SINGLE-ROW MIXED-INTEGER PROGRAMS: THEORY AND COMPUTATIONS

A Thesis
Presented to
The Academic Faculty

by

Ricardo Fukasawa

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

School of Industrial and Systems Engineering
Georgia Institute of Technology
August 2008

SINGLE-ROW MIXED-INTEGER PROGRAMS: THEORY AND COMPUTATIONS

Approved by:

Professor William J. Cook, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor George L. Nemhauser
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Ellis Johnson
School of Industrial and Systems Engineering
Georgia Institute of Technology

Professor Robin Thomas
School of Mathematics
Georgia Institute of Technology

Professor Zonghao Gu
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: TBD

To my son Felipe and my wife Marta.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. William Cook, for his support throughout my PhD. His invaluable advices were instrumental not only for my success during my studies, but also to steer me in the right direction and help me grow to be a more mature researcher. His ideas and personality were a great source of inspiration for me. And last, but not least, his financial support since the start of my PhD helped me have exposure to state-of-the-art research since I got to GeorgiaTech.

I would like to thank all my friends and colleagues at GeorgiaTech not only for helpful discussions, insights and advice on several different issues, but especially for making these last five years a lot of fun. In particular, I would like to thank Marcos Goycoolea and Daniel Espinoza, who since I arrived have helped me in all different aspects of my life and have become very good friends. Also I would like to give special thanks to Alejandro Toriello, Amandeep Parmar, Bernardo Pagnoncelli, Claudio Santiago, Dan Faissol, Dan Steffy, Doug Altner, Eser Kirkizlar, Faram Engineer, Fatma Kilinc-Karzan, Gizem Keysan, Helder Inacio, Jim Luetdke, Juan Pablo Vielma, Yi He, among many others whose name I am failing to mention that were part of my life here.

I would also like to thank all my friends from Brazil, for their continued friendship and support in spite of the distance. In particular, I'd like to thank Alexandre Belloni and Mauricio Palmeira, who have been friends of mine since high school and still today are very present in my life.

I would like to thank all the professors at GeorgiaTech for all that I have learned from you. I specifically want to thank Dr. R. Gary Parker for all his advice, support and availability in helping me with all my problems.

I would like to thank all my family, for supporting me in all my decisions and encouraging me to pursue my dreams. In particular, I would like to thank my wife Marta and my son Felipe, for helping me put things into perspective and for giving a sense of purpose to all my endeavours.

Finally, I would also like to thank David Applegate, Sanjeeb Dash and Oktay Günlük for mentoring me during two great summers at AT&T Research and IBM Research and for their continued support afterwards.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Single-row mixed-integer programming (MIP) problems have been studied thoroughly under many different perspectives over the years. While not many practical applications can be modeled as a single-row MIP, their importance lies in the fact that they are simple, natural and very useful relaxations of generic MIPs.

This thesis will focus on such MIPs and present theoretical and computational advances in their study.

Chapter 1 presents an introduction to single-row MIPs, a historical overview of results and a motivation of why we will be studying them. It will also contain a brief review of the topics studied in this thesis as well as our contribution to them.

In Chapter 2, we introduce a generalization of a very important structured single-row MIP: Gomory's master cyclic group polyhedron (MCGP). We will show a structural result for the generalization, characterizing all facet-defining inequalities for it. This structural result allows us to develop relationships with MCGP, extend it to the mixed-integer case and show how it can be used to generate new valid inequalities for MIPs.

Chapter 3 presents research on an algorithmic view on how to maximally lift continuous and integer variables. Connections to tilting and fractional programming will also be presented. Even though lifting is not particular to single-row MIPs, we envision that the general use of the techniques presented should be on easily solvable MIP relaxations such as single-row MIPs. In fact, Chapter 4 uses the lifting algorithm presented.

Chapter 4 presents an extension to the work of Goycoolea [62] which attempts

to evaluate the effectiveness of Mixed Integer Rounding (MIR) and Gomory mixed-integer (GMI) inequalities. By extending his work, natural benchmarks arise, against which any class of cuts derived from single-row MIPs can be compared.

Finally, Chapter 5 is dedicated to dealing with an important computational problem when developing any computer code for solving MIPs, namely the problem of numerical accuracy. This proble arises due to the intrinsic arithmetic errors in computer floating-point arithmetic. We propose a simple approach to deal with this issue in the context of generating MIR/GMI inequalities.

CHAPTER I

INTRODUCTION

A mixed-integer programming problem (MIP) is an optimization problem (P) of the form:

$$\begin{aligned}
\min \quad & c^T x & \\
\text{s.t.} \quad & Ax \leq d & (1)\\
& l \leq x \leq u & (2)\\
& x_i \in \mathbb{Z}, \forall i \in I & (3)
\end{aligned}$$

where c, l and u are n-dimensional vectors, x is an n-dimensional vector of variables, A is an $m \times n$ matrix, b is an m-dimensional vector and $I \subseteq \{1, \ldots, n\}$. We assume that the components of l can be $-\infty$ and the components of u can be $+\infty$ and that all data is rational.

Given a MIP problem (P) as above, we call P its feasible region, $conv(P)$ the convex hull of its feasible points and P_{LP} the feasible region of the *continuous relaxation* of (P), i.e., the set of points satisfying (1) and (2). In addition, define $cone(P)$ as the recession cone of $conv(P)$ (which coincides with the recession cone of P_{LP}).

MIPs are a very powerful tool in Operations Research, whose importance comes from its widespread applicability and successful application in many contexts like airline operations, logistics, medical treatment, telecommunications, energy and others. Therefore, the study of MIPs has immediate interest in many areas and this fact has led to the great development of the field in the last 50 years. This thesis presents progress in the field by studying an important subclass of problems: single-row MIPs.

In this chapter we review briefly the basic concepts and ideas that will be used throughout the thesis. Section 1.1 reviews cutting planes, which are the main focus of this thesis. Section 1.2 reviews some results on single-row MIPs and highlights the importance of such types of problems. The remaining sections of this chapter describe the contributions of this thesis in each particular aspect of single-row MIPs, as well as presenting a brief introduction to each of them.

1.1 Cutting planes

One of the most successful approaches to solving MIPs is the use of *cutting planes*. The idea is to relax the integrality constraints, obtaining the continuous relaxation of the problem, whose optimal solution x^*_{LP} is easy to obtain, but may not satisfy the integrality requirements. The continuous relaxation is then modified by adding to the formulation valid inequalities separating x^*_{LP} from P, in other words, we add inequalities of the form $\pi^T x \leq \pi_o$ that are satisfied by all points of P but not satisfied by x^*_{LP}. It is common to use the terms *cuts* or *cutting planes* to refer to such inequalities, so we will use these terms throughout.

The hope is that the addition of such cuts to P_{LP} will give a good approximation of $conv(P)$. Indeed, if we were able to obtain a full description of $conv(P)$ by linear inequalities, then the solution of (P) would be reduced to solving a linear program over $conv(P)$. For that reason, among the infinite number of valid inequalities for P it is preferable to add to P_{LP} the ones that define facets of $conv(P)$, since these are the only ones that are in some sense essential to the description of $conv(P)$. For more details, see [85].

The cutting plane algorithm was first proposed by Dantzig, Fulkerson and Johnson [34] to solve the Traveling Salesman Problem and was later studied by Gomory [54] to solve integer programs. Nowadays it is much more common (and more effective) to use cutting planes within a branch-and-bound framework to solve large MIPs. Such an approach was first implemented by Hong [70] and was later given the name *branch-and-cut* by Padberg and Rinaldi [89].

1.2 Single-row mixed-integer programs

We say that a MIP is a single-row MIP whenever the system (1) is defined by a single constraint. In other words, it is a problem of the form:

$$
\begin{aligned}
\min \quad & c^T x \\
\text{s.t.} \quad & a^T x \leq b \\
& l \leq x \leq u \\
& x_i \in \mathbb{Z}, \forall i \in I
\end{aligned}
$$

where a is an n-dimensional vector and b is a scalar.

Single-row mixed-integer programming problems have been studied thoroughly under many different perspectives over the years. While not many practical applications can be modeled as a single-row MIP, their importance lies in the fact that they are simple, natural and very useful relaxations of more general MIPs. In fact, if we consider the original MIP:

$$
\begin{aligned}
(P) \quad \min \quad & c^T x & \\
\text{s.t.} \quad & Ax \leq d & (1) \\
& l \leq x \leq u & (2) \\
& x_i \in \mathbb{Z}, \forall i \in I & (3)
\end{aligned}
$$

and aggregate constraints (1) by using multipliers $\lambda \in \mathbb{Q}_+^m$, we get the following single-row MIP:

$$
\begin{aligned}
(K) \quad \min \quad & c^T x \\
\text{s.t.} \quad & \lambda A x \leq \lambda d \\
& l \leq x \leq u \\
& x_i \in \mathbb{Z}, \forall i \in I.
\end{aligned}
$$

Note that any point that is feasible for (P) is also feasible for (K), therefore $P \subseteq K$ and any inequality that is valid for K is also valid for P. Throughout we will often use the term *mixed-integer knapsack (MIK) problem* to refer to a single-row mixed-integer program. Moreover, we will use *mixed-integer knapsack set* to refer to the feasible region of a mixed-integer knapsack problem. We will refer to any inequality valid for a given mixed-integer knapsack set as a *mixed-integer knapsack cut*.

