

NEW APPROACHES TO INTEGER PROGRAMMING

A Thesis
Presented to
The Academic Faculty

by

Karthekeyan Chandrasekaran

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Algorithms, Combinatorics and Optimization in the
School of Computer Science

Georgia Institute of Technology
August 2012

NEW APPROACHES TO INTEGER PROGRAMMING

Approved by:

Santosh Vempala, Advisor
School of Computer Science
Georgia Institute of Technology

Shabbir Ahmed
School of Industrial and Systems
Engineering
Georgia Institute of Technology

William Cook
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Santanu Dey
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Richard Karp
Department of Electrical Engineering
and Computer Sciences
University of California, Berkeley

George Nemhauser
School of Industrial and Systems
Engineering
Georgia Institute of Technology

Date Approved: 24th May 2012

To my parents, Nalini and Chandrasekaran.

ACKNOWLEDGEMENTS

My biggest thanks to my advisor Santosh Vempala. His constant reminder to strive for excellence has helped me push myself and motivated me to be a better researcher. I can only wish to attain as much clarity in my explanations as he does.

I am very grateful to the ACO program led by Robin Thomas. It has provided me an invaluable opportunity to appreciate the various aspects of theoretical research. I consider myself lucky for having been able to interact with its exceptional faculty. I also thank ARC and NSF for their generous research and travel support.

Many faculty had a positive impact on me during my time at Georgia Tech. I thank Bill Cook for his alluring treatment of linear inequalities. I thank Shabbir Ahmed, Santanu Dey, George Nemhauser and Prasad Tetali for their feedback, thoughtful questions and for all the support, help and advice. I thank all the theory faculty including Nina Balcan, Dana Randall and Vijay Vazirani for their constant encouragement. I thank Richard Karp for exposing me to a refreshing simplistic take on theoretical computer science and for all the help and advice.

I thank my collaborators Daniel Dadush, Amit Deshpande, Navin Goyal, Bernhard Haeupler, and Laci Vegh for all the insightful discussions. Amit and Navin have been wonderful mentors and more importantly, friends.

I thank all the theory students and postdocs at Georgia Tech for maintaining an enriching research environment. Subruk deserves a particular mention for being an amazing friend and a limitless source of information. I thank Elena, Laci and Will for all their help, advice and outings. I thank Akshay, Amanda, Anand, Elena, Florin, Luis, Prateek, Sarah(s), Steven, Subruk, and Ying for providing timely constructive feedback on my writings.

Finally, I thank my girlfriend Sneha for being a pillar of strength and a source of joy. She has been very patient in all my endeavours and I owe more to her love than I can write here.

TABLE OF CONTENTS

DEDICATION	iii
ACKNOWLEDGEMENTS	iv
LIST OF FIGURES	viii
SUMMARY	ix
I INTRODUCTION	1
1.1 Contributions	3
1.2 Tools and Directions	6
II COMBINATORIAL APPROACH: FEEDBACK VERTEX SET IN IMPLICIT HITTING SET MODEL	8
2.1 Implicit Hitting Set Problems	8
2.1.1 Generic vs online algorithms	11
2.1.2 Results	12
2.2 Preliminaries	13
2.2.1 Related Work	13
2.3 Algorithms	14
2.3.1 An Exact Algorithm	14
2.3.2 Augment-BFS	15
2.3.3 Hit-and-Prune	16
2.4 FVS in Random Graphs	16
2.4.1 Grow induced BFS	18
2.4.2 Analysis	19
2.4.3 Lower Bound	30
2.5 Planted FVS	32
2.5.1 Model	33
2.5.2 Properties of Planted Subset	33
2.5.3 Hitting Set is Planted FVS	35
2.5.4 Pruning Algorithm to recover Planted FVS	36
2.6 Conclusion	38

III GEOMETRIC APPROACH: DISCREPANCY-BASED ALGORITHM FOR IP	39
3.1 Introduction	39
3.1.1 Motivation	40
3.1.2 Results	41
3.2 New tool: Discrepancy for IP	42
3.3 New model for random IP	43
3.4 Preliminaries	44
3.4.1 Related work	44
3.4.2 Concentration Inequalities	45
3.5 Discrepancy of Gaussian Matrix	46
3.5.1 Partial Vector	49
3.5.2 Bounding Lengths of Gaussian Subvectors	53
3.5.3 Discrepancy Bound	54
3.6 Phase-transition in random IP	57
3.6.1 Radius for Integer Infeasibility	57
3.6.2 Threshold Radius	60
3.7 Algorithm to find Integer Point	62
3.7.1 Algorithmic Partial Vector	64
3.7.2 Finding an Integer Point	66
3.8 Conclusion	69
IV LP APPROACH: CONVERGENCE OF CUTTING PLANES FOR MIN-COST PERFECT MATCHING	71
4.1 Introduction	71
4.1.1 Motivation	72
4.1.2 Results	74
4.2 Perfect Matchings and Linear Programs	75
4.3 A Cutting Plane Algorithm for Perfect Matching	77
4.3.1 Half-integrality for faster cut-generation	77
4.3.2 Retain cuts with positive dual values	78
4.3.3 Algorithm C-P-Matching	79

4.4	Factor-critical matchings	80
4.5	Analysis	82
4.5.1	Overview and Organization	83
4.5.2	Contractions and Half-Integrality	86
4.5.3	Structure of Dual Solutions	89
4.5.4	Uniqueness	100
4.5.5	Half-Integral Primal-Dual Algorithm	104
4.5.6	Convergence	111
4.6	Conclusion	118
	REFERENCES	120

LIST OF FIGURES

1	Exact Algorithm for IHS	15
2	Algorithm Augment-BFS	16
3	Algorithm Hit-and-Prune(k)	16
4	BFS Exploration	18
5	Algorithm Grow-induced-BFS	20
6	Planted Model	33
7	Random IP instance $P(n, m, x_0, R)$	44
8	Algorithm Round-IP	63
9	Algorithm Edge-Walk	65
10	Cutting plane method for matchings	74
11	Non-integral Solution to Bipartite Relaxation	75
12	Counterexample to the half-integrality conjecture.	78
13	Algorithm C-P-Matching	80
14	Algorithm Positively-fitting-dual-opt	96
15	Possible primal updates in the Half-integral Primal-Dual algorithm	105

SUMMARY

Integer Programming (IP) is a powerful and widely-used formulation for combinatorial problems. The study of IP over the past several decades has led to fascinating theoretical developments, and has improved our ability to solve discrete optimization problems arising in practice. This thesis makes progress on algorithmic solutions for IP by building on combinatorial, geometric and Linear Programming (LP) approaches.

We use a combinatorial approach to give an approximation algorithm for the feedback vertex set problem (FVS) in a recently developed Implicit Hitting Set framework [44, 45]. Our algorithm is a simple online algorithm which finds a nearly optimal FVS in random graphs. We also propose a planted model for FVS and show that an optimal hitting set for a polynomial number of subsets is sufficient to recover the planted subset.

Next, we present an unexplored geometric connection between integer feasibility and the classical notion of *discrepancy* of matrices [66]. We exploit this connection to show a phase transition from infeasibility to feasibility in random IP instances. A recent algorithm for small discrepancy solutions [56] leads to an efficient algorithm to find an integer point for random IP instances that are feasible with high probability.

Finally, we give a provably efficient implementation of a cutting-plane algorithm for perfect matchings. In our algorithm, cuts separating the current optimum are easy to derive while a small LP is solved to identify the cuts that are to be retained for later iterations. Our result gives a rigorous theoretical explanation for the practical efficiency of the cutting plane approach for perfect matching evident from implementations [38, 69].

In summary, this thesis contributes to new models and connections, new algorithms and rigorous analysis of well-known approaches for IP.

CHAPTER I

INTRODUCTION

Integer Programming (IP) is a widely-used formulation for discrete optimization/combinatorial problems. In a discrete optimization problem, the variables belong to a discrete set, typically a subset of integers. This integer restriction leads to a dichotomy regarding solvability of problems: some combinatorial problems can be solved easily by exploiting the restriction, while others become very difficult to solve. Researchers in theoretical computer science and optimization have long sought to understand this dichotomy in the solvability of combinatorial problems.

Most combinatorial problems involve either verifying feasibility or optimizing an objective function subject to a given set of constraints. In the feasibility version of IP, the goal is to verify if there exists an integer point $x \in \mathbb{R}^n$ satisfying a given set of linear constraints $Ax \leq b$, where $A \in \mathbb{R}^{m \times n}$ is a matrix and $b \in \mathbb{R}^m$ is a vector. The linear constraints describe the facets of a polytope, and thus, one would like to verify whether a given polytope contains an integer point. In the optimization version, the goal is to identify an integer point $x \in \mathbb{R}^n$ that maximizes a given objective function $c^T x$, where $c \in \mathbb{R}^n$, subject to a set of linear constraints $Ax \leq b$:

$$\begin{aligned} & \max c^T x \\ & \text{subject to } Ax \leq b, \\ & \quad x \in \mathbb{Z}^n. \end{aligned}$$

For several decades integer programs have been used to model a variety of combinatorial problems including network design, scheduling, routing, facility location, matching, and so forth [59, 48]. The quest to find essential inequalities that describe the convex-hull of integer points in a polytope led to the development of *polyhedral theory* [64]. This, in turn, inspired the (now broadly accepted) notion of calling an algorithm *efficient* if it runs in

time polynomial in the size of the input representation. The discovery of the ellipsoid algorithm for Linear Programming (LP) [46] and the pioneering work of Grötschel, Lovász and Schrijver [39], Karp and Papadimitriou [43] and Padberg and Rao [60] linked polyhedral characterizations of polytopes with the ability to design efficient algorithms by establishing *optimization equals separation*. The pursuit of an efficient algorithmic solution for general integer programs gave birth to several new avenues which have become a rich field of study by themselves. For example, this opened the door to algorithmic aspects of lattice problems resulting in the discovery of lattice basis reduction [49, 50, 41]; the cutting plane method to solve IPs led to the theory of valid inequalities and the emergence of cutting plane proofs as a powerful proof technique in mathematical logic [18]. We refer the reader to [19] for a more elaborate historical overview of this area.

The algorithmic complexity of IP was studied even before the theory of NP-completeness was introduced. In fact, Dantzig suggested the possibility of IP being a *complete* problem before the Cook-Levin theory of NP-completeness [22]. Combinatorial structure of certain IP instances have been exploited to solve them efficiently either directly (e.g., matching, minimum cut, etc.), or by devising an efficient separation oracle to be used in conjunction with the ellipsoid algorithm. The feasibility version of general IP is NP-complete and appears in Karp's original list [42]. The best-known theoretical bound on the running time for general IP is essentially $n^{O(n)}$ from 1987 (here, n is the number of variables) [41]. Some popular heuristics employed in practice to solve IPs are the cutting plane method and the branch-and-bound method. Despite the poor theoretical running time guarantees of both methods, a combination of the two is implemented in commercial IP solvers to obtain good bounds on the objective value.

A fundamental challenge in designing an algorithm for the feasibility version of IP seems to be the lack of a *duality theory* for IP. Duality theory for LP gives an efficient method to show that no point satisfies a given set of linear constraints. On the contrary, we lack an efficient method to show that no integer point satisfies a given set of linear constraints. Given that IP is NP-complete, but abounds in practice, it is natural to pursue the directions that have been explored for other NP-complete problems – (1) approximation algorithms,

(2) heuristic algorithms and (3) algorithms for probabilistic instances. Approximation algorithms have been designed for various combinatorial problems by utilizing their underlying combinatorial structure. The heuristic direction for IP has been well-explored through the advent of branch-and-bound and cutting-plane methods; yet, there is no rigorous explanation for why these heuristic methods are fast in practice, especially for problems solvable in polynomial-time using other algorithms. Although probabilistic instances have been studied for many combinatorial problems, very little attention has been paid to probabilistic IP instances in the literature.

In this thesis, we pursue these three directions to solving IPs and make contributions to models, algorithms and analysis. On the algorithmic front, our contributions are motivated by three prominent approaches: combinatorial, geometric and LP-based.

- We develop a combinatorial approach to analyze simple online algorithms for the *Feedback Vertex Set* problem (FVS) in random graphs and planted random graphs.
- We present an unexplored geometric connection between integer feasibility and the classical notion of *discrepancy* of matrices. We exploit this connection to show a phase transition from infeasibility to feasibility in random IP instances, and give an algorithm to find an integer point for random IP instances that are feasible with high probability.
- We give an LP-based cutting plane algorithm to find a minimum-cost *perfect matching*. Moreover, our cutting plane implementation is provably efficient.

Our contributions include new models for planted FVS and random IP instances. In the following section we give a detailed exposition of our contributions.

1.1 Contributions

Combinatorial Approach: Implicit Hitting Set

The multi-genome alignment problem (MGA) is a fundamental problem in computational biology. In a fresh attempt to find an efficient heuristic to solve this problem, Karp and Moreno-Centeno give a *hitting set* formulation of this problem [44].

The hitting set problem is a classic NP-hard problem that models many well-known combinatorial problems including vertex cover and set cover. Here, we are given a universe U of discrete elements and a collection \mathcal{T} of subsets of U . A set is said to be a *hitting set* for the collection if it has non-empty intersection with every subset in the collection. The objective is to find a hitting set of minimum cardinality. A simple greedy algorithm finds a hitting set whose size is at most $\log |U|$ times the size of the smallest hitting set.

In the hitting set formulation of MGA given by Karp and Moreno-Centeno, the number of subsets $|\mathcal{T}|$ is exponential in the size of the universe. Under this setting, each iteration of the greedy algorithm would run for exponential time, thereby ruling out the possibility of using it in practice. This motivated them to define the *implicit hitting set* (IHS) framework. They also gave a heuristic algorithm with strong experimental results to find a small hitting set for MGA in this framework. There has been a great need to understand the significance of this framework towards developing approximation algorithms for the combinatorial optimization problems mentioned in their list of IHS problems [45].

In Chapter 2, we use a combinatorial approach to give an approximation algorithm for the feedback vertex set problem in the Implicit Hitting Set framework. Our algorithm is a simple online algorithm that finds a nearly optimal feedback vertex set in random graphs when the cycles in the graph are revealed in a *Breadth-First-Search* order. We also propose a planted model for the feedback vertex set problem similar to the well-studied planted clique/planted dense-subgraph problems. In this model, we show that an optimal hitting set for a polynomial number of subsets is sufficient to recover the planted subset.

Geometric Approach: The Discrepancy Connection

Our geometric approach is inspired by the simple idea that a polytope is likely to contain an integer point if it contains a large ball. We quantify the radius of the largest inscribed ball that guarantees integer feasibility as a function of the constraint matrix describing the polytope. We use this approach to initiate the study of integer feasibility of probabilistic IP instances.

Probabilistic analysis models uncertainty by assuming inputs are drawn from a distribution. It attempts to explain the efficient behavior of heuristic algorithms on “typical” instances of problems arising in practice. Probabilistic instances have been studied for numerous combinatorial problems e.g., random satisfiability [13, 12, 17, 8, 30], random knapsack [6] and various other graph problems on random graphs [7]. The theory of random satisfiability and random graph instances revealed a fascinating *phase transition* phenomenon. Among Karp’s original list of 21 NP-complete problems, IP is one of the few that has not been satisfactorily understood for random instances.

In Chapter 3, we propose a natural model for random IP instances — instances generated by random constraint matrices. In this model, we show a transition from integer infeasibility to feasibility as the radius of the largest inscribed ball increases. From a purely probabilistic perspective, our main contribution is a bound on the *discrepancy* of random Gaussian matrices. We also use a recent algorithm due to Lovett and Meka [56] to obtain an efficient algorithm to find integer points in random polytopes.

LP-based Approach: An Efficient Cutting-Plane Algorithm

The cutting plane algorithm is a leading approach to solve IPs. It proceeds by solving a linear relaxation of the problem (obtained by dropping integrality constraints) and repeatedly adding inequalities (cuts) separating the optimum from the convex hull of integer solutions while the optimum is not integral. The cutting plane algorithm was originally proposed and used by Dantzig, Fulkerson and Johnson to solve a Traveling Salesman Problem on 49 vertices in 1954 [21]. The cuts that they added were ingenious without a rigorous justification. In 1958, Gomory gave a methodical way to generate cuts efficiently [34, 35, 36]. He also showed that at most 2^n cuts are needed to obtain an integer solution for binary integer programs. Improving on this remains an important open problem.

From an algorithmic point of view, cutting plane algorithms have been implemented in commercial IP solvers for more than three decades now. Despite a tremendous body of work, a rigorous explanation for the efficiency of the performance of cutting plane algorithm in practice is missing. It is striking that the only known bound for cutting plane algorithms

over the past fifty-plus years is Gomory’s 2^n bound for binary integer programs. A more realistic step towards investigating the practical efficiency of cutting plane algorithms for general IPs is to prove a bound on the number of cuts needed for combinatorial problems known to be solvable efficiently. Matching is one of the most fundamental problems that can be solved efficiently. A cutting plane algorithm for matching has been discussed by several authors including Padberg and Rao [61] and Lovász and Plummer [54]. Grötschel and Holland [38] and Trick [69] provided experimental evidence to suggest that this algorithm is as efficient as Edmonds’ landmark minimum-cost perfect matching algorithm [28]. Their results have eluded a rigorous explanation thus far.

In Chapter 4, we provide this by giving a provably efficient implementation of a cutting-plane algorithm to find perfect matchings. A major step in our analysis involves showing that there exists a short sequence of LPs that converge to the final integral solution. Moreover, the optima to each LP in the sequence is half-integral and is supported on a disjoint union of odd cycles and matching edges. This enables us to identify potential new cuts easily. Our algorithm adopts an LP-based approach to retain/drop inequalities from earlier rounds. This method of dropping inequalities adds further theoretical justification for dropping cuts in computational implementations of cutting plane algorithms.

1.2 Tools and Directions

We conclude this chapter with a brief summary of our tools and future directions.

- (i)
 - The main tool in our combinatorial approach is a new model for planted FVS and an analysis of simple algorithms for FVS in random graph in the IHS framework.
 - An interesting direction is to develop approximation algorithms for other IHS problems given in Karp’s list [45] as well as to solve their planted and random versions exactly.
- (ii)
 - The new geometric connection between discrepancy and integer feasibility that we develop gives a general method to check feasibility of arbitrary IP instances.

- Our method could potentially be exploited for combinatorial IP instances. Another immediate question is to analyze the integer feasibility behavior of IP instances whose constraint matrices are chosen from more general distributions, e.g., logconcave distributions.
- (iii)
- Our main insight in the cutting plane implementation for perfect matchings is an LP-based machinery to generate cuts leading to strong structural properties of the intermediate LP-optima.
 - It remains open to use this machinery for other closely related combinatorial problems to give an efficient cutting plane algorithm (e.g., subtour elimination).

CHAPTER II

COMBINATORIAL APPROACH: FEEDBACK VERTEX SET IN IMPLICIT HITTING SET MODEL

In this Chapter, we adopt a combinatorial approach to solving large instances of the hitting set problem. A hitting set for a collection of sets is a set that has non-empty intersection with each set in the collection; the hitting set problem is to find a hitting set of minimum cardinality. Motivated by instances where the collection of sets is large, Karp introduced the notion of implicit hitting set problems (IHS) [45]. In IHS, the collection of sets to be hit is not listed explicitly; instead an oracle is provided, which, given a set H , either confirms that H is a hitting set or returns a set that H does not hit. Karp showed a number of classic examples of implicit hitting set problems including feedback vertex set, max-cut, TSP, etc. Although the framework proved to be helpful in designing heuristics [44, 45], algorithms in the framework with approximation guarantees were elusive.

We develop approximation algorithms by presenting a simple on-line algorithm for the minimum feedback vertex set (FVS) problem in the IHS framework. In particular, our algorithm gives a nearly-optimal FVS in random graphs. We also propose a planted model for FVS in directed random graphs. Here, we show that a hitting set for a polynomial-sized subset of cycles is a hitting set for the planted random graph and this allows us to exactly recover the planted FVS. The results in this chapter are joint work with Richard Karp, Erick Moreno-Centeno and Santosh Vempala and appeared in *Proceedings of the Symposium on Discrete Algorithms, 2011* [9].

2.1 Implicit Hitting Set Problems

The hitting set problem is a classic NP-hard problem that models many well-known combinatorial problems including vertex cover and set cover. The input to the problem consists of a universe U of discrete elements and a collection \mathcal{T} of subsets of U . A subset $H \subseteq U$ is said to be a *hitting set* for the collection if $H \cap S \neq \emptyset$ for every $S \in \mathcal{T}$. The objective is to

find a hitting set of minimum cardinality. The hitting set problem can be formulated as a 0 – 1 integer program. For each element $i \in U$, let x_i be a 0 – 1 variable indicating whether element i is selected. The integer program is given by

$$\begin{aligned} \min \quad & \sum_{i \in U} x_i \\ \text{s.t.} \quad & \sum_{i \in S} x_i \geq 1 \quad \forall S \in \mathcal{T}, \\ & x_i \in \{0, 1\} \quad \forall i \in U. \end{aligned}$$

The following greedy algorithm gives the best-known approximation guarantee: repeatedly pick an element from the universe that is present in the largest number of sets yet to be hit. This algorithm finds a hitting set whose size is at most $\log_2 |U|$ times the size of the smallest hitting set.

In a fresh attempt to solve the fundamental problem of multi-genome alignment (MGA) in computational biology, Karp and Moreno-Centeno formulated it as a hitting set problem [44]. Their hitting set formulation contained an exponential number of subsets to be hit. Consequently, obtaining a hitting set with approximation factor $\log_2 |U|$ using the greedy algorithm which examines all subsets is unreasonable for practical applications. Motivated by the possibility of algorithms that run in time polynomial in the size of the universe, Karp introduced the implicit hitting set framework [45].

In an Implicit Hitting Set problem (IHS), the input is a universe U and an *efficient implicit oracle*. The functionality of an implicit oracle is very similar to that of the conventional *separation oracle*: the implicit oracle is an efficient algorithm that, on input $H \subseteq U$, runs in time $\text{poly}(|U|)$ to return a subset that is not hit by H , if one exists. Thus, the collection \mathcal{T} of subsets to be hit is not specified explicitly. The objective is to find a small hitting set by making at most $\text{poly}(|U|)$ queries to the implicit oracle.

Implicit Hitting Set problems are special cases of implicit optimization problems where the constraints are not listed explicitly but are specified implicitly using a succinct representation or an auxiliary algorithm. IP is a well-known example of an implicit optimization problem — the constraints that define the convex-hull of integer solutions are implied by the constraints of a linear program.

We now present some well-known examples of implicit hitting set problems. We describe the hitting set formulation by specifying the universe and the subsets to be hit. We discuss the existence of an efficient implicit oracle for the problem of interest in this chapter. Efficient implicit oracles can be designed for the rest of the problems similarly [44].

- **Feedback Vertex Set in a Graph or Digraph**

Universe: Set of vertices of graph or digraph G .

Subsets: Vertex sets of simple cycles in G .

Given a graph $G(V, E)$, the goal is to find a subset $S \subseteq V$ of smallest cardinality so that every cycle in the graph contains at least one vertex from S . We note that the number of cycles could be exponential in the size of the graph. Yet, one can design an efficient implicit oracle: in order to check whether a proposed set H hits all cycles (i.e., is a feedback vertex set) or find a cycle that is not hit by H , use a breadth-first search procedure to identify cycles after removing the subset of vertices H from the graph. The existence of a polynomial time implicit oracle shows that FVS is an instance of the implicit hitting set problem.

- **Feedback Edge Set in a Digraph**

Universe: Set of edges of digraph G .

Subsets: Edge sets of simple cycles in G .

- **Max Cut**

Universe: Set of edges of graph G .

Subsets: Edge sets of simple odd cycles in G .

- **k-Matroid Intersection**

Universe: Common ground set of k matroids.

Subsets: Subsets in the k matroids.

- **Maximum Feasible Set of Linear Inequalities**

Universe: A finite set of linear inequalities.

Subsets: Minimal infeasible subsets of the set of linear inequalities.

- **Undirected TSP**

Universe: Set of edges of graph $G = (V, E)$.

Subsets: Triplets of edges incident to a node, and edge sets of simple cycles of length strictly less than $|V|$.

- **Synchronization in an Acyclic Digraph**

Universe: A collection U of pairs of vertices drawn from the vertex set of an acyclic digraph G .

Subsets: Minimal collection C of pairs from U with the property that, if each pair in C is contracted to a single vertex, then the resulting digraph contains a cycle.

2.1.1 Generic vs online algorithms

The implicit oracle leads to a natural generic algorithm: first (1) propose a candidate hitting set H , then (2) use the oracle to check if the candidate set hits all the subsets, and if not, obtain a subset S that has not been hit, and finally (3) refine H based on S (by adding/deleting elements) and repeat until a hitting set is found. This algorithm is indeed a generalization of online algorithms for hitting set problems. In the online hitting set problem, the universe is specified in advance while the subsets to be hit arrive online. On obtaining a subset, the algorithm has to decide which new element to include in the hitting set and commit to the element. Thus, the online algorithm is a restricted version of the generic algorithm where the refinement procedure can only add elements. Moreover, only those subsets that have not been hit by the candidate set are revealed by the implicit oracle thereby saving the algorithm from having to examine all subsets in \mathcal{T} .

It is perhaps tempting to use known algorithms for online hitting set to solve implicit hitting set problems, especially by considering the implicit oracle as an online oracle that reveals subsets to be hit in adversarial order. The current best-known online algorithm due to Alon, Awerbuch and Azar [2] finds a hitting set whose size is at most $O(\log |U| \log |\mathcal{T}|)$ times the size of the smallest hitting set. The main drawback of this algorithm is the main motivation for defining the *implicit* hitting set framework – the number $|\mathcal{T}|$ of underlying subsets to be hit could be exponential in the size of the universe $|U|$. Further, there is

no need to assume adversarial order since the implicit oracle is an efficient algorithm that could be designed to output sets in a favorable order.

2.1.2 Results

We apply the implicit hitting set framework and specialize the generic algorithm for FVS. In this framework, we are allowed to examine only a polynomial number of cycles. In order to find a small FVS, we need the oracle to output cycles that have not yet been hit in a natural, yet helpful manner. If the oracle is adversarial, this could force the algorithm to examine almost all cycles. We consider two natural oracles: one that outputs cycles in breath-first search (BFS) order and another that outputs cycles in increasing order of size. We design simple refinement strategies using these oracles that lead to algorithms with good performance guarantees in random graphs.

Random Graph Models. By undirected random graphs, we refer to the Erdős-Rényi model $G_{n,p}$ – these are graphs on n vertices in which each edge is chosen independently with probability p . By directed random graphs, we refer to the following well-established model $D_{n,p}$ [67] – these are directed graphs obtained by picking an instance of a random graph $G_{n,2p}$ and orienting each undirected edge $\{u, v\}$ in one of the two directions $\{u \rightarrow v, v \rightarrow u\}$ with equal probability.

We prove that if cycles in random graph $G_{n,p}$ are obtained in a breadth-first search ordering, there exists an efficient algorithm that examines a polynomial collection \mathcal{T}' of cycles to build an FVS of size $n - (1/p) \log(np)(1 - o(1))$ with probability at least $3/4$, when $p = o(1)$ ¹. A similar result for directed random graphs using the same algorithm follows by ignoring the orientation of the edges. Our algorithm is an online algorithm i.e., it commits to only adding and not deleting vertices from the candidate FVS. We also show a matching lower bound for FVS in random graphs.

Next, we address the *planted* FVS problem in the IHS framework. We propose a general planted model for the feedback vertex set problem in directed graphs. In this model, a subset of δn vertices, for some constant $0 < \delta \leq 1$, is chosen to be the feedback vertex

¹Throughout this chapter, $o(1)$ is with respect to n .

set; the subgraph induced on the complement is a random directed acyclic graph (DAG) and all the other arcs are present independently with probability p . The objective is to recover the planted feedback vertex set. We prove that the optimal hitting set for cycles of bounded size is the planted FVS. Consequently, ordering the cycles according to their sizes and finding an approximately optimal hitting set for the small cycles is sufficient to recover the planted FVS. This also leads to an online algorithm when cycles are revealed in increasing order of their size with ties broken arbitrarily.

Organization. The rest of the chapter is organized as follows. In Section 2.2, we present the related work for the minimum FVS problem. In Section 2.3, we present an exact algorithm for IHS and focus on specializing this algorithm to the FVS problem. In Section 2.4, we analyze the performance of the specialized algorithm for FVS in random graphs and show near-optimality of this algorithm. In Section 2.5, we present the model for planted FVS formally and give an algorithm to recover the planted subset by examining cycles in increasing order of size.

2.2 Preliminaries

2.2.1 Related Work

FVS is a well-studied problem in the context of designing VLSI layouts and deadlock removal in Operating Systems. The decision version of the feedback vertex set problem is NP-complete for both directed as well as undirected graphs [42]. For undirected graphs, the FVS can be approximated upto a factor of two [5, 4, 3]. It is also known that the approximation ratio cannot be improved unless the approximation factor of Vertex Cover can be improved beyond two [3]. It remains open if one can approximate the FVS in directed graphs upto a constant factor. The best-known approximation factor for directed graphs is $O(\log k \log \log k)$, where k is the size of the minimum FVS, due to Seymour [65]. The FVS problem is known to be solvable in polynomial time for various families of graphs including cubic graphs [51, 70], permutation graphs [52] and interval and comparability graphs [53].

A result of Fernandez de la Vega [26] shows that $G_{n,p}$ has an induced tree of size at least $(2/p) \log np(1 - o(1))$, when $p = o(1)$. This gives the best possible existential result

for FVS in random graphs: there exists an FVS of size at most $n - (2/p) \log np(1 - o(1))$ with high probability in $G_{n,p}$, when $p = o(1)$. We note that this result is not algorithmic although de la Vega gives a greedy algorithm to obtain the largest induced tree of size $(1/p) \log np(1 - o(1))$ in [25]. This algorithm is based on growing the induced forest from the highest labeled vertex and does not fall in the implicit hitting set framework (when the graph is revealed as a collection of cycles).

A tight lower bound on the size of FVS in $D_{n,p}$ is due to Spencer and Subramanian [67]. The lower bound is a consequence of the following result on the dual problem – maximum induced acyclic subgraph.

Theorem 2.2.1 (Spencer-Subramanian [67]). *Consider the random graph $D_{n,p}$, where $np \geq W$, for some fixed constant W . Let $r = (2/\log(1-p)^{-1})(\log(np) + 3e)$. Every subgraph induced by any subset of r vertices in G contains a cycle in it with high probability.*

2.3 Algorithms

In this section, we mention an exact algorithm for implicit hitting set problems. We then focus on specializing this algorithm to find a small FVS in directed and undirected graphs.

2.3.1 An Exact Algorithm

We give an exact algorithm for solving instances of the implicit hitting set problem optimally with the aid of an oracle and a subroutine for the exact solution of (explicit) hitting set problems. The guiding principle is to build up a short list of important subsets that dictate the solution, while limiting the number of times the subroutine is invoked, since its computational cost is high.

A set $H \subset U$ is called *feasible* if it is a hitting set for the implicit hitting set problem, and *optimal* if it is feasible and of minimum cardinality among all feasible hitting sets. Whenever the oracle reveals that a set H is not feasible, it returns $c(H)$, a subset that H does not hit. Each generated subset $c(H)$ is added to a growing list Γ of subsets. A set H is called Γ -feasible if it hits every subset in Γ and Γ -optimal if it is Γ -feasible and of minimum cardinality among all Γ -feasible subsets. If a Γ -optimal set K is feasible then it is

necessarily optimal since K is a valid hitting set for the implicit hitting set problem which contains subsets in Γ , and K is the minimum hitting set for subsets in Γ . Thus the goal of the algorithm is to construct a feasible Γ -optimal set.

1. Initialize $\Gamma \leftarrow \emptyset$.
2. Repeat:
 - (a) $H \leftarrow U$.
 - (b) Repeat while there exists a Γ -feasible set $H' = (H \cup X) - Y$ such that $X, Y \subseteq U$, $|X| < |Y|$:
 - i. If H' is feasible then $H \leftarrow H'$; else $\Gamma \leftarrow \Gamma \cup \{c(H')\}$.
 - (c) Construct a Γ -optimal set K .
 - (d) If $|H| = |K|$ then return H and halt (H is optimal); if K is feasible then return K and halt (K is optimal); else $\Gamma \leftarrow \Gamma \cup \{c(K)\}$.

Figure 1: Exact Algorithm for IHS

Remark. Since the exact algorithm uses a subroutine that solves an NP-hard problem optimally, its worst-case execution time is exponential in $|U|$. Its effectiveness in practice depends on the choice of the missed subset that the oracle returns [44].

2.3.2 Augment-BFS

In this section, we specialize our exact algorithm for IHS to solve a candidate instance, namely the FVS. We use an oracle that outputs cycles in a breadth-first-order. Instead of the exact algorithm for solving the (explicit) hitting set problem, we use a simpler strategy of picking a vertex from each missed cycle. Essentially, the algorithm considers cycles according to a breadth-first search ordering and maintains an induced tree on a set of vertices denoted as surviving vertices. The vertices deleted in the process will constitute a feedback vertex set. Having built an induced tree on surviving vertices up to a certain depth i , the algorithm is presented with cycles obtained by a one-step BFS exploration of the surviving vertices at depth i . For each such cycle, the algorithm picks a vertex at depth $i + 1$ to delete. The vertices at depth $i + 1$ that are not deleted are added to the set of surviving vertices, thereby leading to an induced tree on surviving vertices up to depth $i + 1$.

- | |
|--|
| <ol style="list-style-type: none"> 1. Start from an arbitrary vertex as a surviving vertex. Initialize $i=1$. 2. Repeat: <ol style="list-style-type: none"> (a) Obtain cycles induced by one step BFS-exploration of the surviving vertices at depth i. Delete vertices at depth $i+1$ that are present in these cycles. Declare the remaining vertices at depth $i+1$ as surviving vertices. (b) If no vertices at depth $i+1$ are surviving vertices, terminate and output the set of all deleted vertices. (c) $i=i+1$. |
|--|

Figure 2: Algorithm Augment-BFS

2.3.3 Hit-and-Prune

In this section, we give another natural algorithm to find an FVS. Here, we use an oracle that returns cycles according to increasing order of size. The idea is essentially to obtain an approximate hitting set for all cycles of length k , where k is an input parameter. This set could potentially contain the smallest hitting set. In a second phase, we prune the set in an attempt to recover a smaller hitting set. This is the algorithm that we will use to recover planted FVS.

- | |
|--|
| <ol style="list-style-type: none"> 1. Obtain cycles in increasing order of size until all cycles of length k are obtained. Let \mathcal{T}' be the subset of cycles. Let S be the empty set. 2. While there exists a cycle $T \in \mathcal{T}'$ such that S does not hit T, <ol style="list-style-type: none"> (a) Add all vertices in T to S. 3. Return H, where $H = \{u \in S : \exists k\text{-cycle through } v \text{ in the subgraph induced by } V \setminus S \cup \{u\}\}$. |
|--|

Figure 3: Algorithm Hit-and-Prune(k)

2.4 FVS in Random Graphs

We consider the feedback vertex set problem for random graph $G_{n,p}$. We show that a simple augmenting approach based on ordering cycles according to a breadth-first search (Algorithm Augment-BFS described in the Section 2.3.2) has a strong performance guarantee. We show that it finds a FVS of size at most $n - (1/p) \log(np)(1 - o(1))$ with probability at least $3/4$, when $p = o(1)$.

We complement our upper bound with a lower bound on the feedback vertex set for $G_{n,p}$ obtained using simple union bound arguments. We show that if $p < 1/2$, then every subgraph induced by any subset of $r = \frac{2}{p} \log(np)(1 + o(1)) + 1$ vertices in $G_{n,p}$ contains a cycle with high probability. This gives an upper bound of $r - 1$ on the maximum induced acyclic subgraph of $G_{n,p}$. So, the size of the minimum FVS for $G_{n,p}$ is at least $n - r + 1 = n - (2/p) \log np$.

In contrast to de la Vega's result, our main contribution to the FVS problem in random graphs is in showing that a simple breadth-first ordering of the cycles is sufficient to find a nearly optimal feedback vertex set. We also note that our algorithm is an online algorithm with good performance guarantee when the cycles are revealed according to a breadth-first ordering. Improving on the size of the FVS returned by our algorithm appears to require making progress on the long-standing open problem of finding an independent set of size $((1 + \epsilon)/p) \log np$ in $G_{n,p}$. Assuming an optimal algorithm for this problem leads to an asymptotically optimal guarantee matching de la Vega's existential bound.

Next, we turn our attention to the directed random graph $D_{n,p}$. The algorithm for undirected graphs can be applied to give a nearly optimal feedback vertex set for $D_{n,p}$. The undirected graph G_D obtained by ignoring the orientation of the edges in $D_{n,p}$ is the random graph $G(n, 2p)$. Moreover, a feedback vertex set in G_D is also a feedback vertex set in $D_{n,p}$. Therefore, by ignoring the orientation of the arcs, the Augment-BFS algorithm used for undirected graphs can be used to obtain a feedback vertex set of size at most $n - (1/2p) \log(2np)$ with probability at least $3/4$. By Theorem 2.2.1, we see that the algorithm is nearly optimal for directed random graphs.

Organization. In Section 2.4.1 we present a variant of algorithm Augment-BFS that would be helpful for the purpose of analysis. In Section 2.4.2, we give an overview of the analysis, prove properties about our algorithm and show an upper bound on the size of the FVS found by the algorithm. Finally in Section 2.4.3, we prove a lower bound on the size of FVS in random graphs thereby showing near-optimality of our algorithm.

2.4.1 Grow induced BFS

We say that a vertex v is a *unique* neighbor of a subset of vertices L if and only if v is adjacent to exactly one vertex in L .

In Algorithm Augment-BFS, we obtain induced cycles in BFS order having deleted the vertices from the current candidate FVS S . We refine the candidate FVS S precisely as follows to obtain an induced BFS tree with unit increase in height: Consider the set $c(S)$ of cycles obtained by one-step BFS exploration from the set of vertices at current depth. Let K denote the set of unexplored vertices in the cycles in $c(S)$ (K is a subset of the vertices obtained by one-step BFS exploration from the set of vertices at current depth). Among the vertices in K include all non-unique neighbors of the set of vertices at current depth into S . Find a large independent set in the subgraph induced by the unique neighbors $R \subseteq K$ of the set of vertices at current depth. Include all vertices in R that are not in the independent set into S . This iterative refinement process is a natural adaptation of the idea behind the generic algorithm to the feedback vertex set problem where one collects a subset of cycles to find a hitting set H for these cycles and proposes H as the candidate set to obtain more cycles that have not been hit.

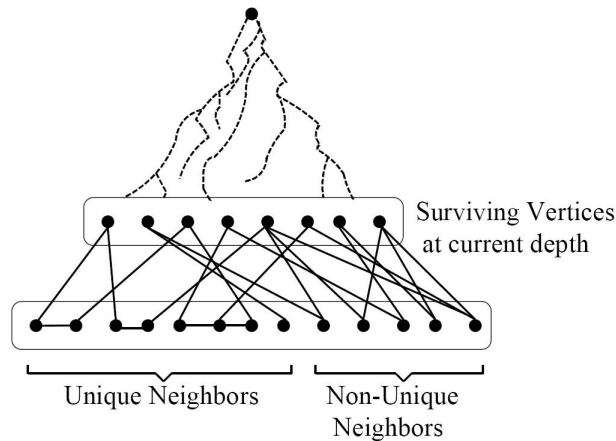


Figure 4: BFS Exploration

Essentially, the algorithm maintains an induced BFS tree by deleting vertices to remove cycles. The set of deleted vertices form a FVS. Consequently at each level of the BFS exploration, one would prefer to add as many vertices from the next level K as possible

maintaining the acyclic property. One way to do this is as follows: Delete all the non-unique neighbors of the current level from K thus hitting all cycles across the current and next level. There could still be cycles using an edge through the unique neighbors. To hit these, add a large independent set from the subgraph induced by the unique neighbors and delete the rest. Observe that this induced subgraph is a random graph on a smaller number of vertices. However, even for random graphs, it is open to find the largest independent set efficiently and only a factor 2 approximation is known.

