SPARSITY IN INTEGER PROGRAMMING

A Dissertation
Presented to
The Academic Faculty

By

Andres Iroume

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology

May 2017

SPARSITY IN INTEGER PROGRAMMING

Approved by:

Dr. Santanu Dey, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Marco Molinaro
Department of Computer Science
Pontificia Universidade Catolica do Rio de Janeiro

Dr. Andy Sun
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Alejandro Toriello
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. David Goldsman
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: December 12, 2016

I think that it is a relatively good approximation to truth - which is much too complicated to allow anything but approximations - that mathematical ideas originate in empirics, although the genealogy is sometimes long and obscure.

John von Neumann

To Stefania

ACKNOWLEDGEMENTS

I would like to first thank my advisor Dr. Santanu Dey for his guidance and support during my studies. I am tremendously grateful for Santanu's dedication and his patience, without which, this thesis would never have been finished. I am also deeply thankful to Dr. Marco Molinaro who was a fundamental part of most of the work in this thesis, both during his time in Georgia Tech as well as later on. I would also like to thank the other members of the committee: Dr. David Goldsman, Dr. Andy Sun and Dr. Alejandro Toriello for their time and feedback that have helped to improve this research.

Thanks to the other members of the ISyE faculty. In particular those who taught the graduate courses and were always willing to answer my questions: Dr. Craig Tovey, Dr. Ton Dieker, Dr. Shabbir Ahmed, Dr. Joel Sokol, Dr. Gary R. Parker, Dr. George Nemhauser, Dr. Robin Thomas and Dr. Anton Kleywegt. Also thanks for the administrative services of Amanda Ford, Pam Morrison and Mark Reese and many others.

I would like to thank all my friends and fellow students in the PhD program. I met many wonderful people from many different countries. Too many people have helped me during this time to list all their names but I hope they know how much I appreciate it. The long hours at ISyE discussing work and many other things with my friends and colleagues were always a source of both motivation and enjoyment. They made my time in Atlanta one of the best ones and I will always be thankful. In the end, the people I met and the time that I spend with them is what made this such a fulfilling experience.

Finally, I would like to specially thank Stefania and my parents for their constant support and encouragement during the last five years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This thesis deals with understanding the effect of sparsity in integer programming. Chapter 1 corresponds to the introduction and outlay of the thesis, Chapter 2 deals with approximating polytopes using sparse cuts under various settings. Chapter 3 discusses a variant on feasibility pump that automatically detects and harnesses sparsity. Chapter 4 deals with the ratio of the number of integral extreme points to the total number of extreme points for a family of random polytopes. Chapter 5 discusses the strength of multi-row aggregation cuts in the context of *sign-pattern* integer programs (IPs).

The paper [40] studied how well polytopes are approximated by using only sparse valid-inequalities. In Chapter 2, we consider less-idealized questions such as: effect of sparse inequalities added to linear-programming relaxation, effect on approximation by addition of a budgeted number of dense valid-inequalities, sparse-approximation of polytope under every rotation and approximation by sparse inequalities in specific directions.

In Chapter 3 we propose a variant on Feasibility pump (FP) which is a successful primal heuristic for mixed-integer linear programs (MILP). The algorithm consists of three main components: rounding fractional solution to a mixed-integer one, projection of infeasible solutions to the LP relaxation, and a randomization step used when the algorithm stalls. While many generalizations and improvements to the original Feasibility Pump have been proposed, they mainly focus on the rounding and projection steps.

We start a more in-depth study of the randomization step in Feasibility Pump. For that, we propose a new randomization step based on the WalkSAT algorithm for solving SAT instances. First, we provide *theoretical analyses* that show the potential of this randomization step; to the best of our knowledge, this is the first time any theoretical analysis of running-time of Feasibility Pump or its variants has been conducted. Moreover, we also conduct computational experiments incorporating the proposed modification into a state-of-the-art Feasibility Pump code that reinforce

the practical value of the new randomization step.

In Chapter 4 we look at a sparse packing instances and their extreme points. For such instances, we try to understand if there is a relation between the ratio of the number of integral extreme points to the total number of extreme points and the sparsity of the constraint matrix. We are able to show that there exists a family of randomly generated packing instances for which we can obtain a lower bound for this ratio that decreases as instances become denser.

Finally, in Chapter 5, we study the strength of *aggregation cuts* for *sign-pattern* integer programs (IPs). Sign-pattern IPs are a generalization of packing IPs and are of the form $\{x \in \mathbb{Z}^n_+ : Ax \leq b\}$ where for a given j, $A_{i,j}$ is either non-negative for all i or non-positive for all i. Our first result is that the *aggregation closure* for such sign-pattern IPs can be *2-approximated* by the *original 1-row closure*. This generalizes a result for packing IPs from [18]. On the other hand, unlike in the case of packing IPs, we show that the *multi-row aggregation closure* cannot be well approximated by the *original multi-row closure*. Therefore for these classes of integer programs general aggregated multi-row cutting planes can perform significantly better than just looking at cuts from multiple original constraints.

CHAPTER 1
INTRODUCTION

In this thesis, we explore different roles that sparsity plays in integer programming (Chapter 2,3 and 4) and the strength of multi-row cuts (Chapter 5). First, in Chapter 2, we are interested in approximating polytopes by sparse constraints under different settings. In Chapter 3, we are interested in improving the randomization step in feasibility pump, a primal heuristic. Our modification of the algorithm allows us to automatically detect decomposable problem and harness this sparse structure to solve problems more efficiently. In Chapter 4, we study the role that sparsity plays in the integrality of the extreme points of a randomly generated family of polytopes. Computational experiments suggest a correlation (for a family of randomly generated polytopes) between a sparse constraint matrix and a higher ratio of integral extreme points. Finally, in Chapter 5 we address another interesting problem in integer programming. We study the strength of aggregation cuts in the context of sign-pattern IPs. In particular, we study the benefits of considering multi-row aggregation cuts (a generalization of aggregation cuts) for these problems.

In this chapter we present the motivation for studying sparsity in integer programming and a brief introduction to each one of the chapters in the thesis.

1.1 Motivation

Understanding how to make use of sparsity has been a main driver in the development of multiple areas of scientific computing and optimization. The following examples correspond to an inexhaustive list of such efforts. Algorithms for solving systems of linear equations with sparse left-hand-side matrices define a field on their own (see for example [24, Chapter 1], or the books [30, 89]). Algorithms for solving linear programs (LP) with sparse constraint matrices [24, Chap-

ter 3], [60, 55], algorithms for quadratic programs (QP) with sparse data [92, 25], polynomial optimization with sparse data [95, 69, 96, 66, 64, 70, 27], solving semi-definite programs (SDP) with sparse data matrices [46, 48, 81] etc.

Surprisingly, the use of sparsity of input data is a very under explored direction of research in the context of Integer Programming (IP). Standard integer programming textbook do not, in general, address IPs with sparse data matrices except in the context of reformulation and decomposition methods to deal with specialized sparsity structure of the constraint matrix. On the other hand, standard techniques such as primal heuristics or cutting-plane selection, which work very well in practice - in fact may be working well due to sparsity of the constraint matrix - however, these consequences of sparsity of constraint matrices for IPs are either not well understood or not known at all.

1.1.1 Previous results on sparse IPs

We examine some of the current mixed-integer linear programming (MILP) techniques that explicitly take advantage of sparsity of constraint matrices. Many of them can be understood as techniques developed for linear programs that have been extended to MILPs.

Reformulation techniques including Dantzig-Wolfe [29] and Benders [11, 52, 53] that use Minkowski-Weyl representation theorem for polyhedra in order to solve linear programs with special structures, such as block diagonal constraint matrix, efficiently. In particular, some of the most significant works in this area correspond to column generation methods to solve cutting-stock problem [56], the branch-and-price algorithm [9], and the integer L-shaped method [68]. Additional work has been done in this direction; see [93] for a review. Recent work includes new decomposition methods that can take advantage of parallelization [4], incorporating Gomorys cuts in the context of a decomposition algorithm [50], and improvements of the integer L-shaped method proposed in [5]. A generalization of this direction of research involves first automatic permutation of rows and columns of the matrix so as to make the constraint matrix look like a bordered

(or doubly-bordered) block diagonal matrix and then apply the branch-and-price algorithm [22, 13, 98, 1].

As mentioned earlier, sparse LPs can be solved efficiently. A MILP solver typically solves a large number of LPs in a *branch-and-bound tree* to solve one MILP instance. Therefore, clearly one of the biggest benefits of sparsity for state-of-the-art MILP solvers is the sparsity of the underlying LP. The following computational result illustrates the importance of understanding sparsity of the linear relaxation of an IP even further. In an interesting study [97], the authors conducted an experiment by adding a very dense valid equality constraint to other constraints in the LP relaxation at each node while solving IP instances from MIPLIB while using CPLEX. The underlying polyhedron at each node is not changed but it makes the constraints dense. Their observation was the following: just by making 9 constraints artificially dense, a 25% increase in time to solve the instances was reported.

There also exist some complexity results and approximation algorithms related to sparse MILPs. Complexity results for sparse binary MILPs are, to the best of our knowledge, as the following: Depending on the non-zero entries of the constraint matrix, a so-called intersection graph [49] is constructed. Then, using non-serial dynamic programming [15] one can construct algorithms or obtain extended formulations whose complexity is proportional to the exponent of the tree-width [88, 87] of the intersection graph [71, 70, 94, 16]. Multiple randomized LP-rounding based approximation algorithms for sparse packing and covering problems have also been proposed [8, 85].

1.1.2 Areas of interest

Cutting-planes have become one of the main pillars in the structure of state-of-the-art mixed integer linear programming solvers. In particular, designing new families of cutting-planes for general MILPs (see the review papers [77, 86]) correspond to important theoretical advances. Even more, some of these cutting-plane families have brought significant speedups in state-of-the-art MILP

solvers [17, 74]. Therefore, understanding the role of sparsity in this context is important.

In particular, it is important to understand the relation between the sparsity structure of the constraint matrix and the resulting sparsity of valid cuts. In MIPLIB instances, one can possibly rearrange the rows and columns [22, 13, 98, 1] so that one sees patterns of blocks of variables in the constraint matrices. If we have a sparse problem, intuitively, we would prefer to generate valid cuts that somehow preserve this sparse structure.

In the presence of a sparsity pattern (for example, coming from a block structure) of the original constraint matrix, cuts, like the ones generated in the classical paper [26] satisfied, once again, this sparsity pattern. In fact many families of single constraint based cuts also satisfy this characteristic [82, 76, 100, 6, 7]. Note that this is not necessarily the case for general families of cuts.

In [40], Dey, Molinaro and Wang explore the problem of approximating the integer hull (convex hull of integer solutions) by only using sparse valid inequalities. This corresponds to an important research direction since, in practice, sparsity is taken into account whenever MIPs solvers are used. Sparsity of a cut (the number of variables involved in it) is one of the criteria these solvers use when deciding whether to add the cut or not to the formulation (of course there are additional considerations). Obviously, if solvers were to consider only sparse valid inequalities (in the setting of a cutting-plane method), this would pose serious limitations in terms of the problems they could solve, thus, it is important to understand the limitations of sparse cuts.

Another important area in Integer Programming corresponds to primal heuristics. Primal heuristics are used to find feasible integral solutions fast and, thus, play a significant role in any method for solving IPs, in particular branch and cut type approaches. Making use of sparsity should also be a concern when trying to develop new primal heuristics. For example, back-tracking of fixings may also be done in order to achieve feasibility when dealing with a sparse problem. Such methods are typically quite successful in the presence of global sparsity structure [28, 54, 12, 51, 75].

1.2 Sparse Approximations

In this context, sparsity refers to the number of variables that are present in a constraint (or equivalently to the number of non-zero coefficients present in that constraint). We say that a constraint is sparse if it involves a small number of variables. Similarly, we say that a constraint is dense if it involves a large number of variables. If n denotes the number of variables, then for any positive integer $k \leq n$, we say that a constraint is k-sparse if it involves exactly k variables (or k non-zero coefficients). Sometimes we refer to this as the sparsity level of a particular constraint, a group of constraints or just the constraint matrix.

In Chapter 2, we are interested in understanding how well we can approximate polytopes by using sparse inequalities under four different settings. In [40], the authors defined the sparse closure of a polytope P as the object obtained by taking the intersection of all sparse valid inequalities for P. They presented bounds on the quality of this approximation in terms of a worst-case-direction metric measuring the distance between P and its sparse approximation.

Since in practice sparsity is one of the criteria considered when solvers decide whether to add a cut or not, the paper [40] was interested in understanding how well would the sparse approximation behave in a general setting. It was shown than whenever the number of vertices of a polytope is polynomial in the size of the input data, its sparse closure approximates well the integer hull. However when the number of vertices increases the sparse approximation no longer performs well, in fact, with high probability the approximation is poor for random 0/1 polytopes with a super polynomial number of vertices.

In our work, we are interested in the following questions: First, when we want to approximate the convex hull of a set of integer points (defined by a set of linear constraints as well as integrality constraints) by considering its linear relaxation and, additionally, only sparse valid inequalities. Second, when we only allow for a budgeted number of dense constraint to strengthen the sparse approximation. Third, the approximation of a polytope under every rotation. Finally, how does the

sparse approximation behave along a randomly selected direction, for optimization purposes.

The motivation for these questions is as follows. First, since in practice, when trying to solve an IP, its linear relaxation may contain dense inequalities, one would like to take this into consideration. The question that arises here is: Are there integer programs for which the sparse approximation intersected with the linear relaxation, does not approximate the integer hull well? Second, if we want to improve the approximation of a polytope by adding a budgeted number of dense inequalities in addition to the sparse ones: Are there polytopes where the quality of approximation by sparse inequalities cannot be significantly improved by adding a polynomial (or even exponential) number of dense valid inequalities? Third, since it is well understood that sparse approximation of polytopes are not invariant under affine transformations (in particular rotations): Are there polytopes that are difficult to approximate under *every* rotation? Finally, since in the context of solving an integer program, one is concerned with approximating the feasible region in the direction of the objective function: Are there polytopes that are difficult to approximate, using sparse inequalities, in *almost all* directions?

We provide positive answers to each one of the previously described questions. This corresponds to an indication that sparse inequalities do not always approximate integer hulls well under the settings considered in this work. However, understanding when sparse inequalities are effective in the afore mentioned settings corresponds to an important research direction given its practical applications.

1.3 Randomization step in feasibility pump

In Chapter 3, we are interested in primal heuristics that can harness the sparsity and structure present in many problem and use it to find feasible solutions fast (both in terms of number of iterations as well as computation time). In particular, we are interested in modifying Feasibility Pump (FP) a successful primal heuristic for mixed-integer linear programs in order to capture this effect. FP was first introduced in [43] and many variants and improvements for it have been studied

[3, 14, 21, 45, 31, 42, 20, 41, 19].

The core steps in FP are rounding and projecting. In the original version, the algorithm starts with the solution to the linear relaxation. If this solution satisfies the integrality constraints, a feasible integral solution has been found. If the solution is fractional, then it is rounded to its closest integral vector so that is satisfies the integrality constraints. Then, this rounded solution is projected onto the linear relaxation. This process is repeated until a feasible integral solution has been found.

Additionally, in the original FP, a randomization step is used to handle stalling solutions (those that are visited multiple times) by randomly perturbing the current solution. In our work, we propose to modify this randomization step so that it takes advantage of the decomposable structures present in many instances. Since decomposable or semi-decomposable instances are naturally sparse, in our modification, FP can *automatically* detect the sparse structure in the constraint matrix and use it advantageously in the randomization step.

The idea comes from a randomized algorithm for solving SAT instances called WalkSAT, first proposed in [90]. WalkSAT finds feasible solutions for instances in conjunctive normal form (CNF), starting from a random assignment, check if is feasible, and if it is not, it flips one literal out of one of the infeasible clauses. If no solution is found after $O(n)$ iterations, it restarts from a new random assignment. When the number of literals present in each clause is bounded, it was shown in [90] that WalkSAT runs, in expectation, faster than enumerating and checking all possible assignments. The upper bound obtained for the running time corresponds to an increasing function in the number of literals in the clauses. This idea can be directly applied to the case of binary programs, where small number of literals in a clause corresponds to a sparse constraint.

WalkSAT has the following property. If we apply it to a block separable problem, at a given point in time, one of the blocks is feasible, then this block will not become infeasible later on. This comes from the fact that those variables that are flipped come from the infeasible constraints of the problem.

In our work, we generalize WalkSAT for mixed-binary programs and show that it continues to have the theoretical guarantees of the original method. Then we combine a randomization step based on this mixed-binary version of WalkSAT with the rounding and projecting steps of FP. We are also able to provide a theoretical analysis of our WalkSAT-based FP in the case of decomposable subset-sum-type instances (to the best of our knowledge no such analysis exists to date).

Based on the theoretical insights, we implemented a modified version of FP with changes only to the randomization step. In order to test our modified randomization step, we used a state-of-the-art version of FP (Feasibility Pump 2.0) and incorporated the WalkSAT ideas on top of the previously discussed randomization step. This allowed for our method to say close to the original FP method in many cases and transition to the WalkSAT version when the original randomization step does not work efficiently.

We compare our method to the original FP 2.0 both for a series of randomly generated 2-stage stochastic programs (that naturally present an almost decomposable structure) as well as a series of problems from MIPLIB 2010 (see [65]). In both case, on average, our version outperform the original FP 2.0. Since our modified randomization step does not affect other components of the algorithm, it can be incorporated to many versions of FP.

1.4 Bounds on the number of extreme points for random polytopes

In Chapter 4, we are interested in understanding the role that sparsity plays in terms of the integral extreme points of a polytope. We want to understand what happens with the ratio between the number of integral extreme points to the total number of extreme points for different sparsity levels.

Some preliminary experiments showed that for a family of randomly generated polytopes this ratio decreases as problems becomes denser.

We propose a family of randomly generated packing instances in order to perform our analysis

and obtain the necessary bounds.

In order to construct a bound on this ratio, we obtain lower bounds for the expected number of integral points as well as upper bound for the total number of extreme points for the family of randomly generated packing instances.

Using the previous results, we show that for the family of randomly generated packing instances, we can obtain a lower bound for the expected value of this ratio that decreases as instances become denser.

1.5 Multi-row cuts and Sign-Pattern IPs

In [18], Bodur et al. studied the aggregation closure of a polyhedron (for definitions, see Chapter 5), an extension of the CG closure and compared it against the following object: the intersection of the aggregation closures of each one of the original constraints (i.e. obtained in a disaggregated manner). We refer to this object as the original 1-row closure of a polyhedron. In the case packing/covering type problems (for details, see Chapter 5) the aggregation closure can be 2 approximated by the original 1-row closure.

Generating cutting-planes from multiple constraints has been a significant research direction recently. Well known approaches, like the infinite group approach can be used in this context. Relaxing the bound on the basic variables together with introducing an infinite number of new non-negative non-basic integer and continuous variables corresponding to distinct columns applied to m rows of the simplex tableaux lead to the *mixed-integer infinite group relaxations* (MIIGR) [57, 58, 59, 62]. In [33, 34, 32, 35, 36] some of the first known families of extreme inequalities of multiple row ($m \geq 2$) infinite group relaxations were presented.

In the case of aggregation cuts, it is possible to extend the notions of aggregation closure and original 1-row closure to the multi-row case. For the aggregation closure, we can consider the object obtained by the convex hull of multiple (namely k) aggregation constraints, which we call as the $k-$aggregation closure. For the original 1-row closure, we can consider the object attained

by consider the closure of all subset of size k of the original constraints, which we call as the original k-row closure.

The fact that in the case packing/covering type problems the aggregation closure can be well approximated by the original 1-row closure is not true for general IPs. Computational experiments show that for general problems the original 1-row closure can be an arbitrarily bad approximation of the aggregation closure. But, again in the case of these problems, the same is true when approximating the 2-aggregation closure by the original 2-row closure.

Note that an important property that packing/covering type problems share is that all the constraints are of the same type. Even more, when considering aggregation cuts, these also correspond to packing/covering type constraints. Thus if a polyhedron is a packing/covering polyhedron, so is its closure and consequently its integer hull.

In Chapter 5, we study a family of problems that we refer to as sign-pattern IPs. Each one of these problems satisfies the property that for each variable, the sign of its coefficients in each constraint and in the objective is the same (for detail see Chapter 5). Thus, as in the case of a packing/covering problem, its closure and its integer hull also correspond to sign-pattern IPs.

As in the previous case, for these problems, it is true that the aggregation closure can be 2 approximated by the original 1-row closure. However, it is no longer the case that the multi-row aggregation closure can be well approximated by the aggregation closure, nor that the 2-aggregation closure can be well approximated by the original 2-row closure.

These results show some further insight into the role that aggregation plays in generating useful cutting planes.

CHAPTER 2

SOME LOWER BOUNDS ON SPARSE OUTER APPROXIMATIONS OF POLYTOPES

The work presented in this chapter has been accepted for publication at Operations Research Letters through a paper authored by Santanu Dey, Andres Iroume and Marco Molinaro [37].

2.1 Introduction

The paper [40] studied how well one can expect to approximate polytopes using valid inequalities that are sparse. The motivation for this study came from the usage of cutting-planes in integer programming (IP) solvers. In principle, facet-defining inequalities of the integer hull of a polytope can be dense, i.e. they can have non-zero coefficients for a high number of variables. In practice, however, most state-of-the-art IP solvers bias their cutting-plane selection towards the use of sparse inequalities. This is done, in part, to take advantage of the fact that linear programming solvers can harness sparsity well to obtain significant speedups.

The paper [40] shows that for polytopes with a polynomial number of vertices, sparse inequalities produce very good approximations of polytopes. However, when the number of vertices increase, the sparse inequalities do not provide a good approximation in general; in fact with high probability the quality of approximation is poor for random 0-1 polytopes with super polynomial number of vertices (see details in [40]).

However the study in [40] is very "idealized" in the context of cutting-planes for IPs, since almost always some dense cutting-planes are used or one is interested in approximating the integer only only along certain directions. In this chapter, we consider some natural extensions to understand the properties of sparse inequalities under more "realistic conditions":

1. All the results in the paper [40] deal with the case when we are attempting to approximate

the integer hull using only sparse inequalities. However, in practice the LP relaxation may have dense inequalities. Therefore we examine the following question: Are there integer programs, such that sparse inequalities do not approximate the integer hull well when added to a linear programming relaxation?

2. More generally, we may consider attempting to improve the approximation of a polytope by adding a few dense inequalities together with sparse inequalities. Therefore we examine the following question: Are there polytopes, where the quality of approximation by sparse inequalities cannot be significantly improved by adding polynomial (or even exponential) number of *arbitrary* valid inequalities?

3. It is clear that the approximations of polytopes using sparse inequalities is not invariant under affine transformations (in particular rotations). This leaves open the possibility that a clever reformulation of the polytope of interest may vastly improve the approximation obtained by sparse cuts. Therefore a basic question in this direction: Are there polytopes that are difficult to approximate under *every* rotation?

4. In optimization one is usually concerned with the feasible region in the direction of the objective function. Therefore we examine the following question: Are there polytopes that are difficult to approximate in *almost all* directions using sparse inequalities?

We are able to present examples that answer each of the above questions in the positive. This is perhaps not surprising: an indication that sparse inequalities do not always approximate integer hulls well even in the more realistic settings considered in this chapter. Understanding when sparse inequalities are effective in all the above settings is an important research direction.

The rest of the chapter is organized as follows. Section 2.2 collects all required preliminary definitions. In Section 2.3 we formally present all the results. In Sections 2.4-2.7 we present proofs of the various results.

2.2 Preliminaries

2.2.1 Definitions

For a natural number n, let $[n]$ denote the set $\{1,\ldots,n\}$ and, for non-negative integer $k \leq n$ let $\binom{[n]}{k}$ denote the set of all subsets of $[n]$ with k elements. For any $x \in \mathbb{R}^n$, let $||x||_1$ denote the l_1 norm of x and $||x||$ or $||x||_2$ denote the l_2 norm of x.

An inequality $\alpha x \leq \beta$ is called *k-sparse* if α has at most k non-zero components. Given a polytope $P \subset \mathbb{R}^n$, P^k is defined as the intersection of all k-sparse cuts valid for P (as in [40]), that is, the best outer-approximation obtained from k-sparse inequalities. We remark that P^k is also a polytope (see [40]).

