in [0,1]^2$, let $f(\mathrm{Ver},\theta_1,\theta_2) := \xi(\theta_1,\theta_2)/\alpha$ and $f(\mathrm{Hor},\theta_1,\theta_2) := \xi(\theta_2,\theta_1)/\alpha$. For a T-shaped cut U, let

$$\mathsf{Pr}(U) := \int_{(\theta_1,\theta_2):V(\theta_1,\theta_2)=U} f(\mathrm{Ver},\theta_1,\theta_2)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2 + \int_{(\theta_1,\theta_2):H(\theta_1,\theta_2)=U} f(\mathrm{Hor},\theta_1,\theta_2)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2. \quad (3.10)$$

We mention that a node set U could be both a horizontal and a vertical T-shaped cut in which case, the probability mass for U comes from both integrals in the above sum. Furthermore $\mathsf{Pr}(\cdot)$ is a probability distribution supported over the set of T-shaped cuts because of the definition of α. Let U be a T-shaped cut chosen according to this distribution.

Claim 3.3. *For $v \in V \setminus \{r,s,t\}$, probability that v is in the chosen T-shaped cut U is at most x_v/α.*

Proof: We may assume that $x_v \neq 0$ since every vertex in a T-shaped cut necessarily has this property. Let $a := \mathsf{dist}_x(r,v)$ and $b := \mathsf{dist}_x(v,r)$. We recall that a vertical T-shaped cut $V(\theta_1,\theta_2)$ is defined as $S^{out}(r,\theta_1) \cup (B^{out}(r,\theta_1) \cap S^{in}(r,\theta_2))$. Thus, $V(\theta_1,\theta_2)$ contains the node v if and only if either (1) $a - x_v < \theta_1 \le a$, or (2) $a < \theta_1$ and $b - x_v < \theta_2 \le b$. Similarly, a horizontal T-shaped cut $H(\theta_1,\theta_2)$ contains the node v if and only if either (3) $b - x_v < \theta_2 \le b$, or (4) $b < \theta_2$ and $a - x_v < \theta_1 \le a$. Therefore, the probability of v being in a random T-shaped cut is at most

$$P := \frac{1}{\alpha}\Bigg(\int_{z_2=0}^{1}\int_{z_1=a-x_v}^{a} \xi(z_1,z_2)\,\mathrm{d}z_1\,\mathrm{d}z_2 + \int_{z_2=b-x_v}^{b}\int_{z_1=a}^{1} \xi(z_1,z_2)\,\mathrm{d}z_1\,\mathrm{d}z_2 \quad (3.11)$$
$$+ \int_{z_2=b-x_v}^{b}\int_{z_1=0}^{1} \xi(z_2,z_1)\,\mathrm{d}z_1\,\mathrm{d}z_2 + \int_{z_2=b}^{1}\int_{z_1=a-x_v}^{a} \xi(z_2,z_1)\,\mathrm{d}z_1\,\mathrm{d}z_2\Bigg) \quad (3.12)$$

By change of variables, we have that

$$P = \frac{1}{\alpha}\int_{y=0}^{x_v}\left(\int_{z=0}^{1}\xi(a-y,z)\,\mathrm{d}z + \int_{z=a}^{1}\xi(z,b-x_v+y)\,\mathrm{d}z \right. \tag{3.13}$$

$$\left. + \int_{z=0}^{1}\xi(b-x_v+y,z)\,\mathrm{d}z + \int_{z=b}^{1}\xi(z,a-y)\,\mathrm{d}z\right)\mathrm{d}y. \tag{3.14}$$

For $0 \le y \le x_v$, we have $a-y \le a$ and $b-x_v+y \le b$. By assumption, $\xi(z_1,z_2)$ is non-negative in the domain. Therefore, we have

$$\int_{y=0}^{x_v}\int_{z=a}^{1}\xi(z,b-x_v+y)\,\mathrm{d}z\,\mathrm{d}y \le \int_{y=0}^{x_v}\int_{z=a-y}^{1}\xi(z,b-x_v+y)\,\mathrm{d}z\,\mathrm{d}y \tag{3.15}$$

and

$$\int_{y=0}^{x_v}\int_{z=b}^{1}\xi(z,a-y)\,\mathrm{d}z\,\mathrm{d}y \le \int_{y=0}^{x_v}\int_{z=b-x_v+y}^{1}\xi(z,a-y)\,\mathrm{d}z\,\mathrm{d}y. \tag{3.16}$$

Hence,

$$P \le \frac{1}{\alpha}\int_{y=0}^{x_v}\left(\int_{z=0}^{1}\xi(a-y,z)\,\mathrm{d}z + \int_{z=a-y}^{1}\xi(z,b-x_v+y)\,\mathrm{d}z \right. \tag{3.17}$$

$$\left. + \int_{z=0}^{1}\xi(b-x_v+y,z)\,\mathrm{d}z + \int_{z=b-x_v+y}^{1}\xi(z,a-y)\,\mathrm{d}z\right)\mathrm{d}y \tag{3.18}$$

$$= \frac{1}{\alpha}\int_{y=0}^{x_v} 1\,\mathrm{d}y = \frac{x_v}{\alpha}, \tag{3.19}$$

where the equality at the beginning of the last row follows from (3.8), since for $0 \le y \le x_v$, we have $(a-y)+(b-x_v+y) = a+b-x_v = \mathsf{dist}_x(r,v)+\mathsf{dist}_x(v,r)-x_v \le 1$ by (3.1), and moreover $a-y \ge a-x_v = \mathsf{dist}_x(r,v)-x_v \ge 0$ and $b-x_v+y \ge b-x_v = \mathsf{dist}_x(v,r)-x_v \ge 0$ by the definition of $\mathsf{dist}_x(\cdot,\cdot)$.

Since every node v is in the random T-shaped cut with probability at most $\frac{x_v}{\alpha}$, expected weight of a random T-shaped cut is at most $\frac{w^T x}{\alpha}$. Therefore, there is a T-shaped cut U with $w(U) \le \frac{w^T x}{\alpha}$.

To prove the Theorem 3.4, it is enough by Lemma 3.2 to show the existence of a function $\xi : [0,1]^2 \to \mathbb{R}_+$ satisfying (3.8) for which

$$\int_0^1\int_0^1 \xi(z_1,z_2)\,\mathrm{d}z_1\,\mathrm{d}z_2 = \frac{1}{2\sqrt{2}}. \tag{3.20}$$

It turns out that such a function exists, but its structure is surprisingly complex. We define two regions where the function ξ will have positive values (see Figure 3.3).

$$\mathcal{R}_1 := \left\{(z_1, z_2) : \frac{1}{\sqrt{2}} < z_1, z_2 \le 1\right\},$$
$$\mathcal{R}_2 := \left\{(z_1, z_2) : \frac{\sqrt{2}-1}{\sqrt{2}} \le z_1 \le \frac{1}{\sqrt{2}},\ z_1 + z_2 \le 1\right\}.$$

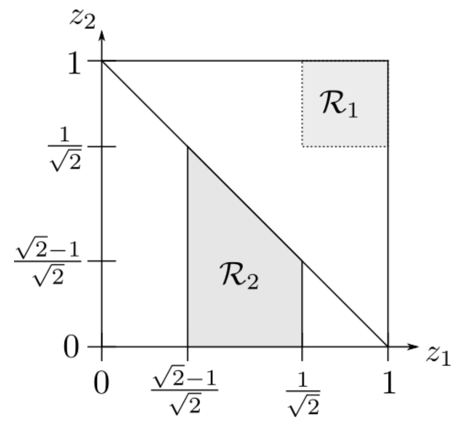


Figure 3.3: The regions $\mathcal{R}_1$ and $\mathcal{R}_2$.

Remark. The reason for this restriction on the support of ξ will become apparent in Section 3.3, where we present an infinite sequence of node-weighted graphs for which the integrality gap of DISTANCE-LP converges to $\sqrt{2}$. It can be seen that $\mathcal{R}_1 \cup \mathcal{R}_2$ consists of the pairs (z_1, z_2) for which the weight of the vertical T-shaped cut $V(z_1, z_2)$ (based on the optimal LP solution x) converges to 1 in the graph sequence. Informally, the region $\mathcal{R}_1 \cup \mathcal{R}_2$ is the region where the complementary slackness conditions allow positive density, if we consider the "limit" of the weighted graph sequence defined in Section 3.3. However, this is not the usual notion of graph limit, so we do not formalize this statement as it is not necessary for the proof.

Proof:[Theorem 3.4] To prove the theorem, we define a function ξ with the above properties. The value of ξ is defined to be 0 for $(z_1, z_2) \in [0,1]^2 \setminus (\mathcal{R}_1 \cup \mathcal{R}_2)$. For every $(z_1, z_2) \in \mathcal{R}_1$, we set $\xi(z_1, z_2) := (\sqrt{2}+1)/\sqrt{2}$. In the region $\mathcal{R}_2$, the value of $\xi(z_1, z_2)$ will depend only on z_1. In particular, we will define a function $\zeta : [\frac{\sqrt{2}-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] \to \mathbb{R}_+$, and define $\xi(z_1, z_2)$ in the region $\mathcal{R}_2$ as

$$\xi(z_1, z_2) := \zeta(z_1). \tag{3.21}$$

Let us examine the properties that are sufficient to be satisfied by ζ in order for ξ to satisfy (3.8).

Claim 3.4. *Condition* (3.8) *is satisfied by ξ if the following hold for ζ:*

$$\zeta\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right) = 0, \tag{3.22}$$

$$(1-y)\zeta(y) + \int_y^{1-y} \zeta(z)\,\mathrm{d}z = \frac{1}{2} \quad \textit{if } \frac{\sqrt{2}-1}{\sqrt{2}} \le y \le \frac{1}{2}, \tag{3.23}$$

$$(1-y)\zeta(y) - \int_{1-y}^{y} \zeta(z)\,\mathrm{d}z = \frac{1}{2} \quad \textit{if } \frac{1}{2} \le y \le \frac{1}{\sqrt{2}}. \tag{3.24}$$

Proof: We consider several cases based on the values of a and b. Let $\Gamma(a, b)$ denote the LHS of (3.8).

By taking $y = \frac{\sqrt{2}-1}{\sqrt{2}}$ in (3.23) and substituting $\zeta(\frac{\sqrt{2}-1}{\sqrt{2}}) = 0$, we obtain that

$$\int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z = \frac{1}{2}. \tag{3.25}$$

Case 1(i): Suppose $a > \frac{1}{\sqrt{2}}$. Then $b \leq \frac{\sqrt{2}-1}{\sqrt{2}}$ since $a + b \leq 1$. Now,

$$\Gamma(a,b) = \int_0^1 \xi(a,z)\,\mathrm{d}z + \int_0^1 \xi(b,z)\,\mathrm{d}z + \int_a^1 \xi(z,b)\,\mathrm{d}z + \int_b^1 \xi(z,a)\,\mathrm{d}z \tag{3.26}$$

$$= \left(1+\frac{1}{\sqrt{2}}\right)\left(1-\frac{1}{\sqrt{2}}\right) + 0 + 0 + \left(1+\frac{1}{\sqrt{2}}\right)\left(1-\frac{1}{\sqrt{2}}\right) = 1. \tag{3.27}$$

Case 1(ii): Suppose $b > \frac{1}{\sqrt{2}}$. Proceeding similar to Case 1(i), we obtain $\Gamma(a,b) = 1$.

Case 2: Suppose $a \leq \frac{\sqrt{2}-1}{\sqrt{2}}$ and $b \leq \frac{\sqrt{2}-1}{\sqrt{2}}$. In this case, we have

$$\Gamma(a,b) = \int_0^1 \xi(a,z)\,\mathrm{d}z + \int_0^1 \xi(b,z)\,\mathrm{d}z + \int_a^1 \xi(z,b)\,\mathrm{d}z + \int_b^1 \xi(z,a)\,\mathrm{d}z \tag{3.28}$$

$$= 0 + 0 + 2\int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z = 1. \tag{3.29}$$

Case 3(i): Suppose $\frac{\sqrt{2}-1}{\sqrt{2}} < a \leq \frac{1}{\sqrt{2}}$ and $b \leq \frac{\sqrt{2}-1}{\sqrt{2}}$. Then

$$\Gamma(a,b) = \int_0^1 \xi(a,z)\,\mathrm{d}z + \int_0^1 \xi(b,z)\,\mathrm{d}z + \int_a^1 \xi(z,b)\,\mathrm{d}z + \int_b^1 \xi(z,a)\,\mathrm{d}z \tag{3.30}$$

$$= \int_0^{1-a} \zeta(a)\,\mathrm{d}z + 0 + \int_a^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{1-a} \zeta(z)\,\mathrm{d}z \tag{3.31}$$

$$= (1-a)\zeta(a) + \int_a^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{1-a} \zeta(z)\,\mathrm{d}z. \tag{3.32}$$

If $a \leq \frac{1}{2}$, then the RHS from (3.32) can be written as

$$\Gamma(a,b) = (1-a)\zeta(a) + \int_a^{1-a} \zeta(z)\,\mathrm{d}z + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z = 1 \tag{3.33}$$

by (3.23) and (3.25). If $a > \frac{1}{2}$, then the RHS from (3.32) can be written as

$$\Gamma(a,b) = (1-a)\zeta(a) - \int_{1-a}^{a} \zeta(z)\,\mathrm{d}z + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z = 1 \tag{3.34}$$

by (3.24) and (3.25).

Case 3(ii): Suppose $a \leq \frac{\sqrt{2}-1}{\sqrt{2}}$ and $\frac{\sqrt{2}-1}{\sqrt{2}} < b \leq \frac{1}{\sqrt{2}}$. Proceeding similar to Case 3(a), we obtain that $\Gamma(a,b) = 1$.

Case 4: Suppose $\frac{\sqrt{2}-1}{\sqrt{2}} < a \leq \frac{1}{\sqrt{2}}$ and $\frac{\sqrt{2}-1}{\sqrt{2}} < b \leq \frac{1}{\sqrt{2}}$. Moreover, we have $a + b \leq 1$.

$$\Gamma(a,b) = \int_0^1 \xi(a,z)\,\mathrm{d}z + \int_0^1 \xi(b,z)\,\mathrm{d}z + \int_a^1 \xi(z,b)\,\mathrm{d}z + \int_b^1 \xi(z,a)\,\mathrm{d}z \tag{3.35}$$

$$= \int_0^{1-a} \zeta(a)\,\mathrm{d}z + \int_0^{1-b} \zeta(b)\,\mathrm{d}z + \int_a^{1-b} \zeta(z)\,\mathrm{d}z + \int_b^{1-a} \zeta(z)\,\mathrm{d}z \tag{3.36}$$

$$= (1-a)\zeta(a) + (1-b)\zeta(b) + \int_a^{1-b} \zeta(z)\,\mathrm{d}z + \int_b^{1-a} \zeta(z)\,\mathrm{d}z. \tag{3.37}$$

We will assume that $a \leq b$ (the other case is similar). If $b \leq \frac{1}{2}$, then

$$\int_a^{1-b} \zeta(z)\,\mathrm{d}z + \int_b^{1-a} \zeta(z)\,\mathrm{d}z = \int_a^{1-a} \zeta(z)\,\mathrm{d}z + \int_b^{1-b} \zeta(z)\,\mathrm{d}z, \tag{3.38}$$

and hence $\Gamma(a,b) = 1$ follows from (3.23). If $b > \frac{1}{2}$, then, $a \leq 1-b < \frac{1}{2} \leq b$ and hence, we have

$$\int_a^{1-b} \zeta(z)\,\mathrm{d}z + \int_b^{1-a} \zeta(z)\,\mathrm{d}z = \int_a^{1-a} \zeta(z)\,\mathrm{d}z - \int_{1-b}^{b} \zeta(z)\,\mathrm{d}z. \tag{3.39}$$

Therefore, $\Gamma(a,b) = 1$ follows from (3.23) and (3.24).

We note that conditions (3.22)-(3.24) on ζ are also necessary for (3.8) to hold. This fact is not needed for the proof of Theorem 3.4, but we prove it to facilitate future investigations on our distribution.

Claim 3.5. *Condition* (3.8) *is satisfied by* ξ *if and only if the following hold for* ζ*:*

$$\zeta\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right) = 0, \tag{3.40}$$

$$(1-y)\zeta(y) + \int_y^{1-y} \zeta(z)\,\mathrm{d}z = \frac{1}{2} \quad \textit{if } \frac{\sqrt{2}-1}{\sqrt{2}} \leq y \leq \frac{1}{2}, \tag{3.41}$$

$$(1-y)\zeta(y) - \int_{1-y}^{y} \zeta(z)\,\mathrm{d}z = \frac{1}{2} \quad \textit{if } \frac{1}{2} \leq y \leq \frac{1}{\sqrt{2}}. \tag{3.42}$$

Proof: The direction showing that (3.40)-(3.42) imply (3.8) was already shown in Claim 3.4. We now argue the necessity of (3.40), (3.41) and (3.42).

To see the necessity of (3.41), we consider $a = b = y$ for $\frac{\sqrt{2}-1}{\sqrt{2}} \leq y \leq \frac{1}{2}$. For this choice of a and b, condition (3.8) necessitates that

$$
\begin{aligned}
1 &= 2\int_0^1 \xi(y,z)\,\mathrm{d}z + 2\int_y^1 \xi(z,y)\,\mathrm{d}z && (3.43)\\
&= 2\int_0^{1-y} \xi(y,z)\,\mathrm{d}z + 2\int_y^{1-y} \xi(z,y)\,\mathrm{d}z && (3.44)\\
&= 2\int_0^{1-y} \zeta(y)\,\mathrm{d}z + 2\int_y^{1-y} \zeta(z)\,\mathrm{d}z && (3.45)\\
&= 2(1-y)\zeta(y) + 2\int_y^{1-y} \zeta(z)\,\mathrm{d}z, && (3.46)
\end{aligned}
$$

which shows the necessity of (3.41). The second equation above is because $\xi(y,z) = \xi(z,y) = 0$ for $z > 1-y$ since $y \le 1/2$.

To see the necessity of (3.42), we consider $a = y$, $b = 1-y$ for some y such that $\frac{1}{2} \le y \le \frac{1}{\sqrt{2}}$. For this choice of a and b, condition (3.8) necessitates that

$$
\begin{aligned}
1 &= \int_0^1 (\xi(y,z) + \xi(1-y,z))\,\mathrm{d}z + \int_y^1 \xi(z,1-y)\,\mathrm{d}z + \int_{1-y}^1 \xi(z,y)\,\mathrm{d}z && (3.47)\\
&= \int_0^1 \xi(y,z)\,\mathrm{d}z + \int_0^1 \xi(1-y,z)\,\mathrm{d}z + 0 + 0 && (3.48)\\
&= \int_0^{1-y} \xi(y,z)\,\mathrm{d}z + \int_0^{y} \xi(1-y,z)\,\mathrm{d}z && (3.49)\\
&= \int_0^{1-y} \zeta(y)\,\mathrm{d}z + \int_0^{y} \zeta(1-y)\,\mathrm{d}z && (3.50)\\
&= (1-y)\zeta(y) + y\zeta(1-y). && (3.51)
\end{aligned}
$$

We note that the bounds on y imply that $\frac{\sqrt{2}-1}{\sqrt{2}} \le 1-y \le \frac{1}{2}$. Hence, by (3.41) applied to $y' := 1-y$, we obtain that

$$
y\zeta(1-y) = \frac{1}{2} - \int_{1-y}^{y} \zeta(z)\,\mathrm{d}z. \tag{3.52}
$$

Substituting (3.52) in (3.51) and rewriting in the required form shows the necessity of (3.42).

To see the necessity of (3.40), we consider $a = b < \frac{\sqrt{2}-1}{\sqrt{2}}$. For this choice of a and b, condition (3.8) necessitates that

$$
1 = 2\int_0^1 \xi(a,z)\,\mathrm{d}z + 2\int_a^1 \xi(z,a)\,\mathrm{d}z = 0 + 2\int_a^1 \xi(z,a)\,\mathrm{d}z = 2\int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \zeta(z)\,\mathrm{d}z. \tag{3.53}
$$

Now, by (3.41) applied to $y = \frac{\sqrt{2}-1}{\sqrt{2}}$, we obtain that

$$\frac{1}{2} = \frac{1}{\sqrt{2}}\zeta\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right) + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \zeta(z)dz = \frac{1}{\sqrt{2}}\zeta\left(\frac{\sqrt{2}-1}{\sqrt{2}}\right) + \frac{1}{2} \tag{3.54}$$

where the second equation is obtained by substituting (3.53). Hence, $\zeta(\frac{\sqrt{2}-1}{\sqrt{2}}) = 0$, showing the necessity of (3.40).

In order to complete the proof of the theorem, we have to find a function $\zeta : [\frac{\sqrt{2}-1}{\sqrt{2}}, \frac{1}{\sqrt{2}}] \to \mathbb{R}_+$ that satisfies properties (3.22)-(3.24). By solving the differential equations corresponding to (3.22)-(3.24), we get that the function satisfying these properties is

$$\zeta(y) := \frac{2y(2-y)-1}{4y(1-y)^2}. \tag{3.55}$$

By substituting the function values, it can be verified that $\int_0^1 \int_0^1 \xi(z_1, z_2)\,\mathrm{d}z_1\,\mathrm{d}z_2 = \frac{1}{2\sqrt{2}}$. We present the calculations needed for verification here. By substituting the function values, we get

$$\begin{aligned}
\int_0^1 \int_0^1 \xi(z_1, z_2)\,\mathrm{d}z_1\,\mathrm{d}z_2 &= \int_{(z_1,z_2)\in\mathcal{R}_1} \xi(z_1, z_2)\,\mathrm{d}z_2\,\mathrm{d}z_1 && (3.56)\\
&\quad + \int_{(z_1,z_2)\in\mathcal{R}_2} \xi(z_1, z_2)\,\mathrm{d}z_2\,\mathrm{d}z_1 && (3.57)\\
&= \left(1 - \frac{1}{\sqrt{2}}\right)^2 \left(1 + \frac{1}{\sqrt{2}}\right) && (3.58)\\
&\quad + \int_{z_1=\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} \int_{z_2=0}^{1-z_1} \xi(z_1, z_2)\,\mathrm{d}z_2\,\mathrm{d}z_1 && (3.59)\\
&= \left(1 - \frac{1}{\sqrt{2}}\right)^2 \left(1 + \frac{1}{\sqrt{2}}\right) + \int_{\frac{\sqrt{2}-1}{\sqrt{2}}}^{\frac{1}{\sqrt{2}}} (1 - z_1)\zeta(z_1)\,\mathrm{d}z_1 && (3.60)\\
&= \frac{\sqrt{2}-1}{2\sqrt{2}} + \frac{\sqrt{2}-1}{2} = \frac{1}{2\sqrt{2}}. && (3.61)
\end{aligned}$$

For clarity, we conclude the section by describing the obtained distribution explicitly. The probability of choosing a given T-shaped cut U is

$$\Pr(U) := \sqrt{2}\left(\int_{(\theta_1,\theta_2):V(\theta_1,\theta_2)=U} \xi(\theta_1,\theta_2)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2 + \int_{(\theta_1,\theta_2):H(\theta_1,\theta_2)=U} \xi(\theta_2,\theta_1)\,\mathrm{d}\theta_1\,\mathrm{d}\theta_2\right), \quad (3.62)$$

where

$$\xi(\theta_1,\theta_2) := \begin{cases} \frac{\sqrt{2}+1}{\sqrt{2}} & \text{if } \frac{1}{\sqrt{2}} < \theta_1, \theta_2 \le 1, \\ \frac{2\theta_1(2-\theta_1)-1}{4\theta_1(1-\theta_1)^2} & \text{if } \frac{\sqrt{2}-1}{\sqrt{2}} \le \theta_1 \le \frac{1}{\sqrt{2}} \text{ and } \theta_1+\theta_2 \le 1, \\ 0 & \text{otherwise.} \end{cases} \quad (3.63)$$

3.3 $\sqrt{2}$-Lower Bound

As mentioned earlier, (s,r,t)-NODE-LIN-3-CUT is equivalent to (s,r,t)-EDGE-LIN-3-CUT. To reduce from (s,r,t)-NODE-LIN-3-CUT to (s,r,t)-EDGE-LIN-3-CUT, replace every vertex v of the graph by two new vertices v^- and v^+ and add an edge from v^- to v^+ with weight that of v. For each edge (u,v) in the original graph, add an edge from u^+ to v^- with infinite weight. To reduce from (s,r,t)-EDGE-LIN-3-CUT to (s,r,t)-NODE-LIN-3-CUT, split each edge in the graph by inserting a new vertex with weight equal to that of the edge. Also, as mentioned in introduction (s,r,t)-EDGE-LIN-3-CUT is equivalent to DIR-SKEW-MULTICUT problem defined in Chapter 2 for $k=2$. DIR-SKEW-MULTICUT is a special case of DIR-MULTICUT where the demand graph H is a bipartite graph with k terminals $s_1,\ldots,s_k$ on one side and k terminals $t_1,\ldots,t_k$ on the other side: (s_i,t_j) is an edge in H iff $i \le j$. To reduce (s,r,t)-EDGE-LIN-3-CUT to DIR-SKEW-MULTICUT, set $s_1 = s, s_2 = r, t_1 = r, t_2 = t$. To reduce DIR-SKEW-MULTICUT to (s,r,t)-EDGE-LIN-3-CUT, set $s = s_1, t = t_2$, add a node r and edges $(t_1,r),(r,s_2)$. By Theorem 2.11, we know that approximability of DIR-SKEW-MULTICUT under UGC coincides with the integrality gap of the DISTANCE-LP. Hence, in this section we present an integrality gap instance for (s,r,t)-NODE-LIN-3-CUT which will in turn show the inapproximability result in Theorem 3.1.

Theorem 3.5. *The integrality gap of the* DISTANCE-LP *is at least* $\sqrt{2}$.

We will construct a sequence of node-weighted graphs for which the integrality gap converges to $\sqrt{2}$. In most previously known integrality gap instances for distance-based linear programs for directed multicut-like problems, the node weights were uniformly set to be one. In contrast, our gap instance assigns non-uniform weights to the nodes.

3.3.1 Gap Instance Construction

Let M be a positive integer. We will construct a graph $G = (V, E)$ on $(M+1)^2 + 3$ nodes with weights on the nodes. For convenience, let us define $V' := \{(i, j) : i, j \in \{0, 1, \ldots, M\}\}$. Thus, we may view V' as the nodes of an $(M+1) \times (M+1)$-grid whose columns and rows are indexed from 0 to M (we will follow the convention that the first index denotes the column while the second index denotes the row). The node set of G is given by $V := \{s, r, t\} \cup V'$. We now define the weights on the nodes. The construction involves a parameter $\alpha \in (0, 1/2)$ that will be determined later. We denote the weight of node (i, j) to be w_{ij} and define[2]

$$w_{ij} := \begin{cases} 0 & \text{if } i+j > M, \\ \frac{1-\alpha}{M} & \text{if } i+j < M, \\ \frac{1}{2} - \frac{(1-\alpha)i}{M} & \text{if } i+j = M,\ i < M\left(1 - \frac{1}{2(1-\alpha)}\right), \\ \alpha & \text{if } i+j = M \text{ and} \\ & M\left(1 - \frac{1}{2(1-\alpha)}\right) \le i \le M\left(\frac{1}{2(1-\alpha)}\right), \\ \frac{1}{2} - \frac{(1-\alpha)j}{M} & \text{if } i+j = M,\ i > M\left(\frac{1}{2(1-\alpha)}\right). \end{cases} \tag{3.64}$$

The edge set E consists of undirected and directed edges where the undirected edges represent directed edges in both directions. The undirected edges consist of the following: every node (i, j) is adjacent to all nodes in $V' \cap \{(i-1, j-1), (i-1, j), (i-1, j+1), (i, j-1), (i, j+1), (i+1, j-1), (i+1, j), (i+1, j+1)\}$. The directed edges consist of $r \to s, t \to r$, $s \to (i, M)$ and $(i, 0) \to r$ for every $i \in \{0, 1, \ldots, M\}$, and $(M, j) \to t$ and $r \to (0, j)$ for every $j \in \{0, 1 \ldots, M\}$ (see Figure 3.4).

We will refer to the subgraph of G induced by the vertex-set V' as a *diagonalized-grid*. We will let leftmost column, rightmost column, bottommost row, topmost row, and diagonal to denote $\{(0, j) : j \in \{0, 1, \ldots, M\}\}$, $\{(M, j) : j \in \{0, 1, \ldots, M\}\}$, $\{(i, 0) : i \in \{0, 1, \ldots, M\}\}$, $\{(i, M) : i \in \{0, 1, \ldots, M\}\}$, and $\{(i, j) : i, j \in \{0, 1, \ldots, M\}, i+j = M\}$ respectively.

3.3.2 Proof of Gap

The following lemma bounds the value of an optimal solution to the linear program.

Lemma 3.3. *An optimal solution to the* DISTANCE-LP *for the node-weighted graph constructed above has weight at most*

$$\left(\frac{1}{M}\right) \sum_{i=0}^{M} \sum_{j=0}^{M} w_{ij}. \tag{3.65}$$

Proof: It is sufficient to exhibit a feasible solution to the linear program whose objective value is as specified in the lemma. We will show that $x_{(i,j)} := 1/M$ for every $i, j \in \{0, 1, \ldots, M\}, i+j \le M$ and

[2]The various boundary conditions in the definition of the node weights will have to use appropriately rounded down and rounded up boundary values. We avoid this technicality in the interests of simplicity.

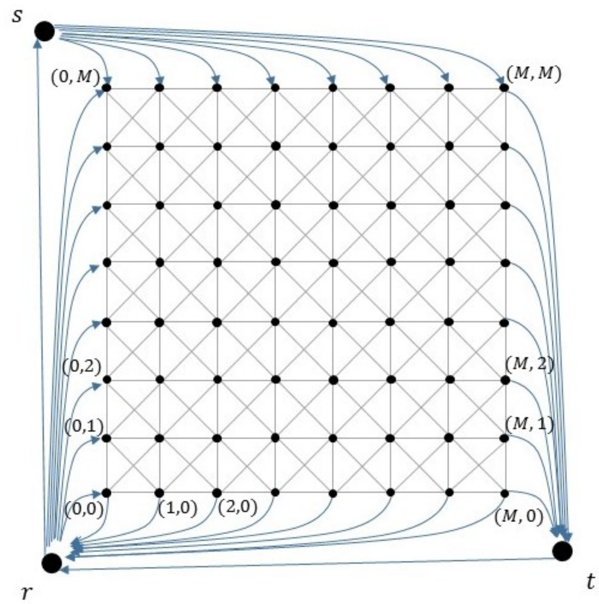


Figure 3.4: The graph corresponding to the integrality gap instance. The black edges are undirected while the blue edges are directed. The node weights are not shown.