The main idea is that since (K) is easier to study than (P), we would be able to obtain strong inequalities (hopefully facet-defining) easily and these would lead to strong inequalities for (P) [32]. In fact, Bixby et al. [20, 21] report that cutting planes play a crucial role in speeding up the practical solution to MIP problems, in particular, the Gomory mixed-integer cut (GMI), the mixed-integer rounding cut (MIR) and the lifted knapsack cover cuts, all of which are MIK cuts.

The literature in MIK cuts is vast and it would be outside the scope of this thesis to mention all of the existing papers. However, for the purpose of introducing MIK cuts, we highlight below some important papers on this subject that have appeared in the literature.

In a seminal paper, which is widely considered to be the paper marking the birth of the field of Integer Programming, Gomory [54] proposes an algorithm for solving pure integer programs based on a class of cutting planes which are MIK cuts. Later, in [55], he proposed a relaxation of general integer programs called the *corner polyhedron*, which is obtained by relaxing nonnegativity constraints for variables that are basic at a given basic feasible solution of the problem. Such a relaxation leads to the study of the so-called *master cyclic group polyhedron*, which provides a framework that allows the derivation of several classes of cuts, all of which are MIK cuts. In fact, any cut that is derived from the master cyclic group polyhedron is a MIK cut, since such a polyhedron is defined by a single constraint. Examples of classes of cuts that are obtained from facets of the master cyclic group polyhedron are the GMI/MIR cut [57], two-step MIR [35], 2-slope, and 3-slope cuts [9]. Several other studies related to the master cyclic group polyhedron exist, for instance see [36, 37, 51, 50, 58, 60, 59].

Another very important paper in the context of MIK cuts is the one by Balas et al. [16]. Before such a paper, Gomory cuts were regarded as impractical, with many authors stating either their unsuccessful experience with them or their beliefs as to why they would not work well in practice (for more details on this see [30]). Balas et al. showed that Gomory cuts could be used in practice with great success within a branch-and-cut framework and after that paper Gomory cuts started to be in the spotlight again, leading to several variations and improvements of it [3, 31, 76, 78] and ultimately becoming the most successful class of cuts in commercial solvers like CPLEX.

Other important papers about MIK cuts are the ones dealing with lifted knapsack cover cuts [15, 18, 32, 64, 66, 65, 69, 90, 100, 104]. The knapsack cover cuts are based on the simple observation that a set of binary variables (called a *cover*) cannot all be 1 since otherwise the knapsack constraint will be violated (assuming all coefficients are nonnegative). Lifting is then performed to strengthen the coefficients of the other variables which are not present in the cover.

More recently, there have also been papers dealing with general MIK cuts that are obtained by a separation routine based on an optimization oracle [13, 23, 48, 62, 74, 103] as well as computational studies regarding several different classes of MIK cuts [17, 22, 37, 38, 49, 51].

Other important recent classes of cuts that have appeared are the mingling cuts [12] and the equivalent Atamtürk's lifted knapsack cuts [10], weight inequalities [99], the $\{0, \frac{1}{2}\}$-cuts [24] and the mod-k cuts [25] among others.

The purpose of this thesis is to advance the existing knowledge in the context of cutting planes for single-row MIPs, both with theoretical results and computational experiments. In what follows, we will give a brief summary of the motivation and the results that are going to be presented throughout this thesis.

1.3 The master cyclic group polyhedron

As mentioned in the previous section, an important polyhedron for generating MIK cuts is the *master cyclic group polyhedron* (MCGP), which is defined as:

$$P(n,r) = conv\left\{(x,y) \in \mathbb{Z}_+^{n-1} \times \mathbb{Z}_+ : \sum_{i=1}^{n-1} ix_i - ny_n = r\right\}, \tag{4}$$

where $r, n \in \mathbb{Z}$, and $n > r > 0$. Note that the Master Cyclic Group Polyhedron is usually presented as

$$P'(n,r) \;=\; conv\left\{x \in \mathbb{Z}_+^{n-1} : \sum_{i=1}^{n-1} ix_i \equiv r \mod n\right\}$$

which is the projection of $P(n,r)$ in the space of x variables. We use (4) as it makes the comparison to MEP simpler.

Note that we may assume, without loss of generality, that any valid inequality $\pi x + \rho_n y_n \geq \pi_o$ for $P(n,r)$ satisfies $\rho_n = 0$, since we can always add to it multiples of the original equality constraint so that such a condition is satisfied. Now, given any facet-defining inequality $\pi x \geq \pi_o$ of $P(n,r)$, define $f : [0,1] \to \mathbb{R}$ as:

$$f(a) := \begin{cases} 0, & \text{if } a = 0 \text{ or } a = 1 \\ \pi_i, & \text{if } a = \frac{i}{n} \text{ for some } i \in \{1,\ldots,n-1\} \\ (1-r)f(\frac{i}{n}) + rf(\frac{i+1}{n}), & \text{if } a = \frac{i+r}{n} \text{ for some } i \in \{0,\ldots,n-1\} \\ & \text{and } 0 < r < 1. \end{cases}$$

It is easy to show that, for a single-row MIP of the form

$$Q = \left\{w \in \mathbb{Z}_+^q : \sum_{j=1}^{q} p_j w_j = s\right\},$$

inequality $\sum_{j=1}^{q} f(\hat{p_j})w_j \geq f(\hat{s})$ is valid for Q. Therefore, facet-defining inequalities for $P(n,r)$ can be used to yield (hopefully strong) valid inequalities for any MIP.

However, this fact would not be useful unless facet-defining inequalities of $P(n,r)$ were easier to obtain than facet-defining inequalities of Q. Fortunately, this is indeed the case as shown in the following theorem by Gomory [55]:

Theorem 1.3.1 (Gomory [55]). *The inequality $\bar{\pi}x \geq 1$ defines a nontrivial facet of $P(n, r)$, for $n > r > 0$, if and only if $\bar{\pi} \in \mathbb{R}^{n-1}$ is an extreme point of*

$$\Theta = \begin{cases} \pi_i + \pi_j \geq \pi_{(i+j)\,mod\,n} & \forall i, j \in \{1, \ldots, n-1\}, \\ \pi_i + \pi_j = \pi_r & \forall i, j \text{ such that } r = (i+j)\,mod\,n, \\ \pi_j \geq 0 & \forall j \in \{1, \ldots, n-1\}, \\ \pi_r = 1. & \end{cases}$$

This basic underlying theorem behind group cuts allowed future research in the area, as mentioned in Section 1.2. In particular, it implies that the separation problem for $P(n, r)$ can be solved by using a small LP. It also gives conditions which one can use to check if or when a given class of inequalities is facet-defining for $P(n, r)$.

1.3.1 Contributions of this thesis

In Chapter 2, we extend the MCGP by considering the *master equality polyhedron* (MEP), which we define as:

$$K(n, r) = conv\left\{(x, y) \in \mathbb{Z}_+^n \times \mathbb{Z}_+^n : \sum_{i=1}^n ix_i - \sum_{i=1}^n iy_i = r\right\}.$$

It is easy to see that MEP contains MCGP as a proper face and, as such, is a generalization of it.

Similar to Gomory [55] we derive a characterization of all facet-defining inequalities of the MEP, which allows easier separation and identification of possible facet-defining inequalities.

Since MCGP is a face of MEP, any facet-defining inequality of MCGP can be lifted to obtain a facet-defining inequality of MEP. We study such liftings and show that the number of possible lifted facets is only linear in the number of variables. This is in contrast to general lifting, in which there is a potentially exponential number of lifted facet-defining inequalities. Still in the context of lifting, one could question whether all facets of MEP can be obtained by lifting facets of MCGP. We show, by means of a constructive general counterexample, that there are facets that cannot be obtained in such a way, thus showing that there exists MEP facets that lead to inequalities that could not be obtained before by using facets of the MCGP.

Finally, we show that all the results for MEP can be extended to the case where there are continuous variables present and show how facet-defining inequalities for MEP can be used to generate valid inequalities for general MIPs.

1.4 Lifting

Lifting is a technique used to obtain valid inequalities for a polyhedron by extending known valid inequalities for lower dimensional restrictions of it. It was first proposed by Gomory [55] within the context of the group problem. Later, Pollatschek [91] observed, in the context of independence systems, that some facets could be obtained by lifting facets of a lower-dimensional polyhedron and Padberg [88] proposed a method for obtaining the lifting coefficients for set packing polyhedra. Several papers afterwards studied lifting in different special cases [15, 69, 86, 100]. Zemel [104] extended the procedure to general binary problems and suggested a procedure that exhaustively characterized all facets that can be obtained by lifting for such problems. Wolsey [101] also extended lifting to the case of general bounded integer programs defined by a single constraint.

In what follows, we will describe in a little more detail the main ideas behind traditional lifting and our contribution to this concept.

Consider the mixed-integer set P:

$$
\begin{aligned}
& Ax \leq d \\
& l \leq x \leq u \\
& x_i \in \mathbb{Z}, \forall i \in I \subseteq N = \{1, \ldots, n\}
\end{aligned}
$$

Let F be a subset of N and $P(F)$ the following restriction of P:

$$A_R x_R \leq d' = d - A_F y_F$$

$$l \leq x \leq u$$

$$x_i \in \mathbb{Z}, \forall i \in I$$

obtained by fixing all variables x_i to y_i for all $i \in F$ (we assume $P \cap \{x_i = y_i\}_{i \in F} \neq \emptyset$). In the above system, $R = N \setminus F$, A_R and A_F are the submatrices of A corresponding to the columns in R and F and x_R and x_F are the components of x corresponding to columns in R and F respectively.