In our analysis, instead of using the two approximate algorithm for the independent set problem, we use the simple heuristic of deleting a vertex for each edge that is present in the subgraph to find an independent set at each level. In order to lower bound the size of the induced tree, it suffices to consider growing the BFS-tree up to a certain height T using this heuristic and then using the 2-approximate algorithm for independent set at height T to terminate the algorithm. The size of the induced tree obtained using Algorithm Augment-BFS is at least as large as the one produced by the process just described. To simplify our analysis, it will be useful to restate the algorithm as Algorithm Grow-induced-BFS (Figure 5).

We remark that improving the approximation factor of the largest independent set problem in $G_{n,p}$ would also improve the size of the FVS produced. Our analysis shows that most of the vertices in the induced BFS tree get added at depth T as an independent set. Moreover, the size of this independent set is close to $(2/p) \log np(1 - o(1))$. Consequently, any improvement on the approximation factor of the largest independent set problem in $G_{n,p}$ would also lead to improving the size of the independent set found at depth T . This would increase the number of vertices in the induced BFS tree and thereby reduce the number of vertices in the feedback vertex set.

2.4.2 Analysis

In this section, we analyze Algorithm Grow-induced-BFS to find the size of the FVS that it returns. For $i = 0, \dots, T$, let L_i be the set of surviving vertices at level i with $l_i := |L_i|$, R_{i+1} be the set of *unique* neighbors of L_i with $r_{i+1} := |R_{i+1}|$, and U_i be the set of

1. Start from an arbitrary vertex v at level 0, set $L_0 = \{v\}$. Mark v as exposed. Fix $c := np$.
2. Explore levels $i = 0, \dots, T-1$, where $T = \left\lceil \frac{\ln(1/16p) - \ln \ln(1/16p)}{\ln(c+20\sqrt{c})} \right\rceil$ in BFS order as follows:
 - (a) Let K_{i+1} be the subset of neighbors of L_i among the unexposed vertices, where L_i is the set of surviving vertices at level i .
 - (b) Mark the vertices in K_{i+1} as exposed.
 - (c) Let $R_{i+1} \subseteq K_{i+1}$ be the subset of vertices in K_{i+1} that are **unique** neighbors of L_i .
 - (d) For every edge (u, v) that is present between vertices $u, v \in R_{i+1}$, add either u or v to W_{i+1} .
 - (e) Set $L_{i+1} = R_{i+1} \setminus W_{i+1}$.
(The set of surviving vertices at level $i+1$, namely L_{i+1} is an independent set in the subgraph induced by R_{i+1} .)
3. On obtaining L_{T-1} , set $R_T =$ **unique** neighbors of L_T among the unexposed vertices. In the subgraph induced by R_T , find an independent set L_T as follows.
 - (a) Fix an arbitrary ordering of the vertices of R_T . Repeat while $R_T \neq \emptyset$:
 - Add the next vertex $v \in R_T$ to L_T . Let $N(v) =$ neighbors of v in R_T . Set $R_T \leftarrow R_T \setminus N(v)$.
4. Return $S = V \setminus \cup_{i=0}^T L_i$ as the feedback vertex set.

Figure 5: Algorithm Grow-induced-BFS

unexposed vertices of the graph after i levels of BFS exploration with $u_i := |U_i|$. Observe that $U_i := V \setminus (L_0 \cup_{j=1}^i K_j)$.

We will use the following Chernoff bound for the concentration of binomial distribution.

Lemma 2.4.1 (Chernoff). *Let $X = \sum_{i=1}^n X_i$ where X_i are i.i.d. Bernoulli random variables with $\Pr(X_i = 1) = p$. Then*

$$\Pr(|X - np| \geq a\sqrt{np}) \leq 2e^{-a^2/2}.$$

2.4.2.1 Large Set of Unique Neighbors

The following lemma gives a concentration of the number of surviving vertices, unexposed vertices and unique neighbors to survivors at a particular level. It shows that upon exploring t levels according to the algorithm, the number of surviving vertices at the t -th level, l_t ,

is not too small while the number of unexposed vertices, u_t , is large. This also helps in proving a lower bound on the number of unique neighbors r_{t+1} to a level of survivors.

Lemma 2.4.2. *Consider Algorithm Grow-induced-BFS to be run on random graph $G_{n,p}$, where $c = np$ and T is the largest integer that satisfies $16Tp(c + 20\sqrt{c})^{T-1} \leq 1/2$. Then, with probability at least $3/4$, $\forall t \in \{0, 1, \dots, T-1\}$,*

$$\begin{aligned} u_t &\leq \left(n - \frac{1}{4} \sum_{i=0}^t (c - 20\sqrt{c})^i \right) \left(1 + \sqrt{\frac{\ln \ln n}{n}} \right), \\ u_t &\geq \left(n - \sum_{i=0}^t (c + 20\sqrt{c})^i \right) \left(1 - \sqrt{\frac{\ln \ln n}{n}} \right), \end{aligned} \quad (1)$$

$$\begin{aligned} l_t &\leq (c + 20\sqrt{c})^t, \\ l_t &\geq (c - 20\sqrt{c})^t (1 - 16Tp(c + 20\sqrt{c})^t) \left(1 - \frac{\sum_{i=0}^t (c + 20\sqrt{c})^i}{n} \right), \end{aligned} \quad (2)$$

$$\begin{aligned} r_t &\leq (c + 20\sqrt{c})^{t+1} \left(1 + \sqrt{\frac{\ln \ln n}{n}} \right), \\ r_t &\geq \frac{(c - 20\sqrt{c})^{t+1}}{4} \left(1 - \frac{\sum_{i=0}^{t+1} (c + 20\sqrt{c})^i}{n} \right) \left(1 - \sqrt{\frac{\ln \ln n}{n}} \right). \end{aligned} \quad (3)$$

The rest of this section will be devoted to prove this Lemma.

Proof. We prove the lemma by induction on t . We will prove the stronger induction hypothesis that every l_i, u_i for $i \in \{0, 1, \dots, t\}$ satisfy their respective concentration bounds with probability at least

$$a_t := 1 - \frac{t}{16T} - \frac{1}{16} \sum_{i=1}^t \frac{1}{i^2}.$$

We will prove the concentration of r_{i+1} as a consequence of l_i and u_i satisfying their respective concentration bounds. We will in fact show that the failure probability of r_{i+1} satisfying its concentration bound conditioned on l_i and u_i satisfying their respective concentration bounds will be at most $1/(32(i+1)^2)$. It immediately follows that with failure probability at most $(t/16T) + (3/32) \sum_{i=1}^t (1/i^2) + (1/32)(t+1)^2 \leq 1/4$, every r_{i+1}, u_i and l_i , for $i \in \{0, 1, \dots, t\}$ satisfies its respective concentration bound leading to the conclusion of the lemma.

For the base case, consider $t = 0$. It is clear that $u_0 = n - 1$ and $l_0 = 1$ satisfy the concentration bounds with probability 1. For the induction step, the induction hypothesis

is the following: With probability at least a_t , the concentration bounds are satisfied for u_i and l_i for every $i \in \{0, 1, \dots, t\}$. We will bound the probability that u_{t+1} or l_{t+1} fails to satisfy its corresponding concentration bound conditioned on the event that u_i, l_i for $i \in \{0, 1, \dots, t\}$ satisfy their respective concentration bounds.

1. To prove the concentration bound for u_{t+1} , observe that u_{t+1} is a binomial distribution with u_t trials and success probability $(1-p)^{l_t}$. Indeed, u_{t+1} is the number of vertices among U_t which are not neighbors of vertices in L_t . For each vertex $x \in U_t$, $\Pr(x \text{ has no neighbor in } L_t) = (1-p)^{l_t}$.

Therefore, by Lemma 2.4.1, we have that

$$\Pr\left(|u_{t+1} - u_t(1-p)^{l_t}| > \gamma_{t+1}\sqrt{u_t(1-p)^{l_t}}\right) \leq 2e^{-\gamma_{t+1}^2/2} = \frac{1}{32(t+1)^2}$$

with $\gamma_{t+1} = \sqrt{4 \ln 8(t+1)}$. Hence, with probability at least $1 - (1/32(t+1)^2)$,

$$u_t(1-p)^{l_t} \left(1 - \sqrt{\frac{4 \ln 8(t+1)}{u_t(1-p)^{l_t}}}\right) \leq u_{t+1} \leq u_t(1-p)^{l_t} \left(1 + \sqrt{\frac{4 \ln 8(t+1)}{u_t(1-p)^{l_t}}}\right).$$

Now, using the inductive bounds on u_t and l_t ,

$$\frac{4 \ln 8(t+1)}{u_t(1-p)^{l_t}} \leq \frac{10 \ln \ln n}{n}$$

since $t+1 \leq T \leq \ln n$,

$$\begin{aligned} (n - \sum_{i=0}^t (c + 20\sqrt{c})^i) &\geq \frac{15n}{16}, \\ (1 - p(c + 20\sqrt{c})^t) &\geq \frac{15}{16} \quad \text{and} \\ \left(1 - \sqrt{\frac{\ln \ln n}{n}}\right) &\geq \frac{1}{2}. \end{aligned}$$

Hence,

$$u_t(1-p)^{l_t} \left(1 - \sqrt{\frac{\ln \ln n}{n}}\right) \leq u_{t+1} \leq u_t(1-p)^{l_t} \left(1 + \sqrt{\frac{\ln \ln n}{n}}\right) \quad (4)$$

Therefore, by inequality (4) and the inductive bounds on u_t and l_t ,

$$\begin{aligned}
u_{t+1} &\geq \left(n - \sum_{i=0}^t (c + 20\sqrt{c})^i\right) \left(1 - \frac{c(c + 20\sqrt{c})^t}{n}\right) \left(1 - \sqrt{\frac{\ln \ln n}{n}}\right) \\
&\geq \left(n - \sum_{i=0}^t (c + 20\sqrt{c})^i - (c + 20\sqrt{c})^{t+1} + \frac{(c + 20\sqrt{c})^{t+1}}{n} \sum_{i=0}^t (c + 20\sqrt{c})^i\right) \\
&\quad \times \left(1 - \sqrt{\frac{\ln \ln n}{n}}\right) \\
&\geq \left(n - \sum_{i=0}^{t+1} (c + 20\sqrt{c})^i\right) \left(1 - \sqrt{\frac{\ln \ln n}{n}}\right)
\end{aligned}$$

which proves the lower bound. The upper bound is obtained by proceeding similarly:

$$\begin{aligned}
u_{t+1} &\leq u_t \left(1 - \frac{c(c - 20\sqrt{c})^t}{n} (1 - 16Tp(c + 20\sqrt{c})^t) \left(1 - \frac{\sum_{i=0}^t (c + 20\sqrt{c})^i}{n}\right)\right) \\
&\quad \times \left(1 + \sqrt{\frac{\ln \ln n}{n}}\right) \\
&\leq u_t \left(1 - \frac{c(c - 20\sqrt{c})^t}{4n}\right) \left(1 + \sqrt{\frac{\ln \ln n}{n}}\right) \\
&\leq \left(n - \frac{\sum_{i=0}^{t+1} (c - 20\sqrt{c})^i}{4n}\right) \left(1 + \sqrt{\frac{\ln \ln n}{n}}\right).
\end{aligned}$$

Thus, u_{t+1} satisfies the concentration bound with failure probability at most $1/(32(t+1)^2)$ conditioned on the event that u_i, l_i for $i \in \{0, 1, \dots, t\}$ satisfy their respective concentration bounds.

2. Next we address the failure probability of r_{t+1} not satisfying its concentration bound conditioned on the event that u_i, l_i for $i \in \{0, 1, \dots, t\}$ satisfy their respective concentration bounds. Claim 2.4.3 proves that the number of *unique* neighbors r_{t+1} is concentrated around its expectation.

Claim 2.4.3. *Let $q_t := pl_t(1-p)^{l_t-1}$. With probability at least $1 - (1/32(t+1)^2)$*

$$q_t u_t \left(1 + \frac{20}{\sqrt{c}}\right) \geq r_{t+1} \geq q_t u_t \left(1 - \frac{20}{\sqrt{c}}\right)$$

when $t+1 \leq T$.

Proof. Observe that r_{t+1} is a binomially distributed random variable with u_t trials and success probability q_t . Indeed, r_{t+1} is the number of vertices among U_t which are adjacent

to exactly one vertex in L_t . For each $u \in U_t$,

$$\Pr(u \text{ is adjacent to exactly one vertex in } L_t) = pl_t(1-p)^{l_t-1} = q_t.$$

Using $\beta_{t+1} = \sqrt{4 \ln 8(t+1)}$, by Lemma 2.4.1, we have that

$$\Pr(|r_{t+1} - q_t u_t| > \beta_{t+1} \sqrt{q_t u_t}) \leq 2e^{-\beta_{t+1}^2/2} = \frac{1}{32(t+1)^2}.$$

Hence, with probability at least $1 - (1/32(t+1)^2)$,

$$q_t u_t \left(1 - \sqrt{\frac{4 \ln 8(t+1)}{q_t u_t}}\right) \leq r_{t+1} \leq q_t u_t \left(1 + \sqrt{\frac{4 \ln 8(t+1)}{q_t u_t}}\right) \quad (5)$$

Claim 2.4.4 proves the concentration of the expected number of unique neighbors of L_t conditioned on the event that u_i, l_i for $i \in \{0, 1, \dots, t\}$ satisfy their respective concentration bounds. This in turn helps in proving that r_{t+1} is concentrated.

Claim 2.4.4. *For $t+1 \leq T$, if u_t and l_t satisfy their respective concentration bounds, then*

1. $q_t u_t \leq c(c + 20\sqrt{c})^t \left(1 + \sqrt{\frac{\ln \ln n}{n}}\right)$,
2. $q_t u_t \geq \frac{c(c-20\sqrt{c})^t}{4} \left(1 - \frac{\sum_{i=0}^{t+1} (c+20\sqrt{c})^i}{n}\right) \left(1 - \sqrt{\frac{\ln \ln n}{n}}\right)$.

Proof. Recall that $q_t = pl_t(1-p)^{l_t-1}$. Hence,

$$\begin{aligned} q_t u_t &\geq p \left(n - \sum_{i=0}^t (c + 20\sqrt{c})^i\right) l_t (1-p)^{l_t-1} \left(1 - \sqrt{\frac{\ln \ln n}{n}}\right) \\ &\geq c(c - 20\sqrt{c})^t \left(1 - \frac{\sum_{i=0}^t (c + 20\sqrt{c})^i}{n}\right)^2 \\ &\quad \times (1 - 16Tp(c + 20\sqrt{c})^t)(1 - p(c + 20\sqrt{c})^t) \left(1 - \sqrt{\frac{\ln \ln n}{n}}\right) \end{aligned}$$

(By the inductive bound on l_t)

$$\geq \frac{c(c - 20\sqrt{c})^t}{4} \left(1 - \frac{\sum_{i=0}^{t+1} (c + 20\sqrt{c})^i}{n}\right) \left(1 - \sqrt{\frac{\ln \ln n}{n}}\right)$$

using Claim 2.4.6 and

$$\begin{aligned} (1 - 16Tp(c + 20\sqrt{c})^t) &\geq \frac{1}{2}, \\ (1 - p(c + 20\sqrt{c})^t) &\geq \frac{1}{2} \quad \text{when } t+1 \leq T. \end{aligned}$$

For the upper bound, we proceed similarly and obtain

$$q_t u_t \leq c(c + 20\sqrt{c})^t \left(1 + \sqrt{\frac{\ln \ln n}{n}} \right).$$

□

Consequently, using Claim 2.4.4,

$$\frac{4 \ln 8(t+1)}{q_t u_t} \leq \frac{400}{c}$$

since, when $t+1 \leq T$,

$$\begin{aligned} \left(1 - \frac{\sum_{i=0}^{t+1} (c + 20\sqrt{c})^i}{n} \right) &\geq \left(\frac{15}{16} \right)^2, \\ \left(1 - \sqrt{\frac{\ln \ln n}{n}} \right) &\geq \frac{1}{2} \quad \text{and} \\ \frac{1}{2} &\geq \frac{4 \ln 8(t+1)}{(c - 20\sqrt{c})^t}. \end{aligned}$$

Hence, by inequality (2), with probability at least $1 - (1/32(t+1)^2)$,

$$q_t u_t \left(1 - \frac{20}{\sqrt{c}} \right) \leq r_{t+1} \leq q_t u_t \left(1 + \frac{20}{\sqrt{c}} \right) \quad (6)$$

when $t+1 \leq T$. □

Claims 2.4.3 and 2.4.4 together show that r_{t+1} satisfies the concentration bounds with failure probability at most $(1/32(t+1)^2)$ conditioned on the event that u_t and l_t satisfy their respective concentration bounds.

3. Finally we address the failure probability of l_{t+1} satisfying its concentration bound conditioned on the event that u_i, l_i for $i \in \{0, 1, \dots, t\}$ satisfy their respective concentration bounds. By Step 2(e) of the algorithm, the number of surviving vertices in level $t+1$ is $l_{t+1} := r_{t+1} - m_{t+1}$, where m_{t+1} denotes the number of edges among the vertices in R_{t+1} . In claim 2.4.3, we showed that the number of **unique** neighbors r_{t+1} is concentrated around its expectation. Claim 2.4.5 proves a concentration which bounds the number of edges among the vertices in R_t . These two bounds will immediately lead to the induction step on l_{t+1} . Thus, the probability that l_{t+1} does not satisfy its concentration bound will at most be the probability that either m_{t+1} or r_{t+1} does not satisfy its respective concentration bound.

Claim 2.4.5. $m_{t+1} \leq 8Tr_{t+1}^2p$ with probability at least $1 - (1/16T)$.

Proof. Recall that m_{t+1} denotes the number of edges among the vertices in R_{t+1} . Since the algorithm has not explored the edges among the vertices in R_{t+1} , m_{t+1} is a random variable following the Binomial distribution with $\binom{r_{t+1}}{2}$ trials and success probability p . By Markov's inequality, we have that for $t + 1 \leq T$,

$$\Pr(m_{t+1} \geq 8Tr_{t+1}^2p) \leq \frac{1}{16T}.$$

Hence, $m_{t+1} \leq 8Tr_{t+1}^2p$ with probability at least $1 - (1/16T)$, . □

We recall that $l_{t+1} = r_{t+1} - m_{t+1}$. The upper bound of the induction step follows using Claim 2.4.4:

$$l_{t+1} \leq r_{t+1} \leq c(c + 20\sqrt{c})^t \left(1 + \sqrt{\frac{\ln \ln n}{n}}\right) \left(1 + \frac{20}{\sqrt{c}}\right) \leq (c + 20\sqrt{c})^{t+1} \left(1 + \sqrt{\frac{\ln \ln n}{n}}\right).$$

For the lower bound, we use Claims 2.4.3 and 2.4.5 conditioned on the event that l_t and u_t satisfy their respective concentration bounds. With failure probability at most

$$\frac{1}{32(t+1)^2} + \frac{1}{16T},$$

we have that

$$\begin{aligned} l_{t+1} &= r_{t+1} - m_{t+1} \geq r_{t+1} - 8Tr_{t+1}^2p = r_{t+1}(1 - 8Tr_{t+1}p) \\ &\geq q_t u_t \left(1 - \frac{20}{\sqrt{c}}\right) \left(1 - 8Tq_t u_t p \left(1 + \frac{20}{\sqrt{c}}\right)\right) \end{aligned}$$

by Claim 2.4.3. Substituting for $q_t = pl_t(1-p)^{l_t-1}$, we get

$$\begin{aligned} l_{t+1} &\geq l_t p (1-p)^{l_t-1} u_t \left(1 - 8Tl_t p^2 (1-p)^{l_t-1} u_t \left(1 + \frac{20}{\sqrt{c}}\right)\right) \left(1 - \frac{20}{\sqrt{c}}\right) \\ &\geq l_t p \left(1 - \frac{20}{\sqrt{c}}\right) (1 - l_t p) (1 - 12Tl_t p^2 (1-p)^{l_t-1} u_t) \\ &\geq l_t p \left(1 - \frac{20}{\sqrt{c}}\right) (1 - l_t p) (1 - 12Tnp^2 l_t (1-p)^{l_t-1}) \\ &\geq c(c - 20\sqrt{c})^t \left(1 - \frac{\sum_{i=0}^t (c + 20\sqrt{c})^i}{n}\right)^2 (1 - 16Tp(c + 20\sqrt{c})^t) \\ &\quad \times (1 - (c + 20\sqrt{c})^t p(1 + 12Tc)) \left(1 - \frac{20}{\sqrt{c}}\right) \end{aligned}$$

using the inductive bound on u_t, l_t . Now, Claim 2.4.6 completes the induction step for the lower bound on l_{t+1} .

Thus, l_{t+1} satisfies the concentration bounds with failure probability at most $(1/32(t+1)^2) + (1/16T)$ conditioned on the event that u_i, l_i for $i \in \{0, 1, \dots, t\}$ satisfy their respective concentration bounds.

Finally, by the union bound, with probability at most $\frac{1}{32(t+1)^2} + \frac{1}{32(t+1)^2} + \frac{1}{16T}$, either u_{t+1} or l_{t+1} does not satisfy its respective concentration bounds conditioned on the event that u_i, l_i for $i \in \{0, 1, \dots, t\}$ satisfy their respective concentration bounds. By induction hypothesis, the failure probability of some u_i, l_i for $i \in \{0, 1, \dots, t\}$ not satisfying their respective concentration bound is at most $1 - a_t$. Hence, the probability that u_i, l_i satisfy their respective concentration bound for every $i \in \{0, 1, \dots, t+1\}$ is at least $a_t(1 - (1/16(t+1)^2) - (1/16T)) \geq a_{t+1}$. Therefore, with probability at least a_{t+1} , every u_i, l_i for $i \in \{0, 1, \dots, t+1\}$ satisfy their respective concentration bounds. This proves the stronger induction hypothesis.

To complete the proof of Lemma 2.4.2, recollect that we showed that the failure probability of r_{i+1} satisfying its concentration bound conditioned on l_i and u_i satisfying their respective concentration bounds is at most $1/(32(i+1)^2)$. By the union bound argument, it immediately follows that with failure probability at most $(t/16T) + (3/32) \sum_{i=1}^t (1/i^2) + (1/32(t+1)^2) \leq 1/4$, every r_{i+1}, u_i and l_i , for $i \in \{0, 1, \dots, t\}$ satisfies its respective concentration bound leading to the conclusion of the lemma. \square

Claim 2.4.6. For $t+1 \leq T$,

1.

$$1 - \frac{\sum_{i=0}^{t+1} (c + 20\sqrt{c})^i}{n} \leq \left(1 - \frac{\sum_{i=0}^t (c + 20\sqrt{c})^i}{n} \right)^2,$$

2.

$$(1 - 16Tp(c + 20\sqrt{c})^{t+1}) \leq (1 - 16Tp(c + 20\sqrt{c})^t) (1 - (c + 20\sqrt{c})^t p(1 + 12Tc)).$$

Proof of Claim 2.4.6. We prove the first part of the Claim by induction. For the base case,

we need to prove that

$$1 + \frac{1}{n^2} - \frac{2}{n} \geq 1 - \frac{c + 20\sqrt{c}}{n} - \frac{1}{n}$$

i.e., $n - 1 \leq (c + 20\sqrt{c})n$

which is true. For the induction step, we need to prove that

$$1 - \frac{\sum_{i=0}^{t+2} (c + 20\sqrt{c})^i}{n} \leq \left(1 - \frac{\sum_{i=0}^t (c + 20\sqrt{c})^i}{n} - \frac{(c + 20\sqrt{c})^{t+1}}{n} \right)^2$$

Now, RHS

$$\begin{aligned} &= \left(1 - \frac{\sum_{i=0}^t (c + 20\sqrt{c})^i}{n} \right)^2 + \frac{(c + 20\sqrt{c})^{2t+2}}{n^2} \\ &\quad - \frac{2(c + 20\sqrt{c})^{t+1}}{n} \left(1 - \frac{\sum_{i=0}^t (c + 20\sqrt{c})^i}{n} \right) \\ &\geq 1 - \frac{\sum_{i=0}^{t+1} (c + 20\sqrt{c})^i}{n} + \frac{(c + 20\sqrt{c})^{2t+2}}{n^2} \\ &\quad - \frac{2(c + 20\sqrt{c})^{t+1}}{n} + \frac{2(c + 20\sqrt{c})^{t+1} \sum_{i=0}^t (c + 20\sqrt{c})^i}{n^2}. \end{aligned}$$

Hence, it is sufficient to prove that

$$\begin{aligned} -\frac{(c + 20\sqrt{c})^{t+2}}{n} &\leq \frac{(c + 20\sqrt{c})^{2t+2}}{n^2} - \frac{2(c + 20\sqrt{c})^{t+1}}{n} + \frac{2(c + 20\sqrt{c})^{t+1} \sum_{i=0}^t (c + 20\sqrt{c})^i}{n^2} \\ \text{i.e., } (c + 20\sqrt{c}) &\geq 2 - \frac{(c + 20\sqrt{c})^{t+1}}{n} - 2 \frac{\sum_{i=0}^t (c + 20\sqrt{c})^i}{n}, \end{aligned}$$

which is true for large enough c when $t + 1 \leq T$.

For the second part of the Claim, we need to prove that

$$(1 - 16Tp(c + 20\sqrt{c})^t) (1 - (c + 20\sqrt{c})^t p(1 + 12Tc)) \geq (1 - 16Tp(c + 20\sqrt{c})^{t+1})$$

$$\text{i.e., } 1 - 16Tp(c + 20\sqrt{c})^t - (c + 20\sqrt{c})^t p(1 + 12Tc) + 18Tp^2(c + 20\sqrt{c})^{2t}(1 + 12Tc)$$

$$\geq 1 - 16Tp(c + 20\sqrt{c})^{t+1}$$

$$\text{i.e., } (1 - 16Tp(c + 20\sqrt{c})^t)(1 + 12Tc) \leq 16T(c + 20\sqrt{c} - 1)$$

which is true since $1 + 12Tc \leq 16T(c + 20\sqrt{c} - 1)$ for large c and the rest of the terms are less than 1 when $t + 1 \leq T$. □

2.4.2.2 Upper Bound

We will need the following theorem due to Frieze [32] in this analysis.

Theorem 2.4.7 (Frieze [32]). *Let $d = np$ and $\epsilon > 0$ be fixed. Suppose $d_\epsilon \leq d = o(n)$ for some sufficiently large fixed constant d_ϵ . Then, almost surely, the size of the independent set in $G_{n,p}$ is at least*

$$\left(\frac{2}{p}\right) (\log np - \log \log np - \log 2 + 1 - 0.5\epsilon).$$

Now, we are ready to analyze the performance of Algorithm Augment-BFS.

Theorem 2.4.8. *For $G_{n,p}$, such that $p = o(1)$, algorithm Augment-BFS runs in polynomial time and produces a feedback vertex set of size at most $n - (1/p) \log(np)(1 - o(1))$ with probability at least $3/4$.*

Proof. The polynomial running time of the algorithm is straightforward. We will bound the size of the FVS returned by the algorithm. For this, we use the fact that the size of the surviving set of vertices is large when the algorithm has explored $T - 1$ levels. Moreover, the number of unexposed vertices is also large. Thus, there is a large independent set among the *unique* neighbors of the surviving vertices. This set along with the surviving vertices up to level $T - 1$ will form a large induced tree. We will now prove that the size of the independent set among the *unique* neighbors of L_{T-1} is large.

By Theorem 2.4.7, if $r_T p > d_\epsilon$ for some constant d_ϵ and $r_T p = o(r_T)$, then there exists an independent set of size $(2/p) \log(r_T p)(1 - o(1))$. It suffices to prove that r_T is large and is such that $r_T p > d_\epsilon$.

We note that the choice of $T = \left\lceil \frac{\ln(1/16p) - \ln \ln(1/16p)}{\ln(c+20\sqrt{c})} \right\rceil$ used in the algorithm satisfies the hypothesis of Lemma 2.4.2. Therefore, using Lemma 2.4.2, with probability at least $3/4$, we have

$$\begin{aligned} r_T &\geq \frac{(c - 20\sqrt{c})^T}{4} \left(1 - \frac{\sum_{i=0}^T (c + 20\sqrt{c})^i}{n}\right) \left(1 - \sqrt{\frac{\ln \ln n}{n}}\right) \\ &\geq \frac{(c - 20\sqrt{c})}{64p} \left(1 - \frac{\sum_{i=0}^T (c + 20\sqrt{c})^i}{n}\right) \left(1 - \sqrt{\frac{\ln \ln n}{n}}\right) \\ &\geq \frac{c - 20\sqrt{c}}{2^8 p} \geq \frac{d_\epsilon}{p} \end{aligned}$$

for sufficiently large c since

$$\left(1 - \frac{\sum_{i=0}^T (c + 20\sqrt{c})^i}{n}\right) \left(1 - \sqrt{\frac{\ln \ln n}{n}}\right) \geq \frac{15}{16} \cdot \frac{1}{2}.$$

Consequently, by Theorem 2.4.7, there exists an independent set of size at least

$$(2/p) \log(r_{TP})(1 - o(1)).$$

Moreover, step 3 of the algorithm finds a 2-approximate independent set (see [37, 58]).

Therefore, the size of the independent set found in step 3 is at least $(1/p) \log r_{TP}(1 - o(1))$, which is greater than

$$\left(\frac{1}{p}\right) \log(c)(1 - o(1)) = \left(\frac{1}{p}\right) \log(np)(1 - o(1)).$$

We note that this set gets added to the tree obtained by the algorithm which increases the number of vertices in the tree while maintaining the acyclic property of the induced subgraph. Hence, with probability at least $3/4$, the induced subgraph has $\sum_{i=0}^{T-1} l_i + (1/p) \log np(1 - o(1))$ vertices. Consequently, the FVS obtained has size at most $n - (1/p) \log np(1 - o(1))$ with probability at least $3/4$.

□

Using the same algorithm for directed random graphs by ignoring the orientation of edges, we have the following Corollary.

Theorem 2.4.9. *For $D_{n,p}$, there exists a polynomial time algorithm that produces a FVS of size at most $n - (1/2p)(\log(np) - o(1))$ with probability at least $3/4$.*

The near-optimality of the algorithm for directed random graphs follows from Spencer and Subramanian's lower bound (Theorem 2.2.1).

2.4.3 Lower Bound

In this section, we prove a lower bound for the Feedback Vertex Set in random graphs. We will need the following bound on the number of ways to partition a positive integer n into k positive integers.

Theorem 2.4.10 (Pribitkin [24]). *Let $p_k(n)$ denote the number of ways to partition n into exactly k parts. Then there exists an absolute constant $A < 1$ such that*

$$p_k(n) < A \frac{e^{c\sqrt{n-k}}}{(n-k)^{3/4}} e^{-\frac{2\sqrt{n-k}}{c}} L_2\left(e^{-\frac{c(k+1/2)}{2\sqrt{n-k}}}\right)$$

where $c = \pi\sqrt{2/3}$ and $L_2(x) = \sum_{m=1}^{\infty} \frac{x^m}{m^2}$ for $|x| \leq 1$.

Remark: Since we will not need such a tight bound, we will use $p_k(n) < C_1 e^{C_2(n-k)}$ for some constants $C_1, C_2 > 0$.

We consider the dual problem for minimum FVS - namely maximum induced acyclic subgraph.

Theorem 2.4.11. *Let $r = \frac{2}{p} \log(np)(1+o(1)) + 1$. If $p < 1/2$, then every subgraph induced by any subset of r vertices in $G_{n,p}$ contains a cycle with high probability.*

Proof. First we note that every induced subgraph on r vertices is a graph from the family $G(r, p)$. We bound the probability that a graph $H = G(r, p)$ is a forest.

$$\begin{aligned} \Pr(H \text{ is a forest}) &\leq \sum_{k=1}^r \sum_{n_1+\dots+n_k=r, n_i>0} \text{No. of forests with } k \text{ trees on } n_1, \dots, n_k \text{ vertices} \\ &\quad \times \Pr(\text{Forest on } k \text{ components}) \\ &= \sum_{k=1}^r \sum_{n_1+\dots+n_k=r, n_i>0} \left(\frac{r!}{\prod_{i=1}^k n_i!} \right) \left(\prod_{i=1}^k n_i^{n_i-2} \right) p^{r-k} (1-p)^{\binom{r}{2}-r+k} \\ &\leq r! (1-p)^{\binom{r}{2}} \sum_{k=1}^r \sum_{n_1+\dots+n_k=r, n_i>0} \left(\frac{p}{1-p} \right)^{r-k} \\ &\leq r! (1-p)^{\binom{r}{2}} \sum_{k=1}^r \sum_{n_1+\dots+n_k=r, n_i>0} (2p)^{r-k} \end{aligned}$$

assuming $p < \frac{1}{2}$. Therefore,

$$\begin{aligned} \Pr(H \text{ is a forest}) &\leq r! (1-p)^{\binom{r}{2}} \sum_{k=1}^r (2p)^{r-k} \sum_{n_1+\dots+n_k=r, n_i>0} 1 \\ &= r! (1-p)^{\binom{r}{2}} \sum_{k=1}^r (2p)^{r-k} p_k(r) \\ &\leq r! (1-p)^{\binom{r}{2}} \sum_{k=1}^r (2p)^{r-k} C_1 e^{C_2(r-k)} \end{aligned}$$

by the remark below Theorem 2.4.10. Consequently, there exists a constant C_3 such that

$$\begin{aligned} \Pr(H \text{ is a forest}) &\leq r!(1-p)^{\binom{r}{2}} \sum_{k=1}^r (C_3 p)^{r-k} \leq r!(1-p)^{\binom{r}{2}} (C_4 p)^r \leq (C_4 r p)^r (1-p)^{\frac{r^2}{2}} \\ &\leq (C_4 n p)^r (1-p)^{\frac{r^2}{2}} \leq e^{(r(\log(C_4 n p) - \frac{pr}{2}))} \end{aligned}$$

which tends to zero when $r > \frac{2}{p}(\log C_4 n p)$. \square

2.5 *Planted FVS*

It is evident from our results in Section 2.4 that the feedback vertex set of a random graph contains most of its vertices for $p = o(1)$. This motivates one to ask if a significantly smaller “planted” feedback vertex set in a random graph can be recovered in the implicit hitting set framework. This question is similar in flavor to the well-studied planted clique problem [40, 1, 31], but posed in the implicit hitting set framework.

We address this question, with a new model $D_{n,\delta,p}$ for the directed planted feedback vertex set problem (see Section 2.5.1). In this section, we prove that for graphs $D_{n,\delta,p}$, for large enough p , it is sufficient to hit cycles of small size to recover the planted feedback vertex set. For example, if $p \geq C_0/n^{1/3}$ for some absolute constant C_0 , then it is sufficient to find the best hitting set for triangles in $D_{n,\delta,p}$. This would be the planted feedback vertex set. More generally, if $p \geq C/n^{1-2/k}$ for some constants C, k , $0 < \delta \leq 9/19$, then the smallest hitting set for the set of cycles of size k in $D = D_{n,\delta,p}$ is the planted feedback vertex set P with high probability.

This also shows that the planted subset is indeed the smallest feedback vertex set in this model. Now, in order to recover the planted feedback vertex set, it is sufficient to obtain cycles in increasing order of their sizes and find the best hitting set for subset of cycles of size k . Moreover, by straightforward counting (see Lemma 2.5.5), the expected number of cycles of length k is at most $(nkp)^k = \text{poly}(n)$ for the mentioned range of p and constant k . Thus, we have a polynomial-sized collection \mathcal{T}' of cycles, such that the optimal hitting set for \mathcal{T}' is also the optimal hitting set for all cycles in $D_{n,\delta,p}$.

However, we note that finding the smallest hitting set even for triangles is NP-hard. Here, we use algorithm Hit-and-Prune to recover the planted feedback vertex set of slightly

smaller size in polynomial time.

Organization. We present the planted model in Section 2.5.1 and prove properties about this model in Section 2.5.2. We use these properties to show that the smallest FVS is the planted subset in Section 2.5.3. Finally, in Section 2.5.4, we give a pruning based algorithm to recover the planted subset proving Theorem 2.5.6.

2.5.1 Model

The planted directed random graph $D_{n,\delta,p}$ on n vertices for $0 < \delta \leq 1$ is obtained as follows: Choose δn vertices arbitrarily to be the planted subset P . Each pair (u, v) where $u \in P, v \in V$, is adjacent independently with probability $2p$ and the corresponding edge is oriented in one of the two directions $\{u \rightarrow v, v \rightarrow u\}$ in $D_{n,\delta,p}$ with equal probability. The arcs between vertices in $V \setminus P$ are obtained in the following manner to ensure that the subgraph induced on $V \setminus P$ is a directed acyclic graph: Pick an arbitrary permutation of the vertices in $V \setminus P$. With the vertices ordered according to this permutation, each forward arc is present with probability p independently; no backward arcs occur according to this ordering (see Figure 6). The goal is to recover the planted subset.

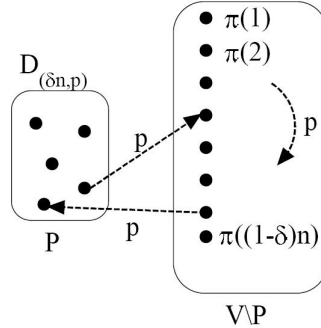


Figure 6: Planted Model

2.5.2 Properties of Planted Subset

The following property is formalized in Lemma 2.5.1: If $S \subseteq |V \setminus P|$ is a subset of vertices of size at least $(1 - \delta)n/10$, then with high probability, every vertex $u \in P$ induces a k -cycle with vertices in S . This will be crucial in showing that the smallest hitting set is indeed

the planted subset and also in showing that algorithm Hit-and-Prune recovers the planted subset.