$x_{(i,j)} := 1$ for every $i, j \in \{0, 1, \ldots, M\}, i + j > M$, is a feasible solution to the linear program. We recall that nodes (i, j) with $i + j > M$ have weight $w_{ij} = 0$. Let us consider the graph H obtained from G by removing all nodes (i, j) with $i + j > M$. To show feasibility of x, it suffices to show that every path from s to r, from r to t and from s to t has at least M intermediate nodes in H.

A path in H from a node $(i, j) \in V(H)$ to r has to cross j intermediate rows and hence has at least j internal nodes. Hence, for every node (i, j) with $i, j \in \{0, 1 \ldots, M\}, i + j \leq M$, the number of internal nodes in every path from (i, j) to r in H is at least j. Now every path from s to r in H has to go through $(0, M)$ and hence has at least M internal nodes from $V(H) \cap V'$. Similarly, every path from r to t in H has at least M internal nodes from $V(H) \cap V'$. Finally, the distance from s to t in H is at least the distance from r to t in H owing to the edge $r \to s$ and hence, the number of internal nodes from $V(H) \cap V'$ in any path from s to t in H is also at least M.

The next lemma shows a lower bound on the objective value of an integral optimum solution.

Lemma 3.4. *An optimal solution to* (s, r, t)-NODE-LIN-3-CUT *in the node-weighted graph constructed above has weight at least* 1.

Proof: Let U^* be an integral optimal solution. We will show the lower bound on the weight of U^* in two steps. We define the *axis-parallel neighbors* of a node (i, j) to be the nodes in $\{(i+1, j), (i-1, j), (i, j+1), (i, j-1)\}$ and a path to be an *axis-parallel path* if all neighbors of a node (i, j) occurring in the path are its axis-parallel neighbors. In the first step of the proof, we show that U^* consists of an axis-parallel path from a node in the topmost row to a node in the bottommost row and an axis-parallel path from a node in the leftmost column to a node in rightmost column.

Lemma 3.5. *The optimal solution U^* contains a set of nodes which form an axis-parallel path P_1 from a node in the bottommost row to a node in the topmost row in G.*

Lemma 3.6. *The optimal solution U^* contains a set of nodes which form an axis-parallel path P_2 from a node in the leftmost column to a node in the rightmost column in G.*

We defer the proof of Lemmas 3.5 and 3.6 to complete the proof of Lemma 3.4. As a second step of the proof, we now show a lower bound on the total weight of the union of the nodes in these two paths. The next claim follows immediately from the definition of axis-parallel paths.

Claim 3.6. *Every axis-parallel path from node (i_1, j_1) to (i_2, j_2) contains at least $|i_2 - i_1| + |j_2 - j_1| - 1$ internal nodes.*

We also have the following claim from the definition of the node weights.

Claim 3.7. *For every node (i, j) for $i, j \in \{0, 1, \ldots, M\}$ with $i + j = M$, we have*

$$w_{ij} = \max\left\{\frac{1}{2} - \frac{(1-\alpha)i}{M}, \alpha, \frac{1}{2} - \frac{(1-\alpha)j}{M}\right\}. \tag{3.66}$$

Let P_1 and P_2 be the node sets guaranteed by Lemmas 3.5 and 3.6 respectively. Without loss of generality, let P_1 be the node set in U^* that induces an axis-parallel path from a node $(a, 0)$ to a node (b, M). Similarly, let P_2 be the node set in U^* that induces an axis-parallel path from a node $(0, c)$ to a node (M, d) (see Figure 3.5).

Since P_1 is an axis-parallel path from a node in the bottommost row to a node in the topmost row, there exists a node in P_1 from the diagonal. Let $(x, M - x)$ be the first node along the axis-parallel path P_1 that is in the diagonal. Let P_1' be the restriction of P_1 from $(a, 0)$ to $(x, M - x)$. Let (x', y') be the first node along the axis-parallel path P_2 that is either in the diagonal or in P_1'. Let P_2' be the restriction of P_2 from $(0, c)$ to (x', y'). By construction, all nodes (p, q) of $P_1' \cup P_2'$ satisfy $p + q \leq M$. We will show that the total weight of the nodes in $P_1' \cup P_2'$ is at least 1. This suffices since $P_1' \cup P_2' \subseteq U^*$. We distinguish two cases.

1. Suppose P_2' is a path from $(0, c)$ to a node (i, j) of P_1', where $i + j \leq M$. By Claim 3.6, the axis-parallel path P_2' has at least $|i - 0| + |j - c| - 1 \geq i - 1$ internal nodes. Furthermore, the path P_1' is the concatenation of an axis-parallel path Q_1 from $(a, 0)$ to (i, j) and an axis-parallel path Q_2 from (i, j) to $(x, M - x)$. Hence, by Claim 3.6, the axis-parallel path P_1' has at least $|i - a| + |j - 0| - 1 + 1 + |x - i| + |M - x - j| - 1 \geq j + |x - i| + |M - x - j| - 1 \geq M - i - 1$ internal nodes. We recall that all these nodes have weight $(1 - \alpha)/M$. Additionally, the nodes $(a, 0)$ and $(0, c)$ have weight $(1 - \alpha)/M$ each. The node $(x, M - x)$ on the diagonal has weight at least α by

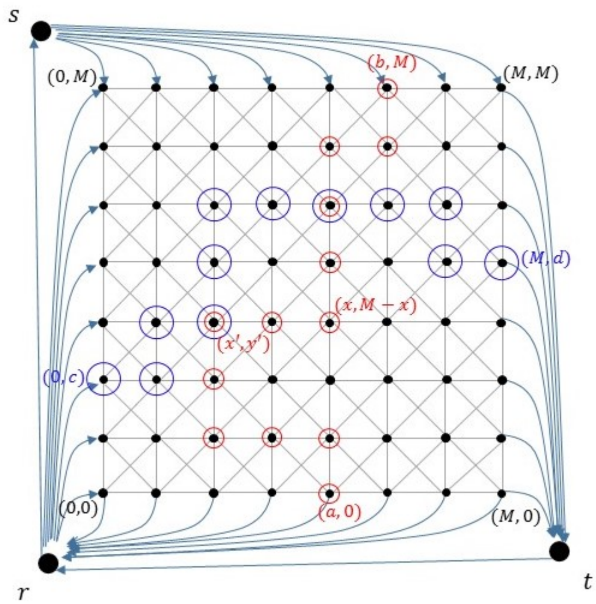


Figure 3.5: The red circled nodes denote the axis-parallel path P_1 and the blue circled nodes denote the axis-parallel path P_2.

Claim 3.7. Combining these, we get that the total weight of the nodes in $P_1' \cup P_2'$ is at least

$$((i-1)+(M-i-1)+2)\left(\frac{1-\alpha}{M}\right)+\alpha=1. \tag{3.67}$$

2. Suppose P_2' is a path from $(0,c)$ to a node $(x', M-x')$ on the diagonal. In this case, we will show that the total weight of the nodes in P_1' and P_2' are each lower bounded by $1/2$. By Claim 3.6, the axis-parallel path P_1' has at least $|x-a|+|M-x-0|-1 \geq M-x-1$ internal nodes each of which has weight $(1-\alpha)/M$. Additionally, the end-node $(a,0)$ also has weight $(1-\alpha)/M$ and the end-node $(x,M-x)$ has weight at least $1/2-(1-\alpha)((M-x)/M)$ by Claim 3.7. Thus, the total weight of the nodes in P_1' is at least

$$((M-x-1)+1)\left(\frac{1-\alpha}{M}\right)+\frac{1}{2}-\frac{(1-\alpha)(M-x)}{M}=\frac{1}{2}. \tag{3.68}$$

We proceed by a similar argument for the total weight of the nodes in P_2'. By Claim 3.6, the axis-parallel path P_2' has at least $|x'-0|+|M-x'-c|-1 \geq x'-1$ internal nodes each of which has weight $(1-\alpha)/M$. Additionally, the end-node $(0,c)$ also has weight $(1-\alpha)/M$ and the end-node $(x', M-x')$ has weight at least $1/2-(1-\alpha)(x'/M)$ by Claim 3.7. Thus, the total weight of the nodes in P_2' is at least

$$((x'-1)+1)\left(\frac{1-\alpha}{M}\right)+\frac{1}{2}-\frac{(1-\alpha)x'}{M}=\frac{1}{2}. \tag{3.69}$$

This completes the proof of Lemma 3.4.

We now prove Lemma 3.5. The proof of Lemma 3.6 is similar.

Proof:[Lemma 3.5] Let U_t^* denote an inclusion wise-minimal subset of U^* such that $G - U_t^*$ contains no path from r to t. We will show that U_t^* contains a path P_1 as required. Showing this is equivalent to showing the following combinatorial statement: every subset of nodes that intersects all paths from left to right in a diagonalized-grid (since G contains a diagonalized-grid) has a subset of nodes that induce an axis-parallel path from a node in the topmost row to a node in the bottommost row in G. We proceed to show this now.

We will show the combinatorial statement using a coloring argument. Let $R := U_t^* \cup \{(i,-1),(i,M+1) : i \in \{0,1,\ldots,M\}\}$ and $B := (V' \setminus U_t^*) \cup \{(-1,j),(M+1,j) : j \in \{0,1,\ldots,M\}\}$. Call the corresponding nodes as red and blue nodes respectively (we observe that the sets R and B have extra nodes in addition to the nodes in the diagonalized-grid of G, but this is only for the purposes of notational convenience in this proof). We will construct an auxiliary graph for the purposes of the proof—for clarity, we will refer to the vertices of G as *nodes* and the vertices of the auxiliary graph as *vertices*.

We construct an undirected graph H as follows. The vertex set of H is given by $V(H) := \{v_{i,j} : i,j \in \{0,1,\ldots,M+1\}\} \cup \{a,b,c,d\}$. We call a vertex $v_{i,j}$ to be in column i and row j. We define $a' := v_{0,0}, b' := v_{M+1,0}, c' := v_{0,M+1}, d' := v_{M+1,M+1}$ and call them to be the *corner vertices*.

The edge set of H is denoted by $E(H)$: a vertex $v_{i,j}$ is adjacent to all vertices in $V(H) \cap \{v_{i-1,j}, v_{i+1,j}, v_{i,j-1}, v_{i,j+1}\}$ (i.e., the undirected grid edges) and vertices a,b,c,d are adjacent to a',b',c',d' respectively.

We note that H is a plane graph that corresponds to a square grid with four pendant vertices that are adjacent to the four corner vertices of the grid. In the following, it is helpful to consider overlaying H on top of G as shown in Figure 3.6 with each internal face of H containing exactly one node from V'. For a vertex $v_{i,j} \in V(H) \setminus \{a,a',b,b',c,c',d,d'\}$, we define $D_1^{i,j} := \{(i,j),(i-1,j-1)\}$ and $D_2^{i,j} := \{(i-1,j),(i,j-1)\}$. We emphasize that $D_1^{i,j}$ and $D_2^{i,j}$ consist of nodes from the original graph G. They are the nodes in the two diagonally opposite faces adjacent to $v_{i,j}$ in the overlay (see Figure 3.6).

We now modify H to obtain a directed subgraph G' as follows: for an edge $e \in E(H)$ with $e = \{v_{i,j}, v_{i+b_1,j+b_2}\}$ where $b_1,b_2 \in \{0,1\}$ and $b_1 + b_2 = 1$, we say that e is *bi-labeled* if $|\{(i,j),(i+b_1-1,j+b_2-1)\} \cap R| = 1$ and $|\{(i,j),(i+b_1-1,j+b_2-1)\} \cap B| = 1$ (i.e., the two faces of the edge contain a node from R and B). In addition, we will call the edges $\{a,a'\},\{b,b'\},\{c,c'\},\{d,d'\}$ to be *trivially bi-labeled*. We delete all edges of H that are *not* bi-labeled. We orient the trivially bi-labeled edges as $a' \to a$, $b \to b'$, $c \to c'$ and $d' \to d$ and orient the remaining bi-labeled edges by the following rule (see Figure 3.7):

1. for an edge $e = \{v_{i,j}, v_{i+1,j}\}$, we will orient the edge as
   (a) $v_{i,j} \to v_{i+1,j}$ if $(i,j) \in R$ and $(i,j-1) \in B$ and declare (i,j) to be the *left node* and $(i,j-1)$ to be the *right node* of the edge,

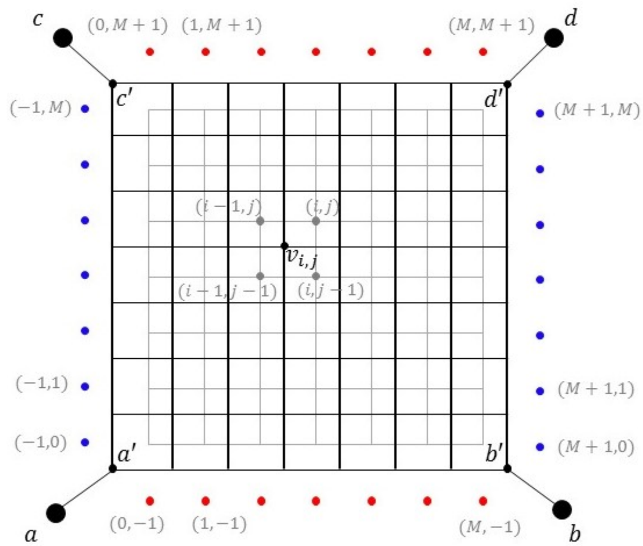


Figure 3.6: The diagonalized-grid of the graph G is shown in gray while the graph H is shown in black. For visual simplicity, we have not included the diagonal edges of the diagonalized-grid. The extra red and blue nodes are also shown. For node $v_{i,j}$, the nodes of G in the two diagonally opposite faces $D_1^{i,j}$ and $D_2^{i,j}$ are also shown.

(b) $v_{i+1,j} \to v_{i,j}$ if $(i,j) \in B$ and $(i,j-1) \in R$ and declare (i,j) to be the right node and $(i,j-1)$ to be the left node of the edge,

2. for an edge $e = \{v_{i,j}, v_{i,j+1}\}$, we will orient the edge as

(a) $v_{i,j+1} \to v_{i,j}$ if $(i,j) \in R$ and $(i-1,j) \in B$ and declare (i,j) to be the left node and $(i-1,j)$ to be the right node of the edge,

(b) $v_{i,j} \to v_{i,j+1}$ if $(i,j) \in B$ and $(i-1,j) \in R$ and declare (i,j) to be the right node and $(i-1,j)$ to be the left node of the edge,

We observe that this orienting rule ensures that the left and right nodes of every oriented edge are red and blue respectively (see Figure 3.7).

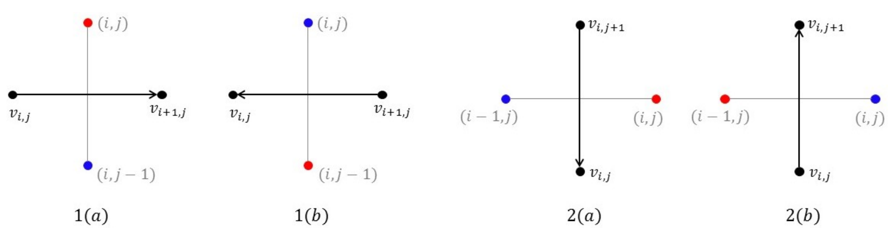


Figure 3.7: Orienting the bi-labeled edges of H.

We make one final modification to G' to obtain G: for each vertex $v_{i,j}$ where $i,j \in \{1,\ldots,M\}$,

(I) if $D_1^{i,j} \subseteq B$ and $D_2^{i,j} \subseteq R$, then (see Figure 3.8) we replace the vertex $v_{i,j}$ by $v_{i,j}^1, v_{i,j}^2$, declare them to be the vertices in row i and column j, and replace the head of the incoming edge from the vertex in column $i-1$, row j by $v_{i,j}^1$, replace the head of the incoming edge from the vertex in column $i+1$, row j by $v_{i,j}^2$, replace the tail of the outgoing edge to the vertex in column i, row $j+1$ by $v_{i,j}^1$, and replace the tail of the outgoing edge to the vertex in column i, row $j-1$ by $v_{i,j}^2$, and

(II) if $D_1^{i,j} \subseteq R$ and $D_2^{i,j} \subseteq B$, then (see Figure 3.9) we replace the vertex $v_{i,j}$ by $v_{i,j}^1, v_{i,j}^2$, declare them to be the vertices in row i and column j, and replace the head of the incoming edge from the vertex in column i, row $j+1$ by $v_{i,j}^1$, replace the head of the incoming edge from the vertex in column i, row $j-1$ by $v_{i,j}^2$, replace the tail of the outgoing edge to the vertex in column $i+1$, row j by $v_{i,j}^1$, and replace the tail of the outgoing edge to the vertex in column $i-1$, row j by $v_{i,j}^2$.

We call the above operation to be a split operation. We emphasize that the operation separates the red nodes in a consistent manner. The left and right nodes of all oriented edges still remain the same after the split operation.

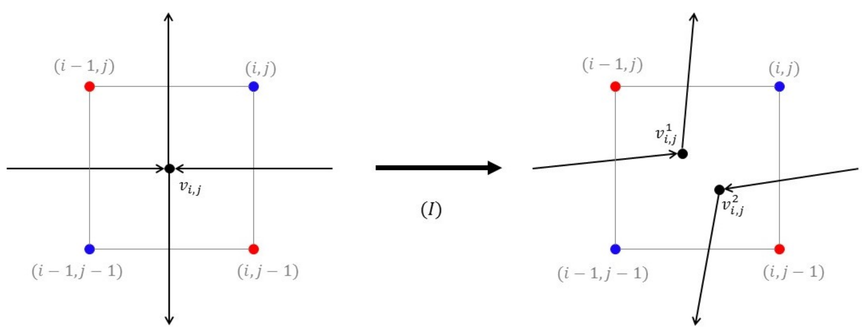


Figure 3.8: Splitting operation (I).

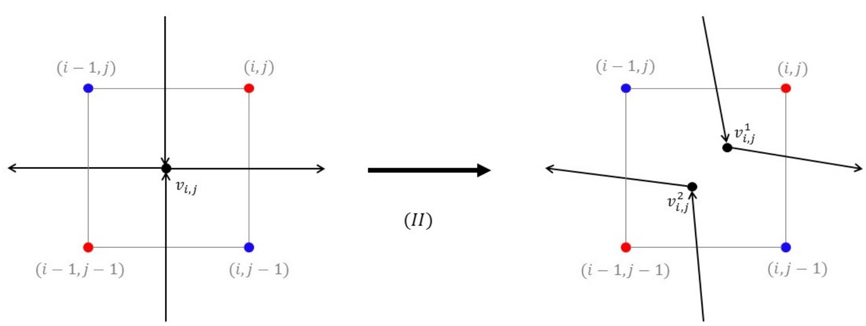


Figure 3.9: Splitting operation (II).

Claim 3.8. *Let $v \in V(G) \setminus \{a,b,c,d\}$. Then, the incoming and outgoing degree of v are either both zero or are both* 1.

Proof: Vertices in G that were obtained by splitting a vertex in G' clearly satisfy the property since they have incoming and outgoing degree to be 1 after the split. So, we may assume that v is a vertex in G' as well as G. Suppose v has incoming degree to be one in G' (the proof for outgoing degree being one is identical).

Suppose v is not a corner vertex. Let $v = v_{i,j}$. Without loss of generality, let the incoming edge be from a vertex in column $i-1$ and row j (see Figure 3.10). Then, $(i-1, j-1) \in B, (i-1, j) \in R$. Based on whether $(i, j-1)$ is in R or B and whether (i, j) is in R or B, we have four cases. One of the cases cannot happen since $v_{i,j}$ is a vertex in both G and G'. The remaining three cases show that the outgoing degree from $v_{i,j}$ is also one in G'.

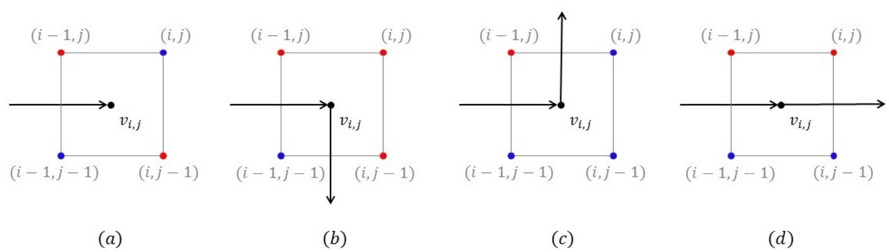


Figure 3.10: Degree of internal vertices: Case (a) is impossible. Cases (b), (c) and (d) have a unique outgoing edge as well.

Suppose v is a corner vertex. Without loss of generality, let $v = v_{0,M} = c'$ (see Figure 3.11). Now, depending on whether $(0, M)$ is in R or B, we have two cases. In both cases, the outgoing degree from $v_{0,M}$ is indeed one.

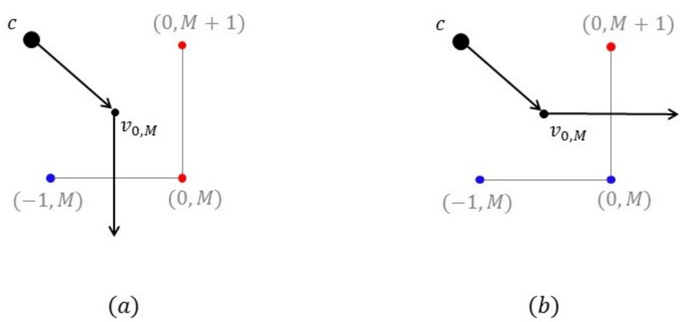


Figure 3.11: Degree of corner vertices.

Thus, the only vertices in G with outgoing degree 1 and incoming degree 0 are b and c while the only vertices in G with incoming degree 1 and outgoing degree 0 are a and d. Hence, by Claim 3.8, there exists a path from c to either a or d in G.

Claim 3.9. *Suppose there exists a path from c to d in G. Then there exists a path in $G - U_t^*$ from s to r.*

Proof: Suppose we have a path from c to d in G. Let P denote the nodes of G along the right of the edges in this path. Thus, P induces a path from a node in the leftmost column to a node in the rightmost column in G. We recall that the right nodes along the edges in the path are blue nodes and are indeed not in U_t^*. Thus, we have a path from a node in the leftmost column to a node in the rightmost column in $G - U_t^*$ and hence a path from r to t in $G - U_t^*$.

Claim 3.9 shows that a path from c to d in G contradicts the fact that U_t^* is a $r \to t$ cut in G. Thus, we must have a path from c to a in G. Claim 3.10 below completes the proof of the lemma.

Claim 3.10. *Suppose there exists a path from c to a in G. Then there exists an axis parallel path from a node in the topmost row to a node in the bottommost row in U_t^*.*

Proof: Suppose we have a path Q from c to a in G. Let P denote the nodes of G along the left of the edges in this path. We recall that the left nodes along the edges in the path are red nodes and hence are in U_t^*. Thus, P is a path from a node in the topmost row to a node in the bottommost row in G. It remains to show that P can be transformed into an axis-parallel path.

Suppose P uses a diagonal edge in G. Without loss of generality, let it be $(i-1, j) \to (i, j-1)$. Let Q' be the path Q projected on G'—i.e., use the projected edges in G'. Then, Q' traverses $v_{i-1,j} \to v_{i,j} \to v_{i,j-1}$. These edges imply that $(i-1, j), (i, j-1) \in R$ and $(i-1, j-1) \in B$. If $(i, j) \in B$, then the split operation to obtain G from G' shows that the edges in Q do not exist in G, a contradiction (see Figure 3.12).

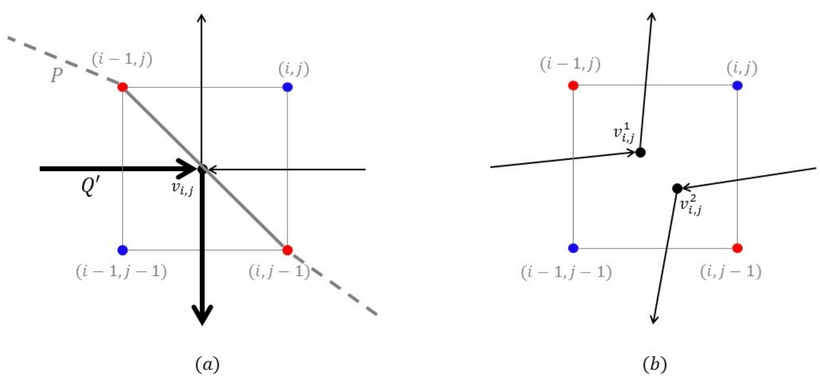


Figure 3.12: Diagonal path P. Path Q' is impossible owing to the split operation.

Thus, we may assume that $(i, j) \in R$ and is hence in U_t^*. In this case, we can ensure that P makes fewer axis-parallel turns by rerouting as $(i-1, j) \to (i, j) \to (i, j-1)\}$ (see Figure 3.13). By rerouting this way for each diagonal edge of P, we obtain the required axis-parallel path.

With Lemmas 3.3 and 3.4, we prove the main theorem of the section. We restate it below for convenience.

Theorem 3.6. *The integrality gap of the* DISTANCE-LP *is at least* $\sqrt{2}$.

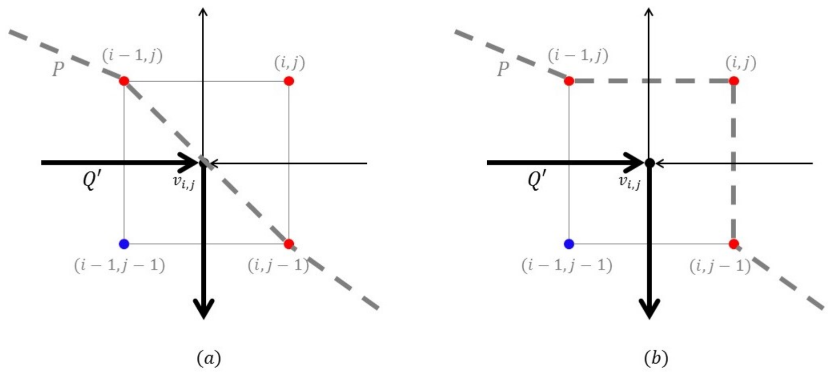


Figure 3.13: Diagonal path P can be made axis-parallel.

Proof: We will use the sequence of instances constructed at the beginning of the section. By Lemmas 3.3 and 3.4, it only remains to fix a choice of α and bound the sum of the node weights. We will pick an α that *minimizes* the sum of the node weights in order to get the largest possible integrality gap.

We now compute the sum of the node weights as a function of α. The following claim follows from the definitions of the node weights.

Claim 3.11.

1.

$$\sum_{i,j\in\{0,\ldots,M\}:\ i+j\neq M} w_{ij} = \frac{(1-\alpha)(M+1)}{2},$$

2.

$$\sum_{\substack{i,j\in\{0,\ldots,M\}:\ i+j=M,\\ i<\left(1-\frac{1}{2(1-\alpha)}\right)M \text{ or } i>\left(\frac{1}{2(1-\alpha)}\right)M}} w_{ij} = \left(\frac{(1-2\alpha)M}{4(1-\alpha)}\right)\left(\frac{1+2\alpha}{2}+\frac{1-\alpha}{M}\right), \textit{ and}$$

3.

$$\sum_{\substack{i,j\in\{0,\ldots,M\}:\ i+j=M,\\ \left(1-\frac{1}{2(1-\alpha)}\right)M\leq i\leq\left(\frac{1}{2(1-\alpha)}\right)M}} w_{ij} = \frac{\alpha^2 M}{1-\alpha}.$$

Using the above claim, we have that

$$\sum_{i=0}^{M}\sum_{j=0}^{M} w_{ij} = \left(\frac{3-4\alpha+2\alpha^2}{4(1-\alpha)}\right)M + 1 - \frac{3\alpha}{2} \tag{3.70}$$

Now, the minimum value of the function $f(\alpha) := (3-4\alpha+2\alpha^2)/(4(1-\alpha))$ in the domain $(0,1/2)$ occurs at $\alpha = 1-1/\sqrt{2}$ and thus the minimum value of the function is $\min_{\alpha\in(0,1/2)} f(\alpha) = 1/\sqrt{2}$. Using this value of α shows that the objective value of an optimal solution to the linear program is at most

$1/\sqrt{2} + \Theta(1/M)$ while the objective value of an optimal integral solution is at least 1. Consequently, the integrality gap of the sequence of instances constructed as above converges to $\sqrt{2}$ when M tends to infinity.

3.4 Results for Related Cut Problems

We prove Theorems 3.2 and 3.3 in this section.

3.4.1 Blocking Arborescences

In this section, we show that the approximability of r-InOut-Node-Blocker and (s,r,t)-Node-Lin-3-Cut coincide. We recall the problem r-InOut-Node-Blocker: Given a node-weighted directed graph with a specified terminal node r, find a minimum weight set of non-terminal nodes to remove so that the resulting graph has no out-r-arborescence *and* no in-r-arborescence. Theorem 3.2 follows from the following result in conjunction with Theorem 3.1.

Theorem 3.7. *There exists an efficient α-approximation algorithm for r-InOut-Node-Blocker if and only if there exists an efficient α-approximation for (s,r,t)-Node-Lin-3-Cut.*

Proof: We need the notion of the Strong-Node-Cut problem: the input is a directed graph with node weights, and the goal is to find a minimum weight subset of nodes whose deletion results in at least two disjoint weakly connected components. We observe that Strong-Node-Cut can be solved in polynomial-time. We first show that r-InOut-Node-Blocker is a combination of (s,r,t)-Node-Lin-3-Cut and Strong-Node-Cut.

Claim 3.12. *For every directed graph $G = (V,E)$ with $r \in V$, the optimal solution to r-InOut-Node-Blocker has value equal to*

$$\min\left(\min_{s,t\in V-r}\{(s,r,t)\text{-Node-Lin-3-Cut } \textit{in } G\}, \text{Strong-Node-Cut } \textit{in } G\right). \tag{3.71}$$

Proof: Let U be an optimal solution of r-InOut-Node-Blocker in $G = (V,E)$ with $r \in V$. The optimal values of both (s,r,t)-Node-Lin-3-Cut in G and Strong-Node-Cut in G are upper bounds for the weight of U. If the weight of U is strictly smaller than Strong-Node-Cut, then $G[V-U]$ is weakly connected. By the definition of U, we have that $G[V-U]$ does not contain an in-r-arborescence and hence it has a strongly connected component C_1 not containing r with $\delta^{out}_{G[V-U]}(C_1) = \emptyset$. Similarly, since $G[V-U]$ does not contain an out-r-arborescence, it has a strongly connected component C_2 not containing r with $\delta^{in}_{G[V-U]}(C_2) = \emptyset$. Since $G[V-U]$ is weakly connected, we have $C_1 \neq C_2$. Since C_1 and C_2 are strongly connected components, they are disjoint. For arbitrary nodes $s \in C_1$ and $t \in C_2$, there are no directed paths from s to r, from r to t and from s to t in $G[V-U]$. Thus U is a feasible solution to (s,r,t)-Node-Lin-3-Cut in G.