Given two polytopes $P, Q \subset \mathbb{R}^n$ such that $P \subseteq Q$ we consider the Hausdorff distance $d(P,Q)$ between them:

$$d(P,Q) := \max_{x\in Q}\left(\min_{y\in P}||x-y||\right).$$

When $P, Q \subset [-1,1]^n$, we have that $d(P,Q)$ is upper bounded by $2\sqrt{n}$, the largest distance between two points in $[-1,1]^n$. In this case, if $d(P,Q) \propto \sqrt{n}$ the error of approximation of P by Q is basically as large as it can be and smaller $d(P,Q)$ (for example constant or of the order of $\sqrt{\log n}$) will indicate better approximations.

Given a polytope $P \subseteq \mathbb{R}^n$ and a vector $c \in \mathbb{R}^n$, we define

$$gap_P^k(c) = \max_{x\in P^k} cx - \max_{x\in P} cx,$$

namely the "gap" between P^k and P in direction c. We first note that $d(P,P^k)$ equals the worst directional gap between P^k and P (the proof is presented in Appendix A.1).

Lemma 1. For every polytope $P \subseteq \mathbb{R}^n$, $d(P,P^k) = \max_{c:||c||=1} gap_P^k(c)$.

For a set $\mathscr{D} = \{\alpha_1 x \leq \beta_1, \ldots, \alpha_d x \leq \beta_d\}$ of (possibly dense) valid inequalities for P, let $P^{k,\mathscr{D}}$

denote the outer-approximation obtained by adding all k-sparse cuts and the inequalities from $\mathscr{D}$:

$$P^{k,\mathscr{D}} = \left(\bigcap_{i=1}^{d} \{x \in \mathbb{R}^n : a_i x \leq b_i\} \right) \bigcap P^k. \tag{2.1}$$

Since $P^{k,\mathscr{D}} \subseteq P^k$ we have that $d(P, P^{k,\mathscr{D}}) \leq d(P, P^k)$ for any set $\mathscr{D}$ of valid inequalities for P.

2.2.2 Important Polytopes

Throughout the chapter, we will focus our attention on the polytopes $\mathscr{P}_{t,n} \subseteq [0,1]^n$ defined as

$$\mathscr{P}_{t,n} = \left\{ x \in [0,1]^n : \sum_{i=1}^{n} x_i \leq t \right\}. \tag{2.2}$$

Notice that for $t = 1$ we obtain a simplex and for $t = n/2$ we obtain half of the hypercube. Moreover different values to t yield very different properties regarding approximability using sparse inequalities, as discussed in [40].

Proposition 2. The following hold:

1. $d(\mathscr{P}_{1,n}, \mathscr{P}^k_{1,n}) = \frac{\sqrt{n}}{k} - \frac{1}{\sqrt{n}}$.
2. $d(\mathscr{P}_{n/2,n}, \mathscr{P}^k_{n/2,n}) = \begin{cases} \sqrt{n}/2 & \text{if } k \leq n/2 \\ \frac{n\sqrt{n}}{2k} - \frac{\sqrt{n}}{2} & \text{if } k > n/2 \end{cases}$.
3. $\mathscr{P}^k_{t,n} = [0,1]^n$ for all $t \leq n$ and $k \leq t$. .

We will also consider symmetrized versions of the polytopes $\mathscr{P}_{t,n}$. To define this symmetrization, for $x \in \mathbb{R}^n$ and $I \subset [n]$ let x^I denote the vector obtained by switching the sign of the components of x not in I:

$$x^I_i = \begin{cases} x_i & \text{if } i \in I \\ -x_i & \text{if } i \notin I. \end{cases}$$

More generally, for a set $P \subseteq \mathbb{R}^n$ we define $P^I = \left\{x^I \in \mathbb{R}^n : x \in P\right\}$.

Definition 3. For a polytope $P \subseteq \mathbb{R}^n_+$, we define its symmetrized version $\overline{P} = \text{conv}\left(\bigcup_{I\subseteq[n]} P^I\right)$.

Note that $\overline{\mathscr{P}_{1,n}}$ is the cross polytope in dimension n; more generally, we have the following external description of the symmetrized versions of $\mathscr{P}_{t,n}$ and $\mathscr{P}^k_{t,n}$ (proof presented in Appendix A.2).

Lemma 4.

$$\overline{\mathscr{P}_{t,n}} = \left\{x \in [-1,1]^n : \ \forall I \subset [n], \sum_{i\in I} x_i - \sum_{i\in[n]\setminus I} x_i \leq t\right\} \tag{2.3}$$

$$\overline{\mathscr{P}_{t,n}}^k = \left\{x \in [-1,1]^n : \ \forall I \in \binom{[n]}{k}, \ \forall I^+, I^- \text{ partition of } I, \sum_{i\in I^+} x_i - \sum_{i\in I^-} x_i \leq t\right\}. \tag{2.4}$$

2.3 Main results

In our first result (Section 2.4), we point out that in the worst case LP relaxations plus sparse inequalities provide a very weak approximation of the integer hull.

Theorem 5. For every even integer n there is a polytope $Q_n \subseteq [0,1]^n$ such that:

1. $\mathscr{P}_{n/2,n} = \text{conv}(Q_n \cap \mathbb{Z}^n)$
2. $d(\mathscr{P}_{n/2,n}, (\mathscr{P}_{n/2,n})^k \cap Q_n) = \Omega(\sqrt{n})$ for all $k \leq n/2$.

In Section 2.5 we consider the second question: How well does the approximation improve if we allowed a budgeted number of dense valid inequalities. Notice that for the polytope $\mathscr{P}_{\frac{n}{2},n}$, while Proposition 2 gives that $d(\mathscr{P}_{\frac{n}{2},n}, \mathscr{P}^k_{\frac{n}{2},n}) \geq \Omega(\sqrt{n})$, adding exactly *one* dense cut ($ex \leq n/2$) to the k-sparse closure (even for $k = 1$) would yield the original polytope $\mathscr{P}_{\frac{n}{2},n}$.

We consider instead the symmetrized polytope $\overline{\mathscr{P}_{\frac{n}{2},n}}$. Notice that while this polytope needs 2^n dense inequality to be described *exactly*, it could be that a small number of dense inequalities, together with sparse cuts, is already enough to provide a good approximation; we observe that in higher dimensions valid cuts for $\overline{\mathscr{P}_{\frac{n}{2},n}}$ can actually cut off significant portions of $[-1,1]^n$ in *multiple orthants*. Nevertheless, we show that *exponentially* many dense inequalities are required to improve the approximation significantly.

Theorem 6. Consider an even integer n and the polytope $P = \overline{\mathscr{P}_{\frac{n}{2},n}}$. For any $k \leq n/100$ and any set $\mathscr{D}$ of valid inequalities for P with $|\mathscr{D}| \leq \exp\left(\frac{n}{600^2}\right)$, we have

$$d\left(P, P^{k,\mathscr{D}}\right) \geq \frac{1}{6}\sqrt{n}.$$

In the proof of this theorem we use a probabilistic approach to count in how many orthants an inequality can significantly cut off the box $[-1,1]^n$.

In Section 2.6 we consider the question of sparse approximation of a polytope when rotations are allowed. We show that again $\overline{\mathscr{P}_{n/2,n}}$ cannot be approximated using sparse inequalities after *any* rotation is applied to it.

Theorem 7. Consider an even integer n and the polytope $P = \overline{\mathscr{P}_{\frac{n}{2},n}}$. For every rotation $R : \mathbb{R}^n \to \mathbb{R}^n$ and $k \leq \frac{n}{200^3}$, we have

$$d\big(R(P), (R(P))^k\big) = \Omega(\sqrt{n}).$$

The proof of this theorem relies on the intuition given by Theorem 6: since $\overline{\mathscr{P}_{\frac{n}{2},n}}$ required exponentially many dense inequalities in order to be well approximated, no rotation is able to align all of them with the axis so that they can be captured by sparse inequalities.

Finally, in Section 2.7 we show that $\overline{\mathscr{P}_{\frac{n}{10},n}}$ and its k-sparse approximation have a large gap in almost every direction.

Theorem 8. Let $n \geq 1000$ be an integer divisible by 10 and consider the polytope $P = \overline{\mathscr{P}_{n/10,n}}$. If $C \in \mathbb{R}^n$ is a random direction uniformly distributed on the unit sphere, then for $k \leq \frac{n}{10}$ we have

$$\mathbb{P}\left(gap_P^k(C) \geq \frac{\sqrt{n}}{20}\right) \geq 1 - \frac{4}{n}.$$

To prove this theorem we rely on the concentration of the value of Lipschitz functions on the sphere (actually we work on the simpler Gaussian space).

2.4 Strengthening of *LP* relaxation by sparse inequalities

We now present a short proof of Theorem 5. Consider the polytope

$$Q_n = \left\{ x \in [0,1]^n : \sum_{i \in I} x_i \leq \frac{n}{2} \quad \forall I \in \binom{[n]}{\frac{n}{2}+1} \right\}.$$

It is straightforward to verify that $\mathscr{P}_{n/2,n} = \mathrm{conv}(Q_n \cap \mathbb{Z}^n)$.

From Part (3) of Proposition 2, $\mathscr{P}^k_{n/2,n} = [0,1]^n$ thus $Q_n \cap \mathscr{P}^k_{n/2,n} = Q_n$. Now $x = \frac{n}{n+2}e$ belongs to Q_n and its projection onto $\mathscr{P}_{n/2,n}$ corresponds to $y = \frac{1}{2}e$. Therefore,

$$d\left(\mathscr{P}_{n/2,n}, \mathscr{P}^k_{n/2,n} \cap Q_n\right) \geq \frac{n-2}{2n+4}\sqrt{n} = \Omega(\sqrt{n}).$$

This concludes the proof of the theorem.

2.5 Strengthening by general dense cuts

Now we turn to the proof of Theorem 6. For that we will need Bernstein's concentration inequality (stated in a slightly weaker but more convenient form).

Theorem 9 ([67], Appendix A.2)**.** Let $X_1, X_2, \ldots, X_n$ be independent random variables such that

$\mathbb{E}[X_i] = 0$ and $|X_i| \leq M \ \forall i$. Let $X = \sum_{i=1}^{n} X_i$ and $\sigma^2 = \mathrm{Var}(X) \leq U$. Then:

$$\mathbb{P}(|X| > w) \leq \exp\left(-\min\left\{\frac{w^2}{4U}, \frac{3w}{4M}\right\}\right).$$

Notice that to prove the theorem it suffices to consider the case $k = \frac{n}{100}$, which is what we do. Recall that $P = \overline{\mathscr{P}_{n/2,n}}$, consider any set $\mathscr{D}$ of valid inequalities for P with $|\mathscr{D}| \leq \exp\left(\frac{n}{600^2}\right)$; for convenience let $d = |\mathscr{D}|$. From Lemma 4 we know P^k contains all the points in $\{-1,1\}^n$. Also note that any $\bar{x} \in \{-1,1\}^n$ achieves the maximal distance in P^k from P, namely $d(P,P^k) = d(P,\bar{x}) = \frac{1}{2}\sqrt{n}$. We then consider a random such "bad" point X, namely X is uniformly distributed in $\{-1,1\}^n$ (equivalently, the X_i's are independent and uniformly distributed over $\{-1,1\}$). We will show that there exists an instantiation of the scaled random $\frac{2X}{3}$ which belongs to $P^{k,\mathscr{D}}$, which will then lower bound the distance $d(P,P^{k,\mathscr{D}})$ by $d(P,\frac{2\bar{x}}{3}) = \frac{1}{6}\sqrt{n}$ (for some $\bar{x} \in \{-1,1\}^n$) and thus prove the result.

To achieve this, consider a single inequality $ax \leq b$ from $\mathscr{D}$ (we assume without loss of generality that $\|a\|_1 = 1$). We claim that with probability more than $1 - \frac{1}{d}$, the point $\frac{2X}{3}$ satisfies this inequality. By symmetry of X, we can assume without loss of generality that $a \geq 0$. To prove this, let $\bar{a}$ be the vector obtained by keeping the k largest components of a and zeroing out the other components (ties are broken arbitrarily), and let $\underline{a} = a - \bar{a}$. Since $\bar{a}x \leq b$ is a k-sparse valid inequality for P and $X \in P^k$, we have that

$$aX = \bar{a}X + \underline{a}X \leq b + \underline{a}X. \tag{2.5}$$

Claim 10. $\mathrm{Var}(\underline{a}X) \leq \frac{b(n-k)}{k^2}$.

Proof. Since $\mathrm{Var}(X_i) = 1$ for all $i \in [n]$, we obtain that

$$\mathrm{Var}(\underline{a}X) \quad = \sum_{i=1}^{n} \underline{a}_i^2 \,\mathrm{Var}(X_i) = ||\underline{a}||^2. \tag{2.6}$$

Note that the kth largest component of a is at most $1/k$ (otherwise $\|a\|_1 > 1$), hence $\underline{a}_i X_i \leq \frac{1}{k}$ for all i, so we have

$$||\underline{a}||^2 = \sum_{i=1}^{n} (\underline{a}_i X_i)^2 \leq \frac{1}{k} \sum_{i=1}^{n} \underline{a}_i X_i. \tag{2.7}$$

Moreover, by comparing averages of the components of $\bar{a}$ and $\underline{a}$ and then using $\bar{a}e \leq b$, we have that

$$\sum_{i=1}^{n} \frac{\underline{a}_i}{n-k} \leq \sum_{i=1}^{n} \frac{\bar{a}_i}{k} \leq \frac{b}{k}. \tag{2.8}$$

Now by using (2.6)-(2.8), we obtain the bound $\mathrm{Var}(\underline{a}X) \leq \frac{b(n-k)}{k^2}$, thus concluding the proof. $\diamond$

Now using the fact that $|\underline{a}_i X_i| \leq \frac{1}{k}$, $\mathbb{E}(\underline{a}X) = 0$ and the above bound on $\mathrm{Var}(\underline{a}X)$, we obtain by an application of Bernstein's inequality (Theorem 9) with $w = 30b\frac{\sqrt{\log d}}{\sqrt{k}}$:

$$\mathbb{P}\left(\underline{a}X \geq 30b \cdot \frac{\sqrt{\log d}}{\sqrt{k}}\right) \leq \exp\left(-\min\left\{\frac{30^2 b \cdot k \cdot \log d}{4(n-k)}, \frac{30}{4} \cdot 3b \cdot \sqrt{k \log d}\right\}\right). \tag{2.9}$$

To upper bound the right-hand side of this expression, first we employ our assumption $d \leq \exp(\frac{n}{600^2})$ and $k = \frac{n}{100}$ to obtain

$$\sqrt{\log d} \leq \frac{1}{600}\sqrt{n} \leq \frac{3 \cdot 99}{30 \cdot 10}\sqrt{n} = \frac{3}{30}\left(\frac{n-k}{k}\right)\sqrt{k}.$$

With this at hand, we have that the minimum in the right-hand side of (2.9) is achieved in the first

term. Moreover, notice that $b \geq 1/2$: the point $p = (\frac{1}{2}, \ldots, \frac{1}{2})$ belongs to P and hence $b \geq ap = \frac{1}{2}\|a\|_1 = 1/2$. Putting these observations together gives

$$\mathbb{P}\left(\underline{a}X \geq 30b \cdot \frac{\sqrt{\log d}}{\sqrt{k}}\right) \leq \exp\left(-\frac{30^2}{4 \cdot 99} b \cdot \log d\right) < \exp(-\log d) = \frac{1}{d}.$$

Then using (2.5) and the above inequality, we obtain that with probability more than $1 - \frac{1}{d}$ we have

$$\begin{aligned} aX &\leq b\left(1 + 30\frac{\sqrt{\log d}}{\sqrt{k}}\right) \\ &= b\left(1 + \frac{1}{2} \cdot 600\frac{\sqrt{\log d}}{\sqrt{n}}\right) \leq b\frac{3}{2}, \end{aligned} \tag{2.10}$$

where the first equality uses $k = \frac{n}{100}$ and the second inequality uses the assumption that $\sqrt{\log d} \leq \frac{1}{600}\sqrt{n}$. Now note that (2.10) implies that the point $\frac{2X}{3}$ satisfies $ax \leq b$ with probability more than $1 - \frac{1}{d}$.

Since $|\mathscr{D}| = d$, we can then take a union bound over the above argument to get that with strictly positive probability $\frac{2X}{3}$ satisfies *all* the inequalities in $\mathscr{D}$. Hence with strictly positive probability $\frac{2X}{3}$ belongs to $P^{k,\mathscr{D}}$ and in particular there is a point $\bar{x} \in \{-1,1\}^n$ such that $\frac{2\bar{x}}{3} \in P^{k,\mathscr{D}}$.

This gives the lower bound $d(P, P^{k,\mathscr{D}}) \geq d\left(P, \frac{2\bar{x}}{3}\right)$; now we lower bound the right-hand side. It is easy to see that the closest point in $\overline{P}$ to $2\bar{x}/3$ is $\bar{x}/2$, the projection onto $\overline{P}$. Since $||2\bar{x}/3 - \bar{x}/2|| = \frac{1}{6}||\bar{x}||$, we obtain that $d(\overline{P}, \bar{x}) \geq \frac{1}{6}\sqrt{n}$ which concludes the proof.

2.6 Sparse approximation of rotations of a polytope

In this section we prove Theorem 7; for that we need to recall some standard definitions from convex geometry.

Definition 11. Given a set $P \subseteq \mathbb{R}^n$:

- We say that P is *centrally symmetric* if $\forall x \in P : -x \in P$.

- For any $\alpha \in \mathbb{R}$ we define the set $\alpha P := \{\alpha x : x \in P\}$.

- The *polar* of P is the set $P^\circ = \{z \in \mathbb{R}^n : zx \leq 1 \ \forall x \in P\}$.

We also need the following classical result about approximating convex set by polytopes with few vertices (see for instance Lemma 4.10 of [10] and [84])

Theorem 12. For every centrally symmetric convex set $S \subseteq \mathbb{R}^k$, there is a polytope S' with at most $\left(\frac{3}{\varepsilon}\right)^k$ vertices such that $S \subseteq S' \subseteq (1+\varepsilon)S$

By applying this result to the polar we obtain approximations with bounded number of *facets* instead of vertices.

Lemma 13. For every centrally symmetric convex set $C \subseteq \mathbb{R}^k$, there is a polytope C' with at most $\left(\frac{3}{\varepsilon}\right)^k$ facets such that $C \subseteq C' \subseteq (1+\varepsilon)C$.

Proof. Consider the (centrally symmetric) convex set $\frac{1}{1+\varepsilon}C^\circ$; applying the above result, we get S with $(3/\varepsilon)^k$ vertices and $\frac{1}{1+\varepsilon}C^\circ \subseteq S \subseteq C^\circ$. Taking polars (and noticing that $(\lambda A)^\circ = (1/\lambda)A^\circ$), we get $C \subseteq S^\circ \subseteq (1+\varepsilon)C$ and S° has at most $(3/\varepsilon)^k$ facets. This concludes the proof. □

The key ideas used in our proof of Theorem 6 are twofold (recall that $P = \overline{\mathscr{P}_{n/2,n}}$):

1. Roughly speaking, $(RP)^k$ is the intersection of polyhedra each of which can be decomposed into a *k-dimensional* polytope plus an $(n-k)-$dimensional lineality space. This allow us to use Lemma 13 above to get a good approximation H of $(RP)^k$ using fewer than $\exp\left(\frac{n}{600^2}\right)$ inequalities.

2. Then argue that $d(RP,(RP)^k) \approx d(RP,H) = \Omega(\sqrt{n})$ since $d(P,R^{-1}(H)) = \Omega(\sqrt{n})$ due to the number of facets of H and Theorem 6.

Proof of Theorem 7. Note that it is sufficient to prove the result for $k = \frac{n}{200^3}$, which is what we do. To make the above ideas precise, observe that $(RP)^k = \bigcap_{K \in \binom{[n]}{k}} Q_K$, where $Q_K = RP + 0^K \times \mathbb{R}^{\bar{K}}$

(we use $\bar{K} := [n] \setminus K$). To approximate each Q_K, using Lemma 13, let $h_K \subseteq \mathbb{R}^k$ be a polytope such that $\text{proj}_K Q_K \subseteq h_K \subseteq (1+\varepsilon)\text{proj}_K Q_K$ and h_K has at most $(3/\varepsilon)^k$ facets. Let $H_k = h_K + 0^K \times \mathbb{R}^{\bar{K}}$; then $Q_K \subseteq H_K \subseteq (1+\varepsilon)Q_K$ and H_K has at most $(3/\varepsilon)^k$ facets.

Now notice that for convex sets A, B, we have $((1+\varepsilon)A) \cap ((1+\varepsilon)B) \subseteq (1+\varepsilon)(A \cap B)$. This gives that if we look at the intersection $\bigcap_{K \in \binom{[n]}{k}} H_K$, we obtain

$$\begin{aligned}(RP)^k = \bigcap_{K \in \binom{[n]}{k}} Q_K &\subseteq \bigcap_{K \in \binom{[n]}{k}} H_K \subseteq (1+\varepsilon) \bigcap_{K \in \binom{[n]}{k}} Q_K \\ &= (1+\varepsilon)(RP)^k. \end{aligned} \tag{2.11}$$

Notice $\bigcap_{K \in \binom{[n]}{k}} H_K$ has at most $\binom{n}{k}\left(\frac{3}{\varepsilon}\right)^k \leq \left(\frac{en}{k}\right)^k \left(\frac{3}{\varepsilon}\right)^k = \left(\frac{3en}{k\varepsilon}\right)^k$ facets. Thus, setting $\varepsilon = \frac{1}{10}$ we get

$$\begin{aligned}\left(\frac{3en}{k\varepsilon}\right)^k &= \left(30 \cdot e \cdot 200^3\right)^{\frac{n}{200^3}} \\ &= \left(\exp(\log(30 \cdot e \cdot 200^3))\right)^{\frac{n}{200^3}} \\ &= \left(\exp\left(\log(30 \cdot e \cdot 200^3) \cdot \frac{n}{200^3}\right)\right) \\ &< \exp\left(\frac{n}{601^2}\right). \end{aligned} \tag{2.12}$$

Then define $H := \bigcap_{K \in \binom{[n]}{k}} H_K$, so that $(RP)^k \subseteq H \subseteq (1+\varepsilon)(RP)^k$.

In order to control the relationship between this multiplicative approximation and the distance $d(.,.)$, we introduce the set $C = R([-1,1]^n)$. Notice that by construction $RP \subseteq H \cap C$.

Claim 14. $d(RP, H \cap C) \geq \frac{1}{6}\sqrt{n}$

Proof. Assume by contradiction that $d(RP, H \cap C) < \frac{1}{6}\sqrt{n}$. Then since distances between points and number of facets of a polytope are invariant under rotation, we obtain that $d(P, R^{-1}(H \cap C)) < \frac{1}{6}\sqrt{n}$ where $R^{-1}(H \cap C)$ is defined using at most $\exp(n/(600)^2)$ inequalities (because C has $2n$ facets, using (2.12) H has at most $\exp(n/(601)^2)$ and for sufficiently large n, $\exp(n/(601)^2) + 2n \leq$

$\exp(n/(600)^2)$). However notice that this contradicts the result of Theorem 6, since $k = \frac{\sqrt{n}}{100} \leq \frac{n}{100}$ and $R^{-1}(H \cap C)$ is defined using at most $2^{n/(600)^2}$ inequalities. $\diamond$

But from (2.11) we have $(1+\varepsilon)(RP)^k \cap C$ contains $H \cap C$, and hence

$$d(RP, (1+\varepsilon)(RP)^k \cap C) \geq \frac{1}{6}\sqrt{n}. \tag{2.13}$$

Claim 15. $d(RP, (RP)^k \cap C) \geq d(RP, (1+\varepsilon)(RP)^k \cap C) - \varepsilon\sqrt{n}$

Proof. Take $\bar{x} \in (1+\varepsilon)(RP)^k \cap C$ and $\bar{y} \in RP$ that achieve $d(\bar{x}, \bar{y}) = d((1+\varepsilon)(RP)^k \cap C, RP)$. Look at the point $\frac{1}{1+\varepsilon}\bar{x}$ and notice it belongs to $(RP)^k \cap C$; let $\tilde{y}$ be the point in RP closest to $\frac{1}{1+\varepsilon}\bar{x}$. Then since $\bar{y}$ is the point in RP closest to $\bar{x}$,

$$d(RP, (1+\varepsilon)(RP)^k \cap C) = d(\bar{x}, \bar{y}) \leq d(\bar{x}, \tilde{y}).$$

By triangle inequality, $d(\bar{x}, \tilde{y}) \leq d(\frac{1}{1+\varepsilon}\bar{x}, \tilde{y}) + d(\frac{1}{1+\varepsilon}\bar{x}, \bar{x}) \leq d(RP, (RP)^k \cap C) + d(\frac{1}{1+\varepsilon}\bar{x}, \bar{x})$. To bound $d(\frac{1}{1+\varepsilon}\bar{x}, \bar{x})$, notice it is equal to $\frac{\varepsilon}{1+\varepsilon}\|\bar{x}\|$; since $\bar{x}$ belongs to C, we can upper bound $\|\bar{x}\| \leq \sqrt{n}$ (this is why we introduced the set C in the argument). Putting these bounds together we obtain the result. $\diamond$

Using (2.13) and Claim 2 we obtain that $d(RP, (RP)^k) \geq d(RP, (RP)^k \cap C) \geq d(RP, (1+\varepsilon)(RP)^k \cap C) - \varepsilon\sqrt{n} \geq (\frac{1}{6} - \frac{1}{10})\sqrt{n}$. This concludes the proof of the theorem. $\square$

2.7 Lower bounds on approximation along most directions

We now prove Theorem 8. The main tool we use in this section is concentration of Lipschitz functions on Gaussian spaces.