Lemma 2.5.1. *Let $D_{n,\delta,p}$ be a planted directed random graph where $p \geq C/n^{1-2/k}$ for some constants C, k, δ . Then, with high probability, for every vertex $v \in P$, there exists a cycle of size k through v in the subgraph induced by $S \cup \{v\}$ in $D_{n,\delta,p}$ if S is a subset of $V \setminus P$ of size at least $|V \setminus P|/10 = (1 - \delta)n/10$.*

Proof. The proof is by second moment method. Let $S \subset V \setminus P$, $|S| \geq (1 - \delta)n/10$, $v \in P$. Let X_v denote the number of cycles of size k through v in the subgraph induced by $S \cup \{v\}$. Then, $\mathbb{E}(X_v) = \binom{(1-\delta)n/10}{k-1} p^k$. Using Chebyshev's inequality, we can derive that

$$\Pr(X_v = 0) \leq \frac{\text{Var}(X_v)}{\mathbb{E}(X_v)^2}.$$

To compute the variance of X_v , we write $X_v = \sum_{A \subseteq S: |A|=k-1} X_A$, where the random variable X_A is 1 when the vertices in A induce a cycle of length k with v and 0 otherwise.

$$\text{Var}(X_v) \leq \mathbb{E}(X_v) + \sum_{A, B \subseteq S: |A|=|B|=k-1, A \neq B} \text{Cov}(X_A, X_B)$$

Now, for any fixed subsets $A, B \subseteq S$, $|A| = |B| = k-1$ and $|A \cap B| = r$, $\text{Cov}(X_A, X_B) \leq p^{2k-r}$ and the number of such subsets is at most $\binom{|S|}{2k-2-r} \binom{k}{r} \leq \binom{n}{2k-2-r} \binom{k}{r}$. Therefore,

$$\begin{aligned} \sum_{r=0}^{k-2} \sum_{A, B \subseteq S: |A|=|B|=k-1, |A \cap B|=r} \frac{\text{Cov}(X_A, X_B)}{\mathbb{E}(X_v)^2} &\leq \sum_{r=0}^{k-2} \frac{\binom{k}{r} \binom{n}{2k-2-r} p^{2k-r}}{\left(\binom{(1-\delta)n/10}{k-1} p^k\right)^2} \\ &\leq \sum_{r=0}^{k-2} \frac{C_r}{(np)^r} \text{ for some constants } C_r = C_r(\delta) \\ &\rightarrow 0 \end{aligned}$$

as $n \rightarrow \infty$ if $p \geq C/n^{1-2/k}$ for some sufficiently large constant C since each term in the summation tends to 0 and the summation is over a finite number of terms. Thus

$$\Pr(X_v = 0) \leq \frac{1}{\left(\binom{(1-\delta)n/10}{k-1} p^k\right)} \leq \frac{1}{((1 - \delta)n/10)^{k-1} p^k}.$$

Therefore,

$$\Pr(X_v \geq 1) \geq 1 - \frac{1}{((1 - \delta)n/10)^{k-1} p^k}$$

and hence

$$\begin{aligned} \Pr(X_v \geq 1 \forall v \in P) &\geq \left(1 - \frac{1}{((1-\delta)n/10)^{k-1}p^k}\right)^{|P|} \\ &= \left(1 - \frac{1}{((1-\delta)n/10)^{k-1}p^k}\right)^{\delta n} \\ &\geq e^{-\frac{10^{k-1}\delta}{2(1-\delta)^{k-1}n^{k-2}p^k}} \rightarrow 1 \end{aligned}$$

as $n \rightarrow \infty$ if $p \geq C/n^{1-2/k}$ for some large constant C . \square

Lemma 2.5.1 leads to the following corollaries.

Corollary 2.5.2. *Let H be a hitting set for the k -cycles in $D_{n,\delta,p}$ where $p \geq C/n^{1-2/k}$ for some constants C, k, δ . If $|H| \leq t\delta n$ where $t \leq 9(1-\delta)/10\delta$, then $H \supseteq P$.*

Proof. Suppose $u \in P$ and $u \notin H$. Then H should contain at least $|V \setminus P| - |V \setminus P|/10$ vertices from $V \setminus P$, else by Lemma 2.5.1, there exists a k -cycle involving u and some $k-1$ vertices among the $|V \setminus P|/10$ vertices that H does not contain contradicting the fact that H hits all cycles of length k . Therefore, $|H| > |V \setminus P| - |V \setminus P|/10 = (1-\delta)9n/10 \geq t\delta n$ by the choice of t . Thus, the size of H is greater than $t\delta n$, a contradiction. \square

Corollary 2.5.3. *If a subset $H \subseteq V$ hits all cycles of length k in $D_{n,\delta,p}$, where $p \geq C/n^{1-2/k}$ for some constants C, k, δ , then $|H| \geq |P|$.*

Proof. If H contains all vertices in P , then we are done. Suppose not. Let $u \in P$ and $u \notin H$. Then H should contain at least $|V \setminus P| - |V \setminus P|/10$ vertices from $V \setminus P$, else by Lemma 2.5.1, there exists a k -cycle involving u and some $k-1$ vertices among the $|V \setminus P|/10$ vertices that H does not contain. This would contradict the fact that H hits all cycles of length k . Therefore, $|H| > |V \setminus P| - |V \setminus P|/10 = (1-\delta)9n/10 \geq \delta n = |P|$ since $\delta \leq 9/19$. \square

2.5.3 Hitting Set is Planted FVS

Corollary 2.5.2 states that every sufficiently small hitting set for the set of k -cycles in $D_{n,\delta,p}$ should contain every vertex from the planted feedback vertex set. Corollary 2.5.3 states that every subset that hits all cycles of length k in $D_{n,\delta,p}$ should be of size at least the size of the smallest feedback vertex set. Thus, the Corollaries indeed show that the planted set P is the smallest hitting set for all cycles in $D_{n,\delta,p}$.

Theorem 2.5.4. *Let $D = D_{n,\delta,p}$ with planted feedback vertex set P , where $p \geq C/n^{1-2/k}$ for some constants C, k , $0 < \delta \leq 9/19$. Then, the smallest hitting set for the set of cycles of size k in D is the planted feedback vertex set P with high probability.*

Proof. By Corollary 2.5.3, we know that if a subset $H \subseteq V$ hits all cycles of length k in $D_{n,\delta,p}$, then $|H| \geq |P|$. Therefore, every hitting set for the subset of k -cycles should be of size at least $|P| = \delta n$. Also, we know that P is a hitting set for the k -cycles since P is a feedback vertex set in $D_{n,\delta,p}$. Thus, the optimum hitting set for the k -cycles is of size exactly $|P|$.

Let H be the smallest hitting set for the k -cycles. Then $|H| = \delta n$. It is easily verified that $t = 1$ satisfies the conditions of Corollary 2.5.2 if $\delta \leq 9/19$. Therefore, $H \supseteq P$. Along with the fact that $H = \delta n = |P|$, we conclude that $H = P$. \square

2.5.4 Pruning Algorithm to recover Planted FVS

In this section, we give an algorithm to recover the planted feedback vertex set in $D_{n,\delta,p}$. Theorem 2.5.4 suggests an algorithm where one would obtain all cycles of length k and find the best hitting set for these set of cycles. Even though the number of k -cycles is polynomial, we do not have a procedure to find the best hitting set for k -cycles. However, by repeatedly taking all vertices of a cycle into the hitting set and removing them from the graph, we do have a simple greedy strategy that finds a k -approximate hitting set. This is the strategy used in Algorithm Hit-and-Prune(k) given in Figure 3.

The idea behind the application of Algorithm Hit-and-Prune(k) to recover the planted subset is the following: The set S obtained at the end of step 2 in the above algorithm is a k -approximate hitting set and hence is of size at most $k\delta n$. Using Corollary 2.5.2, it is clear that S contains P - indeed, if S does not contain all vertices in P , then S should contain most of the vertices in $V \setminus P$ contradicting the fact that the size of S is at most $k\delta n$. Further, owing to the choice of δ , it can be shown that S does not contain at least $|V \setminus P|/10$ vertices from $V \setminus P$. Therefore, by Lemma 2.5.1, every vertex $v \in P$ induces a k -cycle with some subset of vertices from $V \setminus S$. Also, since $V \setminus P$ is a DAG no vertex $v \in V \setminus P$ induces cycles with any subset of vertices from $V \setminus S \subseteq V \setminus P$. Consequently, a

vertex v induces a k -cycle with vertices in $V \setminus S$ if and only if $v \in P$. Thus, the vertices in P are identified exactly.

The following Lemma will exhibit that Algorithm Hit-and-Prune(k) examines a polynomial number of cycles.

Lemma 2.5.5. *The expected number of cycles of length k in $D_{n,\delta,p}$ is at most $(nkp)^k$.*

Proof. The proof is by straightforward counting. The expected number of cycles of length k is at most

$$\begin{aligned} \sum_{i=1}^k \binom{|P|}{i} \binom{|R|}{k-i} k! p^k &= \sum_{i=1}^k \binom{\delta n}{i} \binom{(1-\delta)n}{k-i} k! p^k \leq \sum_{i=1}^k (\delta n)^i ((1-\delta)n)^{k-i} (kp)^k \\ &= ((1-\delta)nkp)^k \sum_{i=1}^k \left(\frac{\delta}{1-\delta}\right)^i = ((1-\delta)nkp)^k (1-\delta) \leq (nkp)^k. \end{aligned}$$

□

Theorem 2.5.6. *For $D_{n,\delta,p}$ with planted FVS P , where $p \geq C/n^{1-2/k}$ for some constants $C, k \geq 3$ and $0 < \delta \leq 1/2k$, Algorithm Hit-and-Prune runs in polynomial time and recovers the planted FVS P with high probability.*

Proof. Since we are using the greedy strategy to obtain a hitting set S for \mathcal{T}' , it is clear the S is a k -approximate hitting set. Therefore $|S| \leq k\delta n$. It is easily verified that $t = k$ satisfies the conditions of Corollary 2.5.2 if $\delta \leq 1/2k$. Thus, all vertices from the planted feedback vertex set P are present in the subset S obtained at the end of step 2 in the algorithm.

By the choice of $\delta \leq 1/2k$, it is true that $|S| \leq k\delta n \leq 9(1-\delta)n/10 = 9|V \setminus P|/10$. Hence, $|V \setminus S| \geq |V \setminus P|/10$.

Since $S \supseteq P$, the subset of vertices $V \setminus S$ does not contain any vertices from the planted set. Also, the number of vertices in $V \setminus S$ is at least $|V \setminus P|/10$. Consequently, by Lemma 2.5.1, each vertex $v \in P$ induces at least one k -cycle with vertices in $V \setminus S$. Since $V \setminus P$ is a directed acyclic graph, none of the vertices $u \in V \setminus P$ induce cycles with vertices in $V \setminus S$. Therefore, a vertex $v \in S$ induces a k -cycle with vertices in $V \setminus S$ if and only if $v \in P$. Hence, the subset H output by Algorithm Hit-and-Prune(k) is exactly the planted feedback vertex set P .

By Lemma 2.5.5, the expected number of cycles seen by the algorithm to store \mathcal{T}' is polynomial. Thus, the algorithm uses polynomial sized storage memory. Finally, since the size of \mathcal{T}' is polynomial, steps 2 and 3 of the algorithm take polynomial time leading to polynomial running time of Algorithm Hit-and-Prune(k) in expectation. \square

2.6 Conclusion

Many well-known combinatorial problems can be reformulated as hitting set problems with an exponential number of subsets to be hit. For some of these problems, there exist efficient procedures to verify whether a candidate set is a hitting set and if not, output a subset that is not hit. The implicit hitting set framework encompasses such problems. The motivation behind introducing this framework lies in the possibility of efficient approximation algorithms for large hitting set instances. We seek algorithms that run in time bounded by a polynomial in the size of the ground set even though the number of subsets to hit is exponential. We initiated the study towards developing such algorithms by showing an algorithm for a combinatorial problem that falls in this framework — the feedback vertex set problem in random graphs. It would be interesting to extend our results to other implicit hitting set problems, such as the ones in Section 2.1 or the problems mentioned by Karp and Moreno-Centeno [44].

CHAPTER III

GEOMETRIC APPROACH: DISCREPANCY-BASED ALGORITHM FOR IP

In this chapter, we adopt a geometric approach to integer programming. We observe a seemingly unexplored connection between the integer feasibility of a polytope and the radius of the largest inscribed ball in the polytope. Here, we refine this geometric connection by studying the radius of the largest inscribed ball that guarantees integer feasibility as a function of the constraint matrix describing the polytope. We observe that this radius function is precisely the classical notion of linear discrepancy of the normalized constraint matrix (normalized so that each row is a unit vector).

We exploit this connection to show a phase-transition phenomenon in random IP instances. We also use a recent algorithm for low discrepancy solutions due to Lovett and Meka [56] to obtain an algorithm to find integer points in random polytopes that are integer feasible with high probability. The results in this chapter are joint work with Santosh Vempala [11].

3.1 Introduction

In this chapter, we give a new approach to IP via classical discrepancy theory. The approach is centered on the simple idea that a polytope is likely to contain an integer point (hence called *integer feasible*) if it contains a large ball. In fact, any polytope in n -dimensional space that contains a Euclidean ball of radius at least $\sqrt{n}/2$ is integer feasible. A natural attempt to refine this radius is to ask for the radius of the largest inscribed ball $r(A)$ as a function of the constraint matrix A describing the polytope. It is unclear whether such a radius function even exists.

3.1.1 Motivation

The main motivation for studying such a radius function comes from a natural model for random IP instances where the constraint matrix A describing a polytope is chosen from some distribution. While IP in its general form is intractable unless $P=NP$, several special instances are very interesting and not well-understood. One simple family of instances is randomly generated IP instances.

Random instances have been studied for several combinatorial problems e.g., random satisfiability [13, 12, 17, 8, 30], random knapsack [6] and various other graph problems on random graphs [7]. IP is one of the few problems in Karp’s original list [42] that has not been satisfactorily understood for random instances. Furst and Kannan studied the random single-row subset-sum IP [33]. Their results were generalized to multi-row IP by Pataki, Tural and Wong [62]. They showed that if each entry in the constraint matrix A is chosen independently and uniformly at random from the discrete set $\{1, 2, \dots, M\}$, then with high probability, a certain reformulation of such random IP instances can be solved very quickly by the branch-and-bound algorithm, if M is sufficiently large. Their requirement that M be larger than the length of the RHS vector b of the IP is very different from the flavor of the results in well-studied random graph and random satisfiability instances. This raises the question of what is the best model for random IP instances.

We believe that the study of various probabilistic models for IP would also reveal the best-suited model for *smoothed analysis* of IP. In their landmark paper, Spielman and Teng introduced smoothed analysis as a robust concept to explain the efficient behavior of the Simplex algorithm for LP in practice [68]. Smoothed analysis assumes that the input instances are specified by an adversary and then perturbed randomly. Intriguingly, they showed that the expected running time of the Simplex algorithm on the resulting instance decreases as the amount of randomness increases. In order to be able to algorithmically solve smoothed IP instances, it seems imperative to algorithmically solve the more fundamental problem of fully random IP instances.

3.1.2 Results

Our main conceptual contribution is a connection between the radius of the largest inscribed ball that guarantees integer feasibility and the linear discrepancy of the constraint matrix. We show that if the radius is at least the *linear discrepancy* of the normalized constraint matrix (each row is normalized to a unit vector), then the polytope contains an integer point.

Next, we exploit this connection to show a phase-transition phenomenon in a new probabilistic IP model. An IP instance $\{x : Ax \leq b\}$ in this model is described by a constraint matrix $A \in \mathbb{R}^{m \times n}$ whose m rows are independent uniform random unit vectors in \mathbb{R}^n . We recall that if each row of the constraint matrix A is a unit vector, then they describe the normals to the facets of the polytope $P = \{x : Ax \leq b\}$. Thus, our probabilistic IP instance is a random polytope in \mathbb{R}^n with m facets whose normal vectors are independent uniform random unit vectors. The radius of the largest ball contained in the polytope determines the right hand side vector b .

We show that for $m = 2^{O(\sqrt{n})}$, there exist constants $c_0 < c_1$ such that, with high probability, the random polytope is integer infeasible if the largest ball contained in the corresponding polytope is centered at $(1/2, \dots, 1/2)$ and has radius at most $c_0 \sqrt{\log(m/n)}$; it is integer feasible for every center if the radius is at least $c_1 \sqrt{\log(m/n)}$. Thus, random polytopes transition from having no integer points to being integer feasible within a constant factor increase in the radius of the largest inscribed ball. When $m = O(n)$, a constant radius ball inscribed in the random polytope guarantees integer feasibility with high probability (as opposed to the \sqrt{n} radius ball needed in the case of arbitrary polytopes). A recent algorithm for finding low-discrepancy solutions [56] leads to a randomized polynomial-time algorithm for finding an integer point if the polytope contains a ball of radius at least $16c_1 \sqrt{\log(m/n)}$.

From a probabilistic perspective, we show that the *discrepancy* of a random $m \times n$ matrix whose entries are independent and identically chosen from the Gaussian $N(0, \sigma^2)$ is $O(\sigma \sqrt{n \log(m/n)})$. In the process, we also upper bound the *linear and hereditary discrepancy* of random Gaussian matrices.

Our bounds on discrepancy of random Gaussian matrices lead to the required lower bound on the radius of the largest inscribed ball required for integer feasibility since random unit vectors are chosen by picking each coordinate from the Gaussian distribution $N(0, \sigma^2 = 1/n)$ and normalizing the resulting vector. The normalizing coefficient is upper bounded by a constant with high probability (by standard application of Chernoff bound), hence the linear discrepancy gives the radius required for integer feasibility with high probability, upto a constant factor.

Organization. The rest of the chapter is organized as follows. In Section 3.2, we define discrepancy and show the connection to integer feasibility of polytopes. In Section 3.3, we introduce the new model for random IP instances and in Section 3.4, we review known bounds on discrepancy and relevant concentration inequalities. In Section 3.5, we bound the discrepancy of random Gaussian matrices and in Section 3.6, we show the phase-transition phenomenon. Finally, in Section 3.7, we give an algorithm to find integer points in random feasible IP instances using a recent algorithm of Lovett and Meka [56].

3.2 *New tool: Discrepancy for IP*

In this section, we present a simple but seemingly powerful connection between IP and classical discrepancy. Given a matrix $A \in \mathbb{R}^{m \times n}$, the discrepancy of the matrix is defined as follows:

$$\text{disc}(A) := \min_{x \in \{-1, +1\}^n} \|Ax\|_\infty.$$

Suppose we would like to verify if a given polytope contains $-1/1$ points (as opposed to integer points). Given a matrix $A \in \mathbb{R}^{m \times n}$, and a real positive value r , consider the polytope $P(A, r) = \{x \in \mathbb{R}^n : |A_i x| \leq r \forall i \in [m]\}$. The discrepancy of a matrix A is exactly the least r so that the polytope $P(A, r)$ contains a $-1/1$ point. The following proposition is an immediate consequence of this definition.

Proposition 3.2.1. *The polytope $P(A, \text{disc}(A)) = \{x \in \mathbb{R}^n : |A_i x| \leq \text{disc}(A) \forall i \in [m]\}$ contains a $-1/1$ point.*

This is a consequence of the point $x \in \{-1, +1\}^n$ that minimizes discrepancy being contained in the polytope $P(A, \text{disc}(A))$. Thus, if we can evaluate the discrepancy of the constraint matrix A describing a given polytope $P = \{x : Ax \leq b\}$, then by verifying if $\|b\|_\infty$ is at least $\text{disc}(A)$, we have a simple heuristic to verify if the polytope contains a $-1/1$ point. Hence, if each row of A is a normalized unit vector, then the polytope $P = \{x : Ax \leq b\}$ contains a $-1/1$ point if it contains a ball of radius at least $\text{disc}(A)$ centered around the origin.

The following related notion of linear discrepancy helps in characterizing integer feasibility of the polytope (as opposed to $-1/1$ feasibility):

$$\text{lindisc}(A) := \max_{x_0 \in [0,1]^n} \min_{x \in \{0,1\}^n} \|A(x - x_0)\|_\infty.$$

Proposition 3.2.2. *Every polytope $P_{x_0}(A) = \{x \in \mathbb{R}^n : |A_i(x - x_0)| \leq b_i \text{ for } i \in [m]\}$ where $b_i \geq \text{lindisc}(A)$ contains an integer point for every $x_0 \in \mathbb{R}^n$.*

Proposition 3.2.2, similar to Proposition 3.2.1, is an immediate consequence of the definition of linear discrepancy. By definition, every polytope $P_{x_0}(A)$ where $b_i \geq \text{lindisc}(A)$ contains a point $x \in \{0, 1\}^n$ for every $x_0 \in [0, 1]^n$. The proposition follows, since by linear transformation, we may assume that x_0 is in the fundamental cube defined by the standard basis unit vectors. Thus, if each row of the matrix $A \in \mathbb{R}^{m \times n}$ is a unit vector, then linear discrepancy of the constraint matrix gives one possible radius of the largest inscribed ball that guarantees integer feasibility of polytopes described by the constraint matrix A .

This approach to verify integer feasibility of arbitrary polytope fails since it is NP-hard to find the discrepancy of a set-system to within a factor of \sqrt{n} [14]. We show that this approach can still be used for random polytopes due to tight bounds on the discrepancy and the linear discrepancy of Gaussian matrices.

3.3 New model for random IP

A random polytope $P(n, m, x_0, R) = \{x \in \mathbb{R}^n : A_i x \leq b_i \forall i \in [m]\}$ is defined as follows: we pick a random $m \times n$ matrix A with i.i.d. rows from a spherically symmetric distribution; and a vector $b \in \mathbb{R}^m$ such that the hyperplane for each constraint is at distance R from x_0 , i.e., $b_i = R\|A_i\| + A_i x_0$, where A_i is the i 'th row of A .

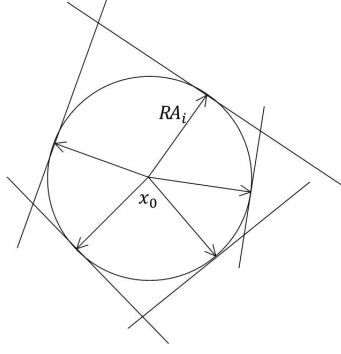


Figure 7: Random IP instance $P(n, m, x_0, R)$

The condition above implies that $P(n, m, x_0, R)$ contains a ball of radius R centered at x_0 . We study the integer feasibility of $P(n, m, x_0, R)$ as a function of the radius R . As the radius R of the random polytope $P(n, m, x_0, R)$ increases, it is likely that the polytope contains an integer point.

3.4 Preliminaries

3.4.1 Related work

The central quantity that leads to all known bounds on discrepancy and linear discrepancy in the literature is hereditary discrepancy defined as follows:

$$\text{herdisc}(A) := \max_{U \subseteq [n]} \text{disc}(A^U)$$

where A^U denotes the submatrix of A containing columns indexed by the set U . The best known bound on discrepancy and hereditary discrepancy of arbitrary matrices is due to Spencer [66].

Theorem 3.4.1 (Spencer [66]). *For any matrix $A \in \mathbb{R}^{m \times n}$ and any subset $U \subseteq [n]$, there exists a point $z \in \{-1, +1\}^{|U|}$ such that*

$$|A_i^U z| \leq 11 \sqrt{|U| \log \frac{2m}{|U|}} \max_{i \in [m], j \in U} |A_{ij}|$$

for every $i \in [m]$.

Lovász, Spencer and Vesztergombi [55] showed the following relation between hereditary discrepancy and linear discrepancy.

Theorem 3.4.2 (Lovász-Spencer-Vesztergombi [55]). *For any matrix A , $\text{lindisc}(A) \leq \text{herdisc}(A)$.*

Hence, every polytope $P = \{x \in \mathbb{R}^n \mid |A_i(x - x_0)| \leq b_i \text{ for } i \in [m]\}$ where

$$b_i = \Omega \left(\max_{i \in [m], j \in [n]} |A_{ij}| \sqrt{n \log(2m/n)} \right)$$

contains an integer point for every $x_0 \in \mathbb{R}^n$.

3.4.2 Concentration Inequalities

We will use the following concentration inequalities to derive the phase-transition radius.

Lemma 3.4.3. *Let Y be a random variable drawn from the Gaussian distribution $N(0, \sigma^2)$.*

Then for any $\lambda > 0$,

$$\Pr(|Y| \leq \lambda\sigma) \leq \min \left\{ 1 - \sqrt{\frac{2}{\pi}} \left(\frac{\lambda}{\lambda^2 + 1} \right) e^{-\frac{\lambda^2}{2}}, \lambda \sqrt{\frac{2}{\pi}} \right\}.$$

Lemma 3.4.4. *Let Y be a random variable drawn from the Gaussian distribution $N(0, \sigma^2)$.*

Then for any $\lambda \geq 1$,

$$\Pr(|Y| \geq \lambda\sigma) \leq 2e^{-\frac{\lambda^2}{2}}.$$

Lemma 3.4.5 (Dasgupta-Schulman [23]). *Let X_1, \dots, X_n be independent random variables each drawn from the Gaussian distribution $N(0, \sigma^2)$. Then for any $\lambda > 0$,*

$$\Pr \left(\left| \sum_{j \in [n]} X_j^2 - n\sigma^2 \right| \geq \lambda \sqrt{n}\sigma^2 \right) \leq 2e^{-\frac{\lambda^2}{24}}.$$

Lemma 3.4.6 (Matoušek-Spencer [57]). *Let X_1, \dots, X_n be independent random variables each drawn uniformly from $\{-1, +1\}$. Then, for any subset $S \subseteq [n]$, any fixed set of vectors a_i , $i \in [m]$, and any $\lambda \geq 0$,*

$$\Pr \left(\left| \sum_{j \in S} a_{ij} X_j \right| \geq \lambda \right) \leq 2e^{-\frac{\lambda^2}{2 \sum_{j \in S} a_{ij}^2}}.$$

3.5 Discrepancy of Gaussian Matrix

In this section, we bound the hereditary discrepancy of Gaussian matrices. This leads to bounds on discrepancy and linear discrepancy.

Implications of known bounds. It is tempting to use known concentration inequalities in conjunction with Spencer’s result (Theorem 3.4.1) to bound the hereditary discrepancy of Gaussian matrices. In this setting, each entry A_{ij} is from $N(0, \sigma^2)$. Using standard concentration for $|A_{ij}|$ and a union bound to bound the maximum entry $|A_{ij}|$ leads to the following weak bound: whp, the polytope $P = \{x \in \mathbb{R}^n \mid |A_i(x - x_0)| \leq b_i \text{ for } i \in [m]\}$ with $b_i = \Omega(\sigma \sqrt{n \log mn \log(2m/n)})$ contains an integer point for any $x_0 \in \mathbb{R}^n$. This is too weak for our purpose (recall that \sqrt{n} radius ball in arbitrary polytopes already guarantees integer feasibility and our goal is to guarantee integer feasibility with smaller inscribed ball in *random* polytopes).

Our Strategy. Our overall strategy to bound hereditary discrepancy is similar to that of Spencer (Theorem 3.4.1). As a first step we show that there exists a partial vector with small discrepancy, i.e., a point $z \in \{0, -1, +1\}^{|S|}$ with at least $|S|/2$ non-zero coordinates such that $|A_i^S z|$ is small for any subset $S \subseteq [n]$. Next for any $U \subseteq [n]$, we repeatedly use the existence of this partial vector to derive a vector $x \in \{-1, 1\}^{|U|}$ with small discrepancy — we start with $x = 0$, $S = U$ and use z to fix at least half of the coordinates of x to $+1$ or -1 ; then we take S to be the set of coordinates that are set to zero in the current x and use z to fix at least half of the remaining coordinates of x to $+1$ or -1 ; we repeat this until all coordinates of x are non-zero. Since at most $|S|/2$ coordinates are set to zero in each round of fixing coordinates, we will repeat at most $\log |U| \leq \log n$ times. The total discrepancy is bounded by the sum of the discrepancies incurred in each round of fixing. Thus, the goal is to bound the discrepancy incurred in each round of fixing.

In Section 3.5.1, we show that the discrepancy incurred when S is the set of coordinates

yet to be fixed is bounded by

$$|A_i^S z| \leq 4 \|A_i^S\| \sqrt{\log \frac{2m}{|S|}} \quad (7)$$

for all vectors A_i , $i \in [m]$ and all subsets $S \subseteq [n]$. This improves on the bound shown by Spencer:

$$|A_i^S z| = O \left(\max_{i \in [m], j \in S} |A_{ij}| \sqrt{|S| \log \frac{2m}{|S|}} \right).$$

We note that Spencer's bound can be recovered from our bound. From inequality (7), it is clear that an upper bound on the length of partial sub-vectors A_i^S leads to an upper bound on the discrepancy incurred by the partial vectors z .

Bounding discrepancy of partial vector. The general bound given in (7) depends on the length of the vector A_i^S . We describe two straightforward approaches which do not lead to tight bounds.

Approach 1. It is straightforward to obtain $\|A_i^S\| \leq 2\sigma \sqrt{|S| \log mn}$ whp for random Gaussian vectors A_i using well-known upper bound on the maximum coefficient of A_i^S . This leads to an upper bound of

$$8\sigma \sqrt{|U| \log(mn) \log \frac{2m}{|U|}}$$

on the discrepancy of A^U . Although this bound on the discrepancy of A^U is good enough when the cardinality of U is smaller than some threshold, it is too large for large U . E.g., when $U = [n]$, this gives a total discrepancy of at most $O(\sigma \sqrt{n \log(mn) \log(2m/n)})$.

Approach 2. Another possible approach to obtain tighter bounds is to bound the length of vector A_i^S when each entry in the vector is from $N(0, \sigma^2)$ (as opposed to bounding the maximum coefficient): using Lemma 3.4.5, for any fixed $S \subseteq [n]$ and $i \in [m]$,

$$\Pr \left(|\|A_i^S\|^2 - |S|\sigma^2| \geq \lambda\sigma^2 \right) \leq 2e^{-\frac{\lambda^2|S|}{24}}.$$

By union bound, we get that

$$\Pr \left(\exists S \subseteq [n], i \in [m] : |\|A_i^S\|^2 - |S|\sigma^2| \geq \lambda\sigma^2 \right) \leq 2e^{-\frac{\lambda^2}{24|S|}} \cdot \binom{n}{|S|} \cdot m \leq 2e^{-\frac{\lambda^2}{24|S|}} n^{|S|} m.$$

Thus, taking $\lambda = |S|\sqrt{48(\log(n) + (1/|S|)\log m)}$ we get

$$\|A_i^S\| \leq 48\sigma|S|\sqrt{\log n + (1/|S|)\log m} \text{ for every } i \in [m] \text{ and } S \subseteq [n] \text{ whp.}$$

Therefore, the discrepancy incurred in each round of fixing is at most

$$196\sigma|S|\sqrt{\left(\log n + \frac{1}{|S|}\log m\right) \log \frac{2m}{|S|}}$$

and the total bound on discrepancy of A^U is

$$O\left(\sigma|U|\sqrt{\log \frac{2m}{|U|}} \left(\sqrt{\log n} + \sqrt{\frac{\log m}{|U|}}\right)\right).$$

This gives the discrepancy to be at most $O(\sigma n \sqrt{\log n \log(2m/n)})$, which is still large. In fact, when each entry is from $N(0, \sigma^2)$, it is possible that there exists a subset of coordinates $S \subseteq [n]$ such that the length of A_i^S is $\Omega(\sigma|S|)$.

New Approach. In order to bound the total discrepancy, we only need to bound the length of the remaining vector after each round of fixing. Let S denote the set of coordinates to be fixed in the current round. The existence lemma (Lemma 3.5.3) picks some subset from S of at least $|S|/2$ coordinates to fix so that the discrepancy is at most $4\|A_i^S\|\sqrt{\log(2m/|S|)}$. Hence, it leaves at most $|S|/2$ coordinates among the possible $|S|$ coordinates for the next round. It is sufficient to bound the probability that there exists a subset $T \subseteq S$ of size at most $|S|/2$ such that the length of the vector A_i^T is large. We do not need the length of A_i^T to be small for every subset $T \subseteq S$. Thus, the union bound is only over the choices of the coordinates yet to be fixed (subsets of S of size at most $|S|/2$) and not over all possible subsets of coordinates. We use this approach in Lemma 3.5.5 to obtain a stronger bound on the length of the vectors $A_i^{S_k}$ for every $i \in [m]$ and every collection of subsets (S_1, S_2, \dots, S_k) where $S_k \subseteq S_{k-1}$ and $|S_k| \leq n2^{-k}$. This helps us obtain the tighter bound for hereditary discrepancy.

Organization. We show the existence of a partial vector with small discrepancy in Section 3.5.1. Then we bound lengths of Gaussian subvectors in Section 3.5.2. We bound the hereditary discrepancy of Gaussian matrices in Section 3.5.3.

3.5.1 Partial Vector

We do not assume any randomness on the choice of the matrix A in this section. We show that for any matrix A and any subset S , there exist two vectors $x, y \in \{+1, -1\}^{|S|}$ such that

1. $|A_i^S x - A_i^S y|$ is small for every $i \in [m]$ and
2. x and y differ in a large number of coordinates.

Thus, taking $z = \frac{x-y}{2}$ gives a vector z so that $z \in \{0, -1, +1\}^{|S|}$ and z has a large number of non-zero coordinates. Further, since $|A_i^S(x - y)|$ is small, $|A_i^S z|$ is also small for every $i \in [m]$. We will quantify the magnitudes formally in Lemma 3.5.3.

Proof Sketch. The existence of vectors x, y will be shown by the probabilistic method. In order to show that there exist vectors $x, y \in \{-1, +1\}^{|S|}$ satisfying condition 1 above, we consider the value $|A_i^S x|$ for every $x \in \{-1, +1\}^{|S|}$. We show that there exist $x, y \in \{-1, +1\}^{|S|}$ so that the difference between $|A_i^S x|$ and $|A_i^S y|$ is small for each $i \in [m]$. For this, we consider a real line for each $i \in [m]$ and equi-partition the i 'th line into small parts for each $i \in [m]$. Then, we show that there exist an exponential number of vectors $x \in \{-1, +1\}^{|S|}$ such that their corresponding $|A_i^S x|$ values fall in the same part for every $i \in [m]$. Thus, we get a set containing exponential number of vectors in $\{-1, +1\}^{|S|}$ so that for any pair of vectors x, y in this set, $|A_i^S x| - |A_i^S y|$ is at most the length of each part corresponding to $i \in [m]$. Therefore, we have an exponential number of vectors satisfying property 1.

Finally, since an exponential number of vectors $x \in \{-1, +1\}^{|S|}$ satisfy property 1, there should exist at least two such vectors x and y with large hamming distance. Thus, among the set of vectors satisfying property 1, there should exist at least two vectors satisfying property 2.

Notation. We define the following function for equi-partitioning. For any $\lambda > 0$, define buckets

$$B_0^\lambda := [-\lambda, \lambda]$$

$$\text{for every positive integer } l, B_l^\lambda := ((2l - 1)\lambda, (2l + 1)\lambda]$$

$$B_{-l}^\lambda := [-(2l + 1)\lambda, -(2l - 1)\lambda).$$

Suppose we have a real vector $t = (t_1, \dots, t_m)$. Then define the bucketing function $P^t(x) = (P_1^{t_1}(x), \dots, P_m^{t_m}(x))$ where

$$P_i^\lambda(x) = j \text{ if } \sum_{k \in S} A_{ik} x_k \in B_j^\lambda.$$

Thus, the length of each part in the equipartition for the i 'th vector is $2t_i$.

Let Y be a random variable chosen from a discrete set D . Then, the entropy of Y is defined to be

$$\text{ENT}(Y) := \sum_{d \in D} \Pr(Y = d) \log \frac{1}{\Pr(Y = d)}.$$

The following lemma shows the existence of a point z with discrepancy of A_i due to z being at most t_i for each $i \in [m]$ if the entropy of the bucketing function is small.

Lemma 3.5.1. *Suppose $x \in \{-1, +1\}^n$ is chosen uniformly at random. If $\text{ENT}(P^t(x)) \leq |S|/5$, then there exists a point $z \in \{0, -1, +1\}^{|S|}$ with at least $|S|/2$ non-zero coordinates such that*

$$|A_i^S z| \leq t_i \text{ for each } i \in [m].$$

Proof. Let $r = |S|$. Since $\text{ENT}(P^t(x)) \leq \frac{r}{5}$, there exists a vector $b = (b_1, \dots, b_m)$ such that $\Pr(P^t(x) = b) \geq 2^{-\frac{r}{5}}$. Since total number of possible choices for x is 2^r , at least $2^{\frac{4r}{5}}$ of the choices for x should map to b . This implies that there exist x, y which differ in at least $r/2$ coordinates such that $P^t(x) = P^t(y)$ [47]. Taking $z = \frac{x-y}{2}$ completes the proof of the lemma. \square

Our next lemma upper bounds the entropy. This is very similar to Lemma 2.3 in [57].

Lemma 3.5.2. *Let S be an arbitrary subset of $[n]$. Let $x \in \{-1, +1\}^{|S|}$ be chosen uniformly at random. Then $\text{ENT}(P_i^{t_i}(x)) \leq G(t_i/\|A_i^S\|)$ for every $i \in [m]$, where*

$$G(\lambda) = \begin{cases} 40e^{-\frac{\lambda^2}{9}} & \text{if } \lambda > 0.1, \\ 40 \ln(1/\lambda) & \text{if } \lambda \leq 0.1. \end{cases}$$

Proof. Suppose we pick x uniformly at random in $\{+1, -1\}^{|S|}$. Let

$$p_k := \Pr\left(P_i^{\lambda\|A_i^S\|}(x) = k\right).$$

Then, $\text{ENT}\left(P_i^{\lambda\|A_i^S\|}(x)\right) = \sum_k -p_k \log(p_k)$. Also,

$$\mathbb{E}(x_j) = 0 \text{ for each } j \in [n],$$

$$\mathbb{E}(x_j^2) = 1 \text{ for each } j \in [n],$$

$$\mathbb{E}(A_i^S x) = 0 \text{ for any } A_i,$$

$$\mathbb{E}\left((A_i^S x)^2\right) = \sum_{j \in S} A_{ij}^2 = \|A_i^S\|^2 \text{ for any } A_i.$$

By Lemma 3.4.6,

$$\Pr(A_i^S x \geq \lambda \|A_i^S\|) \leq e^{-\frac{\lambda^2}{2}}.$$

Define

$$g_k := e^{-\frac{\lambda^2(2k-1)^2}{8}}, \quad k \geq 1$$

$$g_0 := 1 - 2e^{-\frac{\lambda^2}{8}}.$$

By Lemma 3.4.6, $p_k, p_{-k} \leq g_k$ and $p_0 \geq g_0$. The function $-x \log x$ is increasing in $(0, 1/e)$ and decreasing in $[1/e, 1]$.

When $\lambda \geq 10$, $g_0 \geq 1/e$ and $g_k < 1/e$ for $k \geq 1$. Therefore,

$$\text{ENT}\left(P_i^{\lambda\|A_i^S\|}(x)\right) \leq -g_0 \log g_0 + 2 \sum_{k=1}^{\infty} -g_k \log g_k \leq 26e^{-\frac{\lambda^2}{9}}.$$

When $0.1 \leq \lambda \leq 10$, by Jensen's inequality, $\sum_{k=-100}^{100} -p_k \log p_k \leq \log |K| \leq 8$. For $|k| \geq 101$, $g_k < 1/e$ and hence

$$\sum_{k=101}^{\infty} -p_k \log p_k \leq \sum_{k=101}^{\infty} -g_k \log g_k \leq \frac{1}{2}.$$

Thus,

$$\text{ENT} \left(P_i^{\lambda \|A_i^S\|}(x) \right) \leq 9 \leq 26e^{-\frac{\lambda^2}{9}}.$$

When $\lambda < 0.1$, by Jensen's inequality, $\sum_{k:|k|\leq\lambda^{-20}} -p_k \log p_k \leq \log |K|$. For $|k| > \lambda^{-20}$, $g_k < 1/e$. Therefore,

$$\text{ENT} \left(P_i^{\lambda \|A_i^S\|}(x) \right) \leq \log(1 + 2\lambda^{-20}) + 2 \sum_{k:|k|\geq\lambda^{-20}} -g_k \log g_k \leq 40 \ln(1/\lambda).$$

□

We now prove the partial vector lemma.

Lemma 3.5.3. *For any set of vectors $A_1, \dots, A_m \in \mathbb{R}^n$ and any subset $S \subseteq [n]$, there exists a point $z \in \{0, -1, +1\}^{|S|}$ with at least $|S|/2$ non-zero coordinates such that*

$$|A_i^S z| \leq 8 \|A_i^S\| \sqrt{\log \frac{2m}{|S|}} \quad \forall i \in [m].$$

Proof. Let $r = |S|$. Suppose we pick x uniformly at random in $\{+1, -1\}^{|S|}$. We show that $\text{ENT}(P^t(x)) \leq r/5$ for $t_i = 8 \|A_i^S\| \sqrt{\log \frac{2m}{r}}$, $i \in [m]$. The existence of a point $z \in \{0, -1, +1\}^{|S|}$ with at least $|S|/2$ non-zero coordinates such that

$$|A_i^S z| \leq t_i \quad \text{for each } i \in [m]$$

follows by Lemma 3.5.1.