Now suppose that we have an inequality $\pi_R x_R \leq \pi_o$ valid for $P(F)$. Typically, lifting consists in obtaining coefficients π_F such that $\pi_R x_R + \pi_F(x_F - y_F) \leq \pi_o$ is valid for P.

One possible way to obtain the coefficients π_F is to do *sequential lifting*, which consists in ordering the variable indices in F as $(f_1, \ldots, f_k)$ and then lifting one variable at a time in that order. The procedure obtains an inequality $\pi_R x_R + \sum_{i=1}^{l} \pi_{f_i}(x_{f_i} - y_{f_i}) \leq \pi_o$ valid for $P(F \setminus \{f_1, \ldots, f_l\})$ from an inequality $\pi_R x_R + \sum_{i=1}^{l-1} \pi_{f_i}(x_{f_i} - y_{f_i}) \leq \pi_o$ valid for $P(F \setminus \{f_1, \ldots, f_{l-1}\})$ for all $l = 1, \ldots, k$. We say that the lifting is *maximal* if the coefficient π_{f_l} is the highest possible, i.e., if $\hat{\pi} > \pi_{f_l}$ then $\pi_R x_R + \sum_{i=1}^{l-1} \pi_{f_i}(x_{f_i} - y_{f_i}) + \hat{\pi}(x_{f_l} - y_{f_l}) \leq \pi_o$ is not valid for $P(F \setminus \{f_1, \ldots, f_l\})$.

It is well-known that the sequence in which variables are lifted affects (in general) the final valid inequality, in the sense that the earlier a variable is lifted, the higher will be its coefficient. Also, it is well-known that sequential lifting does not yield all possible lifted facets of a polyhedron. As an alternative to sequential lifting, Zemel [104] proposed the so-called *simultaneous lifting* where all variables are lifted simultaneously. Gu et al. [67] later generalized both frameworks by partitioning F into several subsets of variables and simultaneously lifting one set at a time.

Sequential lifting is usually considered very expensive in terms of computing time and therefore, one of the most used approaches is *sequence-independent lifting* (see for instance [10, 11, 66, 67, 102]), which consists in deriving a special class of superadditive functions that can easily generate the lifted cut coefficients. As its name suggests, in sequence-independent lifting, the lifting coefficients do not depend on the order in which the variables are lifted. In that sense, sequence-independent lifting can be seen as a simultaneous lifting approach. One great advantage to this approach is that it is much cheaper in terms of computing time. However, there are two drawbacks:

1. The lifted coefficients may not be as high as they could be if one was performing maximal sequential lifting.

2. It requires the computation of a superadditive valid lifting function, which is usually hard to compute and requires a thorough analysis of a particular problem in advance.

1.4.1 Contributions of this thesis

In Chapter 3 we derive a lifting procedure that uses an optimization oracle for a MIP problem to sequentially compute maximal lifting coefficients. Such an approach is better than sequence-independent lifting in the sense that it computes the maximal lifting coefficients and it does not require any a priori study of the polyhedron for which lifting is performed and, therefore, can be quickly implemented to perform lifting on any type of polyhedra. However, it has the drawback of being computationally more expensive than sequence-independent lifting.

The procedure is an extension of the algorithm by Easton and Gutierrez [46], which in turn are closely related to Dinkelback's algorithm for fractional programming (see for instance [96]) and the Newton-Rhapson method for finding roots of a function. Such an approach has connections to tilting and fractional programming and we also present these in this chapter. We note that, in the special cases of tilting and fractional programming, our algorithm is essentially the same as that of Applegate et al. [6]

and Espinoza [47] for tilting and Dinkelback's algorithm for fractional programming. However, one of the contributions of this chapter is to present a framework under which all these problems can be unified and solved and where results from one area can be applied to the others.

One other interesting feature of the algorithm in the context of lifting is that it deals with lifting from general faces of a polyhedron, not only ones obtained by fixing variables at certain values. It is important to note, however, that Louveaux and Wolsey [77] recently presented lifting in a very general way, also considering lifting from general faces and noted that lifting from faces defined by constraints can be seen as lifting the slack variables of such constraints. However, their paper presents a general framework, and a review of several results on lifting, but does not present a general way to lift.

Another contribution of the proposed algorithm is that, as far as we are aware, this is the first algorithmic approach proposed to lift general integer and continuous variables without any further assumptions. Indeed Richard et al. [92, 93] studied the lifting of continuous variables coupled with binary integer variables in a single constraint, while other papers deal with lifting of continuous variables in more specific problems [39, 40, 41, 66]. Meanwhile, several other papers treat lifting of integer variables, but usually under the assumption that the variables are either binary or at least bounded.

We note that, in spite of the fact that the algorithm is not specific for single-row MIPs, we envision that the general use of the techniques presented should be on easily solvable MIP relaxations such as single-row MIPs. In fact, Chapter 4 will use the lifting algorithm presented.

Finally, this new way of looking at lifting gives rise to a new way of thinking about simultaneous lifting of several variables (we call this *multi-lifting*), which we briefly introduce as a future research direction.

1.5 Benchmarking mixed-integer knapsack cuts

As mentioned in Section 1.2, the most successful classes of cuts for the practical solution of MIPs are the Mixed Integer Rounding (MIR) and Gomory mixed-integer (GMI) cuts. Due to this success, several improvements have been tried over the years over these types of cuts, such as [3, 31, 76, 78]. However, in spite of these efforts, it has been very hard to make a significant and consistent improvement over the performance of MIR/GMI cuts on the solution of general MIPs.

This lack of significant improvement has led to several papers over the last few years that try to assess the strength of GMI/MIR cuts. For example, there have been several papers presenting how much gap is closed by adding all possible cuts in a given class:

- Boyd [23] and Yan and Boyd [103] study MIK cuts generated from formulation rows.
- Fischetti and Lodi [49] study rank-1 Chvátal-Gomory cuts.
- Bonami et al. [22] study rank-1 projected Chvátal-Gomory cuts.
- Balas and Saxena [17] and Dash et al. [38] study rank-1 MIR cuts.
- Fischetti and Lodi [48] study rank-1 MIK cuts for binary problems.
- Fischetti and Saturni [51] study group cuts.

Another different type of study has been proposed by Dash and Günlük [37], in which they show that GMI cuts are strong in the context of group cuts. They proceed to do the following experiment: Solve the LP relaxation, add GMI cuts over the optimal tableau rows and reoptimize to obtain the optimal solution x^* to the new LP relaxation. They show that often it is the case that x^* is contained in the corner polyhedron defined by the optimal tableau rows of the original LP relaxation.

In other words, they show that the combined effect of GMI cuts "push" the LP relaxation solution inside the convex hull of each group problem defined by the rows of the original optimal LP tableau and thus any group cuts that can be derived from those rows would not yield any performance improvement over the GMI cuts.

Goycoolea [62] later extended this idea to show that the same does not hold true if we consider mixed-integer knapsack cuts, that is, after adding GMI cuts from the original optimal LP tableau, there are still MIK cuts that can be obtained from the same tableau rows that are still violated. However, in this context, a question still remains to be answered. Even in the instances that have violated MIK cuts that can be added after GMI cuts, how much of the remaining gap can be closed by using all possible MIK cuts?

1.5.1 Contributions of this thesis

Chapter 4 presents a contribution to try to answer the question posed above. The idea is that we will add all possible MIK cuts that can be derived from the original optimal LP tableau rows (or from other MIK sets). In that sense, this is also related to the above mentioned works that compute how much gap is closed by adding all possible cuts in a given class. By adding all such cuts, we will be able to evaluate how close to that gap the MIR/GMI cuts take us. Also, since we are obtaining MIK cuts, these experiments serve the purpose of a benchmark against which any other subclass of MIK cuts can be compared against to empirically determine its efficacy.

In principle, such benchmarks can be computed by using a separation linear program with an optimization oracle subroutine, much like the work of Applegate et al. [5], Espinoza [47] and Goycoolea [62]. However, such a direct approach is prohibitive in terms of computational time and therefore several ideas have to be applied in order to make running times more reasonable. These ideas will be presented in this chapter as well as the benchmark comparison results.

1.6 Numerically accurate cuts

Finally, Chapter 5 will be dedicated to dealing with an important computational problem that exists for any MIP solver based on floating point arithmetic, namely the problem of numerical accuracy. The issue is that several decisions throughout the solution of MIPs are based on the fact that numerical calculations are performed accurately, for example, fathoming a node by bound, fixing variables by reduced cost, generating valid cuts, etc.

By design, floating-point calculations are subject to an error and, since most MIP solvers are based on floating-point arithmetic, they are also subject to these errors and are therefore not guaranteed to return the correct optimal solution. In a related paper, Margot [79] recently shows that floating-point based cut generators from the COIN-OR library [28] often generate cutting planes that are invalid even for small binary instances. In floating-point based MIP solvers, such problems are dealt with in an heuristic fashion, using tolerances to determine, for example, when a number is integer or when a node should be fathomed among other decisions; restricting the rank of the MIR/GMI cuts that are used; discarding cuts that have "bad numerical properties"; "cleaning" the resulting cutting planes among others.