Theorem 16 (Inequality (1.6) of [72])**.** Let $G_1, G_2, \ldots, G_n$ be independent standard Gaussian random variables, and let $f : \mathbb{R}^n \to \mathbb{R}$ be an L-Lipschitz function, namely for all $x, x' \in \mathbb{R}^n$, $|f(x) -$

$f(x')| \leq L \cdot \|x - x'\|$. Then letting $Z = f(G_1, G_2, \ldots, G_n)$, for $t > 0$ we have

$$\mathbb{P}\left(|Z - \mathbb{E}(Z)| \geq t\right) \leq 2\exp\left(-\frac{t^2}{2L^2}\right)$$

To prove Theorem 8, recall that $P = \overline{\mathscr{P}_{n/10,n}}$. Let $G = (G_1, G_2, \ldots, G_n)$ be a random vector whose components are independent standard Gaussians. It is well-known that $\frac{G}{\|G\|_2}$ is uniformly distributed in the sphere (see for instance [72], page 55). Notice that $gap_P^k(\cdot)$ is positive homogeneous, so $gap_P^k\left(\frac{G}{\|G\|}\right) = \frac{1}{\|G\|} \cdot gap_P^k(G)$.

Our first step is to lower bound $gap_P^k(G)$ with high probability, starting by lower bounding the maximization of G over P^k.

Claim 17. With probability at least $1 - \frac{1}{n}$, $\max_{x \in P^k} Gx \geq 0.7n$.

Proof. Since $k \leq \frac{n}{10}$, we have that $P^k = [-1,1]^n$ (Lemma 4). It then follows that

$$\max_{x \in P^k} Gx = \sum_{i=1}^{n} |G_i|. \tag{2.14}$$

The random variables $|G_i|$ have *folded normal* distribution [73], for which is known that $\mathbb{E}[|G_i|] = \sqrt{2/\pi} \geq 0.79$. Since the function $(x_1, \ldots, x_n) \mapsto \sum_{i=1}^n |x_i|$ is $\sqrt{n}$-Lipschitz, we can use Theorem 16 to obtain the bound

$$\mathbb{P}\left(\sum_{i=1}^{n} |G_i| < 0.7n\right) \leq 2\exp\left(-\frac{0.09^2 n}{2}\right) \leq \frac{1}{n},$$

where the last inequality holds if $n \geq 1000$. Equation (2.14) then concludes the proof. ◇

Next we upper bound the maximization of G over P.

Claim 18. With probability at least $1 - \frac{2}{n}$, $\max_{x \in P} Gx \leq 0.6n$.

Proof. Letting $ext(P)$ denote the set of extreme points of P, notice that $\max_{x \in P} Gx = \max_{v \in ext(P)} Gv$,

so it suffices to upper bound the latter. Also notice that the extreme points of P are exactly the points in $\{-1,0,1\}^n$ with at most $\frac{n}{10}$ non-zero entries (Lemma 4).

Consider $v \in ext(P)$; we verify that $Gv \leq 0.6n$ with probability at least $1-2e^{-0.6n}$. One way of seeing this, is by noticing that since v has at most $\frac{n}{10}$ non-zero entries, $Gv = \sum_{i:v_i=1} G_i + \sum_{i:v_i=-1} -G_i$ is a function of G that has at most $\frac{n}{10}$ terms and is $\sqrt{\frac{n}{10}}$-Lipschitz, so Theorem 16 gives

$$\mathbb{P}(Gv > 0.6n) = \mathbb{P}(Gv - \mathbb{E}[Gv] > 0.6n) \leq 2e^{-0.6n}, \tag{2.15}$$

and the result follows. (Another way to see this is to use the fact that Gv is a centered Gaussian with variance at most $\frac{n}{10}$ and use a tail bound for the latter.)

Now notice that P has $\sum_{i=1}^{n/10} \binom{n}{i} 2^i \leq \frac{n}{10}\binom{n}{n/10} 2^{n/10}$ extreme points. Since $\binom{n}{t} \leq (\frac{en}{t})^t$ for all $0 < t < n$, the number of extreme points of P can be upper bounded by

$$\exp\left(\ln\left(\frac{n}{10}\right) + \frac{n}{10}\left(\ln(10e) + \ln 2\right)\right) \leq \frac{2}{n} e^{0.6n},$$

where the last inequality uses $n \geq 30$.

Then taking a union bound of (2.15) over all extreme points of P gives that with probability at least $1-\frac{2}{n}$ for all $v \in ext(P)$ we have $Gv \leq 0.6n$. This concludes the proof. $\diamond$

Finally, standard results give that $\|G\|_2 \leq 2\sqrt{n}$ with probability at least $1-2e^{-0.5n}$ (for instance, notice by Jensen's inequality $\mathbb{E}[\|G\|]^2 \leq \mathbb{E}[\|G\|^2] = n$ and apply Theorem 16 to $\|G\|$). Using the fact $n \geq 30$, we then get $\mathbb{P}(\|G\| \leq 2\sqrt{n}) \geq 1-\frac{1}{n}$. Then taking a union bound over this event and the events $\max_{x \in P^k} Gx \geq 0.7n$ and $\max_{x \in P} Gx \leq 0.6n$ gives that with probability at least $1-\frac{4}{n}$ we have $gap_P^k\left(\frac{G}{\|G\|}\right) = \frac{1}{\|G\|} \cdot gap_P^k(G) \geq \frac{\sqrt{n}}{20}$. This concludes the proof of Theorem 8.

CHAPTER 3

IMPROVING THE RANDOMIZATION STEP IN FEASIBILITY PUMP

The work presented in this chapter was submitted to SIAM Journal on Optimization through a paper authored by Santanu Dey, Andres Iroume, Marco Molinaro and Domenico Salvagnin on September 30th, 2016 [38].

3.1 Introduction

Primal heuristics are used within mixed-integer linear programming (MILP) solvers for finding good integer feasible solutions quickly [44]. *Feasibility pump* (FP) is a very successful primal heuristic for mixed-binary LPs that was introduced in [43]. At its core, Feasibility Pump is an *alternating projection method*, as described below.

Algorithm 1 Feasibility Pump (Naïve version)

1: **Input:** mixed-binary LP (with binary variables x and continuous variables y)

2: Solve the linear programming relaxation, and let $(\bar{x}, \bar{y})$ be an optimal solution

3: **while** $\bar{x}$ is not integral **do**

4: (Round) Round each coordinate of $\bar{x}$ to the closest integer, call the obtained vector $\widetilde{x}$

5: (Project) Let $(\bar{x}, \bar{y})$ be the point in the LP relaxation that minimizes $\sum_i |x_i - \widetilde{x}_i|$

6: Return $(\bar{x}, \bar{y})$

The scheme presented above may *stall*, since the same infeasible integer point may be visited in Step 4 at different iterations. Whenever this happens, the paper [43] recommends a *randomization step*, that after Step 4 flips the value of some of the binary variables as follows: Defining the *fractionality* of variable x_i as $|\bar{x}_i - \tilde{x}_i|$ and let NN be the number of variables with positive fractionality, randomly generate a positive integer TT and flip $\min\{TT, NN\}$ variables with largest fractionality.

Together with a few other tweaks, this surprisingly simple method works very well. On MIPLIB 2003 instances, FP finds feasible solutions for 96.3% of the instances in reasonable time [43].

Due to its success, many improvements and generalizations of FP (both for MILPs and mixed integer non-linear programs(MINLPs)) have been studied [3, 14, 21, 45, 31, 42, 20, 41, 19]. However, the focus of these improvements has been on the projection and rounding steps or generalization for MINLPs; to the best of our knowledge, they use essentially the same randomization step as proposed in the original algorithm [43] (and its generalization to the general integer MILP case of [14]).

Moreover, even though FP is so successful and so many variants have been proposed, there is very limited theoretical analysis of its properties [20]. In particular, to the best of our knowledge there is no known bounds on expected running-time of FP.

3.2 Our contributions

In this chapter, we start a more in-depth study of the randomization step in Feasibility Pump. For that, we propose a new randomization step RANDWALKSAT$_\ell$ and provide both *theoretical analysis* as well as *computational experiments* in a state-of-the-art Feasibility Pump code that show the potential of this method.

Theoretical justification of RANDWALKSAT$_\ell$. The new randomization step RANDWALKSAT$_\ell$ is inspired by the classical algorithm WALKSAT [90] for solving SAT instances (see also [83, 79]). The key idea of RANDWALKSAT$_\ell$ is that whenever Feasibility Pump stalls, namely an infeasible mixed-binary solution is revisited, it should flip a binary variable that participates in an *infeasible constraint*. More precisely, RANDWALKSAT$_\ell$ constructs a *minimal (projected) infeasibility certificate* for this solution and *randomly picks* a binary variable in it to be flipped (see Section 3.3 for exact definitions).

While the vague intuition that such randomization is trying to "fix" the infeasible constraint is

clear, we go further and provide theoretical analyses that formally justify this and highlight more subtle advantageous properties of RANDWALKSAT$_\ell$.

First, we analyze what happens if we simply repeatedly use *only* the new proposed randomization step RANDWALKSAT$_\ell$, which gives a simple primal heuristic that we denote by MBWALKSAT. Not only we show that MBWALKSAT is guaranteed to find a solution if one exists, but its behavior is related to the *(almost) decomposability* and *sparsity* of the instance. To make this precise, consider a decomposable mixed-binary set with k blocks:

$$\begin{gathered} P^I = P_1^I \times \ldots \times P_k^I, \text{ where for all } i \in [k] \text{ we have} \\ P_i^I = P_i \cap (\{0,1\}^{n_i} \times \mathbb{R}^{d_i}),\ P_i = \{(x^i, y^i) \in [0,1]^{n_i} \times \mathbb{R}^{d_i} : A^i x^i + B^i y^i \leq b^i\}. \qquad (3.1) \\ \text{Let } P = P_1 \times \ldots \times P_k \text{ denote the LP relaxation of } P^I. \end{gathered}$$

Note that since we allow $k = 1$, this also captures a general mixed-binary set. We then have the following running-time guarantee for the primal heuristic MBWALKSAT.

Theorem 1. Consider a feasible decomposable mixed-binary set as in equation (3.1). Let s_i be such that each constraint in P_i^I has at most s_i binary variables, and define $c_i := \min\{s_i \cdot (d_i + 1), n_i\}$. Then with probability at least $1 - \delta$, MBWALKSAT with parameter $\ell = 1$ returns a feasible solution within $\ln(k/\delta) \sum_i n_i 2^{n_i \log c_i}$ iterations. In particular, this bound is at most $\bar{n} k 2^{\bar{n} \log \bar{n}} \cdot \ln(k/\delta)$, where $\bar{n} = \max_i n_i$.

There are a few interesting features of this bound that indicates good properties of the proposed randomization step, apart from the fact that it is already able to find feasible solutions by itself. First, it depends on the *sparsity* s_i of the blocks, giving better running times on sparser problems. More importantly, the bound indicates that the algorithm works almost *independently* on each of the blocks, that is, it just takes about 2^{n_i} iterations to find a solution for each of the blocks, instead of $2^{n_1 + \ldots + n_k}$ of a complete enumeration over the whole problem. In fact, the proof of Theorem 1 makes explicit this almost independence of the algorithm over the blocks, and motivates the uses

of *minimal* infeasibility certificates. Moreover, we note the important point that the algorithm is not provided the knowledge of the decomposability of the instance, it just *automatically* runs "fast" when the problem is decomposable. This gives some indication that the proposed randomization could still exhibit good behavior on the *almost decomposable* instances often found in practice (see discussion in [39]).

RANDWALKSAT$_\ell$ in conjunction with FP. Next, we analyze RANDWALKSAT$_\ell$ in the context of Feasibility Pump by adding it as a randomization step to the Naïve Feasbility Pump algorithm (Algorithm 1); we call the resulting algorithm WFP. This now requires understanding the complicated interplay of the randomization, rounding and projection steps: While in practice rounding and projection greatly help finding feasible solutions, their worst-case behavior is difficult to analyze and in fact they could take the iterates far away from feasible solutions. Although the general case is elusive at this point, we are nonetheless able to analyze the running time of WFP for *decomposable subset-sum* instances.

Definition 19. A *separable subset-sum set* is one of the form

$$\{(x^1, x^2, \ldots, x^k) \in \{0,1\}^{n_1+n_2+\ldots+n_k} : a^i x^i = b_i \ \ \forall i\} \tag{3.2}$$

for non-negative (a^i, b_i)'s.

While this may seem like a simple class of problems, on these instances Feasibility Pump with the original randomization step from [43] (without restarts) may not even converge, as illustrated next.

Remark 20. Consider the feasible subset-sum problem

$$\begin{aligned} \max\ & x_2 \\ s.t.\ & 3x_1 + x_2 = 3 \\ & x_1, x_2 \in \{0,1\}. \end{aligned}$$

Consider the execution of the original Feasibility Pump algorithm (without restarts). The starting point is an optimal LP solution; without loss of generality, suppose it is the solution $(\frac{2}{3}, 1)$. This solution is then rounded to the point $(1,1)$, which is infeasible. This point is then ℓ_1-projected to the LP, giving back the point $(\frac{2}{3}, 1)$, which is then rounded again to $(1,1)$. At this point the algorithm has stalled and applies the randomization step. Since only variable x_2 has strictly positive fractionality $|\frac{2}{3} - 1| = \frac{1}{3}$, only the first coordinate of $(1,1)$ is a candidate to be flipped. So suppose this coordinate is flipped. The infeasible point $(0,1)$ obtained is then ℓ_1-projected to the LP, giving again the point $(\frac{2}{3}, 1)$. This sequence of iterates repeats indefinitely and the algorithm does not find the feasible solution $(1,0)$.

The issue in this example is that the original randomization step never flips a variable with zero fractionality. Moreover, in Appendix B we show that even if such flips are considered, there is a more complicated subset-sum instance where the algorithm stalls.

On the other hand, we show that algorithm WFP with the proposed randomization step always finds a feasible solution of feasible subset-sum instances, and moreover its running time again depends on the sparsity and the decomposability of the instance (in order to simplify the proof, we assume that $\widetilde{x} \notin P$, then ℓ_1-proj$(P, \widetilde{x})$ is a vertex of P; notice that since ℓ_1-proj$(P, \widetilde{x})$ is a linear programming problem and subset-sum instances are bounded, there is always a vertex satisfying the desired properties from ℓ_1-proj).

Theorem 2. Consider a feasible separable subset-sum set P as in (3.2). Then with probability at least $1 - \delta$, WFP with $\ell = 2$ returns a feasible solution within $T = \lceil \ln(k/\delta) \rceil \sum_i n_i 2^{2n_i \log n_i} \leq$

$\bar{n}k\,2^{2\bar{n}\log\bar{n}}\cdot\ln(k/\delta)$ iterations, where $\bar{n}=\max_i n_i$.

To the best of our knowledge this is the first theoretical analysis of the running-time of a variant of Feasibility Pump algorithm, even for a special class of instances. As in the case of repeatedly using just RANDWALKSAT$_\ell$, the algorithm WFP essentially works independently on each of the blocks (inequalities) of the problem, and has reduced running time on sparser instances.

The high-level idea of the proof Theorem 2 is to: 1) Show that the combination of projection plus rounding is *idempotent* for these instances, namely applying them once or repeatedly yields the same effect (Lemma 24); 2) Show that a round of randomization step plus projection plus rounding has a non-zero probability of generating an iterate closer to a feasible solution (Lemma 27).

Computational experiments. While the analyses above give insights on the usefulness of using RANDWALKSAT$_\ell$ in the randomization step of FP, in order to attest its practical value it is important to understand how it interacts with complex engineering components present in current Feasibility Pump codes. To this end, we considered the state-of-the-art code of [45] and modified its randomization step based on RANDWALKSAT$_\ell$. While the full details of the experiments are presented in Section 3.5, we summarize some of the main findings here.

We conducted experiments on MIPLIP 2010 [65] instances and on randomly generated two-stage stochastic models. In the first testbed there was a small but consistent improvement in both running-time and number of iterations. More importantly, the success rate of the heuristic improved consistently. In the second testbed, the new algorithm performs even better, according to all measures. It is somewhat surprising that our small modification of the randomization step could provide noticeable improvements over the code in [45], specially considering that it already includes several improvements over the original Feasibility Pump (e.g. constraint propagation). In addition, the proposed modification is generic and could be easily incorporated in essentially any Feasibility Pump code. Moreover, for virtually all the seeds and instances tested the modified

algorithm performed better than the original version in [45]; this indicates that, in practice, the modified randomization step dominates the previous one.

The rest of the chapter is organized as follows: Section 3.3 we discuss and present out analysis of the proposed randomization scheme RANDWALKSAT$_\ell$, Section 3.4 presents the analysis of the new randomization scheme RANDWALKSAT$_\ell$ in conjunction with feasibility pump, and Section 3.5 describes details of our empirical experiments.

Notation. We use $\mathbb{R}_+$ to denote the non-negative reals, and $[k] := \{1, 2, \ldots, k\}$. For a vector $v \in \mathbb{R}^n$, we use $\text{supp}(v) \subseteq [n]$ to denote its support, namely the set of coordinates i where $v_i \neq 0$. We also use $\|v\|_0 = |\text{supp}(v)|$, and $\|v\|_1 = \sum_i |v_i|$ to denote the ℓ_1 norm.

3.3 New randomization step RANDWALKSAT$_\ell$

3.3.1 Description of the randomization step

We start by describing the WALKSAT algorithm [90], that serves as the inspiration for the proposed randomization step RANDWALKSAT$_\ell$, in the context of pure-binary linear programs. The vanilla version of WALKSAT starts with a random point $\bar{x} \in \{0,1\}^n$; if this point is feasible, the algorithm returns it, and otherwise selects any constraint violated by it. The algorithm then select a random index i from the support of the selected constraint and flips the value of the entry $\bar{x}_i$ of the solution. This process is repeated until a feasible solution is obtained. It is known that this simple algorithm finds a feasible solution in expected time at most 2^n (see [80] for a proof for 3-SAT instances), and Schöning [90] showed that if the algorithm is restarted at every $3n$ iterations, a feasible solution is found in expected time at most a polynomial factor from $(2(1-\frac{1}{s}))^n$, where s is the largest support size of the constraints.

Based on this WALKSAT algorithm, to obtain a randomization step for mixed-binary problems we are going to work on the projection onto the binary variables, so instead of looking for violated constraints we look for a *certificate of infeasibility* in the space of binary variables. Importantly, we

use a **minimal** certificate, which makes sure that for decomposable instances the certificate does not "mix" the different blocks of the problem.

Now we proceed with a formal description of the proposed randomization step RANDWALKSAT$_\ell$. Consider a mixed-binary set

$$P^I = P \cap (\{0,1\}^n \times \mathbb{R}^d), \text{ where } P = \{(x,y) \in [0,1]^n \times \mathbb{R}^d : Ax + By \leq b\}. \tag{3.3}$$

We use $\text{proj}_{bin} P$ to denote the projection of P onto the binary variables x.

Definition 21 (Projected certificates)**.** Given a mixed-binary set P^I as in (3.3) and a point $(\bar{x}, \bar{y}) \in \{0,1\}^n \times \mathbb{R}^d$ such that $\bar{x} \notin \text{proj}_{bin} P$, a *projected certificate* for $\bar{x}$ is an inequality $\lambda Ax + \lambda By \leq \lambda b$ with $\lambda \in \mathbb{R}^m_+$ such that: (i) $\bar{x}$ does not satisfy this inequality; (ii) $\lambda B = 0$. A *minimal* projected certificate is one where the support of the vector λ is minimal (i.e. the certificate uses a minimal set of the original inequalities).

Standard Fourier-Motzkin theory guarantees us that projected certificates always exist, and furthermore Caratheodory's theorem [91] guarantees that minimal projected certificates use at most $d+1$ inequalities. Together these give the following lemma.

Lemma 22. Consider a mixed-binary set P^I as in (3.3) and a point $(\bar{x}, \bar{y}) \in \{0,1\}^n \times \mathbb{R}^d$ such that $\bar{x} \notin \text{proj}_{bin} P$. There exists a vector $\lambda \in \mathbb{R}^m_+$ with support of size at most $d+1$ such that $\lambda Ax + \lambda By \leq \lambda b$ is a minimal projected certificate for $\bar{x}$. Moreover, this minimal projected certificate can be obtained in polynomial-time (by solving a suitable LP).

For completeness, see Appendix B for a proof of Lemma 22.

Now we can formally define the randomization step RANDWALKSAT$_\ell$ (notice that the condition $\lambda B = 0$ guarantees that a projected certificate has the form $ax \leq b$).

Algorithm 2 RANDWALKSAT$_\ell(\bar{x})$

1: //Assumes that $\bar{x}$ does not belong to $\text{proj}_{bin} P$

2: Let $ax \leq b$ be a minimal projected certificate for $\bar{x}$

3: Sample ℓ indices from the support $\text{supp}(a)$ uniformly and independently, let $\mathbf{I}$ be the set of indices obtained

4: (Flip coordinates) For all $i \in \mathbf{I}$, set $\bar{x}_i \leftarrow 1 - \bar{x}_i$

Note that in the pure-binary case and $\ell = 1$, this is reduces to the main step executed during WALKSAT. We remark that the flexibility of introducing the parameter ℓ will be needed in Section 3.4.

3.3.2 Analyzing the behavior of RANDWALKSAT$_\ell$

In this section we consider the behavior of the algorithm MBWALKSAT that tries to find a feasible mixed-binary solution by just repeatedly applying the randomization step RANDWALKSAT$_\ell$.

Algorithm 3 MBWALKSAT

1: **input parameter:** Integer $\ell \geq 1$

2: (Starting solution) Consider any mixed-binary point $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in \{0,1\}^n \times \mathbb{R}^d$

3: **loop**

4: **if** $\bar{\mathbf{x}}$ does not belong to $\text{proj}_{bin} P$ **then**

5: RANDWALKSAT$_\ell(\bar{\mathbf{x}})$

6: **else**

7: (Output feasible lift of $\bar{\mathbf{x}}$) Find $\bar{\mathbf{y}} \in \mathbb{R}^d$ such that $(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \in P$, return $(\bar{\mathbf{x}}, \bar{\mathbf{y}})$

As mentioned in the introduction, we show that this algorithm find a feasible solution if such exists, and the running-time improves with the sparsity and decomposability of the instance. Recall the definition of a decomposable mixed-binary problem from equation (3.1), and let certSupp_i

denote the maximum support size of a minimal projected certificate for the instance P_i^I which consists only of the ith block.

Theorem 3 (Theorem 1 restated)**.** Consider a feasible decomposable mixed-binary set as in equation (3.1). Then with probability at least $1-\delta$, MBWALKSAT with parameter $\ell = 1$ returns a feasible solution within $T = \lceil \ln(k/\delta) \rceil \sum_i n_i 2^{n_i \log \text{certSupp}_i}$ iterations.

In light of Lemma 22, if each constraint in P_i has at most s_i integer variables, we have $\text{certSupp}_i \leq \min\{s_i \cdot (d_i+1), n_i\}$, and thus this statement indeed implies Theorem 1 stated in the introduction. We remark that similar guarantees can be obtained for general ℓ, but we focus on the case $\ell = 1$ to simplify the exposition.