By sub-additivity of entropy function,

$$\text{ENT}(P^t(x)) \leq \sum_{i=1}^m \text{ENT}(P_i^{t_i}(x)).$$

Due to the choice of t_i , we have that $t_i / \|A_i^S\| = 8\sqrt{\log(2m/r)} > 0.1$. Therefore, by Lemma 3.5.2,

$$\text{ENT}(P_i^{t_i}(x)) \leq 40e^{-(64/9) \log \frac{2m}{r}}.$$

Thus,

$$\text{ENT}(P^t(x)) \leq 40me^{-7 \log \frac{2m}{r}} \leq \frac{r}{5}.$$

□

3.5.2 Bounding Lengths of Gaussian Subvectors

In this section we bound lengths of Gaussian subvectors. These bounds will be useful in conjunction with the partial vector lemma (Section 3.5.1) to bound discrepancy of Gaussian matrices.

Lemma 3.5.4. *If each entry A_{ij} is drawn i.i.d. from $N(0, \sigma^2)$, then with high probability*

$$\max_{i \in [m]} \|A_i^S\| \leq 2\sigma \sqrt{|S| \log mn}$$

for every subset $S \subseteq [n]$.

Proof. If each entry A_{ij} is drawn i.i.d. from $N(0, \sigma^2)$, then by Lemma 3.4.4 the maximum entry $|A_{ij}|$, $i \in [m], j \in [n]$ is at most $2\sigma \sqrt{\log mn}$ with high probability. \square

Next we obtain an upper bound on the length of A_i^S when $|S|$ is large.

Lemma 3.5.5. *Suppose we have a matrix $A \in \mathbb{R}^{m \times n}$ where $m \geq n \geq \log m$ and each entry A_{ij} is drawn from $N(0, \sigma^2)$. For any collection of subsets $S_0, S_1, S_2, \dots, S_{\log(n/\log m)}$ of the set $[n]$, where $S_0 \subseteq [n]$, $S_k \subseteq S_{k-1}$, $|S_k| \leq n2^{-k}$ for $k = 0, 1, \dots, \log(n/\log m)$, the following holds with high probability.*

$$\|A_i^{S_k}\|^2 \leq 16n2^{-k}\sigma^2 \tag{8}$$

for every $i \in [m]$ and $k = 0, 1, \dots, \log(n/\log m)$.

Proof. Let S be a collection of subsets $S_0, S_1, \dots, S_{\log(n/\log m)}$ of $[n]$ such that $S_0 \subseteq [n]$, $S_k \subseteq S_{k-1}$ and $|S_k| \leq n2^{-k}$ for $k = 0, 1, \dots, \log(n/\log m)$. We will show that (8) holds for every possible S .

We say that a subset S_k of the collection S is *heavy* if there exists $i \in [m]$ such that it violates (8). We denote a collection S of subsets to be *heavy* if there exists $k \in \{0, 1, \dots, \log(n/\log m)\}$ such that S_k is heavy.

Thus, a collection is heavy if one of its subsets is heavy. We will bound the probability that there exists a heavy collection. Therefore,

$$\Pr(\exists S : S \text{ is heavy}) \leq \sum_{k=0}^{\log(n/\log m)} \Pr\left(\exists S_k \subseteq S_{k-1}, |S_k| \leq n2^{-k} : S_k \text{ is heavy}\right)$$

We bound each term in the above sum as follows. For $k = 0$,

$$\begin{aligned}
\Pr(S_0 \subseteq [n] \text{ is heavy}) &\leq \Pr\left(\exists S_0 \subseteq [n], i \in [m] : \|A_i^{S_0}\|^2 > 16n\sigma^2\right) \\
&\leq \Pr\left(\exists i \in [m] : \|A_i\|^2 > 16n\sigma^2\right) \\
&\leq 2e^{-5n} \cdot m \quad (\text{Using Lemma 3.4.5}) \\
&\leq 2e^{-4\log m} \quad (n \geq \log m).
\end{aligned}$$

For each $k = 1, 2, \dots, \log(n/\log m)$,

$$\begin{aligned}
&\Pr\left(\exists S_k \subseteq S_{k-1}, |S_k| \leq n2^{-k} : S_k \text{ is heavy}\right) \\
&\leq \Pr\left(\exists S_k \subseteq S_{k-1}, |S_k| \leq n2^{-k}, i \in [m] : \|A_i^{S_k}\|^2 > 16n2^{-k}\sigma^2\right) \\
&\leq 2e^{-\frac{256n^2 2^{-2k}}{48|S_k|}} \cdot 2^{n2^{-(k-1)}} \cdot m \quad (\text{Using Lemma 3.4.5}) \\
&\leq 2e^{-\left(\frac{5n}{2^k} - \frac{n}{2^{k-1}} - \log m\right)} \quad (|S_k| \leq n2^{-k}) \\
&\leq 2e^{-2\log m} \quad (k \leq \log(n/\log m)).
\end{aligned}$$

Thus,

$$\Pr(\exists S : S \text{ is heavy}) \leq \frac{2}{m^2} \cdot \log\left(\frac{n}{\log m}\right) \rightarrow 0.$$

□

3.5.3 Discrepancy Bound

We will derive an upper bound on the hereditary discrepancy. The bound on the discrepancy of submatrix A^S that we derive is independent of the size of S . This is unlike Spencer's result (Theorem 3.4.1) where the discrepancy of A^S is bounded by a function of $|S|$.

Theorem 3.5.6. *Suppose we have m vectors $A_1, \dots, A_m \in \mathbb{R}^n$, such that A_{ij} is drawn from the distribution $N(0, \sigma^2)$ for each $i \in [m], j \in [n]$. Then, for any $S \subseteq [n]$, with high probability, there exists a point $x \in \{-1, 1\}^{|S|}$ such that,*

$$|A_i^S x| \leq 32\sigma \left(\sqrt{n \log \frac{2m}{n}} + \sqrt{\log m \log mn \log \frac{2m}{\log m}} \right) \text{ for every } i \in [m].$$

Proof. We use Lemma 3.5.3 repeatedly to fix the coordinates of x . We start with $S_0 = S$. By Lemma 3.5.3 there exists a point $z_0 \in \{0, -1, +1\}^{|S_0|}$ containing at most $|S_0|/2$ zeros. Let S_1 denote the subset of coordinates of z_0 that are zero. Then we set $z(j) = z_0(j)$ for every $j \notin S_1$. We take $S = S_1$. By Lemma 3.5.3 there exists a point $z_1 \in \{0, -1, +1\}^{|S|}$ containing at most $|S|/2$ zeros. Let S_2 denote the subset of coordinates of z_1 that are zero. Then we set $x(j) = z_1(j)$ for every $j \notin S_2$. We repeat this until the number of coordinates of x that are yet to be set is at most a constant with high probability. We set these remaining coordinates to be $-1/1$ arbitrarily. The discrepancy incurred by x due to this arbitrary setting is at most a constant.

We use Lemma 3.5.5 to bound the discrepancy incurred when the number of coordinates to be fixed is greater than $\log m$ and Lemma 3.5.4 to bound the discrepancy incurred when the number of coordinates to be fixed is at most $\log m$.

By Lemma 3.5.3, the discrepancy incurred by x while setting its coordinates using subset S_k , $k \in \{0, 1, \dots, \log(n/\log m)\}$ is at most

$$|A_i^{S_k} z_k| \leq 8 \|A_i^{S_k}\| \sqrt{\log \frac{2m}{|S_k|}} \leq 32\sigma \sqrt{n2^{-k} \log \frac{2m}{n2^{-k}}}$$

with high probability. Here, the second inequality is by using Lemma 3.5.5.

Thus, the discrepancy incurred by x due to $z_0, z_1, \dots, z_{\log(n/\log m)}$ is at most

$$\sum_{k=0}^{\log \frac{n}{\log m}} |A_i^{S_k} z_k| \leq \sum_{k=0}^{\log \frac{n}{\log m}} 32\sigma \sqrt{n2^{-k} \log \frac{2m}{n2^{-k}}} \leq 32\sigma \sqrt{2n \log \frac{2m}{n}}$$

with high probability.

For $k \geq \log(n/\log m)$, the number of coordinates $|S_k| \leq \log m$. By Lemma 3.5.3, the discrepancy incurred by x while setting its coordinates using subset S_k , $k \in \{\log(n/\log m) + 1, \dots, \log |S|\}$ is at most

$$|A_i^{S_k} z_k| \leq 8 \|A_i^{S_k}\| \sqrt{\log \frac{2m}{|S_k|}} \leq 16\sigma \sqrt{n2^{-k} \log(mn) \log \frac{2m}{n2^{-k}}}$$

with high probability. Here, the second inequality is by using Lemma 3.5.4 and $|S_k| \leq n2^{-k}$.

Thus, the discrepancy incurred by x due to $z_{\log(n/\log m)+1}, \dots, z_{\log |S|}$ is at most

$$\begin{aligned} \sum_{k=\log \frac{n}{\log m}}^{k=\log |S|} |A_i^{S_k} z_k| &\leq \sum_{k=\log \frac{n}{\log m}}^{k=\log |S|} 16\sigma \sqrt{n2^{-k} \log(mn) \log \frac{2m}{n2^{-k}}} \\ &\leq 32\sigma \sqrt{\log m \log(mn) \log \frac{2m}{\log m}} \end{aligned}$$

with high probability. Hence, the total discrepancy is bounded by

$$\max_{i \in [n]} |A_i^S x| \leq 64\sigma \left(\sqrt{n \log \frac{2m}{n}} + \sqrt{\log m \log(mn) \log \frac{2m}{\log m}} \right)$$

with high probability. \square

The following upper bound on linear discrepancy follows from Theorems 3.4.2 and 3.5.6. For the sake of completeness, we also give a direct proof. Our proof strategy is well-known (see Corollary 8 in [66]).

Theorem 3.5.7. *Let $A \in \mathbb{R}^{m \times n}$ be a random matrix with i.i.d. entries from $N(0, \sigma^2)$. For any $x_0 \in \mathbb{R}^n$, with high probability, there exists a point $x \in \mathbb{Z}^n$ obtained by rounding each coordinate of x_0 either up or down such that, for every $i \in [m]$,*

$$|A_i(x - x_0)| \leq \frac{1}{2} \sigma \sqrt{n} R_1.$$

Proof. We will find x by rounding x_0 . Without loss of generality, let x_0 be such that $x_0(j) \in [0, 1]$ for each $j \in [n]$. Let the vector x_0 be rational. Suppose each coordinate in x_0 can be expressed in binary using at most p bits. We will round in p phases - each phase will reduce the number of bits needed to express each coordinate in the rounded vector by one.

Consider the binary expansion $x_0(j) = \sum_{k=0}^p \delta_{j,k} 2^{-k}$, $\delta_{j,k} \in \{0, 1\}$. Let S denote the set of coordinates of x_0 which require precision at the p -th bit, i.e., $S = \{j : \delta_{j,p} = 1\}$. Now, by Theorem 3.5.6, there exists a point $z \in \{-1, +1\}^{|S|}$ such that

$$|A_i^S z| \leq 64\sigma \left(\sqrt{|S| \log \frac{2m}{|S|}} + \sqrt{\log m \log(m|S|) \log \frac{2m}{\log m}} \right).$$

Now, consider the following rounding procedure to obtain x_1 : Set $z(j) = 0$ for every $j \notin S$ and $x_1 = x_0 + z2^{-p}$. It is clear that the number of bits needed to express x_1 is at most $p - 1$. This is because, exactly those coordinates which required precision at the p -th bit

were rounded. Further, they were rounded in a manner so that the p -th bit is set to 0. This is because $z(j) \in \{+1, -1\}$ for every $j \in S$ (rounding could possibly change the $p-1$ -th bit in each coordinate). We also have that

$$|A_i(x_1 - x_0)| = |A_i z 2^{-p}| \leq 64\sigma \left(\sqrt{|S| \log \frac{2m}{|S|}} + \sqrt{\log m \log (m|S|) \log \frac{2m}{\log m}} \right) \cdot 2^{-p}.$$

We repeat this rounding procedure at most $p-1$ times thereby reducing the number of bits of precision by at least one each time. Thus, the final \bar{x} obtained needs one bit of precision for each coordinate and hence $\bar{x} \in \{0, 1\}^n$. Finally,

$$\begin{aligned} |A_i(\bar{x} - x_0)| &\leq \sum_{k=1}^{p-1} |A_i(x_k - x_{k-1})| \\ &\leq 64\sigma \sum_{k=1}^p 2^{-k} \left(\sqrt{|S_k| \log \frac{2m}{|S_k|}} + \sqrt{\log m \log (mn) \log \frac{2m}{\log m}} \right) \\ &\leq 64\sigma \sum_{j=1}^p 2^{-j} \left(\sqrt{n \log \frac{2m}{n}} + \sqrt{\log m \log (mn) \log \frac{2m}{\log m}} \right) \quad (\text{Since } |S_k| \leq n) \\ &\leq 32\sigma \left(\sqrt{n \log \frac{2m}{n}} + \sqrt{\log m \log mn \log \frac{2m}{\log m}} \right) \cdot \sum_{j=1}^{p-1} 2^{-j} \\ &\leq 64\sigma \left(\sqrt{n \log \frac{2m}{n}} + \sqrt{\log m \log (mn) \log \frac{2m}{\log m}} \right). \end{aligned}$$

□

3.6 Phase-transition in random IP

In this section, we show a phase-transition phenomenon for probabilistic IP instances in the model described in Section 3.3. In Section 3.6.1, we show a threshold on the radius for integer infeasibility. In Section 3.6.2, we derive the upper bound on radius for integer feasibility from discrepancy bounds and show the phase-transition phenomenon.

3.6.1 Radius for Integer Infeasibility

To get an infeasibility threshold, we begin with a lower bound R_0 so that any 0/1 point is infeasible for $P(n, m, x_0, R_0)$ for every $x_0 \in [0, 1]$, and then extend this to ensure that all integer points are infeasible. The following lemma shows an upper bound on the radius required so that the random polytope $P(n, m, 0, R)$ does not contain an integer point with all nonzero coordinates.

Lemma 3.6.1. For $m \geq 1000n$, let $A \in \mathbb{R}^{m \times n}$ be a matrix whose entries are chosen i.i.d. from the Gaussian distribution $N(0, \sigma^2)$. With probability at least $1 - 2^{-n}$, there does not exist $x \in \mathbb{Z}^n \cap \{x \in \mathbb{R}^n : |x_j| \geq 1 \ \forall j \in [n]\}$ such that

$$|A_i x| \leq \sigma \sqrt{n \log(2m/n)} \text{ for every } i = 1, \dots, m.$$

Proof. For each $r > 0$, we define the set

$$U_r := \mathbb{Z}^n \cap \{x : \|x\| = r, |x_j| > 0 \ \forall j \in [n]\}.$$

We will show that with probability at least $1 - 2^{-n}$, there does not exist $x \in \cup_{r \geq 0} U_r$ satisfying all the m inequalities. We observe that U_r is non-empty only if $r \geq \sqrt{n}$. Fix $r \geq \sqrt{n}$ and a point $X \in U_r$. Now, for $i \in [m]$, since each A_{ij} is chosen from $N(0, \sigma^2)$, the dot product

$$A_i x = \sum_{j=1}^n A_{ij} x_j$$

is distributed according to the Gaussian distribution $N(0, r^2 \sigma^2)$. Let

$$P_x := \Pr \left(|A_i x| \leq \sigma \sqrt{n \log \frac{2m}{n}} \ \forall i \in [m] \right),$$

$$P_r := \Pr \left(\exists x \in U_r : |A_i x| \leq \sigma \sqrt{n \log \frac{2m}{n}} \ \forall i \in [m] \right).$$

By union bound,

$$P_r \leq \sum_{x \in U_r} P_x \leq |U_r| \max_{x \in U_r} P_x.$$

We will obtain an upper bound on P_x that depends only on r . To bound the size of the set U_r , we observe that every point in U_r is an integer point on the surface of a sphere of radius r centered around the origin and hence is contained in an euclidean ball of radius $r + 1$ centered around the origin. Thus, $|U_r|$ can be bounded by the volume of the sphere of radius $r + 1 \leq 2r$ centered around the origin:

$$|U_r| \leq \text{vol}(2r\mathbb{B}_0) \leq \left(2r \sqrt{\frac{2\pi e}{n}} \right)^n \leq \left(\frac{10r}{\sqrt{n}} \right)^n.$$

Next we bound P_r . We have two cases.

Case 1. Let $r \in \left[\sqrt{n}, \sqrt{n \log(2m/n)} \right]$. Since $A_i x$ is distributed according to $N(0, r^2 \sigma^2)$, by

Lemma 3.4.3,

$$\Pr \left(|A_i x| \leq \sigma \sqrt{n \log \frac{2m}{n}} \right) \leq 1 - \frac{1}{\sqrt{2\pi}} \left(\frac{r \sqrt{n \log \frac{2m}{n}}}{r^2 + n \log \frac{2m}{n}} \right) \cdot \left(\frac{n}{2m} \right)^{\frac{n}{2r^2}}.$$

Since each A_{ij} is chosen independently, we have that

$$\begin{aligned} P_x &= \prod_{i=1}^m \Pr \left(|A_i x| \leq \sigma \sqrt{n \log \frac{2m}{n}} \right) < \left(1 - \frac{1}{\sqrt{2\pi}} \left(\frac{r \sqrt{n \log \frac{2m}{n}}}{r^2 + n \log \frac{2m}{n}} \right) \cdot \left(\frac{n}{2m} \right)^{\frac{n}{2r^2}} \right)^m \\ &\leq e^{-\frac{1}{\sqrt{2\pi}} \left(\frac{r \sqrt{n \log \frac{2m}{n}}}{r^2 + n \log \frac{2m}{n}} \right) \cdot \left(\frac{n}{2m} \right)^{\frac{n}{2r^2}} \cdot m} \cdot e^{-n \log \frac{10r}{\sqrt{n}}} \leq e^{-n \log \frac{10r}{\sqrt{n}}} \leq \left(\frac{\sqrt{n}}{10r} \right)^n. \end{aligned}$$

Therefore, by union bound, it follows that

$$P_r \leq e^{-\frac{1}{\sqrt{2\pi}} \left(\frac{r \sqrt{n \log \frac{2m}{n}}}{r^2 + n \log \frac{2m}{n}} \right) \cdot \left(\frac{n}{2m} \right)^{\frac{n}{2r^2}} \cdot m + n \log \frac{10r}{\sqrt{n}}} \leq e^{-n \log \frac{10r}{\sqrt{n}}} \leq \left(\frac{\sqrt{n}}{10r} \right)^n.$$

Case 2. Let $r > \sqrt{n \log(2m/n)}$. Since $A_i x$ is distributed according to $N(0, r^2 \sigma^2)$, by Lemma 3.4.3, we have that

$$\Pr \left(|A_i x| \leq \sigma \sqrt{n \log \frac{2m}{n}} \right) \leq \frac{1}{r} \sqrt{\frac{2}{\pi} n \log \frac{2m}{n}} \leq \frac{4}{5r} \sqrt{n \log \frac{2m}{n}}.$$

The random variables $A_1 x, \dots, A_m x$ are independent and identically distributed. Therefore,

$$P_x = \prod_{i=1}^m \Pr \left(|A_i x| \leq \sigma \sqrt{n \log \frac{2m}{n}} \right) \leq \left(\frac{4}{5r} \sqrt{n \log \frac{2m}{n}} \right)^m.$$

Hence, by union bound,

$$P_r \leq e^{-n \left(\frac{m}{n} \log \left(\frac{5r}{4\sqrt{n \log \frac{2m}{n}}} \right) - \log \frac{10r}{\sqrt{n}} \right)} \leq e^{-n \left(\frac{m}{2n} \log \left(\frac{5r}{4\sqrt{n \log \frac{2m}{n}}} \right) \right)} \leq \left(\frac{4\sqrt{n \log \frac{2m}{n}}}{5r} \right)^{\frac{m}{2}}.$$

Finally,

$$\Pr \left(\exists x \in \cup_{r \geq \sqrt{n}} U_r : |A_i x| \leq \sigma \sqrt{n \log \frac{2m}{n}} \forall i \in [m] \right) = \sum_{r \geq \sqrt{n}} P_r$$

$$\begin{aligned}
\text{and } \sum_{r \geq \sqrt{n}} P_r &= \sum_{r \in [\sqrt{n}, \sqrt{n \log \frac{2m}{n}}]} P_r + \sum_{r > \sqrt{n \log \frac{2m}{n}}} P_r \\
&\leq \frac{1}{10^n} \int_{r=\sqrt{n}}^{\infty} \left(\frac{\sqrt{n}}{r}\right)^n dr + \left(\frac{4}{5}\right)^{\frac{m}{2}} \int_{r=\sqrt{n \log \frac{2m}{n}}}^{\infty} \left(\frac{\sqrt{n \log \frac{2m}{n}}}{r}\right)^{\frac{m}{2}} dr \\
&\leq \frac{1}{10^n} \cdot \frac{\sqrt{n}}{n-1} + \left(\frac{4}{5}\right)^{\frac{m}{2}} \cdot \left(\frac{2\sqrt{n \log \frac{2m}{n}}}{m-2}\right) \\
&\leq \frac{1}{2^n} \quad (\text{since } m \geq 1000n).
\end{aligned}$$

□

We obtain the following Corollary from Lemma 3.6.1 by the choice of x_0 .

Corollary 3.6.2. *For $m \geq 1000n$, let $A \in \mathbb{R}^{m \times n}$ be a matrix whose entries are chosen i.i.d. from the Gaussian distribution $N(0, \sigma^2)$. Let $x_0 := (1/2, \dots, 1/2) \in \mathbb{R}^n$. Then,*

$$\Pr \left(\exists x \in \mathbb{Z}^n : |A_i(x - x_0)| \leq \frac{\sigma}{2} \sqrt{n \log \frac{2m}{n}} \quad \forall i \in [m] \right) \leq \frac{1}{2^n}.$$

Proof. There exists $x \in \mathbb{Z}^n$ such that

$$|A_i(x - x_0)| \leq \frac{\sigma}{2} \sqrt{n \log \frac{2m}{n}} \quad \forall i \in [m]$$

if and only if there exists $x \in \mathbb{Z}^n \cap \{x \in \mathbb{R}^n : x_j \geq 1 \quad \forall j \in [n]\}$ such that

$$|A_i x| \leq \sigma \sqrt{n \log \frac{2m}{n}} \quad \forall i \in [m].$$

The result follows by Lemma 3.6.1. □

3.6.2 Threshold Radius

We now have all the ingredients needed to show the threshold radius.

Theorem 3.6.3. *Let $m \geq 1000n$ and*

$$R_0 = \sqrt{\frac{1}{6} \log \frac{2m}{n}}, \quad R_1 = 256 \left(\sqrt{\log \frac{2m}{n}} + \sqrt{\frac{\log m \log(mn) \log(2m/\log m)}{n}} \right).$$

Then, with probability at least $1 - 2me^{-n/96}$,

1. for every $x_0 \in \mathbb{R}^n$, the random polytope $P(n, m, x_0, R)$ contains an integer point when $R \geq R_1$, and
2. for $x_0 = (1/2, \dots, 1/2)$, the random polytope $P(n, m, x_0, R)$ does not contain an integer point when $R \leq R_0$ if $m = 2^{O(\sqrt{n})}$.

Proof. Let $P = \{x \in \mathbb{R}^n : a_i x \leq b_i \forall i \in [m]\}$, where each a_i is chosen from a spherically symmetric distribution. Then $\alpha_i = a_i / \|a_i\|$ for $i \in [m]$ is distributed randomly on the unit sphere. A random unit vector α_i can be obtained by drawing each coordinate from the Gaussian distribution $N(0, \sigma^2 = 1/n)$ and normalizing the resulting vector. Thus, we may assume $\alpha_i = A_i / \|A_i\|$ where each coordinate A_{ij} is drawn from the Gaussian distribution $N(0, 1/n)$. Here, we show that the probability that there exists a vector A_i that gets scaled by more than a constant is at most $2me^{-n/96}$.

Taking $r = n$ and $\sigma^2 = 1/n$ in Lemma 3.4.5, we have

$$\Pr\left(\exists i \in [m] : \left| \|A_i\|^2 - 1 \right| > \frac{1}{2}\right) \leq 2me^{-\frac{n}{96}}.$$

Hence, with probability at least $1 - 2me^{-n/96}$, we have that $\sqrt{1/2} \leq \|A_i\| \leq \sqrt{3/2}$ for every $i \in [m]$. We now show the upper and lower bounds.

1. Since P contains a ball of radius R_1 , $P \supseteq Q$ where

$$Q = \{x \in \mathbb{R}^n \mid |\alpha_i(x - x_0)| \leq R_1 \text{ for } i \in [m]\}$$

Using Theorem 3.5.7 and $\sigma^2 = 1/n$, we know that there exists $x \in \mathbb{Z}^n$ such that for every $i \in [m]$

$$|A_i(x - x_0)| \leq 64 \left(\sqrt{\log \frac{2m}{n}} + \sqrt{\frac{\log m \log(mn)}{n} \log \frac{2m}{\log m}} \right).$$

Thus, with probability at least $1 - 2me^{-n/96}$, there exists $X \in \mathbb{Z}^n$ satisfying

$$|\alpha_i(x - x_0)| = \frac{|A_i(x - x_0)|}{\|A_i\|} \leq 128 \left(\sqrt{\log \frac{2m}{n}} + \sqrt{\frac{\log m \log(mn)}{n} \log \frac{2m}{\log m}} \right)$$

for every $i \in [m]$. Thus the polytope Q is integer feasible and consequently P is also integer feasible.

2. For $x_0 = (1/2, \dots, 1/2)$, let

$$P = \left\{ X \in \mathbb{R}^n : |A_i(x - x_0)| \leq \|A_i\| \sqrt{\frac{1}{6} \log \frac{2m}{n}} \quad \forall i \in [m] \right\}.$$

Then, P contains a ball of radius R_0 centered around x_0 and hence is an instance of the random polytope $P(n, m, x_0, R_0)$. Further, with probability at least $1 - 2me^{-n/96}$, P is contained in

$$Q = \left\{ x \in \mathbb{R}^n : |A_i(x - x_0)| \leq \frac{1}{2} \sqrt{\log \frac{2m}{n}} \quad \forall i \in [m] \right\}.$$

By Corollary 3.6.2, with high probability, we have that $Q \cap \mathbb{Z}^n = \emptyset$. Thus, with probability at least $1 - 2me^{-n/96}$, we have that $P \cap \mathbb{Z}^n = \emptyset$.

□

Remark. For $m = 2^{O(\sqrt{n})}$, the second term in R_1 is of the same order as the first and so the two thresholds are within a constant factor of each other. Thus, in this case, the transition between infeasibility and feasibility happens within a constant factor increase in the radius.

3.7 Algorithm to find Integer Point

We complement our integer feasibility result for random polytope with an algorithm to find an integer point. With $R = \Omega(\sqrt{\log m})$ and $x_0 = (1/2, \dots, 1/2)$, there is a trivial algorithm — pick a random 0/1 vector; most such vectors will be feasible in $P(n, m, x_0, R)$. But with smaller R , and arbitrary centers x_0 , only an exponentially small fraction of nearby integer vectors might be feasible, so such direct sampling/enumeration would not give a feasible integer point.

We will show an algorithm to find small linear discrepancy solutions for Gaussian matrices. The algorithm to find integer points in random polytopes with large inscribed ball follows. Our algorithm for small linear discrepancy solutions for Gaussian matrices is essentially an extension of Lovett-Meka's algorithm for constructive discrepancy minimization [56]. The algorithm runs in phases. In each phase, we start with the current point $x \in [0, 1]^n$ and perform a random walk to arrive at a partial vector y with at least half of the non-integer coordinates of x being close to integers in y . Further, the discrepancy overhead

incurred by y (i.e., $|A_i(y - x)|$) is small. Thus, each phase could be viewed as an equivalent of the partial-vector lemma (Lemma 3.5.3 in Section 3.5.1).

In the description of the algorithm, let $\mathcal{N}(\mathcal{V})$ be the standard multi-dimensional Gaussian distribution in the subspace \mathcal{V} — a random vector G from $\mathcal{N}(\mathcal{V})$ is obtained by taking an orthonormal basis $\{b_1, \dots, b_d\}$ for \mathcal{V} and setting $G = \sum_{k=1}^d G_k b_k$, where $G_k \sim \mathcal{N}(0, 1)$. The algorithm is described in Figure 8. The sub-routine Edge-Walk is a mild variation of the algorithm used in Theorem 4 of Lovett-Meka [56]. Its functionality is summarized in Lemma 3.7.1.

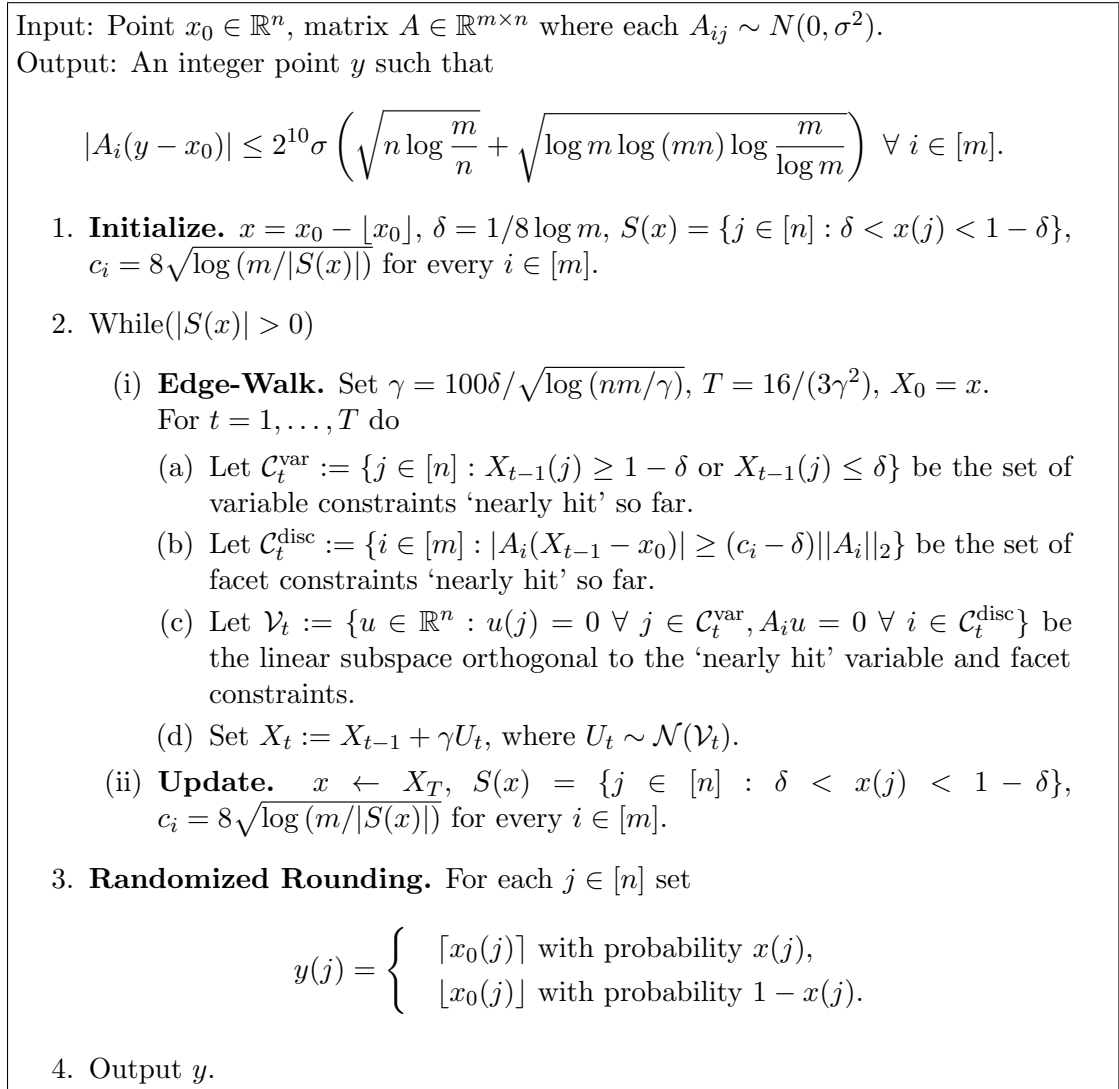


Figure 8: Algorithm Round-IP

Lemma 3.7.1. *Let $A_1, \dots, A_m \in \mathbb{R}^n$ be vectors. Let $x_0 \in [0, 1]^n$ be a “starting” point.*

Let $c_1, \dots, c_m \geq 0$ be thresholds such that $\sum_{i=1}^m \exp(-c_i^2/16) \leq n/16$. Let $\delta \in (0, 0.5]$ be a small approximation parameter. Then Algorithm Edge-Walk is a randomized algorithm which with probability at least 0.1 finds a point $x \in [0, 1]^n$ such that

1. $|A_i(x - x_0)| \leq c_i \|A_i\|_2$ for every $i \in [m]$, and
2. the size of the set $\{j : x_j \geq 1 - \delta \text{ or } x_j \leq \delta\}$ is at least $n/2$.

Moreover, the algorithm runs in time $O((m + n)^3 \delta^{-3} \log(nm/\delta))$.

In Algorithm Round-IP, we repeatedly invoke the Edge-Walk algorithm. Each such call reduces $|S(x)|$ by at least $1/2$. Thus, Step 2 terminates in at most $\log n$ calls to the Edge-Walk algorithm. Further, the total discrepancy overhead incurred by x is at most the sum of the discrepancy overhead incurred in each call to the Edge-Walk algorithm. Also observe that in each call, we have $\sum_{i=1}^m \exp(-c_i^2/16) \leq |S(x)|/16$. The sum of the discrepancy overheads is bounded similar to the proof of Theorem 3.5.6 using Lemmas 3.5.4 and 3.5.5. Finally, in Step 3, we do a randomized rounding. By standard Chernoff bound, the discrepancy incurred due to randomized rounding will be shown to be small.

3.7.1 Algorithmic Partial Vector

We state the polynomial time algorithm Edge-Walk here and elaborate on its functionality. Given a point x_0 as input, the algorithm finds a point x such that at least half the coordinates of x are close to being integers and the discrepancy overhead incurred by x is small. This algorithm is essentially an application of Lovett-Meka's algorithm [56]. Here, we restate a mild variation of Theorem 4 of Lovett-Meka. Lemma 3.7.1 can be obtained from this theorem as a straightforward Corollary by taking $a = 0, b = 1$.

Theorem 3.7.2 (Lovett-Meka [56]). *Let $v_1, \dots, v_m \in \mathbb{R}^n$ be vectors, and a, b be reals such that $|a|, |b| \leq 1$. Let $x_0 \in [a, b]^n$ be a "starting" point. Let $c_1, \dots, c_m \geq 0$ be thresholds such that $\sum_{i=1}^m \exp(-c_i^2/16) \leq n/16$. Let $\delta \in (0, 0.1]$ be a small approximation parameter. Then Algorithm Edge-Walk given in Figure 9 is a randomized algorithm which with probability at least 0.1 finds a point $x \in [a, b]^n$ such that*

1. $|v_i^T(x - x_0)| \leq c_i \|v_i\|_2$ for every $i \in [m]$,

2. The size of the set $\{j : x_j \geq b - \delta \text{ or } x_j \leq a + \delta\}$ is at least $n/2$.

Moreover, the algorithm runs in time $O((m+n)^3 \delta^{-3} \log(nm/\delta))$.

The algorithm starts from the point x_0 and performs a random walk with small step size. During the walk, if the current point is close to being an integer or close to a facet of the polytope, then the random walk continues in an orthogonal subspace. Thus, once the walk almost hits a face of the polytope, it stays on/close to that face; if a coordinate of the current point is close to being an integer, then the walk stays on the subspace defined by fixing that coordinate. After a sufficiently large number of steps, several coordinates of the current point are close to being integers while the number of facets of the polytope that the current point is close to is also small.

Input: Point $x_0 \in \mathbb{R}^n$, vectors $v_1, \dots, v_m \in \mathbb{R}^n$, real values a, b such that $|a|, |b| < 1$, $a < b$, $\delta \in [0, 0.1]$ and c_1, \dots, c_m such that $\sum_{j=1}^m \exp(-c_j^2/16) \leq n/16$.
Output: A point x such that $|v_i^T(x - x_0)| \leq c_i \|v_i\|_2$ for every $i \in [m]$ and $|\{j : x_j \geq b - \delta \text{ or } x_j \leq a + \delta\}| \geq n/2$.

1. Let $\gamma > 0$ be a small step size so that $\gamma = \delta / \sqrt{\log(nm/\gamma)}$. Let $T = 16/(3\gamma^2)$. For $t = 1, \dots, T$ do
 - (a) Let $\mathcal{C}_t^{\text{var}} := \{j \in [n] : X_{t-1}(j) \geq b - \delta \text{ or } X_{t-1}(j) \leq a + \delta\}$ be the set of variable constraints ‘nearly hit’ so far.
 - (b) Let $\mathcal{C}_t^{\text{disc}} := \{i \in [m] : |v_i^T(X_{t-1} - x_0)| \geq (c_i - \delta) \|v_i\|_2\}$ be the set of facet constraints ‘nearly hit’ so far.
 - (c) Let $\mathcal{V}_t := \{u \in \mathbb{R}^n : u(j) = 0 \forall j \in \mathcal{C}_t^{\text{var}}, v_i^T u = 0 \forall i \in \mathcal{C}_t^{\text{disc}}\}$ be the linear subspace orthogonal to the ‘nearly hit’ variable and facet constraints.
 - (d) Set $X_t := X_{t-1} + \gamma U_t$, where $U_t \sim \mathcal{N}(\mathcal{V}_t)$.
2. Return X_T .

Figure 9: Algorithm Edge-Walk

Since the proof of correctness of Theorem 3.7.2 using the above algorithm hardly deviates from that of Lovett-Meka’s, we refer the reader to Lovett-Meka for a complete proof. The only non-trivial Claim that needs to be verified from their proof is Claim 13. The Claim follows in a straightforward manner since only the closeness of the current solution to the variable constraints matter and not the exact constraints. We also need $|a|, |b| \leq 1$ for their Claim 15 to hold.

3.7.2 Finding an Integer Point

We now show the performance of Algorithm Round-IP for Gaussian matrices.

Theorem 3.7.3. *There is a randomized polynomial-time algorithm that takes as input a random matrix $A \in \mathbb{R}^{m \times n}$ with i.i.d. entries from $N(0, \sigma^2)$ and a point $x_0 \in \mathbb{R}^n$, and outputs an integer point y such that for every $i \in [m]$,*

$$|A_i(y - x_0)| \leq 2^{10} \sigma \left(\sqrt{n \log \frac{m}{n}} + \sqrt{\log m \log(mn) \log \frac{m}{\log m}} \right)$$

with high probability. Moreover, the algorithm runs in time

$$O(n \log n (m + n)^3 (\log m)^3 \log nm \log m).$$

Proof. Without loss of generality, we may assume that $x_0 \in [0, 1]^n$ and our objective is to find $x \in \{0, 1\}^n$ with low discrepancy overhead. We use Algorithm Round-IP. We will show that it succeeds with probability at least $1/n$ in finding a point $x \in \{0, 1\}^n$ such that

$$|A_i(x - x_0)| \leq 2^{11} \sigma \left(\sqrt{n \log \frac{m}{n}} + \sqrt{\log m \log(mn) \log \frac{m}{\log m}} \right).$$

The success probability of this algorithm can be amplified by repeating it n times.