What is important to notice is that all of these measures are heuristics, and were derived through a process of engineering the solvers to make them commit errors less frequently. However, there are no guarantees that these heuristic rules will allow the solver to obtain the correct optimal solution. Moreover, the rule to restrict the rank and/or number of rounds of MIR/GMI cuts that are added may result in a significant loss in dual bound quality, which is well-known to be an important factor in the overall solution time of branch-and-bound based MIP algorithms.

There have been other attempts to cope with the numerical errors in floating point arithmetic like using rational arithmetic [7] or interval arithmetic [73, 87]. However, these approaches are hard to implement and in case of rational arithmetic, can lead to arithmetic computations that are orders of magnitude slower than standard floating-point arithmetic.

1.6.1 Contributions of this thesis

In Chapter 5 we present a simple approach to deal with this issue in the context of generating MIR/GMI cuts. Our proposed approach can be implemented with little effort in any floating-point based MIP solver and allows the generation of MIR/GMI cuts that are always guaranteed to be valid, in spite of the numerical inaccuracies of floating point calculations. We analyze the benefits and tradeoffs of using such an approach with computational experiments over MIPLIB and TSPLIB instances.

CHAPTER II

THE MASTER EQUALITY POLYHEDRON

2.1 Introduction

In this chapter, we study a structured polyhedron called the Master Equality polyhedron (MEP), which generalizes the Master Cyclic Group polyhedron (MCGP) proposed by Gomory [55] in the sense that the MCGP is a proper face of the MEP. As is the case for the MCGP, the study of the MEP leads to a framework that can be used for generating MIK cuts. In what follows, we review the MCGP and the Capacitated Vehicle Routing problem. The reason for looking at the latter is that the MEP appears as a natural substructure in it (and in several other combinatorial optimization problems) and therefore, the results that we obtain can be applied to generate strong cuts for these problems.

2.1.1 The Master Cyclic Group polyhedron

The Master Cyclic Group polyhedron is defined as

$$P(n,r) = conv\left\{(x,y) \in \mathbb{Z}_+^{n-1} \times \mathbb{Z}_+ : \sum_{i=1}^{n-1} ix_i - ny_n = r\right\}, \tag{5}$$

where $r, n \in \mathbb{Z}$, and $n > r > 0$. Note that the MCGP is usually presented as

$$P'(n,r) \;=\; conv\left\{x \in \mathbb{Z}_+^{n-1} : \sum_{i=1}^{n-1} ix_i \equiv r \mod n\right\}$$

which is the projection of $P(n,r)$ in the space of x variables. We use (5) as it makes the comparison to the MEP simpler.

The MCGP is an important structured polyhedron that can be used to generate valid inequalities for any single-row MIP and consequently for any MIP. For example, the Gomory mixed-integer cut (also known as the mixed-integer rounding (MIR) inequality) can be derived from a facet of $P(n,r)$ [57]. To see how facets of the MCGP can be used in general to generate valid inequalities for MIP, consider the following feasible region defined by a generic single equality constraint

$$Q = \left\{ w \in \mathbb{Z}_+^q : \sum_{j=1}^{q} p_j w_j = s \right\}$$

where $\{p_i\}_{i=1}^q$ and r are rational numbers.

We can rewrite Q as:

$$Q = \left\{ w \in \mathbb{Z}_+^q : \sum_{j=1}^{q} \lfloor p_j \rfloor w_j - \lfloor s \rfloor + \sum_{j=1}^{q} \hat{p_j} w_j = \hat{s} \right\}.$$

Since $p_j \in \mathbb{Q}$ for all $j = 1, \ldots, q$ and $s \in \mathbb{Q}$, let n be an integer such that $n\hat{p_j} \in \mathbb{Z}$ for all $j = 1, \ldots, q$ and $n\hat{s} \in \mathbb{Z}$. Now define $y_n = -(\sum\limits_{j=1}^{q} \lfloor p_j \rfloor w_j - \lfloor s \rfloor)$. Note that since $\sum\limits_{j=1}^{q} \lfloor p_j \rfloor w_j \leq \lfloor s \rfloor$, we have that $y_n \geq 0$ and integer. So any point in Q can be mapped to a point in:

$$\left\{ w \in \mathbb{Z}_+^q, y_n \in \mathbb{Z}_+ : \sum_{j=1}^{q} n\hat{p_j} w_j - n y_n = n\hat{s} \right\}$$

and consequently it can also be mapped to a point in

$$\left\{ (x,z) \in \mathbb{Z}_+^{n-1} \times \mathbb{Z}_+ : \sum_{i=1}^{n-1} i x_i - n y_n = n\hat{s} \right\}$$

by defining $x_i = \sum\limits_{j:n\hat{p_j}=i} w_j$. Now note that any valid inequality for $P(n, n\hat{s})$ has an equivalent representation with the coefficient of y_n being zero, since we can add any multiple of $\sum_{i=1}^{n-1} i x_i - n y_n = n\hat{s}$ to it. Therefore, any facet-defining inequality $\pi x \geq \pi_o$ of $P(n, n\hat{s})$ yields a valid inequality for P:

$$\sum_{j=1}^{q} \pi_{n\hat{p_j}} w_j \geq \pi_o.$$

This is the reason why facet-defining inequalities of $P(n,r)$ are important. Indeed, as mentioned in Section 1.2, several papers have studied the structure and computational aspects of MCGP and derived facets for it [9, 36, 35, 37, 51, 50, 57, 58, 60, 59].

However, this mapping from Q to $P(n,r)$ would not be useful unless facet-defining inequalities of $P(n,r)$ were easier to obtain than facet-defining inequalities of Q. Fortunately, this is indeed the case as shown in the following theorem by Gomory [55]:

Theorem 2.1.1 (Gomory [55]). *The inequality $\bar{\pi}x \geq 1$ defines a nontrivial facet of $P(n,r)$, for $n > r > 0$, if and only if $\bar{\pi} \in \mathbb{R}^{n-1}$ is an extreme point of*

$$\Theta = \begin{cases} \pi_i + \pi_j \geq \pi_{(i+j)\, mod n} & \forall i, j \in \{1, \dots, n-1\}, \\ \pi_i + \pi_j = \pi_r & \forall i, j \text{ such that } r = (i+j)\, mod n, \\ \pi_j \geq 0 & \forall j \in \{1, \dots, n-1\}, \\ \pi_r = 1. & \end{cases}$$

This is one of the basic underlying theorems behind group cuts and which allowed future research in the area. By extending the MCGP, we show that a similar result can be obtained for the MEP and, like was the case for the MCGP, we hope that it will lead to several different new classes of inequalities for MIPs.

2.1.2 The Capacitated Vehicle Routing problem

In this section, we introduce the Capacitated Vehicle Routing Problem (CVRP). The reason for doing so is to later present a connection of MEP to it and motivate how facet-defining inequalities for MEP can be useful for it.

Let $G = (V, E)$ be an undirected graph with vertices $V = \{0, 1, \dots, n\}$ and edges $E = \{1, \dots, m\}$. Vertex 0 is called the *depot*, and each remaining vertex $i \in \{1, \dots, n\}$ represents a *client* with an associated positive demand d_i. Call V_+ the set of clients, i.e. $V_+ = \{1, \dots, n\}$. Each edge $e \in E$ has a nonnegative length c_e. Given G and two positive integers K and C, the CVRP consists of finding routes for K vehicles satisfying the following constraints: (i) each route starts and ends at the depot, (ii) each client is visited by a single vehicle, and (iii) the total demand of all clients in any route is at most C. The goal is to minimize the sum of the lengths of all routes. This classical NP-hard problem is a natural generalization of the Travelling Salesman Problem (TSP), and has widespread application itself. The CVRP was first proposed in 1959 by Dantzig and Ramser [33] and has received close attention from the optimization community since then.

There are several formulations for solving the CVRP, but we will focus only on one in particular, since this is the one for which the connection with the MEP exists. Later, we will comment on why we believe that using such a formulation can be very good in practice.

Let x_{ij}^d be a binary variable that takes value 1 if and only if a vehicle goes from i to j with d units undelivered (i.e. it can still serve clients whose total demand is less than or equal to d). Also, let $D = (V, A)$ be the directed graph corresponding to $G = (V, E)$, where each edge $\{i, j\} \in E$ is replaced by two arcs (i, j) and (j, i) in A, with both arcs having the same length c_e. Moreover, for any $S \subseteq V$ let $\delta^+(S) := \{(i, j) \in A : i \in S,\ j \notin S\}$ and $\delta^-(S) := \{(i, j) \in A : i \notin S,\ j \in S\}$. Also, let $d(S) := \sum_{i \in S} d_i$. Then, a valid formulation for the CVRP is as follows:

$$
\begin{aligned}
\min \sum_{a\in A} c_a \sum_{d=0}^{C} x_a^d & & & \\
\text{s.t.} \quad \sum_{a\in\delta^-(\{i\})} \sum_{d=0}^{C} x_a^d &= 1 & \forall\, i \in V_+ \quad & (6)\\
\sum_{a\in\delta^+(\{i\})} \sum_{d=0}^{C} x_a^d &= 1 & \forall\, i \in V_+ \quad & (7)\\
\sum_{a\in\delta^-(\{0\})} \sum_{d=0}^{C} x_a^d &= K & & (8)\\
\sum_{a\in\delta^+(\{0\})} \sum_{d=0}^{C} x_a^d &= K & & (9)\\
\sum_{d=0}^{C} \sum_{a\in\delta^-(\{i\})} d x_a^d - \sum_{d=0}^{C} \sum_{a\in\delta^+(\{i\})} d x_a^d &= d_i & \forall\, i \in V_+ \quad & (10)\\
x_a^d &\in \{0,1\} & \forall\, a \in A, d \in 0,\ldots,C \quad & (11)
\end{aligned}
$$

Constraint (6) ensures that exactly one vehicle enters in each client, while constraint (7) ensures that exactly one vehicle leaves each client. Constraints (8) and (9) guarantee that exactly K vehicles leave and return to the depot. Finally, constraint (10) are *flow conservation* type of constraints, which guarantee that the vehicle that visits client i delivers exactly d_i units, since if it enters vertex i with d units undelivered, it must leave with $d - d_i$. A formulation based on the same idea already appears in Gouveia [61] for a closely related problem, the Capacitated Minimum Spanning Tree problem. Indeed in that paper, Gouveia mentions that such a formulation can be used for other problems like CVRP and Steiner Tree.