The high-level idea of the proof of Theorem 3 is the following:

1. First we show that if we run MBWALKSAT over a single block P_i^I, then with high probability the algorithm returns a feasible solution within $n_i \; 2^{n_i \log \text{certSupp}_i} \cdot \ln(1/\delta)$ iterations. This analysis is inspired by the one given by Schöning [90] and argues that with a small, but non-zero, probability the iteration of the algorithm makes the iterate $\bar{\boldsymbol{x}}$ closer (in Hamming distance) to a fixed solution x^* for the instance.

2. Next, we show that when running MBWALKSAT over the whole decomposable instance each iteration only depends on **one** of the blocks P_i^I; this uses the minimality of the certificates. So in effect the execution of MBWALKSAT can be split up into independent executions over each block, and thus we can put together the analysis from Item 1 for all blocks with a union bound to obtain the result.

For the remainder of the section we prove Theorem 3. We start by considering a general mixed-binary set as in equation (3.3). Given such mixed-binary set P^I, we use $\text{certSupp} = \text{certSupp}(P^I)$ to denote the maximum support size of all minimal projected certificates.

Theorem 4. Consider the execution of MBWALKSAT over a feasible mixed-binary program as in equation (3.3). The probability that MBWALKSAT does not find a feasible solution within the first T iterations is at most $(1-p)^{\lfloor T/n \rfloor}$, where $p = \text{certSupp}^{-n}$. In particular, for $T = n \cdot 2^{n \log(\text{certSupp})} \cdot \lceil \ln(1/\delta) \rceil$ this probability is at most δ (this follows from the inequality $(1-x) \leq e^{-x}$ valid for $x \geq 0$).

Proof. Consider a fixed solution $x^* \in \text{proj}_{bin} P$. To analyze MBWALKSAT, we only keep track of the Hamming distance of the (random) iterate $\bar{\mathbf{x}}$ to x^*; let $\boldsymbol{X}_t$ denote this (random) distance at iteration t, for $t \geq 1$. If at some point this distance vanishes, i.e. $\boldsymbol{X}_t = 0$, we know that $\bar{\mathbf{x}} = x^*$ and thus $\bar{\mathbf{x}} \in \text{proj}_{bin} P$; at this point the algorithm returns a feasible solution for P^I.

Fix an iteration t. To understand the probability that $\boldsymbol{X}_t = 0$, suppose that in this iteration $\bar{\mathbf{x}}$ does not belong to $\text{proj}_{bin} P$, and let $ax \leq b$ be the minimal projected certificate for it used in RANDWALKSAT$_1$. Since the feasible point x^* satisfies the inequality $ax \leq b$ but $\bar{\mathbf{x}}$ does not, there must be at least one index $\mathbf{i}^*$ in the support of a such where x^* and $\bar{\mathbf{x}}$ differ. Then if algorithm MBWALKSAT makes a "lucky move" and chooses $\mathbf{I} = \{\mathbf{i}^*\}$ in Line 3, the modified solution after flipping this coordinate (the next line of the algorithm) is one unit closer to x^* in Hamming distance, hence $\boldsymbol{X}_{t+1} = \boldsymbol{X}_t - 1$. Moreover, since $\mathbf{I}$ is independent of $\mathbf{i}$, the probability of choosing $\mathbf{I} = \{\mathbf{i}^*\}$ is $1/|\text{supp}(a)| \geq 1/\text{certSupp}$.

Therefore, if we start at iteration t and for all the next $\boldsymbol{X}_t$ iterations either the iterate belongs to $\text{proj}_{bin} P$ or the algorithm makes a "lucky move", it terminates by time $t + \boldsymbol{X}_t$. Thus, with probability at least $(1/\text{certSupp})^{\boldsymbol{X}_t} \geq (1/\text{certSupp})^n = p$ the algorithm terminates by time $t + \boldsymbol{X}_t \leq t + n$.

To conclude the proof, let $\alpha = \lfloor T/n \rfloor$ and call iterations $i \cdot n, \ldots, (i+1) \cdot n - 1$ the i-th block of iterations. If the algorithm has not terminated by iteration $i \cdot n - 1$, then with probability at least p it terminates within the next n iterations, and hence within the i-th block. Putting these bounds together for all α blocks, the probability that the algorithm *does not* stop by the end of block α is at most $(1-p)^\alpha$. This concludes the proof. □

Going back to decomposable problems, we now make formal the claim that minimal projected certificates for decomposable mixed-binary sets do not mix the constraints from different blocks. Notice that projected certificates for a decomposable mixed-binary set as in equation (3.1) have the form $\sum_i \lambda^i A^i x^i \leq \sum_i \lambda^i b^i$ and $\lambda^i B^i = 0$ for all $i \in [k]$.

Lemma 23. Consider a decomposable mixed-integer set as in equation (3.1). Consider a point $\bar{x} \notin \text{proj}_{bin} P$ and let $\sum_i \lambda^i A^i x^i \leq \sum_i \lambda^i b^i$ be a minimal projected certificate for $\bar{x}$. Then this certificate uses only inequalities from one block P^j, i.e. there is j such that $\lambda^i = 0$ for all $i \neq j$. Moreover, $\bar{x}^j \notin \text{proj}_{bin} P_j$.

Proof. Let $\bar{x} = (\bar{x}^1, \bar{x}^2, \ldots, \bar{x}^k)$ and call the certificate $(ax \leq b) \triangleq (\sum_i \lambda^i A^i x^i \leq \sum_i \lambda^i b^i)$. By definition of projected certificate we have $\sum_i \lambda^i A^i \bar{x}^i > \sum_i \lambda^i b^i$, and thus by linearity there must be an index j such that $\lambda^j A^j \bar{x}^j > \lambda^j b^j$. Moreover, as remarked earlier, decomposability implies that the certificate satisfies $\lambda^i B^i = 0$ for all i, so in particular for j. Thus, the inequality $\lambda^j (A^j, B^j)(x^j, y^j) \leq \lambda^j b^j$ obtained by combining only the inequalities form P_j is a projected certificate for $\bar{x}$. The minimality of the original certificate $ax \leq b$ implies that $\lambda^i = 0$ for all $i \neq j$. This concludes the first part of the proof.

Moreover, since $\lambda^j A^j \bar{x}^j > \lambda^j b^j$ and $\lambda^j B^j = 0$ we have that $\lambda^j (A^j, B^j)(\bar{x}^j, y) > \lambda^j b^j$ for all y, and hence $\bar{x}^j$ does not belong to $\text{proj}_{bin} P_j$. This concludes the proof. □

We can finally prove the desired theorem.

Proof of Theorem 3. We use the natural decomposition $\bar{\mathbf{x}} = (\bar{\mathbf{x}}^1, \ldots, \bar{\mathbf{x}}^k) \in \{0,1\}^{n_1} \times \ldots \times \{0,1\}^{n_k}$ of the iterates of the algorithm. From Lemma 23, we have that for each scenario, each iteration of MBWALKSAT is associated with just one of the blocks P_j^I's, namely the P_j^I containing all the inequalities in the minimal projected certificate used in this iteration; let $\mathbf{J}_t \in [k]$ denote the (random) index j of the block associated to iteration t. Notice that at iteration t, only the binary variables $x^{\mathbf{J}_t}$ can be modified by the algorithm.

Let $T_i = n_i 2^{n_i \log n_i} \lceil \ln(k/\delta) \rceil$. Applying the proof of Theorem 4 to the iterations $\{t : \mathbf{J}_t = i\}$ with index i, we get that with probability at least $1 - \frac{\delta}{k}$ the algorithm finds some $\bar{\mathbf{x}}^i$ in $\text{proj}_{bin} P_i$ within the first T_i of these iterations. Moreover, after the algorithm finds such a point, it does not change it (that is, the remaining iterations have index $\mathbf{J}_t \neq i$, due to the second part of Lemma 23).

Therefore, by taking a union bound we get that with probability at least $1 - \delta$, *for all* $i \in [k]$ the algorithm finds $\bar{\mathbf{x}}^i \in \text{proj}_{bin} P_i$ within the first T_i iterations with index i (for a total of $\sum_i T_i = T$ iterations). When this happens, the total solution $\bar{\mathbf{x}}$ belongs to $\text{proj}_{bin} P$ and the algorithm returns. This concludes the proof. □

3.4 Randomization step RANDWALKSAT$_\ell$ within Feasibility Pump

In this section we incorporate the randomization step RANDWALKSAT$_\ell$ into the Naïve Feasibility Pump, the resulting algorithm being called WFP. We describe this algorithm in a slightly different way and using a notation more convenient for the analysis.

Consider a mixed-binary set P^I as in equation (3.3). Given a 0/1 point $\widetilde{x} \in \{0,1\}^n$, let ℓ_1-proj$(P, \widetilde{x})$ denote a point (x, y) in P where $\|\widetilde{x} - x\|_1$ is as small as possible. Also, for a vector $v \in [0,1]^p$, we use round(v) to denote the vector obtained by rounding each component of v to the closest integer; we use the convention that $\frac{1}{2}$ is rounded to 1, but any consistent rounding would suffice. Notice that operations 'ℓ_1-proj' and 'round' correspond precisely to Steps 5 and 4 in the Naïve Feasibility Pump. With this notation, algorithm WFP can be described as follows.

Note that stalling in the above algorithm is determined using the condition $\widetilde{\mathbf{x}}^t = \widetilde{\mathbf{x}}^{t-1}$. What about 'long cycle' stalling, that is $\widetilde{\mathbf{x}}^t = \widetilde{\mathbf{x}}^{t'}$ where $t' < t-1$, but $\widetilde{\mathbf{x}}^{t'}, \ldots, \widetilde{\mathbf{x}}^{t-1}$ are all distinct binary vectors. As it turns out (assuming no numerical errors) a consistent rounding rule implies that stalling will always occur with cycles of length two.

Theorem 5. With consistent rounding, long cycles cannot occur.

We present a proof of 5 in Appendix B. For the remainder of the section, we analyze the

Algorithm 4 WFP

input parameter: integer $\ell \geq 1$
Let $(\bar{x}^0, \bar{y}^0)$ be an optimal solution of the LP relaxation
Let $\widetilde{x}^0 = \text{round}(\bar{x}^0)$
for t = 1,2,... **do**
    $(\bar{\mathbf{x}}^t, \bar{\mathbf{y}}^t) = \ell_1\text{-proj}(P, \widetilde{\mathbf{x}}^{t-1})$
    $\widetilde{\mathbf{x}}^t = \text{round}(\bar{\mathbf{x}}^t)$
    if $(\widetilde{\mathbf{x}}^t, \bar{\mathbf{y}}^t) \in P$ **then** ▷ equivalently, $\widetilde{\mathbf{x}}^t \in \text{proj}_{bin}(P)$
        Return $(\widetilde{\mathbf{x}}^t, \bar{\mathbf{y}}^t)$
    if $\widetilde{\mathbf{x}}^t = \widetilde{\mathbf{x}}^{t-1}$ **then** ▷ iterations have stalled
        $\widetilde{\mathbf{x}}^t = \text{RANDWALKSAT}_\ell(\widetilde{\mathbf{x}}^t)$

behavior of algorithm WFP on separable subset-sum instances, proving Theorem 2 stated in the introduction.

3.4.1 Running time of WFP for separable subset-sum instances: Proof of Theorem 2

Notice that the projection operators 'ℓ_1-proj' and 'round' now present also act on each block independently, namely given a point $x = (x^1, \ldots, x^k) \in \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_k}$, if $(\check{x}^1, \ldots, \check{x}^k) = \ell_1\text{-proj}(P, x)$ then $\check{x}^i = \ell_1\text{-proj}(P_i, x^i)$ for all $i \in [k]$, and similarly for 'round'. Therefore, as in the proof of Theorem 3, it suffices to analyze the execution of algorithm WFP over a single block/inequality of the separable subset-sum problem. More precisely, it suffices to prove the following guarantee for WFP on a general subset-sum instance.

Theorem 6. Consider a feasible subset-sum problem $P \subseteq \mathbb{R}^n$. Then for every $T \geq 1$, the probability that WFP with $\ell = 2$ does not find a feasible solution within the first $2T$ iterations is at most $(1-p)^{\lfloor T/n \rfloor}$, where $p = (1/n^2)^n$. In particular, for $T = n \cdot 2^{2n \log n} \cdot \lceil \ln(1/\delta) \rceil$ this probability is at most δ.

The high-level idea of the proof of this theorem is the following. We use a similar strategy as before, where we consider a fixed feasible solution x^* and track its distance to the iterates $\widetilde{\boldsymbol{x}}^t$ generated by algorithm WFP. However, while again the randomization step RANDWALKSAT_2

brings $\widetilde{\boldsymbol{x}}^t$ closer to x^* with small but non-zero probability, the issue is that the projections 'ℓ_1-proj' and 'round' in the next iterations could send the iterate even further from x^*. To analyze the algorithm we then use the structure of subset-sum instances to: 1) First control the combination 'ℓ_1-proj + round' in Steps 5 and 6, showing that in this case they are *idempotent*, namely applying them once or repeatedly yields the same effect (Lemma 24); 2) Strengthen the analysis of Theorem 3 to show that a round of RANDWALKSAT$_2$ *plus* 'ℓ_1-proj + round' still has a non-zero probability of generating a point closer to x^* (Lemma 27). For this, it will be actually important that we use $\ell = 2$ in algorithm WFP (actually $\ell \geq 2$ suffices).

For the remainder of the section we prove Theorem 6. To simplify the notation we omit the polytope P from the notation of ℓ_1-proj. We assume that our subset-sum problem $P = \{x \in [0,1]^n : ax = b\}$ is such that *all* coordinates of a are positive, since components with $a_i = 0$ do not affect the problem (more precisely, after the first iteration of the algorithm, the value of $\widetilde{\boldsymbol{x}}_i^t$ is set to 0 or 1 and does not change anymore, and this value does not affect the feasibility of the solutions $\widetilde{\boldsymbol{x}}^t$'s). Also remember that subset-sum problems only have binary variables.

Given a point $\widetilde{x} \in \{0,1\}^n$, let AltProj$(\widetilde{x}) \in \{0,1\}^n$ be the effect of applying to $\widetilde{x}$ ℓ_1-proj$(.)$ and then round$(.)$. Notice that if $\widetilde{x}$ belongs to P, then AltProj$(\widetilde{x}) = \widetilde{x}$. Then algorithm WFP can be thought as performing a AltProj operation, then checking if the iterate obtained either belongs to P (in which case it exits) of if it equals the previous iterate (in which case it applies RANDWALKSAT$_2$); if neither of these occur, then another AltProj operation is performed. So an important component for analyzing this algorithm is getting a good control over a sequence of AltProj operations. For that, define the iterated operation $\text{AltProj}^t(\widetilde{x}) = \text{AltProj}\left(\text{AltProj}^{t-1}(\widetilde{x})\right)$ (with $\text{AltProj}^1 = \text{AltProj}$) and if the sequence $(\text{AltProj}^t(\widetilde{x}))$ stabilizes at a point, let $\text{AltProj}^*(\widetilde{x})$ denote this point.

A crucial observation, given by the next lemma, is that for subset-sum instances the operation of AltProj is idempotent, namely it stabilizes after just one operation.

Lemma 24. Let P be a subset-sum instance. Then for every $\widetilde{x} \in \{0,1\}^n$, $\text{AltProj}^*_P(\widetilde{x}) = \text{AltProj}_P(\widetilde{x})$.

Proof. Again to simplify the notation we omit the polyhedron P when writing ℓ_1-proj and AltProj. Let $\bar{x} = \ell_1\text{-proj}(\widetilde{x})$ and recall it is an extreme point of P. Clearly, if $\widetilde{x} \in P$ then $\text{AltProj}(\widetilde{x}) = \widetilde{x}$ and hence $\text{AltProj}^*(\widetilde{x}) = \text{AltProj}(\widetilde{x})$. Similarly, if $\bar{x}$ is a 0/1 point then $\text{AltProj}(\widetilde{x}) = \bar{x}$, and again $\text{AltProj}^*(\widetilde{x}) = \text{AltProj}(\widetilde{x})$.

Thus, assume that $\widetilde{x} \notin P$ and $\bar{x}$ is not a 0/1 point. Since $\bar{x}$ is an extreme point of the subset-sum LP P it has exactly 1 fractional coordinate, so by permuting indices we assume without loss of generality:

1. $\bar{x}_1 = \cdots = \bar{x}_k = 1$.
2. $\bar{x}_{k+1} \in (0,\ 1)$.
3. $\bar{x}_{k+2} = \cdots = \bar{x}_n = 0$
4. $a_{k+2} \geq a_{k+3} \geq \cdots \geq a_n$.
5. $a_1 \leq a_2 \leq a_3 \leq \cdots \leq a_k$.

Now we look at the points obtained after applying round(.) and ℓ_1-proj(.) to $\bar{x}$, namely let $\widetilde{x}' := \text{round}(\bar{x}) = \text{AltProj}(\widetilde{x})$ and let $\bar{x}' := \ell_1\text{-proj}(\widetilde{x}')$. Notice that $\bar{x}'$ is obtained by solving:

$$\begin{array}{ll} \min & \sum_{\{j \mid \widetilde{x}'_j = 0\}} x_j + \sum_{\{j \mid \widetilde{x}'_j = 1\}} (1 - x_j) \\ \text{s.t.} & ax = b \\ & 0 \leq x \leq 1. \end{array} \tag{3.4}$$

Case 1: $\bar{x}_{k+1} < 1/2$. Then $\widetilde{x}'_i = 1$ for all $i \leq k$, $\widetilde{x}'_i = 0$ for all $i \geq k+1$; also notice $\widetilde{x}' \leq \bar{x}$, and hence $a\widetilde{x}' < b$; thus $\bar{x}'$ is obtained from $\widetilde{x}$' by increasing some components of 0 value. We have three subcases:

a. If $a_{k+1} > a_{k+2}$: then a_{k+1} is the largest coordinate of a where $\widetilde{x}'$ has value 0, so it follows from (3.4) that $\bar{x}'$ is obtained from $\widetilde{x}'$ by raising its $(k+1)$-component from 0 to $\bar{x}_{k+1}$. Thus,

$\bar{x}' = \bar{x}$, and hence $\text{AltProj}(\text{AltProj}(\widetilde{x})) = \text{round}(\bar{x}')$ equals $\text{round}(\bar{x}) = \text{AltProj}(\widetilde{x})$; this implies $\text{AltProj}^*(\widetilde{x}) = \text{AltProj}(\widetilde{x})$.

b. If $a_{k+1} < a_{k+2}$: then $\bar{x}'$ is obtained from $\widetilde{x}$ by raising its $(k+2)$-component to a value that is at most $\bar{x}_{k+1} < 1/2$. Now, $\text{round}(\bar{x}') = \widetilde{x}'$, so again we get $\text{AltProj}(\text{AltProj}(\widetilde{x})) = \text{round}(\bar{x}') = \widetilde{x}' = \text{AltProj}(\widetilde{x})$ and we are done.

c. If $a_{k+1} = a_{k+2}$: Since $\bar{x}'$ is a vertex of the subset-sum LP P, again it only has 1 fractional component (either $k+1$ or $k+2$) and then it is easy to see that $\bar{x}'$ is equal to the one in either Case (a) or Case (b) above; thus the result also holds for this case.

Case 2: $\bar{x}_{k+1} \geq 1/2$. Then $\widetilde{x}'$ is such that $\widetilde{x}'_i = 1$ for all $i \leq k+1$ and $\widetilde{x}' = 0$ for all $i \geq k+2$; also notice $\widetilde{x}' \geq \bar{x}$ and hence $a\widetilde{x}' > b$. Now, consider $\bar{x}' = \ell_1\text{-proj}(\widetilde{x}')$:

a. If $a_k < a_{k+1}$: This is analogous to Case 1a: $\bar{x}'$ is obtained by lowering the $(k+1)$-coordinate of $\widetilde{x}'$ from 1 to $\bar{x}$, and thus $\bar{x}' = \bar{x}$; the rest of the proof is identical to Case 1a.

b. If $a_k > a_{k+1}$: In this case, $\bar{x}'$ is obtained by lowering the k-component of $\widetilde{x}'$. Since $a\bar{x} = a\bar{x}' = b$, and k and $(k+1)$ are the only components where $\bar{x}$ and $\bar{x}'$ differ, we have: $a_k + a_{k+1}\bar{x}_{k+1} = a_k\bar{x}'_k + a_{k+1}$. Hence $\bar{x}'_k = 1 - \frac{a_{k+1}}{a_k}(1 - \bar{x}_{k+1}) \geq 1/2$ and $\text{round}(\bar{x}') = \widetilde{x}'$; the rest of the proof is identical to Case 1b.

c. If $a_k = a_{k+1}$: Identical to Case 1c.

□

Therefore, there is not much loss in looking at a "compressed" version of algorithm WFP that packs repeated applications of AltProj until stalling happens into a single AltProj^*; more formally, we have the following algorithm (stated in the pure-binary case to simplify the notation).

Algorithm 5 WFP-Compressed

input parameter: integer $\ell \geq 1$
Let $\bar{x}^0$ be an optimal solution of the LP relaxation
Let $\widetilde{z}^0 = \text{round}(\bar{x}^0)$
for τ = 1,2,... **do**
    $\bar{\boldsymbol{z}}^\tau = \text{AltProj}^*(\widetilde{\boldsymbol{z}}^{\tau-1})$
    if $\widetilde{\boldsymbol{z}}^\tau \in P$ **then**
        Return $\widetilde{\boldsymbol{z}}^\tau$
    $\widetilde{\boldsymbol{z}}^\tau = \text{RANDWALKSAT}_\ell(\widetilde{\boldsymbol{z}}^\tau)$

Intuitively, Lemma 24 should imply that packing the repeated applications of AltProj into a single AltProj* should not save more than 1 iteration. To see this more formally, assume that both algorithms use as starting point the same optimal solution of the LP, so $\widetilde{z}^0 = \bar{x}^0$. Now condition on a scenario where we have $\widetilde{\boldsymbol{z}}^\tau = \widetilde{\mathbf{x}}^t$ at the beginning of iterations τ and t of algorithms WFP-Compressed and WFP respectively (for $\tau, t \geq 1$). Then we claim that either both algorithms return at the current iteration, or $\widetilde{\boldsymbol{z}}^{\tau+1}$ has the same distribution as either $\widetilde{\mathbf{x}}^{t+1}$ or $\widetilde{\mathbf{x}}^{t+2}$ (at the beginning of they respective iterations): If $\widetilde{\boldsymbol{z}}^\tau = \widetilde{\mathbf{x}}^t \in P$, then both algorithms return; if $\widetilde{\mathbf{x}}^t \notin P$ but $\widetilde{\mathbf{x}}^t = \widetilde{\mathbf{x}}^{t-1}$, then both algorithms WFP-Compressed and WFP employ RANDWALKSAT_2 over $\widetilde{\boldsymbol{z}}^\tau = \widetilde{\mathbf{x}}^t$, in which case $\widetilde{\boldsymbol{z}}^{\tau+1}$ has the same distribution as $\widetilde{\mathbf{x}}^{t+1}$; finally, if $\widetilde{\mathbf{x}}^t \neq \widetilde{\mathbf{x}}^{t-1}$, then WFP at the beginning of the next iteration will have $\widetilde{\mathbf{x}}^{t+1} = \text{AltProj}(\widetilde{\mathbf{x}}^t)$, which by Lemma 24 (and $t \geq 1$) equals $\widetilde{\mathbf{x}}^t$ itself, and so it will employ RANDWALKSAT_2 to $\widetilde{\mathbf{x}}^{t+1} = \widetilde{\mathbf{x}}^t$ and again we have that $\widetilde{\mathbf{x}}^{t+2}$ has the same distribution as $\widetilde{\boldsymbol{z}}^{\tau+1}$.

Therefore, since we can employ this argument to couple iterations $\leq \tau$ of WFP-Compressed with iterations $\leq 2\tau$ of WFP, we have the following result.

Lemma 25. Consider the application of algorithms WFP and WFP-Compressed over the subset-sum problem P. Then the probability that algorithm WFP returns after at most $2T$ iterations is at least the probability that algorithm WFP-Compressed after at most T iterations.