Let \bar{x} denote the vector at the end of Step 2 in Algorithm Round-IP and let x_k denote the vector x in Algorithm Round-IP after k calls to the Edge-Walk algorithm. Let $S_k = \{j \in [n] : x_k(j) \geq 1 - \delta \text{ or } x_k(j) \leq \delta\}$. First observe that during the k 'th call to the Edge-Walk subroutine, we have $\sum_{i=1}^m \exp(-c_i^2/16) \leq |S_k|/16$. By Lemma 3.7.1, the discrepancy overhead incurred in the k 'th run of the Edge-Walk sub-routine for $k \in \{0, 1, \dots, \log(n/\log m)\}$ is

$$|A_i^{S_k}(x_k - x_{k-1})| \leq 8 \|A_i^{S_k}\| \sqrt{\log \frac{m}{|S_k|}} \leq 2^7 \sigma \sqrt{n 2^{-k} \log \frac{2m}{n 2^{-k}}}$$

with high probability. Here, the second inequality is by using Lemma 3.5.5 and $|S_k| \leq n 2^{-k}$.

Thus, the discrepancy overhead incurred after $\log(n/\log m)$ calls to the Edge-Walk subroutine is

$$\sum_{k=0}^{\log \frac{n}{\log m}} |A_i^{S_k}(x_k - x_{k-1})| \leq \sum_{k=0}^{\log \frac{n}{\log m}} 2^7 \sigma \sqrt{n 2^{-k} \log \frac{m}{n 2^{-k}}} \leq 2^{10} \sigma \sqrt{n \log \frac{m}{n}}$$

with high probability.

For $k \geq \log(n/\log m)$, the number of non-integer coordinates $|S_k| \leq \log m$. By Lemma 3.7.1, the discrepancy overhead incurred in the k 'th call to the Edge-Walk sub-routine, where $k \in \{\log(n/\log m) + 1, \dots, \log n\}$ is

$$|A_i^{S_k}(x_k - x_{k-1})| \leq 8 \|A_i^{S_k}\| \sqrt{\log \frac{m}{|S_k|}} \leq 2^4 \sigma \sqrt{n 2^{-k} \log(mn) \log \frac{m}{n 2^{-k}}}$$

with high probability. Here, the second inequality is by using Lemma 3.5.4 and $|S_k| \leq n 2^{-k}$.

Thus, the discrepancy overhead incurred by Step 2 in rounds $k = \log(n/\log m)$ to $\log n$ is

$$\begin{aligned} \sum_{k=\log \frac{n}{\log m}}^{k=\log n} |A_i^{S_k}(x_k - x_{k-1})| &\leq \sum_{k=\log \frac{n}{\log m}}^{k=\log n} 2^4 \sigma \sqrt{n 2^{-k} \log(mn) \log \frac{m}{n 2^{-k}}} \\ &\leq 2^7 \sigma \sqrt{\log m \log(mn) \log \frac{m}{\log m}} \end{aligned}$$

with high probability.

Since each call to the Edge-Walk subroutine reduces $s(x)$ by at least half, we call the Edge-Walk subroutine at most $\log n$ times. Each call succeeds with probability $1/2$. Hence, with probability at least $1/2^{\log n} = 1/n$, at the end of Step 2, we obtain a point \bar{x} such that $\bar{x} \in [0, 1]^n$ and $\bar{x}(j) \geq 1 - \delta$ or $\bar{x}(j) \leq \delta$ for every $j \in [n]$ and the total discrepancy overhead is bounded as follows:

$$\max_{i \in [n]} |A_i(\bar{x} - x_0)| \leq 2^{10} \sigma \left(\sqrt{n \log \frac{m}{n}} + \sqrt{\log m \log(mn) \log \frac{m}{\log m}} \right).$$

Next we show that the randomized rounding performed in Step 3 incurs small discrepancy. Consider a coordinate $j \in [n]$ that is rounded. Then,

$$\mathbb{E}(y(j) - \bar{x}(j)) = 0,$$

$$\text{Var}(y(j) - \bar{x}(j)) \leq \delta,$$

and thus,

$$\Delta_i^2 := \text{Var} \left(\sum_{j=1}^n A_{ij}(y(j) - \bar{x}(j)) \right) \leq \|A_i\|^2 \delta.$$

Therefore, for $i \in [m]$, by Chernoff bound,

$$\Pr \left(\left| \sum_{j=1}^n A_{ij}(y(j) - \bar{x}(j)) \right| \geq 4\Delta_i \sqrt{\log m} \right) \leq \frac{2}{m^2}.$$

Hence, by union bound, we get that $|A_i(y - \bar{x})| \leq 4\Delta_i \sqrt{\log m} \leq 4\|A_i\|$ for every $i \in [m]$ with high probability. Now, applying Lemma 3.5.5 for $k = 0$, we get that $|A_i(y - \bar{x})| \leq 4\sigma\sqrt{n}$. Thus,

$$|A_i(y - x_0)| \leq |A_i(y - \bar{x})| + |A_i(\bar{x} - x_0)| \leq 2^{11}\sigma \left(\sqrt{n \log \frac{m}{n}} + \sqrt{\log m \log(mn) \log \frac{m}{\log m}} \right).$$

The running time of the algorithm is obvious – each call to the Edge-Walk subroutine takes $O((m+n)^3 \log(nm \log m)(\log m)^3)$ time, the number of calls is $O(\log n)$ and the number of steps before each call is $O(n)$. \square

We use Theorem 3.7.3 to obtain an algorithm to find integer points in random polytopes containing a ball of large radius.

Theorem 3.7.4. *Given a random polytope $P = P(n, m, x_0, R)$ where $R \geq 16R_1$, there is a randomized polynomial-time algorithm to find an integer point $x \in P$ with probability at least $1 - 2me^{-n/96}$.*

Proof. We use Theorem 3.7.3 to derive Theorem 3.7.4.

Let

$$R_{ALG} = 2^{11}\sigma \left(\sqrt{n \log \frac{2m}{n}} + \sqrt{\log m \log(mn) \log \frac{2m}{\log m}} \right)$$

and $\alpha_i = A_i/\|A_i\|$, $i \in [m]$. Solve the following linear programming problem to find the center of the largest ball contained in the polytope.

$$\max R$$

$$R \leq b_i - A_i x, \quad \forall i \in [m].$$

Let (x_0, R) be a solution to the above LP. Since P contains a ball of radius R_{ALG} , there exists $\beta_i \geq R_{ALG} \|A_i\|$ for every $i \in [m]$ such that

$$P \supseteq \{x \in \mathbb{R}^n \mid |A_i(x - x_0)| \leq \beta_i \text{ for } i \in [m]\}.$$

Observe that the polytope

$$Q := \{x \in \mathbb{R}^n \mid |\alpha_i(x - x_0)| \leq R_{ALG} \text{ for } i \in [m]\}$$

is contained in P . We will show that there exists a randomized polynomial-time algorithm to find an integer point in Q that succeeds with probability at least $(1 - 2me^{-n/96})/2$. This success probability is over the choice of A_i s.

Since each A_i is drawn from a spherically symmetric distribution, $\alpha_i = A_i/\|A_i\|$ is distributed uniformly on the unit sphere. A random unit vector α_i on a sphere is obtained by drawing each coordinate a_{ij} i.i.d. from the Gaussian distribution $N(0, 1/n)$ and scaling the resulting vector by $\|a_i\| = \sqrt{\sum_{j=1}^n a_{ij}^2}$. Similar to the proof of Theorem 3.6.3, a_i gets scaled by at most 2 for every $i \in [m]$ with probability at least $1 - 2me^{-n/96}$. Using Theorem 3.7.3 and $\sigma^2 = 1/n$, we know that there exists a randomized polynomial time algorithm that succeeds with high probability to find $x \in \mathbb{Z}^n$ such that for every $i \in [m]$

$$|a_i(x - x_0)| \leq 2^{10} \left(\sqrt{\log \frac{m}{n}} + \sqrt{\frac{\log m \log(mn)}{n} \log \frac{m}{\log m}} \right).$$

Thus, the same randomized polynomial-time algorithm finds a point $x \in \mathbb{Z}^n$ satisfying

$$|\alpha_i(x - x_0)| = \frac{|a_i(x - x_0)|}{\|a_i\|} \leq 2^{11} \left(\sqrt{\log \frac{2m}{n}} + \sqrt{\frac{\log m \log(mn)}{n} \log \frac{2m}{\log m}} \right)$$

for every $i \in [m]$. The success probability of the algorithm reduces by a factor of $1 - 2me^{-n/96}$ due to the randomness in the input. \square

3.8 Conclusion

We exhibit a new geometric connection between integer feasibility of IP and the classical notion of discrepancy of matrices. We exploit this connection to show a phase transition in random IP instances defined by random constraint matrices. We believe that our result acts as a starting point for the study of natural random IP instances. We conclude by considering some open problems.

An immediate question is to understand the integer feasibility behavior of random IP instances for more general distributions of the constraint matrix, for example, constraint

matrices in which the rows are not necessarily independent of each other. In particular, constraint matrices whose joint distribution is logconcave.

Another open question is to perform optimization over random polytopes, even along a random objective direction. The ability to verify feasibility of random polytope does not immediately lead to an algorithm to optimize over such polytopes, mainly due to the conditioning induced while testing feasibility. Even though we expect the optimum to be concentrated for any fixed direction c , an algorithm to find the optimum point seems elusive. Perhaps, an algorithm to verify integer feasibility of random polytope would help make progress towards optimization – in this chapter, we only provide an algorithm that finds an integer point in random instances that are feasible with high probability. More specifically, how efficient are cutting-plane and branch-and-bound algorithms for optimization over random polytopes?

From a purely probabilistic perspective, our work also raises the issue of the existence of a sharp threshold. Does there exist a sharp threshold radius R^* so that the random polytope $P(n, m, x_0, R)$ is integer infeasible whp for some center x_0 if $R < R^*$ and is integer feasible for all center x_0 whp if $R \geq R^*$?

Propositions 3.2.1 and 3.2.2 hold for arbitrary constraint matrices and do not require any randomness. It would be interesting to understand if these observations could be used to solve IP formulations of combinatorial feasibility and optimization problems. The main bottleneck is the ability to compute the discrepancy of a matrix. Since it is NP-hard to compute discrepancy of arbitrary matrices within a factor of \sqrt{n} [14], it appears that these propositions cannot even be used as heuristics to check the feasibility of arbitrary IP instances. However, one might be able to overcome this bottleneck by exploiting the structure of the constraint matrix describing certain combinatorial polytopes.

CHAPTER IV

LP APPROACH: CONVERGENCE OF CUTTING PLANES FOR MIN-COST PERFECT MATCHING

In this chapter we study the efficiency of cutting plane algorithms. Cutting plane algorithms are commonly used for NP-hard combinatorial problems to obtain lower bounds. Due to their practical efficiency, they are also used for problems known to be solvable in polynomial time using other algorithms. A classic example is the minimum-cost perfect matching problem.

The cutting plane approach to matchings has been discussed by several authors over the past decades [61, 38, 54, 69, 29], but its convergence has long been an open question. We prove that the cutting plane approach using *Edmonds' blossom inequalities* converges in polynomial time for the minimum-cost perfect matching problem. Our new insight is an LP-based method to retain/drop cuts. This careful cut retention procedure leads to a sequence of intermediate linear programs with a linear number of constraints whose optima are half-integral and supported by a disjoint union of odd cycles and edges. This structural property of the optima is instrumental in finding violated blossom inequalities (cuts) in linear time. Further, the number of cycles in the support of the half-integral optima acts as a potential function to show efficient convergence to an integral solution. The results in this chapter are joint work with László Végh and Santosh Vempala [10].

4.1 Introduction

The cutting plane algorithm introduced by Gomory [34, 35, 36] is a leading approach to solve IPs. It can be summarized as follows:

1. Solve the linear programming relaxation (LP) of the given IP to obtain a basic optimal solution x .
2. If x is integral, terminate. If x is not integral, find a linear inequality that is valid for

the convex hull of all integer solutions but violated by x .

3. Add the inequality, possibly drop some other inequalities and solve the resulting LP to obtain a basic optimal solution x . Go back to Step 2.

For the approach to be efficient, we require the following: (a) an efficient method for finding a violated inequality (called a cutting plane) and (b) a bound on the number of iterations needed for convergence. In his seminal work, Gomory [36] gave an efficient method for finding cutting planes and showed a bound of 2^n on the number of iterations for convergence for 0 – 1 integer programs.

There is a rich theory on the choice of cutting planes, both in general and for specific problems of interest. Several years after Gomory’s introduction of cutting plane algorithms, the complete characterization of the Matching polytope by Edmonds [27] inspired Chvátal to revisit Gomory’s cutting plane generation method from a polyhedral perspective [15]. He viewed the cutting plane procedure as a recursive procedure for deriving all valid linear inequalities for the integer-hull of a polytope. He showed that Gomory’s systematic approach of adding cuts can be used to derive the integer hull within a finite number of rounds. This gave birth to the study of *closure properties* of polytopes under various cut choices.

Given the lack of a *Farkas Lemma* and a *duality theory* for IPs, it was believed that cutting planes could serve as a means to show integer infeasibility of polytopes efficiently. This led to the study of cutting plane proof techniques as a powerful proof technique in mathematical logic [18]. Researchers have since attempted to prove/disprove the efficiency of this proof technique by showing exponential lower bounds for particular cut choices [16]. In the most general case, Pudlák showed an exponential lower bound on the length of cutting plane proofs for any systematic cut-generation procedure [63].

4.1.1 Motivation

Despite the enormous history of work in this area, there are no polynomial bounds on the convergence of the cutting plane method even for specific problems. Gomory’s bound of 2^n remains the best known bound for general 0-1 integer programs. It is possible that this bound can be significantly improved for IPs with small Chvátal-Gomory rank [29]. A more

realistic possibility is that the approach is provably efficient for combinatorial optimization problems that are known to be solvable in polynomial time. An ideal candidate could be a problem that (a) has a polynomial-size IP-description (the LP-relaxation is polynomial-size), and (b) the convex-hull of integer solutions has a polynomial-time separation oracle.

Perhaps the first such interesting problem is minimum-cost perfect matching: *given a graph with costs on the edges, find a perfect matching of minimum total cost.*

The polyhedral characterization of matching was discovered by Edmonds [27]. Basic solutions of the linear program below (extreme points of the polytope) correspond to perfect matchings of the graph. We use $\delta(S)$ to denote the set of edges which have exactly one end point in S and $\delta(u)$ to denote $\delta(\{u\})$. Finally, $x(\delta(S)) = \sum_{uv \in \delta(S)} x(e)$.

$$\begin{aligned} \min \sum_{uv \in E} c(uv)x(uv) & \tag{P} \\ x(\delta(u)) = 1 \quad \forall u \in V & \tag{degree constraints} \\ x(\delta(S)) \geq 1 \quad \forall S \subsetneq V, |S| \text{ odd}, 3 \leq |S| \leq |V| - 3 & \tag{blossom inequalities} \\ x \geq 0 & \tag{nonnegativity constraints} \end{aligned}$$

The relaxation with only the degree and nonnegativity constraints, known as the *bipartite relaxation*, suffices to characterize the convex-hull of perfect matchings in bipartite graphs, and serves as a natural starting relaxation for the cutting plane approach. The inequalities corresponding to sets of odd cardinality greater than 1 are called *blossom* inequalities. These inequalities have Chvátal rank 1, i.e., applying one round of all possible Gomory cuts to the bipartite relaxation suffices to recover the perfect matching polytope of any graph. Moreover, although the number of blossom inequalities is exponential in the size of the graph, for any point not in the perfect matching polytope, a violated (blossom) inequality can be found in polynomial time [61]. This suggests a natural cutting plane algorithm (Figure 10), described by Padberg and Rao [61] and by Lovász and Plummer in their classic book on matching theory [54]. Grötschel and Holland [38], Trick [69], and Fischetti and Lodi [29] found experimental evidence that this algorithm converges quickly, but it has been an open problem to rigorously explain their findings.

1. Start with the bipartite relaxation.
2. While the current solution is fractional,
 - (a) Find a blossom inequality violated by the current solution and add it to the LP.
 - (b) Solve the new LP.

Figure 10: Cutting plane method for matchings

Most polynomial time algorithms are variants of Edmonds' primal-dual algorithm for weighted matching [27]. It is perhaps tempting to interpret this as a cutting plane algorithm, by adding cuts corresponding to the shrunk sets in the iterations of Edmonds' algorithm. However, there is no correspondence between the solution x of the LP given by non-negativity, degree constraints and a family \mathcal{F} of blossom inequalities, and the partial matching M in the iteration of Edmonds' algorithm when \mathcal{F} is the set of shrunk nodes. In particular, the next odd set S shrunk by Edmonds' algorithm might not even be a cut for x (i.e., $x(\delta(S)) \geq 1$). It is even possible that the bipartite relaxation already has an integer optimal solution, whereas Edmonds' algorithm proceeds by shrinking and unshrinking a long sequence of odd sets.

4.1.2 Results

In this chapter, we resolve the polynomial-time convergence of the cutting plane method for matchings. We show an implementation of the cutting plane method so that

1. every intermediate LP is defined by blossom inequalities corresponding to a laminar family of odd sets,
2. the optimum of every intermediate LP is half-integral and supported by a disjoint union of odd cycles and edges, and
3. the total number of iterations to arrive at a minimum-cost perfect matching for a n -vertex graph is $O(n \log n)$.

A collection of sets is called a *laminar* family if any two sets in the collection are either disjoint or nested.

Moreover, the set of inequalities used at each step can be identified by solving an LP of the same size as the current LP. To our knowledge, this is the first polynomial bound on the convergence of a cutting plane method (for matchings or in general). It is easy to verify that a laminar family of nontrivial odd sets may have at most $n/2$ members, hence every intermediate LP has at most $3n/2$ inequalities apart from the non-negativity constraints.

Organization. In Section 4.2, we review basic definitions and well-known properties about perfect matchings. In Section 4.3 we motivate a half-integrality conjecture and present our cutting plane implementation. In Section 4.4 we formulate a notion of factor-critical sets and fitting duals that play a central role in the analysis of our cutting plane algorithm. In Section 4.5 we prove properties about intermediate primal and dual solutions and analyze the running time of our algorithm.

4.2 Perfect Matchings and Linear Programs

Let $G = (V, E)$ be a graph, $c : E \rightarrow \mathbb{R}$ be a cost function on the edges and assume G has a perfect matching. The integer programming formulation of this problem is given by:

$$\begin{aligned} \min \quad & \sum_{uv \in E} c(uv)x(uv) \\ x(\delta(u)) &= 1 \quad \forall u \in V \\ x &\geq 0 \\ x &\in \mathbb{Z}^n \end{aligned}$$

It is well-known that basic feasible solutions x to the LP relaxation of this problem need not be integral (e.g., see Figure 11).

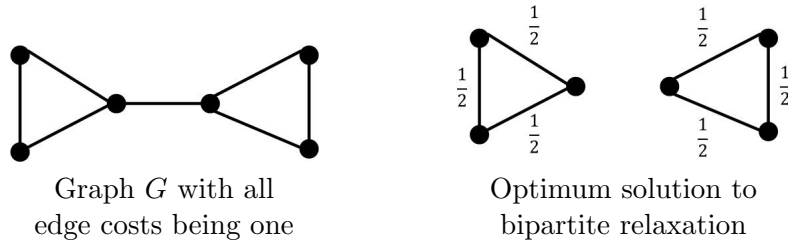


Figure 11: Non-integral Solution to Bipartite Relaxation

The LP relaxation is known as the bipartite relaxation. The bipartite relaxation suffices to characterize the convex-hull of perfect matchings in bipartite graphs, and serves as a natural starting relaxation. The bipartite relaxation and its dual are specified as follows.

$$\begin{array}{ll}
\min \sum_{uv \in E} c(uv)x(uv) & (P_0(G, c)) \\
x(\delta(u)) = 1 \quad \forall u \in V & \\
x \geq 0 &
\end{array}
\qquad
\begin{array}{ll}
\max \sum_{u \in V} \pi(u) & (D_0(G, c)) \\
\pi(u) + \pi(v) \leq c(uv) \quad \forall uv \in E &
\end{array}$$

We call a vector $x \in \mathbb{R}^E$ *proper-half-integral* if $x(e) \in \{0, 1/2, 1\}$ for every $e \in E$ and $\text{supp}(x)$ is a disjoint union of edges and odd cycles. The bipartite relaxation of any graph has the following well-known property.

Proposition 4.2.1. *Every basic feasible solution x of $P_0(G, c)$ is proper-half-integral.*

The polyhedral characterization of the matching problem was discovered by Edmonds [27] as an exponential-sized LP. Let \mathcal{O} be the set of all odd subsets of V of size at least 3, and let \mathcal{V} denote the set of one element subsets of V . Edmonds showed that the following primal program characterizes minimum-cost perfect matching, i.e., the solution to the primal problem is necessarily integral. Here, we give the primal and dual program.

$$\begin{array}{ll}
\min \sum_{uv \in E} c(uv)x(uv) & (P(G, c)) \\
x(\delta(u)) = 1 \quad \forall u \in V & \\
x(\delta(S)) \geq 1 \quad \forall S \in \mathcal{O} & \\
x \geq 0 &
\end{array}
\qquad
\begin{array}{ll}
\max \sum_{S \in \mathcal{V} \cup \mathcal{O}} \Pi(S) & (D(G, c)) \\
\sum_{S \in \mathcal{V} \cup \mathcal{O}: uv \in \delta(S)} \Pi(S) \leq c(uv) \quad \forall uv \in E & \\
\Pi(S) \geq 0 \quad \forall S \in \mathcal{O} &
\end{array}$$

Let $\mathcal{F} \subset \mathcal{O}$ be a laminar family of odd sets of cardinality at least 3. All basic feasible solutions to $P(G, c)$ are integral. Moreover, any basic solution can be fully defined by using only a laminar collection of tight inequalities (these will generate all the other tight inequalities) [20].

4.3 A Cutting Plane Algorithm for Perfect Matching

In this section we present the cutting plane algorithm for perfect matching. Before presenting our algorithm, we elaborate on the significance of half-integral primal solutions to generate faster cuts in Section 4.3.1 and justify the choice of dropping cuts that were added in earlier rounds in Section 4.3.2.

4.3.1 Half-integrality for faster cut-generation

For arbitrary graphs, the bipartite relaxation has the nice property that any basic solution is half-integral and thus its support is a disjoint union of edges and odd cycles. This makes it particularly easy to find violated blossom inequalities – any odd component of the support gives one. This is also the simplest heuristic that is employed in the implementations [38, 69] for finding violated blossom inequalities. However, if we have a fractional solution in a later phase, there is no guarantee that we can find an odd connected component whose blossom inequality is violated, and therefore sophisticated and significantly slower separation methods are needed for finding cutting planes, e.g., the Padberg-Rao procedure [61].

At first sight, maintaining half-integrality seems to be impossible. Figure 12 shows an example where the starting solution consists of two odd cycles. There is only one reasonable way to impose cuts, and it leads to a non half-integral basic feasible solution. Observe however, that in the example, the bipartite relaxation also has an integer optimal solution. The problem here is the existence of multiple basic optimal solutions. To avoid such degeneracy, we will ensure that all linear systems that we encounter have unique optimal solutions.

This uniqueness can be achieved by a simple deterministic perturbation of the integer cost function, which increases the input size polynomially. Given an integer cost function $c : E \rightarrow \mathbb{Z}$ on the edges, let us define the perturbation \tilde{c} by ordering the edges arbitrarily, and increasing the cost of edge number i by $1/2^i$.

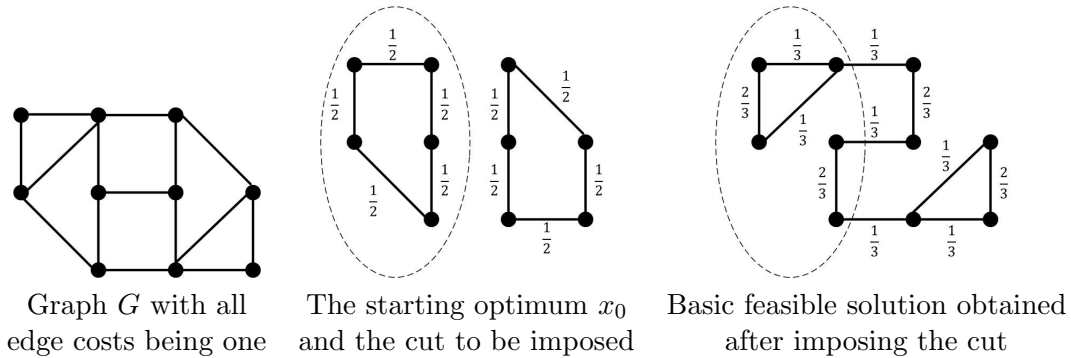


Figure 12: Counterexample to the half-integrality conjecture.

4.3.2 Retain cuts with positive dual values

The main difficulty that we have to overcome is the following: ensuring unique optimal solutions does not suffice to maintain half-integrality of optimal solutions upon adding any sequence of blossom inequalities (or even adding a laminar family of blossom inequalities). In fact, these inequalities have to be chosen with some care and it is also crucial that we eliminate certain older ones that are no longer useful. Eliminating cutting planes that have zero dual values in any later iteration is common in most implementations of the cutting plane algorithm; although this is done mainly to keep the number of inequalities from blowing up, another justification is that a cut with zero dual value may not be a facet contributing to the current LP optimum.

At any iteration, inequalities that are tight for the current optimal solution are natural candidates for retaining in the next iteration while the new inequalities are determined by odd cycles in the support of the current optimal solution. However, it turns out that keeping all tight inequalities does not maintain half-integrality. Our main algorithmic insight is that the choice of cuts for the next iteration can be determined by examining optimal dual solutions to the current LP. We use a restricted type of dual optimal solution (later called *positively-fitting dual*) that can be computed either by solving a single LP of the same complexity or combinatorially. Moreover, we also ensure that the set of cuts imposed in any LP are laminar and correspond to blossom inequalities.

4.3.3 Algorithm C-P-Matching

The intermediate LPs in our cutting plane algorithm will consist of bipartite relaxation constraints and blossom inequalities corresponding to some laminar family.

$$\begin{array}{ll}
\min \sum_{uv \in E} c(uv)x(uv) & (P_{\mathcal{F}}(G, c)) \\
x(\delta(u)) = 1 \quad \forall u \in \mathcal{V} & \\
x(\delta(S)) \geq 1 \quad \forall S \in \mathcal{F} & \\
x \geq 0 & \\
\max \sum_{S \in \mathcal{V} \cup \mathcal{F}} \Pi(S) & (D_{\mathcal{F}}(G, c)) \\
\sum_{S \in \mathcal{V} \cup \mathcal{F}: uv \in \delta(S)} \Pi(S) \leq c(uv) \quad \forall uv \in E & \\
\Pi(S) \geq 0 \quad \forall S \in \mathcal{F} &
\end{array}$$

We will use $\Pi(v)$ to denote $\Pi(\{v\})$ for dual solutions. For an arbitrary laminar family $\mathcal{F} \subseteq \mathcal{O}$, basic feasible solutions to $P_{\mathcal{F}}(G, c)$ can be half-integral but non-proper (i.e., the optimum could contain even cycles).

Assume we are given a dual feasible solution Γ to $D_{\mathcal{F}}(G, c)$. We say that a dual optimal solution Π to $D_{\mathcal{F}}(G, c)$ is Γ -*extremal*, if it minimizes

$$h(\Pi, \Gamma) = \sum_{S \in \mathcal{V} \cup \mathcal{F}} \frac{|\Pi(S) - \Gamma(S)|}{|S|}$$

among all dual optimal solutions Π . A Γ -extremal dual optimal solution can be found by solving a single LP if we are provided with the primal optimal solution to $P_{\mathcal{F}}(G, c)$ (see Section 4.5.3.3).

The cutting plane implementation that we propose is shown in Figure 13. From the previous set of cuts, we keep only those which have a positive value in an extremal dual optimal solution; let \mathcal{H}' denote their set. We add the new cut set \mathcal{H}'' based on odd cycles in the support of the current solution. However, in order to maintain laminarity of the cut family, we do not add the vertex sets of these cycles but instead their union with all the sets in \mathcal{H}' that they intersect. We shall show that these unions are also odd sets and thus give blossom inequalities. We note that in the first iteration, there is no need to solve the dual LP as \mathcal{F} will be empty.

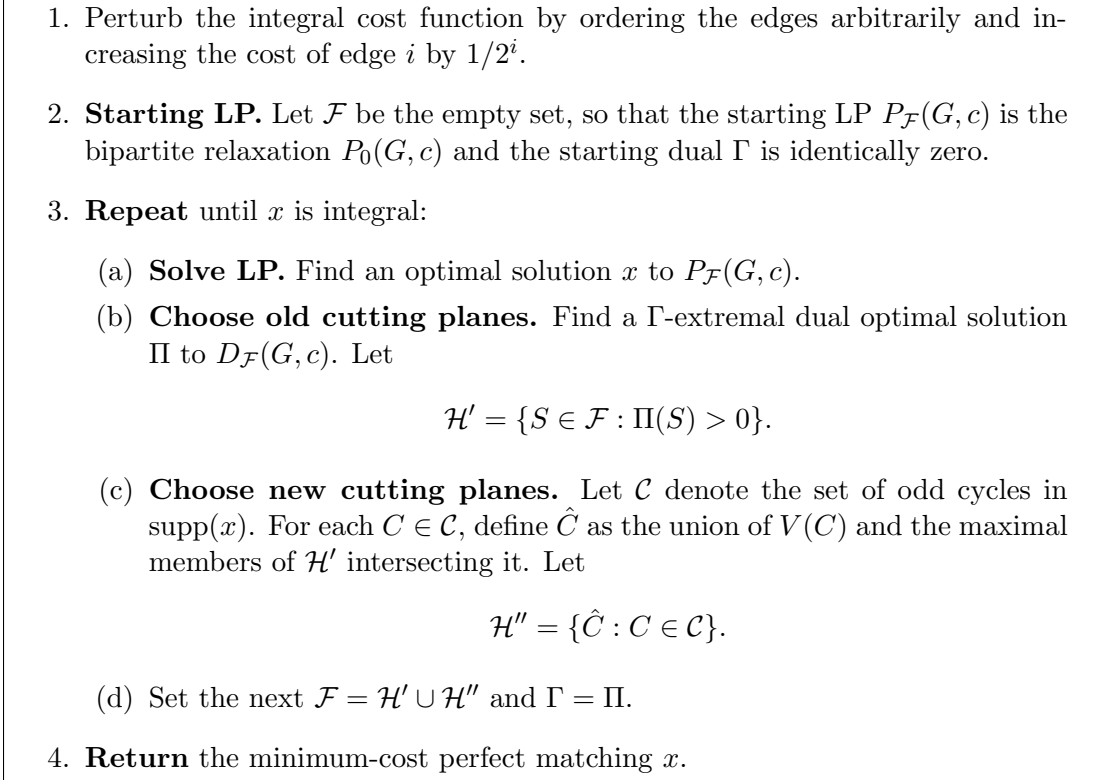


Figure 13: Algorithm C-P-Matching

4.4 Factor-critical matchings

In what follows, we formulate a notion of factor-critical sets and factor-critical duals, that play a central role in the analysis of our algorithm and are extensions of concepts central to the analysis of Edmonds' algorithm.

Let $H = (V, E)$ be a graph and \mathcal{F} be a laminar family of subsets of V . We say that an edge set $M \subseteq E$ is an \mathcal{F} -*matching*, if it is a matching, and for any $S \in \mathcal{F}$, $\delta_M(S) \leq 1$. For a set $S \subseteq V$, we call a set M of edges to be an (S, \mathcal{F}) -*perfect-matching* if it is an \mathcal{F} -matching covering precisely the vertex set S .

A set $S \in \mathcal{F}$ is defined to be \mathcal{F} -*factor-critical* in H , if for every node $u \in S$, there exists an $(S \setminus \{u\}, \mathcal{F})$ -perfect-matching using the edges of H . Given a laminar family \mathcal{F} and a feasible solution Π to $D_{\mathcal{F}}(G, c)$, let $G_{\Pi} = (V, E_{\Pi})$ denote the graph of tight edges. A set $S \in \mathcal{F}$ is (Π, \mathcal{F}) -*factor-critical* if S is \mathcal{F} -factor-critical in G_{Π} . The corresponding matching M_u is called the Π -*critical-matching* for u in S . If \mathcal{F} is clear from the context, then we simply call the set S to be Π -factor-critical.

A feasible solution Π to $D_{\mathcal{F}}(G, c)$ is an \mathcal{F} -fitting dual, if every $S \in \mathcal{F}$ is (Π, \mathcal{F}) -factor-critical, and $\Pi(Z) > 0$ for every non-maximal member of \mathcal{F} . A family $\mathcal{F} \subseteq \mathcal{O}$ is called a *proper odd family*, if \mathcal{F} is laminar, and there exists an \mathcal{F} -fitting dual solution. This will be a crucial notion: the set of cuts imposed in every iteration of the cutting plane algorithm will be a proper odd family. The following observation provides some context and motivation for these definitions.

Proposition 4.4.1. *Let \mathcal{F} be the set of contracted sets at some stage of Edmonds' matching algorithm. Then the corresponding dual solution Π in the algorithm is an \mathcal{F} -fitting-dual. \square*

We call Π to be an \mathcal{F} -positively-fitting dual, if Π is a feasible solution to $D_{\mathcal{F}}(G, c)$, and every $S \in \mathcal{F}$ such that $\Pi(S) > 0$ is (Π, \mathcal{F}) -factor-critical. Clearly, every \mathcal{F} -fitting dual is also an \mathcal{F} -positively-fitting, but the converse is not true. This property will be satisfied by the extremal dual optimal solutions found in every iteration.

We conclude this section by summarizing elementary properties of Π -critical matchings.

Lemma 4.4.2. *Let \mathcal{F} be a laminar odd family, Π a feasible solution to $D_{\mathcal{F}}(G, c)$, and let $S \in \mathcal{F}$ be a (Π, \mathcal{F}) -factor-critical set. For $u, v \in S$, let M_u, M_v be the Π -critical-matchings for u, v respectively.*

(i) For every $T \in \mathcal{F}$ such that $T \subsetneq S$,

$$|M_u \cap \delta(T)| = \begin{cases} 1 & \text{if } u \in S \setminus T, \\ 0 & \text{if } u \in T. \end{cases}$$

(ii) Assume the symmetric difference of M_u and M_v contains an even cycle C . Then the symmetric difference $M_u \Delta C$ is also a Π -critical matching for u .

Proof. (i) M_u is a perfect matching of $S \setminus \{u\}$, hence for every $T \subsetneq S$,

$$|M_u \cap \delta(T)| \equiv |T \setminus \{u\}| \pmod{2}.$$

By definition of M_u , $|M_u \cap \delta(T)| \leq 1$ for any $T \subsetneq S$, $T \in \mathcal{F}$, implying the claim.

(ii) Let $M' = M_u \Delta C$. First observe that since C is an even cycle, $u, v \notin V(C)$. Hence M' is a perfect matching on $S \setminus \{u\}$ using only tight edges w.r.t. Π . It remains to show that

$|M' \cap \delta(T)| \leq 1$ for every $T \in \mathcal{F}$, $T \subsetneq S$. Let γ_u and γ_v denote the number of edges in $C \cap \delta(T)$ belonging to M_u and M_v , respectively. Since these are critical matchings, we have $\gamma_u, \gamma_v \leq 1$. On the other hand, since C is a cycle, $|C \cap \delta(T)|$ is even and hence $\gamma_u + \gamma_v = |C \cap \delta(T)|$ is even. These imply that $\gamma_u = \gamma_v$. The claim follows since $|M' \cap \delta(T)| = |M_u \cap \delta(T)| - \gamma_u + \gamma_v$. \square

4.5 Analysis

Our analysis to show half-integrality is based on the intimate relationship that matchings have with factor-critical graphs (deleting any node leaves the graph with a perfect matching): for example, contracted sets are factor-critical both in the unweighted [28] and weighted [27] matching algorithms by Edmonds. The definitions of factor-criticality given in Section 4.4 help establish the existence of positive-fitting duals. This in conjunction with the following *uniqueness* property is needed to prove the existence of a proper-half-integral solution in each step. We require that the cost function $c : E \rightarrow \mathbb{R}$ satisfies:

For every proper odd family \mathcal{F} , $P_{\mathcal{F}}(G, c)$ has a unique optimal solution. (*)

In Section 4.5.4, we show that arbitrary integer cost function can be perturbed to achieve this property.

We use the number of odd cycles in the support of an optimal half-integral solution as a potential function to show convergence. We first show that $\text{odd}(x_{i+1}) \leq \text{odd}(x_i)$, where x_i, x_{i+1} are consecutive optimal LP solutions, and $\text{odd}(\cdot)$ is the number of odd cycles in the support. We further show that the cuts added in iterations where $\text{odd}(x_i)$ does not decrease continue to be retained until $\text{odd}(x_i)$ decreases. Since the maximum size of a laminar family of nontrivial odd sets is $n/2$, we get a bound of $O(n \log n)$ on the number of iterations.

The proof of the potential function behavior is quite intricate. It proceeds by designing a *half-integral* version of Edmonds primal-dual algorithm for minimum-weight perfect matching, and arguing that the optimal solution to the extremal dual LP must correspond to the one found by this primal-dual algorithm. We emphasize that this algorithm is used only in the analysis. Nevertheless, it is rather remarkable that even for analyzing the cutting plane approach, comparison with a new extension of Edmonds' classic algorithm provides the answer.

4.5.1 Overview and Organization

The proof of efficient convergence of our cutting plane algorithm is established in two parts. In the first part, we show that half-integrality of the intermediate optimum solutions is guaranteed by the existence of an \mathcal{F} -positively-fitting dual optimal solution to $D_{\mathcal{F}}(G, c)$. In Section 4.5.2, we show that if x is the unique optimal solution to $P_{\mathcal{F}}(G, c)$ and if there exists an \mathcal{F} -positively-fitting dual optimal solution, then x is proper-half-integral.

This proper-half-integral property is shown using a basic contraction operation. Given Π , an \mathcal{F} -positively-fitting dual optimal solution for the laminar odd family \mathcal{F} , contracting every set $S \in \mathcal{F}$ with $\Pi(S) > 0$ preserves primal and dual optimal solutions (similar to Edmonds' primal-dual algorithm). This is shown in Lemma 4.5.1. Moreover, if we had a unique primal optimal solution x to $P_{\mathcal{F}}(G, c)$, its image x' in the contracted graph is the unique optimal solution; if x' is proper-half-integral, then so is x . Now, half-integrality of the optimum follows: we contract all maximal sets $S \in \mathcal{F}$ with $\Pi(S) > 0$. The image x' of the unique optimal solution x is the unique optimal solution to the bipartite relaxation in the contracted graph, and consequently, half-integral.

\mathcal{F} -positively-fitting dual optimal solutions are hence quite helpful, but their existence is far from obvious. We next show that if \mathcal{F} is a proper odd family, then the extremal dual optimal solutions found in the algorithm are in fact \mathcal{F} -positively-fitting dual optimal solutions. Furthermore, the next set of cuts is a proper odd family.

In order to show that a proper odd family \mathcal{F} always admits an \mathcal{F} -positively-fitting dual optimum, and that every extremal dual solution satisfies this property, we need a deeper understanding of the structure of dual optimal solutions. Section 4.5.3 is dedicated to this analysis. Let Π be an \mathcal{F} -fitting dual solution, and Ψ an arbitrary dual optimal solution to $D_{\mathcal{F}}(G, c)$. The main Lemma 4.5.10 shows the following relation between Ψ and Π inside $S \in \mathcal{F}$, if S is tight for a primal optimal solution x . Let $\Pi_S(u)$ and $\Psi_S(u)$ denote the sum of the dual values of sets containing u that are strictly contained inside S in solutions Π and Ψ respectively, and let $\Delta = \max_{u \in S} (\Pi_S(u) - \Psi_S(u))$. We show that every edge in $\text{supp}(x) \cap \delta(S)$ is incident to a node $u \in S$ such that $\Pi_S(u) - \Psi_S(u) = \Delta$. We also show (Lemma 4.5.12) that if $S \in \mathcal{F}$ is both Π - and Ψ -factor-critical, then Π and Ψ are identical

inside S .