Now, pick any set $S \subseteq V_+$ and sum constraints (10) for all $i \in S$. We obtain the following constraint that is valid for all feasible CVRP solutions:

$$\sum_{d=0}^{C} \sum_{a\in\delta^-(S)} d x_a^d - \sum_{d=0}^{C} \sum_{a\in\delta^+(S)} d x_a^d = d(S)$$

Let:

$$ECC(S) := conv \left\{ x : \begin{array}{c} \sum_{d=0}^{C} \sum\limits_{a \in \delta^-(S)} dx_a^d - \sum_{d=0}^{C} \sum_{a \in \delta^+(S)} dx_a^d = d(S) \\ x \geq 0 \\ x \text{ integer vector} \end{array} \right\}$$

and call the facets defined by the nonnegativity constraints as trivial. We now prove that it suffices to aggregate all variables in $\delta^-(S)$ with a given index d into one single variable and likewise, aggregate all variables in $\delta^+(S)$ with a given index d into another single variable.

Proposition 2.1.2. *Let*

$$P = \left\{ x \in \mathbb{Z}_+^n : Ax = b, x \geq 0 \right\}.$$

Any facet-defining inequality $\pi x \geq \pi_o$ of P which is not a nonnegativity constraint satisfies:

$$\pi_i = \pi_j, \quad \forall i, j : A_i = A_j.$$

Proof. Suppose $\pi x \geq \pi_o$ is a facet-defining inequality of P that is not a nonnegativity constraint. Further, assume that $\pi x \geq \pi_o$ satisfies $\pi_i > \pi_j$ for some i, j such that $A_i = A_j$. Since this facet-defining inequality is not a nonnegativity constraint, there exists a point $\hat{x} \in P$ that has $\hat{x}_i > 0$ and such that $\pi \hat{x} = \pi_o$. But then, define $\bar{x}$ as:

$$\bar{x}_k := \begin{cases} \hat{x}_k & \text{if } k \notin \{i, j\} \\ \hat{x}_i + \hat{x}_j & \text{if } k = j \\ 0 & \text{if } k = i \end{cases}$$

and note that $\bar{x} \in P$ and $\pi \bar{x} < \pi \hat{x} = \pi_o$, which is a contradiction. ∎

Therefore we may define $w_d = \sum\limits_{a \in \delta^-(S)} x_a^d$ and $z_d = \sum\limits_{a \in \delta^+(S)} x_a^d$, obtaining the following set:

$$F(S) = conv \left\{ (w, z) \in \mathbb{Z}_+^{C+1} \times \mathbb{Z}_+^{C+1} : \sum_{d=0}^{C} dw_d - \sum_{d=0}^{C} dz_d = d(S) \right\}$$

and note that all facet-defining inequalities for $ECC(S)$ can be obtained from facet-defining inequalities of $F(S)$ by Proposition 2.1.2. Also, since $ECC(S)$ is a relaxation of the CVRP, any cut that is valid for $F(S)$ will give us a cut that is also valid for the CVRP. In the next section the connection between set $F(S)$ and the MEP will become obvious and thus it should also become clear how the MEP can be used to generate cuts for the CVRP.

Notice that the only part of the model that is related to the MEP are constraints (10) and therefore, any other problem that can be reformulated using those constraints can benefit from cuts of the MEP. Gouveia [61] and Uchoa [97] mention such connections and indeed cuts for $F(S)$ are used in Uchoa et al. [98] with great success for the capacitated minimum spanning tree problem. In that paper, valid inequalities for $F(S)$ (which are called Homogeneous Extended Capacity cuts) were used to reduce the integrality gap by more than 50% on average, making it possible to solve several open instances for the first time. Therefore the benefit of using strong inequalities for the MEP can be quite substantial to these types of problems.

One possible argument against the formulation presented is that it has a large number of variables, especially for instances with large C. We note, however that, if such a formulation is used implicitly as part of a branch-and-cut-and-price algorithm, there is no added complexity in terms of the solution of the LP relaxation. For more details on this, see [98].

2.1.3 The Master Equality Polyhedron

We now define the Master Equality Polyhedron (MEP):

$$K(n,r) = conv\left\{(x,y) \in \mathbb{Z}_+^n \times \mathbb{Z}_+^n : \sum_{i=1}^{n} ix_i - \sum_{i=1}^{n} iy_i = r\right\} \tag{12}$$

where $n, r \in \mathbb{Z}$ and $n > 0$. Without loss of generality we assume that $r \geq 0$. To the best of our knowledge, $K(n,r)$ was first defined by Uchoa [97] in a slightly different form, although its polyhedral structure was not studied in that paper.

Recall that an important polyhedron that leads to many classes of cuts for MIPs is the MCGP, which is defined as:

$$P(n,r) = conv\left\{(x,y) \in \mathbb{Z}_+^{n-1} \times \mathbb{Z}_+ : \sum_{i=1}^{n-1} ix_i - ny_n = r\right\}.$$

Another important polyhedron that is related to the MCGP and that can be used to obtain facets of the MCGP (see [9]) is called the Master Knapsack polyhedron (MKP), which is defined as

$$K(r) = conv\left\{x \in \mathbb{Z}_+^r : \sum_{i=1}^{r} ix_i = r\right\}, \tag{13}$$

where $r \in \mathbb{Z}$ and $r > 0$. It is easy to see that both the MCGP and the MKP are lower dimensional faces of the MEP, and as such, the MEP is a generalization of both, with a potentially richer family of facets.

Two key results for the MCGP and the MKP are the ones of Gomory [55] and Araóz [8], where an explicit characterization of the polar of their respective nontrivial facets is given. We give a similar description of the nontrivial facets of the MEP for $n \geq r \geq 0$. Like in the case of the MCGP and the MKP, such structural result enables us to derive several other results for the MEP. For instance, using this result, we obtain a polynomial time algorithm to separate over the MEP for all $r \geq 0$ (including $r > n$). In addition, we also analyze some structural properties of the MEP and describe how to obtain valid inequalities for general MIPs using facet-defining inequalities for the MEP.

Notice that, for $n > r > 0$, it is easy to obtain valid (facet-defining) inequalities for the MEP using valid (facet-defining) inequalities for the MCGP. To see how, note that

$$\sum_{i=1}^{n} ix_i - \sum_{i=1}^{n} iy_i = r \iff \sum_{i=1}^{n-1} i(x_i + y_{n-i}) - n(\sum_{i=1}^{n} y_i - x_n) = r. \tag{14}$$

Furthermore, $\sum_{i=1}^{n} y_i - x_n \geq 0$ for all $(x,y) \in K(n,r)$. Therefore, any valid inequality $\sum_{i=1}^{n-1} \pi_i x_i + \rho_n y_n \geq \beta$ for $P(n,r)$ leads to a valid inequality $\sum_{i=1}^{n-1} \pi_i x_i + \sum_{i=1}^{n-1}(\pi_{n-i} +$

$\rho_n)y_i + \rho_n y_n - \rho_n x_n \geq \beta$ for $K(n,r)$. In fact, as the MCGP is a lower dimensional face of the MEP, any valid inequality for the MCGP can lead to multiple valid inequalities (including the inequality presented above) for the MEP via lifting. Also, any facet-defining inequality of the MEP can be obtained by lifting some valid inequality of the MCGP. However, the important question is: Can every facet-defining inequality of the MEP be obtained by lifting facet-defining inequalities of the MCGP? Answering this question will tell us if, in some sense, the MEP can give us any new valid inequalities for general MIPs that the MCGP did not. Our characterization of the facets of the MEP allows us to study lifted inequalities from the MCGP and show that not all facets of the MEP can be obtained by lifting facets of the MCGP.

In the next section, we present our characterization of the polar of the nontrivial facets of $K(n,r)$, for any $n > 0$ and any r satisfying $0 < r \leq n$. In Section 2.3 we discuss how to lift facets of $P(n,r)$ to obtain facets of $K(n,r)$ and in Section 2.4, we show that not all facets of $K(n,r)$ can be obtained by lifting. In Section 2.5, we study $K(n,r)$ when $r = 0$. In Section 2.6, we describe how to separate an arbitrary point from $K(n,r)$ for any r, including the case $r > n$. Section 2.7 is dedicated to extending the results of $K(n,r)$ to the mixed-integer case. In Section 2.8, we follow the approach of Gomory and Johnson [57] to derive valid inequalities for mixed-integer programs from facets of $K(n,r)$ via *interpolation*. We conclude in Section 2.9 with some remarks on directions for further research on $K(n,r)$.