Now we turn to the proof of the theorem. The 'if' part follows from Claim 3.12 above. To see the other direction, consider an instance $G = (V, E)$ of (s, r, t)-NODE-LIN-3-CUT. Clearly, we may assume that s, r and t have infinite weights. For each node $v \in V$, add an arc from t to v and an arc from v to s. This step does not affect the values of the feasible solutions to (s, r, t)-NODE-LIN-3-CUT. Let G' denote the graph thus arising.

We claim that the feasible solutions with finite weight of (s, r, t)-NODE-LIN-3-CUT and those of r-INOUT-NODE-BLOCKER coincide in G'. Indeed, assume first that U is a solution of (s, r, t)-NODE-LIN-3-CUT in G'. As $G'[V - U]$ does not contain a directed path from s to r or from r to t, there exists no in-r-arborescence or out-r-arborescence in $G'[V - U]$, hence U is also a solution of r-INOUT-NODE-BLOCKER in G'. Now assume that U is a solution of r-INOUT-NODE-BLOCKER in G' with finite weight, that is, $s, t \notin U$. If $G'[V - U]$ contains a directed path from s to r or from r to t or from s to t, then the arcs that were added to G can be used to obtain either an in-r-arborescence or an out-r-arborescence, a contradiction. Hence no such path exists and so U is also a solution of (s, r, t)-NODE-LIN-3-CUT in G'.

By the above, an α-approximate solution to r-INOUT-NODE-BLOCKER in the extended graph is also an α-approximate solution to (s, r, t)-NODE-LIN-3-CUT in G, thus concluding the proof of the theorem.

The above proof ideas also extend to show that there exists an efficient α-approximation algorithm for r-INOUT-EDGE-BLOCKER if and only if there exists an efficient α-approximation for (s, r, t)-EDGE-LIN-3-CUT, which is equivalent to (s, r, t)-NODE-LIN-3-CUT.

3.4.2 Hardness of approximation of $\{s, *\}$-EDGE-BICUT

In this section, we improve on the hardness of approximation of $\{s, *\}$-EDGE-BICUT. We recall the problem $\{s, *\}$-EDGE-BICUT: Given an edge-weighted directed graph $G = (V, E)$ with a node s, find a minimum weight subset of edges to remove so that the resulting graph has a node t such that s cannot reach t and t cannot reach s. Theorem 3.3 follows from the following result in conjunction with Theorem 3.1.

Theorem 3.8. *There exists an approximation preserving reduction from* (s, r, t)-EDGE-LIN-3-CUT *to* $\{s, *\}$-EDGE-BICUT.

Proof: Given an instance of (s, r, t)-EDGE-LIN-3-CUT $G = (V, E)$ with edge weights $w \in \mathbb{R}_+^E$ and nodes $s, r, t \in V$, we construct an instance of $\{r, *\}$-EDGE-BICUT as follows: add a new node u to G; add arcs $r \to s$, $t \to u$, $u \to s$ and arcs $t \to v, v \to s$ for every $v \in V$ with infinite weight. Let $G' = (V', E')$ denote the resulting graph with edge-weights $w' \in \mathbb{R}_+^{E'}$. We now show that this reduction is an approximation-preserving reduction.

Suppose $F \subseteq E$ is a feasible solution for (s, r, t)-EDGE-LIN-3-CUT for the given instance $G = (V, E)$ with edge weights $w \in \mathbb{R}_+^E$. Then, the subset $F \subset E'$ is also a feasible solution to $\{r, *\}$-EDGE-BICUT in G' with the same weight: Since r cannot reach t in $G - F$ and the only incoming arc into u in G' is from t, the node r cannot reach u in $G' - F$; since s cannot reach r in $G - F$ and the only outgoing arc from u in G' is to s, the node r cannot be reached by u in $G' - F$.

Suppose $F \subseteq E'$ is a feasible solution for $\{r, *\}$-EDGE-BICUT in G' with finite cost. Then, F cannot contain any of the newly added arcs and hence $F \subseteq E$. We show that the subset F is a feasible solution to (s, r, t)-EDGE-LIN-3-CUT in G with the same weight. Let v be a node that cannot reach r and cannot be reached by r in $G' - F$. If r can reach t in $G - F$, then r can reach v in $G' - F$ owing to the infinite weight arc $t \to v$ in G', a contradiction. Thus, r cannot reach t in $G - F$. If s can reach t in $G - F$, then owing to the infinite weight arc $r \to s$ in G', it follows that r can reach t in $G' - F$, a contradiction. Thus, s cannot reach t in $G - F$. If s can reach r in $G - F$, then owing to the infinite weight arc $v \to s$ in G', it follows that v can reach r in $G' - F$, a contradiction. Thus, s cannot reach r in $G - F$.

3.5 Concluding Remarks

In this chapter, we proved tight bounds on the approximability of Linear-3-cut. It opens up the possibility of resolving the approximability of a more general problem, Linear-k-Cut. In Linear-k-cut problem, where we are given a directed graph and a set of terminals $s_1, \ldots, s_k$. The goal is to remove minimum weight set of edges or nodes such that there is no path left from s_i to s_j for $i < j$. Linear-k-cut is equivalent to the DIR-SKEW-MULTICUT problem considered in Chapter 2. Thus, resolving the approximability of Linear-k-cut will resolve the approximability of DIR-SKEW-MULTICUT as well.

Our proof for Linear-3-Cut consists of two parts. First part shows there exists a distribution over T-shaped cuts which achieve a $\sqrt{2}$-approximation factor. Optimality of this factor is proved in the second part by exhibiting instances with integrality gap limiting to $\sqrt{2}$. We can generalize the notion of T-shaped cuts for Linear-k-cuts for $k > 3$. However, there are two issues in generalizing the result. First issue is in finding the optimal distribution over the T-shaped cuts for Linear-k-Cut for $k > 3$. Optimal distribution over T-cuts was very complicated even for $k = 3$ and has been extremely hard to find for $k > 3$. Second issue is about the optimality of T-shaped cuts. Even if we could find the best distribution for T-shaped cuts, it is not necessary that this gives the tight approximation ratio for Linear-k-cut for $k > 3$. Best distribution over T-shaped Cuts achieving the tight approximation ratio may have been a coincidence for $k = 3$.

Chapter 4

Integrality gap results for Multiway Cut

In this chapter, we improve the hardness results known for a special case of MULTICUT, known as Multiway Cut. From the results in section 2.4 and that of Manokaran *et al.* [68], we know that assuming UGC, hardness matches the integrality gap of certain LPs. Hence, our focus in this chapter is on improving the integrality gap lower bounds.

EDGE-WT-MWC: The input is an undirected graph $G = (V, E)$ and a set of terminals $s_1, \dots, s_k \in V$ along with non-negative edge weights $w(e), e \in E$. The goal is to find minimum weight set of edges $E' \subseteq E$ such that for $i \neq j \in [1, k]$, there is no path from s_i to s_j in $G - E'$.

DIR-MWC: The input is a directed graph $G = (V, E)$ and a set of terminals $s_1, \dots, s_k \in V$ along with non-negative edge weights $w(e), e \in E$. The goal is to find minimum weight set of edges $E' \subseteq E$ such that for $i \neq j \in [1, k]$, there is no path from s_i to s_j in $G - E'$.

For EDGE-WT-MWC, Manokaran *et al.* [68] showed that the integrality gap of CKR relaxation gives a matching hardness of approximation bound for EDGE-WT-MWC assuming UGC. The CKR relaxation takes a geometric perspective of the problem. For a graph $G = (V, E)$ with edge weights $w : E \to \mathbb{R}_+$ and terminals $t_1, \dots, t_k$, the CKR relaxation is given by

$$\min \sum_{e=\{u,v\} \in E} w(e) \|x^u - x^v\|_1 \tag{4.1}$$

$$x^u \in \Delta_k \ \forall \ u \in V, \tag{4.2}$$

$$x^{t_i} = e_i \ \forall \ i \in [k], \tag{4.3}$$

where $\Delta_k := \{(x_1, \dots, x_k) \in [0,1]^k : \sum_{i=1}^k x_i = 1\}$ is the $(k-1)$-dimensional simplex and $e^i \in \{0,1\}^k$ is the extreme point of the simplex along the i-th coordinate axis, i.e., $e^i_j = 1$ if and only if $j = i$.

Current best upper bound on the integrality gap of CKR relaxation is 1.2965 due to Sharma and Vondrák [77] and current best lower bound is 1.20 due to Angelidakis, Makarychev and Manurangsi [3]. In this work, we improve on the lower bound by constructing an instance with integrality gap 1.20016.

Theorem 4.1. *For every constant $\epsilon > 0$, there exists an instance of* EDGE-WT-MWC *such that the integrality gap of the CKR relaxation for that instance is at least* $1.20016 - \epsilon$.

The above result in conjunction with the result of Manokaran et al. immediately implies the following corollary:

Corollary 4.1. EDGE-WT-MWC *is UGC-hard to approximate within a factor of* $1.20016 - \epsilon$ *for every* $\epsilon > 0$.

$$
\begin{array}{llll}
 & \textbf{DIR-MWC-REL} & & \\
\min & \sum_{e \in E} w_e x_e & & \\
 & \sum_{e \in p} x_e & \geq 1 & p \in \mathcal{P}_{ij}, i \neq j \\
 & x_e & \geq 0 & e \in E
\end{array}
$$

Figure 4.1: LP Relaxation for DIR-MWC

From the result of Zosin *et al.* [70] and the algorithm in section 2.3.1, we know that the integrality gap of the DIR-MWC-REL (flow-cut gap) for DIR-MWC is at most 2. Hence, DIR-MWC admits a 2-approximation algorithm and does not admit a $2-\epsilon$ approximation algorithm for any $\epsilon > 0$ assuming UGC [61]. We show that the same bound is tight even for a more restrictive case of $k = 2$, referred to as $\{s,t\}$-EDGE-BICUT.

$\{s,t\}$-EDGE-BICUT: The input is a directed graph $G = (V, E)$ and two vertices $s, t \in V$ along with non-negative edge weights $w(e), e \in E$. The goal is to find minimum weight set of edges $E' \subseteq E$ such that there is no path from s to t *and* no path from t to s in $G - E'$.

Theorem 4.2. *Integrality gap of* DIR-MWC-REL *for* $\{s,t\}$-EDGE-BICUT *is* 2 *even in planar directed graphs.*

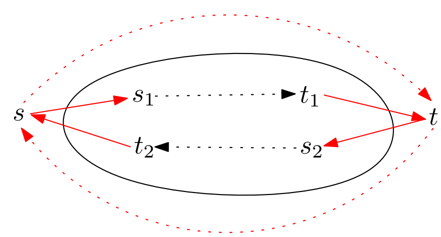


Figure 4.2: Reduction from DIR-MULTICUT with two demand pairs $(s_1, t_1), (s_2, t_2)$ to $\{s,t\}$-EDGE-BICUT

Subsequent to our construction, Julia Chuzhoy obtained an alternative non-recursive construction with an integrality gap of 2 for $\{s,t\}$-EDGE-BICUT. This can be deduced from the fact that $\{s,t\}$-EDGE-BICUT is equivalent to DIR-MULTICUT with two demand pairs, and the fact that DIR-MULTICUT with k demand pairs has integrality gap equal to k [74]. Figure 4.2 shows the reduction from an instance of DIR-MULTICUT with two demand pairs to $\{s,t\}$-EDGE-BICUT. Reduction from $\{s,t\}$-EDGE-BICUT to DIR-MULTICUT with two demand pairs is trivial.

Using theorem 2.11, we obtain the following corollary:

Corollary 4.2. $\{s,t\}$-EDGE-BICUT *is UGC-hard to approximate within a factor of* $2-\epsilon$ *for every* $\epsilon > 0$.

Organization: In Section 4.2, we show that the integrality gap of DIR-MWC-REL is 2 for $\{s,t\}$-EDGE-BICUT. In Section 4.1, we show that the integrality gap of CKR relaxation is at least 1.20016

4.1 Improving the integrality gap bound for CKR-relaxation

4.1.1 Non-opposite Cuts

Before outlining our techniques, we briefly summarize the background literature that we build upon to construct our instance. We rely on two significant results from the literature. In the context of the k-way cut problem, a *cut* is a function $P : \Delta_k \to [k+1]$ such that $P(e^i) = i$ for all $i \in [k]$, where we use the notation $[k] := \{1, 2, \ldots, k\}$. The approximation ratio $\tau_k(\mathcal{P})$ of a distribution $\mathcal{P}$ over cuts is given by its *maximum density*:

$$\tau_k(\mathcal{P}) := \sup_{x,y \in \Delta_k, x \neq y} \frac{\Pr_{P \sim \mathcal{P}}(P(x) \neq P(y))}{(1/2)\|x - y\|_1} \tag{4.4}$$

Moreover, for

$$\tau_k^* := \min_{\mathcal{P}} \tau_k(\mathcal{P}), \tag{4.5}$$

Karger et al. [56] showed that for every $\epsilon > 0$, there is an instance of multiway cut with k terminals for which the integrality gap of the CKR relaxation is at least $\tau_k^* - \epsilon$. Thus, Karger et al.'s result reduced the problem of constructing an integrality gap instance for multiway cut to proving a lower bound on τ_k^*.

Next, Angelidakis, Makarychev and Manurangsi [3] reduced the problem of lower bounding τ_k^* further by showing that it is sufficient to restrict our attention to *non-opposite cuts* as opposed to all cuts. A cut P is a *non-opposite cut* if $P(x) \in \text{Support}(x) \cup \{k+1\}$ for every $x \in \Delta_k$. Let $\Delta_{k,n} := \Delta_k \cap ((1/n)\mathbb{Z})^k$. For a distribution $\mathcal{P}$ over cuts, let

$$\tau_{k,n}(\mathcal{P}) := \max_{x,y \in \Delta_{k,n}, x \neq y} \frac{\Pr_{P \sim \mathcal{P}}(P(x) \neq P(y))}{(1/2)\|x - y\|_1}, \text{ and} \tag{4.6}$$

$$\tilde{\tau}_{k,n}^* := \min\{\tau_{k,n}(\mathcal{P}) : \mathcal{P} \text{ is a distribution over non-opposite cuts}\}. \tag{4.7}$$

Angelidakis, Makarychev and Manurangsi showed that $\tilde{\tau}_{k,n}^* - \tau_K^* = O(kn/(K-k))$ for all $K > k$. Thus, in order to lower bound τ_K^*, it suffices to lower bound $\tilde{\tau}_{k,n}^*$. That is, it suffices to construct an instance that has *large integrality gap against non-opposite cuts*.

As a central contribution, Angelidakis, Makarychev and Manurangsi constructed an instance showing that $\tilde{\tau}_{3,n}^* \geq 1.2 - O(1/n)$. Now, by setting $n = \Theta(\sqrt{K})$, we see that τ_K^* is at least $1.2 - O(1/\sqrt{K})$. Furthermore, they also showed that their lower bound on $\tilde{\tau}_{3,n}^*$ is almost tight, i.e., $\tilde{\tau}_{3,n}^* \leq 1.2$. The salient feature of this framework is that in order to improve the lower bound on τ_K^*, it suffices to improve $\tilde{\tau}_{k,n}^*$ for some $4 \leq k < K$.

The main technical challenge towards improving $\tilde{\tau}_{4,n}^*$ is that one has to deal with the 3-dimensional simplex Δ_4. Indeed, all known gap instances including that of Angelidakis, Makarychev and Manurangsi are constructed using the 2-dimensional simplex. In the 2-dimensional simplex, the properties of non-opposite cuts are easy to visualize, and their cut-values are convenient to characterize using simple geometric observations. However, the values of non-opposite cuts in the 3-dimensional simplex become difficult to characterize. Our main contribution is a simple argument based on properties of lower-dimensional simplices that overcomes this technical challenge. We construct a 3-dimensional instance that has gap larger than 1.2 against non-opposite cuts.

Theorem 4.3. $\tilde{\tau}^*_{4,n} \geq 1.20016 - O(1/n)$.

One of the by-products of our technique is a generalization of a result on *Sperner admissible labelings* due to Mirzakhani and Vondrák [69] that might be of independent combinatorial interest (see Theorem 4.6).

4.1.2 Outline of Ideas

In order to construct an instance that has gap strictly larger than 1.2 against non-opposite cuts, we present four instances that have large gap against different types of cuts, and then compute the convex combination of these instances that gives the best gap against all non-opposite cuts.

All of our four instances are defined as edge-weights on the graph $G = (V, E)$ with node set $\Delta_{4,n}$ and edge set $E_{4,n} := \{xy : x, y \in \Delta_{4,n}, \|x - y\|_1 = 2/n\}$, where the terminals are the four unit vectors. We identify $\Delta_{3,n}$ with the facet of $\Delta_{4,n}$ defined by $x_4 = 0$. Our first three instances are 2-dimensional instances, i.e. only edges induced by $\Delta_{3,n}$ have positive weight. The fourth instance has uniform weight on $E_{4,n}$.

We first explain the motivation behind Instances 1,2, and 4, since these are easy to explain. Let

$$L_{ij} := \{xy \in E_{4,n} : \text{Support}(x), \text{Support}(y) \subseteq \{i, j\}\}. \tag{4.8}$$

- Instance 1 is simply the instance of Angelidakis, Makarychev and Manurangsi [3] on $\Delta_{3,n}$. It has gap $1.2 - \frac{1}{n}$ against all non-opposite cuts, since non-opposite cuts in $\Delta_{4,n}$ induce non-opposite cuts on $\Delta_{3,n}$. Additionally, we show in Lemma 4.5 that the gap is strictly larger than 1.2 by a constant if the following two conditions hold:
  - there exist $i, j \in [3]$ such that L_{ij} contains only one edge whose end-nodes have different labels (a cut with this property is called a *non-fragmenting cut*), and
  - $\Delta_{3,n}$ has a lot of nodes with label 5.
- Instance 2 has uniform weight on L_{12}, L_{13} and L_{23}, and 0 on all other edges. Here, a cut in which each L_{ij} contains at least two edges whose end-nodes have different labels (a *fragmenting cut*) has large weight. Consequently, this instance has gap at least 2 against such cuts.
- Instance 4 has uniform weight on all edges in $E_{4,n}$. A beautiful result due to Mirzakhani and Vondrák [69] implies that non-opposite cuts with no node of label 5 have large weight. Consequently, this

instance has gap at least 3/2 against such cuts. We extend their result in Lemma 4.1 to show that the weight remains large if $\Delta_{3,n}$ has few nodes with label 5.

At first glance, the arguments above seem to already imply that a convex combination of these instances already gives a gap strictly larger than 1.2 for all non-opposite cuts. Unfortunately, this is not the case: using only these three instances, we could not obtain a gap better than 1.2. The issue is that a fragmenting cut may have near-minimum weight in Instance 1 and simultaneously very small weight in Instance 4. Instance 3 is constructed specifically against these types of cuts. It has positive uniform weight on 3 equilateral triangles, incident to e^1, e^2 and e^3 on the face $\Delta_{3,n}$. We call the edges of these triangles *red edges*. The side length of these triangles is a parameter, denoted by c, that is optimized at the end of the proof. Essentially, we show that if a non-opposite cut has small weight both on Instance 1 and Instance 4, then it must contain red edges.

Our lower bound of 1.20016 is obtained by optimizing the coefficients of the convex combination and the parameter c.

4.1.3 A 3-dimensional gap instance against non-opposite cuts

We will focus on the graph $G = (V, E)$ with the node set $V := \Delta_{4,n}$ being the discretized 3-dimensional simplex and the edge set $E_{4,n} := \{xy : x, y \in \Delta_{4,n}, \|x - y\|_1 = 2/n\}$. The four terminals $s_1, \dots, s_4$ will be the four nodes of the simplex, namely $s_i = e_i$ for $i \in [4]$. In this context, a cut is a function $P : V \to [5]$ such that $P(s_i) = i$ for all $i \in [4]$. The *cut-set* corresponding to P is defined as

$$\delta(P) := \{xy \in E_{4,n} : P(x) \neq P(y)\}. \tag{4.9}$$

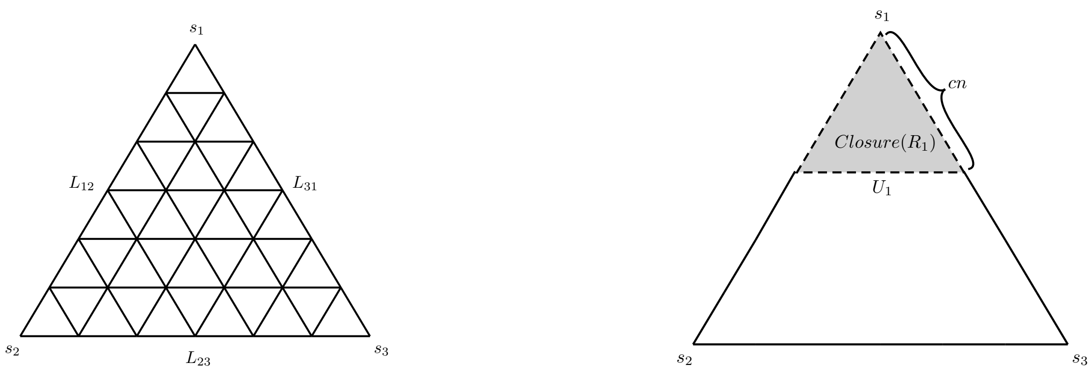
For a set S of nodes, we will also use $\delta(S)$ to denote the set of edges with exactly one end node in S. Given a weight function $w : E_{4,n} \to \mathbb{R}_+$, the *cost* of a cut P is $\sum_{e \in \delta(P)} w(e)$. Our goal is to come up with weights on the edges so that the resulting 4-way cut instance has gap at least 1.20016 against non-opposite cuts.

We will denote the boundary nodes and the boundary edges between terminals s_i and s_j as V_{ij} and L_{ij} respectively, i.e.,

$$V_{ij} := \left\{x \in \Delta_{4,n} : \text{Support}(x) \subseteq \{i, j\}\right\}, \text{ and} \tag{4.10}$$

$$L_{ij} := \{xy \in E_{4,n} : \text{Support}(x), \text{Support}(y) \subseteq \{i, j\}\}. \tag{4.11}$$

Let $c \in (0, 1/2)$ be a constant to be fixed later, such that cn is integral. For each $k \in [3]$ and $\{i, j\} =$



(a) One face of the simplex with edge-sets L_{12}, L_{23} and L_{31}.

(b) Definition of red nodes and edges near terminal s_1. Dashed part corresponds to (R_1, Γ_1).

Figure 4.3: Notation on Face(s_1, s_2, s_3).

$[3] \setminus \{k\}$, we define the node sets

$$U_k := \{x \in \Delta_{4,n} : x_4 = 0,\ x_k = 1-c,\ x_i = \lambda c,\ x_j = (1-\lambda)c \text{ for some } \lambda \in [0,1]\}, \quad (4.12)$$

$$R_k := U_k \cup \{x \in V_{ik} \cup V_{jk} : x_k \geq 1-c\}, \text{ and} \quad (4.13)$$

$$\text{Closure}(R_k) := \{x \in \Delta_{4,n} : x_4 = 0,\ x_k \geq 1-c,\ 0 \leq x_i, x_j \leq c\}. \quad (4.14)$$

Moreover, for each $k \in [3]$ and $\{i,j\} = [3] \setminus \{k\}$, we define the edge set

$$\Gamma_k := \left\{xy \in E_{4,n} : x \in V_{ik} \cap R_k,\ y = x + \frac{1}{n}e_k - \frac{1}{n}e_i\right\} \quad (4.15)$$

$$\cup \left\{xy \in E_{4,n} : x \in V_{jk} \cap R_k,\ y = x + \frac{1}{n}e_k - \frac{1}{n}e_j\right\} \quad (4.16)$$

$$\cup \left\{xy \in E_{4,n} : x \in U_k,\ y = x + \frac{1}{n}e_i - \frac{1}{n}e_j\right\}. \quad (4.17)$$

We will refer to the nodes in R_k as red[1] nodes near terminal s_k and the edges in Γ_k as the red edges near terminal s_k (see Figure 4.3b). Let Face(s_1, s_2, s_3) denote the subgraph of G induced by the nodes whose support is contained in $\{1,2,3\}$. We emphasize that the red edges and red nodes are present only in Face(s_1, s_2, s_3) and that the total number of red edges is exactly $9cn$.

Gap instance as a convex combination: Our gap instance is a convex combination of the following four instances.

1. **Instance I_1.** Our first instance constitutes the 3-way cut instance constructed by Angelidakis, Makarychev and Manurangsi [3] that has gap 1.2 against non-opposite cuts. To ensure that the total weight of all the edges in their instance is exactly n, we will scale their instance by 6/5. Let us denote the resulting instance as J. In I_1, we simply use the instance J on Face(s_1, s_2, s_3) and set

[1] We use the term "red" as a convenient way for the reader to remember these nodes and edges. The exact color is irrelevant.

the weights of the rest of the edges in $E_{4,n}$ to be zero.

2. **Instance** I_2**.** In this instance, we set the weights of the edges in L_{12}, L_{23}, L_{13} to be $1/3$ and the weights of the rest of the edges in $E_{4,n}$ to be zero.

3. **Instance** I_3**.** In this instance, we set the weights of the red edges to be $1/9c$ and the weights of the rest of the edges in $E_{4,n}$ to be zero.

4. **Instance** I_4**.** In this instance, we set the weight of every edge in $E_{4,n}$ to be $1/n^2$.

We note that the total weight of all edges in each of the above instances is $n + O(1)$. For multipliers $\lambda_1, \ldots, \lambda_4 \geq 0$ to be chosen later that will satisfy $\sum_{i=1}^{4} \lambda_i = 1$, let the instance I be the convex combination of the above four instances, i.e., $I = \lambda_1 I_1 + \lambda_2 I_2 + \lambda_3 I_3 + \lambda_4 I_4$. By the properties of the four instances, it immediately follows that the total weight of all edges in the instance I is also $n + O(1)$.

Gap of the Convex Combination: The following theorem is the main result of this section.

Theorem 4.4. *For every $n \geq 10$ and $c \in (0, 1/2)$ such that cn is integer, every non-opposite cut on I has cost at least the minimum of the following two terms:*

(i) $\lambda_2 + (1.2 - \frac{1}{n})\lambda_1 + \min_{\alpha \in \left[0, \frac{1}{2}\right]} \left\{0.4\alpha\lambda_1 + 3\left(\frac{1}{2} - \alpha\right)\lambda_4\right\}$

(ii) $2\lambda_2 + (1.2 - \frac{2}{n})\lambda_1 + 3\min\left\{\frac{2\lambda_3}{9c}, \min_{\alpha \in \left[0, \frac{c^2}{2}\right]} \left\{0.4\alpha\lambda_1 + 3\left(\frac{c^2}{2} - \alpha\right)\lambda_4\right\}\right\}$

Before proving Theorem 4.4, we see its consequence.

Corollary 4.3. *There exist constants $c \in (0, 1/2)$ and $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$ with $\sum_{i=1}^{4} \lambda_i = 1$ such that the cost of every non-opposite cut in the resulting convex combination I is at least $1.20016 - O(1/n)$.*

Proof: Suppose that cn is integer. By Theorem 4.4, the cost of every non-opposite cut on I is at least $\min\{\mu_1, \mu_2\} - O(1/n)\lambda_1$ where

$$\mu_1 := \lambda_2 + 1.2\lambda_1 + \min_{\alpha \in \left[0, \frac{1}{2}\right]} \left\{0.4\alpha\lambda_1 + 3\left(\frac{1}{2} - \alpha\right)\lambda_4\right\} \text{ and } \tag{4.18}$$

$$\mu_2 := 2\lambda_2 + 1.2\lambda_1 + 3\min\left\{\frac{2\lambda_3}{9c}, \min_{\alpha \in \left[0, \frac{c^2}{2}\right]} \left\{0.4\alpha\lambda_1 + 3\left(\frac{c^2}{2} - \alpha\right)\lambda_4\right\}\right\}. \tag{4.19}$$

In order to get the largest possible gap from this, we will identify constants $c \in (0, 1/2)$ and $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$ such that $\sum_{i=1}^{4} \lambda_i = 1$ so that $\min\{\mu_1, \mu_2\}$ is maximized. We mention that the optimal choice for λ_4 is

$$\lambda_4 = \frac{0.4\lambda_1}{3}. \tag{4.20}$$

We avoid going over the details behind this choice of λ_4 as it is not insightful. After this choice, we can simplify μ_1 and μ_2 as

$$\mu_1 = \lambda_2 + 1.4\lambda_1, \text{ and} \tag{4.21}$$

$$\mu_2 = 2\lambda_2 + 1.2\lambda_1 + 3\min\left\{\frac{2\lambda_3}{9c}, 0.2c^2\lambda_1\right\}. \tag{4.22}$$

Our goal is to maximize $\min\{\mu_1, \mu_2\}$. Again, we omit the details and mention that the optimal choice for λ_3 is to set

$$\frac{2\lambda_3}{9c} = 0.2c^2\lambda_1, \text{ i.e., } \lambda_3 = 0.9c^3\lambda_1. \tag{4.23}$$

With this choice of λ_3, we have that

$$\mu_1 = \lambda_2 + 1.4\lambda_1, \text{ and} \tag{4.24}$$

$$\mu_2 = 2\lambda_2 + 1.2\lambda_1 + 0.6c^2\lambda_1. \tag{4.25}$$

Now, in order to maximize $\min\{\mu_1, \mu_2\}$, the optimal choice is to ensure that $\mu_1 = \mu_2$, which gives

$$\lambda_2 = (0.2 - 0.6c^2)\lambda_1 \tag{4.26}$$

and

$$\min\{\mu_1, \mu_2\} = (1.6 - 0.6c^2)\lambda_1. \tag{4.27}$$

Thus, we now have λ_2, λ_3 and λ_4 as a function of λ_1 and c. We further note that $\sum_{i=1}^{4} \lambda_i = 1$. Substituting the values of λ_2, λ_3 and λ_4 into this equation and solving for λ_1, we obtain that

$$\lambda_1 = \frac{1}{\frac{4}{3} - 0.6c^2 + 0.9c^3}. \tag{4.28}$$

With this choice of λ_1, we have that

$$\min\{\mu_1, \mu_2\} = \frac{1.6 - 0.6c^2}{\frac{4}{3} - 0.6c^2 + 0.9c^3}. \tag{4.29}$$

Maximizing the RHS function, which is a function of c, subject to the constraint that $c \in (0, 1/2)$, we obtain that

$$\min\{\mu_1, \mu_2\} = 1.20016 \tag{4.30}$$

at $c = 0.074125$. Substituting back, we obtain the constants

$$\lambda_1 = 0.751652, \tag{4.31}$$
$$\lambda_2 = 0.147852, \tag{4.32}$$
$$\lambda_3 = 0.000275, \text{ and} \tag{4.33}$$
$$\lambda_4 = 0.100221. \tag{4.34}$$

We note that the function in (4.29) is differentiable, so changing c to the nearest multiple of $1/n$ decreases (4.29) by $O(1/n)$. Hence, the cost of every non-opposite cut on the instance I given by the above values of c and multipliers $\lambda_1, \ldots, \lambda_4$ is at least $1.20016 - O(1/n)$.