If $\Psi(S) > 0$ but S is not Ψ -factor-critical, the above property (called *consistency* later) enables us to modify Ψ by moving towards Π inside S , and decreasing $\Psi(S)$ so that optimality is maintained. Thus, we either get that Ψ and Π are identical inside S thereby making S to be Ψ -factor-critical or $\Psi(S) = 0$. A sequence of such operations converts an arbitrary dual optimal solution to an \mathcal{F} -positively-fitting dual optimal one, leading to a combinatorial procedure to obtain positively-fitting dual optimal solutions (Section 4.5.3.2). Moreover, such operations decrease the secondary objective value $h(\Psi, \Pi)$ and thus show that every Π -extremal dual optimum is also an \mathcal{F} -positively-fitting dual optimum.

As a consequence of every intermediate set of cuts being a proper odd family and uniqueness of primal optimal solutions, we can guarantee that the primal optimal solutions obtained during the execution of the algorithm are proper-half-integral. In the second part of our analysis, we show efficient convergence by considering the number of odd cycles, $\text{odd}(x)$, in the support of the current primal optimal solution x . We first show that this quantity is non-increasing during the algorithm. This is already true if we choose an arbitrary \mathcal{F} -positively-fitting dual optimal solution Π in each iteration of the algorithm. In order to show that the number of cycles cannot remain the same and has to strictly decrease within a polynomial number of iterations, we need the more specific choice of extremal duals.

Suppose the number of odd cycles does not decrease between iterations i and j . Then, with the choice of extremal duals to retain cuts, we show that all cuts added between iterations i and j are retained until round $j + 1$. The main ingredient to prove this progress is by coupling our intermediate primal and dual solutions with the solutions of a *Half-integral Primal-Dual* algorithm for half-integral matching, a variation of Edmonds' primal-dual weighted matching algorithm that we design for this purpose. We assume familiarity with Edmonds' algorithm [27]. We describe only one phase of this algorithm needed by our argument. However, it can be easily extended to a strongly polynomial combinatorial algorithm that finds a minimum cost perfect matching starting from an optimal solution to the bipartite relaxation, and maintaining a proper half integral solution in every intermediate phase. In contrast, Edmonds' algorithm always maintains an integral matching, starting

from the empty set, and gradually extending the set of nodes covered by the matching.

The Half-integral Primal-Dual algorithm starts from an optimal proper-half-integral partial matching x in G , leaving a set W of nodes exposed. Further, we have a dual Π on a laminar support $\mathcal{V} \cup \mathcal{K}$ with $\mathcal{K} \subseteq \mathcal{O}$; x and Π satisfy primal-dual slackness conditions. The algorithm transforms x to a proper-half-integral perfect matching and Π to a dual solution with support contained in $\mathcal{V} \cup \mathcal{K}$, satisfying complementary slackness. We sketch how we apply the Half-integral Primal-Dual algorithm to show convergence.

Let us consider two consecutive primal solutions x_i and x_{i+1} , with duals Π_i and Π_{i+1} . We contract every set $S \in \mathcal{O}$ with $\Pi_{i+1}(S) > 0$; let \hat{G} be the resulting graph. As we have seen by Lemma 4.5.4 the image x'_{i+1} of x_{i+1} is the unique optimal solution to the bipartite relaxation in \hat{G} . The image x'_i of x_i is proper-half-integral in \hat{G} with some exposed nodes W ; let Π'_i be the image of Π_i . Every exposed node in W corresponds to a cycle in $\text{supp}(x_i)$. We start the Half-integral Primal-Dual in \hat{G} with the solutions x'_i and Π'_i , and we prove that it must terminate with the primal solution x'_{i+1} . The analysis of the Half-integral Primal-Dual algorithm reveals that the total number of exposed nodes and odd cycles does not increase; this will imply that the number of odd cycles in the cutting plane algorithm does not increase.

To prove convergence, we show that if the number of cycles does not decrease between phases i and $i + 1$, then the Half-integral Primal-Dual algorithm also terminates with the extremal dual optimal solution Π'_{i+1} . This enables us to couple the performance of Half-integral Primal-Dual between phases i and $i + 1$ and between $i + 1$ and $i + 2$: the (alternating forest) structure built in the former iteration carries over to the latter one. As a consequence, all cuts added in iteration i will be imposed in all subsequent phases until the number of odd cycles decreases.

Thus, during each iteration of adding cuts in our algorithm either the number of cycles decreases or the size of the laminar family of cuts increases. Since the size of a laminar family is $O(n)$, polynomial convergence follows.

4.5.2 Contractions and Half-Integrality

We define an important contraction operation and derive some fundamental properties. Let \mathcal{F} be a laminar odd family, Π be a feasible solution to $D_{\mathcal{F}}(G, c)$, and let $S \in \mathcal{F}$ be a (Π, \mathcal{F}) -factor-critical set. Let us define

$$\Pi_S(u) := \sum_{T \in \mathcal{V} \cup \mathcal{F}: T \not\subseteq S, u \in T} \Pi(T)$$

to be the total dual contribution of sets inside S containing u .

By contracting S w.r.t. Π , we mean the following: Let $G' = (V', E')$ be the contracted graph on node set $V' = (V \setminus S) \cup \{s\}$, s representing the contraction of S . Let \mathcal{V}' denote the set of one-element subsets of V' . For a set $T \subseteq V$, let T' denote its contracted image. Let \mathcal{F}' be the set of nonsingular images of the sets of \mathcal{F} , that is, $T' \in \mathcal{F}'$ if $T \in \mathcal{F}$, and $T' \setminus \{s\} \neq \emptyset$. Let E' contain all edges $uv \in E$ with $u, v \notin S$ and for every edge uv with $u \in S, v \in V - S$ add an edge sv . Let us define the image Π' of Π to be $\Pi'(T') = \Pi(T)$ for every $T' \in \mathcal{V}' \cup \mathcal{F}'$ and the image x' of x to be $x'(u'v') = x(uv)$. Define the new edge costs

$$c'(u'v') = \begin{cases} c(uv) & \text{if } uv \in E[V \setminus S], \\ c(uv) - \Pi_S(u) & \text{if } u \in S, v \in V \setminus S. \end{cases}$$

Lemma 4.5.1. *Let \mathcal{F} be a laminar odd family, x be an optimal solution to $P_{\mathcal{F}}(G, c)$, Π be a feasible solution to $D_{\mathcal{F}}(G, c)$. Let S be a (Π, \mathcal{F}) -factor-critical set, and let G', c', \mathcal{F}' denote the graph, costs and laminar family respectively obtained by contracting S w.r.t. Π and let x', Π' be the images of x, Π respectively. Then the following hold.*

- (i) Π' is a feasible solution to $D_{\mathcal{F}'}(G', c')$. Furthermore, if a set $T \in \mathcal{F}$, $T \setminus S \neq \emptyset$ is (Π, \mathcal{F}) -factor-critical, then its image T' is (Π', \mathcal{F}') -factor-critical.
- (ii) Suppose Π is an optimal solution to $D_{\mathcal{F}}(G, c)$ and $x(\delta(S)) = 1$. Then x' is an optimal solution to $P_{\mathcal{F}'}(G', c')$ and Π' is optimal to $D_{\mathcal{F}'}(G', c')$.
- (iii) If x is the unique optimum to $P_{\mathcal{F}}(G, c)$, and Π is an optimal solution to $D_{\mathcal{F}}(G, c)$, then x' is the unique optimum to $P_{\mathcal{F}'}(G', c')$. Moreover, x' is proper-half-integral if and only if x is proper-half-integral. Further, assume C' is an odd cycle in $\text{supp}(x')$

and let T be the pre-image of $V(C')$ in G . Then, $\text{supp}(x)$ inside T consists of an odd cycle and matching edges.

Proof. (i) For feasibility, it is sufficient to verify

$$\sum_{T' \in \mathcal{V}' \cup \mathcal{F}': u'v' \in \delta(T')} \Pi'(T') \leq c'(u'v') \quad \forall u'v' \in E'.$$

If $u, v \neq s$, this is immediate from feasibility of Π to $D_{\mathcal{F}_i}(G, c)$. Consider an edge $sv' \in E(G')$. Let uv be the pre-image of this edge.

$$\sum_{T' \in \mathcal{V}' \cup \mathcal{F}': sv' \in \delta(T')} \Pi'(T') = \Pi(S) + \sum_{T \in \mathcal{F}: uv \in \delta(T), T \setminus S \neq \emptyset} \Pi(T) \leq c(uv) - \Pi_S(u) = c'(sv').$$

We also observe that $u'v'$ is tight in G' w.r.t Π' if and only if the pre-image uv is tight in G w.r.t Π .

Let $T \in \mathcal{F}$ be (Π, \mathcal{F}) -factor-critical with $T \setminus S \neq \emptyset$. Now, it is sufficient to verify that T' is (Π', \mathcal{F}') -factor-critical whenever T contains S . Let $u' \in T'$ be the image of u and consider the projection M' of the Π -critical-matching M_u . The support of this projection is contained in the tight edges with respect to Π' . Let $Z' \subsetneq T'$, $Z' \in \mathcal{F}'$ and let Z be the pre-image of Z' . If $u' \neq s$, then $|M' \cap \delta(Z')| = |M_u \cap \delta(Z)| \leq 1$ and since $|M_u \cap \delta(S)| = 1$ by Lemma 4.4.2, the matching M' is a $(T' \setminus \{u'\}, \mathcal{F}')$ -perfect-matching. If $u' = s$, then $u \in S$. By Lemma 4.4.2, $M_u \cap \delta(S) = \emptyset$ and hence, M' misses s . Also, $|M_u \cap \delta(Z)| \leq 1$ implies $|M' \cap \delta(Z')| \leq 1$ and hence M' is a $(T' \setminus \{s\}, \mathcal{F}')$ -perfect-matching.

(ii) Since $x(\delta(S)) = 1$, we have $x'(\delta(v)) = 1$ for every $v \in V'$. It is straightforward to verify that $x'(\delta(T')) \geq 1$ for every $T' \in \mathcal{F}'$ with equality if $x(\delta(T)) = 1$. Thus, x' is feasible to $P_{\mathcal{F}'}(G', c')$. Optimality follows as x' and Π' satisfy complementary slackness, using that the image of tight edges is tight, as shown by the argument for part (i).

(iii) For uniqueness, consider an arbitrary optimal solution y' to $P_{\mathcal{F}'}(G', c')$. Let M_u be the Π -critical matching for u in S . Define $\alpha_u = y'(\delta(u))$ for every $u \in S$, i.e., α_u is the total y' value on the images of edges $uv \in E$ with $v \in V - S$. Take $w = \sum_{u \in S} \alpha_u M_u$ and

$$y(uv) = \begin{cases} y'(u'v') & \text{if } uv \in E \setminus E[S], \\ w(uv) & \text{if } uv \in E[S]. \end{cases}$$

Then, y is a feasible solution to $P_{\mathcal{F}}(G, c)$ and y satisfies complementary slackness with Π . Hence, y is an optimum to $P_{\mathcal{F}}(G, c)$ and thus by uniqueness, we get $y = x$. Consequently, $y' = x'$.

The above argument also shows that x must be identical to w inside S . Suppose x' is proper-half-integral. First, assume s is covered by a matching edge in x' . Then $\alpha_u = 1$ for some $u \in S$ and $\alpha_v = 0$ for every $v \neq u$. Consequently, $w = M_u$ is a perfect matching on $S - u$. Next, assume s is incident to an odd cycle in x' . Then $\alpha_{u_1} = \alpha_{u_2} = 1/2$ for some nodes $u_1, u_2 \in S$, and $w = \frac{1}{2}(M_{u_1} + M_{u_2})$. The uniqueness of x implies the uniqueness of both M_{u_1} and M_{u_2} . Then by Lemma 4.4.2(ii), the symmetric difference of M_{u_1} and M_{u_2} may not contain any even cycles. Hence, $\text{supp}(w)$ contains an even path between u_1 and u_2 , and some matching edges. Consequently, x is proper-half-integral. The above argument immediately shows the following.

Claim 4.5.2. *Let C' be an odd (even) cycle such that $x'(e) = 1/2$ for every $e \in C'$ in $\text{supp}(x')$ and let T be the pre-image of the set $V(C')$ in G . Then, $\text{supp}(x) \cap E[T]$ consists of an odd (even) cycle C and a (possibly empty) set M of edges such that $x(e) = 1/2 \forall e \in C$ and $x(e) = 1 \forall e \in M$. \square*

Next, we prove that if x is proper-half-integral, then so is x' . It is clear that x' being the image of x is half-integral. If x' is not proper-half-integral, then $\text{supp}(x')$ contains an even $1/2$ -cycle, and thus by Claim 4.5.2, $\text{supp}(x)$ must also contain an even cycle, contradicting that it was proper-half-integral. Finally, if C' is an odd cycle in $\text{supp}(x')$, then Claim 4.5.2 provides the required structure for x inside T . \square

Corollary 4.5.3. *Assume x is the optimal solution to $P_{\mathcal{F}}(G, c)$ and there exists an \mathcal{F} -positively-fitting dual optimum Π . Let \hat{G}, \hat{c} be the graph, and cost obtained by contracting all maximal sets $S \in \mathcal{F}$ with $\Pi(S) > 0$ w.r.t. Π , and let \hat{x} be the image of x in \hat{G} .*

(i) \hat{x} and $\hat{\Pi}$ are the optimal solutions to the bipartite relaxation $P_0(\hat{G}, \hat{c})$ and $D_0(\hat{G}, \hat{c})$ respectively.

(ii) If x is the unique optimum to $P_{\mathcal{F}}(G, c)$, then \hat{x} is the unique optimum to $P_0(\hat{G}, \hat{c})$. If

\hat{x} is proper-half-integral, then x is also proper-half-integral.

We now prove the main lemma of this section.

Lemma 4.5.4. *Let \mathcal{F} be a laminar odd family and assume $P_{\mathcal{F}}(G, c)$ has a unique optimal solution x . If there exists an \mathcal{F} -positively-fitting dual optimal solution, then x is proper-half-integral.*

Proof. Let Π be an \mathcal{F} -positively-fitting dual optimum, and let x be the unique optimal solution to $P_{\mathcal{F}}(G, c)$. Contract all maximal sets $S \in \mathcal{F}$ with $\Pi(S) > 0$, obtaining the graph \hat{G} and cost \hat{c} . Let \hat{x} be the image of x in \hat{G} . By Corollary 4.5.3(ii), \hat{x} is unique optimum to $P_0(\hat{G}, \hat{c})$. By Proposition 4.2.1, \hat{x} is proper-half-integral and hence by Corollary 4.5.3(ii), x is also proper-half-integral. \square

4.5.3 Structure of Dual Solutions

In this section, we show two properties about positively-fitting dual optimal solutions – (1) an optimum Ψ to $D_{\mathcal{F}}(G, c)$ can be transformed into an \mathcal{F} -positively-fitting dual optimal (Section 4.5.3.2) if \mathcal{F} is a proper odd family and (2) a Γ -extremal dual optimal solution to $D_{\mathcal{F}}(G, c)$ as obtained in the algorithm is also an \mathcal{F} -positively-fitting dual optimal solution (Section 4.5.3.3). In Section 4.5.3.1, we first show some lemmas characterizing arbitrary dual optimal solutions.

4.5.3.1 Consistency of Dual Solutions

Assume $\mathcal{F} \subseteq \mathcal{O}$ is a proper odd family, with Π being an \mathcal{F} -fitting dual solution, and let Ψ be an arbitrary dual optimal solution to $D_{\mathcal{F}}(G, c)$. Note that optimality of Π is not assumed. Let x be an optimal solution to $P_{\mathcal{F}}(G, c)$; we do not assume uniqueness in this section. We shall describe structural properties of Ψ compared to Π ; in particular, we show that if we contract a Π -factor-critical set S , the images of x and Ψ will be primal and dual optimal solutions in the contracted graph.

Consider a set $S \in \mathcal{F}$. We say that the dual solutions Π and Ψ are *identical* inside S , if $\Pi(T) = \Psi(T)$ for every set $T \subsetneq S$, $T \in \mathcal{F} \cup \mathcal{V}$. We defined $\Pi_S(u)$ in the previous section;

we also use this notation for Ψ , namely, let $\Psi_S(u) := \sum_{T \in \mathcal{V} \cup \mathcal{F}: T \subseteq S, u \in T} \Psi(T)$ for $u \in S$. Let us now define

$$\Delta_{\Pi, \Psi}(S) := \max_{u \in S} (\Pi_S(u) - \Psi_S(u)).$$

We say that Ψ is *consistent* with Π inside S , if $\Pi_S(u) - \Psi_S(u) = \Delta_{\Pi, \Psi}(S)$ holds for every $u \in S$ that is incident to an edge $uv \in \delta(S) \cap \text{supp}(x)$. Consistency is important as it enables us to preserve optimality when contracting a set $S \in \mathcal{F}$ w.r.t. Π . Assume Ψ is consistent with Π inside S , and $x(\delta(S)) = 1$. Let us contract S w.r.t. Π to obtain G' and c' as defined in Section 4.5.2. Define

$$\Psi'(T') = \begin{cases} \Psi(T) & \text{if } T' \in (\mathcal{F}' \cup \mathcal{V}') \setminus \{s\}, \\ \Psi(S) - \Delta_{\Pi, \Psi}(S) & \text{if } T' = \{s\} \end{cases}$$

Lemma 4.5.5. *Let $\mathcal{F} \subseteq \mathcal{O}$ be a proper odd family, with Π being an \mathcal{F} -fitting dual solution and let Ψ be an optimal solution to $D_{\mathcal{F}}(G, c)$. Let x be an optimal solution to $P_{\mathcal{F}}(G, c)$. Suppose Ψ is consistent with Π inside $S \in \mathcal{F}$ and $x(\delta(S)) = 1$. Let G', c', \mathcal{F}' denote the graph, costs and laminar family obtained by contraction. Then the image x' of x is an optimum to $P_{\mathcal{F}'}(G', c')$, and Ψ' (as defined above) is an optimum to $D_{\mathcal{F}'}(G', c')$.*

Proof. Feasibility of x' follows as in the proof of Lemma 4.5.1(ii). For the feasibility of Ψ' , we have to verify $\sum_{T' \in \mathcal{V}' \cup \mathcal{F}': uv \in \delta(T')} \Psi'(T') \leq c'(uv)$ for every edge $uv \in E(G')$. This follows immediately for every edge uv such that $u, v \neq s$ since Ψ is a feasible solution for $D_{\mathcal{F}}(G, c)$. Consider an edge $uv \in E(G)$, $u \in S$. Let $sv \in E(G')$ be the image of uv in G' , and let $\Delta = \Delta_{\Pi, \Psi}(S)$.

$$\begin{aligned} c(uv) &\geq \sum_{T \in \mathcal{V} \cup \mathcal{F}: uv \in \delta(T)} \Psi(T) \\ &= \Psi_S(u) + \Psi(S) + \sum_{T \in \mathcal{F}: uv \in \delta(T), T-S \neq \emptyset} \Psi(T) \\ &= \Psi_S(u) + \Delta + \sum_{T' \in \mathcal{V}' \cup \mathcal{F}': sv \in \delta(T')} \Psi'(T'). \end{aligned}$$

In the last equality, we used the definition $\Psi'(s) = \Psi(S) - \Delta$. Therefore, using $\Pi_S(u) \leq$

$\Psi_S(u) + \Delta$, we obtain

$$\sum_{T' \in \mathcal{V}' \cup \mathcal{F}': sv \in \delta(T')} \Psi'(T') \leq c(uv) - \Psi_S(u) - \Delta \leq c(uv) - \Pi_S(u) = c'(uv). \quad (9)$$

Thus, Ψ' is a feasible solution to $D_{\mathcal{F}'}(G', c')$. To show optimality, we verify complementary slackness for x' and Ψ' . If $x'(uv) > 0$ for $u, v \neq s$, then $x(uv) > 0$. Thus, the tightness of the constraint for uv w.r.t. Ψ' in $D_{\mathcal{F}'}(G', c')$ follows from the tightness of the constraint w.r.t. Ψ in $D_{\mathcal{F}}(G, c)$. Suppose $x'(sv) > 0$ for an edge $sv \in E(G')$. Let $uv \in E(G)$ be the pre-image of sv for some $u \in S$. Then the tightness of the constraint follows since both the inequalities in (9) are tight – the first inequality is tight since uv is tight w.r.t. Ψ , and the second is tight since $\Pi_S(u) - \Psi_S(u) = \Delta(S)$ by the consistency property. Finally, if $\Psi'(T') > 0$ for some $T' \in \mathcal{F}'$, then $\Psi(T) > 0$ and hence $x(\delta(T)) = 1$ and therefore, $x'(\delta(T')) = 1$. \square

Lemma 4.5.6. *Let \mathcal{F} be a proper odd family with Π being an \mathcal{F} -fitting dual, and let x be an optimal solution to $P_{\mathcal{F}}(G, c)$. If $x(\delta(S)) = 1$ for some $S \in \mathcal{F}$, then $x(\delta(T)) = 1$ for every $T \subsetneq S$, $T \in \mathcal{F}$, and all edges in $\text{supp}(x) \cap E[S]$ are tight w.r.t. Π .*

Proof. Let $\alpha_u = x(\delta(u, V \setminus S))$ for each $u \in S$, and for each $T \subseteq S$, $T \in \mathcal{F}$, let $\alpha(T) = \sum_{u \in T} \alpha_u = x(\delta(T, V \setminus S))$. Note that $\alpha(S) = x(\delta(S)) = 1$. Let us consider the following pair of linear programs.

$$\begin{array}{ll} \min \sum_{uv \in E[T]} c(uv)z(uv) & (P_{\mathcal{F}}[S]) \\ z(\delta(u)) = 1 - \alpha_u \quad \forall u \in S & \\ z(\delta(T)) \geq 1 - \alpha(T) \quad \forall T \subsetneq S, T \in \mathcal{F} & \\ z(uv) \geq 0 \quad \forall uv \in E[S] & \end{array} \quad \begin{array}{ll} \max \sum_{T \subsetneq S, T \in \mathcal{V} \cup \mathcal{F}} (1 - \alpha(T))\Gamma(T) & (D_{\mathcal{F}}[S]) \\ \sum_{\substack{T \subsetneq S, T \in \mathcal{V} \cup \mathcal{F} \\ uv \in \delta(T)}} \Gamma(T) \leq c(uv) \quad \forall uv \in E[S] & \\ \Gamma(Z) \geq 0 \quad \forall Z \subsetneq T, Z \in \mathcal{F} & \end{array}$$

For a feasible solution z to $P_{\mathcal{F}}[S]$, let x^z denote the solution obtained by replacing $x(uv)$ by $z(uv)$ for edges uv inside S , that is,

$$x^z(e) = \begin{cases} x(e) & \text{if } e \in \delta(S) \cup E[V \setminus S], \\ z(e) & \text{if } e \in E[S]. \end{cases}$$

Claim 4.5.7. *The restriction of x inside S is feasible to $P_{\mathcal{F}}[S]$, and for every feasible solution z to $P_{\mathcal{F}}[S]$, x^z is a feasible solution to $P_{\mathcal{F}}(G, c)$. Consequently, z is an optimal solution to $P_{\mathcal{F}}[S]$ if and only if x^z is an optimal solution to $P_{\mathcal{F}}(G, c)$.*

Proof. The first part is obvious. For feasibility of x^z , if $u \notin S$ then $x^z(u) = x(u) = 1$. If $u \in S$, then $x^z(u) = z(u) + x(\delta(u, V \setminus S)) = 1 - \alpha_u + \alpha_u = 1$. Similarly, if $T \in \mathcal{F}$, $T \setminus S \neq \emptyset$, then $x^z(\delta(T)) = x(T) \geq 1$. If $T \subseteq S$, then $x^z(\delta(T)) = z(\delta(T)) + x(\delta(T, V \setminus S)) \geq 1 - \alpha(T) + \alpha(T) = 1$.

Optimality follows since $c^T x^z = \sum_{uv \in E[S]} c(uv)z(uv) + \sum_{uv \in E \setminus E[S]} c(uv)x(uv)$. \square

Claim 4.5.8. *Let $\bar{\Pi}$ denote the restriction of Π inside S , that is, $\bar{\Pi}(T) = \Pi(T)$ for every $T \in \mathcal{V} \cup \mathcal{F}$, $T \subseteq S$. Then $\bar{\Pi}$ is an optimal solution to $D_{\mathcal{F}}[S]$.*

Proof. Since Π is an \mathcal{F} -fitting dual and $S \in F$, we have a Π -critical-matching M_u inside S for each $u \in S$. Let $w = \sum_{u \in S} \alpha_u M_u$. The claim follows by showing that w is feasible to $P_{\mathcal{F}}[S]$ and that w and $\bar{\Pi}$ satisfy complementary slackness.

The degree constraint $w(\delta(u)) = 1 - \alpha_u$ is straightforward and using Lemma 4.4.2, if $T \subseteq S$, $T \in \mathcal{F}$, then $w(\delta(T)) = \sum_{u \in S \setminus T} \alpha_u = 1 - \alpha(T)$. The feasibility of Π to $D_{\mathcal{F}}(G, c)$ immediately shows feasibility of $\bar{\Pi}$ to $D_{\mathcal{F}}[S]$.

Complementary slackness also follows since by definition, all M_u 's use only tight edges w.r.t. Π (equivalently, w.r.t. $\bar{\Pi}$). Also, for every odd set $T \subseteq S$, $T \in \mathcal{F}$, we have that $w(\delta(T)) = 1 - \alpha(T)$ as verified above. Thus, all odd set constraints are tight in the primal. \square

By Claim 4.5.7, the solution obtained by restricting x to $E[S]$ must be optimal to $P_{\mathcal{F}}[S]$ and thus satisfies complementary slackness with $\bar{\Pi}$. Consequently, every edge in $E[S] \cap \text{supp}(x)$ must be tight w.r.t. $\bar{\Pi}$ and equivalently w.r.t. Π . By the definition of the \mathcal{F} -fitting-property, every non-maximal set T of \mathcal{F} satisfies $\Pi(T) > 0$, and in particular, every set $T \subseteq S$, $T \in \mathcal{F}$ satisfies $\bar{\Pi}(T) = \Pi(T) > 0$. Thus, complementary slackness gives $x(\delta(T)) = 1$.

\square

We need one more claim to show a strong structure property about dual solutions.

Claim 4.5.9. *Let $S \in \mathcal{F}$ be an inclusionwise minimal set of \mathcal{F} . Let Λ and Γ be feasible solutions to $D_{\mathcal{F}}(G, c)$, and suppose S is (Λ, \mathcal{F}) -factor-critical. Then $\Delta_{\Lambda, \Gamma}(S) = \max_{u \in S} |\Lambda_S(u) - \Gamma_S(u)|$. Further, if $\Delta_{\Lambda, \Gamma}(S) > 0$, define*

$$A^+ := \{u \in T : \Gamma(u) = \Lambda(u) + \Delta_{\Lambda, \Gamma}(S)\},$$

$$A^- := \{u \in T : \Gamma(u) = \Lambda(u) - \Delta_{\Lambda, \Gamma}(S)\}.$$

Then $|A^-| > |A^+|$.

Proof. Let $\Delta = \max_{u \in S} |\Lambda_S(u) - \Gamma_S(u)|$, and define the sets A^- and A^+ with Δ instead of $\Delta_{\Lambda, \Gamma}(S)$. Since S is (Λ, \mathcal{F}) -factor-critical, for every $a \in S$, there exists an $(S \setminus \{a\}, \mathcal{F})$ perfect matching M_a using only tight edges w.r.t. Λ , i.e., $M_a \subseteq \{uv : \Lambda(u) + \Lambda(v) = c(uv)\}$ by the minimality of S . Further, by feasibility of Γ , we have $\Gamma(u) + \Gamma(v) \leq c(uv)$ on every $uv \in M_a$. Thus, if $u \in A^+$, then $v \in A^-$ for every $uv \in M_a$. Since $\Delta > 0$, we have $A^+ \cup A^- \neq \emptyset$ and therefore $A^- \neq \emptyset$, and consequently, $\Delta = \Delta_{\Lambda, \Gamma}(S)$. Now pick $a \in A^-$ and consider M_a . This perfect matching M_a matches each node in A^+ to a node in A^- . Thus, $|A^-| > |A^+|$. \square

We now prove the main lemma of this section.

Lemma 4.5.10. *Let $\mathcal{F} \subseteq \mathcal{O}$ be a proper odd family, with Π being an \mathcal{F} -fitting dual solution and let Ψ be an optimal solution to $D_{\mathcal{F}}(G, c)$. Let x be an optimal solution to $P_{\mathcal{F}}(G, c)$. Then Ψ is consistent with Π inside every set $S \in \mathcal{F}$ such that $x(\delta(S)) = 1$. Further, $\Delta_{\Pi, \Psi}(S) \geq 0$ for all such sets.*

Proof. We prove this by induction on $|V|$, and subject to this, on $|S|$. Let S be a (Π, \mathcal{F}) -factor-critical set. First, consider the case when S is an inclusion-wise minimal set. Then, $\Pi_S(u) = \Pi(u)$, $\Psi_S(u) = \Psi(u)$ for every $u \in S$. By Claim 4.5.9, we have that $\Delta := \Delta_{\Pi, \Psi}(S) \geq 0$. We are done if $\Delta = 0$. Otherwise, define the sets A^- and A^+ as in the claim using $\Delta_{\Pi, \Psi}(S)$.

Now consider an edge $uv \in E[S] \cap \text{supp}(x)$. By complementary slackness, we have $\Psi(u) + \Psi(v) = c(uv)$. By dual feasibility, we have $\Pi(u) + \Pi(v) \leq c(uv)$. Hence, if $u \in A^-$,

then $v \in A^+$. Consequently, we have

$$\begin{aligned} |A^-| &= \sum_{u \in A^-} x(\delta(u)) = x(\delta(A^-, V \setminus S)) + x(\delta(A^-, A^+)) \\ &\leq 1 + \sum_{u \in A^+} x(\delta(u)) = 1 + |A^+| \leq |A^-|. \end{aligned}$$

Thus, we should have equality throughout. Hence, $x(\delta(A^-, V \setminus S)) = 1$. This precisely means that Ψ is consistent with Π inside S .

Next, let S be a non-minimal set. Let $T \in \mathcal{F}$ be a maximal set strictly contained in S . By Lemma 4.5.6, $x(\delta(T)) = 1$, therefore the inductual claim holds for T : Ψ is consistent with Π inside T , and $\Delta(T) = \Delta_{\Pi, \Psi}(T) \geq 0$.

We contract T w.r.t. Π and use Lemma 4.5.5. Let the image of the solutions x , Π , and Ψ be x' , Π' and Ψ' respectively and the resulting graph be G' with cost function c' . Then x' and Ψ' are optimum to $P_{\mathcal{F}'}(G', c')$ and $D_{\mathcal{F}'}(G', c')$ respectively, and by Lemma 4.5.1(i), Π' is an \mathcal{F}' -fitting dual. Let t be the image of T by the contraction. Now, consider the image S' of S in G' . Since G' is a smaller graph, it satisfies the induction hypothesis. Let $\Delta' = \Delta_{\Pi', \Psi'}(S')$ in G' . By induction hypothesis, $\Delta' \geq 0$. The following claim verifies consistency inside S and thus completes the proof. \square

Claim 4.5.11. *For every $u \in S$, $\Pi_S(u) - \Psi_S(u) \leq \Pi'_{S'}(u') - \Psi'_{S'}(u')$, and equality holds if there exists an edge $uv \in \delta(S) \cap \text{supp}(x)$. Consequently, $\Delta' = \Delta$.*

Proof. Let u' denote the image of u . If $u' \neq t$, then $\Pi'_{S'}(u') = \Pi_S(u)$, $\Psi'_{S'}(u') = \Psi_S(u)$ and therefore, $\Pi_S(u) - \Psi_S(u) = \Pi'_{S'}(u') - \Psi'_{S'}(u')$. Assume $u' = t$, that is, $u \in T$. Then $\Pi_S(u) = \Pi_T(u) + \Pi(T)$, $\Psi_S(u) = \Psi_T(u) + \Psi(T)$ by the maximal choice of T , and therefore

$$\begin{aligned} \Pi_S(u) - \Psi_S(u) &= \Pi_T(u) - \Psi_T(u) + \Pi(T) - \Psi(T) \\ &\leq \Delta(T) + \Pi(T) - \Psi(T) \\ &= \Pi'(t) - \Psi'(t) \quad (\text{Since } \Pi'(t) = \Pi(T), \Psi'(t) = \Psi(T) - \Delta(T)) \\ &= \Pi'_{S'}(t) - \Psi'_{S'}(t). \end{aligned} \tag{10}$$

Assume now that there exists a $uv \in \delta(S) \cap \text{supp}(x)$. If $u \in T$, then using the consistency inside T , we get $\Pi_T(u) - \Psi_T(u) = \Delta(T)$, and therefore (10) gives $\Pi_S(u) - \Psi_S(u) = \Pi'_{S'}(t) -$

$$\Psi'_{S'}(t) = \Delta'. \quad \square$$

Claim 4.5.9 can also be used to derive the following important property.

Lemma 4.5.12. *Given a laminar odd family $\mathcal{F} \subset \mathcal{O}$, let Λ and Γ be two dual feasible solutions to $D_{\mathcal{F}}(G, c)$. If a subset $S \in \mathcal{F}$ is both (Λ, \mathcal{F}) -factor-critical and (Γ, \mathcal{F}) -factor-critical, then Λ and Γ are identical inside S .*

Proof. Consider a graph $G = (V, E)$ with $|V|$ minimal, where the claim does not hold for some set S . Also, choose S to be the smallest counterexample in this graph. First, assume $S \in \mathcal{F}$ is a minimal set. Then consider Claim 4.5.9 for Λ and Γ and also by changing their roles, for Γ and Λ . If Λ and Γ are not identical inside S , then $\Delta = \max_{u \in S} |\Lambda_S(u) - \Gamma_S(u)| > 0$. The sets A^- and A^+ for Λ and Γ become A^+ and A^- for Γ and Λ . Then $|A^-| > |A^+| > |A^-|$, a contradiction.

Suppose now S contains $T \in \mathcal{F}$. It is straightforward to see that T is also (Λ, \mathcal{F}) -factor-critical and (Γ, \mathcal{F}) -factor-critical by definition. Thus, by the minimal choice of the counterexample S , we have that Λ and Γ are identical inside T . Now, contract the set T w.r.t. Λ , or equivalently, w.r.t. Γ . Let Λ', Γ' denote the contracted solutions in G' , and let \mathcal{F}' be the contraction of \mathcal{F} . Then, by Lemma 4.5.1(i), these two solutions are feasible to $D_{\mathcal{F}'}(G', c')$, and S' is both Λ' -factor-critical and Γ' -factor-critical. Now, Λ' and Γ' are not identical inside S' , contradicting the minimal choice of G and S . \square

4.5.3.2 Finding a Positively-fitting Dual Optimal Solution

Let $\mathcal{F} \subseteq \mathcal{O}$ be a proper odd family with Π being an \mathcal{F} -fitting dual. Let Ψ be a dual optimum solution to $D_{\mathcal{F}}(G, c)$. Our goal is to satisfy the property that for every $S \in \mathcal{F}$, if $\Psi(S) > 0$, then Ψ and Π are identical inside S . By Lemma 4.5.12, it is equivalent to showing that Ψ is \mathcal{F} -positively-fitting. We modify Ψ by the algorithm shown in Figure 14.

The correctness of the algorithm follows by showing that the modified solution $\bar{\Psi}$ is also dual optimal, and it is closer to Π .

Lemma 4.5.13. *Let $\mathcal{F} \subseteq \mathcal{O}$ be a proper odd family with Π being an \mathcal{F} -fitting dual and let Ψ be a dual optimum solution to $D_{\mathcal{F}}(G, c)$. Suppose we consider a maximal set S such that*

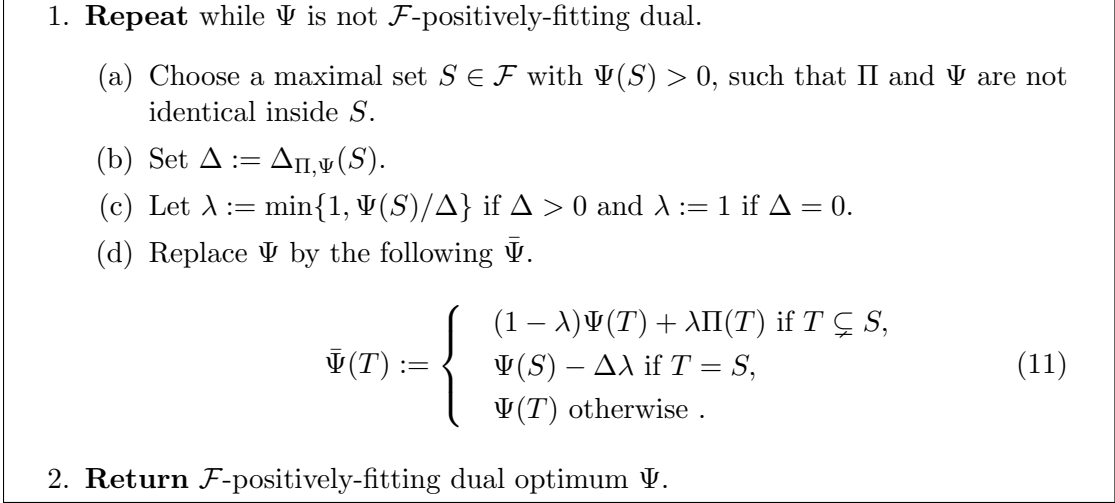


Figure 14: Algorithm Positively-fitting-dual-opt

Π and Ψ are not identical inside S , take $\lambda = \min\{1, \Psi(S)/\Delta_{\Pi, \Psi}(S)\}$ if $\Delta_{\Pi, \Psi}(S) > 0$ and $\lambda = 1$ if $\Delta_{\Pi, \Psi}(S) = 0$ and set $\bar{\Psi}$ as in (11). Then, $\bar{\Psi}$ is also a dual optimal solution to $D_{\mathcal{F}}(G, c)$, and either $\bar{\Psi}(S) = 0$ or Π and Ψ are identical inside S .

Proof. Let x be an optimal solution to $P_{\mathcal{F}}(G, c)$. Since $\Psi(S) > 0$, we have $x(\delta(S)) = 1$ and by Lemma 4.5.10, we have $\Delta = \Delta_{\Pi, \Psi}(S) \geq 0$. Now, the second conclusion is immediate from definition: if $\lambda = 1$, then we have that Π and $\bar{\Psi}$ are identical inside S ; if $\lambda < 1$, then we have $\bar{\Psi}(S) = 0$. For optimality, we show feasibility and verify the primal-dual slackness conditions.

The solution $\bar{\Psi}$ might have positive components on some sets $T \subsetneq S, T \in \mathcal{F}$ where $\Pi(T) > 0$. However, $x(\delta(T)) = 1$ for all such sets by Lemma 4.5.6. The choice of λ also guarantees $\bar{\Psi}(S) \geq 0$. We need to verify that all inequalities in $D_{\mathcal{F}}(G, c)$ are maintained and that all tight constraints in $D_{\mathcal{F}}(G, c)$ w.r.t. Ψ are maintained. This trivially holds if $uv \in E[V \setminus S]$. If $uv \in E[S] \setminus \text{supp}(x)$, the corresponding inequality is satisfied by both Π and Ψ and hence also by their linear combinations. If $uv \in E[S] \cap \text{supp}(x)$, then uv is tight for Ψ by the optimality of Ψ , and also for Π by Lemma 4.5.6.