2.2 Polyhedral analysis of $K(n,r)$ when $n \geq r > 0$

Throughout this section, we assume $n \geq r > 0$. The case $r = 0$ is studied in Section 2.5. We start with some notation and some basic polyhedral properties of $K(n,r)$.

Let $e_i \in \mathbb{R}^{2n}$ be the unit vector with a one in the component corresponding to x_i and let $f_i \in \mathbb{R}^{2n}$ be the unit vector with a one in the component corresponding to y_i, for $i = 1, \ldots, n$.

Lemma 2.2.1. *$dim(K(n,r)) = 2n - 1$.*

Proof. Clearly $dim(K(n,r)) \leq 2n-1$ as all points in $K(n,r)$ satisfy $\sum_{i=1}^{n} ix_i - \sum_{i=1}^{n} iy_i = r$. Let U be the set of $2n$ points $p_1 = re_1$, $p_i = re_1 + e_i + if_1$ for $i = 2, \ldots, n$, and $q_i = (r+i)e_1 + f_i$ for $i = 1, \ldots, n$. The vectors of U are affinely independent, as $\{u - p_1 : u \in U, u \neq p_1\}$ is a set of linearly independent vectors. As $U \subseteq K(n,r)$, $dim(K(n,r)) \geq 2n-1$. ∎

Lemma 2.2.2. *The nonnegativity constraints of $K(n,r)$ are facet-defining if $n \geq 2$.*

Proof. Let U be defined as in the proof of Lemma 2.2.1. For any $i \neq 1$, the vectors in $U \setminus \{p_i\}$ and $U \setminus \{q_i\}$ are affinely independent, and satisfy $x_i = 0$ and $y_i = 0$, respectively. Therefore $x_i \geq 0$ and $y_i \geq 0$ define facets of $K(n,r)$ for $i \geq 2$. To see that $y_1 \geq 0$ is facet-defining, replace p_i $(2 \leq i \leq n)$ in U by $p_i' = re_1 + ne_i + if_n$ to get a set of affinely independent vectors $U' \subseteq K(n,r)$. All points in U' other than q_1 satisfy $y_1 = 0$. Finally, let V be the set of points $t_0 = e_n + (n-r)f_1$, $t_i = t_0 + ie_n + nf_i$ for $i = 1, \ldots, n$, and $s_i = t_0 + e_i + if_1$ for $i = 2, \ldots, n-1$. V is contained in $K(n,r)$, and its elements are affinely independent vectors as $\{v - t_0 : v \in V, v \neq t_0\}$ is a set of linearly independent vectors. The points in V also satisfy $x_1 = 0$. ∎

Clearly, $K(n,r)$ is an unbounded polyhedron. We next characterize all the extreme rays (one-dimensional faces of the recession cone) of $K(n,r)$. We represent an extreme ray $\{u + \lambda v : u, v \in \mathbb{R}_+^{2n}, \lambda \geq 0\}$ of $K(n,r)$ simply by the vector v. Let $r_{ij} = je_i + if_j$ for any $i, j \in \{1, \ldots, n\}$.

Lemma 2.2.3. *The set of extreme rays of $K(n,r)$ is given by $R = \{r_{ij} : 1 \leq i, j \leq n\}$.*

Proof. Let (c,d) be an extreme ray of $K(n,r)$, that is, $\sum_{i=1}^{n} ic_i - \sum_{j=1}^{n} jd_j = 0$. Since (c,d) is a one-dimensional face of the recession cone, at least $2n-2$ of the nonnegativity constraints hold as equality. As (c,d) can not have a single nonzero component, it must have exactly two nonzero components. Thus, (c,d) is of the form $\alpha e_i + \beta f_j$ and since $\sum_{i=1}^n ic_i - \sum_{j=1}^n jd_j = 0$, we have that $i\alpha - j\beta = 0 \Rightarrow \alpha = \frac{j}{i}\beta$. Since (c,d) is a ray, we may scale it so that $\beta = i$, and we have shown that R contains all extreme rays of $K(n,r)$.

To complete the proof, it suffices to notice that a conic combination of 2 or more rays in R gives a ray with at least 3 nonzero entries and therefore all rays in R are extreme. ∎

As $K(n,r)$ is not a full-dimensional polyhedron, any valid inequality $\pi x + \rho y \geq \pi_o$ for $K(n,r)$ has an equivalent representation with $\rho_n = 0$. If a valid inequality does not satisfy this condition, one can add an appropriate multiple of the equation $\sum_{i=1}^n ix_i - \sum_{i=1}^n iy_i = r$ to it. We state this formally in Observation 2.2.4, and subsequently assume all valid inequalities have $\rho_n = 0$. This is one of many possible choices of normalization and it was chosen to make the relation to facets of $P(n,r)$ easier, since Gomory's characterization also satisfies that property.

Observation 2.2.4. *If $\pi x + \rho y \geq \pi_o$ defines a valid inequality for $K(n,r)$, we may assume $\rho_n = 0$.*

We classify the facets of $K(n,r)$ as *trivial* and *nontrivial* facets.

Definition 2.2.5. *The following facet-defining inequalities of $K(n,r)$ are called trivial:*

$$x_i \geq 0, \quad \forall i = 1,\ldots,n, \tag{15}$$

$$y_i \geq 0, \quad \forall i = 1,\ldots,n-1. \tag{16}$$

All other facet-defining inequalities of $K(n,r)$ are called nontrivial.

According to this definition, the inequality $y_n \geq 0$ defines a nontrivial facet. There is nothing special about the $y_n \geq 0$ inequality except that it is the only nonnegativity constraint that does not comply directly with the $\rho_n = 0$ assumption. With this distinction between $y_n \geq 0$ and the other trivial facets, our results are easier to state and prove.

2.2.1 Characterization of the nontrivial facets

Let $N = \{1, \ldots, n\}$. We next state our main result:

Theorem 2.2.6. *Consider an inequality $\pi x + \rho y \geq \pi_o$ with $\rho_n = 0$. It defines a nontrivial facet of $K(n,r)$ if and only if the following conditions hold: (i) $\pi_o > 0$, and (ii) $(\pi, \rho, \pi_o)/\pi_o$ is an extreme point of $T \subseteq \mathbb{R}^{2n+1}$ where T is defined by the following linear equations and inequalities:*

$$\pi_i + \rho_j \geq \pi_{i-j}, \qquad \forall i,j \in N, \qquad i > j, \tag{SA1}$$

$$\pi_i + \pi_j \geq \pi_{i+j}, \qquad \forall i,j \in N, \qquad i + j \leq n, \tag{SA2}$$

$$\rho_k + \pi_i + \pi_j \geq \pi_{i+j-k}, \qquad \forall i,j,k \in N, \qquad 1 \leq i + j - k \leq n, \tag{SA3}$$

$$\pi_i + \pi_{r-i} = \pi_o, \qquad \forall i \in N, \qquad i < r, \tag{EP1}$$

$$\pi_r = \pi_o, \tag{EP2}$$

$$\pi_i + \rho_{i-r} = \pi_o, \qquad \forall i \in N \qquad i > r, \tag{EP3}$$

$$\rho_n = 0, \tag{NC1}$$

$$\pi_o = 1. \tag{NC2}$$

The proof of this theorem requires several preliminary results which will be presented in this section. Note that $\rho_n = 0$ is not a restrictive assumption in the above theorem since any valid inequality has an equivalent representation with $\rho_n = 0$.

We call constraints (SA1)-(SA3) *relaxed subadditivity* conditions as they are implied by the first four of the following pairwise subadditivity conditions:

$$\pi_i + \rho_i \geq 0, \qquad \forall i \in N, \tag{SA0}$$

$$\pi_i + \rho_j \geq \pi_{i-j}, \qquad \forall i, j \in N, \qquad i > j, \tag{SA1}$$

$$\pi_i + \rho_j \geq \rho_{j-i}, \qquad \forall i, j \in N, \qquad i < j, \tag{SA1'}$$

$$\pi_i + \pi_j \geq \pi_{i+j}, \qquad \forall i, j \in N, \qquad i + j \leq n, \tag{SA2}$$

$$\rho_i + \rho_j \geq \rho_{i+j}, \qquad \forall i, j \in N, \qquad i + j < n, \tag{SA2'}$$

$$\rho_i + \rho_j \geq \rho_{i+j}, \qquad \forall i, j \in N, \qquad i + j = n. \tag{SA2''}$$

We distinguish the cases in (SA2') and (SA2") since (SA2") is not satisfied by the nonnegativity constraint on y_n and we will need this distinction to establish certain structural properties later on.

Lemma 2.2.7. *Let* $(\pi, \rho) \in \mathbb{R}^{2n}$ *satisfy the pairwise subadditivity conditions* (SA0), (SA1), (SA1'), *and* (SA2), *then* (π, ρ) *satisfies* (SA3) *as well.*

Proof. Let $i, j, k \in N$ be such that $1 \leq i + j - k \leq n$ and without loss of generality assume that $i \leq j$. If $i + j \leq n$, then using (SA2) and (SA1) we have $\rho_k + \pi_i + \pi_j \geq \rho_k + \pi_{i+j} \geq \pi_{i+j-k}$.