Corollary 4.3 implies Theorem 4.3 in the following way.

Proof:[Theorem 4.3] Let $w(e)$ denote the weight of edge $e \in E_{4,n}$ in the instance I defined above. Let $\mathcal{P}$ be a distribution over non-opposite cuts. Then

$$\begin{aligned}
\tau_{k,n}(\mathcal{P}) &= \max_{x,y \in \Delta_{k,n}, x \neq y} \frac{\Pr_{P\sim\mathcal{P}}(P(x) \neq P(y))}{(1/2)\|x-y\|_1} && (4.35)\\
&\geq \max_{xy \in E_{4,n}} \frac{\Pr_{P\sim\mathcal{P}}(P(x) \neq P(y))}{(1/2)\|x-y\|_1} && (4.36)\\
&= \max_{xy \in E_{4,n}} \frac{\Pr_{P\sim\mathcal{P}}(P(x) \neq P(y))}{1/n} && (4.37)\\
&\geq \sum_{xy \in E_{4,n}} \frac{w(xy)\Pr_{P\sim\mathcal{P}}(P(x) \neq P(y))}{(1/n)(\sum_{e \in E_{4,n}} w(e))} && (4.38)\\
&\geq \frac{1.20016 - O(1/n)}{1 + O(1/n)} = 1.20016 - O(1/n), && (4.39)
\end{aligned}$$

where the last inequality follows from Corollary 4.3 and the fact that the total weight of all edges in the instance I is $n + O(1)$.

The rest of the section is devoted to proving Theorem 4.4. We rely on two main ingredients in the proof. The first ingredient is a statement about non-opposite cuts in the 3-dimensional discretized simplex. We prove this in Section 4.1.4, where we also give a generalization to higher dimensional simplices, which might be of independent interest.

Lemma 4.1. *Let P be a non-opposite cut on $\Delta_{4,n}$ with $\alpha(n+1)(n+2)$ nodes from Face(s_1, s_2, s_3) labeled as 1, 2, or 3 for some $\alpha \in [0, 1/2]$. Then, $|\delta(P)| \geq 3\alpha n(n+1)$.*

The next ingredient involves properties of the 3-way cut instance constructed by Angelidakis, Makarychev and Manurangsi [3]. We need two properties that are summarized in Lemma 4.5 and Corollary 4.5. We prove these properties in Section 4.1.5. We define a cut $Q : \Delta_{3,n} \to [4]$ to be a *fragmenting cut* if $|\delta(Q) \cap L_{ij}| \geq 2$ for every distinct $i, j \in [3]$; otherwise it is a non-fragmenting cut. We recall that J denotes the instance obtained from the 3-way cut instance of Angelidakis, Makarychev and

Manurangsi by scaling it up by 6/5.

Lemma 4.2. *Let $Q : \Delta_{3,n} \to [4]$ be a non-opposite cut with αn^2 nodes labeled as 4. If Q is a non-fragmenting cut and $n \geq 10$, then the cost of Q on J is at least $1.2 - \frac{1}{n} + 0.4\alpha$.*

Corollary 4.4. *Let $Q : \Delta_{3,n} \to [4]$ be a non-opposite cut and $n \geq 10$. For each $i \in [3]$, let*

$$A_i := \begin{cases} \{v \in \mathit{Closure}(R_i) : Q(v) = 4\} & \text{if } \delta(Q) \cap \Gamma_i = \emptyset, \\ \emptyset & \text{otherwise}. \end{cases} \tag{4.40}$$

Then, the cost of Q on J is at least $1.2 - \frac{2}{n} + 0.4\sum_{i=1}^{3} |A_i|/n^2$.

We now have the ingredients to prove Theorem 4.4.

Proof:[Theorem 4.4] Let $P : \Delta_{4,n} \to [5]$ be a non-opposite cut. Let Q be the cut P restricted to $\text{Face}(s_1, s_2, s_3)$, i.e., for every $v \in \Delta_{4,n}$ with $\text{Support}(v) \subseteq [3]$, let

$$Q(v) := \begin{cases} P(v) & \text{if } P(v) \in \{1,2,3\}, \\ 4 & \text{if } P(v) = 5. \end{cases} \tag{4.41}$$

We consider two cases.

Case 1: Q is a non-fragmenting cut. Let the number of nodes in $\text{Face}(s_1, s_2, s_3)$ that are labeled by Q as 4 (equivalently, labeled by P as 5) be $\alpha(n+1)(n+2)$ for some $\alpha \in \left[0, \frac{1}{2}\right]$. Since $|\{x \in \text{Face}(s_1, s_2, s_3) : Q(x) = 4\}| \geq \alpha n^2$, Lemma 4.5 implies that the cost of Q on J, and hence the cost of P on I_1, is at least $1.2 - \frac{1}{n} + 0.4\alpha$. Moreover, the cost of P on I_2 is at least 1 since at least one edge in L_{ij} should be in $\delta(P)$ for every pair of distinct $i, j \in [3]$. To estimate the cost on I_4, we observe that the number of nodes on $\text{Face}(s_1, s_2, s_3)$ labeled by P as 1, 2, or 3 is $(1/2 - \alpha)(n+1)(n+2)$. By Lemma 4.1, we have that $|\delta(P)| \geq 3(1/2 - \alpha)n(n+1)$ and thus, the cost of P on I_4 is at least $3(1/2 - \alpha)$. Therefore, the cost of P on the convex combination instance I is at least

$$\lambda_2 + \left(1.2 - \frac{1}{n}\right)\lambda_1 + \min_{\alpha \in \left[0, \frac{1}{2}\right]} \left\{0.4\alpha\lambda_1 + 3\left(\frac{1}{2} - \alpha\right)\lambda_4\right\}. \tag{4.42}$$

Case 2: Q is a fragmenting cut. Then, the cost of P on I_2 is at least 2 as a fragmenting cut contains at least 2 edges from each L_{ij} for distinct $i, j \in [3]$.

We will now compute the cost of P on the other instances. Let $r := |\{i \in [3] : \delta(P) \cap \Gamma_i \neq \emptyset\}|$, i.e., r is the number of red triangles that are intersected by the cut P. We will derive lower bounds on the cost of the cut in each of the three instances I_1, I_3 and I_4 based on the value of $r \in \{0, 1, 2, 3\}$. For each

$i \in [3]$, let

$$A_i := \begin{cases} \{v \in \text{Closure}(R_i) : P(v) = 5\} & \text{if } \delta(P) \cap \Gamma_i = \emptyset, \\ \emptyset & \text{otherwise,} \end{cases} \tag{4.43}$$

and let $\alpha := |A_1 \cup A_2 \cup A_3|/((n+1/c)(n+2/c))$. Since $c < 1/2$, the sets A_i and A_j are disjoint for distinct $i, j \in [3]$. We note that $\alpha \in [0, (3-r)c^2/2]$ since $|A_i| \le (cn+1)(cn+2)/2$ and $A_i \cap A_j = \emptyset$.

In order to lower bound the cost of P on I_1, we will use Corollary 4.5. Recall that Q is the cut P restricted to Face(s_1, s_2, s_3), so the cost of P on I_1 is the same as the cost of Q on J. Moreover, by Corollary 4.5, the cost of Q on J is at least $1.2 - \frac{2}{n} + 0.4\alpha$, because $\alpha \le \sum_{i=1}^{3} |A_i|/n^2$. Hence, the cost of P on I_1 is at least $1.2 - \frac{2}{n} + 0.4\alpha$.

The cost of P on I_3 is at least $2r/9c$ by the following claim.

Claim 4.1. *Let $i \in [3]$. If $\delta(P) \cap \Gamma_i \neq \emptyset$, then $|\delta(P) \cap \Gamma_i| \ge 2$.*

Proof: The subgraph (R_i, Γ_i) is a cycle. If $P(x) \neq P(y)$ for some $xy \in \Gamma_i$, then the path $\Gamma_i - xy$ must also contain two consecutive nodes labeled differently by P.

Next, we compute the cost of P on the instance I_4. If $r = 3$, then the cost of P on I_4 is at least 0. Suppose $r \in \{0, 1, 2\}$. For a red triangle $i \in [3]$ with $\delta(P) \cap \Gamma_i = \emptyset$, we have at least $(cn+1)(cn+2)/2 - |A_i|$ nodes from Closure(R_i) that are labeled as 1, 2, or 3. Moreover, the nodes in Closure(R_i) and Closure(R_j) are disjoint for distinct $i, j \in [3]$. Hence, the number of nodes in Face(s_1, s_2, s_3) that are labeled as 1, 2, or 3 is at least $(3-r)(cn+1)(cn+2)/2 - \alpha(n+1/c)(n+2/c) = ((3-r)c^2/2 - \alpha)(n+1/c)(n+2/c)$, which is at least $((3-r)c^2/2 - \alpha)(n+1)(n+2)$, since $c \le 1$. Therefore, by Lemma 4.1, we have $|\delta(P)| \ge 3((3-r)c^2/2 - \alpha)n^2$ and thus, the cost of P on I_4 is at least $3((3-r)c^2/2 - \alpha)$.

Thus, the cost of P on the convex combination instance I is at least $2\lambda_2 + (1.2 - \frac{2}{n})\lambda_1 + \gamma(r, \alpha)$ for some $\alpha \in [0, (3-r)c^2/2]$, where

$$\gamma(r, \alpha) := \begin{cases} \frac{6\lambda_3}{9c}, & \text{if } r = 3, \\ 0.4\alpha\lambda_1 + \frac{2r}{9c}\lambda_3 + 3\left(\frac{(3-r)c^2}{2} - \alpha\right)\lambda_4, & \text{if } r \in \{0, 1, 2\}. \end{cases} \tag{4.44}$$

In particular, the cost of P on the convex combination instance I is at least $2\lambda_2 + (1.2 - \frac{2}{n})\lambda_1 + \gamma^*$, where

$$\gamma^* := \min_{r \in \{0,1,2,3\}} \min_{\alpha \in \left[0, \frac{(3-r)c^2}{2}\right]} \gamma(r, \alpha). \tag{4.45}$$

Now, Claim 4.2 completes the proof of the theorem.

Claim 4.2.

$$\gamma^* \ge 3 \min\left\{\frac{2\lambda_3}{9c}, \min_{\alpha \in \left[0, \frac{c^2}{2}\right]} \left\{0.4\alpha\lambda_1 + 3\left(\frac{c^2}{2} - \alpha\right)\lambda_4\right\}\right\}. \tag{4.46}$$

Proof: Let $\gamma(r) := \min_{\alpha \in [0,(3-r)c^2/2]} \gamma(r, \alpha)$. If $r = 3$, then the claim is clear. We consider the three remaining cases.

(I) Say $r = 0$. Then,

$$\gamma(0) = \min_{\alpha \in \left[0, \frac{3c^2}{2}\right]} \left\{ 0.4\alpha\lambda_1 + 3\left(\frac{3c^2}{2} - \alpha\right)\lambda_4 \right\} = 3 \min_{\alpha \in \left[0, \frac{c^2}{2}\right]} \left\{ 0.4\alpha\lambda_1 + 3\left(\frac{c^2}{2} - \alpha\right)\lambda_4 \right\}. \quad (4.47)$$

(II) Say $r = 1$. Then,

$$\begin{aligned}
\gamma(1) &= \min_{\alpha \in [0,c^2]} \left\{ 0.4\alpha\lambda_1 + \frac{2}{9c}\lambda_3 + 3\left(c^2 - \alpha\right)\lambda_4 \right\} && (4.48)\\
&= \frac{2}{9c}\lambda_3 + \min_{\alpha \in \left[0, \frac{c^2}{2}\right]} \left\{ 2 \cdot 0.4\alpha\lambda_1 + 3\left(c^2 - 2\alpha\right)\lambda_4 \right\} && (4.49)\\
&= \frac{2}{9c}\lambda_3 + 2 \min_{\alpha \in \left[0, \frac{c^2}{2}\right]} \left\{ 0.4\alpha\lambda_1 + 3\left(\frac{c^2}{2} - \alpha\right)\lambda_4 \right\} && (4.50)\\
&\geq 3 \min \left\{ \frac{2\lambda_3}{9c}, \min_{\alpha \in \left[0, \frac{c^2}{2}\right]} \left\{ 0.4\alpha\lambda_1 + 3\left(\frac{c^2}{2} - \alpha\right)\lambda_4 \right\} \right\}, && (4.51)
\end{aligned}$$

where the last inequality is from the identity $x + 2y \geq 3\min\{x, y\}$ for all $x, y \in \mathbb{R}$.

(III) Say $r = 2$. Then,

$$\begin{aligned}
\gamma(2) &= \min_{\alpha \in \left[0, \frac{c^2}{2}\right]} \left\{ 0.4\alpha\lambda_1 + \frac{4}{9c}\lambda_3 + 3\left(\frac{c^2}{2} - \alpha\right)\lambda_4 \right\} && (4.52)\\
&= \frac{4}{9c}\lambda_3 + \min_{\alpha \in \left[0, \frac{c^2}{2}\right]} \left\{ 0.4\alpha\lambda_1 + 3\left(\frac{c^2}{2} - \alpha\right)\lambda_4 \right\} && (4.53)\\
&\geq 3 \min \left\{ \frac{2\lambda_3}{9c}, \min_{\alpha \in \left[0, \frac{c^2}{2}\right]} \left\{ 0.4\alpha\lambda_1 + 3\left(\frac{c^2}{2} - \alpha\right)\lambda_4 \right\} \right\}, && (4.54)
\end{aligned}$$

where the last inequality is from the identity $2x + y \geq 3\min\{x, y\}$ for all $x, y \in \mathbb{R}$.

4.1.4 Size of non-opposite cuts in $\Delta_{k,n}$

In this section, we prove Lemma 4.1. In fact, we prove a general result for $\Delta_{k,n}$, that may be useful for obtaining improved bounds by considering higher dimensional simplices. Our result is an extension of a theorem of Mirzakhani and Vondrák [69] on Sperner-admissible labelings.

A labeling $\ell : \Delta_{k,n} \to [k]$ is *Sperner-admissible* if $\ell(x) \in \text{Support}(x)$ for every $x \in \Delta_{k,n}$. We say that $x \in \Delta_{k,n}$ has an *inadmissible label* if $\ell(x) \notin \text{Support}(x)$. Let $H_{k,n}$ denote the hypergraph whose node set is $\Delta_{k,n}$ and whose hyperedge set is

$$\mathcal{E} := \left\{ \left\{ \frac{n-1}{n}x + \frac{1}{n}e_1, \frac{n-1}{n}x + \frac{1}{n}e_2, \dots, \frac{n-1}{n}x + \frac{1}{n}e_k \right\} : x \in \Delta_{k,n-1} \right\}. \tag{4.55}$$

Each hyperedge $e \in \mathcal{E}$ has k nodes, and if $x, y \in e$, then there exist distinct $i, j \in [n]$ such that $x - y = \frac{1}{n}e_i - \frac{1}{n}e_j$. We remark that $H_{k,n}$ has $\binom{n+k-1}{k-1}$ nodes and $\binom{n+k-2}{k-1}$ hyperedges. Geometrically, the hyperedges correspond to simplices that are translates of each other and share at most one node. Given a labeling ℓ, a hyperedge of $H_{k,n}$ is *monochromatic* if all of its nodes have the same label. Mirzakhani and Vondrák proved the following result.

Theorem 4.5 (Proposition 2.1 in [69])**.** *Let ℓ be a Sperner-admissible labeling of $\Delta_{k,n}$. Then, the number of monochromatic hyperedges in $H_{k,n}$ is at most $\binom{n+k-3}{k-1}$, and therefore the number of non-monochromatic hyperedges is at least $\binom{n+k-3}{k-2}$.*

Our result is an extension to the case when there are some inadmissible labels on a single face of $\Delta_{k,n}$. We will denote the nodes $x \in \Delta_{k,n}$ with $\mathrm{Support}(x) \subseteq [k-1]$ as $\mathrm{Face}(s_1, \dots, s_{k-1})$.

Theorem 4.6. *Let ℓ be a labeling of $\Delta_{k,n}$ such that all inadmissible labels are on $\mathrm{Face}(s_1, \dots, s_{k-1})$ and the number of nodes with inadmissible labels is $\beta\frac{(n+k-2)!}{n!}$ for some β. Then, the number of non-monochromatic hyperedges of $H_{k,n}$ is at least*

$$\left(\frac{1}{(k-2)!} - \beta\right)\frac{(n+k-3)!}{(n-1)!}. \tag{4.56}$$

Proof: Let $Z := \{x \in \mathrm{Face}(s_1, \dots, s_{k-1}) : \ell(x) = k\}$, i.e. Z is the set of nodes in $\mathrm{Face}(s_1, \dots, s_{k-1})$ having an inadmissible label. Let us call a hyperedge of $H_{k,n}$ *inadmissible* if the label of one of its nodes is inadmissible.

Claim 4.3. *There are at most $\beta\frac{(n+k-3)!}{(n-1)!}$ inadmissible monochromatic hyperedges.*

Proof: Let $\mathcal{E}'$ be the set of inadmissible monochromatic hyperedges. Each hyperedge $e \in \mathcal{E}'$ has exactly $k-1$ nodes from $\mathrm{Face}(s_1, \dots, s_{k-1})$ and they all have the same label as e is monochromatic. Thus, each $e \in \mathcal{E}'$ contains $k-1$ nodes from Z. We define an injective map $\varphi : \mathcal{E}' \to Z$ by letting $\varphi(e)$ to be the node $x \in e \cap Z$ with the largest 1st coordinate. Notice that if $x = \varphi(e)$, then the other nodes of e are $x - (1/n)e_1 + (1/n)e_i$ $(i = 2, \dots, k)$, and all but the last one are in Z. In particular, x_1 is positive.

Let $Z' \subseteq Z$ be the image of φ. For $x \in Z$ and $i \in \{2, \dots, k-1\}$, let

$$Z_x^i := \{y \in Z : y_j = x_j \ \forall j \in [k-1] \setminus \{1, i\}\}. \tag{4.57}$$

Since $y_k = 0$ and $\|y\|_1 = 1$ for every $y \in Z$, the nodes of Z_x^i are on a line containing x. It also follows that $Z_x^i \cap Z_x^j = \{x\}$ if $i \neq j$. Let

$$Z'' := \{x \in Z : \exists i \in \{2, \ldots, k-1\} \text{ such that } x_i \geq y_i \ \forall y \in Z_x^i\}. \tag{4.58}$$

We observe that if $x \in Z'$, then for each $i \in \{2, \ldots, k-1\}$, the node $y = x - (1/n)e_1 + (1/n)e_i$ is in Z and hence, $y \in Z_x^i$ with $y_i > x_i$. In particular, this implies that $Z' \cap Z'' = \emptyset$. We now compute an upper bound on the size of $Z \setminus Z''$, which gives an upper bound on the size of Z' and hence also on the size of $\mathcal{E}'$, as $|Z'| = |\mathcal{E}'|$. For each node $x \in Z \setminus Z''$ and for every $i \in \{2, \ldots, k-1\}$, let z_x^i be the node in $Z'' \cap Z_x^i$ with the largest ith coordinate. Clearly $z_x^i \neq z_x^j$ if $i \neq j$, because $Z_x^i \cap Z_x^j = \{x\}$.

For given $y \in Z''$ and $i \in \{2, \ldots, k-1\}$, we want to bound the size of $S := \{x \in Z \setminus Z'' : z_x^i = y\}$. Consider $a \in S$. Then, $z_a^i = y$ implies that the node in $Z'' \cap Z_a^i$ with the largest i-th coordinate is y. That is, $y_j = a_j$ for all $j \in [k-1] \setminus \{1, i\}$ and moreover $y_i \geq a_i$. If $y_i = a_i$, then $y = a$, so a is in Z'' which contradicts $a \in S$. Thus, $y_i > a_i$ for any $a \in S$, i.e. the nodes in S are on the line Z_y^i and their i-th coordinate is strictly smaller than y_i. This implies that $|S| \leq n y_i$. Consequently, the size of the set $\{x \in Z \setminus Z'' : y = z_x^i \text{ for some } i \in \{2, \ldots, k-1\}\}$ is at most n, since $\sum_{i=2}^{k-2} y_i \leq \|y\|_1 = 1$.

For each $x \in Z \setminus Z''$, we defined $k-2$ distinct nodes $z_x^2, \ldots, z_x^{k-1} \in Z''$. Moreover, for each $y \in Z''$, we have at most n distinct nodes x in $Z \setminus Z''$ for which there exists $i \in \{2, \ldots, k-1\}$ such that $y = z_x^i$. Hence, $(k-2)|Z \setminus Z''| \leq n|Z''|$, and therefore $|Z \setminus Z''| \leq (n/(n+k-2))|Z|$. This gives

$$|\mathcal{E}'| = |Z'| \leq |Z \setminus Z''| \leq \frac{n}{n+k-2}|Z| \leq \beta \frac{(n+k-2)!}{n!} \frac{n}{n+k-2} = \beta \frac{(n+k-3)!}{(n-1)!}, \tag{4.59}$$

as required.

Let ℓ' be a Sperner-admissible labeling obtained from ℓ by changing the label of each node in Z to an arbitrary admissible label. By Theorem 4.5, the number of monochromatic hyperedges for ℓ' is at most $\binom{n+k-3}{k-1}$. By combining this with the claim, we get that the number of monochromatic hyperedges for ℓ is at most $\binom{n+k-3}{k-1} + \beta \frac{(n+k-3)!}{(n-1)!}$. Since $H_{k,n}$ has $\binom{n+k-2}{k-1}$ hyperedges, the number of non-monochromatic hyperedges is at least

$$\binom{n+k-2}{k-1} - \binom{n+k-3}{k-1} - \beta \frac{(n+k-3)!}{(n-1)!} = \binom{n+k-3}{k-2} - \beta \frac{(n+k-3)!}{(n-1)!} \tag{4.60}$$

$$= \left(\frac{1}{(k-2)!} - \beta\right) \frac{(n+k-3)!}{(n-1)!}. \tag{4.61}$$

We now derive Lemma 4.1 from Theorem 4.6. We restate Lemma 4.1 for convenience.

Lemma 4.1. *Let P be a non-opposite cut on $\Delta_{4,n}$ with $\alpha(n+1)(n+2)$ nodes from Face(s_1, s_2, s_3) labeled as 1, 2, or 3 for some $\alpha \in [0, 1/2]$. Then, $|\delta(P)| \geq 3\alpha n(n+1)$.*

Proof:[Lemma 4.1] Let ℓ be the labeling of $\Delta_{4,n}$ obtained from P by setting $\ell(x) = 4$ if $P(x) = 5$, and

$\ell(x) = P(x)$ otherwise. This is a labeling with $(\frac{1}{2} - \alpha)(n+1)(n+2)$ nodes having an inadmissible label, all on Face(s_1, s_2, s_3). We apply Theorem 4.6 with parameters $k = 4$, $\beta = \frac{1}{2} - \alpha$, and the labeling ℓ. By the theorem, the number of non-monochromatic hyperedges in $H_{4,n} = (\Delta_{4,n}, \mathcal{E})$ under labeling ℓ is at least $\alpha n(n+1)$.

We observe that for each hyperedge $e = \{u_1, u_2, u_3, u_4\} \in \mathcal{E}$, the subgraph $G[e]$ induced by the nodes in e contains 6 edges. Also, for any two hyperedges e_1 and e_2, the edges in the induced subgraphs $G[e_1]$ and $G[e_2]$ are disjoint as e_1 and e_2 can share at most one node. Moreover, for each non-monochromatic hyperedge $e \in \mathcal{E}$, at least 3 edges of $G[e]$ are in $\delta(P)$. Thus, the number of edges of G that are in $\delta(P)$ is at least $3\alpha n(n+1)$.

4.1.5 Properties of the 3-way cut instance in [3]

In this section, we prove Lemma 4.5 and Corollary 4.5 which are properties of the gap instance in [3].

The gap instance in [3]: We start by summarizing the relevant background about the gap instance against non-opposite 3-way cuts designed by Angelidakis, Makarychev and Manurangsi [3]. For our purposes, we scale the costs of their instance by a factor of 6/5 as it will be convenient to work with them. We describe this scaled instance now.

Let $\mathcal{G} = (\Delta_{3,n}, E_{3,n})$ where $E_{3,n} := \{xy : x, y \in \Delta_{3,n}, \|x - y\|_1 = 2/n\}$. Their instance is obtained by dividing $\Delta_{3,n}$ into a middle hexagon $H := \{x \in \Delta_{3,n} : x_i \leq 2/3 \ \forall\ i \in [3]\}$ and three corner triangles T_1, T_2, T_3, where $T_i := \{x \in \Delta_{3,n} : x_i > 2/3\}$. To define the edge costs, we let $\rho := 3/5n$. The cost of the edges in $\mathcal{G}[H]$ is ρ. The cost of the non-boundary edges in $\mathcal{G}[T_i]$ that are not parallel to the opposite side of e^i is also ρ. The cost of the non-boundary edges in $\mathcal{G}[T_i]$ that are parallel to the opposite side of e_i are zero. The cost of the boundary edges in L_{ij} are as follows: the edge closest to e^i has cost $(n/3)\rho$, the second closest edge to e^i has cost $(n/3 - 1)\rho$, and so on. See Figure 4.4 for an example. We will denote the resulting graph with edge-costs as J. The cost of a subset F of edges on the instance J is $Cost_J(F) := \sum_{e \in F} w(e)$.

For a subset of edges $F \subset E_{3,n}$, let $\mathcal{G} - F$ denote the graph $(\Delta_{3,n}, E_{3,n} \setminus F)$. We need the following two results about their instance.

Lemma 4.3. *[3] For every non-opposite cut $Q : \Delta_{3,n} \to [4]$, the cost of Q on instance J is at least* $1.2 - \frac{1}{n}$.

Lemma 4.4. *For $\{i, j, k\} = [3]$ and for every subset F of edges in $E_{3,n}$ such that s_i cannot reach V_{jk} in $\mathcal{G} - F$, the cost of F on J is at least* $0.4 - (\frac{1}{n})/3$.

Although Lemma 4.4 is not explicitly stated in [3], its proof appears under Case 1 in the Proof of Lemma 3 of [3]. The factor 0.4 that we have here is because, we scaled their costs by a factor of 6/5.

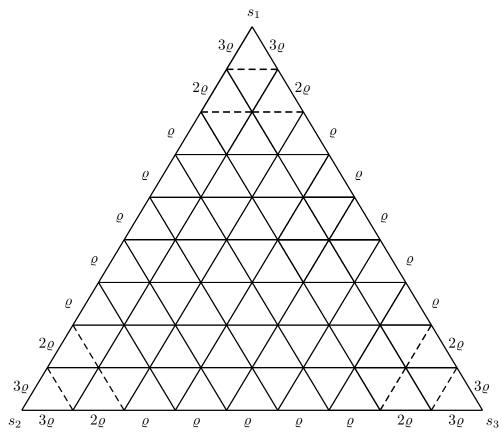


Figure 4.4: The instance in [3] for $n = 9$.

We next define non-oppositeness as a property of the cut-set as it will be convenient to work with this property for cut-sets rather than for cuts.

Definition 4.1. *A set $F \subseteq E_{3,n}$ of edges is a non-opposite cut-set if there is no path from s_1 to V_{23} in $\mathcal{G} - F$, no path from s_2 to V_{13} in $\mathcal{G} - F$, and no path from s_3 to V_{12} in $\mathcal{G} - F$.*

We summarize the connection between non-opposite cut-sets and non-opposite cuts.

Proposition 4.1. *(i) If $Q : \Delta_{3,n} \to [4]$ is a non-opposite cut, then $\delta(Q)$ is a non-opposite cut-set.*

(ii) For every non-opposite cut-set $F \subseteq E_{3,n}$, the cost of F on instance J is at least $1.2 - \frac{1}{n}$.

Proof:

(i) Suppose not. Without loss of generality, suppose there exists a path from s_1 to V_{23} in $\mathcal{G} - \delta(Q)$. Then, by the definition of $\delta(Q)$, all nodes of the path have the same label, so there exists a node $u \in V_{23}$ that is labeled as 1, contradicting the fact that Q is a non-opposite cut.

(ii) Consider a labeling $L : \Delta_{3,n} \to [4]$ where $L = i$ if the node v is reachable from terminal s_i in $\mathcal{G} - F$ and $L(v) = 4$ if the node v is reachable from none of the three terminals in $\mathcal{G} - F$. Since F is a non-opposite cut-set, it follows that ℓ is a non-opposite cut. Moreover, $\delta(L) \subseteq F$. Therefore, the claim follows by Lemma 4.3.

Proof of Lemma 4.5: We now restate and prove Lemma 4.5.

Lemma 4.5. *Let $Q : \Delta_{3,n} \to [4]$ be a non-opposite cut with αn^2 nodes labeled as 4. If Q is a non-fragmenting cut and $n \geq 10$, then the cost of Q on J is at least $1.2 - \frac{1}{n} + 0.4\alpha$.*

Proof: We first show that the labeling Q may be assumed to indicate reachability in the graph $\mathcal{G} - \delta(Q)$.

Claim 4.4. *For every non-opposite non-fragmenting cut $Q : \Delta_{3,n} \to [4]$, there exists a labeling $Q' : \Delta_{3,n} \to [4]$ such that*

(i) *a node $v \in \Delta_{3,n}$ is reachable from s_i in $\mathcal{G} - \delta(Q')$ iff $Q'(v) = i$,*

(ii) $Cost_J(\delta(Q')) \le Cost_J(\delta(Q))$,

(iii) *the number of nodes in $\Delta_{3,n}$ that are labeled as 4 by Q is at most the number of nodes in $\Delta_{3,n}$ that are labeled as 4 by Q', and*

(iv) *Q' is a non-opposite non-fragmenting cut.*

Proof:

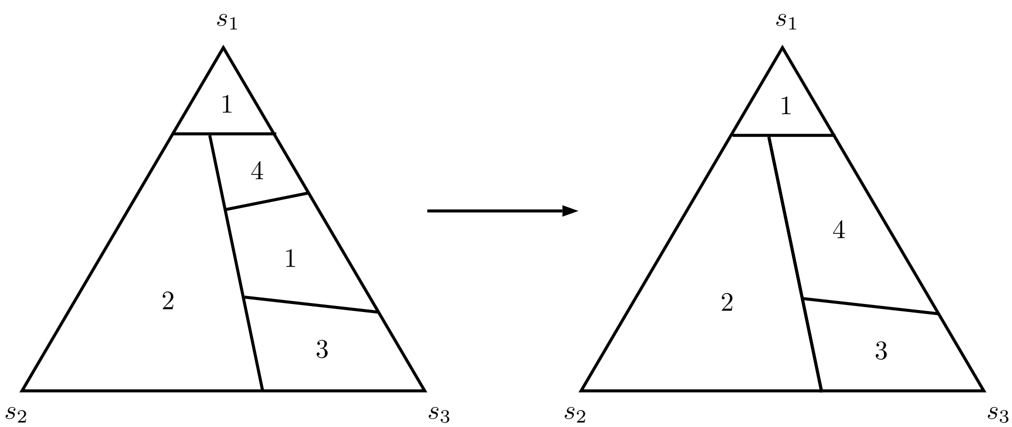


Figure 4.5: An example of a cut Q and the cut Q' obtained in the proof of Claim 4.4.