It remains to verify the constraint corresponding to edges uv with $u \in S, v \in V \setminus S$. The contribution of $\sum_{T \in \mathcal{F}: uv \in \delta(T), T \setminus S \neq \emptyset} \Psi(T)$ is unchanged. The following claim completes the proof of optimality. □

Claim 4.5.14. $\bar{\Psi}_S(u) + \bar{\Psi}(S) \leq \Psi_S(u) + \Psi(S)$ with equality whenever $uv \in \text{supp}(x)$.

Proof.

$$\bar{\Psi}(T) - \Psi(T) = \begin{cases} \lambda(\Pi(T) - \Psi(T)) & \text{if } T \subsetneq S, \\ -\Delta\lambda & \text{if } T = S. \end{cases}$$

Thus,

$$\begin{aligned} \bar{\Psi}_S(u) + \bar{\Psi}(S) &= \lambda(\Pi_S(u) - \Psi_S(u)) + \bar{\Psi}(S) - \Psi(S) + \Psi_S(u) + \Psi(S) \\ &= \lambda(\Pi_S(u) - \Psi_S(u) - \Delta) + \Psi_S(u) + \Psi(S). \end{aligned}$$

Now, $\Pi_S(u) - \Psi_S(u) \leq \Delta$, and equality holds whenever $uv \in \text{supp}(x) \cap \delta(S)$ by the consistency of Ψ and Π inside S (Lemma 4.5.10). \square

Corollary 4.5.15. *Let \mathcal{F} be a proper-odd-family with Π being an \mathcal{F} -fitting dual feasible solution. Algorithm *Positively-fitting-dual-opt* in Figure 14 transforms an arbitrary dual optimal solution Ψ to an \mathcal{F} -positively-fitting dual optimal solution in at most $|\mathcal{F}|$ iterations.*

Proof. The correctness of the algorithm follows by Lemma 4.5.13. We bound the running time by showing that no set $S \in \mathcal{F}$ is processed twice. After a set S is processed, by Lemma 4.5.13, either Π and Ψ will be identical inside S or $\Psi(S) = 0$. Once Π and Ψ become identical inside a set, it remains so during all later iterations.

The value $\Psi(S)$ could be changed later only if we process a set $S' \supsetneq S$ after processing S . Let S' be the first such set. At the iteration when S was processed, by the maximal choice it follows that $\Psi(S') = 0$. Hence $\Psi(S')$ could become positive only if the algorithm had processed a set $Z \supsetneq S'$, $Z \in \mathcal{F}$ between processing S and S' , a contradiction to the choice of S' . \square

4.5.3.3 Extremal Dual Solutions

Assume $\mathcal{F} \subseteq \mathcal{O}$ is a proper odd family, with Π being an \mathcal{F} -fitting dual. Let x be the unique optimal solution to $P_{\mathcal{F}}(G, c)$. Let $\mathcal{F}_x = \{S \in \mathcal{F} : x(\delta(S)) = 1\}$ the collection of tight sets

for x . A Π -extremal dual can be found by solving the following LP.

$$\begin{aligned}
\min h(\Psi, \Pi) &= \sum_{S \in \mathcal{V} \cup \mathcal{F}_x} \frac{r(S)}{|S|} & (D_{\mathcal{F}}^*) \\
-r(S) &\leq \Psi(S) - \Pi(S) \leq r(S) \quad \forall S \in \mathcal{V} \cup \mathcal{F}_x \\
\sum_{S \in \mathcal{V} \cup \mathcal{F}_x: uv \in \delta(S)} \Psi(S) &= c(uv) \quad \forall uv \in \text{supp}(x) \\
\sum_{S \in \mathcal{V} \cup \mathcal{F}_x: uv \in \delta(S)} \Psi(S) &\leq c(uv) \quad \forall uv \in E \setminus \text{supp}(x) \\
\Psi(S) &\geq 0 \quad \forall S \in \mathcal{F}_x
\end{aligned}$$

The support of Ψ is restricted to sets in $\mathcal{V} \cup \mathcal{F}_x$. Primal-dual slackness implies that the feasible solutions to this program coincide with the optimal solutions of $D_{\mathcal{F}}(G, c)$, hence an optimal solution to $D_{\mathcal{F}}^*$ is also an optimal solution to $D_{\mathcal{F}}(G, c)$.

Lemma 4.5.16. *Let $\mathcal{F} \subset \mathcal{O}$ be a proper odd family with Π being an \mathcal{F} -fitting dual. Then, a Π -extremal dual is also an \mathcal{F} -positively-fitting dual optimal solution.*

Proof. We will show that whenever $\Psi(S) > 0$, the solutions Ψ and Π are identical inside S . Assume for a contradiction that this is not true for some $S \in \mathcal{F}$. Let $\lambda = \min\{1, \Psi(S)/\Delta_{\Pi, \Psi}(S)\}$ if $\Delta_{\Pi, \Psi}(S) > 0$ and $\lambda = 1$ if $\Delta_{\Pi, \Psi}(S) = 0$. Define $\bar{\Psi}$ as in (11). By Lemma 4.5.13, $\bar{\Psi}$ is also optimal to $D_{\mathcal{F}}(G, c)$ and thus feasible to $D_{\mathcal{F}}^*$. We show $h(\bar{\Psi}, \Pi) < h(\Psi, \Pi)$, which is a contradiction.

For every $T \in \mathcal{V} \cup \mathcal{F}_x$, let $\tau(T) = |\Psi(T) - \Pi(T)| - |\bar{\Psi}(T) - \Pi(T)|$. With this notation,

$$h(\Psi, \Pi) - h(\bar{\Psi}, \Pi) = \sum_{T \in \mathcal{V} \cup \mathcal{F}_x} \frac{\tau(T)}{|T|}.$$

If $T \setminus S = \emptyset$, then $\bar{\Psi}(T) = \Psi(T)$ and thus $\tau(T) = 0$. If $T \subsetneq S$, $T \in \mathcal{V} \cup \mathcal{F}$, then $|\bar{\Psi}(T) - \Pi(T)| = (1 - \lambda)|\Psi(T) - \Pi(T)|$, and thus $\tau(T) = \lambda|\Psi(T) - \Pi(T)|$. Since $\bar{\Psi}(S) = \Psi(S) - \Delta\lambda$, we have $\tau(S) \geq -\Delta\lambda$.

Let us fix an arbitrary $u \in S$, and let $\gamma = \max_{T \subseteq S: u \in T, T \in \mathcal{V} \cup \mathcal{F}_x} |T|$.

$$\begin{aligned}
h(\Psi, \Pi) - h(\bar{\Psi}, \Pi) &= \sum_{T \in \mathcal{V} \cup \mathcal{F}_x} \frac{\tau(T)}{|T|} \\
&\geq \sum_{T \subseteq S: u \in T, T \in \mathcal{V} \cup \mathcal{F}_x} \frac{\tau(T)}{|T|} + \frac{\tau(S)}{|S|} \\
&\geq \frac{\lambda}{\gamma} \sum_{T \subseteq S: u \in T, T \in \mathcal{V} \cup \mathcal{F}_x} |\Psi(T) - \Pi(T)| - \frac{\Delta \lambda}{|S|} \\
&\geq \frac{\lambda}{\gamma} (\Pi_S(u) - \Psi_S(u)) - \frac{\Delta \lambda}{|S|}.
\end{aligned}$$

Case 1: If $\Delta > 0$, then pick $u \in S$ satisfying $\Pi_S(u) - \Psi_S(u) = \Delta$. Then the above inequalities give

$$h(\Psi, \Pi) - h(\bar{\Psi}, \Pi) \geq \Delta \lambda \left(\frac{1}{\gamma} - \frac{1}{|S|} \right) > 0.$$

The last inequality follows since $|S| > \gamma$.

Case 2: If $\Delta = 0$, then $\lambda = 1$ and therefore,

$$h(\Psi, \Pi) - h(\bar{\Psi}, \Pi) \geq \frac{1}{\gamma} \sum_{T \subseteq S: u \in T, T \in \mathcal{V} \cup \mathcal{F}_x} |\Psi(T) - \Pi(T)|$$

Now, if Π and Ψ are not identical inside S , then there exists a node $u \in S$ for which the RHS is strictly positive. Thus, in both cases, we get $h(\bar{\Psi}, \Pi) < h(\Psi, \Pi)$, a contradiction to the optimality of Ψ to $D_{\mathcal{F}}^*$. \square

We are now ready to prove the main lemma of this section showing that intermediate laminar set of blossoms lead to a proper-odd-family.

Lemma 4.5.17. *Suppose that in an iteration of Algorithm C-P-Matching, \mathcal{F} is a proper odd family with Π being an \mathcal{F} -fitting dual solution. Then the Π -extremal dual solution Ψ is an \mathcal{F} -positively-fitting dual optimal solution. Furthermore, the next set of cuts $\mathcal{H} = \mathcal{H}' \cup \mathcal{H}''$ is a proper odd family with Ψ being an \mathcal{H} -fitting dual.*

Proof. By Lemma 4.5.4, the unique optimal x to $P_{\mathcal{F}}(G, c)$ is proper-half-integral. We already know that a Π -extremal dual solution Ψ is also \mathcal{F} -positively-fitting by Lemma 4.5.16. We need to show that the next family of cuts is a proper-odd-family. Recall that the set of cuts for the next round is defined as $\mathcal{H}' \cup \mathcal{H}''$, where $\mathcal{H}' = \{T \in \mathcal{F} : \Psi(T) > 0\}$, and

\mathcal{H}'' is defined based on some cycles in $\text{supp}(x)$. We need to show that every member of $\mathcal{H}' \cup \mathcal{H}''$ is Ψ -factor-critical. This is straightforward for members of \mathcal{H}' by the definition of the \mathcal{F} -positively-fitting property.

It remains to show that the members of \mathcal{H}'' are also Ψ -factor-critical. These are defined for odd cycles $C \in \text{supp}(x)$. Now, $\hat{C} \in \mathcal{H}''$ is the union of $V(C)$ and the maximal members S_1, \dots, S_ℓ of \mathcal{F}' intersecting $V(C)$. We have $\Psi(S_j) > 0$ for each $j = 1, \dots, \ell$ and hence $x(\delta(S_j)) = 1$.

Let $u \in \hat{C}$ be an arbitrary node; we will construct the Ψ -critical matching \hat{M}_u . Let us contract all sets S_1, \dots, S_ℓ to nodes s_1, \dots, s_ℓ wrt Ψ . We know by Lemma 4.5.1(iii) that the image x' of x is proper-half-integral and that the cycle C projects to a cycle in x' . The proper-half-integral property guarantees that C is contracted into an odd cycle C' in $\text{supp}(x')$. Let u' be the image of u ; the pre-image of every edge in C' is a tight edge w.r.t. Ψ in the original graph since Ψ is an optimum to the dual problem $D_{\mathcal{F}}(G, c)$. Since C' is an odd cycle, there is a perfect matching $M'_{u'}$ that covers every node in $V(C') \setminus \{u'\}$ using the edges in C' .

Assume first $u \in S_j$ for some $1 \leq j \leq \ell$. Then $u' = s_j$. The pre-image \hat{M} of $M'_{u'}$ in the original graph contains exactly one edge entering each S_k for $k \neq j$ and no edges entering S_j . Consider the Ψ -critical matching M_u for u in S_j . For $k \neq j$, if $a_k b_k \in \hat{M} \cap \delta(S_k)$, $a_k \in S_k$, then, let M_{a_k} be the Ψ -critical matching for a_k in S_k . The union of \hat{M} , M_u and the M_{a_k} 's give a Ψ -critical matching for u inside \hat{C} .

If $u \in \hat{C} \setminus (\cup_{j=1}^{\ell} S_j)$, then similarly there is a Ψ -critical matching M_{a_k} inside every S_k . The union of \hat{M} and the M_{a_k} 's give the Ψ -critical matching for u inside \hat{C} . We also have $\Psi(S) > 0$ for all non-maximal sets $S \in \mathcal{H}' \cup \mathcal{H}''$ since the only members with $\Psi(S) = 0$ are those in \mathcal{H}'' , and they are all maximal ones. \square

4.5.4 Uniqueness

In Section 4.5.3.2, we proved that if \mathcal{F} is a proper odd family, then there always exists an \mathcal{F} -positively-fitting optimal solution (Corollary 4.5.15). This argument did not use uniqueness. Indeed, it will also be used to derive that a perturbation of the original integer cost function

satisfies our uniqueness assumption (*). We will need the following simple claim.

Claim 4.5.18. *For a graph $G = (V, E)$, let $a, b : E \rightarrow \mathbb{R}_+$ be two vectors on the edges with $a(\delta(v)) = b(\delta(v))$ for every $v \in V$. If a and b are not identical, then there exists an even length closed walk C such that for every odd edge $e \in C$, $a(e) > 0$ and for every even edge $e \in C$, $b(e) > 0$.*

Proof. Due to the degree constraints, $z = a - b$ satisfies $z(\delta(v)) = 0$ for every $v \in V$, and since a and b are not identical, z has nonzero components. If there is an edge $uv \in E$ with $z(uv) > 0$ then there must be another edge $uw \in E$ with $z(uw) < 0$. This implies the existence of an alternating even closed walk C where for every odd edge $e \in C$, $0 < z(e) = a(e) - b(e)$, and for every even edge, $0 > z(e) = a(e) - b(e)$. This proves the claim. \square

We are now ready to prove the statement about intermediate solutions being unique optimal solutions upon perturbation.

Lemma 4.5.19. *Let $c : E \rightarrow \mathbb{Z}$ be an integer cost function, and \tilde{c} as follows. Choose an arbitrary ordering of the edges, and increase the cost of the i 'th edge by $1/2^i$. Then \tilde{c} satisfies (*).*

Proof. Let \tilde{c} denote the perturbation of the integer cost $c : E \rightarrow \mathbb{Z}$. Consider a graph $G = (V, E)$, perturbed cost \tilde{c} and proper odd family \mathcal{F} where (*) does not hold. Choose a counterexample with $|\mathcal{F}|$ minimal. Let x and y be two different optimal solutions to $P_{\mathcal{F}}(G, \tilde{c})$. Since \mathcal{F} is a proper-odd-family, by Corollary 4.5.15, there exists an \mathcal{F} -positively-fitting dual optimal solution, say Π .

First, assume $\mathcal{F} = \emptyset$. Then x and y are both optimal solutions to the bipartite relaxation $P_0(G, \tilde{c})$. As they are not identical, Claim 4.5.18 gives an even closed walk C such that $x(e) > 0$ on every even edge and $y(e) > 0$ on every odd edge. Let γ_1 and γ_2 be the sum of edge costs on even and on odd edges of C , respectively. Then for some $\varepsilon > 0$, we could modify x by decreasing $x(e)$ by ε on even edges and increasing on odd edges, and y the other way around. These give two other optimal matchings \bar{x} and \bar{y} , with $\tilde{c}^T \bar{x} = \tilde{c}^T x + (\gamma_2 - \gamma_1)\varepsilon$ and $\tilde{c}^T \bar{y} = \tilde{c}^T y + (\gamma_1 - \gamma_2)\varepsilon$. Since \bar{x} and \bar{y} are both optimal, this gives $\gamma_1 = \gamma_2$. However, the

fractional parts of γ_1 and γ_2 must be different according to the definition of the perturbation, giving a contradiction.

If $\mathcal{F} \neq \emptyset$, after modifying x, y to non-identical optimal solutions a, b respectively (if necessary), we will similarly identify an alternating closed walk C in $\text{supp}(a) \cup \text{supp}(b)$ with the additional property that if C intersects $\delta(S)$ for some member $S \in \mathcal{F}$, then it does so in exactly one even and one odd edge. The modifications \bar{a} and \bar{b} defined as above would again be feasible, implying $\gamma_1 = \gamma_2$.

We first claim that $\Pi(S) > 0$ must hold for all $S \in \mathcal{F}$. Indeed, if $\Pi(S) = 0$ for some $S \in \mathcal{F}$, then x and y would be two different optimal solutions to $P_{\mathcal{F} \setminus \{S\}}(G, \tilde{c})$, contradicting the minimal choice of \mathcal{F} . Let $T \in \mathcal{F} \cup \{V\}$ be a smallest set with the property that $x(\delta(u, V - T)) = y(\delta(u, V - T))$ for every $u \in T$, but x and y are not identical inside T . Note that V trivially satisfies this property and hence such a set exists. Let \mathcal{S} denote the collection of maximal sets S in \mathcal{F} such that $S \subsetneq T$ (\mathcal{S} could possibly be empty).

Inside each maximal set $S \in \mathcal{S}$, we modify x and y such that they are still both optimal and different from each other after the modification. Since $\Pi(S) > 0$, we have $x(\delta(S)) = 1$, $y(\delta(S)) = 1$ and S is Π -factor-critical. For $u \in S$, let M_u denote the Π -critical matching for u inside S . Let $\alpha_u := x(\delta(u, V - S))$, $\alpha'_u = y(\delta(u, V - S))$ and $w := \sum_{u \in S} \alpha_u M_u$, $w' := \sum_{u \in S} \alpha'_u M_u$. If x and y are not identical inside S and $\alpha_u = \alpha'_u$ for every $u \in S$, then S contradicts the minimality of T . Hence, for each maximal set $S \in \mathcal{S}$ we have two cases – either (1) x and y are identical inside S , and thus for every $u \in S$, $\alpha_u = \alpha'_u$ or (2) $\alpha_u \neq \alpha'_u$ for some $u \in S$.

Fix $S \in \mathcal{S}$. The following modified solutions

$$a(e) := \begin{cases} x(e) & \text{if } e \in \delta(S) \cup E[V \setminus S], \\ w(e) & \text{if } e \in E[S], \end{cases}$$

$$b(e) := \begin{cases} y(e) & \text{if } e \in \delta(S) \cup E[V \setminus S], \\ w'(e) & \text{if } e \in E[S], \end{cases}$$

are also optimal solutions since we used only tight edges w.r.t. Π for these modifications.

Claim 4.5.20. *Consider two edges $u_1v_1 \in \delta(S) \cap \text{supp}(a)$, $u_2v_2 \in \delta(S) \cap \text{supp}(b)$, $u_1, u_2 \in S$. Then there exists an even alternating path P_S inside S between u_1 and u_2 such that $a(e) > 0$ for every even edge and $b(e) > 0$ for every odd edge. Also, consider the extended path $P'_S = u_1v_1P_Su_2v_2$. If there exists a set $Z \subsetneq S, Z \in \mathcal{F}$ such that $V(P'_S) \cap Z \neq \emptyset$, then P'_S intersects $\delta(Z)$ in exactly one even and one odd edge.*

Proof. By the modification, $\text{supp}(a) \cap E[S]$ and $\text{supp}(b) \cap E[S]$ contain the Π -critical-matchings M_{u_1} and M_{u_2} respectively.

By the modification, $\text{supp}(a) \cap E[S]$ contains the Π -critical-matching M_{u_1} and $\text{supp}(b) \cap E[S]$ contains the Π -critical-matching M_{u_2} . Then the symmetric difference of M_{u_1} and M_{u_2} contains an $u_1 - u_2$ alternating path satisfying the requirements (by Lemma 4.4.2). \square

We perform the above modifications inside every $S \in \mathcal{S}$, and denote by a and b the result of all these modifications. Let us now contract all sets in \mathcal{S} w.r.t. Π ; let $G', \mathcal{C}', \mathcal{F}'$ denote the resulting graph, costs and laminar family respectively. By Lemma 4.5.1(ii), the images a' and b' are both optimal solutions in $P'_{\mathcal{F}'}(G', \mathcal{C}')$.

We claim that a' and b' are still not identical inside T' . Indeed, assume a' and b' are identical inside T' . Then we must have had case (1) for every $S \in \mathcal{S}$. Note that we only modified x and y on the edge set $\cup_{S \in \mathcal{S}} E[S]$, and hence the contracted images must be identical: $x' = a'$ and $y' = b'$, and consequently, $x' = y'$. As x and y are not identical inside T , they must differ inside at least one $S \in \mathcal{S}$. This contradicts the fact that case (1) applied for every $S \in \mathcal{S}$.

By the assumption on T , we also have $a'(\delta(u', V' - T')) = b'(\delta(u', V' - T'))$ for every $u' \in T'$. Hence Claim 4.5.18 is applicable, giving an alternating closed walk C' with $a'(e) > 0$ on every even edge and $b'(e) > 0$ on every odd edge. Now, we can extend C' to an even alternating closed walk C in the original graph using the paths P_S as in the above claim. The resulting closed walk C will have the property that if there exists a set $Z \subsetneq S, Z \in \mathcal{F}$ such that $V(C) \cap Z \neq \emptyset$, then C intersects $\delta(Z)$ in exactly one even and one odd edge. \square

4.5.5 Half-Integral Primal-Dual Algorithm

4.5.5.1 Outline

Let us now turn to the description of the Half-integral Primal-Dual algorithm. We assume familiarity with Edmonds' algorithm [27] in the following sketch (see following section for complete description). In every step of the algorithm, we maintain z to be a proper-half-integral partial matching with exposed set $T \subseteq W$, Λ to be a dual solution satisfying complementary slackness with z , and the support of Λ is $\mathcal{V} \cup \mathcal{L}$ for some $\mathcal{L} \subseteq \mathcal{K}$. We start with $z = x$, $T = W$, $\Lambda = \Pi$ and $\mathcal{L} = \mathcal{K}$. We work on the graph G^* resulting from the contraction of the maximal members of \mathcal{L} .

By changing the dual Λ , we grow an alternating forest of tight edges in G^* , using only edges e with $z(e) = 0$ or $z(e) = 1$. The forest is rooted at the exposed set of nodes T . The solution z will be changed according to the following three scenarios. (a) If we find an alternating path between two exposed nodes, we change z by alternating along this path as in Edmonds' algorithm. (b) If we find an alternating path P from an exposed node to a $1/2$ -cycle C in $\text{supp}(z)$, we change z by alternating along P , and replacing it by a blossom (an alternating odd cycle) on C . (c) If we find an alternating path P from an exposed node to a blossom C , then we change z by alternating along P and replacing the blossom by a $1/2$ -cycle on C . If none of these cases apply, we change the dual Λ in order to extend the alternating forest. If $\Lambda(S)$ decreases to 0 for some $S \in \Lambda$, then we remove S from Λ and unshrink it in G^* .

The modifications are illustrated in Figure 4.5.5.1. Note that in case (c), Edmonds' algorithm would instead contract C . In contrast, we do not perform any contractions, but allow $1/2$ -cycles in the Half-integral Primal-Dual algorithm. For the special starting solution $x \equiv 0$, $\Pi \equiv 0$, our Half-integral Primal-Dual algorithm will return a proper-half-integral optimum to the bipartite relaxation.

4.5.5.2 The Complete Half-integral Primal-Dual Algorithm

The Half-integral Primal-Dual algorithm will be applied in certain contractions of G , but here we present it for a general graph $G = (V, E)$ and cost c . We use the terminology of

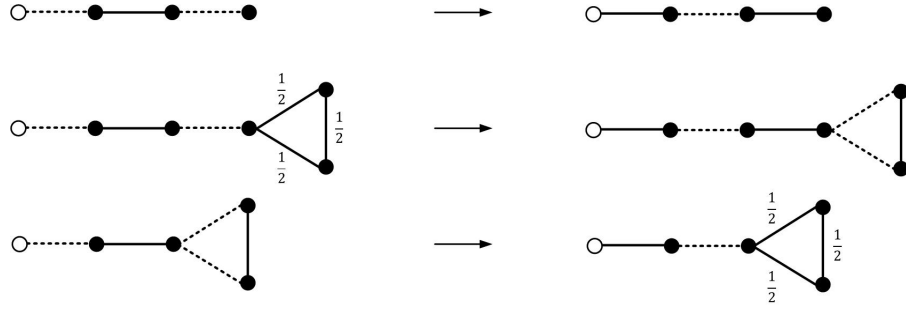


Figure 15: Possible primal updates in the Half-integral Primal-Dual algorithm

Edmonds' weighted matching algorithm [27] as described by Schrijver [64, Vol A, Chapter 26].

Let $W \subseteq V$, and let $\mathcal{F} \subset \mathcal{O}$ be a laminar family of odd sets that are disjoint from W . Let \mathcal{V}^W denote the set of one-element subsets of $V \setminus W$. The following primal $P_{\mathcal{F}}^W(G, c)$ and dual $D_{\mathcal{F}}^W(G, c)$ programs describe fractional matchings that leave the set of nodes in W exposed (unmatched) while satisfying the blossom inequalities corresponding to a laminar family \mathcal{F} . The primal program is identical to $P_{\mathcal{F}}(G \setminus W, c)$ while optimal solutions to $D_{\mathcal{F}}(G \setminus W, c)$ that are feasible to $D_{\mathcal{F}}^W(G, c)$ are also optimal solutions to $D_{\mathcal{F}}^W(G, c)$.

$$\begin{array}{ll}
 \min \sum_{uv \in E} c(uv)x(uv) & (P_{\mathcal{F}}^W(G, c)) \\
 x(\delta(u)) = 1 \quad \forall u \in V - W \\
 x(\delta(u)) = 0 \quad \forall u \in W \\
 x(\delta(S)) \geq 1 \quad \forall S \in \mathcal{F} \\
 x \geq 0
 \end{array}
 \qquad
 \begin{array}{ll}
 \max \sum_{S \in \mathcal{V}^W \cup \mathcal{F}} \Pi(S) & (D_{\mathcal{F}}^W(G, c)) \\
 \sum_{S \in \mathcal{V}^W \cup \mathcal{F}: uv \in \delta(S)} \Pi(S) \leq c(uv) \quad \forall uv \in E \\
 \Pi(S) \geq 0 \quad \forall S \in \mathcal{F}
 \end{array}$$

The algorithm is iterative. In each iteration, it maintains a set $T \subseteq W$, a subset $\mathcal{L} \subseteq \mathcal{F}$ of cuts, a proper-half-integral optimal solution z to $P_{\mathcal{L}}^T(G, c)$, and an \mathcal{L} -fitting dual optimal solution Λ to $D_{\mathcal{L}}^T(G, c)$ such that $\Lambda(S) > 0$ for every $S \in \mathcal{L}$. In the beginning $T = W$, $\mathcal{L} = \mathcal{F}$ and the algorithm terminates when $T = \emptyset$.

We work on the graph $G^* = (\mathcal{V}^*, E^*)$, obtained the following way from G : We first remove every edge in E that is not tight w.r.t. Λ , and then contract all maximal members

of \mathcal{L} w.r.t. Λ . The node set of \mathcal{V}^* is identified with the pre-images. Let c^* denote the contracted cost function and z^* the image of z . Since E^* consists only of tight edges, $\Lambda(u) + \Lambda(v) = c^*(uv)$ for every edge $uv \in E^*$. Since \mathcal{F} is disjoint from W , the nodes in \mathcal{L} will always have degree 1 in z^* .

In the course of the algorithm, we may decrease $\Lambda(S)$ to 0 for a maximal member S of \mathcal{L} . In this case, we remove S from \mathcal{L} and modify G^* , c^* and z^* accordingly. This operation will be referred as ‘unshrinking’ S . New sets will never be added to \mathcal{L} .

The algorithm works by modifying the solution z^* and the dual solution Λ^* . An edge $uv \in E^*$ is called a 0-edge/ $\frac{1}{2}$ -edge/1-edge according to the value $z^*(uv)$. A modification of z^* in G^* naturally extends to a modification of z in G . Indeed, if $S \in \mathcal{L}^*$ is a shrunk node, and z^* is modified so that there is an 1-edge incident to S in G^* , then let u_1v_1 be the pre-image of this edge in G , with $u_1 \in S$. Then modify z inside S to be identical with the Λ -critical-matching M_{u_1} inside S . If there are two half-edges incident to S in G^* , then let u_1v_1, u_2v_2 be the pre-image of these edges in G , with $u_1, u_2 \in S$. Then modify z inside S to be identical with the convex combination $(1/2)(M_{u_1} + M_{u_2})$ of the Λ -critical-matchings M_{u_1} and M_{u_2} inside S .

A walk $P = v_0v_1v_2 \dots v_k$ in G^* is called an alternating walk, if every odd edge is a 0-edge and every even edge is a 1-edge. If every node occurs in P at most once, it is called an alternating path. By *alternating along the path P* , we mean modifying $z^*(v_iv_{i+1})$ to $1 - z^*(v_iv_{i+1})$ on every edge of P . If k is odd, $v_0 = v_k$ and no other node occurs twice, then P is called a *blossom* with base v_0 . The following claim is straightforward.

Claim 4.5.21 ([64, Thm 24.3]). *Let $P = v_0v_1 \dots v_{2k+1}$ be an alternating walk. Either P is an alternating path, or it contains a blossom C and an even alternating path from v_0 to the base of the blossom.* □

Half-integral Primal-Dual

Input. A subset $W \subseteq V$, a proper odd family $\mathcal{F} \subset \mathcal{O}$ with all members disjoint from W , a proper-half-integral optimal solution w to $P_{\mathcal{F}}^W(G, c)$, and an \mathcal{F} -fitting dual optimal solution Γ to $D_{\mathcal{F}}^W(G, c)$.

Output. A proper-half-integral optimal solution z to $P_{\mathcal{L}}(G, c)$ and an \mathcal{L} -fitting dual optimal solution Λ to $P_{\mathcal{L}}(G, c)$ for some $\mathcal{L} \subseteq \mathcal{F}$.

We initialize $z = w$, $\mathcal{L} = \mathcal{F}$, $\Lambda = \Gamma$, and $T = W$. The algorithm terminates if $T = \emptyset$. Let G^* , z^* be as defined above. Let $R \supseteq T$ be the set of exposed nodes and nodes incident to $\frac{1}{2}$ -edges in z^* .

Case I: There exists an alternating T - R -walk in G^ .* Let $P = v_0 \dots v_{2k+1}$ denote a shortest such walk.

- (a) If P is an alternating path, and $v_{2k+1} \in T$, then change z by alternating along P .
- (b) If P is an alternating path, and $v_{2k+1} \in R - T$, then let C denote the odd cycle containing v_{2k+1} . Change z by alternating along P , and replacing z on C by a blossom with base v_{2k+1} .
- (c) If P is not a path, then by Claim 4.5.21, it contains an even alternating path P_1 to a blossom C . Change z by alternating along P_1 , and setting $z^*(uv) = 1/2$ on every edge of C .

Case II: There exists no alternating T - R -walk in G^ .* Define

$$\begin{aligned} \mathcal{B}^+ &:= \{S \in \mathcal{V}^* : \exists \text{ an even alternating path from } T \text{ to } S\}, \\ \mathcal{B}^- &:= \{S \in \mathcal{V}^* : \exists \text{ an odd alternating path from } T \text{ to } S\}. \end{aligned}$$

For some $\varepsilon > 0$, reset

$$\Lambda(S) := \begin{cases} \Lambda(S) + \varepsilon & \text{if } S \in \mathcal{B}^+, \\ \Lambda(S) - \varepsilon & \text{if } S \in \mathcal{B}^-. \end{cases}$$

Choose ε to be the maximum value such that Λ remains feasible.

- (a) If some new edge becomes tight, then E^* is extended.
- (b) If $\Lambda(S) = 0$ for some $S \in \mathcal{L} \cap \mathcal{B}^-$ after the modification, then unshrink the node S .

The scenarios in *Case I* are illustrated in Figure 4.5.5.1. In Case II, we observe that $T \in \mathcal{B}^+$ and further, $\mathcal{B}^+ \cap \mathcal{B}^- = \emptyset$ (otherwise, there exists a $T - T$ alternating walk and hence we should be in case I). The correctness of the output follows immediately due to complementary slackness. We show the termination of the algorithm along very similar lines as the proof of termination of Edmonds' algorithm.

Let $\beta(z)$ denote the number of exposed nodes plus the number of cycles in $\text{supp}(z)$. We first note that $\beta(z) = \beta(z^*)$. This can be derived from Lemma 4.5.1(iii) (We apply this Lemma in $G \setminus T$, observing that $P_{\mathcal{L}}^T(G, c)$ is identical to $P_{\mathcal{L}}(G \setminus T, c)$). Our next lemma (Lemma 4.5.22) shows that $\beta(z)$ is non-increasing. If $\beta(z)$ is unchanged during a certain number of iterations of the algorithm, we say that these iterations form a *non-decreasing phase*. We say that the algorithm itself is non-decreasing, if $\beta(z)$ does not decrease anytime. In the next section, we investigate properties of non-decreasing phases. These results will also show that every non-decreasing phase may contain at most $|V| + |\mathcal{F}|$ iterations and therefore the algorithm terminates in strongly polynomial time.

Lemma 4.5.22. *Let z be an arbitrary solution during the algorithm, and let α be the number of odd cycles in $\text{supp}(w)$ that are absent in $\text{supp}(z)$. Then $|W| + \text{odd}(w) \geq \beta(z) + 2\alpha$. At termination, $|W| + \text{odd}(w) \geq \text{odd}(z) + 2\alpha$.*

Proof. Initially, $\beta(z) = |W| + \text{odd}(w)$. Let us check the possible changes in $\beta(z)$ during an iteration of the algorithm. In Case I(a), the number exposed nodes decreases by two. In Case I(b), both the number of exposed nodes and the number of cycles decrease by one. In Case I(c), the number of exposed nodes decreases by one, but we obtain a new odd cycle, hence $\beta(z)$ remains unchanged. In Case II, z is not modified.

The only way to remove a cycle from $\text{supp}(z)$ is by performing the operation in Case I(b). This must be executed α times, therefore $\beta(z) \leq \beta(w) - 2\alpha$. Further, there are no exposed nodes at the end of the Half-integral Primal-Dual. Thus, on termination $\beta(z) = \text{odd}(z)$, and the claim follows. \square

4.5.5.3 The non-decreasing scenario

Let us now analyze the first non-decreasing phase \mathcal{P} of the algorithm, starting from the input w . These results will also be valid for later non-decreasing phases as well. Consider an intermediate iteration with z , Λ being the solutions, \mathcal{L} being the laminar family and T being the exposed nodes. Recall that $R \supseteq T$ is the set of exposed nodes and the node sets of the 1/2-cycles. Let us define the set of outer/inner nodes of G^* as those having even/odd length alternating walk from R in G^* . Let \mathcal{N}_o and \mathcal{N}_i denote their sets, respectively. Clearly, $\mathcal{B}^+ \subseteq \mathcal{N}_o$, $\mathcal{B}^- \subseteq \mathcal{N}_i$ in Case II of the algorithm.

Lemma 4.5.23. *If \mathcal{P} is a non-decreasing phase, then if a node in \mathcal{V}^* is outer in any iteration of phase \mathcal{P} , it remains a node in \mathcal{V}^* and an outer node in every later iteration of \mathcal{P} . If a node is inner in any iteration of \mathcal{P} , then in any later iteration of \mathcal{P} , it is either an inner node, or it has been unshrunk in an intermediate iteration.*

Proof. Since \mathcal{P} is a non-decreasing phase, Cases I(a) and (b) can never be performed. We show that the claimed properties are maintained during an iteration.

In Case I(c), a new odd cycle C is created, and thus C is added to R . Let $P_1 = v_0 \dots v_{2\ell}$ denote the even alternating path with $v_0 \in T$, $v_{2\ell} \in C$. If a node $u \in \mathcal{V}^*$ had an even/odd alternating walk from v_0 before changing the solution, it will have an even/odd walk alternating from $v_{2\ell} \in R$ after changing the solution.

In Case II, the alternating paths from T to the nodes in \mathcal{B}^- and \mathcal{B}^+ are maintained when the duals are changed. The only nontrivial case is when a set S is unshrunk; then all inner and outer nodes maintain their inner and outer property by the following: if $u_1 v_1$ is a 1-edge and $u_2 v_2$ is a 0-edge entering S after unshrinking, with $u_1, u_2 \in S$, we claim that there exists an even alternating path inside S from u_1 to u_2 using only tight edges wrt Λ . Indeed, during the unshrinking, we modify z to M_{u_1} inside S . Also, by the Λ -factor-critical property, all edges of M_{u_2} are tight w.r.t. Λ . Hence the symmetric difference of M_{u_1} and M_{u_2} contains an alternating path from u_1 to u_2 .

We have to check that vertices in $\mathcal{N}_o - \mathcal{B}^+$ and $\mathcal{N}_i - \mathcal{B}^-$ also maintain their outer and inner property. These are the nodes having even/odd alternating paths from an odd cycle,

but not from exposed nodes. The nodes in these paths are disjoint from $\mathcal{B}^- \cup \mathcal{B}^+$ and are thus maintained. Indeed, if $(\mathcal{B}^- \cap \mathcal{N}_o) \setminus \mathcal{B}^+ \neq \emptyset$ or $(\mathcal{B}^+ \cap \mathcal{N}_i) \setminus \mathcal{B}^- \neq \emptyset$, then we would get an alternating walk from T to an odd cycle, giving the forbidden Case I(b). \square

The termination of the algorithm is guaranteed by the following simple corollary.

Corollary 4.5.24. *The non-decreasing phase \mathcal{P} may consist of at most $|V| + |\mathcal{F}|$ iterations.*

Proof. Case I may occur at most $|W|$ times as it decreases the number of exposed nodes. In Case II, either \mathcal{N}_i is extended, or a set is unshrunk. By Lemma 4.5.23, the first scenario may occur at most $|V|$ times and the second at most $|\mathcal{F}|$ times. \square

In the rest of the section, we focus on the case when the entire algorithm is non-decreasing.

Lemma 4.5.25. *Assume the Half-integral Primal-Dual algorithm is non-decreasing. Let Γ be the initial dual and z, Λ be the terminating solution and \mathcal{L} be the terminating laminar family. Let \mathcal{N}_o and \mathcal{N}_i denote the final sets of outer and inner nodes in G^* .*

- *If $\Lambda(S) > \Gamma(S)$ then S is an outer node in \mathcal{V}^* .*
- *If $\Lambda(S) < \Gamma(S)$, then either $S \in \mathcal{F} \setminus \mathcal{L}$, (that is, S was unshrunk during the algorithm and $\Lambda(S) = 0$) or S is an inner node in \mathcal{V}^* , or S is a node in \mathcal{V}^* incident to an odd cycle in $\text{supp}(z)$.*

Proof. If $\Lambda(S) > \Gamma(S)$, then $S \in \mathcal{B}^+$ in some iteration of the algorithm. By Lemma 4.5.23, this remains an outer node in all later iterations. The conclusion follows similarly for $\Lambda(S) < \Gamma(S)$. \square

Lemma 4.5.26. *Assume the Half-integral primal-dual algorithm is non-decreasing. Let z, Λ be the terminating solution, \mathcal{L} be the terminating laminar family and G^* the corresponding contracted graph, \mathcal{N}_o and \mathcal{N}_i be the sets of outer and inner nodes. Let $\Theta : \mathcal{V}^* \rightarrow \mathbb{R}$ be an arbitrary optimal solution to the dual $D_0(G^*, c^*)$ of the bipartite relaxation. If $S \in \mathcal{V}^*$ is incident to an odd cycle in $\text{supp}(z)$, then $\Lambda(S) = \Theta(S)$. Further $S \in \mathcal{N}_o$ implies $\Lambda(S) \leq \Theta(S)$, and $S \in \mathcal{N}_i$ implies $\Lambda(S) \geq \Theta(S)$.*

Proof. For $S \in \mathcal{N}_o \cup \mathcal{N}_i$, let $\ell(S)$ be the length of the shortest alternating path. The proof is by induction on $\ell(S)$. Recall that there are no exposed nodes in z , hence $\ell(S) = 0$ means that S is contained in an odd cycle C . Then $\Theta(S) = \Lambda(S)$ is a consequence of Lemma 4.5.12: both Θ and Λ are optimal dual solutions in G^* , and an odd cycle in the support of the primal optimum z is both Λ -factor-critical and Θ -factor-critical.