Therefore, it suffices to upper bound the number of iterations of WFP-Compressed until it returns. To avoid ambiguity, let $\boldsymbol{z}^\tau$ be the value of $\widetilde{\boldsymbol{z}}^\tau$ at the *beginning* of iteration τ of WFP-

Compressed. Notice that $z^1 = \text{AltProj}^*(\widetilde{x}^0)$, and $\boldsymbol{z}^{\tau+1} = \text{AltProj}^*(\text{RANDWALKSAT}_2(\boldsymbol{z}^\tau))$ for $\tau \geq 2$. It suffices to show that with probability at least $1-(1-p)^{T/n}$, there is $\tau \leq T/2$ such that $\boldsymbol{z}^\tau$ belongs to P.

To do so, for $\widetilde{x} \in \{0,1\}^n$ and $I \subseteq [n]$ let $\text{flip}(\widetilde{x}, I)$ denote the 0/1 vector obtained starting from $\widetilde{x}$ and flipping the value of all coordinates that belongs to I. Notice that (up to scaling) the only possible projected certificates for our subset-sum problem are $ax \geq b$ and $ax \leq b$. Since we have assumed that the vector a has full support, it follows that on this problem $\text{RANDWALKSAT}_2(\widetilde{x}) = \text{flip}(\widetilde{x}, \mathbf{I})$ for $\mathbf{I}$ being the set obtained by sampling independently two indices uniformly from $[n]$.

The next lemma then shows that there is always a "lucky choice" of set $\mathbf{I}$ in $\text{RANDWALKSAT}_2(\boldsymbol{z}^\tau)$ that brings $\boldsymbol{z}^{\tau+1} = \text{AltProj}^*(\text{RANDWALKSAT}_2(\boldsymbol{z}^\tau))$ closer to a fixed solution x^* to the subset-sum problem.

The following definition is convenient.

Definition 26. A point $\widetilde{x} \in \{0,1\}^n$ is called a stalling solution if $\text{AltProj}(\widetilde{x}) = \widetilde{x}$.

Lemma 27. Let $x^* \in \{0,1\}^n$ be a feasible solution to the subset-sum problem. Consider $\widetilde{x} \in \{0,1\}^n$ with $a\widetilde{x} \neq b$ that satisfies the fixed point condition $\text{AltProj}(\widetilde{x}) = \widetilde{x}$. Then there is a set $I \subseteq [n]$ of size at most 2 such that the point $x' = \text{AltProj}_P^*(\text{flip}(\widetilde{x}, I))$ is closer to x^* than $\widetilde{x}$, namely $\|x' - x^*\|_0 \leq \|\widetilde{x} - x^*\|_0 - 1$.

Proof. Again to simplify the notation we omit P from ℓ_1-proj and AltProj, and use $\text{flip}(\widetilde{x}, j)$ instead of $\text{flip}(\widetilde{x}, \{j\})$ in the singleton case.

We start with a couple of claims.

Claim 1 Suppose $\widetilde{x} \in \{0,1\}^n$ is a stalling point. If $a\widetilde{x} < b$, then there is $k \notin \text{supp}(\widetilde{x})$ such that $\ell_1\text{-proj}(\widetilde{x})_i = \widetilde{x}_i$ for all $i \neq k$, and $\ell_1\text{-proj}(\widetilde{x})_k \in (0, \frac{1}{2})$. Similarly, if $a\widetilde{x} > b$, then there is $k \in \text{supp}(\widetilde{x})$ such that $\ell_1\text{-proj}(\widetilde{x})_i = \widetilde{x}_i$ for all $i \neq k$, and $\ell_1\text{-proj}(\widetilde{x})_k \in [\frac{1}{2}, 1)$.

Proof of Claim 1. We only prove the first statement, the proof of the second is completely analogous. Since $\widetilde{x}$ is stalling we have that round(ℓ_1-proj($\widetilde{x}$)) $= \widetilde{x}$, and since ℓ_1-proj($\widetilde{x}$) is an extreme point of the subset-sum problem P it has at most 1 fractional component, and hence only differs in one component k from

$$\text{round}(\ell_1\text{-proj}(\widetilde{x})) = \widetilde{x}.$$

Since $a \cdot \ell_1$-proj($\widetilde{x}$) $= b > a \cdot \widetilde{x}$, we have that $\widetilde{x}_k = 0$ and ℓ_1-proj($\widetilde{x}$)$_k > 0$; since round(ℓ_1-proj($\widetilde{x}$)$_k$) $= \widetilde{x}_k = 0$, we have ℓ_1-proj($\widetilde{x}$)$_k < \frac{1}{2}$. □

Claim 2 Consider a point $\widetilde{x} \in \{0,1\}^n$.

1. If the objective value of (3.4) is strictly less than $\frac{1}{2}$, then AltProj($\widetilde{x}$) $= \widetilde{x}$.

2. If the objective value of (3.4) is strictly less than 1, then $\|\text{AltProj}(\widetilde{x}) - \widetilde{x}\|_0 \leq 1$.

Proof of Claim 2. Let $\bar{x} = \ell_1$-proj($\widetilde{x}$) be an optimal solution for (3.4). Proof of Part 1: the assumption implies that $|\bar{x}_i - \widetilde{x}_i| < \frac{1}{2}$ for all i, which directly implies that AltProj($\widetilde{x}$) $=$ round($\bar{x}$) $= \widetilde{x}$.

Proof of Part 2: the assumption implies that there can be at most one index j with $|\bar{x}_j - \widetilde{x}_j| \geq \frac{1}{2}$, which implies that for all $i \neq j$, AltProj($\widetilde{x}$)$_i$ $=$ round($\bar{x}_i$) $= \widetilde{x}_i$ and the result follows. □

Now we are ready to present the proof of Lemma 27. Let x^* and $\tilde{x}$ be as in the statement of the Lemma. From Lemma 24 we know that

$$\text{AltProj}^*(\text{flip}(\widetilde{x}, J)) = \text{AltProj}(\text{flip}(\widetilde{x}, J)),$$

so it suffices to work with the right-hand side instead. Since $\widetilde{x} \neq x^*$ we have supp($\widetilde{x}$) $\neq$ supp(x^*). We separate the proof in three cases depending on the relationship between these supports.

Case 1: supp($\widetilde{x}$) $\subsetneq$ supp(x^*): Pick any $j \in \text{supp}(x^*) \setminus \text{supp}(\widetilde{x})$ and notice that $\|\text{flip}(\widetilde{x}, j) - x^*\|_0 = \|\widetilde{x} - x^*\|_0 - 1$. Notice that both supp($\widetilde{x}$) and supp(flip($\widetilde{x}, j$)) are contained in the support of x^*, and hence we have $a\widetilde{x} \leq b$ and $a \cdot \text{flip}(\widetilde{x}, j) \leq b$. Moreover, since flip($\widetilde{x}, j$) $\geq \widetilde{x}$, it is easy to see that the

optimal value of (3.4) for $\text{flip}(\widetilde{x}, j)$ is *strictly less than* that for $\widetilde{x}$ (we need to raise fewer variables to make the point satisfy $ax = b$), which by Claim 1 is at most $\frac{1}{2}$. Thus, employing Part 1 of Claim 2 to $\text{flip}(\widetilde{x}, j)$ gives that $\text{AltProj}(\text{flip}(\widetilde{x}, j)) = \text{flip}(\widetilde{x}, j)$, which is the desired point closer to x^*.

Case 2: $\text{supp}(x^*) \subsetneq \text{supp}(\widetilde{x})$: The proof is the same as above, with the only change that we take $j \in \text{supp}(\widetilde{x}) \setminus \text{supp}(x^*)$.

Case 3: The supports $\text{supp}(x^*)$ and $\text{supp}(\widetilde{x})$ are not contained in one another. In this case $a\widetilde{x}$ can be either $< b$ or $> b$:

1. If $a\widetilde{x} < b$. Take $m \in \text{supp}(x^*) \setminus \text{supp}(\widetilde{x})$. If $a \cdot \text{flip}(\widetilde{x}, m) \leq b$, then we can argue exactly as in Case 1 to get that $\text{AltProj}(\text{flip}(\widetilde{x}, m)) = \text{flip}(\widetilde{x}, m)$, which is closer to x^* than $\widetilde{x}$. So consider the case $a \cdot \text{flip}(\widetilde{x}, m) > b$. Take $i \in \text{supp}(\widetilde{x}) \setminus \text{supp}(x^*)$ and consider $\text{flip}(\widetilde{x}, \{m, i\})$, which is 2 units closer to x^* in Hamming distance.

   We claim that the optimal value of (3.4) for $\text{flip}(\widetilde{x}, \{m, i\})$ is strictly less than 1. Suppose $a \cdot \text{flip}(\widetilde{x}, \{m, i\}) \leq b$; since $a \cdot \text{flip}(\widetilde{x}, m) > b$ (notice $\text{flip}(\widetilde{x}, m)$ is obtained from $\text{flip}(\widetilde{x}, \{m, i\})$ by increasing coordinate i to 1), this means that we can make $\text{flip}(\widetilde{x}, \{m, i\})$ satisfy $ax = b$ by increasing coordinate i to a value *strictly less* than 1, thus upper bounding the optimum of (3.4). On the other hand, consider $a \cdot \text{flip}(\widetilde{x}, \{m, i\}) > b$; notice $a \cdot \text{flip}(\widetilde{x}, i) \leq a \cdot \widetilde{x} < b$ (the last uses a running assumption), and thus again we can make $\text{flip}(\widetilde{x}, \{m, i\})$ satisfy $ax = b$ by decreasing coordinate m to a value strictly smaller than 1. This proves the claim.

   With this claim in place, we can just employ Part 2 of Claim 2 to $\text{flip}(\widetilde{x}, \{m, i\})$ and triangle inequality to obtain that $\|\text{AltProj}(\text{flip}(\widetilde{x}, \{m, i\})) - x^*\|_0$ is at most

$$1 + \|\text{flip}(\widetilde{x}, \{m, i\}) - x^*\|_0 = 1 + \|\widetilde{x} - x^*\|_0 - 2,$$

   which gives the desired result.

2. If $a\bar{x} > b$. The proof of this case mirrors that of the above case (only with the inequalities $<$

and $>$ reversed throughout).

□

Notice that since z^τ is obtained from AltProj$^*(.)$, it satisfies the fixed point condition AltProj$(z^\tau) = z^\tau$. Thus, as long as z^τ does not belong to P we can apply the above lemma to obtain that with probability at least $\frac{1}{n^2}$ we have $\mathbf{I}$ in RANDWALKSAT$_2$ equal to the set I in the lemma and thus the iterate moves closer to a feasible solution; more formally we have the following.

Corollary 28. Let $x^* \in \{0,1\}^n$ be a feasible solution to the subset-sum problem P. Then

$$\mathbb{P}\Big(\|z^{\tau+1} - x^*\|_0 \leq \|z^\tau - x^*\|_0 - 1 \;\Big|\; z^\tau \notin P\Big) \geq \frac{1}{n^2}.$$

Now we can conclude the proof of Theorem 6 arguing just like in the proof of Theorem 4.

Proof of Theorem 6. Consider $x^* \in P$ and let $\boldsymbol{Z}_\tau = \|z^\tau - x^*\|_0$. Notice that $\boldsymbol{Z}_\tau = 0$ implies $z^\tau = x^*$ and hence $z^\tau \in P$. Corollary 28 gives that $\mathbb{P}(\boldsymbol{Z}_{\tau+1} \leq \boldsymbol{Z}_\tau - 1 \mid z^\tau \notin P) \geq \frac{1}{n^2}$. Therefore, if we start at iteration τ and for all the next $\boldsymbol{Z}_\tau$ iterations either the iterate $z^{\tau'}$ belongs to P or the algorithm reduces $\boldsymbol{Z}_{\tau'}$, it terminates by time $\tau + \boldsymbol{Z}_\tau$. Thus, with probability at least $(1/n^2)^{\boldsymbol{Z}_\tau} \geq (1/n^2)^n = p$ the algorithm terminates by time $t + \boldsymbol{Z}_\tau \leq t + n$.

To conclude the proof, let $\alpha = \lfloor T/n \rfloor$ and call time steps $i \cdot n, \ldots, (i+1) \cdot n - 1$ the i-th block of time. From the above paragraph, the probability that there is τ in the ith block of time such that $z^\tau \in P$ conditioned on $z^{i \cdot n - 1} \notin P$ is at least p. Using the chain rule of probability gives that the probability that there is no $z^\tau \in P$ within any of the α blocks is at most $(1-p)^\alpha$. This concludes the proof. □

3.5 Computations

In this section, we describe the algorithms that we have implemented and report computational experiments comparing the performance of the original Feasibility Pump 2.0 algorithm from [45],

which we denote by FPORIG, to our modified code that uses the new perturbation procedure. The code is based on the current version of the Feasibility Pump 2.0 code (the one available on the NEOS servers), which is implemented in C++ and linked to IBM ILOG CPLEX 12.6.3 [61] for preprocessing and solving LPs. All features such as constraint propagation which are part of the Feasibility Pump 2.0 code have been left unchanged.

All algorithms have been run on a cluster of identical machines, each equipped with an Intel Xeon CPU E3-1220 V2 running at 3.10GHz and 16 GB of RAM. Each run had a time limit of half an hour.

3.5.1 WalkSAT-based perturbation

In preliminary tests, we implemented the algorithm WFP as described in the previous section. However, its performance was not competitive with FPORIG. In hindsight, this can be justified by the following reasons:

- Picking a fixed ℓ can be tricky. Too small or too big a value can lead to slow convergence in practical implementations.
- Using RANDWALKSAT$_\ell$ at each perturbation step can be overkill, as in most cases the original perturbation scheme does just fine.
- Computing the minimal certificate is too expensive, as it requires solving LPs.

For the reasons above, we devised a more conservative implementation of a perturbation procedure inspired by WALKSAT, which we denote by WFPBASE. The algorithm works as follows. Let $F \subset [n]$ be the set of indices with positive fractionality $|\widetilde{x}_j - \bar{x}_j|$. If $TT \leq |F|$, then the perturbation procedure is just the original one in FPORIG. Else, let S be the union of the supports of the constraints that are not satisfied by the current point $(\widetilde{x}, \bar{y})$. We select the $|F|$ indices with largest fractionality $|\widetilde{x}_j - \bar{x}_j|$ and select uniformly at random $\min\{|S|, TT - |F|\}$ indices from S, and flip the values in $\widetilde{x}$ for all the selected indices.

Note also that the above procedure applies only to the case in which a cycle of length one is detected. In case of longer cycle, we use the very same restart strategy of FPORIG.

3.5.2 Computational results

We tested the two algorithms on two classes of models: two-stage stochastic models, and the MIPLIB 2010 dataset.

Two-stage stochastic models. In order to validate the hypothesis suggested by the theoretical results that our walkSAT-based perturbation should work well on almost-decomposable models, we tested WFPBASE on two-stage stochastic models. These are the deterministic equivalent of two-stage stochastic programs and have the form

$$\begin{aligned} & Ax + D^i y^i \leq b^i \quad , i \in \{1, \ldots, k\} \\ & x \in \{0,1\}^p \\ & y^i \in \{0,1\}^q \quad , i \in \{1, \ldots, k\}. \end{aligned}$$

The variables x are the first-stage variables, and y^i are the second-stage variables for the ith scenario. Notice that these second-stage variables are different for each scenario, and are only coupled through the first-stage variables x. Thus, as long as the number of scenarios is reasonably large compared to dimensions of $x, y^1, \ldots, y^k$, these problems are to some extent almost-decomposable.

For our experiments we randomly generated instances of this form as follows: (1) the entries in A and the D^i's are independently and uniformly sampled from $\{-10, \ldots, 10\}$; (2) to guarantee feasibility, a 0/1 point is sampled uniformly at random from $\{0,1\}^{p+k \cdot q}$ and the right-hand sides b^i are set to be the smallest ones that make this point feasible. We generated 50 instances, 5 for each setting of parameters $k = \{5, 15, 25, 35, 45\}$, $p = \{10, 20\}$, $q = 10$.

We compared the two algorithms FPORIG and WFPBASE over these instances using ten dif-

ferent random seeds. A seed by seed comparison is reported in Table 3.1. In the tables, `#found` denotes the number of models for which a feasible solution was found, while `time` and `itr.` report the shifted geometric means [2] of running times and iterations, respectively.

Table 3.1: Aggregated results on two-stage stochastic models.

	# found		time (s)		itr.	
Seed	FPORIG	WFPBASE	FPORIG	WFPBASE	FPORIG	WFPBASE
1	28	**31**	4.12	**3.36**	124.43	**76.02**
2	26	**35**	4.06	**3.17**	122.51	**82.85**
3	25	**37**	4.00	**3.02**	117.74	**72.50**
4	26	**36**	4.28	**3.40**	119.82	**75.17**
5	25	**31**	4.20	**3.44**	124.41	**81.66**
6	26	**35**	3.98	**3.56**	122.74	**79.73**
7	25	**27**	4.22	**3.98**	126.77	**91.59**
8	28	**38**	3.82	**3.10**	112.91	**73.92**
9	25	**31**	4.22	**3.67**	117.61	**83.46**
10	25	**32**	4.12	**3.57**	116.92	**88.23**

Notice that WFPBASE performed substantially better than FPORIG, in agreement with our theoretical results. Using the walkSAT-based perturbation the average number of successful instances increased by 28%, while average runtime was reduced by 17% and average number of iterations was reduced by 33%.

MIPLIB 2010. We also compared the algorithms on a subset of models from MIPLIB 2010 [65]. The subset is defined by the models for which at least one of the two algorithms took more than 20 iterations to find a feasible solution (if any); the remaining models are basically too easy and not useful for comparing the two perturbation procedures. We are thus left with a subset of 82 models. Again we compared the two algorithms using ten different random seeds. A seed by seed comparison is reported in Table 3.2.

Even though the improvement in this heterogeneous testbed was less dramatic as in the two-

stage stochastic models, as expected, WFPBASE still consistently dominates FPORIG: it can find more solutions in 7 out 10 cases (in the remaining 3 cases it is a tie), taking always less time and almost always fewer iterations. On average over the seeds, WFPBASE increased the number of successfully solved instances by 6%, reduced by the computation time by 8.4% and reduced the number of iterations by 5.9%.

In conclusion, given that the suggested modification is very simple to implement, and appears to dominate FPORIG consistently, it suggests it is a good idea to add it as a feature in all future feasibility pump codes.

Table 3.2: Aggregated results on MIPLIB2010.

	# found		time (s)		itr.	
Seed	FPORIG	WFPBASE	FPORIG	WFPBASE	FPORIG	WFPBASE
1	33	**34**	1070.35	**1068.09**	**103.38**	104.59
2	**34**	**34**	1073.03	**1004.84**	108.65	**104.05**
3	34	**39**	1125.44	**976.16**	107.10	**96.18**
4	34	**36**	1045.10	**976.31**	101.30	**96.24**
5	31	**32**	1033.60	**974.56**	96.67	**94.36**
6	**34**	**34**	974.47	**880.05**	99.61	**91.20**
7	33	**36**	972.96	**877.45**	102.39	**95.04**
8	29	**32**	1085.82	**1049.22**	104.63	**103.22**
9	**37**	**37**	1065.50	**937.19**	101.44	**91.73**
10	32	**37**	1096.99	**913.50**	103.01	**90.85**

CHAPTER 4

THE RATIO OF THE NUMBER INTEGRAL EXTREME POINTS TO THE TOTAL NUMBER OF EXTREME POINTS

4.1 Introduction

In this Chapter, we are interested in understanding the role that the sparsity of the constraint matrix plays in the integrality of extreme points. We want to understand if the ratio of the number of integral extreme points to the total number of extreme points of a polytope varies as a function of sparsity. Preliminary computational results showed that for a family of randomly generated polytopes this ratio decreased as a function of sparsity.

The Chapter is structured as follows. First, we introduce a model for random packing polytopes that we use to derive our results, notation and definitions. Second, we present the main result for this Chapter. Finally, we present the proof for this result including bounds for the number of (integral) extreme points.

4.2 Preliminaries

In this section, we define the model we use for generating random packing instances. Additionally, we present basic definitions and notation. Finally, we present a well known concentration inequality (in a convenient form) that we use in our analysis.

4.2.1 Generating packing instances

A setting for generating random packing instances was described in [40, 63]: where for $n, m, M \in \mathbb{N}$ we construct $P = \text{conv}\left(\left\{x \in \{0,1\}^n : A^j x \leq \frac{\sum_{i=1}^n A_i^j}{2}, \forall j \in [m]\right\}\right)$, where the A_i^js are chosen independently and uniformly in the set $\{0, 1, ..., M\}$.

In this work, we are interested in the linear relaxation of these packing instances (and not the convex hull of the feasible 0/1 points). The previous model can be understood as a particular case of the following.

For parameters $n,m,N,M \in \mathbb{Z}$ ($M \geq N$) and $p \in [0,1]$, we consider the random polytope $P(n,m,N,M,p)$ defined as

$$P(n,m,N,M,p) = \left\{ x \in [0,1]^n : A^j x \leq \frac{\sum_{i=1}^n A_i^j}{2}, \forall j \in [m] \right\},$$

where with probability p, each coefficient A_i^j is chosen independently and uniformly in the set $\{N,...,M\}$ and with probability $1-p$ it is 0. Note, that these polytopes correspond to packing instances as long as $N \geq 0$.

For the reminder of this chapter, we focus on the case where $N = M = 1$ and $p \in (0,1)$ (and thus omit N and M from our notation). Even more, since we are interested in *sparse* instances we will require p to be *small* (we specify exactly how small later). By sparsity or sparsity level, we refer to the number of non-zero coefficients per constraints.

For the instances described above, we have that $P(n,m,p)$ corresponds to all the points x in $\mathbb{R}^n$ satisfying the following constraints

$$\sum_{i=1}^{n} A_i^j x_i \leq B_j, \ \forall j = 1,\ldots,m,$$
$$x_i \leq 1, \ \forall i = 1,\ldots,n,$$
$$x_i \geq 0, \ \forall i = 1,\ldots,n,$$

where $A_i^j \sim Bernoulli(p)$ and for m, j positive integer, $j \leq m$, B_j denotes $\frac{\sum_{i=1}^n A_i^j}{2}$, i.e. the right hand side on the j-th constraint in our model.

4.2.2 Definitions for counting (integral) extreme points

We use the notation $[n]$ to denote the set $\{1,\dots,n\}$.

Definition 29. For n,m,k positive integers such that $k \leq n$ and $p \in (0,1)$ and the random polytope $P = P(n,m,p)$, we define the following random variables

1. N_{EP}: the number of extreme points in P.

2. N_{EP}^I: the number of 0/1 extreme points in P.

Note that is this setting N_{EP}^I corresponds to the number of feasible 0/1 points in $P(n,m,p)$ as, in this case, no integral point can be expressed as a non-trivial combination of extreme points.

Since we are interested in lower bounding the ratio of the number of integral extreme points to the total number of extreme points and in particular, we will work (mainly) in finding lower bounds for N_{EP}^I and upper bound for N_{EP} (or their respective expectations).

4.2.3 Bernstein's inequality

Theorem 7 ([67], Appendix A.2)**.** Let $X_1, X_2, \dots, X_n$ be independent random variables such that $\mathbb{E}[X_i] = 0$ and $|X_i| \leq M \ \forall i$. Let $X = \sum_{i=1}^n X_i$ and $\sigma^2 = \mathrm{Var}(X) \leq U$. Then, for $w > 0$

$$\mathbb{P}(|X| > w) \leq \exp\left(-\min\left\{\frac{w^2}{4U}, \frac{3w}{4M}\right\}\right).$$

4.3 Statement of the main result

Theorem 8. Let n be a positive integer and $p \in (0, 2/3]$, then there exist a m such that

$$\mathbb{E}\left(\frac{N_{EP}^{I}}{N_{EP}}\right) \geq 2^{-\frac{1}{2}\exp\left(p\frac{\sqrt{n}}{1-p}\right)} 3^{-n} 2^{\lfloor \frac{n}{2} - n^{3/4} \rfloor - 1}.$$

Note that for the lower bound on the ratio stated in Theorem 8, we have that as instances become denser (i.e. p decreases) the lower bound increases.

4.4 Proofs

In order to prove Theorem 8 we use the following two results.

Proposition 1. Let n be a positive integer, let $p \in (0, 2/3]$ and let m be a positive integer such that $m \leq \frac{1}{2}\exp\left(p\frac{\sqrt{n}}{1-p}\right)$, then

$$\mathbb{E}\left(N_{EP}^{I}\right) \geq 2^{\lfloor \frac{n}{2} - n^{3/4} \rfloor - 1}.$$

Proposition 2. Let n, m be a positive integers, then

$$N_{EP} \leq 2^m 3^n.$$

Now, the proof of Theorem 8.