For $i \in [3]$, let S_i be the set of nodes that can be reached from s_i in $\mathcal{G} - \delta(Q)$. Consider a labeling Q' defined by

$$Q'(v) := \begin{cases} i & \text{if } v \in S_i, \text{ and} \\ 4 & \text{if } v \in \Delta_{3,n} \setminus (S_1 \cup S_2 \cup S_3). \end{cases} \tag{4.62}$$

See Figure 4.5 for an example of a cut Q and the cut Q' obtained as above. We prove the required properties for the labeling Q' below.

(i) By definition, $Q'(v) = i$ iff v is reachable from s_i in $\mathcal{G} \setminus \delta(Q')$.

(ii) Since $\delta(Q') = (\delta(S_1) \cup \delta(S_2) \cup \delta(S_3)) \cap \delta(Q)$, we have that $\delta(Q') \subseteq \delta(Q)$. Hence, $Cost_J(\delta(Q')) \le Cost_J(\delta(Q))$.

(iii) Let $i \in [3]$. Since all nodes of S_i are labeled as i by Q, the nodes labeled as i by Q' is a subset of the set of nodes labeled as i by Q. This implies that Q' is also a non-opposite cut and that the number of nodes in $\Delta_{3,n}$ that are labeled as 4 by Q is at most the number of nodes in $\Delta_{3,n}$ that are labeled as 4 by Q'.

(iv) Since Q is a non-fragmenting cut, there exist distinct $i, j \in [3]$ such that $|\delta(Q) \cap L_{ij}| = 1$. Since $\delta(Q') \subseteq \delta(Q)$, we have that $|\delta(Q') \cap L_{ij}| \leq 1$. On the other hand, Q' labels s_i by i and s_j by j and hence, $|\delta(Q') \cap L_{ij}| \geq 1$. Combining the two, we have that $|\delta(Q') \cap L_{ij}| = 1$ and hence Q' is a non-fragmenting cut.

Let $\mathcal{G}' := \mathcal{G} - \delta(Q)$. By Claim 4.4, we may henceforth assume that

$$\text{For every node } v \in V,\ v \text{ is reachable from } s_i \text{ in } G' \text{ iff } Q(v) = i. \tag{4.63}$$

In order to show a lower bound on the cost of Q, we will modify Q to obtain a non-opposite cut while reducing its cost by 0.4α. [3] showed that the cost of every non-opposite cut on J is at least $1.2 - \frac{1}{n}$. Therefore, the cost of Q on J must be at least $1.2 - \frac{1}{n} + 0.4\alpha$.

Since Q is a non-fragmenting cut, there exist distinct $i, j \in [3]$ such that $|\delta(Q) \cap L_{ij}| = 1$. Without loss of generality, suppose that $i = 1$ and $j = 3$. For $i \in [3]$, let $S_i := \{v \in \Delta_{3,n} \mid Q(v) = i\}$, i.e. S_i is the set of nodes that can be reached from s_i in $\mathcal{G}'$. Let $B := \{v \in \Delta_{3,n} \mid Q(v) = 4\}$ be the set of nodes labeled as 4 by Q. Then, $|B| = \alpha n^2$. We note that S_1, S_2, and S_3 are components of G', and the set B is the union of the remaining components.

Let V_{ij} be the set of end nodes of edges in L_{ij}. We say that a node $v \in \Delta_{3,n}$ can reach V_{ij} in G' if there exists a path from v to some node $w \in V_{ij}$ in G'. We observe that all nodes in V_{13} are reachable from either s_1 or s_3 in G'. In particular, this means that no node of B can reach V_{13} in G'. We partition the node set B based on reachability as follows (see Figure 4.6):

$$B_1 := \{v \in B \mid \ v \text{ cannot reach } V_{12} \text{ and } V_{23} \text{ in } G'\}, \tag{4.64}$$
$$B_2 := \{v \in B \mid \ v \text{ can reach } V_{12} \text{ but not } V_{23} \text{ in } G'\}, \tag{4.65}$$
$$B_3 := \{v \in B \mid \ v \text{ can reach } V_{23} \text{ but not } V_{12} \text{ in } G'\}, \text{ and} \tag{4.66}$$
$$B_4 := \{v \in B \mid v \text{ can reach } V_{12} \text{ and } V_{23} \text{ in } G'\}. \tag{4.67}$$

For $r \in [4]$, let $\beta_r := |B_r|/n^2$. We next summarize the properties of the sets defined above.

Proposition 4.2. *The sets B_1, B_2, B_3, B_4 defined above satisfy the following properties:*

(i) For every distinct $r, p \in [4]$, we have $B_r \cap B_p = \emptyset$.

(ii) For every $r \in [4]$, we have $\delta(B_r) \subseteq \delta(Q)$, i.e. B_r is the union of some components of G'.

(iii) For every $r \in [4]$ and every edge $e \in \delta(B_r)$, one end node of e is in B_r and the other one is in $S_1 \cup S_2 \cup S_3$.

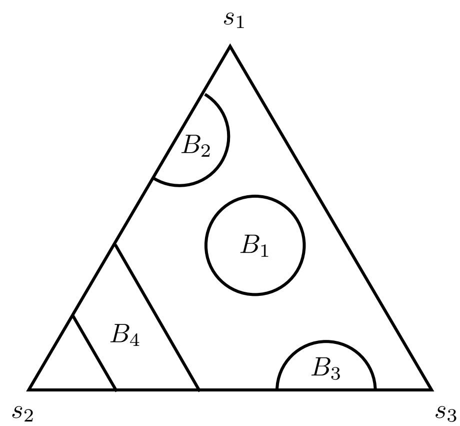


Figure 4.6: Partition of B into B_1, B_2, B_3, B_4.

(iv) For every distinct $r, p \in [4]$, we have $\delta(B_r) \cap \delta(B_p) = \emptyset$.

(v) $B = \cup_{r=1}^{4} B_r$, $\sum_{r=1}^{4} \beta_r = \alpha$, and $\beta_r \le 0.66$ for every $r \in [4]$.

Proof:

(i) The disjointness property follows from the definition of the sets.

(ii) Suppose $\delta(B_r)$ is not a subset of $\delta(Q)$ for some $r \in [4]$. Without loss of generality, let $r = 1$ (the proof for the other cases are similar). Then, there exists an edge $uv \in E_{3,n} \setminus \delta(Q)$ with $u \in B_1, v \in B \setminus B_1$. Since v is in $B \setminus B_1$, it follows that the node v can reach either V_{12} or V_{13} in G'. Moreover, since the edge uv is in $\mathcal{G}'$, it follows that the node u can also reach either V_{12} or V_{13} in G', and hence $u \notin B_1$. This contradicts the assumption that $u \in B_1$.

(iii) Let $uv \in \delta(B_r)$ with $u \in B_r$ and $v \notin B_r$. Since $Q(u) = 4$, the node u is not reachable from any of the terminals in G'. Suppose that the node v is also not reachable from any of the terminals in G'. Then, by the reachability assumption, it follows that $Q(v) = 4$. Hence, the edge uv has both end-nodes labeled as 4 by Q and therefore $uv \notin \delta(Q)$. Thus, we have an edge $uv \in \delta(B_i) \setminus \delta(Q)$ contradicting part (ii).

(iv) Follows from parts (i) and (iii).

(v) By definition, we have that $B = \cup_{r=1}^{4} B_r$. Since the sets B_1, B_2, B_3, B_4 are pair-wise disjoint, they induce a partition of B and hence $|B| = \sum_{r=1}^{4} |B_r|$. Consequently, $\sum_{r=1}^{4} \beta_r n^2 = \alpha n^2$ and thus, $\sum_{r=1}^{4} \beta_r = \alpha$. Next, we note that $|\Delta_{3,n}| = (n+1)(n+2)/2$. Since $B_r \subseteq B \subseteq \Delta_{3,n}$, we have that $\beta_r = |B_r|/n^2 \le |\Delta_{3,n}|/n^2 \le (1 + 1/n)(1 + 2/n)/2 \le 0.66$ since $n \ge 10$.

By Proposition 4.1 (i), the cut-set $\delta(Q)$ is a non-opposite cut-set. The following claim shows a way to modify $\delta(Q)$ to obtain a non-opposite cut-set with strictly smaller cost if $\beta_r > 0$.

Claim 4.5. *For every $r \in [4]$, there exists $E_r \subseteq \delta(B_r), E'_r \subseteq \mathcal{G}[B_r]$ such that*

1. $E_r \subseteq \delta(S_i)$ *for some* $i \in [3]$,

2. $(\delta(Q) \setminus E_r) \cup E'_r$ *is a non-opposite cut-set and*

3. $Cost_J(E_r) - Cost_J(E'_r) \geq 0.4\beta_r$.

Proof: We consider the cases $r = 1, 2, 4$ individually as the proofs are different for each of them. The case of $r = 3$ is similar to the case of $r = 2$. We begin with a few notations that will be used in the proof. For distinct $i, j \in [3]$, and for $t \in \{0, 1, \ldots, 2n/3\}$, let $V^t_{ij} := \{u \in \Delta_{3,n} : u_k = 1 - t/n \text{ for } \{k\} = [3] \setminus \{i, j\}\}$. Thus, V^t_{ij} denotes the set of nodes that are on the line parallel to V_{ij} and at distance t/n from it. We will call the sets V^t_{ij} as lines for convenience. Let L^t_{ij} denote the edges of $E_{3,n}$ whose end-nodes are in V^t_{ij}. Thus, the edges in L^t_{ij} are parallel to L_{ij} (see Figure 4.7).

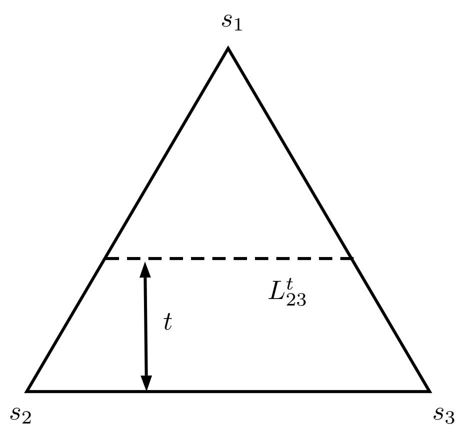


Figure 4.7: The set of edges L^t_{23}.

1. **Suppose** $r = 1$. We partition the set $\delta(B_1)$ of edges into three sets $X_i := \delta(B_1) \cap \delta(S_i)$ for $i \in [3]$ (see Figure 4.8).

   By Proposition 4.2 (iii), we have that (X_1, X_2, X_3) is a partition of B_1. Let

$$E_1 := \arg\max\{Cost_J(F) : F \in \{X_1, X_2, X_3\}\} \text{ and} \tag{4.68}$$

$$E'_1 := \emptyset. \tag{4.69}$$

   We now show the required properties for this choice of E_1 and E'_1.

   (a) Since $E'_1 = \emptyset$, we need to show that $\delta(Q) \setminus E_1$ is a non-opposite cut-set. Let $\mathcal{G}'' := \mathcal{G} - (\delta(Q) \setminus E_1)$. For each edge $e \in E_1$, the end node of e in $\Delta_{3,n} \setminus B_1$ is reachable from a terminal s_i in G' iff it is reachable from s_i in $\mathcal{G}''$. Therefore, for each node $v \in \Delta_{3,n} \setminus B_1$ and a terminal s_i for $i \in [3]$, we have that v is reachable from s_i in $\mathcal{G}'$ iff v is reachable from s_i in $\mathcal{G}''$. Since $\delta(Q)$ is a non-opposite

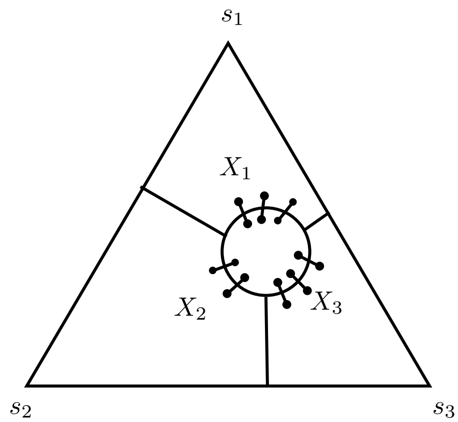


Figure 4.8: Partition of $\delta(B_1)$ into X_i's.

cut-set, it follows that s_i cannot reach V_{jk} in $\mathcal{G}'$ for $\{i,j,k\} = [3]$. Since $B_1 \cap (V_{12} \cup V_{23} \cup V_{13}) = \emptyset$, the terminal s_i cannot reach V_{jk} in $\mathcal{G}''$ for $\{i,j,k\} = [3]$. Hence, $\delta(Q) \setminus E_1$ is a non-opposite cut-set.

(b) We note that none of the nodes in B_1 can reach V_{12}, V_{23} and V_{13} in G'. Therefore, if there exists a node from B_1 in V_{ij}^t for some $t \in \{1,\ldots,n\}$, then at least two edges in L_{ij}^t should be in $\delta(B_1)$ (see Figure 4.9). Therefore, if $V_{ij}^t \cap B_1 \neq \emptyset$, then $|\delta(B_1) \cap L_{ij}^t| \geq 2$.

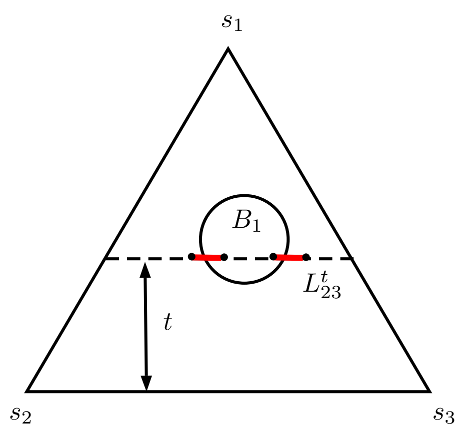


Figure 4.9: $B_1 \cap V_{23}^t \neq \emptyset$ implies that $|\delta(B_1) \cap L_{23}^t| \geq 2$.

Every node $v \in B_1$ is in at least two lines among V_{ij}^t for distinct $i,j \in [3]$ and $t \in \{1,\ldots,2n/3\}$. Each line V_{ij}^t for $t \in \{1,\ldots,2n/3\}$ has at most n nodes. Hence, the number of lines with non-empty intersection with B_1 is at least $2|B_1|/n$. For each line that has a non-empty intersection with B_1, we have at least two edges in $\delta(B_1)$. Hence,

$$\left|\delta(B_1) \cap \left(\cup_{i,j\in[3],t\in\{1,\ldots,2n/3\}} L_{ij}^t\right)\right| \geq 4 \cdot \frac{|B_1|}{n}. \tag{4.70}$$

The cost of each edge in $\cup_{i,j\in[3],t\in\{1,\ldots,2n/3\}} L_{ij}^t$ is $3/5n$. So,

$$Cost_J(\delta(B_1)) \geq Cost_J\left(\delta(B_1)\cap\left(\cup_{i,j\in[3],t\in\{1,\ldots,2n/3\}} L_{ij}^t\right)\right) \geq \frac{12}{5}\frac{|B_1|}{n^2} = \frac{12}{5}\beta_1. \tag{4.71}$$

Since we set E_1 to be the X_i with maximum cost, we get that $Cost_J(E_1) \geq (4/5)\beta_1$. Moreover, $Cost_J(E_1') = 0$ as $E_1 = \emptyset$. Hence, $Cost_J(E_1) - Cost_J(E_1') \geq (4/5)\beta_1 \geq 0.4\beta_1$.

2. **Suppose** $r = 2$. We assume that $B_2 \neq \emptyset$ as otherwise, the claim is trivial. Similar to the previous case, we partition the set $\delta(B_2)$ into three sets $X_i := \delta(B_2)\cap\delta(S_i)$ for $i \in [3]$ (see Figure 4.10).

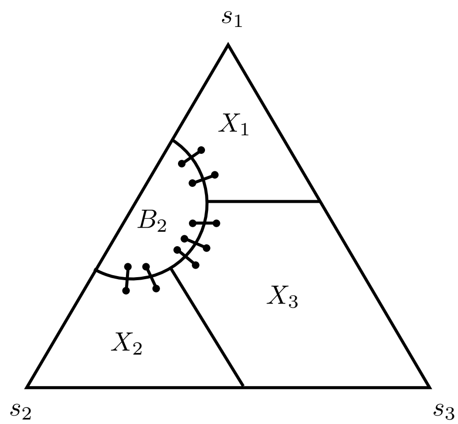


Figure 4.10: Partition of $\delta(B_2)$ into X_i's.

We also define

$$Z := X_3 \cap \delta(B_2 \cap V_{12}) \tag{4.72}$$

and let

$$E_2 := X_1 \text{ and } E_2' := \emptyset \text{ if } Cost_J(X_1) \geq 0.4\beta_2, \tag{4.73}$$
$$E_2 := X_2 \text{ and } E_2' := \emptyset \text{ if } Cost_J(X_2) \geq 0.4\beta_2, \tag{4.74}$$
$$E_2 := X_3 \setminus Z \text{ and } E_2' := \delta_{\mathcal{G}}(B_2 \setminus V_{12}, B_2 \cap V_{12}) \text{ if } Cost_J(X_1), Cost_J(X_2) < 0.4\beta_2. \tag{4.75}$$

We emphasize that the last case is the only situation where we use a non-empty set for E_2'. We now show the required properties for this choice of E_2 and E_2'.

(a) Let $\mathcal{G}'' := \mathcal{G} - ((\delta(Q)\setminus E_2)\cup E_2')$. For each edge $e \in E_2$, the end node of e in $\Delta_{3,n}\setminus B_2$ is reachable from a terminal s_i in $\mathcal{G}'$ iff it is reachable from s_i in $\mathcal{G}''$. Therefore, for each node $v \in \Delta_{3,n}\setminus B_2$ and a terminal s_i for $i \in [3]$, we have that v is reachable from s_i in $\mathcal{G}'$ iff v is reachable from s_i in $\mathcal{G}''$. Since $B_2 \cap V_{13} = \emptyset$ and s_2 cannot reach V_{13} in $\mathcal{G}'$, we have that s_2 cannot reach V_{13} in $\mathcal{G}''$. Similarly, s_1 cannot reach V_{23} in $\mathcal{G}''$. It remains to argue that s_3 cannot reach V_{12} in $\mathcal{G}''$. We

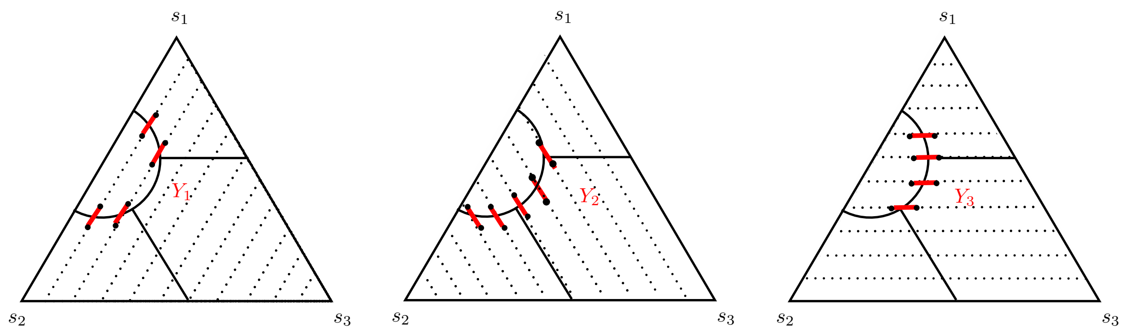


Figure 4.11: Partition of $\delta(B_2)$ into Y_i's. The shaded region is B_2.

have two cases.

i. Suppose $E_2 = X_1$ or $E_2 = X_2$. We note that X_1 and X_2 are the set of edges in $\delta(B_2)$ whose end nodes outside B_2 are reachable from s_1 (and s_2 respectively) in $\mathcal{G}'$. So, if $E_2 = X_1$ or if $E_2 = X_2$, then the set of nodes reachable by s_3 in $\mathcal{G}'$ and $\mathcal{G}''$ remains the same. Since s_3 cannot reach V_{12} in $\mathcal{G}'$, we have that s_3 cannot reach V_{12} in $\mathcal{G}''$.

ii. Suppose $E_2 = X_3$. We will show that $\delta(B_2 \cap V_{12}) \subseteq (\delta(Q) \setminus E_2) \cup E_2'$. Consequently, the nodes of $B_2 \cap V_{12}$ are not reachable from s_3 in $\mathcal{G}''$. Since nodes of $V_{12} \setminus B_2$ are not reachable from s_3 in $\mathcal{G}'$, we have that s_3 cannot reach V_{12}.
We now show that $\delta(B_2 \cap V_{12}) \subseteq (\delta(Q) \setminus E_2) \cup E_2'$. Let $uv \in \delta(B_2 \cap V_{12})$ with $u \in B_2 \cap V_{12}$ and $v \notin B_2 \cap V_{12}$. If $v \in S_1 \cup S_2$, then $uv \in \delta(B_2) \subseteq \delta(Q)$ and $uv \notin X_3 \supseteq E_2$. Hence, $uv \in (\delta(Q) \setminus E_2) \cup E_2'$. If $v \in S_3$, then $uv \in Z$ and hence $uv \notin E_2$. Moreover, $uv \in \delta(B_2) \subseteq \delta(Q)$, hence $uv \in (\delta(Q) \setminus E_2) \cup E_2'$. If $v \in B$, then $v \in B_2$ by Proposition 4.2 (iv) and hence $e \in E_2' \subseteq (\delta(Q) \setminus E_2) \cup E_2'$.

(b) If $Cost_J(X_1)$ or $Cost_J(X_2)$ is at least $0.4\beta_2$, then we are done. So, let us assume that $Cost_J(X_1)$, $Cost_J(X_2) \leq 0.4\beta_2$. Let Y_1, Y_2 and Y_3 be the set of edges in $\delta(B_2)$ that are parallel to L_{12}, L_{13} and L_{23} respectively (see Figure 4.11). Formally,

$$Y_1 := \delta(B_2) \cap \left(\cup_{t \in \{0,1,\ldots,n\}} L_{12}^t\right) \tag{4.76}$$
$$Y_2 := \delta(B_2) \cap \left(\cup_{t \in \{0,1,\ldots,n\}} L_{13}^t\right) \tag{4.77}$$
$$Y_3 := \delta(B_2) \cap \left(\cup_{t \in \{0,1,\ldots,n\}} L_{23}^t\right) \tag{4.78}$$

Claims 4.6 and 4.7 will help us derive the required inequality on the cost.

Claim 4.6.

$$Cost_J(E_2') \leq Cost_J(Y_2) + Cost_J(Y_3) - Cost_J(Z). \tag{4.79}$$

Proof: We proceed in two steps: (1) we will show a one-to-one mapping f from edges in E_2' to edges in $(Y_2 \cup Y_3) \setminus Z$ such that the cost of every edge $e \in E_2'$ is the same as the cost of the mapped edge $f(e)$ in the instance J, i.e., $w(e) = w(f(e))$ for every $e \in E_2'$ and (2) we will show that that $Z \subseteq Y_2 \cup Y_3$. Now, by observing that the sets Y_2 and Y_3 are disjoint, we get that $Cost_J(E_2') \leq Cost_J(Y_2) + Cost_J(Y_3) - Cost_J(Z)$.

We now define the one-to-one mapping $f : E_2' \rightarrow (Y_2 \cup Y_3) \setminus Z$. Let $e = uv \in E_2'$ such that $u \in B_2 \cap V_{12}, v \in B_2 \setminus V_{12}$. Since E_2' only contains edges between $B_2 \cap V_{12}$ and $B_2 \setminus V_{12}$, it does not contain an edge parallel to L_{12}. Therefore, $e \in L_{13}^t$ or $e \in L_{23}^t$ for some $t \in \{1, \ldots, n\}$. Suppose $e \in L_{13}^t$ for some $t \in \{1, \ldots, n\}$. Since the nodes of B_2 cannot reach V_{23} in $\mathcal{G}'$, there exists an edge in $\delta(B_2) \cap L_{13}^t$. We map e to an arbitrary edge in $\delta(B_2) \cap L_{13}^t \subseteq Y_2$ (see Figure 4.12). We note that the set Z contains the set of edges incident to $B_2 \cap V_{12}$ whose other end node is in S_3. Since both u and v are in B_2, it follows that $L_{13}^t \cap Z = \emptyset$. So, our mapping of e is indeed to an edge in $Y_2 \setminus Z$. Similarly, if $e \in L_{23}^t$ for some $t \in \{1, \ldots, n\}$, then we map e to an arbitrary edge in $\delta(B_2) \cap L_{23}^t \subseteq Y_3 \setminus Z$. This mapping is a one-to-one mapping as E_2' contains at most one edge from L_{13}^t for each $t \in \{1, 2, \ldots, n\}$ and at most one edge from L_{23}^t for each $t \in \{1, 2, \ldots, n\}$. Moreover, for each $t \in \{1, 2, \ldots, n\}$, the cost of all edges in L_{13}^t are identical and the cost of all edges in L_{23}^t are identical.

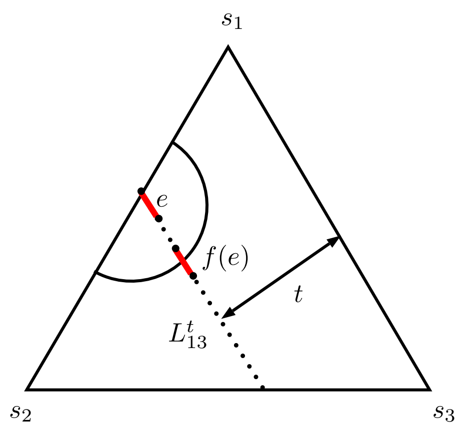


Figure 4.12: Mapping from E_2' to $(Y_2 \cup Y_3) \setminus Z$. The shaded region is B_2.

We now show that $Z \subseteq Y_2 \cup Y_3$. The set Z contains all edges whose one end node is in $B_2 \cap V_{12}$ and another end node is in S_3. Since $V_{12} \cap S_3 = \emptyset$, the set Z does not contain any edge between $B_2 \cap V_{12}$ and $V_{12} \setminus B_2$. Hence, $Y_1 \cap Z = \emptyset$. Since Z is a subset of X_3 which is a subset of $Y_1 \cup Y_2 \cup Y_3$, it follows that $Z \subseteq Y_2 \cup Y_3$.

Claim 4.7.

$$Cost_J(Y_1) \geq \frac{6}{5}\beta_2. \tag{4.80}$$

Proof: We first show a lower bound on the size of the set $W := \{t \in \{0, 1, \ldots, 2n/3\} : V_{12}^t \cap B_2 \neq \emptyset\}$. If $B_2 \cap V_{12}^t \neq \emptyset$ for some $t \in \{2n/3 + 1, \ldots, n\}$, then $B_2 \cap V_{12}^t \neq \emptyset$ for all $t \in \{0, 1, \ldots, 2n/3\}$ and hence, $|W| \geq 2n/3$. Otherwise, $B_2 \cap V_{12}^t = \emptyset$ for all $t \in \{2n/3 + 1, \ldots, n\}$. In this case,

$B_2 \subseteq \cup_{t=0}^{2n/3} V_{12}^t$. For $t \geq 1$, each line V_{12}^t has at most n nodes. For $t = 0$, the set B_2 can contain at most $n-1$ nodes from V_{12}^0 which are not s_1 or s_2. Hence, $|W| \geq |B_2|/n = \beta_2 n$. Thus, we have that $|W| \geq \min\{2n/3, \beta_2 n\} = \beta_2 n$ as $\beta_2 \leq 0.66$.

Since the nodes of B_2 cannot reach V_{23} and V_{13} in $\mathcal{G}'$, we have that $|\delta(B_2) \cap L_{12}^t| \geq 2$ if $B_2 \cap V_{12}^t \neq \emptyset$. Hence,

$$\left|\delta(B_2) \cap \left(\cup_{t=0}^{2n/3} L_{12}^t\right)\right| \geq 2|W| \geq 2\beta_2 n. \tag{4.81}$$

Each edge in $\cup_{t=0}^{2n/3} L_{12}^t$ has cost at least $3/5n$. Hence,

$$Cost_J\left(\delta(B_2) \cap \left(\cup_{t=0}^{2n/3} L_{12}^t\right)\right) \geq \frac{6}{5}\beta_2. \tag{4.82}$$

Since $Y_1 = \delta(B_2) \cap \left(\cup_{t=0}^{n} L_{12}^t\right) \supseteq \delta(B_2) \cap \left(\cup_{t=0}^{2n/3} L_{12}^t\right)$, we get that $Cost_J(Y_1) \geq (6/5)\beta_2$.

We now derive the required inequality on the cost as follows:

$$\begin{aligned}
Cost_J(E_2) - Cost_J(E_2') &= Cost_J(X_3 \setminus Z) - Cost(E_2') && (4.83)\\
&\geq Cost_J(X_3) - Cost_J(Z) - Cost_J(Y_2) - Cost_J(Y_3) + Cost_J(Z) \quad \text{(By Claim 4.6)} && (4.84)\\
&\geq Cost_J(X_3 \cap Y_1) + Cost_J(X_3 \cap Y_2) + Cost_J(X_3 \cap Y_3) - Cost_J(Y_2) - Cost_J(Y_3) && (4.85)\\
&= Cost_J(X_3 \cap Y_1) - Cost_J((X_1 \cup X_2) \cap Y_2) - Cost_J((X_1 \cup X_2) \cap Y_3) && (4.86)\\
&= Cost_J(Y_1) - Cost_J((X_1 \cup X_2) \cap Y_1) - Cost_J((X_1 \cup X_2) \cap Y_2) - Cost_J((X_1 \cup X_2) \cap Y_3) && (4.87)\\
&= Cost_J(Y_1) - Cost_J(X_1 \cup X_2) && (4.88)\\
&\geq \frac{6}{5}\beta_2 - 0.4\beta_2 - 0.4\beta_2 \quad \text{(By Claim 4.7 and } Cost_J(X_1), Cost_J(X_2) \leq 0.4\beta_2\text{)} && (4.89)\\
&= 0.4\beta_2. && (4.90)
\end{aligned}$$

3. **Suppose** $r = 4$. We assume that $B_4 \neq \emptyset$, as otherwise the claim is trivial. We partition $\delta(B_4)$ into $X_1 := \delta(B_4) \cap \delta(S_2)$ and $X_2 := \delta(B_4) \setminus X_1$ (see Figure 4.13), and let $E_4 := X_1$ and $E_4' := \emptyset$.