For the induction step, assume the claim for $\ell(S) \leq i$. Consider a node $U \in \mathcal{V}^*$ with $\ell(U) = i + 1$. There must be an edge f in E^* between S and U for some S with $\ell(S) = i$. This is a 0-edge if i is even and a 1-edge if i is odd.

Assume first i is even. By induction, $\Lambda(S) \leq \Theta(S)$. The edge f is tight for Λ , and $\Theta(S) + \Theta(U) \leq c^*(f)$. Consequently, $\Lambda(U) \geq \Theta(U)$ follows. Next, assume i is odd. Then $\Lambda(S) \geq \Theta(S)$ by induction. Then, $\Lambda(U) \leq \Theta(U)$ follows as f is tight for both Λ and Θ . \square

4.5.6 Convergence

Let us consider two consecutive solutions in Algorithm C-P-Matching. Let x be the unique proper-half-integral optimal solution to $P_{\mathcal{F}}(G, c)$ and Π be an \mathcal{F} -positively-fitting dual optimal solution to $D_{\mathcal{F}}(G, c)$. We define $\mathcal{H}' = \{S : S \in \mathcal{F}, \Pi(S) > 0\}$ and \mathcal{H}'' based on odd cycles in x , and use the proper odd family $\mathcal{H} = \mathcal{H}' \cup \mathcal{H}''$ for the next iteration. Let y be the unique proper-half-integral optimal solution to $P_{\mathcal{H}}(G, c)$, and let Ψ be an \mathcal{H} -positively-fitting dual optimal solution to $D_{\mathcal{H}}(G, c)$. We already know that Π is an \mathcal{H} -fitting dual feasible solution to $D_{\mathcal{H}}(G, c)$ by Lemma 4.5.17.

Let us now contract all maximal sets $S \in \mathcal{H}$ with $\Psi(S) > 0$ w.r.t. Ψ to obtain the graph $\hat{G} = (\hat{V}, \hat{E})$ with cost \hat{c} . Note that by Lemma 4.5.12, Π and Ψ are identical inside S , hence this is the same as contracting w.r.t. Π . Let \hat{x} , \hat{y} , $\hat{\Pi}$, and $\hat{\Psi}$ be the images of x , y , Π , and Ψ , respectively.

Let $\bar{\mathcal{H}}'' = \{S : S \in \mathcal{H}'', \Psi(S) > 0\}$, and let $W = \cup \bar{\mathcal{H}}''$ denote the union of the members of $\bar{\mathcal{H}}''$. Let \hat{W} denote the image of W . Then \hat{W} is the set of exposed nodes for \hat{x} in \hat{G} , whereas the image of every set in $\mathcal{H}'' \setminus \bar{\mathcal{H}}''$ is an odd cycle in \hat{x} . Let $\zeta = \{T \in \mathcal{H}' : T \cap W = \emptyset\}$, $\mathcal{K} = \{T \in \zeta : \Psi(T) = 0\}$ and $\hat{\zeta}$ and $\hat{\mathcal{K}}$ be their images. Here, we observe that $\hat{\mathcal{K}}$ contains all sets in $\hat{\zeta}$ of size at least 3.

Using the notation above, we will couple the solutions of the Half-integral Primal-Dual algorithm and the solutions of our cutting plane algorithm. For this, we will start the Half-integral Primal-Dual algorithm in \hat{G} with \hat{W} , from the initial primal and dual solutions \hat{x} and $\hat{\Pi}$. Claim 4.5.27(ii) justifies the validity of this input choice for the Half-integral Primal-Dual.

Claim 4.5.27. (i) For every $\hat{\mathcal{L}} \subseteq \hat{\mathcal{K}}$, \hat{y} is the unique optimal solution to $P_{\hat{\mathcal{L}}}(\hat{G}, \hat{c})$ and $\hat{\Psi}$ is an optimal solution to $D_{\hat{\mathcal{L}}}(\hat{G}, \hat{c})$.

(ii) \hat{x} is a proper-half-integral optimal solution to $P_{\hat{\mathcal{K}}}^{\hat{W}}(\hat{G}, \hat{c})$ and $\hat{\Pi}$ is a $\hat{\mathcal{K}}$ -positively-fitting dual optimal solution to $D_{\hat{\mathcal{K}}}^{\hat{W}}(\hat{G}, \hat{c})$.

Proof. (i) For $\hat{\mathcal{L}} = \emptyset$, both conclusions follow by Corollary 4.5.3 – y is the unique optimal solution to $P_{\mathcal{H}}(G, c)$ and Ψ is an \mathcal{H} -positively-fitting dual optimum.

For an arbitrary $\hat{\mathcal{L}} \subseteq \hat{\mathcal{K}}$, since $y(\delta(S)) \geq 1$ for every $S \in \mathcal{H}'$ in the pre-image of $\hat{\mathcal{L}}$, \hat{y} is a feasible solution to $P_{\hat{\mathcal{L}}}(\hat{G}, \hat{c})$. Since \hat{y} is optimum to the bipartite relaxation, this implies optimality of \hat{y} to $P_{\hat{\mathcal{L}}}(\hat{G}, \hat{c})$. Now, $\hat{\Psi}$ is optimum to $D_{\hat{\mathcal{L}}}(\hat{G}, \hat{c})$ since $\hat{\Psi}$ satisfies complementary slackness with \hat{y} . Uniqueness follows since a different optimal solution to $P_{\hat{\mathcal{L}}}(\hat{G}, \hat{c})$ would also be optimal to $P_0(\hat{G}, \hat{c})$, but \hat{y} is known to be the unique optimum to $P_0(\hat{G}, \hat{c})$.

(ii) We will use Lemma 4.5.1 on an appropriate graph. We first setup the parameters that we will be using to apply this Lemma. First observe that x is the unique optimum to $P_{\mathcal{F}}(G, c)$ as well as $P_{\mathcal{H}'}(G, c)$ and Π is an optimum to $D_{\mathcal{H}'}(G, c)$.

Let G^W denote the graph $G \setminus W$, c^W denote the cost function restricted to this graph. Let Π^W denote Π restricted to the set ζ and x^W denote the solution x projected on G^W . Since $x(\delta(W, V \setminus W)) = 0$, x^W is the unique optimal solution to $P_{\zeta}(G^W, c^W)$. By complementary slackness, Π^W is optimum to $D_{\zeta}(G^W, c^W)$.

We now apply Lemma 4.5.1(iii) by considering the graph G^W with cost c^W and the laminar odd family ζ on the vertices of G^W . Now, Π^W is a feasible solution to $D_{\zeta}(G^W, c^W)$, x^W is an optimum to $P_{\zeta}(G^W, c^W)$. Let $S \in \zeta \setminus \mathcal{K}$, i.e., $\Psi(S) > 0$. Since $S \in \zeta \setminus \mathcal{H}'$ we have $\Pi(S) = \Pi^W(S) > 0$ and hence S is a Π^W -factor-critical set. Contract all such sets S wrt Π^W .

Then, by the conclusion of the Lemma, we have that \hat{x}^W is the unique proper-half-integral optimum to $P_{\hat{\zeta}}(\hat{G}^{\hat{W}}, \hat{c}^{\hat{W}})$. Since \hat{K} is the set of nonsingular elements of \hat{N} , it is equivalent to saying that \hat{x} is the unique optimum to $P_{\hat{\mathcal{K}}}^{\hat{W}}(\hat{G}, \hat{c})$. By Lemma 4.5.1(ii), we also have that $\hat{\Pi}^W$ is optimum to $D_{\hat{\zeta}}(\hat{G}^{\hat{W}}, \hat{c}^{\hat{W}})$ and hence is optimum to $D_{\hat{\mathcal{K}}}(\hat{G}^{\hat{W}}, \hat{c}^{\hat{W}}) = D_{\hat{\mathcal{K}}}(\hat{G} \setminus \hat{W}, \hat{c})$. Recall that every optimal solution to $D_{\hat{\mathcal{K}}}(\hat{G} \setminus \hat{W}, \hat{c})$ that is feasible to $D_{\hat{\mathcal{K}}}^{\hat{W}}(\hat{G}, \hat{c})$ is also optimal to $D_{\hat{\mathcal{K}}}^{\hat{W}}(\hat{G}, \hat{c})$. Observe that $\hat{\Pi}$ is 0 on every node and subset of \hat{W} . Consequently, all nonzero values of $\hat{\Pi}$ and $\hat{\Pi}^W$ coincide. Since $\hat{\Pi}$ is feasible to $D_{\hat{\mathcal{K}}}^{\hat{W}}(\hat{G}, \hat{c})$, we get that $\hat{\Pi}$ is also optimum to $D_{\hat{\mathcal{K}}}^{\hat{W}}(\hat{G}, \hat{c})$. Further, $\hat{\Pi}$ is $\hat{\mathcal{K}}$ -positively-fitting by definition of $\hat{\mathcal{K}}$. \square

Lemma 4.5.28. *Suppose we start the Half-integral Primal-Dual algorithm in \hat{G} , \hat{c} , $\hat{\mathcal{K}}$, \hat{W} , from the initial primal and dual solutions \hat{x} and $\hat{\Pi}$. Then the output \hat{z} of the algorithm is equal to \hat{y} .*

Proof. Since \hat{z} is the output of the Half-integral Primal-Dual algorithm, it is an optimal solution to $P_{\hat{\mathcal{L}}}(\hat{G}, \hat{c})$ for some $\hat{\mathcal{L}} \subseteq \hat{\mathcal{K}}$. By Claim 4.5.27(i), \hat{y} is the unique optimal solution to this program, and consequently, $\hat{z} = \hat{y}$. \square

We are now ready to prove the non-increasing property of the number of odd cycles.

Lemma 4.5.29. *Assume the cost function c satisfies (*). Then $\text{odd}(x)$ is non-increasing during the algorithm.*

Proof. Using the notation above, let us start the Half-integral Primal-Dual algorithm in \hat{G} , \hat{c} , $\hat{\mathcal{K}}$, \hat{W} , from the initial primal and dual solutions \hat{x} and $\hat{\Pi}$. Let \hat{z} be the output of the Half-integral Primal-Dual algorithm.

By Lemma 4.5.28, $\hat{z} = \hat{y}$. We first observe that $\text{odd}(x) = |W| + \text{odd}(\hat{x})$. This easily follows by Lemma 4.5.1(iii), applied in $G \setminus W$. Let $\alpha = |\mathcal{H}'' \setminus \bar{\mathcal{H}}''|$. There is an odd cycle in $\text{supp}(x)$ corresponding to each member of $\mathcal{H}'' \setminus \bar{\mathcal{H}}''$. None of these cycles may be contained in $\text{supp}(\hat{z}) = \text{supp}(\hat{y})$ as otherwise the corresponding cut in \mathcal{H}'' would be violated by \hat{y} . Thus Lemma 4.5.22 implies $\text{odd}(\hat{y}) = \text{odd}(\hat{z}) \leq |W| + \text{odd}(\hat{x}) - 2\alpha$ and Lemma 4.5.1(iii) implies $\text{odd}(y) = \text{odd}(\hat{y})$. Hence, $\text{odd}(y) \leq \text{odd}(x) - 2\alpha$. \square

The next claim formulates a simple consequence of the above proof.

Claim 4.5.30. *If $\text{odd}(y) = \text{odd}(x)$, then $\mathcal{H}'' = \bar{\mathcal{H}}''$. Further, the Half-integral Primal-Dual algorithm applied in \hat{G} , \hat{c} , $\hat{\mathcal{K}}$, \hat{W} , with starting solution \hat{x} , $\hat{\Pi}$ is non-decreasing. \square*

This claim already implies that the new cuts are retained in the next iteration when the number of odd cycles did not decrease. To analyze the scenario $\text{odd}(x) = \text{odd}(y)$, the results in Section 4.5.5.3 are applicable since the algorithm was non-decreasing. Let us start the Half-integral Primal-Dual algorithm in \hat{G} , \hat{c} , $\hat{\mathcal{K}}$, \hat{W} , from the initial primal and dual solutions \hat{x} and $\hat{\Pi}$. Consider the final dual solution $\hat{\Lambda}$ with corresponding laminar family $\hat{\mathcal{L}}$ and define Λ in G as follows.

If $S \subsetneq T$ for some $T \in \mathcal{H}$, $\Psi(T) > 0$, then set $\Lambda(S) = \Psi(S)$ (this defines the dual solutions for sets and nodes inside T that were contracted to obtain \hat{G}). If $\hat{S} \in \hat{\mathcal{L}} \cup \hat{\mathcal{V}}$, then set $\Lambda(S) = \hat{\Lambda}(\hat{S})$ for its pre-image S (this defines the dual solutions for sets and nodes on or outside T that were contracted to obtain \hat{G}).

Claim 4.5.31. *Λ is an optimal solution to $D_{\mathcal{H}}(G, c)$.*

Proof. We show feasibility here. Optimality follows by complementary slackness with y and the definition of shrinking. The nontrivial part to show feasibility is to verify that $\Lambda(S) \geq 0$ if $S \in \mathcal{H}$. This is straightforward if $S \subseteq T$ for some $T \in \mathcal{H}$, $\Psi(T) > 0$, or if $\hat{S} \in \hat{\mathcal{L}}$. Let $S \in \mathcal{H}$ be a maximal set with $\Psi(S) > 0$, and let s denote its image in \hat{G} . Then during the Half-integral Primal-Dual algorithm, $\hat{\Lambda}(s)$ might have decreased below 0. We show that this cannot be the case.

By Claim 4.5.27(i), $\hat{\Psi}$ is an optimal solution to the bipartite relaxation $D_0(\hat{G}, \hat{c})$. If $\hat{\Lambda}(s) < 0$, then $\hat{\Lambda}(s) < 0 < \hat{\Pi}(s)$. We started the algorithm with the dual solution $\Gamma = \hat{\Pi}$, therefore by Lemma 4.5.25, either (1) s is an inner node or (2) $s \in \mathcal{V}^*$ and s is incident to an odd cycle in $\text{supp}(z)$. In both cases, by Lemma 4.5.26, we get that $\Lambda(S) = \hat{\Lambda}(s) \geq \hat{\Psi}(s) = \Psi(S) \geq 0$. \square

Lemma 4.5.32. *Assume $\text{odd}(x) = \text{odd}(y)$ for the consecutive solutions x and y . Then $\hat{\Lambda} = \hat{\Psi}$ and hence $\Lambda = \Psi$.*

Proof. By Lemma 4.5.12, if S is such that $\Psi(S) > 0$, then Ψ is identical to Π inside S , and so by definition, also to Λ . Therefore, it suffices to prove $\hat{\Lambda} = \hat{\Psi}$. By Claim 4.5.27(i), $\hat{\Psi}$ is

an optimal solution to $D_0(\hat{G}, \hat{c})$. The following claim completes the proof by showing that $h(\Lambda, \Pi) \leq h(\Psi, \Pi)$, and equality can hold only if Λ and Ψ are identical. \square

Claim 4.5.33. *For every $S \in \hat{\mathcal{V}} \cup \hat{\mathcal{K}}$, $|\hat{\Lambda}(S) - \hat{\Pi}(S)| \leq |\hat{\Psi}(S) - \hat{\Pi}(S)|$ and equality holds only if $\hat{\Lambda}(S) = \hat{\Psi}(S)$.*

Proof. The claim will follow by showing that for every $S \in \hat{\mathcal{V}} \cup \hat{\mathcal{K}}$, either $\hat{\Pi}(S) \leq \hat{\Lambda}(S) \leq \hat{\Psi}(S)$ or $\hat{\Pi}(S) \geq \hat{\Lambda}(S) \geq \hat{\Psi}(S)$. By Claim 4.5.30, $\text{odd}(x) = \text{odd}(y)$ implies that the Half-integral primal-dual algorithm was non-decreasing. Let \hat{z} be the terminating primal solution, and let $G^* = (\mathcal{V}^*, E^*)$, c^* be the corresponding contraction of maximal sets of $\hat{\mathcal{L}}$ in \hat{G} , \hat{c} . Note that $\hat{\Psi}(S) = 0$ if $S \in \hat{\mathcal{L}}$. Let $\Theta : \mathcal{V}^* \rightarrow \mathbb{R}$ be defined as follows:

$$\Theta(S^*) = \begin{cases} \hat{\Psi}(S) - \Delta_{\hat{\Lambda}, \hat{\Psi}}(S) & \text{if } S \in \hat{\mathcal{L}}, \\ \hat{\Psi}(S) & \text{if } S \in \mathcal{V}^* \setminus \hat{\mathcal{L}}. \end{cases}$$

Claim 4.5.34. *Θ is an optimal solution to $D_0(G^*, c^*)$. Further, $\Theta(S) \leq 0$ holds for every $S \in \hat{\mathcal{L}}$.*

Proof. If $S \in \hat{\mathcal{L}}$, then $\hat{\Lambda}(S) > 0$ and hence $\hat{y}(\delta(S)) = 1$. $\hat{\mathcal{L}}$ is a proper odd family in \hat{G} with $\hat{\Lambda}$ being an $\hat{\mathcal{L}}$ -fitting dual (by definition of the algorithm). By Claim 4.5.27(i), \hat{y} and $\hat{\Psi}$ are optimal solutions to $P_{\hat{\mathcal{L}}}(\hat{G}, \hat{c})$ and to $D_{\hat{\mathcal{L}}}(\hat{G}, \hat{c})$, respectively. Hence, by Lemma 4.5.10, $\hat{\Psi}$ is consistent with $\hat{\Lambda}$ inside every set $S \in \hat{\mathcal{L}}$ such that $\hat{y}(\delta(S)) = 1$. Further, $\Delta_{\hat{\Lambda}, \hat{\Psi}}(S) \geq 0$ for every $S \in \hat{\mathcal{L}}$. Thus, by Lemma 4.5.5, Θ is an optimum dual solution to $D_0(G^*, c^*)$. The second part follows since $\hat{\Psi}(S) = 0$ and $\Delta_{\hat{\Lambda}, \hat{\Psi}}(S) \geq 0$ if $S \in \hat{\mathcal{L}}$. \square

Now consider $S \subseteq \mathcal{V}$. If $\hat{\Lambda}(S) > \hat{\Pi}(S)$, then by Lemma 4.5.25, we have that $S \in \mathcal{V}^*$ and $S \in \mathcal{N}_o$. Consequently, by Lemma 4.5.26, $\Theta(S) \geq \hat{\Lambda}(S)$. If $S \in \hat{\mathcal{L}}$, then $0 \geq \Theta(S^*) \geq \hat{\Lambda}(S)$, a contradiction to $\hat{\Lambda}(S) > 0$. Thus, $S = \{s\}$ for some $s \in \hat{\mathcal{V}}$. Hence, $\hat{\Psi}(s) = \Theta(s) \geq \hat{\Lambda}(s) > \hat{\Pi}(s)$.

If $\hat{\Lambda}(S) < \Pi(S)$, then by Lemma 4.5.25, we have that either (1) $S \in \hat{\mathcal{K}} \setminus \hat{\mathcal{L}}$, that is, $\hat{\Lambda}(S) = 0$ and S was unshrunk or (2) $S \in \mathcal{N}_i$ or (3) $S \in \mathcal{V}^*$ and S is incident to an odd cycle C in $\text{supp}(z)$. If (1), then $\hat{\Psi}(S) = 0 = \hat{\Lambda}(S) < \Pi(S)$. In both cases (2) and (3),

Lemma 4.5.26 gives $\Theta(S) \leq \hat{\Lambda}(S)$. If $S \in \hat{\mathcal{L}}$, then $\hat{\Psi}(S) = 0 \leq \hat{\Lambda}(S) < \hat{\Pi}(S)$. If $S = \{s\}$ for some $s \in \hat{V}$, then $\hat{\Psi}(s) = \Theta(s) \leq \hat{\Lambda}(s) \leq \hat{\Pi}(s)$. \square

We are now ready to prove the main lemma needed to show convergence.

Lemma 4.5.35. *Assume the cost function c satisfies (*) and that $\text{odd}(x)$ does not decrease between iterations i and j . Let \mathcal{F}_k be the set of blossom inequalities imposed in the k 'th iteration and $\mathcal{H}_k'' = \mathcal{F}_k \setminus \mathcal{F}_{k-1}$ be the subset of new inequalities. Then,*

$$\bigcup_{k=i+1}^j \mathcal{H}_k'' \subseteq \mathcal{F}_{j+1}.$$

Proof. Let x_i be the solution in the i 'th iteration (above, we used $x = x_i$ and $y = x_{i+1}$). Assume the number of odd cycles does not decrease between iterations i and j . By Claim 4.5.30, if we run the Half-integral Primal-Dual algorithm between x_k and x_{k+1} , for $i \leq k < j$, it is always non-decreasing.

We first run the Half-integral Primal-Dual on the contracted graph $\hat{G} = \hat{G}_i$ starting from primal solution $\hat{x} = \hat{x}_i$ and dual solution $\hat{\Pi} = \hat{\Pi}_i$. Lemmas 4.5.28 and 4.5.32 show that it terminates with the primal optimal solution $\hat{y} = \hat{x}_{i+1}$ and dual optimal solution $\hat{\Lambda} = \hat{\Psi}$.

For $j = i + 1$, the statement follows by Claim 4.5.30 since $\bar{\mathcal{H}}'' = \mathcal{H}''$ means that all cuts added in iteration i have positive dual value in iteration $i + 1$. Further, all sets in \mathcal{H}'' were contracted to exposed nodes in \hat{x}_i . By Lemma 4.5.25, these will be outer nodes on termination of the Half-Integral Primal-Dual algorithm as well. Let G^* be the contracted graph upon termination of the Half-Integral Primal-Dual algorithm.

Let $\mathcal{J} = \mathcal{J}' \cup \mathcal{J}''$ be the set of cuts imposed in the $(i + 2)$ 'th round, with $\mathcal{J} = \{S \in \mathcal{H} : \Psi(Z) > 0\}$, and let \mathcal{J}'' be defined according to odd cycles in x_{i+1} . Let Φ be the extremal dual optimal solution to $D_{\mathcal{J}}(G, c)$.

Let us run the Half-integral Primal-Dual algorithm from x_{i+1} to x_{i+2} . We start the algorithm with the contracted graph \hat{G}_{i+1} , which results by contracting all sets with $\Phi(S) > 0$, $S \in \mathcal{J}$. Let \hat{G}_{i+1}^* be the initial contraction of \hat{G}_{i+1} used by the algorithm.

The key observation is that while the underlying graphs \hat{G}_i and \hat{G}_{i+1} are different, \hat{G}_{i+1}^* can be obtained from G^* by contracting those odd cycles corresponding to the members of

\mathcal{J}'' . Every other node that was inner or outer node in G^* will also be inner or outer node in \hat{G}_{i+1}^* , including the members of \mathcal{H}'' . By Lemma 4.5.25, the members of \mathcal{H}'' will be outer nodes at termination, along with the new outer nodes \mathcal{J}'' .

Iterating this argument one can show that every set that was imposed based on an odd cycle between iterations i and k will be outer nodes at the termination of the Half-integral Primal-Dual from x_k to x_{k+1} . \square

We conclude the chapter with the main theorem about the convergence of Algorithm C-P-Matching.

Theorem 4.5.36. *For any graph $G = (V, E)$ on n nodes and cost function $c : E \rightarrow \mathbb{Z}$, let \tilde{c} denote the perturbation of c . Using cost function \tilde{c} , there exists a sequence of sets of blossom inequalities (cutting planes) such that*

- (i) *every LP encountered has a laminar family of blossom inequalities,*
- (ii) *the optimum of each LP is unique and half-integral, and*
- (iii) *the total number of iterations to arrive at a minimum-cost perfect matching is $O(n \log n)$.*

Moreover, the set of inequalities used at each step can be identified by solving an LP of the same size as the current LP. The optimal solution returned for \tilde{c} is also optimal to the original cost c .

Proof. We use Algorithm C-P-Matching given in Figure 13 for a perturbed cost function. By Lemma 4.5.19, the perturbed cost function satisfies (*). Let i denote the index of the iteration. We prove by induction on i that every intermediate solution x_i is proper-half-integral and (i) follows immediately by the choice of the algorithm. The proper-half-integral property holds for the initial solution x_0 by Proposition 4.2.1. The induction step follows by Lemmas 4.5.4 and 4.5.17 and the uniqueness property. Further, by Lemma 4.5.29, the number of odd cycles in the support does not increase.

Assume the number of cycles in the i 'th phase is ℓ , and we have the same number of odd cycles ℓ in a later iteration j . Between iterations i and j , the set \mathcal{H}_k'' always contains

ℓ cuts, and thus the number of cuts added is at least $\ell(j - i)$. By Lemma 4.5.35, all cuts in $\bigcup_{k=i+1}^j \mathcal{H}_k''$ are imposed in the family \mathcal{F}_{j+1} . Since \mathcal{F}_{j+1} is a laminar odd family, it can contain at most $n/2$ subsets, and therefore $j - i \leq n/2\ell$. Consequently, the number of cycles must decrease from ℓ to $\ell - 1$ within $n/2\ell$ iterations. Since $\text{odd}(x_0) \leq n/3$, the number of iterations is at most $O(n \log n)$.

Finally, we show that optimal solution returned by the algorithm using \tilde{c} is also optimal for the original cost function. Let M be the optimal matching returned by \tilde{c} , and assume for a contradiction that there exists a different perfect matching M' with $c(M') < c(M)$. Since c is integral, it means $c(M') \leq c(M) - 1$. In the perturbation, since $c(e) < \tilde{c}(e)$ for every $e \in E$, we have $c(M) < \tilde{c}(M)$ and since $\sum_{e \in E} (\tilde{c}(e) - c(e)) < 1$, we have $\tilde{c}(M') < c(M') + 1$. This gives $\tilde{c}(M') < c(M') + 1 \leq c(M) < \tilde{c}(M)$, a contradiction to the optimality of M for \tilde{c} . \square

4.6 Conclusion

The cutting plane algorithm is a well-known heuristic for solving IP in practice. It has been studied extensively in order to develop *deeper* cuts, which intuitively appear to expedite the process of convergence to integer solution. Yet, a rigorous theoretical treatment of the practical efficiency of cutting plane algorithms using simple cut-generation methods has remained elusive. A reasonable first step is to find efficient cutting-plane implementations for integer polytopes with polynomial-time separation oracles. We make progress towards this by showing an efficient cutting plane implementation for minimum-cost perfect matching.

Our initial motivation was to bound the number of iterations of the cutting plane method using the Padberg-Rao separation procedure as a cut-generation procedure. This question remains open and any analysis would have to deal with non-half-integral solutions. In retrospect, it is plausible that our approach which avoids the use of Padberg-Rao procedure is more efficient in practice.

Within our algorithm, Lemma 4.5.4 shows that it is sufficient to use positively-fitting dual optimal solutions to maintain proper-half-integrality. Can we prove efficient convergence of our cutting plane algorithm using positively-fitting dual optimal solutions (without

using extremal dual solutions)? We believe that such a proof of convergence should depend on whether the following adversarial variation of Edmonds' algorithm for perfect matching runs in polynomial time: We run the Edmonds' perfect matching algorithm, but after every few iterations, the adversary replaces the current dual solution with a different one, still satisfying complementary slackness with the (unchanged) primal solution.

Given the encouraging results in this chapter, it would be interesting to prove efficient convergence of the cutting plane method for other combinatorial polytopes. For example, one could try a similar approach for finding an optimal solution for b -matchings. Another direction could be to try this approach for optimizing over the subtour elimination polytope.

REFERENCES

- [1] ALON, N., KRIVELEVICH, M., and SUDAKOV, B., “Finding a large hidden clique in a random graph,” in *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pp. 594–598, 1998.
- [2] ALON, N., AWERBUCH, B., and AZAR, Y., “The online set cover problem,” in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pp. 100–105, 2003.
- [3] BAFNA, V., BERMAN, P., and FUJITO, T., “A 2-approximation algorithm for the undirected feedback vertex set problem,” *SIAM Journal of Discrete Mathematics*, vol. 12, no. 3, pp. 289–297, 1999.
- [4] BAR-YEHUDA, R. and GEIGER, D., “Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and Bayesian inference,” *SIAM Journal on Computing*, vol. 27, no. 4, pp. 942–959, 1998.
- [5] BECKER, A. and GEIGER, D., “Optimization of Pearl’s method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem,” *Artificial Intelligence*, vol. 83, pp. 167–188, May 1996.
- [6] BEIER, R. and VÖCKING, B., “Random knapsack in expected polynomial time,” in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pp. 232–241, 2003.
- [7] BOLLOBÁS, B., *Random Graphs*. Cambridge studies in advanced mathematics, Cambridge University Press, 2001.
- [8] BRODER, A., FRIEZE, A., and UPFAL, E., “On the satisfiability and maximum satisfiability of random 3-cnf formulas,” in *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete algorithms*, SODA '93, pp. 322–330, 1993.
- [9] CHANDRASEKARAN, K., KARP, R., MORENO-CENTENO, E., and VEMPALA, S., “Algorithms for implicit hitting set problems,” in *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pp. 614–629, 2011.
- [10] CHANDRASEKARAN, K., VÉGH, L., and VEMPALA, S., “The cutting plane method is polynomial for perfect matchings,” *Submitted*, 2012.
- [11] CHANDRASEKARAN, K. and VEMPALA, S., “A discrepancy-based approach to random integer programs,” *Submitted*, 2012.
- [12] CHAO, M. T. and FRANCO, J., “Probabilistic analysis of a generalization of the unit-clause literal selection heuristics for the k -satisfiability problem,” *Information Sciences*, vol. 51, pp. 289–314, Aug 1990.
- [13] CHAO, M. T. and FRANCO, J., “Probabilistic analysis of two heuristics for the 3-satisfiability problem,” *SIAM Journal on Computing*, vol. 15, pp. 1106–1118, 1986.

- [14] CHARIKAR, M., NEWMAN, A., and NIKOLOV, A., “Tight hardness for minimizing discrepancy,” in *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’11, pp. 1607–1614, 2011.
- [15] CHVÁTAL, V., “Edmonds polytopes and a hierarchy of combinatorial problems,” *Discrete Mathematics*, vol. 4, pp. 305–337, Apr 1973.
- [16] CHVÁTAL, V., COOK, W., and HARTMANN, M., “On cutting-plane proofs in combinatorial optimization,” *Linear Algebra and its Applications*, vol. 114-115, pp. 455–499, Mar 1989.
- [17] CHVÁTAL, V. and REED, B., “Mick gets some (the odds are on his side),” in *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science*, FOCS ’92, pp. 620–627, 1992.
- [18] COOK, W., COULLARD, C. R., and TURÁN, G., “On the complexity of cutting-plane proofs,” *Discrete Appl. Math.*, vol. 18, pp. 25–38, Nov. 1987.
- [19] COOK, W., “Fifty-plus years of combinatorial integer programming,” *50 Years of Integer Programming 1958-2009*, pp. 387–430, 2010.
- [20] CUNNINGHAM, W. H. and MARSH, A. B., “A primal algorithm for optimum matching,” *Polyhedral Combinatorics*, vol. 8, pp. 50–72, 1978.
- [21] DANTZIG, G., FULKERSON, R., and JOHNSON, S., “Solution of a large-scale traveling-salesman problem,” *Operations Research*, vol. 2, pp. 393–410, Nov 1954.
- [22] DANTZIG, G., “On the significance of solving some linear programs with some integer variables,” *Econometrica*, vol. 28, pp. 30–34, 1960.
- [23] DASGUPTA, S. and SCHULMAN, L. J., “A two-round variant of EM for Gaussian mixtures,” in *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, UAI ’00, pp. 152–159, 2000.
- [24] DE AZEVEDO PRIBITKIN, W., “Simple upper bounds for partition functions,” *The Ramanujan Journal*, vol. 18, no. 1, pp. 113–119, 2007.
- [25] DE LA VEGA, F., “Induced trees in sparse random graphs,” *Graphs and Combinatorics*, vol. 2, pp. 227–231, 1986.
- [26] DE LA VEGA, F., “The largest induced tree in a sparse random graph,” *Random Structures and Algorithms*, vol. 9, no. 1-2, pp. 93–97, 1996.
- [27] EDMONDS, J., “Maximum matching and a polyhedron with 0, 1-vertices,” *Journal of Research of the National Bureau of Standards*, vol. 69, no. 1-2, pp. 125–130, 1965.
- [28] EDMONDS, J., “Paths, trees, and flowers,” *Canadian Journal of Mathematics*, vol. 17, no. 3, pp. 449–467, 1965.
- [29] FISCHETTI, M. and LODI, A., “Optimizing over the first Chvátal closure,” *Mathematical Programming*, vol. 110, no. 1, pp. 3–20, 2007.
- [30] FRIEDGUT, E., “Sharp thresholds of graph properties and the k-sat problem,” *Journal of the American Mathematical Society*, vol. 12, pp. 1017–1054, 1998.

- [31] FRIEZE, A. and KANNAN, R., “A new approach to the planted clique problem,” in *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, FSTTCS '08, pp. 187–198, 2008.
- [32] FRIEZE, A. M., “On the independence number of random graphs,” *Discrete Mathematics*, vol. 81, no. 2, pp. 171–175, 1990.
- [33] FURST, M. L. and KANNAN, R., “ Succinct certificates for almost all subset sum problems,” *SIAM Journal on Computing*, vol. 18, pp. 550–558, Jun 1989.
- [34] GOMORY, R., “Outline of an algorithm for integer solutions to linear programs,” *Bulletin of the American Mathematical Society*, vol. 64, pp. 275–278, 1958.
- [35] GOMORY, R., “Solving linear programs in integers,” in *Proceedings of Symposia in Applied Mathematics*, vol. 10, pp. 211–215, July 1960.
- [36] GOMORY, R., “An algorithm for integer solutions to linear programs,” in *Recent advances in Mathematical Programming*, pp. 269–302, 1963.
- [37] GRIMMETT, G. and MCDIARMID, C., “On colouring random graphs,” *Mathematical Proceedings of Cambridge Philosophical Society*, vol. 77, pp. 313–324, 1975.
- [38] GRÖTSCHEL, M. and HOLLAND, O., “Solving matching problems with linear programming,” *Mathematical Programming*, vol. 33, no. 3, pp. 243–259, 1985.
- [39] GRÖTSCHEL, M., LOVÁSZ, L., and SCHRIJVER, A., “The ellipsoid method and its consequences in combinatorial optimization,” *Combinatorica*, vol. 1, pp. 169–197, 1981.
- [40] JERRUM, M., “Large cliques elude the metropolis process,” *Random Structures and Algorithms*, vol. 3, pp. 347–359, 1992.
- [41] KANNAN, R., “Minkowski’s convex body theorem and integer programming,” *Mathematics of Operations Research*, vol. 12, pp. 415–440, Aug 1987.
- [42] KARP, R., “Reducibility among combinatorial problems,” in *Complexity of Computer Computations*, pp. 85–103, 1972.
- [43] KARP, R. and PAPADIMITRIOU, C., “On linear characterization of combinatorial optimization problems,” *SIAM Journal on Computing*, vol. 11, pp. 620–632, 1982.
- [44] KARP, R. and MORENO-CENTENO, E., “The implicit hitting set approach to solve combinatorial optimization problems with an application for multi-genome alignment,” *Manuscript*, 2010.
- [45] KARP, R. M., “Heuristic algorithms in computational molecular biology,” *Journal of Computer and System Sciences*, vol. 77, pp. 122–128, Jan 2011.
- [46] KHACHIYAN, L., “A polynomial algorithm in linear programming (in Russian),” *Doklady Akademii Nauk SSSR*, vol. 244, pp. 1093–1096, 1979.
- [47] KLEITMAN, D., “On a combinatorial conjecture of Erdős,” *Journal of Combinatorial Theory*, vol. 1, pp. 209–214, 1966.
- [48] KORTE, B. and VYGEN, J., *Combinatorial Optimization*. Springer, 2000.

- [49] LENSTRA, A. K., LENSTRA, H. W., and LOVÁSZ, L., “Factoring polynomials with rational coefficients,” *Mathematische Annalen*, vol. 261, pp. 515–534, 1982.
- [50] LENSTRA, H. W., “Integer programming with a fixed number of variables,” *Mathematics of Operations Research*, vol. 8, no. 4, pp. 538–548, 1983.
- [51] LI, D. M. and LIU, Y. P., “A polynomial algorithm for finding the minimum feedback vertex set of a 3-regular simple graph,” *Acta Math Sci*, vol. 19, no. 4, pp. 375–381, 1999.
- [52] LIANG, Y. D., “On the feedback vertex set problem in permutation graphs,” *Information Processing Letters*, vol. 52, no. 3, pp. 123–129, 1994.
- [53] LIANG, Y. D. and CHANG, M.-S., “Minimum feedback vertex sets in cocomparability graphs and convex bipartite graphs,” *Acta Informatica*, vol. 34, no. 5, pp. 337–346, 1997.
- [54] LOVÁSZ, L. and PLUMMER, M., *Matching Theory*. North Holland, 1986.
- [55] LOVÁSZ, L., SPENCER, J., and VESZTERGOMBI, K., “Discrepancy of set-systems and matrices,” *European Journal of Combinatorics*, vol. 7, pp. 151–160, April 1986.
- [56] LOVETT, S. and MEKA, R., “Constructive Discrepancy Minimization by Walking on the Edges,” *arXiv: abs/1203.5747*, 2012.
- [57] MATOUŠEK, J. and SPENCER, J., “Discrepancy in arithmetic progressions,” *American Mathematical Society*, vol. 9, pp. 195–204, January 1996.
- [58] MCDIARMID, C., “Colouring random graphs,” *Annals of Operations Research*, vol. 1, pp. 183–200, 1984.
- [59] NEMHAUSER, G. and WOLSEY, L., *Integer and Combinatorial Optimization*. Wiley-Interscience, 1999.
- [60] PADBERG, M. W. and RAO, M. R., “The Russian method for linear programming III: Bounded integer programming,” *Research Report 81-39, New York University*, 1981.
- [61] PADBERG, M. and RAO, M., “Odd minimum cut-sets and b -matchings,” *Mathematics of Operations Research*, vol. 7, no. 1, pp. 67–80, 1982.
- [62] PATAKI, G., TURAL, M., and WONG, E. B., “Basis reduction and the complexity of branch-and-bound,” in *Proceedings of the twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '10*, pp. 1254–1261, 2010.
- [63] PUDLAK, P., “Lower bounds for resolution and cutting plane proofs and monotone computations,” *The Journal of Symbolic Logic*, vol. 62, pp. 981–998, Sep 1997.
- [64] SCHRIJVER, A., *Combinatorial Optimization*. Springer, 2003.
- [65] SEYMOUR, P. D., “Packing directed circuits fractionally,” *Combinatorica*, vol. 15, no. 2, pp. 281–288, 1995.
- [66] SPENCER, J., “Six standard deviations suffice,” *Transactions of the American Mathematical Society*, vol. 289, pp. 679–706, 1985.

- [67] SPENCER, J. and SUBRAMANIAN, C., “On the size of induced acyclic subgraphs in random digraphs,” *Discrete Mathematics and Theoretical Computer Science*, vol. 10, no. 2, pp. 47–54, 2008.
- [68] SPIELMAN, D. A. and TENG, S. H., “Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time,” *Journal of the ACM*, vol. 51, pp. 385–463, May 2004.
- [69] TRICK, M., *Networks with Additional Structured Constraints*. PhD thesis, Georgia Institute of Technology, 1987.
- [70] UENO, S., KAJITANI, Y., and GOTOH, S., “On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three,” *Discrete Mathematics*, vol. 72, no. 1-3, pp. 355–360, 1988.