If, on the other hand, $i + j > n$, we consider three cases:

Case 1: $k < j$. Using (SA1) and (SA2), we have: $\rho_k + \pi_i + \pi_j \geq \pi_{j-k} + \pi_i \geq \pi_{i+j-k}$.

Case 2: $k > i$. Using (SA1') and (SA1), we have: $\rho_k + \pi_i + \pi_j \geq \rho_{k-i} + \pi_j \geq \pi_{i+j-k}$.

Case 3: $k = i = j$. Using (SA0), we have: $\rho_k + \pi_k \geq 0 \iff \rho_k + \pi_k + \pi_k \geq \pi_k$. ∎

As we show later, any nontrivial facet-defining inequality $\pi x + \rho y \geq \pi_o$ for $K(n, r)$ satisfies (SA1)-(SA3) as well as (SA0), (SA1') and (SA2'). Based on Gomory's characterization of the MCGP using pairwise subadditivity conditions, it would seem more natural to have a characterization of the nontrivial facets using the pairwise subadditivity conditions above instead of the relaxed subadditivity conditions. We show in Section 2.2.4 that Theorem 2.2.6 does not hold if (SA3) is replaced with (SA0), (SA1') and (SA2').

The equations (EP1)-(EP3) essentially state that the following $n - \lfloor \frac{r-1}{2} \rfloor$ affinely independent points, which we call the *elementary points* of $K(n, r)$,

$$\{e_i + e_{r-i} \,:\, 1 \le i < r\} \cup \; e_r \; \cup \{e_i + f_{i-r}, \,:\, r < i \le n\}$$

lie on every nontrivial facet of $K(n, r)$. In other words, $K(n, r)$ has a face of dimension at least $n - \lfloor \frac{r-1}{2} \rfloor - 1$ where all nontrivial facets intersect.

The last two constraints (NC1) and (NC2) are normalization constraints that are necessary to have a unique representation of nontrivial facets.

Note that the definition of T in Theorem 2.2.6 is similar to that of a polar. However, T is not the polar of $K(n, r)$, as it does not contain extreme points of the polar that correspond to the trivial inequalities. In addition, some of the extreme rays of the polar are not present in T. It is possible to interpret T as an important subset of the polar that contains all extreme points of the polar besides the ones that lead to the trivial inequalities. In the rest of this section we develop the required analysis to prove Theorem 2.2.6.

2.2.2 Basic properties of T

We start with a basic observation which states that any valid inequality for $K(n, r)$ has to be valid for its extreme rays and elementary points.

Observation 2.2.8. *Let $\pi x + \rho y \ge \pi_o$ be a valid inequality for $K(n, r)$, then the following holds:*

$$j\pi_i + i\rho_j \ge 0, \qquad \forall i, j \in N \tag{R1}$$

$$\pi_i + \pi_{r-i} \ge \pi_o, \quad \forall i \in N, i < r, \tag{P1}$$

$$\pi_r \ge \pi_o, \tag{P2}$$

$$\pi_i + \rho_{i-r} \ge \pi_o, \quad \forall i \in N, i > r. \tag{P3}$$

We next show that nontrivial facet-defining inequalities satisfy the relaxed subadditivity conditions and they are tight at the elementary points of $K(n, r)$.

Lemma 2.2.9. *Let $\pi x + \rho y \geq \pi_o$ be a nontrivial facet-defining inequality of $K(n, r)$, then it satisfies* (SA1)-(SA3) *as well as* (SA0), (SA1'), (SA2') *and* (EP1)-(EP3).

Proof. Due to Observation 2.2.8, $i\pi_i + i\rho_i \geq 0$ for all $i \in N$ and therefore (SA0) holds. Next, let $(x^*, y^*) \in K(n, r)$ be such that $\pi x^* + \rho y^* = \pi_o$ and $x^*_{i-j} > 0$. Such a point exists because the facet-defining inequality we consider is nontrivial. Then $(x^*, y^*) + (e_i + f_j - e_{i-j})$ is also contained in $K(n, r)$. Therefore, (SA1) holds. Proofs for (SA2), (SA3), (SA1') and (SA2') are analogous.

Finally, let (x', y') and (x'', y'') be integral points in $K(n, r)$ lying on the facet defined by $\pi x + \rho y \geq \pi_o$ such that $x'_i > 0$ and $x''_{r-i} > 0$. Then $(\bar{x}, \bar{y}) = (x', y') + (x'', y'') - e_i - e_{r-i} \in K(n, r)$. Therefore

$$\pi\bar{x} + \rho\bar{y} = \pi x' + \rho y' + \pi x'' + \rho y'' - \pi_i - \pi_{r-i} = 2\pi_o - \pi_i - \pi_{r-i} \geq \pi_o.$$

The last inequality above implies that $\pi_i + \pi_{r-i} \leq \pi_o$ and therefore (P1) $\Rightarrow$ (EP1). The proofs of (EP2) and (EP3) are analogous. ∎

Observation 2.2.10. *If $\pi x + \rho y \geq \pi_o$ is a facet-defining inequality of $K(n, r)$ which is not a nonnegativity constraint, then it also satisfies* (SA2").

We next show that the normalization condition (NC2) does not eliminate any nontrivial facets.

Lemma 2.2.11. *Let $\pi x + \rho y \geq \pi_o$ be a nontrivial facet-defining inequality of $K(n, r)$, that satisfies $\rho_n = 0$. Then $\pi_o > 0$.*

Proof. By (R1), we have, for all $i \in N$, $n\pi_i + i\rho_n \geq 0$ and therefore $\pi_i \geq 0$ since $\rho_n = 0$. Also by (EP2), we have $\pi_o = \pi_r$ which implies that $\pi_o \geq 0$.

Assume $\pi_o = 0$. As $\pi \geq 0$, using (EP1) we have $\pi_i = 0$ for $i = 1, \ldots, r$. But then, (SA2) implies that

$$0 + \pi_{i-1} \geq \pi_i \geq 0, \text{ for } i = 2, \ldots, n.$$

Starting with $i = r + 1$, we can inductively show that $\pi_i = 0$ for all $i \in N$. This also implies that $\rho_k = 0$ for $1 \leq k \leq n - r$ by (EP3). In addition $\rho_k \geq 0$ for $n - r + 1 \leq k \leq n$ by (SA3).

Therefore, if $\pi_o = 0$, then $\pi = 0$, $\rho \geq 0$ and therefore $\pi x + \rho y \geq 0$ can be written as a conic combination of the nonnegativity facets, which is a contradiction. Thus $\pi_o > 0$. ∎

Combining Lemmas 2.2.9 and 2.2.11 we have therefore established the following.

Corollary 2.2.12. *Let $\pi x + \rho y \geq \pi_o$ be a nontrivial facet-defining inequality of $K(n, r)$ that satisfies $\rho_n = 0$. Then $\frac{1}{\pi_o}(\pi, \rho, \pi_o) \in T$.*

In the following result, we show that a subset of the conditions presented in Theorem 2.2.6 suffices to ensure the validity of an inequality for $K(n, r)$.

Lemma 2.2.13. *Let (π, ρ, π_o) satisfy* (SA1)*,* (SA2)*,* (SA3) *and* (EP2)*. Then $\pi x + \rho y \geq \pi_o$ defines a valid inequality for $K(n, r)$.*

Proof. We will prove this by contradiction. Assume that $\pi x + \rho y \geq \pi_o$ satisfies (EP2), (SA1), (SA2) and (SA3) but $\pi x + \rho y \geq \pi_o$ does not define a valid inequality for $K(n, r)$, $r > 0$. Let (x^*, y^*) be an integer point in $K(n, r)$ that has minimum L_1 norm among all points violated by $\pi x + \rho y \geq \pi_o$.

Clearly, $(x^*, y^*) \neq 0$ and therefore $||(x^*, y^*)||_1 > 0$. If $||(x^*, y^*)||_1 = 1$, then $x^* = e_r$ and $y^* = 0$, and as $\pi_r = \pi_o$, (x^*, y^*) does not violate the inequality, we have a contradiction. Therefore $||(x^*, y^*)||_1 \geq 2$. We next consider three cases.

Case 1: Assume that $y^* = 0$. Then $\sum_{i=1}^n i x_i^* = r$. By successively applying (SA2), we obtain

$$\pi_o > \sum_{i=1}^n \pi_i x_i^* \geq \sum_{i=1}^n \pi_{i x_i^*} \geq \pi_{\sum_{i=1}^n i x_i^*} = \pi_r$$

which contradicts (EP2).

Case 2: Assume that there exists $i > j$ such that $x_i^* > 0$ and $y_j^* > 0$. Let $(x', y') = (x^*, y^*) + (e_{i-j} - e_i - f_j)$. Clearly, $(x', y') \in K(n, r)$, and $||(x', y')||_1 = ||(x^*, y^*)||_1 - 1$. Moreover, as $\pi x + \rho y \geq \pi_o$ satisfies (SA1), $\pi x' + \rho y' = \pi x^* + \rho y^* + \pi_{i-j} - \pi_i - \rho_j \leq \pi x^* + \rho y^* < \pi_o$, which contradicts the choice of (x^*, y^*).