   We now show the required properties for this choice of E_4 and E_4'. Let us fix a node $v \in B_4$ and a path $v, u_1, \ldots, u_t$ from v to L_{12} in $\mathcal{G}[B_4]$, and a path $v, w_1, \ldots, w_{t'}$ from v to L_{23} in $\mathcal{G}[B_4]$ (see Figure 4.13). Let $S := \{v, u_1, \ldots, u_t, w_1, \ldots, w_{t'}\}$. We note that $S \subseteq B_4$.

   (a) Since $E_4' = \emptyset$, we need to show that $\delta(Q) \setminus E_4$ is a non-opposite cut-set. Let $\mathcal{G}'' := \mathcal{G} - (\delta(Q) \setminus E_4)$. We first observe that there are no paths between S and V_{13} in $\mathcal{G} - X_2$. Hence, there is no path from s_1 or s_3 to an end node of $E_4 = X_1$ in $\mathcal{G}'$. Moreover, there is no path from s_1 to V_{23} or from s_3 to V_{12} in $\mathcal{G}'$. So, there is no path from s_1 to V_{23} or from s_3 to V_{12} in $\mathcal{G}''$. Also, since $X_2 \subseteq \delta(Q) \setminus X_1$ and there is no path from s_2 to V_{13} in $\mathcal{G} - X_2$, it follows that there is no path from s_2 to V_{13} in $\mathcal{G}''$. Hence, $\delta(Q) \setminus E_4$ is a non-opposite cut-set.

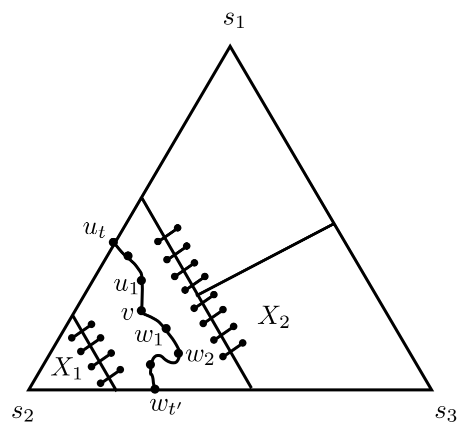


Figure 4.13: Partition of $\delta(B_4)$ into X_1 and X_2. The shaded region is B_4.

(b) We note that there are no paths between s_2 and S in $\mathcal{G} - E_4$. Moreover, all paths in $\mathcal{G}$ between s_2 and V_{13} go through S. Hence, there are no paths between s_2 and V_{13} in $\mathcal{G} - E_4$. The cost of any such subset of nodes can be lower bounded by the Lemma 4.4. Thus, $Cost_J(E_4) - Cost_J(E'_4) \geq 0.4 - (\frac{1}{n})/3 \geq 0.4\beta_4$. The last inequality is because $\beta_4 \leq 0.66$ by Proposition 4.2 and $n \geq 10$.

For $r \in [4]$, let E_r and E'_r be the sets given by Claim 4.5. We will show that

$$F := \left(\delta(Q) \setminus \left(\cup_{r=1}^{4} E_r\right)\right) \cup \left(\cup_{r=1}^{4} E'_r\right) \tag{4.91}$$

is a non-opposite cut-set and that $Cost_J(\delta(Q)) \geq Cost_J(F) + 0.4\alpha$. Then, we use Proposition 4.1 (ii) to conclude that $Cost_J(\delta(Q)) \geq 1.2 - \frac{1}{n} + 0.4\alpha$.

Claim 4.8. *F is a non-opposite cut-set.*

Proof: Let $\mathcal{G}'' = \mathcal{G} - F$, and for $i \in [3]$, let S'_i be the set of nodes reachable from s_i in $\mathcal{G}''$. Since $E'_r \subseteq \mathcal{G}[B]$ for every r, S'_i is a superset of S_i, and $\mathcal{G}''[S_i] = G'[S_i]$, which is connected. By the first property of Claim 4.5, for every $r \in [4]$ there exists $i \in [3]$ such that $E_r \subseteq \delta(B_r) \cap \delta(S_i)$. This implies, together with Proposition 4.2 (ii), that the sets S'_i are disjoint. It also implies the following property:

($\star$) For every $r \in [4]$, there exists $i \in [3]$ such that $\delta_{\mathcal{G}''}(B_r) \subseteq \delta(S_i)$.

Suppose for contradiction that for some distinct $i, j, k \in [3]$, there exists a path P in $\mathcal{G}''$ from s_i to some $v \in V_{jk}$. Since $\delta(Q)$ is a non-opposite cut, the node v is not in S_i. Also, since $v \in S'_i$ and we have seen above that S'_i is disjoint from S'_j and S'_k, it follows that $v \notin S'_j \cup S'_k \supseteq S_j \cup S_k$. Hence, $v \notin S_1 \cup S_2 \cup S_3$, and therefore $v \in B_r$ for some $r \in [4]$.

Let u be the last node of S_i on the path P. By property ($\star$), the end segment of P starting at the node after u is entirely in $\mathcal{G}[B_r] \setminus E'_r$. Since $\mathcal{G}''[S_i]$ is connected, we can replace the $s_i - u$ part of P by a path in $\mathcal{G}''[S_i]$, and obtain an $s_i - v$ path in $\mathcal{G}''$ that uses only edges in $G'[S_i] \cup (G'[B_r] \setminus E'_r)$ and a single edge in $E_r \subseteq \delta(S_i) \cap \delta(B_r)$. Hence, this is also a path in $E \setminus ((\delta(Q) \setminus E_r) \cup E'_r)$. But we have already seen in Claim 4.5 that $(\delta(Q) \setminus E_r) \cup E'_r$ is a non-opposite cut-set, so $v \notin V_{jk}$, a contradiction.

To show that $Cost_J(\delta(Q)) \geq Cost_J(F) + 0.4\alpha$, we first observe that $E_i \subset \delta(B_i) \subset \delta(Q)$ for $i \in [4]$ and E_i's are mutually disjoint since $\delta(B_i)$'s are mutually disjoint by Proposition 4.2 (iv). Therefore,

$$
\begin{aligned}
Cost_J(F) &\leq Cost_J(\delta(Q) \setminus (\cup_{i=1}^4 E_i)) + Cost_J(\cup_{i=1}^4 E'_i) && (4.92)\\
&= Cost_J(\delta(Q)) - \sum_{i=1}^{4}(Cost_J(E_i) - Cost_J(E'_i)) && (4.93)\\
&\leq Cost_J(\delta(Q)) - \sum_{i=1}^{4} 0.4\beta_i \qquad \text{(By Claim 4.5)} && (4.94)\\
&\leq Cost_J(\delta(Q)) - 0.4\alpha \qquad \text{(By Proposition 4.2 (v))}. && (4.95)
\end{aligned}
$$

Proof of Corollary 4.5: We restate and prove Corollary 4.5 now.

Corollary 4.5. *Let $Q : \Delta_{3,n} \to [4]$ be a non-opposite cut and $n \geq 10$. For each $i \in [3]$, let*

$$
A_i := \begin{cases} \{v \in Closure(R_i) : Q(v) = 4\} & \text{if } \delta(Q) \cap \Gamma_i = \emptyset, \\ \emptyset & \text{otherwise}. \end{cases} \tag{4.96}
$$

Then, the cost of Q on J is at least $1.2 - \frac{2}{n} + 0.4\sum_{i=1}^{3} |A_i|/n^2$.

Proof:[Corollary 4.5] Let $A := A_1 \cup A_2 \cup A_3$. We will show that $Cost_J(\delta(A))$ is at least $0.4\sum_{i=1}^{3} |A_i|/n^2 - 1/n$ and that there exists a non-opposite non-corner cut Q' satisfying $\delta(Q') = \delta(Q) \setminus \delta(S)$. By Lemma 4.5, $Cost_J(\delta(Q')) \geq 1.2 - 1/n$ and hence the corollary follows.

We first show a lower bound on the total cost of the edges in $\delta(A)$.

Claim 4.9. $Cost_J(\delta(A)) \geq 0.4\sum_{i=1}^{3} |A_i|/n^2 - \frac{1}{n}$.

Proof: We will consider a specific non-opposite non-corner cut to give a lower bound on the cost of $\delta(A)$ on J. Let Q_0 be defined as follows (see Figure 4.14):

$$
Q_0(x) := \begin{cases} 1 & \text{if } x_1 \geq 1/2, \\ 2 & \text{if } x_1 < 1/2,\ x_2 \geq 1/2, \\ 3 & \text{otherwise}. \end{cases} \tag{4.97}
$$

Then Q_0 is 'tight' in J, i.e., $Cost_J(\delta(Q_0)) \leq 1.2$. Indeed, $Cost_J(\delta(Q_0)) \leq 2(n/2)(1.2/(2n)) = 1.2$ as needed. Moreover, Q_0 is a non-opposite non-corner cut. We now combine $\delta(A)$ and $\delta(Q_0)$ into a single

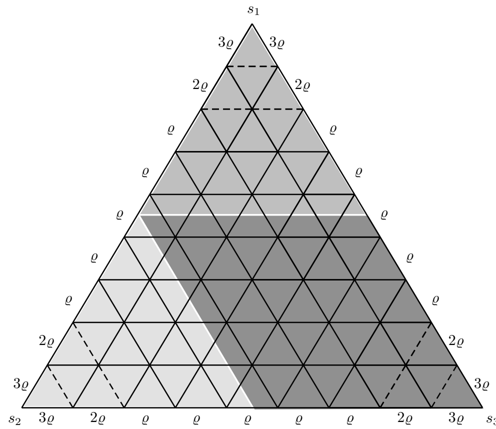


Figure 4.14: The labeling Q_0.

cut by defining

$$Q_0'(x) := \begin{cases} Q_0(x) & \text{if } x \notin A, \\ 4 & \text{otherwise.} \end{cases} \tag{4.98}$$

We observe that Q_0' is a non-opposite cut as it is obtained from a non-opposite cut by relabeling a subset of nodes that lie in the strict interior of Closure(R_i) as 4. As $A_i \neq \emptyset$ implies $\delta(A_i) \cap \Gamma_i = \emptyset$, we have that $\delta(Q_0')$ intersects each side of the triangle the same number of times as $\delta(Q_0)$. That is, Q_0' is also a non-corner cut. Therefore, we can apply Lemma 4.5 for Q_0'. The number of nodes labeled by Q_0' as 4 is exactly equal to $|A|$. Hence,

$$Cost_J(\delta(Q_0')) \geq 1.2 - \frac{1}{n} + 0.4\frac{|A|}{n^2}. \tag{4.99}$$

Since $Q_0(v) = i$ for each $v \in$ Closure(R_i) and by $\delta(A_i) \cap \Gamma_i = \emptyset$ for $i \in [3]$, we have $\delta(Q_0') = \delta(A) \cup \delta(Q_0)$ and hence,

$$Cost_J(\delta(A)) + Cost_J(\delta(Q_0)) \geq Cost_J(\delta(A) \cup \delta(Q_0)) = Cost_J(\delta(Q_0')) \geq 1.2 - \frac{1}{n} + 0.4\frac{|A|}{n^2}. \tag{4.100}$$

Recall that $Cost_J(\delta(Q_0)) \leq 1.2$, implying $Cost_J(\delta(A)) \geq 0.4|A|/n^2 - 1/n$.

Let $K \subseteq [3]$ denote the set of indices i for which $\delta(Q) \cap \Gamma_i = \emptyset$ and let Q' be a labeling obtained from Q by setting

$$Q'(v) := \begin{cases} i & \text{if } v \in \text{Closure}(R_i) \text{ for some } i \in K, \\ Q(v) & \text{otherwise.} \end{cases} \tag{4.101}$$

Claim 4.10. *Q' is a non-opposite cut with $Cost_J(\delta(Q')) \leq Cost_J(\delta(Q)) - Cost_J(\delta(A))$.*

Proof: The cut Q is a non-opposite cut and $Q'(v) \in$ Support(v) for each relabeled node v, hence Q' is also a non-opposite cut. For any index $i \in K$, $\delta(Q) \cap \Gamma_i = \emptyset$ implies $Q(v) = i$ for $v \in R_i$. Thus we have $\delta(Q') \subseteq \delta(Q) \setminus \delta(A)$ and the claim follows.

As Q' is a non-opposite cut, Lemma 4.3 implies $Cost_J(\delta(Q')) \geq 1.2 - 1/n$. By Claim 4.9, $Cost_J(\delta(A)) \geq 0.4 \sum_{i=1}^{3} |A_i|/n^2 - 1/n$. These together with Claim 4.10 imply that $Cost_J(\delta(Q)) \geq 1.2 - 2/n + 0.4 \sum_{i=1}^{3} |A_i|/n^2$, finishing the proof of the corollary.

4.2 Flow-cut gap 2 for $\{s,t\}$-EDGE-BICUT

Proof: [Theorem 4.2] The proof is based on recursively defined sequence of graphs $G_0, G_1, \ldots, G_h$ with increasing integrality gap; we will use α_i to denote the integrality gap (we also refer to this as the flow-cut gap) in G_i. The two terminals will be denoted by s, t. The symmetry in the construction will ensure that in G_i the s-t cut value will be equal to the t-s cut value; we refer to these common values as the one-way cut value and the optimum value of a cut that separates s from t and t from s as the two-way cut value. The graph G_0 is shown in Fig 4.15 and it is easy to see that $\alpha_0 = 1$.

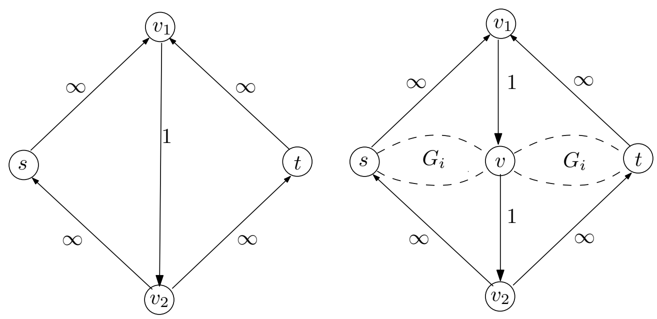


Figure 4.15: G_0 on the left and constructing G_{i+1} from G_i shown on the right.

The iterative construction of G_{i+1} from G_i is shown at a high-level in figure 4.15. A formal description is as follows. To obtain G_{i+1} with terminals s, t we start with two copies of G_i with terminals s_1, t_1 and s_2, t_2 (denoted by H, H') and two new vertices v_1, v_2. We set $s = s_1$, $t = t_2$ and identify t_1 and s_1 as the center vertex v shown in the figure. We add edges (v_1, v) and (v, v_2) with weight 1 and four other edges $\{(s, v_1), (t, v_1), (v_2, s), (v_2, t)\}$ each with weight infinity. Finally, we scale the weights of the edges of H and H' such that the two-way cut value in each of them is $\frac{\alpha_i}{2-\alpha_i}$. It is easy to observe inductively that each graph in the sequence is planar and moreover the graph can be embedded such that s and t are on the outer face.

The following proposition is easy to establish based on the symmetry in the construction of the graphs.

Proposition 4.3. *The s-t cut value and the t-s cut value in G_{i+1} are the same.*

Now, we calculate α_{i+1} in terms of α_i. We refer to the copy of G_i containing s and v with scaled capacities as H, and the one containing v and t as H'.

Lemma 4.6. *For $i \geq 0$, $\alpha_{i+1} = \frac{4-\alpha_i}{3-\alpha_i}$. For $i \geq 0$, the ratio of the one-way cut value to the two-way cut value in G_i is $\frac{1}{\alpha_i}$.*

Proof: Proof by induction on i. For the base case we see that $\alpha_0 = 1$ and in G_0 the one-way cut value and two-way cut value are both 1 and hence the ratio is equal to $1 = \frac{1}{\alpha_0}$.

We now prove the induction step. For this purpose, we estimate the one-way cut value and the two-way cut value in G_{i+1}.

Minimum two-way cut: Any finite value cut that separates s from t has to cut at least one of the two edges $(v_1, v), (v, v_2)$. We consider two cases.

Case 1: Both $(v_1, v), (v, v_2)$ are cut. To separate s and t it is best to pick a two-way cut between s and v in H (or symmetrically between v and t in H'). Thus, the total cost is $2 + \frac{\alpha_i}{2-\alpha_i} = \frac{4-\alpha_i}{2-\alpha_i}$.

Case 2: Only one of the edges $(v_1, v), (v, v_2)$ is cut. Without loss of generality this edge is (v, v_2). Since (v_1, v) is not cut s and t can reach v via v_1. Thus, any two-way cut in G needs to use a one-way cut in H to separate v from s and a one-way cut in H' to separate v from t. The cost of each of these one-way cuts is, by induction, $\frac{1}{\alpha_i} \cdot \frac{\alpha_i}{2-\alpha_i} = \frac{1}{2-\alpha_i}$. Thus, the total cost is $1 + \frac{2}{2-\alpha_i} = \frac{4-\alpha_i}{2-\alpha_i}$.

In both cases the cost is the same and hence the optimal two-way cut in G_{i+1} is $\frac{4-\alpha_i}{2-\alpha_i}$.

Minimum one-way cut: We now calculate one-way cut from s to t. At least one of the edges $(v_1, v), (v, v_2)$ has to be cut. Also, either there is no path from s to v or no path from v to t. Thus, the cost of the one-way cut from s to t is at least $1 + \frac{1}{2-\alpha_i} = \frac{3-\alpha_i}{2-\alpha_i}$. Moreover it is easy to see that this is achievable by removing (v_1, v) and one-way cut from s to v in H.

Optimum fractional solution value: We now calculate the optimum for DIR-MWC-REL on G_{i+1}. We consider the following feasible solution x. Assign 0 to the infinite weight edges and $1/2$ to each of edges (v_1, v) and (v, v_2). For the edges in the graphs H and H' we take an optimum solution y to DIR-MWC-REL on G_i and scale it down by $1/2$ and assign these values to the edges of H and H'. Feasibility of y for G_i implies that distance from s to v and v to s in H according to x is $1/2$ (since we scaled down by $1/2$). It is easy to verify that distance of s to t and from t to s is 1 in the fractional solution x in G_{i+1}. Now we analyze the cost of this solution $\sum_{e \in E(G_{i+1})} w_e x_e$. We have a total contribution of 1 from the two edges (v_1, v) and (v, v_2). We claim that $\sum_{e \in E(H)} w_e x_e = \frac{1}{2} \cdot \frac{1}{\alpha_i} \cdot \frac{\alpha_i}{2-\alpha_i}$ since the cost of the two-way cut in H is chosen to be $\frac{\alpha_i}{2-\alpha_i}$, the integrality gap is α_i and we scaled down y by $1/2$ to obtain x in H. Same holds for H'. Thus, the total fractional cost of this solution is $1 + \frac{1}{2-\alpha_i} = \frac{3-\alpha_i}{2-\alpha_i}$. We can see that this is an optimum solution by exhibiting a multicommodity flow of the same value for the pairs (s, t) and (t, s) in G_{i+1}. Route one unit of flow from s to t along the path $s \to v_1 \to v \to v_2 \to t$. In H there exists a feasible flow of total value $\frac{1}{\alpha_i} \cdot \frac{\alpha_i}{2-\alpha_i} = \frac{1}{2-\alpha_i}$. Let $f(s, v)$ and $f(v, s)$ be the amount of flow from s to v and v to s respectively. By duplicating this flow in H' we see that a flow of value $\frac{1}{2-\alpha_i}$ exists between s and t in G_{i+1} via H and H'. Thus there is a total flow of value at least $1 + \frac{1}{2-\alpha_i}$ in G_{i+1} and this is optimal.

We can now put together the preceding bounds to prove the lemma. The flow-cut gap in G_{i+1} is seen to be the ration of the two-way cut value $\frac{4-\alpha_i}{2-\alpha_i}$ and the maximum flow value $\frac{3-\alpha_i}{2-\alpha_i}$. Hence $\alpha_{i+1} = \frac{4-\alpha_i}{3-\alpha_i}$ as desired. The ratio of one-way cut value $\frac{3-\alpha_i}{2-\alpha_i}$ and the two-way cut value $\frac{4-\alpha_i}{2-\alpha_i}$ in G_{i+1} is $\frac{3-\alpha_i}{4-\alpha_i}$ which is equal to $\frac{1}{\alpha_{i+1}}$. This completes the inductive proof.

We have a sequence of numbers α_i where $\alpha_0 = 1$ and $\alpha_{i+1} = \frac{4-\alpha_i}{3-\alpha_i}$. It is easy to argue that this sequence converges to 2. This proves that the integrality gap of DIR-MWC-REL is in the limit equal to 2.

4.3 Concluding Remarks

Our work opens the possibility of closing the gap between the upper and lower bound known for the integrality gap of CKR-relaxation. We overcame the obstacle in analyzing 3-dimensional instances by considering a convex combination of three 2-dimensional instances and a simple 3-dimensional instance. We proved a lower bound of 1.20016 on the integrality gap which can be further improved by minor modifications in the construction. For example, we put the 2-dimensional instance on a single face of our final isntance. Instead, we can put the 2-dimensional instance on all the faces and improve the gap. However, such modifcations are not sufficient to push the lower bound close the known upper bound of 1.2965. Proving a lower bound close this upper bound would require some new ideas.

Chapter 5

New LP relaxations for Subset Feedback Set problems

In the classical Feedback Vertex Set problem (FVS) the input is a node-weighted graph $G = (V, E)$ and the goal is to find a minimum weight set of nodes whose removal makes the graph acyclic. FVS is interesting for its applications as well as connections to graph theory and combinatorial optimization. In this chapter we restrict our attention to *undirected* graphs. FVS is easily seen to generalize the VERTEX COVER problem and inherits NP-Hardness as well as the hardness of approximation bounds for VERTEX COVER. We could also consider the Feedback Edge Set problem (FES) where the goal is remove a minimum weight set of edges to make it acyclic. FES is polynomial-time solvable; the complement of the edge-set of a maximum weight spanning tree in G can be easily seen to be an optimum solution. FVS and FES can also be viewed as hitting set problems where the goal is to find edges or nodes to intersect all cycles. In this chapter we consider the more general *subset* feedback problems.

SUBSET FEEDBACK VERTEX SET (SUBSET-FVS): Input is an undirected graph $G = (V, E)$ along with non-negative node weights $w(v), v \in V$, and a set $S = \{s_1, \dots, s_k\} \subset V$ of terminals. A cycle is interesting if it contains a terminal. The goal is to find a minimum weight set of nodes $V' \subset V$ that intersect all interesting cycles.

SUBSET FEEDBACK EDGE SET (SUBSET-FES): Input is an undirected graph $G = (V, E)$ along with a non-negative edge weights $w(e), e \in E$, and a set $S = \{s_1, \dots, s_k\} \subset V$ of terminals. A cycle is interesting if it contains a terminal. The goal is to find a minimum weight set of edges $E' \subset E$ that intersect all interesting cycles.

SUBSET-FVS generalizes two well-known NP-complete problems. When $S = V$, we obtain FVS. When $|S| = 1$ it can be shown to be equivalent to the node-weighted Multiway Cut Problem (NODE-WT-MWC). Recall that in NODE-WT-MWC the input consists of a node-weighted graph G and a set of terminals T; the goal is to remove a minimum weight set of nodes such that there is no path left between any two terminals. NODE-WT-MWC can be reduced to SUBSET-FVS by adding a new node s of infinite weight and making it adjacent to each terminal in T. In a similar vein SUBSET-FES generalizes edge-weighted Multiway Cut Problem (EDGE-WT-MWC).

FVS, SUBSET-FVS and SUBSET-FES all admit constant factor approximation algorithms. In particular there is a 2-approximation for FVS [5, 8] and SUBSET-FES [34], and a 8-approximation for SUBSET-FVS [35]. There is a natural LP relaxation for these problems when viewed as a hitting set problem. For instance, consider FVS. The relaxation has a variable $z(v) \in [0, 1]$ for each $v \in V$, and for each cycle C, a constraint $\sum_{v \in C} z(v) \geq 1$. This LP relaxation has an $\Omega(\log n)$ integrality gap [34]. Algorithms for feedback problems in undirected graphs have mainly relied on combinatorial techniques at the high-level. The non-trivial 2-approximation algorithm for FVS from [5] has been later interpreted as a primal-dual algorithm by Chudak *et al.* [26], however, the underlying LP is not known to be solvable in polynomial-time and does not generalize to SUBSET-FES or SUBSET-FVS. The 2-approximation for SUBSET-FES [34] is simple and combinatorial but delicate to analyze. The 8-approximation for SUBSET-FVS [35] is very complicated to describe and analyze; the algorithm is combinatorial at the high-level but solves a sequence of relaxed multicommodity flow LPs to optimality.

In our work, we describe new LP relaxations for SUBSET-FES and SUBSET-FVS and derive constant factor approximations through them. Our results are captured by the following theorem.

Theorem 5.1. *There are polynomial-sized integer programming formulations for* SUBSET-FES *and* SUBSET-FVS *whose linear programming relaxations have an integrality gap of at most* 13.

The approximation bound of 13 that we are able to establish is weaker than the existing approximation ratios for the problems. However, we do not know of an integrality gap worse than 2 for the LP relaxations we propose. We believe that related formulations and ideas would lead to improved algorithms for SUBSET-FES and SUBSET-FVS.

Our formulation and algorithms are simple and are based on a new perspective on the problem. Our analysis uses only elementary arguments but is not as straight forward.

Other Related Work: Figure 5.1 taken from Vazirani's book on approximation [82] shows the relationship of SUBSET-FES and SUBSET-FVS to several well-known problems, some of which we already discussed. A natural open problem here is whether SUBSET-FVS has a 2-approximation.

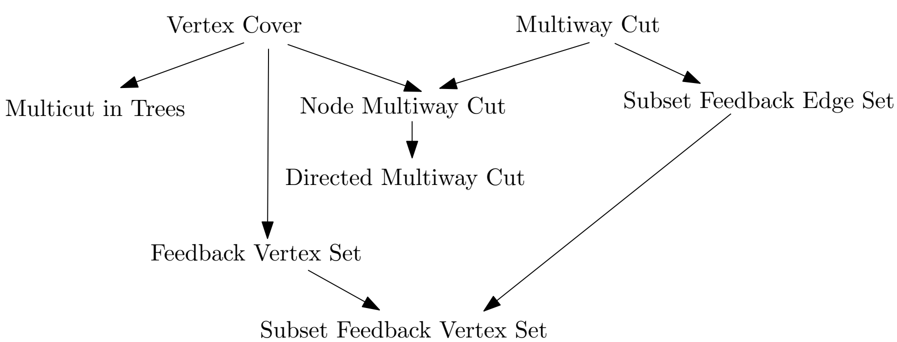


Figure 5.1: Approximation preserving reductions related to feedback problems. Figure is reproduced from [82]. All problems except SUBSET-FVS have a 2-approximation with Multiway Cut admitting an approximation better than 2.

Unweighted FVS is related to the well-known Erdos-Pósa theorem [33] which states that if k is the size of the smallest cardinality feedback vertex set in a graph G then there are $\Omega(k/\log k)$ node disjoint cycles in G. This immediately shows, via duality, that the integrality gap of the standard LP that we discussed in the introduction is $O(\log n)$ for unweighted FVS. The first constant factor approximation for the unweighted FVS problem is due to Bar-Yehuda *et al.* who obtained a 4-approximation [7]; the same paper also obtained a $2\Delta^2$ approximation for weighted FVS where Δ is the maximum degree. SUBSET-FVS also admits a Δ-approximation [34] which can be better than the 8-approximation [35] in some instances. Goemans and Williamson [51] considered FVS, SUBSET-FVS and other related problems in planar graphs and obtained a 9/4 approximation via a primal-dual algorithm with respect to the standard hitting set LP; recall that this LP has an $\Omega(\log n)$ integrality gap in general graphs.

Feedback problems are of much interest in directed graphs as well. FVS and FES are equivalent in directed graphs, and similarly SUBSET-FVS and SUBSET-FES. Leighton and Rao [65] obtained an $O(\log^2 n)$-approximation for FVS in directed graphs using their separator algorithms. Building on Seymour's work [75] that related the fractional packing of cycles to the minimum feedback vertex set[1], Even *et al.* [36] obtained a $O(\log k \log\log k)\}$ approximation for SUBSET-FVS. It is also known that, assuming the Unique Game Conjecture, FVS in directed graphs does not admit a constant factor approximation [53, 81].

Recall the multiway cut problem discussed in Chapter 4. SUBSET-FES and SUBSET-FVS generalize the edge-weighted multiway cut (EDGE-WT-MWC) and node-weighted multiway cut problem (NODE-WT-MWC) respectfully. Labeling based CKR relaxation for EDGE-WT-MWC has been extensively studied (see Chapter 4). Although it is slightly less natural, NODE-WT-MWC can also be viewed as a partition problem indirectly via the hypergraph-cut problem (see [17, 18] and references). Despite this indirect connection, a labeling based LP and rounding for NODE-WT-MWC has not explicitly been written in the literature. Our work here gives such an LP in the more general context of SUBSET-FVS. Labeling problems such as metric labeling [63], zero-extension [14] and submodular-cost labeling [18] have provided powerful tools to address a variety of problems and our work here gives yet another application to an interesting class of problems.

We refer the reader to a survey on feedback set problems for additional information [40]. There has also been extensive work on fixed-parameter algorithms for feedback problems — see [28].

Organization: The rest of the chapter is organized as follows. In Section 5.1, we discuss the basic idea of new LP formulations. In Section 5.2, we describe the LP formulation for SUBSET-FES, and then discuss our two-step rounding scheme. Building upon these ideas, in Section 5.3 we describe a similar LP formulation and rounding scheme for SUBSET-FVS.

5.1 The idea for the new LP formulations

We outline the key idea that allows us to develop new LP relaxations for SUBSET-FES and SUBSET-FVS. It is easier to explain it for SUBSET-FES. First, it is convenient to simplify the instance via well-known

[1] Erdos-Pósa theorem relates integer packing of cycles to feedback vertex sets in undirected graphs. The relationship between integer packing of cycles and feedback vertex sets in directed graphs is much more difficult and is addressed by Reed *et al.* [73].