Proof. For Y, Z random variables and $L, U \in \mathbb{R}_+$ such that $\mathbb{E}(Y) \geq L$ and $0 < Z \leq U$ we have that

$$\mathbb{E}\left(\frac{Y}{Z}\right) \geq \mathbb{E}\left(\frac{Y}{U}\right) \geq \frac{L}{U}.$$

Then by Propositions 1 and 2 and for n, m, p as in the statements of both Propositions, we

obtain that

$$\mathbb{E}\left(\frac{N_{EP}^{I}}{N_{EP}}\right) \geq 2^{-m}3^{-n}2^{\lfloor \frac{n}{2}-n^{3/4}\rfloor -1}.$$

By taking m as its upper bound in Proposition 1, we obtain the result.

□

4.4.1 Proof of Proposition 1

We are interested with finding the number of integral extreme points of polytopes of the form

$$P = \left\{ x \in [0,1]^n : \sum_{i=1}^{n} A_j^i x_i \leq \frac{\sum_{i=1}^{n} A_j^i}{2} \quad \forall j = 1,\ldots,m \right\}.$$

Since for all i and for all j, A_j^i is i.i.d and $A_j^i \sim Bernoulli(p)$, we have that j is of the form: $\sum_{i\in I_j} x_i \leq \frac{|I_j|}{2}$, where $I_j = \left\{ i : A_j^i = 1 \right\}$. This implies that the right hand side is either an integer or a half integer.

Proof. Let $N_{IEP}^{I,k}$ denote the number of extreme points with exactly k non-zero coefficients. We can write the expression $\mathbb{E}\left(N_{EP}^{I}\right)$ in term of $\mathbb{E}\left(N_{EP}^{I,k}\right)$ for any $k \in [n]$ and then

$$\begin{aligned}
\mathbb{E}\left(N_{EP}^{I}\right) &= \sum_{k=0}^{n} \mathbb{E}\left(N_{EP}^{I,k}\right) \\
&= \sum_{k=0}^{n} \binom{n}{k} \mathbb{P}\left(X^k \text{ feasible}\right) \\
&= \sum_{k=0}^{n} \binom{n}{k} \mathbb{P}\left(A^j X^k \leq B_j \; \forall j \in [m]\right).
\end{aligned}$$

We can bound the expression $\mathbb{P}\left(A^j X^k \leq B_j \ \forall j \in [m]\right)$ by

$$\begin{aligned}
\mathbb{P}\left(A^j X^k \leq B_j \ \forall j \in [m]\right) &= 1 - \mathbb{P}\left(A^j X^k > B_j \text{ for some } j\right) \\
&\geq 1 - \sum_{j=1}^{m} \mathbb{P}\left(A^j X^k > B_j\right) \text{ (union bound)} \\
&= 1 - m\mathbb{P}\left(A^j X^k > B_j\right).
\end{aligned}$$

We remove the index j for simplicity. Now, we work on the expression $\mathbb{P}\left(AX^k > B\right)$. Since for all $i \in [n]$: A_i corresponds Bernoulli with parameter p and X^k corresponds to a random 0/1 point with exactly k ones, we have that

$$\begin{aligned}
\mathbb{P}\left(AX^k > B\right) &= \mathbb{P}\left(2\sum_{i=1}^{n} A_i X_i^k > \sum_{i=1}^{n} A_i\right) \\
&= \mathbb{P}\left(\sum_{i=1}^{n} A_i\left(2X_i^k - 1\right) > 0\right) \text{ (by independence)} \\
&= \mathbb{P}\left(\sum_{i:\ X_i^k=1} A_i - \sum_{i:\ X_i^k=0} A_i > 0\right) \text{ (equal in distribution)} \\
&= \mathbb{P}\left(\sum_{i=1}^{k} A_i > \sum_{i=k+1}^{n} A_i\right).
\end{aligned}$$

We use Bernstein's inequality (Theorem 7) to obtain the following bound. For $\mathbb{P}\left(\sum_{i=1}^{k} A_i > \sum_{i=k+1}^{n} A_i\right)$ we have that

$$\mathbb{P}\left(\sum_{i=1}^{k} A_i > \sum_{i=k+1}^{n} A_i\right) = \mathbb{P}\left(\sum_{i=1}^{k} A_i - kp - \sum_{i=k+1}^{n} A_i + (n-k)p > (n-2k)p\right).$$

For all $i \in [n]$ let X_i be defined as

$$X_i = \begin{cases} A_i - p & \text{if } i \leq k \\ p - A_i & \text{if } i \geq k+1. \end{cases}$$

Note that for every $i \in [n]$: $\mathbb{E}(X_i) = 0$, $|X_i| \leq 1$ and $\mathrm{Var}(X_i) = p(1-p)$. Let $S_n = \sum_{i=1}^n X_i$ and let $t = (n-2k)p$. Note that for $k < n/2$, we have that $t > 0$. Then, by Bernstein's inequality, we have

$$\begin{aligned} \mathbb{P}\left(\sum_{i=1}^k A_i > \sum_{i=k+1}^n A_i\right) &\leq \mathbb{P}\left(|S_n| > t\right) \\ &\leq \exp\left(-\min\left\{\frac{t^2}{4np(1-p)}, \frac{3t}{4}\right\}\right). \end{aligned}$$

For $p \leq 2/3$, we have that $\frac{t^2}{4np(1-p)} \leq \frac{3t}{4}$ and thus

$$\mathbb{P}\left(\sum_{i=1}^k A_i > \sum_{i=k+1}^n A_i\right) \leq \exp\left(-\frac{(n-2k)^2 p}{4n(1-p)}\right).$$

Let $k = \lfloor \frac{n}{2} - n^{3/4} \rfloor$, so that $\frac{(n-2k)^2 p}{4n(1-p)} \geq p\frac{\sqrt{n}}{1-p}$ and then for m as in the statement of Proposition 1 (such that $m \leq \frac{1}{2}\exp\left(\frac{(n-2k)^2 p}{4n(1-p)}\right)$), we have that

$$\mathbb{P}\left(\sum_{i=1}^k A_i > \sum_{i=k+1}^n A_i\right) \leq \frac{1}{2m}.$$

Since $\mathbb{E}\left(N_{EP}^I\right) \geq \sum_{i=0}^n \binom{n}{i}\left(1 - m\mathbb{P}\left(AX^i > B\right)\right)$ and for k as above, $\mathbb{P}\left(AX^k > B\right) \leq \frac{1}{2m}$, we have that

$$\mathbb{E}\left(N_{EP}^{I}\right) \geq \sum_{i=0}^{k} \binom{n}{i} \left(1 - m\mathbb{P}\left(AX^{i} > B\right)\right)$$
$$\geq \frac{1}{2}\left(\frac{n}{k}\right)^{k} \geq 2^{k-1}.$$

□

4.4.2 Proof of Proposition 2

Consider the problem in standard form

$$\sum_{i \in I_j} x_i + y_j = \lfloor \frac{|I_j|}{2} \rfloor \quad \forall j \in [m], \tag{4.1}$$

$$x_i + z_i = 1 \quad \forall i \in [n], \tag{4.2}$$

$$x, y, z \geq 0.$$

We can obtain upper bounds on N_{EP} by bounding the number of basic (feasible) solutions. For the problem in standard, a basis is defined by $n+m$ linearly independent columns of the constraint matrix. A trivial bound comes from the maximum number of possible bases $\binom{2n+m}{n+m} \leq \left(e\frac{2n+m}{n+m}\right)^{n+m}$.

Proof of Proposition 2. First note that a basic feasible solution satisfies the following.

Claim 30. A basic feasible solution cannot have more than m fractional x variables.

Proof. Clearly if $m \geq n$, then the claim is trivially true. If $m < n$ and for l a positive integer such that $m + l \leq n$, consider a feasible solution with $m + l$ fractional x variables. By constraint (2), we also have at least $m + l$ fractional z variables. Constraint (2) also implies that we must have at least $n - m - l$ additional non-zero x or z variables. Together with the fractional $2m + 2l$ non-zero x and z variables, the solution has at least $n + m + l$ non-zero variables and thus cannot be basic. □

Claim 30 implies that we cannot have more than m fractional z variables, since the number of fractional x and z variables are equal.

Consider a typical basic feasible solution:

1. Number of basic x variables: We assume there are exactly $i \leq n$ basic variables. Clearly there are $\binom{n}{i}$ ways of selecting such variables.

2. Number of basic z variables: Since there are $(n-i)$ non-basic x variables, we must have $(n-i)$ basic z variables. We assume there are exactly j additional basic z variables among the remaining $(n-(n-i))$ z variables. Claim 30 implies that $j \leq \min\{m,i\}$.

3. Number of basic y variables: The total number of basic x and z variables is $(n+j)$. Therefore there are exactly $(m-j)$ basic y variables.

Based on the above discussion an upper bound on the total number of basic feasible solution is

$$\sum_{i=0}^{n} \binom{n}{i} \sum_{j=0}^{\min\{m,i\}} \binom{i}{j} \binom{m}{m-j}.$$

We can directly bound $\binom{m}{m-j} \leq 2^m$. For $\sum_{j=0}^{\min\{m,i\}} \binom{i}{j}$, consider the two cases: (i) $m \leq i$: $\sum_{j=0}^{m} \binom{i}{j} \leq 2^i$. (ii) $m > i$: $\sum_{j=0}^{i} \binom{i}{j} = 2^i$. Finally, since $\sum_{i=0}^{n} \binom{n}{i} 2^i = 3^n$, we obtain the bound: $2^m 3^n$.

□

CHAPTER 5
THE STRENGTH OF MULTI-ROW AGGREGATION CUTS FOR SIGN-PATTERN INTEGER PROGRAMS

5.1 Introduction

In a recent paper [18], Bodur et al. studied the strength of *aggregation cuts*. An aggregation cut is obtained as follows: (i) By suitably weighing and adding the constraints of a given IP formulation one can obtain a relaxation which is defined by a single constraint together with variable bounds. (ii) All the valid inequalities for the integer hull of this knapsack-like set are called as aggregation cuts. The set obtained by adding all such aggregation cuts (for all possible aggregations) is called the aggregation closure. Such cuts are commonly used in practice by state of the art solvers [100, 101, 99, 78, 47]. A very special subclass of the aggregation cuts are the cuts where one just uses the original constraints of the problem (without non-trivial aggregation) as the knapsack-like relaxation. The weaker closure obtained from such cuts is called as the original 1-row closure [18]. The paper [18] shows that for packing and covering IPs, the aggregation closure can be *2-approximated* by the original 1-row closure. In contrast, they show that for general IPs, the aggregation closure can be arbitrarily stronger than the original 1-row closure.

The notion of aggregations cuts can be generalized to multi-row aggregation cuts. Essentially by using k different set of weights on the constraints of the problem one can produce a relaxation that involves k constraints together with variable bounds. We call the valid inequalities for the integer hull of such k-row relaxations as k-row or multi-row aggregation cuts. Analogous to the case of aggregation cuts, we can also define the notion of k-row aggregation closure and the original k-row closure. It is shown in [18] that for packing and covering IPs the k-row aggregation closure can be approximated by the original k-row closure within a multiplicative factor that depends only

on k.

Observe that for packing and covering IPs all the coefficients of all the variables in all the constraints have the same sign. Therefore when we aggregate constraints we are not able to "cancel" variables, i.e., the support of an aggregated constraint is exactly equal to the union of supports of the original constraints used for the aggregation. A natural conjecture for the fact that the (resp. multi-row) aggregation closure is well approximated by the original (resp. multi-row) 1-row closure for packing and covering problems, is the fact that such cancellations do not occur for these problems. Indeed one of the key ideas used to obtain good candidate aggregations in the heuristic described in [78] is to use aggregations that maximize the chances of a cancellation.

In order to study the effect of cancellations, we study the strength of aggregation closures vis-à-vis original row constraints for *sign-pattern IPs*. A sign-pattern IP is a problem of the form $\{x \in \mathbb{Z}^n_+ : Ax \leq b\}$ where a given variable has exactly the same sign in every constraint, i.e. for a given j, $A_{i,j}$ is either non-negative for all i or non-positive for all i. Thus aggregations do not create cancellations.

Our study produces surprising results. On the one hand we are able to show that the aggregation closure for such sign-pattern IPs is *2-approximated* by the original 1-row closure. On the other hand the multi-row aggregation closure cannot be well approximated by the original multi-row closure. Therefore these classes of integer programs show results that are in between packing and covering IPs on one side and general IPs on the other side.

The structure of the rest of the chapter is as follows. In Section 5.2 we provide definitions and statements of all our main results. In Section 5.3 we present the proofs for our results related to sign-pattern IPs.

5.2 Definitions and statement of results

5.2.1 Definitions

For an integer n, we use the notation $[n]$ to describe the set $\{1,\dots,n\}$ and for $k \leq n$ non-negative integer, we use the notation $\binom{[n]}{k}$ to describe all subsets of $[n]$ of size k. For $i \in [n]$, we denote by e_i the i^{th} vector of the standard basis of $\mathbb{R}^n$. The convex hull of a set S is denoted as $\text{conv}(S)$. For a set $S \subset \mathbb{R}^n$ and a positive scalar α we define $\alpha S := \{\alpha u \mid u \in S\}$. We use $\mathscr{C}(P)$ and P^I to denote the CG closure and the convex hull of integer feasible solutions of P respectively.

Sign-pattern IPs

Definition 31. Let n be an integer, let $J^+, J^- \subset [n]$ s.t. $J^+ \cap J^- = \emptyset$ and $J^+ \cup J^- = [n]$. We call a polyhedron P with m constraints a (J^+, J^-) sign-pattern polyhedron if it is of the form

$$P = \left\{ x \in \mathbb{R}^n_+ : \sum_{j \in J^+} a^i_j x_j - \sum_{j \in J^-} a^i_j x_j \leq b^i \quad \forall i \in [m] \right\},$$

where $a^i_j, b^i \geq 0,\ \forall i = 1,\dots,m, \forall j = 1,\dots,n$. Additionally, we require $a^i_j \leq b^i\ \forall j \in J^+,\ \forall i = 1,\dots,m$.

Definition 32. For points $x, y \in \mathbb{R}^n$, we say that y is (J^+, J^-) sign-pattern dominated by x if $y_j \leq x_j\ \forall j \in J^+$ and $y_j \geq x_j\ \forall j \in J^-$.

Clearly for P a (J^+, J^-) *sign-pattern* polyhedron, if $x \in P$ and $y \geq 0$ is (J^+, J^-) *sign-pattern* dominated by x then, $y \in P$. Similarly, for $x \in P$ we have that $\bar{x} \in \mathbb{Z}^n_+$ defined as

$$\bar{x}_j = \begin{cases} \lfloor x_j \rfloor & \text{if } j \in J^+ \\ \lceil x_j \rceil & \text{if } j \in J^- \end{cases}$$

is in P^I.

Definition 33. Given two polyhedra $P \supset Q$ contained in $\mathbb{R}^n_+$ and a positive scalar α, we say that Q is an α-approximation of P if

$$P \subset \alpha Q.$$

Closures

Given a polyhedron P, we are interested in cuts for the pure integer set $P \cap \mathbb{Z}^n$.

Definition 34. For $P = \{x \geq 0 : Ax \leq b\}$, $k \geq 1$ integer, and $\lambda_1, \ldots, \lambda_k \in R^m_+$ let

$$P(\lambda_1, \ldots, \lambda_k) = \{x \geq 0 : \lambda_1 Ax \leq \lambda_1 b, \ldots, \lambda_k Ax \leq \lambda_k b\}.$$
$$P^I(\lambda_1, \ldots, \lambda_k) = \text{conv}\left(\left\{x \in \mathbb{Z}^n_+ : \lambda_1 Ax \leq \lambda_1 b, \ldots, \lambda_k Ax \leq \lambda_k b\right\}\right).$$

Definition 35. Given a polyhedron $P = \left\{x \in \mathbb{R}^n_+ \mid Ax \leq b\right\}$, we define its aggregation closure $\mathscr{A}(P)$ as

$$\mathscr{A}(P) = \bigcap_{\lambda \in R^m_+} P^I(\lambda).$$

We can generalize the *aggregation-closure* to consider simultaneously k aggregations, where $k \in \mathbb{Z}$ and $k \geq 1$. More precisely, for a polyhedron P the *k-aggregation* closure is defined as

$$\mathscr{A}_k(P) := \bigcap_{\lambda_1, \ldots, \lambda_k \in \mathbb{R}^m_+} P^I(\lambda_1, \ldots, \lambda_k).$$

Similarly, the original 1-row closure *1-*$\mathscr{A}(P)$ is defined as

$$1\text{-}\mathscr{A}(P) := \bigcap_{i\in[n]} P^I(e_i).$$

We can generalize the original 1-row closure, to an original k-row closure k-$\mathscr{A}(P)$. More precisely, for a polyhedron P the original k-row closure is defined as

$$k\text{-}\mathscr{A}(P) := \bigcap_{K\in\binom{[m]}{k}} P^I(e_{i_1},\ldots,e_{i_k}).$$

Where, in the last equation for each $K \in \binom{[m]}{k}$: $\{i_1,\ldots,i_k\} = K$.

5.2.2 Statement of results

The first result compares the aggregation closure with the LP relaxation of a (J^+,J^-) *sign-pattern* polyhedron.

Theorem 9. For a (J^+,J^-) sign-pattern polyhedron P, we have that $\mathscr{A}(P)$ can be 2-approximated by P (and thus by 1-$\mathscr{A}(P)$).

This results is equivalent to the one obtained in [18] for the case of packing problems. Next we show that, in general, for (J^+,J^-) *sign-pattern* IPs, the *aggregation-closure* does not do a good job at approximating the convex hull.

Theorem 10. There is a family of (J^+,J^-) sign-pattern polyhedra with 2 constraints for which $\mathscr{A}$ is an arbitrarily bad approximation to $\mathscr{A}_2$ (and thus to the convex hull), i.e. for each $\alpha > 0$, there is a (J^+,J^-) sign-pattern polyhedron P such that $\mathscr{A}(P)$ is not an α-approximation of $\mathscr{A}_2(P)$.

For such a family of polyhedra, we have that $\frac{z^{LP}}{z^{IP}}$ is arbitrarily large at the same time that $\frac{z^{\mathscr{A}}}{z^{IP}}$ is also arbitrarily large. The previous results show how for (J^+,J^-) *sign-pattern* polyhedra,

aggregating multiple constraints can have significant benefits. This is different than for the case of packing/covering problems (where the improvement is bounded).

The next result shows that the *aggregation-closure* considering simultaneously 2 aggregations ($\mathscr{A}_2$) can be arbitrarily stronger than the original 2-row closure (2-$\mathscr{A}$).

Theorem 11. There is a family of (J^+, J^-) sign-pattern polyhedra with 4 constraints for which *2-*$\mathscr{A}$ is an arbitrarily bad approximation to $\mathscr{A}_2$ (and thus to the convex hull), i.e. for each $\alpha > 0$, there is a (J^+, J^-) sign-pattern polyhedron P such that 2-$\mathscr{A}(P)$ is not an α-approximation of $\mathscr{A}_2(P)$.

The previous results establish α-approximation comparison between the sets P and $\mathscr{A}(P)$, $\mathscr{A}(P)$ and $\mathscr{A}_2(P)$, and 2-$\mathscr{A}(P)$ and $\mathscr{A}_2(P)$. These results are combined with the results obtained in [18] regarding packing and covering problems in Table 5.1. For polyhedra of each type with m constraints we have the following upper bounds for α.

Table 5.1: Containment relation for different classes of polyhedra.

	Packing (from [18])	Covering (from [18])	Sign-pattern (this chapter)
$P \subset \alpha\mathscr{A}(P)$	2	2	2
$\mathscr{A}(P) \subset \alpha\mathscr{A}_2(P)$	$3 \quad \text{if } m \geq 2$	$3 \quad \text{if } m \geq 2$	$\infty \quad \text{if } m \geq 2$
2-$\mathscr{A}(P) \subset \alpha\mathscr{A}_2(P)$	$\begin{cases} 1 & \text{if } m = 2 \\ 3 & \text{if } m \geq 3 \end{cases}$	$\begin{cases} 1 & \text{if } m = 2 \\ 3 & \text{if } m \geq 3 \end{cases}$	$\begin{cases} 1 & \text{if } m = 2 \\ ? & \text{if } m = 3 \\ \infty & \text{if } m \geq 4 \end{cases}$

5.3 Proofs

5.3.1 Proof of Theorem 9

First, we need some general properties for (J^+, J^-) *sign-pattern* LPs.

Proposition 3. Consider a (J^+, J^-) *sign-pattern* polyhedron defined by one constraint $P = \left\{x \geq 0 : \sum_{j \in J^+} a_j x_j - \sum_{j \in J^-} a_j x_j \leq b\right\}$ and let c be a vector with the same sign-pattern as a, i.e. $c_j \geq 0 \; \forall j \in J^+$ and $c_j \leq 0 \; \forall j \in J^-$.

1. $z^{LP} = \max_{x \in P} c^\top x$ is bounded if and only if $\max_{j \in J^+} \frac{c_j}{a_j} \leq \min_{j \in J^-} \frac{c_j}{a_j}$.
2. If z^{LP} is bounded, there exists an optimal solution x^{LP} such that $x_j^{LP} = c_j/a_j$ for $j \in \operatorname{argmax}_{j \in J^+} c_j/a_j$ and $x_k^{LP} = 0$ for $k \in [n] \setminus \{j\}$.
3. If z^{LP} is bounded, then $z^{LP} \leq 2z^{IP}$, where $z^{IP} = \max_{x \in P^I} c^\top x$.

Proof. Clearly $0 \in P$, thus the (J^+, J^-) *sign-pattern* LP cannot be infeasible. Consider its dual

$$\begin{aligned} \min \quad & by \\ s.t. \quad & a_j y \geq c_j \quad \forall j \in J^+ \\ & a_j y \leq c_j \quad \forall j \in J^- \\ & y \geq 0 \end{aligned}$$

which is feasible if and only if $\max_{j \in J^+} \frac{c_j}{a_j} \leq \min_{j \in J^-} \frac{c_j}{a_j}$.

If z^{LP} is bounded, there exists an optimal solution that is an extreme point. Since the problem is defined by a single non-trivial constraint, each extreme point can have at most one non-zero coefficient, thus a maximizer over the set of extreme points must be of the form described previously.

Additionally, if the problem is bounded, there exists an extreme point x^{LP} of the form $x_{j^*}^{LP} = b/a_{j^*} \geq 1$ for some $j^* \in J^+$ and $x_j^{LP} = 0 \; \forall j \in [n] \setminus \{j^*\}$ that is optimal. Clearly $\lfloor x^{LP} \rfloor \in P^I$ (since it is non-negative and (J^+, J^-) *sign-pattern* dominated by x^{LP}), thus $\frac{z^{LP}}{z^{IP}} \leq \frac{b/a_{j^*}}{\lfloor b/a_{j^*} \rfloor}$. Finally, since P is a (J^+, J^-) *sign-pattern* polyhedron $b/a_{j^*} \geq 1$ and thus $\frac{z^{LP}}{z^{IP}} \leq 2$. □

In order to prove Theorem 9, we need some preliminary results.

Proposition 4. Consider a (J^+, J^-) *sign-pattern* polyhedron P, then P^I is also a (J^+, J^-) sign-pattern polyhedron.

Proof. First, since P is a (J^+, J^-) sign-pattern polyhedron, $0, e_1, \ldots, e_n \in P$, then P^I is a non-empty polyhedron. We show that for every facet $ax \leq b$, we must have $a_j \geq 0$ for $j \in J^+$ and $a_j \leq 0$ for $j \in J^-$.

For $j \in J^-$. Note that the recession cone of P^I is the same as the recession cone of P. Then for every facet $ax \leq b$: $a_j \leq 0$ (otherwise e_j would not be in the recession cone of P^I).

For $j \in J^+$, assume that there exists a facet $ax \leq b$ s.t. $a_j < 0$. Consider $a' = a - a_j e_j$ (we zero out the j-th component), if $a'x \leq b$ is valid, it corresponds to a stronger constraint than $ax \leq b$. In order to show that is valid, assume that there exists $x \in P \cap \mathbb{Z}^n$ s.t. $a'x > b$. Clearly $x_j \geq 1$ (since otherwise $a'x = ax \leq b$). Consider $x' = x - x_j e_j$ (clearly $x' \in P \cap \mathbb{Z}^n$), then $b < a'x = ax' \leq b$, a contradiction. □

Proposition 5. Let P be a (J^+, J^-) sign-pattern polyhedron defined by one constraint, then $P \subset 2P^I$.

Proof. Assume that this is not the case and let $x' \in \frac{1}{2}P$ s.t. $x' \notin P^I$. Since P^I is a (J^+, J^-) *sign-pattern* polyhedron, each facet defining inequality $ax \leq b$ satisfies $a_j \geq 0 \; \forall j \in J^+$ and $a_j \leq 0 \; \forall j \in J^-$. Since $x' \notin P^I$, for one of these facets, we have: $ax' > b$. Now, if we consider a as an objective: $\max_{x \in P^I} ax \leq b$ and thus defines a bounded problem. Since the IP is feasible and bounded and P is defined by rational data, the LP is also bounded (see [23]). By Proposition 3, $ax' \leq \frac{1}{2} z^{LP} \leq z^I \leq b$ a contradiction. □

Observation 1. Let $\phi : \mathbb{R}^n \to \mathbb{R}^n$ be a bijective map, let $\{S_i\}_{i \in I}$ be a collection of subsets in R^n and let $\phi(S) := \{\phi(x) : x \in S\}$. Then $\phi\left(\cap_{i \in I} S_i\right) = \cap_{i \in I} \phi(S_i)$.

Now, we prove Theorem 9.

Proof. By definition, we have that $P \subset P(\lambda), \; \forall \lambda \in R^m_+$. Since $P(\lambda)$ corresponds to a (J^+, J^-) *sign-pattern* polyhedron defined by one constraint, by Proposition 5, we have that $P(\lambda) \subset 2P^I(\lambda)$

and thus $P \subset 2P^I(\lambda),\ \forall \lambda \in R^m_+$. Then, taking intersection over all $\lambda \in R^m_+$ and by Observation 1 we have

$$\begin{aligned} P &\subset \bigcap_{\lambda \in R^m_+} 2P^I(\lambda) \\ &= 2 \bigcap_{\lambda \in R^m_+} P^I(\lambda) \\ &= 2\mathscr{A}(P) \end{aligned}$$

Since $1\mathscr{A}(P)$ is contained in P, we have that $1\mathscr{A}(P)$ corresponds to a *2-approximation* of $\mathscr{A}(P)$.

□

5.3.2 Proof of Theorem 10

In order to prove Theorem 10, consider the following family of (J^+, J^-) *sign-pattern* polyhedra and $M \geq 2$ integer

$$\begin{aligned} \max \quad & x_1 - (M-1)x_2 \\ s.t. \quad & x_1 - M(M-1)x_2 \leq 1 \\ & x_1 \leq M+1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Note that the integral solutions of interest are $(1,0)$ and $(\bar{x}_1, \bar{x}_2)$, where $\bar{x}_1, \bar{x}_2 \in \mathbb{Z}_+$, $\bar{x}_1 \leq M+1$ and $\bar{x}_2 \geq 1$ (clearly in terms of objective function value, all of the latter solutions are dominated by $(M+1,1)$). Now the optimal IP value z^{IP} corresponds to $2 = M+1-(M-1)$ (since the value of

$(M+1,1)$ is greater than the value of $(1,0)$). While for the *LP* relaxation, we have that the point $\left(M+1,\frac{1}{M-1}\right)$ is optimal and it's value is: M. Thus, in this case we have that: $\frac{z^{LP}}{z^{IP}} = \frac{M}{2}$.

See example of feasible set in Figure 5.1.

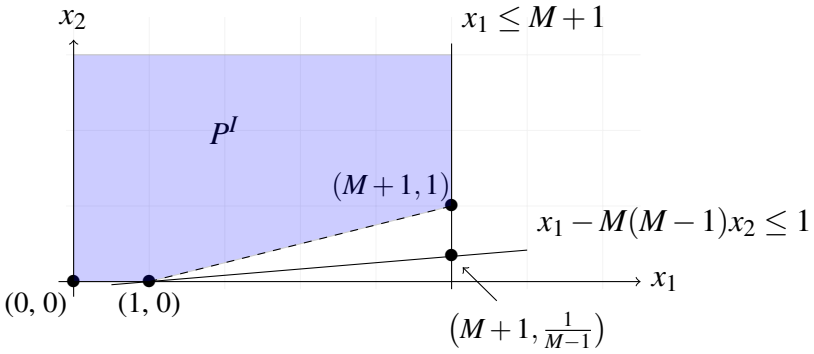

Figure 5.1: Feasible region P^I.

Trivially, since we have only two constraints $\mathscr{A}_2(P) = P^I$.

Now, the proof of Theorem 10.

Proof. It follows from Proposition 5 that for any (J^+,J^-) *sign-pattern* polyhedron $\frac{z^{LP}}{z^{\mathscr{A}}} \in [1,2]$. For the family of (J^+,J^-) *sign-pattern* polyhedra previously described we have that $z^{IP} = z^{\mathscr{A}_2}$ and therefore

$$\frac{z^{\mathscr{A}}}{z^{\mathscr{A}_2}} = \frac{z^{LP}}{z^{IP}} \cdot \frac{z^{\mathscr{A}}}{z^{LP}} \geq \frac{M}{2}\frac{1}{2}.$$

Since M can be arbitrarily large, $\mathscr{A}(P)$ cannot be an α-*approximation* of $\mathscr{A}_2(P)$ for any finite value of α.

□

5.3.3 Proof of Theorem 11

In order to show the proof for Theorem 11, we introduce the following family of instances.

For $M \geq 2$ an even integer:

$$
\begin{aligned}
\max \quad & x_1 - \frac{M}{2}x_2 - \frac{M}{2}x_3 - \frac{M}{2}x_4 & & \\
s.t. \quad & x_1 - Mx_2 - Mx_3 \leq 1 & & (5.1)\\
& x_1 - Mx_2 - Mx_4 \leq 1 & & (5.2)\\
& x_1 - Mx_3 - Mx_4 \leq 1 & & (5.3)\\
& x_1 \leq M+1 & & (5.4)\\
& x_1, x_2, x_3, x_4 \geq 0. & &
\end{aligned}
$$

It is not difficult to see that $z^{IP} = 1$. The point $(1,0,0,0)$ has value 1 and for any feasible solution such that $x_1 \geq 2$, we must have $x_2 + x_3 + x_4 \geq 2$ (from constraints $(1)-(3)$) thus the value in this case is at most 1. Additionally, $z^{LP} = \frac{M}{4} + 1$ since the feasible point $\left(M+1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$ achieves that value and by aggregating constraints $(1)-(4)$ and dividing by 4, we obtain the valid inequality: $x_1 - \frac{M}{2}x_2 - \frac{M}{2}x_3 - \frac{M}{2}x_4 \leq \frac{M}{4} + 1$.

Now, the proof of Theorem 11.

Proof. We show that for the family of problems previously described $\frac{z^{2\text{-}\mathscr{A}}}{z^{\mathscr{A}_2}} = \frac{M}{4} + 1$ and thus *2-*$\mathscr{A}$ can be an arbitrarily bad approximation of $\mathscr{A}_2$.

First, we show that $z^{2\text{-}\mathscr{A}} = \frac{M}{4} + 1$ by showing that the optimal point for the LP relaxation is also feasible for *2-*$\mathscr{A}$. To conclude the proof, we show that $z^{\mathscr{A}_2} = z^I$ by providing an upper bound on $z^{\mathscr{A}_2}$ coming from a particular selection of multipliers.

In the case of *2-*$\mathscr{A}$, we verify that $(x, y_1, y_2, y_3) = \left(M+1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$ is in $P^I(e_{i_1}, e_{i_2})$, where

$\{e_{i_1}, e_{i_2}\} = K$ corresponds to an arbitrary selection of two constraints, i.e. any $K \in \binom{[4]}{2}$. Let S'_K be those variables with non positive coefficients that are present in the inequalities in K. If constraint (4) is in K, let l denote the smallest index in S'_K. Otherwise, let l denote the index of the variable (out of $\{x_2, x_3, x_4\}$) that is present in both constraints $\{e_{i_1}, e_{i_2}\}$ (note that there must always be one such index).

Then it can be verified that the points $(M+1,0,0,0) + e_l^\top$ and $(M+1,1,1,1) - e_l^\top$ are in $P^I(e_{i_1}, e_{i_2})$ and so is the midpoint $\left(M+1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$. The latter point has value: $M+1-\frac{M}{2}\frac{3}{2} = \frac{M}{4}+1$, thus, in terms of objective function value, *2-*$\mathscr{A}$ does not provide any extra improvement when compared to the LP relaxation.

Now, in order to show that $z^{\mathscr{A}_2} = 1$. Since $P^I \subset \mathscr{A}_2(P)$, we have that $z^{\mathscr{A}_2} \geq 1$, and by the definition of $\mathscr{A}_2$, we have that

$$z^{\mathscr{A}_2} \leq z^{P^I(\lambda,\mu)} \quad \forall \lambda, \mu \in \mathbb{R}^4_+.$$

Consider $\bar{\lambda} = (1,1,0,0)$, $\bar{\mu} = (0,0,1,1)$ and $c^\top = \left(1, -\frac{M}{2}, -\frac{M}{2}, -\frac{M}{2}\right)$, then the problem $\max\left\{c^\top x : \; x \in P^I(\bar{\lambda}, \bar{\mu})\right\}$ corresponds to

$$\begin{aligned} \max \quad & x_1 - \frac{M}{2}x_2 - \frac{M}{2}x_3 - \frac{M}{2}x_4 \\ s.t. \quad & 2x_1 - 2Mx_2 - Mx_3 - Mx_4 \leq 2 \\ & 2x_1 \qquad\qquad - Mx_3 - Mx_4 \leq M+2 \\ & x_1, x_2, x_3, x_4 \geq 0. \end{aligned}$$

Note that x_3 and x_4 have the same coefficients in the objective and in every constraint, thus by dropping x_4 and rearranging the constraints we obtain a problem with the same optimal objective

function value

$$\begin{aligned} \max \quad & x_1 - \frac{M}{2}x_2 - \frac{M}{2}x_3 \\ s.t. \quad & x_1 \leq 1 \qquad\qquad + Mx_2 + \frac{M}{2}x_3 \\ & x_1 \leq 1 + \frac{M}{2} \qquad\qquad + \frac{M}{2}x_3 \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

Now, a simple case analysis for each value of x_1, shows that $z^{PI(\bar{\lambda},\bar{\mu})} = 1$. Let z denote the best objective function value for each case:

- Case $(x_1 = 0)$: it is easy to see that $z \leq 0$.
- Case $(x_1 = 1)$: it is easy to see that $z \leq 1$ (in fact, it is equal to 1 when $x_2 = x_3 = 0$).
- Case $\left(2 \leq x_1 \leq \frac{M}{2} + 1\right)$: note that in this case the first constraint forces either x_2 or x_3 to be at least one. Thus $z \leq \frac{M}{2} + 1 - \frac{M}{2} = 1$.
- Case $\left(k\frac{M}{2} + 2 \leq x_1 \leq (k+1)\frac{M}{2} + 1, \text{ for } k \geq 1 \text{ integer}\right)$: similar to the previous case. Now, since $x_1 \geq k\frac{M}{2} + 2$, the second constraint forces $x_3 \geq k$, this together with the first constraint forces $x_2 + x_3 \geq k + 1$. Then, $z \leq (k+1)\frac{M}{2} + 1 - \frac{M}{2} - k\frac{M}{2} = 1$.

□

Appendices

APPENDIX A

TECHNICAL PROOFS CHAPTER 2

A.1 Proof Gap Equivalence

Proof of Lemma 1. It is not difficult to see that for $P^k = P$, the lemma holds, since $d(P,P^k) = gap_P^k(c) = 0\ \forall c : ||c|| = 1$. When, $P^k \neq P$, we have that $d(P,P^k) = d(x^0,y^0) > 0$ is attained at $x^0 \in ext(P^k)$ and $y^0 \in P$, the orthogonal projection of x^0 P (see [40]). Thus, $y^0 \in F = \{z \in \mathbb{R}^n : az = b\} \cap P$, a face of P such that $a = (x^0 - y^0)$, $b = (x^0 - y^0)y^0$ and $P \subseteq \{z \in \mathbb{R}^n : az \leq b\}$. Let $c = (x^0 - y^0)/||x^0 - y^0||$, we have: $\max_{x \in P} cx = cy^0$. On the other hand, $\max_{z \in P^k} cz = cx^0$, since otherwise, if $\exists \bar{x} \in P^k$ with $c\bar{x} > cx^0$, let $\bar{y}$ denote the orthogonal projection of $\bar{x}$ onto $\{z \in \mathbb{R}^n : az = b\}$. Then, for all $z \in P$ we have $d(\bar{x},z) \geq d(\bar{x},\bar{y}) > d(x^0,y^0)$ (the last inequality follows from the fact that $c\bar{x} > cx^0$, $c\bar{y} = cy^0$ and $\bar{x} - \bar{y}/||\bar{x} - \bar{y}|| = c$), a contradiction. So, we obtain

$$\begin{aligned} d(P,P^k) &= ||x^0 - y^0|| = c(x^0 - y^0) \\ &= \max_{x \in P^k} cx - \max_{x \in P} cx = gap_P^k(c). \end{aligned}$$

Now, assume by contradiction that $\exists c'$ s.t. $gap_P^k(c') > gap_P^k(c)$ and $||c'|| = 1$. Let $x' \in P^k, y' \in P$ denote the points at which $gap_P^k(c')$ is attained. Using the definition of $d(P,P^k)$ and the relation between c and c'

$$\begin{aligned} d(P,P^k) \geq ||x' - y'|| &= \frac{(x' - y')}{||x' - y'||}(x' - y') \\ &= \max_{c:||c||=1} c(x' - y') \geq c'(x' - y') \\ &= gap_P^k(c') > gap_P^k(c) = d(P,P^k), \end{aligned}$$

a contradiction. Thus, we must have $d(P,P^k) = \max_{c:||c||=1} gap_P^k(c)$. □

A.2 Proof Description Equivalence

A polytope $P \subseteq \mathbb{R}^n_+$ is called *down-monotone* if whenever $x \in P$ and $0 \leq y \leq x$, we have $y \in P$. We begin with some preliminary results about the symmetrization we employ.

Lemma 36. For a down-monotone polytope $P \subseteq \mathbb{R}^n_+$ we have $\overline{P} = \bigcup_{I\subseteq[n]} P^I$.

Proof. It is sufficient to prove that the set $\bigcup_{I\subseteq[n]} P^I$ is convex. For that, consider $y^1, y^2 \in \bigcup_{I\subseteq[n]} P^I$; by definition, let $x^1, x^2 \in P$ be such that there are sets I_1, I_2 giving $(x^1)^{I_1} = y^1$ and $(x^2)^{I_2} = y^2$. For any $\lambda \in [0,1]$, consider $y = \lambda y^1 + (1-\lambda)y^2$; we show $y \in \bigcup_{I\subseteq[n]} P^I$.

By construction we have:

$$y_i = \begin{cases} \lambda x_i^1 + (1-\lambda)x_i^2 & i \in I^1 \cap I^2 \\ \lambda x_i^1 - (1-\lambda)x_i^2 & i \in I^1 \backslash I^2 \\ -\lambda x_i^1 + (1-\lambda)x_i^2 & i \in I^2 \backslash I^1 \\ -\lambda x_i^1 - (1-\lambda)x_i^2 & i \in [n] \backslash I^1 \cup I^2 \end{cases}$$

Now, let $\bar{I} = \{i \in [n] : y_i \geq 0\}$. Then define $x := y^{\bar{I}}$, which is nonnegative by construction. By non-negativity of the x^i's, we have $|\lambda x_i^1 - (1-\lambda)x_i^2| \leq \lambda x_i^1 + (1-\lambda)x_i^2$ and $|-\lambda x_i^1 + (1-\lambda)x_i^2| \leq \lambda x_i^1 + (1-\lambda)x_i^2$, thus $x \leq x^1 + (1-\lambda)x^2 \in P$. Since P is down-monotone, we have that x belongs to P. Since $y = x^{\bar{I}}$, this gives that y belongs to $\bigcup_{I\subseteq[n]} P^I$, concluding the proof. □

Lemma 37. For a down-monotone polytope $P \subseteq \mathbb{R}^n_+$ we have $(\overline{P})^k = \overline{P^k}$.

Proof. We break the proof into a couple of claims.

Claim 38. $(\overline{P})^k \cap \mathbb{R}^n_+ = P^k = \overline{P^k} \cap \mathbb{R}^n_+$.

Proof. For the first equality, notice that since $\overline{P}^k \supseteq P^k$ it suffices to prove $(\overline{P})^k \cap \mathbb{R}^n_+ \subseteq P^k$. For any $x \in (\overline{P})^k \cap \mathbb{R}^n_+$ and $I \subseteq \binom{[n]}{k}$, there exists $y \in \overline{P}$ such that $y|_I = x|_I$. Moreover, using the fact that

$x \geq 0$ and the symmetry in the definition of $\overline{P}$, there is one such y which is non-negative, and hence $y \in P$. But again using $x|_I = y|_I$, we get that $x \in P^k$.

For the second equality, since P is down-monotone we have that P^k is down monotone. Therefore, from Lemma 36 $\overline{P^k} = \bigcup_{I \subseteq [n]} (P^k)^I$, which implies $\overline{P^k} \cap \mathbb{R}^n_+ = P^k$. $\diamond$

Claim 39. Consider $z \in (\overline{P})^k$ and let $y = z^I$ for some $I \subseteq [n]$. Then $y \in (\overline{P})^k$.

Proof. First note that it is straight forward to verify that if $\alpha x \leq b$ is a valid inequality for $\overline{P}$, then for every $I \subseteq [n]$ the inequality $a^I x \leq b$ is also a valid inequality for $\overline{P}$. Then the point y must belong to $(\overline{P})^k$, since otherwise y would be separated by some k-sparse cut $ax \leq b$ and so z would be separated by the k-sparse cut $a^I x \leq b$. $\diamond$

Now we conclude the proof of the lemma. For the direction $(\overline{P})^k \subseteq \overline{P^k}$, let $z \in (\overline{P})^k$ and let $I = \{i \in [n] : z_i \geq 0\}$ and $x = z^I$. Then using Claim 39 we get $x \in (\overline{P})^k \cap \mathbb{R}^n_+$. Thus by Claim 38 we have $x \in P^k$ and hence $z \in \overline{P^k}$, concluding this part of the proof. For the direction $\overline{P^k} \subseteq (\overline{P})^k$, let $z \in \overline{P^k}$. Let $I = \{i \in [n] : z_i \geq 0\}$ and $x = z^I$. The point $x \in \overline{P^k} \cap \mathbb{R}^n_+$. Thus, by Claim 38 we have that $x \in (\overline{P})^k \cap \mathbb{R}^n_+$. However, by Claim 39 we have that $z \in (\overline{P})^k$. This concludes the proof. $\square$

From [40], it is straightforward to see that for $t \leq n$ the k-sparse apporximation of $\mathscr{P}_{t,n}$ corresponds to:

$$\mathscr{P}^k_{t,n} = \left\{ x \in [0,1]^n : \sum_{i \in I} x_i \leq t, \ \ \forall I \in \binom{[n]}{k} \right\}.$$

The next result together with Lemma 37 implies Lemma 4.

Proposition 40. Consider non-negative vectors $a^1, \ldots, a^m \in \mathbb{R}^n_+$ and define the polyhedron $P = \{x \in \mathbb{R}^n_+ \,|\, a^i x \leq b_i \ \forall i \in [m]\}$. Then $\overline{P} = \{x \,|\, (a^i)^I x \leq b_i \ \ \forall I \subseteq [n], \ \forall i \in [m]\}$.

Proof. ($\overline{P} \subseteq \{x \,|\, (a^i)^I x \leq b_i \ \forall I \subseteq [n], \ \forall i \in [m]\}$) Consider $z \in \overline{P}$ and define $I = \{i \in [n] : z_i \geq 0\}$. Then $z^I \in \overline{P} \cap \mathbb{R}^n_+$ and thus $z^I \in P$ (from Lemma 36). Now observe that $(a^i)^I z = a_i z^I \leq b_i$ where the last inequality follows from that fact that $z^I \in P$. This concludes this part of the proof.

$(\{x \,|\, (a^i)^I x \leq b_i \ \forall I \subseteq [n],\ \forall i \in [m]\} \subseteq \overline{P})$ Consider $z \in \{x \,|\, (a^i)^I x \leq b_i \ \forall I \subseteq [n],\ \forall i \in [m]\}$. Let $I = \{i \in [n] : z_i \geq 0\}$. Then observe that $a_i z^I = (a^i)^I z \leq b_i$ for all $i \in [m]$ and $z^I \in \mathbb{R}^n_+$. Thus, $z^I \in P$ or equivalently, $z \in \overline{P}$. This concludes the proof. $\square$

APPENDIX B

TECHNICAL PROOFS CHAPTER 3

B.1 Minimal projected certificates can be found in polynomial time

Consider the following LP:

$$\begin{aligned} \max \quad & \lambda A\bar{x} - \lambda b \\ \text{s.t.} \quad & B^T\lambda = 0 \\ & e^T\lambda = 1 \\ & \lambda \geq 0, \end{aligned}$$

where e is the all-ones vector. Since we assumed a projected certificate exists, this LP is feasible and has strictly positive optimal value.

An optimal extreme point solution provides a projected certificate that can be computed in polynomial time [91]; we just need to verify that there cannot exist a projected certificate with smaller support. Let λ^* be an extreme point optimal solution, and by contradiction assume that $\widetilde{\lambda}$ gives a projected certificate and is such that $\text{supp}(\widetilde{\lambda})$ is strictly contained in $\text{supp}(\lambda^*)$. Since $\widetilde{\lambda} \geq 0$ and also different from 0, by scaling we can assume without loss of generality that $e^T\widetilde{\lambda} = 1$, and thus $\widetilde{\lambda}$ is a feasible solution for the LP above. This implies that

$$\begin{aligned} B^T\left(\lambda^* - \widetilde{\lambda}\right) &= 0 \\ e^T\left(\lambda^* - \widetilde{\lambda}\right) &= 0, \end{aligned}$$

so the assumption $\text{supp}(\widetilde{\lambda}) \subsetneq \text{supp}(\lambda^*)$ implies that the columns of the matrix $\begin{bmatrix} B^T \\ e^T \end{bmatrix}$ in the support of λ^* are linearly dependent. But since λ^* is an extreme point, it is a basic solution, namely the columns of the matrix in the support of λ^* are linearly independent. This reaches a contradiction and concludes the proof.

B.2 Original Feasibility Pump stalls even when flipping variables with zero fractionality is allowed

In Section 3.2 we showed that the original Feasibility Pump without restarts may stall; we now show that this is still the case even if variables with zero fractionality can be flipped in the perturbation step.

Let TT, the number of variables to be flipped, be randomly selected from the set $[t,T] \cap \mathbb{Z}$, where $T \in \mathbb{Z}_{++}$ is a pre-determined constant in the FP code (independent of the instance). Moreover assume the reasonable convention that for two variables with equal fractionality, we break ties using their index number, that is, if the x_i and x_j have the same fractionality and $i < j$, then x_i is picked before x_j to be flipped.

Consider the following subset-sum problem:

$$\begin{aligned} \max \quad & x_{T+2} \\ \text{s.t.} \quad & 5x_1 + \cdots + 5x_{T+1} + 2x_{T+2} = 5T+5 \\ & x_i \in \{0,1\} \ \forall \ i \in [T+2] \end{aligned}$$

Clearly the LP optimal solution $\bar{x}^0$ is of the form $\bar{x}^0_{T+2} = 1$, $\bar{x}^0_i = \frac{3}{5}$ for some $i \in [T+1]$ and $\bar{x}^0_j = 1$ for all $j \in [T+1] \setminus \{i\}$. Rounding this we obtain $\tilde{x}^0$ which is of the form $\tilde{x}^0_{T+2} = 1$ and $\tilde{x}^0_j = 1$ for all $j \in [T+1]$. It is also straightforward to verify that $\tilde{x}^0$ is a stalling solution (see Definition 26). So that algorithm randomly selects TT from the set $[t,T] \cap \mathbb{Z}$ and flips TT variables. Note that

only x_i has a fractionality of $|\frac{3}{5}-1|$ and all the other variables have a fractionality of 0 for some $i \in [T+1]$. So using the convention for breaking ties, we flip x_i and $TT-1$ other variables. Since $TT \leq T < T+1$, the new point $\widetilde{x}$ is of the form $\widetilde{x}_{T+2}=1$ and $\widetilde{x}_j=0$ for $j \in S \subseteq [T+1]$ and $\widetilde{x}_j=1$ for $j \in [T+1] \setminus S$. (Note that S can also be $\emptyset$ since we make no assumption on t).