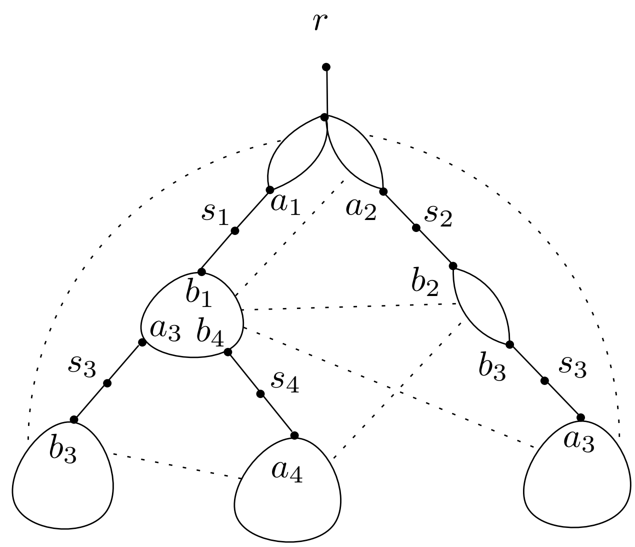


Figure 5.2: Block-cut-vertex tree of $H = G - F$. Dotted edges are edges in F that are cut

reductions; in the simplified instance each terminal s_i has degree 2 and is connected by infinite-weight edges to its neighbors a_i, b_i that are not adjacent to any other terminals. Thus, in any feasible solution of finite weight, the edges $s_i a_i$ and $s_i b_i$ are not cut; we think of these edges as *special* edges. Now, consider any minimal feasible solution $F \subset E$ such that the graph $H = G - F$ has no cycle containing a terminal. We can assume without loss of generality that G is connected, and hence the graph H is also connected by minimality of F. Consider the block-cut-vertex tree T of H [2]. No non-trivial block of H contains a terminal (otherwise there would be a cycle containing it). Thus, in H, each terminal is a cut-vertex and each special edge is a cut-edge. We can root T at a block r that does not contain any terminals; for simplicity assume that r is a single node. See Fig 5.2. Consider $k+1$ labels where terminal s_i has label i for $1 \leq i \leq k$ and the root has label $k+1$. By rooting T at r we obtain a natural label assignment for each node u of G as follows: label u by the index of the first terminal (or r) on a path in T from the block containing u to the root r. This labeling has the following property. The end points of each non-special edge in H have the same labels, and by minimality of F, the end points of edges in F receive different labels. It is important to note that the end points of some of the special edges receive different labels but they can never be cut. Thus, the problem can be viewed as finding a labeling of the nodes to minimize the cost of non-special edges whose end points receive different labels.

However, a labeling by itself does not suffice to obtain a good lower bound. One can assign the label $k+1$ to all non-terminal nodes and no non-special edges are cut. An additional property of the labeling obtained from reasoning via the block-cut-vertex tree is the following: for each terminal s_i, *exactly* one of the two neighbors a_i, b_i should be assigned the label i (the label of s_i). We can thus add this “spreading” constraint. The resulting labeling LP gets us most of the way; using just this LP, we

[2]A block in a graph G is a maximal 2-node-connected component of G. The block-cut-vertex tree is a standard decomposition of a graph into its blocks and we refer the reader to books on graph theory such as [30, 84] for more information.

can reduce the original instance to one in which each connected component "essentially" has only one interesting cycle which can be solved easily. To obtain a single LP we add constraints that ensure that the length of each cycle is at least one. To round the LP with assignment variables we borrow ideas from algorithms for multiway cut [15] and metric labeling [63] but we note that the rounding we have is subtly different because of the special edges and the spreading constraints; unlike those other problems we are not disconnecting the terminals.

For SUBSET-FVS we use a similar labeling procedure but need additional variables to take into account node weights. It is easier to understand the formulation and analysis for SUBSET-FES before seeing the description and analysis for SUBSET-FVS. In fact, some parts of the analysis for SUBSET-FVS rely on the analysis for SUBSET-FES. For this reason, we first discuss SUBSET-FES in detail.

5.2 LP-based constant factor approximation for SUBSET-FES

In this section we describe our LP-relaxation based algorithm and analysis for SUBSET-FES. First, we will assume without loss of generality that the input instance has a certain restricted structure; similar assumptions have been used previously [34] and are easy to justify. The assumptions are: (i) The input graph G is connected. (ii) Each terminal s_i has degree 2 and is connected by infinite weight edges to its neighbors a_i, b_i. (iii) No two terminals are connected by an edge or share a neighbor. (iv) There exists a special non-terminal vertex r with a single infinite weight edge incident to it.

We briefly justify these assumptions: If G is not connected, the problem can be solved separately in each connected component. If a terminal s is not degree two, we can sub-divide each edge incident to s by adding a new node; then, remove s from the set of terminals and add the new nodes to the set of terminals. If $e = uv$ is an edge of weight $w(e)$, sub-dividing e by adding a node q and setting $w(uq) = \infty$ and $w(qv) = w(e)$ does not change the problem. This can be used to justify the other assumptions as well.

For technical reasons, we also assume that every interesting cycle contains *at two* terminals. This can also be justified by subdividing the edges incident on a terminal and adding the new nodes as terminals. We perform one more reduction step to ensure that no two terminals are connected by an edge or share a neighbor. Note that in the new instance that satisfies these assumption, the number of terminals could be much larger than in the original instance.

Remark 5.1. *We refer readers who may wonder about the need or the utility of the simplifying reductions, to Section 5.4. There we provide some additional details and examples.*

5.2.1 LP formulation

Recall the structure of a minimal feasible solution discussed in Section 5.1. If $F \subset E$ is a minimal feasible solution then $H = (V, E \setminus F)$ is connected and each terminal is a cut vertex in H. Let T be the

$$
\begin{array}{llll}
 & \textsc{Subset-FES-Rel} & & \\
\min & \displaystyle\sum_{e\in E\setminus E_s} w(e)z(e) & & \\
 & s.t & & \\
 & \displaystyle\sum_{i=1}^{k+1} x(u,i) & = \quad 1 & u \in V \\
 & x(s_i,i) & = \quad 1 & 1 \le i \le k \\
 & x(r,k+1) & = \quad 1 & \\
 & x(a_i,i)+x(b_i,i) & = \quad 1 & 1 \le i \le k \\
 & z(e)-\dfrac{1}{2}\displaystyle\sum_{i=1}^{k+1}|x(u,i)-x(v,i)| & \ge \quad 0 & e = uv \in E \setminus E_s \\
 & z(e) & = \quad 0 & e \in E_s \\
 & \displaystyle\sum_{e\in C} z(e) & \ge \quad 1 & C \in \mathbb{C} \\
 & x(u,i) & \ge \quad 0 & u \in V, 1 \le i \le k+1 \\
 & z(e) & \ge \quad 0 & e \in E
\end{array}
$$

Figure 5.3: LP Relaxation for SUBSET-FES

block-cut-vertex tree of H rooted at r. Each vertex u is labeled by the index of the first terminal on a path in T from the block containing u to r, or by $k+1$ if there is no such terminal. We will use E_s to denote the set of special edges $\cup_{i=1}^{k}\{(s_i,a_i),(s_i,b_i)\}$ that are incident to the terminals and have infinite weight. Any non-special edge with different labels on the end points is cut. Special edges are never cut and for each terminal, one of the incident special edge always has different labels on the end points. We formulate an integer program based on this structure which can then be relaxed to obtain a linear program. We have two types of binary variables, the labeling variables $x(u,i)$ for each $u \in V$ and $i \in \{1,\dots,k+1\}$, and the edge variables $z(e)$ for each $e \in E$. $x(u,i)$ is an indicator variable for whether u is assigned label i. $z(e)$ is an indicator variable for whether e is cut or not. The following constraints explain our reasoning:

- Each node u is labeled by exactly one label: $\sum_{i=1}^{k+1} x(u,i) = 1$ for all $u \in V$.
- Terminals are labeled by their own index, $x(s_i,i) = 1$ for each i. Root r is labeled $k+1$, $x(r,k+1) = 1$.
- For each s_i, exactly one of a_i, b_i is labeled i: $x(a_i,i)+x(b_i,i) = 1$ for $1 \le i \le k$.
- Non-special edge $e = uv$ is cut (that is $z(e) = 1$) if u, v receive different labels and is not cut ($z(e) = 0$) if u, v receive same labels. Hence, $z(e) \ge \frac{1}{2}\sum_{i=1}^{k+1}|x(u,i)-x(v,i)|$. This can be written as a linear constraint with additional variables that we suppress for ease of notation.

- Special edges are not cut: $z(e) = 0$ for $e \in E_s$.

We also have an additional constraint. Let $\mathbb{C}$ be the set of interesting cycles. For any $C \in \mathbb{C}$, at least one of the edge is cut: hence $\sum_{e \in C} z(e) \geq 1$. There are an exponential number of such constraints but we can express them compactly via triangle inequalities and it is also easy to see that we can separate over them efficiently in polynomial-time. This constraint is essential for LP to have a bounded integrality gap. See Section 5.4 for an example illustrating this fact. Objective is to minimize $\sum_{e \in E \setminus E_s} w(e)z(e)$. We can drop the constraints that upper bound the variables by 1. The full description of the LP relaxation is given in Figure 5.3.

5.2.2 Rounding scheme and analysis

Theorem 5.2. *There is a polynomial-time algorithm that given a feasible solution $\boldsymbol{x}, \boldsymbol{z}$ to* SUBSET-FES-REL *outputs a feasible integral solution of weight at most $13 \sum_{e \in E \setminus E_s} w(e)z(e)$.*

Given a feasible solution $\mathbf{x}, \mathbf{z}$ to SUBSET-FES-REL, we round it in two steps. In the first step, we round the fractional solution using the labeling variables $x(u,i)$ to find a subset $E' \subset E$ of edges such that removing E' yields a graph $G' = G - E'$ that has very restricted structure. In particular each connected component of G' has essentially only one interesting cycle; more formally all interesting cycles in each component have the same *signature* which is defined formally below. Solving an instance in which all cycles have the same signature is easy; we can find an optimal solution. Letting E'' denote the edge set removed in the second step (we take the union of the solutions from each component), the final output of the algorithm is $E' \cup E''$.

Definition 5.1. Let $C = s_{i_1}, c_{i_1}, A_1, c'_{i_2}, s_{i_2}, c_{i_2}, A_2, \ldots, c'_{i_t}, s_{i_t}, c_{i_t}, A_t, c'_{i_1}, s_{i_1}$ be an interesting cycle where $s_{i_1}, \ldots, s_{i_t}$ are terminals, and for $1 \leq j \leq t$, $c_{i_j}, c'_{i_j} \in \{a_{i_j}, b_{i_j}\}$ and $c_{i_j}, A_j, c'_{i_{j+1}}$ is a path with no terminals; here A_j can be empty. Signature of C denoted by $sig(C)$ is defined as $s_{i_1}, c_{i_1}, c'_{i_2}, s_{i_2}, c_{i_2}, \ldots, c'_{i_t}, s_{i_t}, c_{i_t}, c'_{i_1}, s_{i_1}$. Given two cycles C_1 and C_2 we say that their signatures are the *same* if the cycles $sig(C_1)$ and $sig(C_2)$ are isomorphic as labeled graphs.

The heart of the rounding and analysis is the first step which is formalized in the lemma below.

Lemma 5.1. *Given a feasible solution $\boldsymbol{x}, \boldsymbol{z}$ to* SUBSET-FES-REL*, there is an efficient algorithm to find a subset of edges $E' \subset E$ with cost at most $12 \sum_{e \in E \setminus E_s} w(e)z(e)$ such that any two interesting cycles in the same connected component of $G' = G - E'$ have the same signature.*

It is useful to see Figure 5.4 to understand what it means for all interesting cycles to have the same signature. In particular, in each connected component H of G' that has an interesting cycle C, removing any terminal from C suffices to kill all interesting cycles in H.

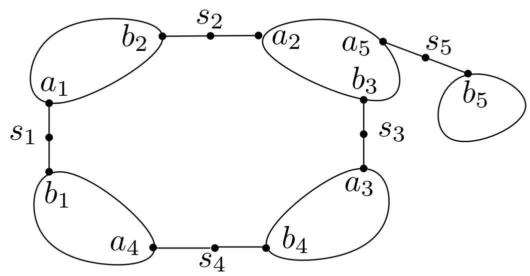


Figure 5.4: Structure of connected component in $G' = G - E'$

Algorithm 5.1 Initial Cut for SUBSET-FES

Given: Feasible solution $\mathbf{x}, \mathbf{z}$ to SUBSET-FES-REL
Pick $\theta \in (1/3, 1/2)$ uniformly at random
For $1 \leq i \leq k$, $B_i := \{u \mid x(u,i) > \theta\}$
$E' := \left(\cup_{i=1}^{k} \delta(B_i)\right) \setminus E_s$
Return E'

Algorithm 5.1 is a simple randomized algorithm that achieves the properties claimed by the preceding lemma. Here, $\delta(S)$ denote the edge boundary of set S, formally defined as $\{uv \in E \mid |\{u,v\} \cap S| = 1\}$. We note that although the algorithm is related to "ball-cutting" type schemes for cut problems such as multiway cut, there are subtle differences. It is important for our analysis that $\theta < 1/2$ which is counter intuitive from a cut perspective. Since $\theta \in (1/3, 1/2)$, B_i and B_j may intersect for $i \neq j$. Another subtlety is the fact that a special edge e may be in $\delta(B_i)$ for some i but is not allowed to be cut. These issues make the analysis tricky.

We define the label set for v, $L(v) = \{i \mid v \in B_i\}$; note that some nodes may not receive a label. We make some simple observations before proceeding with the proof of Lemma 5.1. In the analysis below, all statements hold for each choice of $\theta \in (1/3, 1/2)$, and randomness plays a role only in analyzing the expected cost of E'.

Lemma 5.2. *A node can have at most two labels that is, for each vertex $v \in V, |L(v)| \leq 2$.*

Proof: A label $i \in L(v)$ implies $x(v,i) > \theta > 1/3$. Since $\sum_{i=1}^{k+1} x(v,i) = 1$, $|L(v)| \leq 2$.

The spreading constraint and the choice of θ ensures some useful properties about the labels of a_i and b_i.

Lemma 5.3. *For each $i \in \{1, \ldots, k\}$, a_i or b_i is labeled i, that is, $i \in L(a_i) \cup L(b_i)$. Moreover if $i \notin L(a_i)$, then $L(b_i) = \{i\}$. Similarly, if $i \notin L(b_i)$, then $L(a_i) = \{i\}$.*

Proof: The constraint $x(a_i,i) + x(b_i,i) = 1$ implies that $x(a_i,i) \geq 1/2 > \theta$ or $x(b_i,i) \geq 1/2 > \theta$. Hence, $a_i \in B_i$ or $b_i \in B_i$ which is equivalent to $i \in L(a_i) \cup L(b_i)$. If $i \notin L(a_i)$, $x(a_i,i) \leq \theta < 1/2$ and therefore $x(b_i,i) = 1 - x(a_i,i) \geq 1 - \theta > 1/2$. Therefore $i \in L(b_i)$. Also, since $\sum_{j=1}^{k+1} x(b_i,j) = 1$, and $x(b_i,i) \geq 1-\theta$, for all $\ell \neq i$, $x(b_i,\ell) \leq \theta$. Hence, $L(b_i) = \{i\}$.

Lemma 5.4. *For any non-special edge $uv \in E \setminus E'$, $L(u) = L(v)$. Hence, for any two nodes u, v connected in G' by a path with only non-special edges, $L(u) = L(v)$.*

Proof: Let $uv \in E \setminus E'$ be a non-special edge with $L(u) \neq L(v)$. Then, there exists i in $L(u) \setminus L(v)$ or $L(v) \setminus L(u)$. In both cases, edge uv is in $\delta(B_i)$ and thus in E'. This leads to contradiction as $uv \in E \setminus E'$.

Now comes an important lemma on the label set of nodes in any interesting cycle remaining in G'.

Lemma 5.5. *Let C be an interesting cycle in G' with $sig(C) = s_{i_0}, c_{i_0}, c'_{i_1}, s_{i_1}, c_{i_1}, \ldots, c'_{i_{t-1}}, s_{i_{t-1}}, c_{i_{t-1}}, c'_{i_0}, s_{i_0}$ where $s_{i_0}, \ldots, s_{i_{t-1}}$ are terminals, and $c_{i_j}, c'_{i_j} \in \{a_{i_j}, b_{i_j}\}$. If $t \geq 2$, then exactly one of the following three conditions hold. Addition and subtraction of the indices here is modulo t.*

- *For $0 \leq j \leq t-1, L(c_{i_j}) = \{i_j, i_{j+1}\}, L(c'_{i_j}) = \{i_j, i_{j-1}\}$*
- *For $0 \leq j \leq t-1, L(c_{i_j}) = \{i_j\}, L(c'_{i_j}) = \{i_{j-1}\}$*
- *For $0 \leq j \leq t-1, L(c_{i_j}) = \{i_{j+1}\}, L(c'_{i_j}) = \{i_j\}$*

Proof: For ease of notation, we rename the terminals and their neighbors such that $sig(C) = s_1, a_1, b_2, s_2, a_2, \ldots, b_t, s_t, a_t, b_1, s_1$ and identify $t+1$ with 1, 0 with t when adding and subtracting indices.

Case 1: $\forall j \in \{1, \ldots, t\}, j \in L(a_j) \cap L(b_j)$. Since a_j is connected by a path containing non-special edges to b_{j+1} in C (and hence G'), by Lemma 5.4, $L(a_j) = L(b_{j+1})$. From assumption $t \geq 2$, we know that $b_{j+1} \neq b_j$ and label $j+1$ is not same as label j. And, since $j \in L(a_j)$ and $j+1 \in L(b_{j+1})$, we get $j, j+1 \in L(a_j), L(b_{j+1})$. Lemma 5.2 states that $|L(v)| \leq 2$ for all v which implies $L(a_j) = L(b_{j+1}) = \{j, j+1\}$. The labels in this case satisfies the first condition in the lemma.

Case 2: $\exists j \in \{1, \ldots, t\}, j \notin L(b_j)$. Since, $j \notin L(b_j)$, by Lemma 5.3, we have $L(a_j) = \{j\}$. Since a_j is connected to b_{j+1} via non-special edges, by Lemma 5.4, $L(a_j) = L(b_{j+1}) = \{j\}$. This implies that $j+1 \notin b_{j+1}$. Arguing inductively along the cycle we can see that $L(a_i) = L(b_{i+1}) = \{i\}$ for $1 \leq i \leq t$. The labels in this case satisfy the second condition in the lemma.

Case 3: $\exists j \in \{1, \ldots, t\}, j \notin L(a_j)$. Similar to Case 2 and labels satisfy third condition in the lemma.

Remark 5.2. *There may be nodes which are not labeled but are connected to terminals in G'. For example, if $v = a_i$ with $x(v,j) \leq \theta$ for all j, then $L(v) = \emptyset$ but v is still connected to s_i in G' since special edges are not cut. Lemma 5.5 implies that such a vertex is not part of any interesting cycle in G'.*

Next, we prove bound on expected cost[3] of the edges cut by Algorithm 5.1.

[3] It is easy to derandomize the algorithm by trying "all possible" values of θ in $(1/3, 1/2)$; we only need to try all values of $x(u,i)$.

Lemma 5.6. $\Pr[e \in E'] \le 12z(e)$*, and hence the expected cost of* E' *is at most* $12\sum_{e\in E\setminus E_s} w(e)z(e)$.

Proof: We focus on non-special edges as no special edge is cut by the algorithm. Let $e = uv$ be any such edge. Edge e is cut if and only if $e \in \delta(B_i)$ for some i. $\Pr[e \in \delta(B_i)] = \Pr[\theta \in [\min(x(u,i), x(v,i)), \max(x(u,i), x(v,i)))]$ which is at most $6|x(u,i) - x(v,i)|$ since θ is chosen uniformly from $(1/3, 1/2)$. By union bound, $\Pr[e \in E'] \le \sum_{i=1}^{k} \Pr[e \in \delta(B_i)] \le 6\sum_{i=1}^{k} |x(u,i) - x(v,i)| \le 12z(e)$.

We now prove the main structural observation, namely Lemma 5.1, that in each connected component of G' all interesting cycles have the same signature.

Proof: [Lemma 5.1] We prove the lemma for the edge set E' returned by Algorithm 5.1. The expected cost of E' is upper bounded by Lemma 5.6.

In the graph $G' = (V, E \setminus E')$, consider any two interesting cycles C_1, C_2. We will first prove that if C_1 and C_2 share a terminal, then their signature must be same. Second, we will prove that if C_1, C_2 do not share a terminal then then they must be in different connected components.

First, consider the case when C_1 and C_2 share a terminal vertex. By renaming terminal nodes and their neighbors, let the cycle C_1 be such that $sig(C_1) = s_1, a_1, b_2, s_2, a_2, \ldots, b_t, s_t, a_t, b_1, s_1$ and the shared terminal be s_1. Since each terminal has degree 2, any cycle through s_1 has to contain a_1, b_1. We can assume without loss of generality that $sig(C_2) = s_{i_1}, c'_{i_1}, c_{i_2}, s_{i_2}, c'_{i_2}, \ldots, c_{i_h}, s_{i_h}, c'_{i_h}, c_{i_1}, s_{i_1}$, where $s_{i_1} = s_1$ and $c'_{i_1} = a_1, c_{i_1} = b_1$. Since $sig(C_1) \ne sig(C_2)$ there is a smallest integer $r \ge 2$ such that $c_{i_r} \ne b_r$ and the prefix of $sig(C_2)$ till c_{i_r} agrees with $sig(C_1)$. From Lemma 5.4 we have $L(a_{r-1}) = L(b_r)$ (via C_1) and $L(a_{r-1}) = L(c_{i_r})$ (via C_2) and hence $L(b_r) = L(c_{i_r})$. Given a SUBSET-FES instance, we assumed that each interesting cycle contains at least two terminals. Hence, $t, h \ge 2$. We will consider Lemma 5.5 applied to C_1 and C_2 and the resulting consistency requirements on the labels, in particular for a_{r-1}, b_r, c_{i_r}. Note that these nodes all receive two distinct labels, or all receive exactly one label. We consider three cases based on the guarantee of Lemma 5.5 applied to C_1, and in each case derive a contradiction.

- For all $j, L(a_j) = \{j, j+1\}, L(b_j) = \{j, j-1\}$. Thus $L(a_{r-1}) = L(b_r) = \{r-1, r\}$ and $L(a_r) = \{r, r+1\}$. Lemma 5.5 applied to C_2 implies that $L(a_{r-1}) = L(c_{i_r}) = \{r-1, i_r\}$. This in particular implies that $i_r = r$, and hence $s_{i_r} = s_r$. Since we have $c_{i_r} \ne b_r$ we must have that $c_{i_r} = a_r$. However, $L(a_r) = \{r, r+1\}$ via C_1 which means that $L(a_{r-1}) \ne L(c_{i_r})$, a contradiction.

- For all $j, L(a_j) = \{j+1\}, L(b_j) = \{j\}$. Thus $L(a_{r-1}) = \{r\}$ and $L(b_r) = \{r\}$ and $L(a_r) = \{r+1\}$. Lemma 5.5 applied to C_2 implies that $L(a_{r-1}) = L(c_{i_r}) = \{r-1\}$ or $L(a_{r-1}) = L(c_{i_r}) = \{i_r\}$. Since, $L(a_{r-1}) = \{r\}$, $L(a_{r-1}) = L(c_{i_r}) = \{i_r\} = \{r\}$. Thus $i_r = r$; since $c_{i_r} \ne b_r$ we have $c_{i_r} = a_r$ but then $L(c_{i_r}) = L(a_r) = \{r\}$, a contradiction.

- For all $j, L(a_j) = \{j\}, L(b_j) = \{j-1\}$. Similar to preceding case.

Next, consider the case when C_1 and C_2 do not share a terminal. By contradiction, assume that they are connected in G'. For an interesting cycle C let $S(C)$ denote the indices of the set of terminals in

$V(C)$. From Lemma 5.5 and Lemma 5.4, for each $v \in V(C)$, $\emptyset \neq L(v) \subseteq S(C)$. Since, $S(C_1) \cap S(C_2) = \emptyset$, we conclude that $V(C_1) \cap V(C_2) = \emptyset$.

Let $u \in C_1, v \in C_2$ be two nodes in G' connected by a path not intersecting C_1 or C_2. Since, each terminal is a degree two node, neighbors of all terminals in $S(C_1)$ and $S(C_2)$ are part of cycle C_1 and C_2, we have that u and v are not terminals. Let the path between u and v in G' be $u, A_1, c_{i_1}, s_{i_1}, c'_{i_1}, A_2, \ldots, A_r, c_{i_r}, s_{i_r}, c'_{i_r}, A_{r+1}, v$ where $s_{i_1}, \ldots, s_{i_r}$ are terminals, $c_{i_j}, c'_{i_j} \in \{a_{i_j}, b_{i_j}\}$ and A_j is a path with no terminals. First, we observe that if $r = 0$ then there is a path connecting u to v without terminals and hence $L(u) = L(v)$; this is not possible since $L(u) \subset S(C_1)$ and $L(v) \subset S(C_2)$ and $S(C_1) \cap S(C_2) = \emptyset$. Assume $r \geq 1$. We have $L(u) = L(c_{i_1})$ and since $L(u) \subset S(C_1)$ and $s_{i_1} \notin V(C_1)$, we obtain that $i_1 \notin L(c_{i_1})$. This implies, via Lemma 5.3 that $L(c'_{i_1}) = \{i_1\}$.

Claim 5.1. *For $1 \leq j \leq r$, $L(c'_{i_j}) = \{i_j\}$.*

Proof: We have already established the base case that $L(c'_{i_1}) = \{i_1\}$. Assume by induction that $L(c'_{i_j}) = \{i_j\}$ for $j = \ell - 1$. By Lemma 5.4, $L(c_{i_\ell}) = L(c'_{i_{\ell-1}}) = \{i_{\ell-1}\}$. By Lemma 5.3, $L(c_{i_\ell}) = \{i_\ell\}$.

Using the above claim, $L(c'_{i_r}) = \{i_r\}$ which by Lemma 5.4 implies that $L(v) = \{i_r\}$. However, $L(v) \subset S(C_2)$ and $s_{i_r} \notin V(C_2)$ which is a contradiction.

Remark 5.3. *The assumption that each interesting cycle contains at least two terminals was used in the proof of Lemma 5.5. Without this assumption, we may not get the property of Lemma 5.1 that each interesting cycle in G' has same signature. For examples and more discussion on this, see Section 5.4.*

A part of the analysis of Algorithm 5.1 will be useful for us when analyzing the algorithm for SUBSET-FVS. The following remark captures the necessary aspects.

Remark 5.4. *Constraints involving variables $z(e)$ are used only while bounding the expected cost of the edge set E' returned by Algorithm 5.1. Hence, given any vector $\mathbf{x}$ satisfying constraints involving labeling variables(first four and $x(u,i) \geq 0$), Algorithm 5.1 returns an edge set E' such that all interesting cycles in a connected components in $G - E'$ have same signature.*

5.2.3 Second Step of Rounding

Next, we will describe the second step of rounding for SUBSET-FES-REL and finish the proof of Theorem 5.2. After the first step of the rounding we are left with a graph G'. In each connected component of G' all interesting cycles have the same signature. Consider one such component H which has an interesting cycle C and without loss of generality let $sig(C) = s_1, a_1, b_2, s_2, a_2, \ldots, s_t, a_t, b_1, s_1$. Note that t could be 1. Since the signatures of all interesting cycles are the same, any such cycle C contains the edges $s_1 a_1$ and $s_1 b_1$. Thus, to remove all interesting cycles in H it is necessary and sufficient to disconnect a_1 from b_1 in $H' = H - \{s_1 a_1, s_1 b_1\}$. This can be easily solved by finding a min-cut between a_1 and b_1 in H'. The following claim charges the cost of this min-cut to the LP solution.

Claim 5.2. *The cost of the min-cut between a_1 and b_1 in H' is at most $\sum_{e\in E(H)} w(e)z(e)$.*

Proof: Consider the feasible solution $\mathbf{x}, \mathbf{z}$ to SUBSET-FES-REL to the original instance and the lengths of the edges $z(e)$ induces on $E(H)$. The constraints of the relaxation imply that $\sum_{e\in C} z(e) \geq 1$ for each $C \in \mathbb{C}$. Since H is a subgraph of G these constrains hold for all interesting cycles in H as well. Since $z(s_1a_1)$ and $z(s_1b_1)$ are both 0, we have that $d_z(a_1, b_1) \geq 1$ in $H' = H - \{s_1a_1, s_1b_1\}$ where $d_z(a_1, b_1)$ is the shortest path distance from a_1 to b_1 according to edge lengths given by $\mathbf{z}$. Thus $\mathbf{z}$ restricted to H' is a feasible fractional solution to the standard distance based LP relaxation for the a_1-b_1 minimum cut problem in H'; this LP relaxation is the same as the dual of the maximum flow LP. Via the maxflow-mincut theorem, the integrality gap of the LP is one and in particular there is an a_1-b_1 mincut in H' of cost at most $\sum_{e\in E(H)} w(e)z(e)$.

Let E'' be the union of all the minimum cuts found in different connected components in G'. It is easy to see that $E' \cup E''$ is a feasible solution. From the preceding claim and the fact that the connected components of G' are edge disjoint we have that $w(E'') \leq \sum_{e\in E\setminus E_s} w(e)z(e)$.

We now finish the proof of Theorem 5.2 that establishes a constant factor upper bound on integrality gap for SUBSET-FES-REL.

Proof: [Theorem 5.2] Let $\alpha = \sum_{e\in E\setminus E_s} w(e)z(e)$ be the objective function value of a feasible solution $\mathbf{x}, \mathbf{z}$ to SUBSET-FES-REL. From Lemma 5.1 we obtain a set of edges E' such that $w(E') \leq 12\alpha$ and $G' = G - E'$ satisfies the property needed for the second step of the algorithm. The set of edges E'' found in the second step satisfy the property that $w(E'') \leq \alpha$. Thus $w(E' \cup E'') \leq 13\alpha$ and $E' \cup E''$ is a feasible integral solution to the given instance. This finishes the proof.

5.3 LP-based constant factor approximation for SUBSET-FVS

In this section we extend the ideas from Section 5.2 to handle SUBSET-FVS. Several of the ideas behind the rounding and the analysis are quite similar, however, there are some non-obvious technical differences that we point out as we go along. We will again assume that the input instance satisfies some restricted structure: (i) The input graph G is connected. (ii) Each terminal has infinite weight, is a degree two vertex with both the neighbors having infinite weight. (iii) No two terminals are connected by an edge or share a neighbor. (iv) There exists a special non-terminal degree one vertex r with infinite weight. (v) Each interesting cycle contains at least two terminals. Justification of these assumptions is very similar to the case of SUBSET-FES.