First note that $\widetilde{x}$ is not a feasible solution since $\widetilde{x}_{T+2}=1$. Moreover,

1. If $S=\emptyset$, then $\widetilde{x}=\widetilde{x}^0$, a stalling solution visited before.

2. If $S \neq \emptyset$, then $5\widetilde{x}_1+\cdots+5\widetilde{x}_{T+1}+2\widetilde{x}_{T+2} < 5T+5$ and on projecting to the LP relaxation we will obtain a point of the form of $\bar{x}^0$. Rounding this again gives us $\tilde{x}^0$, a stalling solution visited before.

This completes the proof.

B.3 No long cycles in stalling

Lemma 41. Suppose that following is a sequence of points visited by Feasibility Pump (without any randomization):

$$(\bar{x}^1,\bar{y}^1) \to (\tilde{x}^1,\bar{y}^1) \to (\bar{x}^2,\bar{y}^2) \to (\tilde{x}^2,\bar{y}^2),$$

where $(\bar{x}^i,\bar{y}^i)$, $i \in \{1,2\}$ are the vertices of the LP relaxation, $\tilde{x}^i$, $i \in \{1,2\}$ are $0-1$ vectors, $\tilde{x}^i = \text{round}(\bar{x}^i)$ and $(\bar{x}^2,\bar{y}^2) = \ell_1\text{-proj}(\tilde{x}^1,\bar{y}^1)$. Then,

$$\|\bar{x}^1-\tilde{x}^1\|_1 \geq \|\bar{x}^2-\tilde{x}^2\|_1.$$

Proof. This result holds due to the fact that we are sequentially projecting using the same norm. In particular, we have that

$$\|\bar{x}^1-\tilde{x}^1\|_1 \geq \|\bar{x}^2-\tilde{x}^1\|_1,$$

since $(\bar{x}^2,\bar{y}^2) = \ell_1\text{-proj}(\tilde{x}^1,\bar{y}^1)$, i.e., $\bar{x}^2$ is a closest point in l_1-norm to $\tilde{x}^1$ in the projection of the LP

relaxation in the x-space. Then

$$\|\bar{x}^2 - \tilde{x}^1\|_1 \geq \|\bar{x}^2 - \tilde{x}^2\|_1,$$

since $\tilde{x}^1$ and $\tilde{x}^2$ are both integer points and $\tilde{x}^2$ is obtained by rounding $\bar{x}^2$ (and a rounded point is the closest integer point in ℓ_1 norm). □

A *long cycle* in feasibility pump is a sequence

$$(\bar{x}^1, \bar{y}^1) \to (\tilde{x}^1, \bar{y}^1) \to (\bar{x}^2, \bar{y}^2) \to (\tilde{x}^2, \bar{y}^2) \to \ldots (\bar{x}^k, \bar{y}^k) \to (\tilde{x}^k, \bar{y}^k)$$

where

1. $(\bar{x}^i, \bar{y}^i)$, $i \in \{1, 2, \ldots, k\}$ are the vertices of the LP relaxation, $\tilde{x}^i$, $i \in \{1, 2, \ldots, k\}$ are $0-1$ vectors, $\tilde{x}^i = \text{round}(\bar{x}^i)$ and $(\bar{x}^{i+1}, \bar{y}^{i+1}) = \ell_1\text{-proj}(\tilde{x}^i, \bar{y}^i)$,
2. $\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^{k-1}$ are unique integer vectors,
3. $\bar{x}^1 = \bar{x}^k$, $\tilde{x}^1 = \tilde{x}^k$, and
4. $k \geq 3$.

The statement of Theorem 5 is that such a scenario cannot occur, assuming 0.5 is always rounded consistently.

Proof of Theorem 5. Without loss of generally, we assume that 0.5 is rounded up to 1. Consider the sub-sequence $(\bar{x}^i, \bar{y}^i) \to (\tilde{x}^i, \bar{y}^i) \to (\bar{x}^{i+1}, \bar{y}^{i+1}) \to (\tilde{x}^{i+1}, \bar{y}^{i+1})$. By Lemma 41, since there is cycling, we have that

$$\|\bar{x}^i - \tilde{x}^i\|_1 = \|\bar{x}^{i+1} - \tilde{x}^i\|_1 = \|\bar{x}^{i+1} - \tilde{x}^{i+1}\|_1.$$

For simplicity and without loss of generality, we may assume that $\tilde{x}^i$ is the all ones vector. (This can be achieved by reflecting on coordinates the LP relaxation and the $[0,\ 1]^n$ hypercube.

Note that under such mappings, the sequence of points in feasibility pump will not be altered. Moreover, a point with value 0.5 in some coordinates $I \subseteq [n]$ will be mapped to a point with 0.5 in the coordinates I.)

Let $\emptyset \neq J \subseteq [n]$ be the set of indices where $\tilde{x}^i_j \neq \tilde{x}^{i+1}_j$, that is $\tilde{x}^{i+1}_j = 0$ for all $j \in J$. Since $\|\bar{x}^{i+1} - \tilde{x}^i\|_1 = \|\bar{x}^{i+1} - \tilde{x}^{i+1}\|_1$, we have

$$\begin{aligned} \sum_{j=1}^{n}(1-\bar{x}^{i+1}_j) &= \sum_{j\in[n]\setminus J}(1-\bar{x}^{i+1}_j) + \sum_{j\in J}\bar{x}^{i+1}_j \\ \Leftrightarrow \sum_{j\in J}\bar{x}^{i+1}_j &= \frac{|J|}{2}. \end{aligned} \tag{B.1}$$

Now observe that since $\tilde{x}^{i+1}_j = 0$ for $j \in J$, we must have that $\bar{x}^{i+1}_j < 0.5$ for all $j \in J$. This contradicts, (B.1).

□

REFERENCES

[1] S. Acer, E. Kayaaslan, and C. Aykanat, "A recursive bipartitioning algorithm for permuting sparse square matrices into block diagonal form with overlap," *SIAM Journal on Scientific Computing*, vol. 35, no. 1, 2013.

[2] T. Achterberg, "Constraint integer programming," PhD thesis, Technische Universität Berlin, 2007.

[3] T. Achterberg and T. Berthold, "Improving the feasibility pump," *Discrete Optimization*, vol. 4, no. 1, pp. 77–86, 2007.

[4] S. Ahmed, "A scenario decomposition algorithm for 0–1 stochastic programs," *Operations Research Letters*, vol. 41, no. 6, pp. 565–569, 2013.

[5] G. Angulo, S. Ahmed, and S. S. Dey, "Improving the integer l-shaped method," *INFORMS Journal on Computing*, vol. 28, no. 3, pp. 483–499, 2016.

[6] E. Balas, "Facets of the knapsack polytope," *Mathematical Programming*, vol. 8, pp. 146–164, 1975.

[7] E. Balas and E. Zemel, "Facets of knapsack polytope from minimal covers," *SIAM Journal on Applied Mathematics*, vol. 34, pp. 119–148, 1984.

[8] N. Bansal, N. Korula, V. Nagarajan, and A. Srinivasan, "On k-column sparse packing programs," in *International Conference on Integer Programming and Combinatorial Optimization*, Springer, 2010, pp. 369–382.

[9] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance, "Branch-and-price: column generation for solving huge integer programs," *Operations Research*, vol. 46, no. 3, pp. 316–329, 1998.

[10] A. Barvinok, "Thrifty approximations of convex bodies by polytopes," *International Mathematics Research Notices*, vol. 2014, no. 16, pp. 4341–4356, 2014.

[11] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische mathematik*, vol. 4, no. 1, pp. 238–252, 1962.

[12] P. Beraldi, G. Ghiani, A. Grieco, and E. Guerriero, "Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs," *Computers & OR*, vol. 35, no. 11, pp. 3644–3656, 2008.

[13] M. Bergner, A. Caprara, F. Furini, M. E. Lübbecke, E. Malaguti, and E. Traversi, "Partial convexification of general mips by dantzig-wolfe reformulation," in *International Conference on Integer Programming and Combinatorial Optimization*, Springer, 2011, pp. 39–51.

[14] L. Bertacco, M. Fischetti, and A. Lodi, "A feasibility pump heuristic for general mixed-integer problems," *Discrete Optimization*, vol. 4, no. 1, pp. 63–76, 2007.

[15] U. Bertele and F. Brioschi, *Nonserial dynamic programming*. Academic Press, 1972.

[16] D. Bienstock and G. Munoz, "Lp approximations to mixed-integer polynomial optimization problems," *arXiv preprint arXiv:1501.00288*, 2015.

[17] R. E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling, "Mixed integer programming: a progress report," *The Sharpest Cut*, pp. 309–325, 2004.

[18] M. Bodur, A. Del Pia, S. S. Dey, M. Molinaro, and S. Pokutta, "Aggregation-based cutting-planes for packing and covering integer programs," *arXiv preprint arXiv:1606.08951*, 2016.

[19] N. L. Boland, A. C. Eberhard, F. G. Engineer, M. Fischetti, M. W. Savelsbergh, and A. Tsoukalas, "Boosting the feasibility pump," *Mathematical Programming Computation*, vol. 6, no. 3, pp. 255–279, 2014.

[20] N. L. Boland, A. C. Eberhard, F. G. Engineer, and A. Tsoukalas, "A new approach to the feasibility pump in mixed integer programming," *SIAM Journal on Optimization*, vol. 22, no. 3, pp. 831–861, 2012.

[21] P. Bonami, G. Cornuéjols, A. Lodi, and F. Margot, "A feasibility pump for mixed integer nonlinear programs," *Mathematical Programming*, vol. 119, no. 2, pp. 331–352, 2009.

[22] R. Borndörfer, C. E. Ferreira, and A. Martin, "Decomposing matrices into blocks," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 236–269, 1998.

[23] R. Byrd, A. Goldman, and M. Heller, "Recognizing unbounded integer programs," *Operations Research*, vol. 35, no. 1, pp. 140–142, 1987.

[24] T. F. Coleman, *Large sparse numerical optimization*. Springer-Verlag New York, Inc., 1984.

[25] T. F. Coleman and L. A. Hulbert, "A direct active set algorithm for large sparse quadratic programs with simple bounds," *Mathematical Programming*, vol. 45, no. 1-3, pp. 373–406, 1989.

[26] H. Crowder, E. L. Johnson, and M. Padberg, "Solving large-scale zero-one linear programming problems," *Operations Research*, vol. 31, no. 5, pp. 803–834, 1983.

[27] E. Dalkiran and H. D. Sherali, "Theoretical filtering of rlt bound-factor constraints for solving polynomial programming problems to global optimality," *Journal of Global Optimization*, vol. 57, no. 4, pp. 1147–1172, 2013.

[28] E. Danna, E. Rothberg, and C. L. Pape, "Exploring relaxation induced neighborhoods to improve MIP solutions," *Mathematical Programming*, vol. 102, no. 1, pp. 71–90, 2005.

[29] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, no. 1, pp. 101–111, 1960.

[30] T. A. Davis, *Direct methods for sparse linear systems*. SIAM, 2006, vol. 2.

[31] M. De Santis, S. Lucidi, and F. Rinaldi, "New concave penalty functions for improving the feasibility pump," *Department of Computer and System Sciences Antonio Ruberti Technical Reports*, vol. 2, no. 10, 2010.

[32] S. S. Dey, "Strong cutting planes for unstructured mixed integer programs using multiple constraints," PhD thesis, Purdue University, 2007.

[33] S. S. Dey and J.-P. P. Richard, "Facets of the two-dimensional infinite group problems," *Mathematics of Operations Research*, vol. 33, pp. 140–166, 2008.

[34] —, "Sequential-merge facets for two-dimensional group problems," in *International Conference on Integer Programming and Combinatorial Optimization*, Springer-Verlag, 2007, pp. 30–42.

[35] —, "Some relations between facets of low- and high-dimensional group problems," *Mathematical Programming*, vol. 123, pp. 285–313, 2010.

[36] S. S. Dey, J.-P. P. Richard, Y. Li, and L. A. Miller, "On the extreme inequalities of infinite group problems," *Mathematical Programming*, vol. 121, pp. 145–170, 2010.

[37] S. S. Dey, A. Iroume, and M. Molinaro, "Some lower bounds on sparse outer approximations of polytopes," *Operations Research Letters*, vol. 43, no. 3, pp. 323–328, 2015.

[38] S. S. Dey, A. Iroume, M. Molinaro, and D. Salvagnin, "Improving the randomization step in feasibility pump," *arXiv preprint arXiv:1609.08121*, 2016.

[39] S. S. Dey, M. Molinaro, and Q. Wang, "Analysis of sparse cutting-planes for sparse milps with applications to stochastic milps," *arXiv preprint arXiv:1601.00198*, 2016.

[40] —, "Approximating polyhedra with sparse inequalities," *Mathematical Programming*, vol. 154, no. 1-2, pp. 329–352, 2015.

[41] C. DAmbrosio, A. Frangioni, L. Liberti, and A. Lodi, "A storm of feasibility pumps for nonconvex minlp," *Mathematical programming*, vol. 136, no. 2, pp. 375–402, 2012.

[42] —, "Experiments with a feasibility pump approach for nonconvex minlps," in *Experimental Algorithms*, Springer, 2010, pp. 350–360.

[43] M. Fischetti, F. Glover, and A. Lodi, "The feasibility pump," *Mathematical Programming*, vol. 104, no. 1, pp. 91–104, 2005.

[44] M. Fischetti and A. Lodi, "Heuristics in mixed integer programming," *Wiley Encyclopedia of Operations Research and Management Science*, 2011.

[45] M. Fischetti and D. Salvagnin, "Feasibility pump 2.0," *Mathematical Programming Computation*, vol. 1, no. 2-3, pp. 201–222, 2009.

[46] K. Fujisawa, M. Kojima, and K. Nakata, "Exploiting sparsity in primal-dual interior-point methods for semidefinite programming," *Mathematical Programming*, vol. 79, no. 1-3, pp. 235–253, 1997.

[47] R. Fukasawa and M. Goycoolea, "On the exact separation of mixed integer knapsack cuts," *Mathematical Programming*, vol. 128, no. 1-2, pp. 19–41, 2011.

[48] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, "Exploiting sparsity in semidefinite programming via matrix completion i: general framework," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 647–674, 2001.

[49] D. Fulkerson and O. Gross, "Incidence matrices and interval graphs," *Pacific Journal of Mathematics*, vol. 15, no. 3, pp. 835–855, 1965.

[50] D. Gade, S. Küçükyavuz, and S. Sen, "Decomposition algorithms with parametric gomory cuts for two-stage stochastic integer programs," *Mathematical Programming*, vol. 144, no. 1-2, pp. 39–64, 2014.

[51] G. Gamrath, T. Berthold, S. Heinz, and M. Winkler, "Structure-based primal heuristics for mixed integer programming," in *Optimization in the Real World*, Springer, 2016, pp. 37–53.

[52] A. M. Geoffrion, "Generalized benders decomposition," *Journal of Optimization Theory and Applications*, vol. 10, no. 4, pp. 237–260, 1972.

[53] A. M. Geoffrion and G. W. Graves, "Multicommodity distribution system design by benders decomposition," *Management Science*, vol. 20, no. 5, pp. 822–844, 1974.

[54] S. Ghosh, "Dins, a mip improvement heuristic," in *International Conference on Integer Programming and Combinatorial Optimization*, Springer, 2007, pp. 310–323.

[55] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, "Sparse matrix methods in optimization," *SIAM Journal on Scientific and Statistical Computing*, vol. 5, no. 3, pp. 562–589, 1984.

[56] P. C. Gilmore and R. E. Gomory, "A linear programming approach to the cutting-stock problem," *Operations research*, vol. 9, no. 6, pp. 849–859, 1961.

[57] R. E. Gomory, "Some polyhedra related to combinatorial problems," *Linear Algebra and its Application*, vol. 2, pp. 341–375, 1969.

[58] R. E. Gomory and E. L. Johnson, "Some continuous functions related to corner polyhedra, part I," *Mathematical Programming*, vol. 3, pp. 23–85, 1972.

[59] —, "Some continuous functions related to corner polyhedra, part II," *Mathematical Programming*, vol. 3, pp. 359–389, 1972.

[60] P Guerrero-Garcıa, "Range-space methods for sparse linear programs," PhD thesis, Ph. D. thesis, Department of Applied Mathematics, University of Málaga, Spain, 2002.

[61] I. ILOG, *CPLEX high-performance mathematical programming engine*, `http://www.ibm.com/software/integration/optimization/cplex/`.

[62] E. L. Johnson, "On the group problem for mixed integer programming," *Mathematical Programming Study*, vol. 2, pp. 137–179, 1974.

[63] K. Kaparis and A. N. Letchford, "Separation algorithms for 0-1 knapsack polytopes," *Mathematical programming*, vol. 124, no. 1-2, pp. 69–91, 2010.

[64] S. Kim, M. Kojima, and P. Toint, "Recognizing underlying sparsity in optimization," *Mathematical Programming*, vol. 119, no. 2, pp. 273–303, 2009.

[65] T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R. E. Bixby, E. Danna, G. Gamrath, A. M. Gleixner, S. Heinz, et al., "Miplib 2010," *Mathematical Programming Computation*, vol. 3, no. 2, pp. 103–163, 2011.

[66] M. Kojima and M. Muramatsu, "A note on sparse sos and sdp relaxations for polynomial optimization problems over symmetric cones," *Computational Optimization and Applications*, vol. 42, no. 1, pp. 31–41, 2009.

[67] V. Koltchinskii, *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: Ecole dEté de Probabilités de Saint-Flour XXXVIII-2008*. Springer, 2011.

[68] G. Laporte and F. V. Louveaux, "The integer l-shaped method for stochastic integer programs with complete recourse," *Operations Research Letters*, vol. 13, no. 3, pp. 133–142, 1993.

[69] J. B. Lasserre, "Convergent sdp-relaxations in polynomial optimization with sparsity," *SIAM Journal on Optimization*, vol. 17, no. 3, pp. 822–843, 2006.

[70] M. Laurent, "Sums of squares, moment matrices and optimization over polynomials," in *Emerging Applications of Algebraic Geometry*, Springer, 2009, pp. 157–270.

[71] S. L. Lauritzen, *Graphical models*. Clarendon Press, 1996, vol. 17.

[72] M. Ledoux and M. Talagrand, *Probability in Banach Spaces: Isoperimetry and Processes*. New York: Springer, 1991.

[73] F. C. Leone, L. S. Nelson, and R. B. Nottingham, "The folded normal distribution," *Technometrics*, vol. 3, no. 4, pp. 543–550, 1961.

[74] A. Lodi, "Mixed integer programming computation," in *50 Years of Integer Programming 1958-2008*, Springer, 2010, pp. 619–645.

[75] M. E. Lubbecke and C. Puchert, "Primal heuristics for mixed integer programs with a staircase structure," 2015.

[76] H. Marchand and L. A. Wolsey, "The 0-1 knapsack problem with a single continuous variable," *Mathematical Programming*, vol. 85, pp. 15–33, 1999.

[77] H. Marchand, A. Martin, R. Weismantel, and L. Wolsey, "Cutting planes in integer and mixed integer programming," *Discrete Applied Mathematics*, vol. 123, no. 1, pp. 397–446, 2002.

[78] H. Marchand and L. A. Wolsey, "Aggregation and mixed integer rounding to solve mips," *Operations Research*, vol. 49, no. 3, pp. 363–371, 2001.

[79] S. Minton, M. D. Johnston, A. B. Philips, and P. Laird, "Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems," *Artificial Intelligence*, vol. 58, no. 1-3, pp. 161–205, 1992.

[80] M. Mitzenmacher and E. Upfal, *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[81] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota, "Exploiting sparsity in semidefinite programming via matrix completion ii: implementation and numerical results," *Mathematical Programming*, vol. 95, no. 2, pp. 303–327, 2003.

[82] G. L. Nemhauser and L. A. Wolsey, "A recursive procedure to generate all cuts for 0-1 mixed integer programs," *Mathematical Programming*, vol. 46, pp. 379–390, 1990.

[83] C. H. Papadimitriou, "On selecting a satisfying truth assignment," in *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, IEEE, 1991, pp. 163–169.

[84] G. Pisier, *The volume of convex bodies and Banach space geometry*. Cambridge University Press, 1999.

[85] D. Pritchard and D. Chakrabarty, "Approximability of sparse integer programs," *Algorithmica*, vol. 61, no. 1, pp. 75–93, 2011.

[86] J.-P. P. Richard and S. S. Dey, "The group-theoretic approach in mixed integer programming," in *50 Years of Integer Programming 1958-2008*, Springer, 2010, pp. 727–801.

[87] N. Robertson and P. D. Seymour, "Graph minors. ii. algorithmic aspects of tree-width," *Journal of Algorithms*, vol. 7, no. 3, pp. 309–322, 1986.

[88] N. Robertson and P. D. Seymour, "Graph minors. iii. planar tree-width," *Journal of Combinatorial Theory, Series B*, vol. 36, no. 1, pp. 49–64, 1984.

[89] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.

[90] U. Schöning, "A probabilistic algorithm for k-sat and constraint satisfaction problems," in *Foundations of Computer Science, 1999. 40th Annual Symposium on*, IEEE, 1999, pp. 410–414.

[91] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[92] P. Toint, "A note about sparsity exploiting quasi-newton updates," *Mathematical Programming*, vol. 21, no. 1, pp. 172–181, 1981.

[93] F. Vanderbeck and L. A. Wolsey, "Reformulation and decomposition of integer programs," in *50 Years of Integer Programming 1958-2008*, Springer, 2010, pp. 431–502.

[94] M. J. Wainwright and M. I. Jordan, "Treewidth-based conditions for exactness of the sherali-adams and lasserre relaxations," *Univ. California, Berkeley, Technical Report*, vol. 671, p. 4, 2004.

[95] H. Waki, S. Kim, M. Kojima, and M. Muramatsu, "Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity," *SIAM Journal on Optimization*, vol. 17, no. 1, pp. 218–242, 2006.

[96] H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto, "Algorithm 883: sparsepop—a sparse semidefinite programming relaxation of polynomial optimization problems," *ACM Transactions on Mathematical Software (TOMS)*, vol. 35, no. 2, p. 15, 2008.

[97] M. Walter, "Sparsity of lift-and-project cutting planes," in *Operations Research Proceedings 2012*, Springer, 2014, pp. 9–14.

[98] J. Wang and T. Ralphs, "Computational experience with hypergraph-based methods for automatic decomposition in discrete optimization," in *International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems*, Springer, 2013, pp. 394–402.

[99] R. Weismantel, "On the 0/1 knapsack polytope," *Mathematical Programming*, vol. 77, no. 3, pp. 49–68, 1997.

[100] L. A. Wolsey, "Faces for a linear inequality in 0–1 variables," *Mathematical Programming*, vol. 8, no. 1, pp. 165–178, 1975.

[101] E. Zemel, "Lifting the facets of zero–one polytopes," *Mathematical Programming*, vol. 15, no. 1, pp. 268–277, 1978.

VITA

Andres Iroume was born in Edinburgh, Scotland. Shortly after, he moved to southern Chile where he grew up. In 2004, he became a student at the School of Mathematical and Physical Sciences, Universidad de Chile. There, he earned a Bachelor of Science in Mathematical Engineering and then a Master of Science in Operations Management. During his studies, Andres was involved in research projects related with natural resource planning and power systems. Andres was also a Teaching Assistant for multiple classes, including calculus, statistics and economics.

In 2011, Andres joined the Ph.D. program for Operations Research at the H. Milton Stewart School of Industrial and Systems Engineering at Georgia Tech. He carried out his dissertation research under the supervision of Dr. Santanu Dey. Andres's broader research interests involve developing theory and methods to solve optimization problems. His work is focused on sparse integer programs, polyhedral theory, primal heuristics and cutting plane theory. During his PhD, Andres was involved in teaching both as a Teaching Assistant and an Instructor, mostly in classes related with optimization.

On a personal level, Andres is married to Stefania Stefansdottir.