5.3.1 LP formulation

Let $A \subset V$ be a minimal feasible solution. Since, terminals and their neighbors have infinite weight, none of these nodes are in A. Each terminal is a cut vertex in $H = G - A$. But, unlike the setting of

$$
\begin{array}{llcrr}
 & \textbf{Subset-FVS-Rel} & & & \\
\min & \sum_{u\in V} w(u)z(u) & & & \\
 & s.t & & & \\
 & z(u)+\sum_{i=1}^{k+1} x(u,i) & = & 1 & u\in V \\
 & x(s_i,i) & = & 1 & 1\le i\le k \\
 & x(r,k+1) & = & 1 & \\
 & x(a_i,i)+x(b_i,i) & = & 1 & 1\le i\le k \\
 & z(u)+x(u,i)-x(v,i) & \ge & 0 & 1\le i\le k+1, uv\in E\setminus E_s \\
 & z(u) & = & 0 & u\in \cup_{i=1}^{k}\{s_i,a_i,b_i\} \\
 & \sum_{u\in C} z(u) & \ge & 1 & C\in\mathbb{C} \\
 & x(u,i) & \ge & 0 & u\in V, 1\le i\le k+1 \\
 & z(u) & \ge & 0 & u\in V
\end{array}
$$

Figure 5.5: LP Relaxation for SUBSET-FVS

SUBSET-FES, H might not be connected even if A is a minimal feasible solution. For simplicity we will first assume that H is connected. As before we can now obtain a labeling by considering a block-cut-vertex tree T of H rooted at r, the block containing the special non-terminal r. This leads to a labeling of nodes in $V\setminus A$ with labels 1 to $k+1$ where a node u is labeled i if s_i is the first terminal on the path from the block containing u to r in T. Note that only nodes in $V\setminus A$ receive a label. As before the property of this labeling is that for any non-special edge uv in $E(H)$, u and v receive same labels. Now we briefly address the case when H may not be connected. For each connected component we pick an arbitrary non-terminal vertex, add a dummy edge connecting it to r and consider the labeling corresponding to block-cut vertex tree of the modified graph. Note that the addition of these dummy edges is a thought experiment to justify the validity of the existence of the labeling.

As in case of SUBSET-FES, we formulate an integer program based on this structure with a few changes. We have two types of binary variables, the labeling variables $x(u,i)$ for each $u\in V$ and $1\le i\le k$ and node variables $z(u)$ for each node $u\in V$. $x(u,i)$ indicates whether or note u is assigned label i and $z(u)$ indicates whether or not u is cut. Here are some of the constraints, a minimal feasible solution satisfies, based on the reasoning via the structure of block-cut-vertex tree T.

- Either a node u is cut or is labeled by exactly one label: $z(u)+\sum_{i=1}^{k+1} x(u,i)=1$ for $u\in V$.
- No terminal s_i or its neighbors a_i, b_i are cut: $z(u)=0$ for $u\in\cup_{i=1}^{k}\{s_i,a_i,b_i\}$.
- Terminal is labeled by its own index, $x(s_i,i)=1$ for $1\le i\le k$. Root r is labeled $k+1$, $x(r,k+1)=1$.

- For each terminal s_i, exactly one of its neighbors is labeled i: $x(a_i, i) + x(b_i, i) = 1$ for $1 \leq i \leq k$.
- For each non-special edge $e = uv$, either one of u, v is cut or they have the same label. This is captured by constraints: $x(u, i) + z(u) \geq x(v, i)$ and $x(v, i) + z(v) \geq x(u, i)$ for all $1 \leq i \leq k + 1$. It is important to note that this constraint holds even for the non-special edges incident on $u \in \cup_{j=1}^{k}\{a_j, b_j\}$.
- If $\mathbb{C}$ is the set of interesting cycles, then for every $C \in \mathbb{C}$, $\sum_{u \in C} z(u) \geq 1$.

The objective is to minimize $\sum_{u \in V} w(u)z(u)$. The full description of the LP relaxation is given in Fig 5.5.

5.3.2 Rounding scheme and analysis

Theorem 5.3. *There is a polynomial-time algorithm that given a feasible solution* $\mathbf{x}, \mathbf{z}$ *to* SUBSET-FVS-REL *outputs a feasible integral solution of weight at most* $13 \sum_{u \in V} w(u)z(u)$.

The rounding scheme is similar to the case of SUBSET-FES but we remove nodes instead. As before, given a solution $\mathbf{x}, \mathbf{z}$, we round the solution in two steps. In first step, we find a subset $V' \subset V$ of nodes such that removing V' yields a graph $G' = G - V'$ such that interesting cycles in each connected component of G' have same signature. Each component behaves like an instance with single cycle and can be solved optimally. Final output of the algorithm is $V' \cup V''$ where V'' is union of the optimal solution over connected components of G'. Following lemma formalizes the first step of the algorithm:

Lemma 5.7. *Given a feasible solution* $\mathbf{x}, \mathbf{z}$ *to* SUBSET-FVS-REL*, there is an efficient algorithm to find a subset of nodes* $V' \subset V$ *with cost at most* $12 \sum_{u \in V} w(u)z(u)$ *such that any two interesting cycles in the same connected component in* $G' = G[V \setminus V']$ *have same signature.*

Algorithm 5.2 shows a simple randomized procedure to find V' which achieves the properties claimed by the preceding lemma. Here, $N(B_i)$ denotes the node boundary of set B_i formally defined as $\{v \mid v \notin B_i, \exists u \in B_i, uv \in E\}$. And, $\delta(B_i)$ denote the edge boundary formally defined as $\{(u, v) \mid |\{u, v\} \cap B_i| = 1\}$.

Algorithm 5.2 Initial Cut for SUBSET-FVS

Given a feasible solution $\mathbf{x}, \mathbf{z}$ to SUBSET-FVS-REL
Pick $\theta \in (1/3, 1/2)$ uniformly at random.
For $1 \leq i \leq k, B_i := \{u \mid x(u, i) > \theta\}$
$V' := \cup_{i=1}^{k}(N(B_i) \setminus \{a_i, b_i\}) \setminus \cup_{i=1}^{k}\{s_i\}$
Return V'

Note that we cannot remove the terminals and their neighbors since they have infinite weight. It is not obvious that V', as defined by the algorithm, does not contain a neighbor of a terminal.

Lemma 5.8. *For* $1 \leq i, j \leq k$*,* $N(B_i) \cap \{a_j, b_j\} = \emptyset$ *if* $j \neq i$*. Thus* V' *does not contain a neighbor of any terminal.*

Proof: If $u \in \{s_j, a_j, b_j\}$ for some j, then $z(u) = 0$ and we prove that $u \notin V'$. Easy to note from Algorithm 5.2 that $s_j \notin V'$. And $u \in \{a_j, b_j\}$ is in V' iff $u \in N(B_i)$ for some $i \neq j$. We prove that this is not possible. Consider $uv \in E$ where $v \neq s_j$. From LP constraint $z(u) + x(u,i) \geq x(v,i)$, and since $z(u) = 0$ we have $x(u,i) \geq x(v,i)$. Also, $x(s_j, i) = 0$ for $i \neq j$. Thus, if for some $v \in N(u)$, $v \in B_i$ for $i \neq j$ then, $u \in B_i$. Equivalently, $u \notin N(B_i)$ for $i \neq j$.

Proof of Lemma 5.7 consists of two parts. The first is to bound the expected cost of the nodes that are cut which is provided in the lemma below. The proof of this lemma is not as straight forward as the one for the case of edges in SUBSET-FES-REL.

Lemma 5.9. $\Pr[u \in V'] \leq 12z(u)$*, and hence the expected cost of* V' *is at most* $12 \sum_u w(u)z(u)$.

Proof: From Lemma 5.8 no terminals or neighbors of terminals are in V'. Consider some other node u. Then u is in V' iff $u \in N(B_i)$ for some $1 \leq i \leq k$. And $u \in N(B_i)$ iff $x(u,i) \leq \theta$ and there exists $v \in N(u)$ such that $x(v,i) > \theta$. Equivalently, $u \in N(B_i)$ iff $\theta \in [x(u,i), \max_{v \in N(u)} x(v,i))$. We will denote the interval $(1/3, 1/2) \cap [x(u,i), \max_{v \in N(u)} x(v,i))$ as $I_i(u)$. Thus, $\Pr[u \in V'] = \Pr[\theta \in \cup_{i=1}^k I_i(u)]$. If $\cup_{i=1}^k I_i(u)$ has length at most $2z(u)$ then, since θ is chosen uniformly at random from the range $(1/3, 1/2)$ of length $1/6$, $\Pr[u \in V'] \leq 12z(u)$. Next, we prove this fact. From LP constraint $z(u)+x(u,i)-x(v,i) \geq 0$, we conclude that $\max_{v \in N(u)} x(v,i) \leq z(u) + x(u,i)$.

If there is only one index i_1 such that $x(u,i_1) > 1/3$, then for $j \neq i_1$, $I_j(u) \subset (1/3, 1/3 + z(u))$. This implies that $\cup_{i=1}^k I_i(u) \subset [x(u,i_1), x(u,i_1) + z(u)) \cup (1/3, 1/3 + z(u))$. This has length at most $2z(u)$. If on the other hand there are two indices i_1, i_2 such that $x(u,i_1), x(u,i_2) > 1/3$ then, by LP constraint $z(u) + \sum_{i=1}^{k+1} x(u,i) = 1$, we get $x(u,j) < 1/3 - z(u)$ for $j \neq i_1, i_2$. This implies that $I_j(u) = \emptyset$ for $j \neq i_1, i_2$ and $\cup_{i=1}^k I_i(u) \subset (x(u,i_1), x(u,i_1) + z(u)) \cup (x(u,i_2), x(u,i_2) + z(u))$. This range also has length at most $2z(u)$. Thus, if $u \in V \setminus \cup_{i=1}^k \{s_i, a_i, b_i\}$, $\Pr[u \in V'] \leq 12z(u)$.

The second part of the proof is to show the property of the signatures in the graph $G' = G - V'$. To prove this, we rely on the analysis that we did in the setting of SUBSET-FES. Let $E' = (\cup_{i=1}^k \delta(B_i)) \setminus E_s$. We will prove that $E' \subseteq \cup_{u \in V'} \delta(u)$. Next, we will prove via the analysis from Section 5.2, that in $G - E'$ each connected component has interesting cycles with the same signature; Remark 5.4 is relevant here. This will prove that $G - V'$ has the desired property.

Proof: [Lemma 5.7] Given feasible solution $\mathbf{x}, \mathbf{z}$ to SUBSET-FVS-REL we define a new set of assignment values $\tilde{x}(u,i)$ as follows: $\tilde{x}(u,i) = x(u,i)$ for $u \in V$ and $1 \leq i \leq k$ and $\tilde{x}(u,k+1) = z(u) + x(u,k+1)$. We have for each $u \in V$, $z(u) + \sum_{i=1}^{k+1} x(u,i) = 1$ and hence $\sum_{i=1}^{k+1} \tilde{x}(u,i) = 1$. Now consider Algorithm 5.1 for SUBSET-FES with input $\tilde{\mathbf{x}}$. Let the edges set returned be E'. We observe that the algorithm only uses labels 1 to k and since $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are identical on these labels, for each i and θ, Algorithm 5.1 and Algorithm 5.2 produce the same sets $B_1, \ldots, B_k$. For this reason, we can use the analysis of Algorithm 5.1 about the structure of the graph $G - E'$ (Lemma 5.1). In particular we have the property that in every connected component of $G - E'$ all interesting cycles have the same signature. We also observe that this property remains true for $G - \tilde{E}$ if $\tilde{E}$ is a superset of E'. To finish the proof of Lemma 5.7 we now prove that $E' \subseteq \cup_{u \in V'} \delta(u)$, that is, removing V' removes every edge in E' and perhaps more.

Claim 5.3. $E' \subseteq \cup_{u \in V'} \delta(u)$.

Proof: Recall that $E' = (\cup_{i=1}^{k} \delta(B_i)) \setminus E_s$. Consider an edge $uv \in E'$. Implies that there is an i such that $v \in B_i$ and $u \notin B_i$, and also that uv is non-special edge. Note that if u is not a terminal or neighbor of a terminal then $u \in N(B_i)$ and therefore $u \in V'$ and hence $uv \in \delta(u)$. We can assume that u is not a terminal for then uv is a special edge. Thus, the only case left to consider is that u is a neighbor of terminal, say $u = a_j$ for some j, and v is not a terminal, otherwise uv is again a special edge. Since $z(u) = 0$ and uv is not a special edge, from the LP constraint $z(u) + x(u,i) \geq x(v,i)$, we have that $x(u,i) \geq x(v,i)$ which implies that $u \in B_i$ if $v \in B_i$ contradicting the fact that $u \in N(B_i)$. Thus $uv \in E'$ implies that $uv \in \cup_{u \in V'} \delta(u)$.

This finishes the proof of the lemma.

5.3.3 Second step of rounding

Here, we will describe the second step of rounding for SUBSET-FVS-REL and finish the proof of Theorem 5.3. The second step of the algorithm is to process the graph $G' = G - V'$ which has very restricted structure. Consider a connected component H of G' which has an interesting cycle C; without loss of generality s_1 is a terminal on C. Since all signatures are identical in H, disconnecting a_1 from b_1 in the graph $H' = H - s_1$ is necessary and sufficient to remove all interesting cycles in H. Since we are in the node-weighted setting we need to find a minimum weight node cut between a_1 and b_1 in H'. In analogy with Claim 5.2 we have the following.

Claim 5.4. *The cost of the minimum node-weighted cut between a_1 and b_1 in H' is at most $\sum_{u \in V(H)} w(u)z(u)$.*

Let V'' be the union of all the minimum node cuts found in each connected component of G'. It is easy to see that $V' \cup V''$ is a feasible solution. From the preceding claim and the fact that the connected components of G' are node disjoint we have that $w(V'') \leq \sum_{u \in V} w(u)z(u)$. We now finish the proof of Theorem 5.3.

Proof: [Theorem 5.3] Let $\alpha = \sum_{u \in V} w(u)z(u)$ be the objective function value of a feasible solution $\mathbf{x}, \mathbf{z}$ to SUBSET-FVS-REL. From Lemma 5.7 we obtain a set of nodes V' such that $w(V') \leq 12\alpha$ and $G' = G - V'$ satisfies the property needed for the second step of the algorithm. The set of nodes V'' found in the second step satisfy the property that $w(V'') \leq \alpha$. Thus $w(V' \cup V'') \leq 13\alpha$ and $V' \cup V''$ is a feasible integral solution to the given instance. This finishes the proof.

5.4 Further Remarks on the LP Relaxations

Cycle length constraint: SUBSET-FES-REL with only labeling constraints does not have a bounded integrality gap. It is essential to add the constraint that each interesting cycle has length at least 1: for each interesting cycle $C \in \mathbb{C}, \sum_{e \in C} z(e) \geq 1$. Consider the third graph in Figure 5.6. Without this constraint, SUBSET-FES-REL has cost 0 with $x(a_i, i) = x(b_{i-1}, i) = 1$ for $i \in \{1,2,3\}$, $x(u,1) = 1, x(w,3) = 1$, and $z(e) = 0$ for all $e \in E$, whereas the optimal solution for SUBSET-FES has cost 1.

However, we note that the cycle length constraint is only useful in the second step which can be solved via simple min-cut computations. Alternatively, we can think of the labeling approach as strengthening the naive cycle constraint-based LP which has $\Omega(\log n)$ integrality gap.

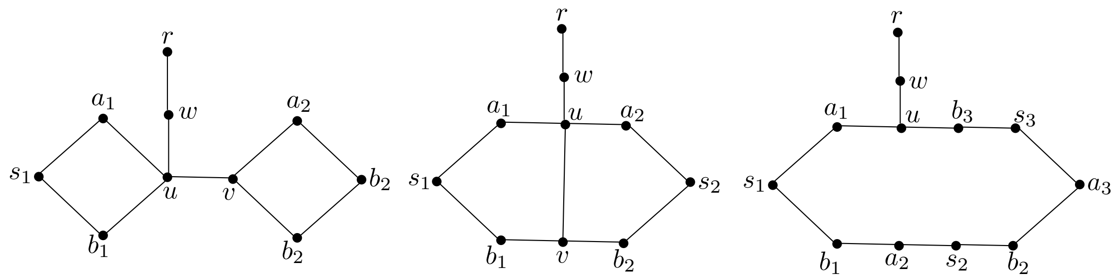


Figure 5.6: Edges incident on u, v have weight 1 except wu which has weight 0. Other edges have infinite weight.

Simplifying Reductions: Given a SUBSET-FES instance we first simplified it via reductions so that we can assume that each terminal is a degree 2 node with incident edges having infinite weight. Also, that no two terminals are connected by an edge or share a neighbor. A natural question here is whether these reductions are for convenience or whether they are essential in enabling the formulation.

Assuming that all edges incident on terminals are of infinite weight, if a terminal has degree more than 2 then we change the spreading constraint to the following constraint: $\sum_{u \in N(s_i)} x(u,i) = |N(s_i)| - 1$. If two terminals s_i, s_j are adjacent, then we can replace the above constraint for s_i, s_j to the following constraint: $\sum_{u \in N(s_i) \setminus \{s_j\}} x(u,i) + \sum_{u \in N(s_j) \setminus \{s_i\}} x(u,j) = |N(s_i)| + |N(s_j)| - 3$. If an edge incident on a terminal does not have infinite weight, we can conceptually split the edge and then write constraints based on the virtual node inserted. Doing these reductions does not change the integrality gap of the LP but simplifies the analysis considerably. Thus, for the most part the simplifying reductions are for convenience.

However, we point out the reduction that allows us to assume that each interesting cycle has at least two terminals, has a more direct impact in terms of our analysis. First, without this assumption, Lemma 5.1 does not hold and Lemma 5.5 does not hold true for cycle with one terminal. There may be two distinct interesting cycles in G', each containing one terminal and connected by a path. We may also have a connected component in G' containing two terminals with an interesting cycle containing both the terminals while another cycle containing just one terminal. Consider the first two graphs in Figure 5.6 with the following feasible solution: $x(a_1, 1) = 1/2, x(a_1, 2) = 1/2, x(w, 3) = 1$ and b_1, u, v, a_2, b_2 have the same assignment x as a_1. For any choice of $\theta \in (1/3, 1/2)$ we will get $L(a_1) = L(b_1) = L(u) = L(v) = L(a_2) = L(b_2) = \{1,2\}, L(w) = \{3\}$ and $E' = \{wu\}$.

We can modify the algorithm to incorporate these special cases. However, the analysis becomes complicated. Also, for the case when cycles in a connected component have different signatures as in the case of second graph of Figure 5.6, the second step of the rounding will lose a factor of 2 giving us an upper bound of 14 on integrality gap of SUBSET-FES-REL. One can write additional constraints to avoid this but it is simpler for our analysis to make the assumption.

5.5 Concluding Remarks

Our work opens up the possibility of obtaining improved approximations for SUBSET-FES and SUBSET-FVS. For both problems the worst-case integrality gap we know comes from corresponding gaps for EDGE-WT-MWC and NODE-WT-MWC respectively. For EDGE-WT-MWC, it lies between 1.20016 and 1.2965 and for NODE-WT-MWC, it is 2.

Another direction that we have briefly explored is to consider SUBSET-FES and SUBSET-FVS when the number of terminals k is small. We believe that we can obtain the following results:

- An algorithm for SUBSET-FVS that runs in time $\exp(k, 1/\varepsilon) \cdot \mathrm{poly}(n)$ and yields a $(4+\varepsilon)$-approximation. An algorithm for SUBSET-FVS that runs in time $\mathrm{poly}(n^k)$ and yields a 2-approximation.

- An algorithm for SUBSET-FVS that runs in time $\exp(k, 1/\varepsilon) \cdot \mathrm{poly}(n)$ and yields a $(2.5930 + \varepsilon)$-approximation. An algorithm for SUBSET-FVS that runs in time $\mathrm{poly}(n^k)$ and yields a 1.2965-approximation.

Note that even when $k = 1$ the problem is APX-Hard.

References

[1] Ittai Abraham, Cyril Gavoille, Anupam Gupta, Ofer Neiman, and Kunal Talwar. Cops, robbers, and threatening skeletons: Padded decomposition for minor-free graphs. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 79–88. ACM, 2014.

[2] Amit Agarwal, Noga Alon, and Moses S Charikar. Improved approximation for directed cut problems. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 671–680. ACM, 2007.

[3] Haris Angelidakis, Yury Makarychev, and Pasin Manurangsi. An improved integrality gap for the călinescu-karloff-rabani relaxation for multiway cut. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 39–50. Springer, 2017.

[4] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998.

[5] Vineet Bafna, Piotr Berman, and Toshihiro Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM Journal on Discrete Mathematics*, 12(3):289–297, 1999.

[6] Egon Balas and Chang Sung Yu. On graphs with polynomially solvable maximum-weight clique problem. *Networks*, 19(2):247–253, 1989.

[7] Reuven Bar-Yehuda, Dan Geiger, Joseph Naor, and Ron M Roth. Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and bayesian inference. *SIAM journal on computing*, 27(4):942–959, 1998.

[8] Ann Becker and Dan Geiger. Optimization of pearl's method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem. *Artificial Intelligence*, 83(1):167–188, 1996.

[9] Kristof Bérczi, Karthekeyan Chandrasekaran, Tamas Király, Euiwoong Lee, and Chao Xu. Global and fixed-terminal cuts in digraphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, volume 81 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:20, 2017.

[10] Kristóf Bérczi, Karthekeyan Chandrasekaran, Tamás Király, and Vivek Madan. A tight-approximation for linear 3-cut. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1393–1406. SIAM, 2018.

[11] Attila Bernáth and Tamás Király. Blocking optimal arborescences. *Mathematical Programming*, 161(1):583–601, 2017.

[12] Niv Buchbinder, Joseph Seffi Naor, and Roy Schwartz. Simplex partitioning via exponential clocks and the multiway cut problem. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing (STOC)*, pages 535–544, 2013.

[13] Niv Buchbinder, Roy Schwartz, and Baruch Weizman. Simplex transformations and the multiway cut problem. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2400–2410, 2017.

[14] Gruia Calinescu, Howard Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. *SIAM Journal on Computing*, 34(2):358–372, 2005.

[15] Gruia Calinescu, Howard J. Karloff, and Yuval Rabani. An improved approximation algorithm for multiway cut. *Journal of Computer and System Sciences*, 60(3):564–574, 2000. Preliminary version in STOC 1998.

[16] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D Sivakumar. On the hardness of approximating multicut and sparsest-cut. *computational complexity*, 15(2):94–114, 2006.

[17] Chandra Chekuri and Alina Ene. Approximation algorithms for submodular multiway partition. In *52nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 807–816. IEEE, 2011.

[18] Chandra Chekuri and Alina Ene. Submodular cost allocation problem and applications. In *International Colloquium on Automata, Languages and Programming (ICALP (1))*, pages 354–366, 2011. A longer version is available on the arXiv, abs/1105.2040.

[19] Chandra Chekuri, Sudeep Kamath, Sreeram Kannan, and Pramod Viswanath. Delay-constrained unicast and the triangle-cast problem. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, pages 804–808. IEEE, 2015.

[20] Chandra Chekuri, Sanjeev Khanna, Joseph Naor, and Leonid Zosin. A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM Journal on Discrete Mathematics*, 18(3):608–625, 2004.

[21] Chandra Chekuri, Sanjeev Khanna, Joseph Seffi Naor, and Leonid Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 109–118. Society for Industrial and Applied Mathematics, 2001.

[22] Chandra Chekuri and Vivek Madan. Constant factor approximation for subset feedback set problems via a new LP relaxation. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 797–807, 2016.

[23] Chandra Chekuri and Vivek Madan. Simple and fast rounding algorithms for directed and node-weighted multiway cut. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 797–807, 2016.

[24] Chandra Chekuri and Vivek Madan. Approximation multicut and the demand graph. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2017.

[25] Kevin KH Cheung, William H Cunningham, and Lawrence Tang. Optimal 3-terminal cuts and linear programming. *Mathematical Programming*, 106(1):1–23, Mar 2006.

[26] Fabián Chudak, Michel X Goemans, Dorit S Hochbaum, and David P Williamson. A primal–dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs. *Operations Research Letters*, 22(4):111–118, 1998.

[27] Julia Chuzhoy and Sanjeev Khanna. Polynomial flow-cut gaps and hardness of directed cut problems. *Journal of the ACM (JACM)*, 56(2):6, 2009.

[28] Marek Cygan, Fedor Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2014. In print.

[29] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994. Preliminary version in STOC 1992.

[30] Reinhard Diestel. *Graph Theory: Springer Graduate Text GTM*, volume 173. Springer New York, 2012.

[31] Alina Ene, Jan Vondrák, and Yi Wu. Local distribution and the symmetry gap: Approximability of multiway partitioning problems. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 306–325. SIAM, 2013.

[32] Robert F. Erbacher, Trent Jaeger, Nirupama Talele, and Jason Teutsch. Directed multicut with linearly ordered terminals. Preprint: `https://arxiv.org/abs/1407.7498`, 2014.

[33] Paul Erdös and Lajos Pósa. On independent circuits contained in a graph. *Canadian Journal of Mathematics. Journal Canadien de Mathématiques*, 17:347–352, 1965.

[34] Guy Even, Joseph Naor, Baruch Schieber, and Leonid Zosin. Approximating minimum subset feedback sets in undirected graphs with applications. *SIAM Journal on Discrete Mathematics*, 13(2):255–267, 2000.

[35] Guy Even, Joseph Naor, and Leonid Zosin. An 8-approximation algorithm for the subset feedback vertex set problem. *SIAM Journal on Computing*, 30(4):1231–1252, 2000. Preliminary version in *Proc. of IEEE FOCS*, 1996.

[36] Guy Even, Joseph Seffi Naor, Satish Rao, and Baruch Schieber. Divide-and-conquer approximation algorithms via spreading metrics. *Journal of ACM(JACM)*, 47:585–616, 2000.

[37] Jittat Fakcharoenphol and Kunal Talwar. An improved decomposition theorem for graphs excluding a fixed minor. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 36–46. Springer, 2003.

[38] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[39] Uriel Feige and Joe Kilian. Zero knowledge and the chromatic number. In *ccc*, page 278. IEEE, 1996.

[40] Paola Festa, Panos M Pardalos, and Mauricio GC Resende. Feedback set problems. In *Handbook of combinatorial optimization*, pages 209–258. Springer, 1999.

[41] Samuel Fiorini, Gwenaël Joret, and Ugo Pietropaoli. Hitting diamonds and growing cacti. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 191–204. Springer, 2010.

[42] Fedor V Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar f-deletion: Approximation, kernelization and optimal fpt algorithms. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 470–479. IEEE, 2012.

[43] LR Ford and DR Fulkerson. Maximal flow through a network. 1954.

[44] Ari Freund and Howard J. Karloff. A lower bound of 8/(7+1/(k-1)) on the integrality ratio of the Călinescu-Karloff-Rabani relaxation for multiway cut. *Information Processing Letters*, 75(1):43 – 50, 2000.

[45] Michael R Garey and David S Johnson. *Computers and intractability*, volume 29. W. H. Freeman New York, 2002.

[46] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Multiway cuts in directed and node weighted graphs. In *ICALP*, pages 487–498, 1994.

[47] Naveen Garg, Vijay V Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi) cut theorems and their applications. *SIAM Journal on Computing*, 25(2):235–251, 1996.

[48] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997.

[49] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Multiway cuts in node weighted graphs. *Journal of Algorithms*, 50(1):49–61, 2004. Preliminary version in *Proc. of ICALP*, 1994.

[50] Michel Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, November 1995.

[51] Michel X Goemans and David P Williamson. Primal-dual approximation algorithms for feedback problems in planar graphs. *Combinatorica*, 18(1):37–59, 1998.

[52] Daniel Golovin, Viswanath Nagarajan, and Mohit Singh. Approximating the k-multicut problem. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 621–630. Society for Industrial and Applied Mathematics, 2006.

[53] Venkatesan Guruswami, Rajsekar Manokaran, and Prasad Raghavendra. Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. In *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 573–582. IEEE, 2008.

[54] Sariel Har-Paled. Computing the k-nearest-neighbors for all vertices via Dijkstra. April 2015. `sarielhp.org/p/14/k\_nn/k\_nn.pdf`.

[55] Johan Håstad. Clique is hard to approximate withinn 1- ε. *Acta Mathematica*, 182(1):105–142, 1999.

[56] David R. Karger, Philip N. Klein, Clifford Stein, Mikkel Thorup, and Neal E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *Mathematics of Operations Research*, 29(3):436–461, 2004. Preliminary version in STOC 1999.

[57] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[58] Leonid G Khachiyan. Polynomial algorithms in linear programming. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 20(1):51–68, 1980.

[59] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th annual ACM Symposium on Theory of Computing*, STOC, pages 767–775, 2002.

[60] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? *SIAM Journal on Computing*, 37(1):319–357, 2007.

[61] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-/spl epsiv. In *Computational Complexity, 2003. Proceedings. 18th IEEE Annual Conference on*, pages 379–386. IEEE, 2003.

[62] Philip Klein, Serge A Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 682–690. ACM, 1993.

[63] Jon M. Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *Journal of the ACM (JACM)*, 49(5):616–639, 2002. Preliminary version in FOCS 1999.

[64] Jean B Lasserre. An explicit exact sdp relaxation for nonlinear 0-1 programs. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 293–303. Springer, 2001.

[65] Tom Leighton and Satish Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. pages 422–431, 1988.

[66] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832, 1999.

[67] László Lovász and Alexander Schrijver. Cones of matrices and set-functions and 0–1 optimization. *SIAM journal on optimization*, 1(2):166–190, 1991.

[68] Rajsekar Manokaran, Joseph Seffi Naor, Prasad Raghavendra, and Roy Schwartz. SDP gaps and UGC hardness for multiway cut, 0-extension, and metric labeling. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 11–20. ACM, 2008.

[69] Maryam Mirzakhani and Jan Vondrák. Sperner's colorings, hypergraph labeling problems and fair division. In *Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 873–886, 2015.

[70] Joseph Naor and Leonid Zosin. A 2-approximation algorithm for the directed multiway cut problem. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 548–548. IEEE Computer Society, 1997.

[71] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 245–254. ACM, 2008.

[72] Prasad Raghavendra and Venkatesan Guruswami. *Approximating np-hard problems efficient algorithms and their limits*. University of Washington, 2009.

[73] Bruce Reed, Neil Robertson, Paul Seymour, and Robin Thomas. Packing directed circuits. *Combinatorica*, 16(4):535–554, 1996.

[74] Michael Saks, Alex Samorodnitsky, and Leonid Zosin. A lower bound on the integrality gap for minimum multicut in directed networks. *Combinatorica*, 24(3):525–530, 2004.

[75] Paul D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.

[76] Adi Shamir. A linear time algorithm for finding minimum cutsets in reducible graphs. *SIAM Journal on Computing*, 8(4):645–655, 1979.

[77] Ankit Sharma and Jan Vondrák. Multiway cut, pairwise realizable distributions, and descending thresholds. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 724–733. ACM, 2014.

[78] Hanif D Sherali and Warren P Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990.

[79] David B Shmoys. Cut problems and their application to divide-and-conquer. *Approximation algorithms for NP-hard problems*, pages 192–235, 1997.

[80] David Steurer and Nisheeth Vishnoi. Connections between multi-cut and unique games. *Technical Report TR09-125*, 2009.

[81] Ola Svensson. Hardness of vertex deletion and project scheduling. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 301–312. Springer, 2012.

[82] Vijay V Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

[83] Ching-Chy Wang, Errol L Lloyd, and Mary Lou Soffa. Feedback vertex sets and cyclically reducible graphs. *Journal of the ACM (JACM)*, 32(2):296–313, 1985.

[84] Douglas West. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.