Case 3: Assume that for any $i, k \in N$, $x_i^* > 0$ and $y_k^* > 0$ imply that $i \leq k$. Let $i \in N$ be such that $x_i^* > 0$, then either there exists another index $j \neq i$ such that $x_j^* > 0$, or $x_i^* \geq 2$ (in which case, let $j = i$). If $i+j \leq n$, let $(x', y') = (x^*, y^*) + (e_{i+j} - e_i - e_j)$. If $i + j > n$, as $y^* \neq 0$, there exists k such that $y_k^* > 0$ and $k \geq i$, and therefore $1 \leq i + j - k \leq n$. Then let $(x', y') = (x^*, y^*) + (e_{i+j-k} - e_i - e_j - f_k)$. In either case, $(x', y') \in K(n, r)$ and $||(x', y')||_1 < ||(x^*, y^*)||_1$. Moreover, as (π, ρ, π_o) satisfy (SA2) and (SA3), in either case $\pi x' + \rho y' \leq \pi x^* + \rho y^* < \pi_o$, which contradicts the choice of (x^*, y^*). ■

Corollary 2.2.14. *Let $(\pi, \rho, \pi_o) \in T$, then $\pi x + \rho y \geq \pi_o$ is a valid inequality for $K(n, r)$.*

Note that by Lemma 2.2.7 an inequality satisfying pairwise subadditivity conditions and (EP2) is valid for $K(n, r)$. We next determine the extreme rays of T.

Lemma 2.2.15. *The extreme rays of T are $(f_k, 0) \in \mathbb{R}^{2n+1}$ for $n - r < k < n$.*

Proof. First note that $(f_k, 0)$ is indeed an extreme ray of T for $n - r < k < n$.

Let (π, ρ, π_o) be an extreme ray of T that is not equivalent to $(f_k, 0)$ for some $n - r < k < n$. Clearly $\pi_o = 0$. In this case, $\pi x + \rho y \geq 0$ is a valid inequality for $K(n, r)$. Using the arguments presented in the proof of Lemma 2.2.11, it is straightforward to establish that $\pi_i = 0$ for all $i \in N$, $\rho_k = 0$ for $1 \leq k \leq n - r$, and $\rho_k \geq 0$ for $n - r + 1 \leq k \leq n$. But then, (π, ρ, π_o) can be written as a conic combination of the rays $(f_k, 0)$ for $n - r < k < n$, a contradiction. ■

2.2.3 Facet characterization

Let

$$\mathcal{F} = \left\{(\pi^k, \rho^k, \pi_o^k)\right\}_{k=1}^{M}$$

be the set of coefficients of nontrivial facets of $K(n,r)$ with $\rho_n = 0$ and $\pi_o = 1$. Note that by Lemma 2.2.11 these two assumptions do not eliminate any nontrivial facets. Also, as $y_n \geq 0$ is a nontrivial facet, $\mathcal{F} \neq \emptyset$. By Lemma 2.2.9, $\mathcal{F} \subseteq T$.

We now proceed to prove Theorem 2.2.6 in two steps.

Lemma 2.2.16. *If $(\pi, \rho, \pi_o) \in \mathcal{F}$, then (π, ρ, π_o) is an extreme point of T.*

Proof. Assume that $(\pi, \rho, \pi_o) \in \mathcal{F}$ but is not an extreme point of T. It can be written as a convex combination of two distinct points in T. The normalization conditions $\rho_n = 0$ and $\pi_o = 1$ imply that any two distinct points in T represent two distinct valid inequalities for $K(n,r)$ in the sense that neither inequality can be obtained from the other by scaling or by adding multiples of the equation $\sum_{i=1}^n ix_i - \sum_{i=1}^n iy_i = r$. Therefore $\pi x + \rho y \geq \pi_o$ can be written as a combination of two distinct valid inequalities for $K(n,r)$, and therefore does not define a facet of $K(n,r)$. ∎

Lemma 2.2.17. *If (π, ρ, π_o) is an extreme point of T, then $(\pi, \rho, \pi_o) \in \mathcal{F}$.*

Proof. Let $(\hat{\pi}, \hat{\rho}, 1)$ be an extreme point of T. By Lemma 2.2.13, $(\hat{\pi}, \hat{\rho}, 1)$ defines a valid inequality for $K(n,r)$ and therefore it is implied by a conic combination of facet-defining inequalities plus a multiple of the equation $\sum_{i=1}^n ix_i - \sum_{i=1}^n iy_i = r$. In other words, there exists multipliers $\lambda \in \mathbb{R}_+^M$ and $\alpha \in \mathbb{R}$ such that

$$\begin{aligned}
\hat{\pi}_i &\geq \sum_{k=1}^{M} \lambda_k \pi_i^k + i\alpha, & \forall\, i \in N && (17)\\
\hat{\rho}_i &\geq \sum_{k=1}^{M} \lambda_k \rho_i^k - i\alpha, & \forall\, i \in N \setminus \{n\} && (18)\\
\hat{\rho}_n &= \sum_{k=1}^{M} \lambda_k \rho_n^k - n\alpha, & && (19)\\
1 &\leq \sum_{k=1}^{M} \lambda_k + r\alpha & && (20)
\end{aligned}$$

hold. The inequalities in (17) follow from the fact that $\hat{\pi}_i = \sum_{k=1}^{M} \lambda_k \pi_i^k + i\alpha + \mu_i e_i$, where $\mu_i \geq 0$, for $i = 1, \ldots, n$. The term $\mu_i e_i$ corresponds to adding μ_i times the trivial facet-defining inequality $x_i \geq 0$. The inequalities in (18) can be derived in a similar manner, from the fact that $y_i \geq 0$ for $i = 1, \ldots, n-1$ are trivial facet-defining inequalities. The equality in (19) is due to the fact that $y_n \geq 0$ is considered to be nontrivial. As $\hat{\rho}_n = 0$ and $\rho_n^k = 0$ for all nontrivial facet-defining inequalities, (19) implies that $\alpha = 0$. Furthermore, $\hat{\rho}_r = 1$ and $\rho_r^k = 1$, for all k, and combining (18) and (20) we can conclude that $\sum_{k=1}^{M} \lambda_k = 1$.

For any $i < r$, inequality (17) for i and $r - i$ combined with the equation (EP1) implies that

$$1 = \hat{\pi}_i + \hat{\pi}_{r-i} \geq \sum_{i=1}^{M} \lambda_k (\pi_i^k + \pi_{r-i}^k) = 1$$

which can hold only if $\hat{\pi}_i = \sum_{i=1}^{M} \lambda_k \pi_i^k$ for all $i < r$.

Similarly, for $i > r$, we use the equation (EP3) to observe that

$$1 = \hat{\pi}_i + \hat{\rho}_{i-r} \geq \sum_{i=1}^{M} \lambda_k (\pi_i^k + \rho_{i-r}^k) = 1$$

and therefore $\hat{\pi}_i = \sum_{i=1}^{M} \lambda_k \pi_i^k$ and $\hat{\rho}_{i-r} = \sum_{i=1}^{M} \lambda_k \rho_{i-r}^k$ for all $i > r$,.

Finally as $\hat{\rho}_i \geq \sum_{i=1}^{M} \lambda_k \rho_i^k$ for $i > n - r$, we can write $(\hat{\pi}, \hat{\rho}, 1)$ as a convex combination of points of $\mathcal{F}$ plus a conic combination of extreme rays of T. This can only be possible if $(\hat{\pi}, \hat{\rho}, 1) \in \mathcal{F}$. Thus, $(\hat{\pi}, \hat{\rho}, 1)$ is a nontrivial facet. ∎

As a final remark, it is interesting to note that conditions (R1) do not appear in the description of T even though they are necessary for any valid inequality. This is because conditions (R1) are implied by (SA1), (SA2) and (SA3). The proof is analogous to the proof of Lemma 2.2.13, so we just state it as an observation.

Observation 2.2.18. *Let* $(\pi, \rho, \pi_o) \in T$. *Then* $j\pi_i + i\rho_j \geq 0, \forall i, j \in N$.

We next show that coefficients of facet defining inequalities are bounded by small numbers.

Lemma 2.2.19. *Let* (π, ρ, π_o) *be an extreme point of* T, *then*

$$0 \leq \pi_k \leq \lceil k/r \rceil \quad \textit{and} \quad -\lceil k/r \rceil \leq \rho_k \leq \lceil n/r \rceil$$

for all $k \in N$.

Proof. Using Observation 2.2.18 with $j = n$ and the fact that $\rho_n = 0$, we have $\pi \geq 0$.

For $k < r$, combining inequality (EP1) $\pi_k + \pi_{r-k} \leq 1$ with $\pi \geq 0$ gives $\pi_k \leq \lceil k/r \rceil = 1$. For $k > r$, let $k = \lfloor k/r \rfloor r + q$, where $0 \leq q < r$. If $q = 0$, by (SA2) we have $\pi_k \leq \lfloor k/r \rfloor \pi_r = \lceil k/r \rceil$ since $\frac{k}{r}$ is integer. Similarly, if $q > 0$ we have $\pi_k \leq \lfloor k/r \rfloor \pi_r + \pi_q = \lfloor k/r \rfloor + \pi_q$, where $\pi_q \leq 1$. Therefore, $0 \leq \pi_k \leq \lceil k/r \rceil$.

The inequality (SA3) with $i = 1$ and $j = k$ implies that $\rho_k \geq -\pi_k$ and therefore $\rho_k \geq -\lceil k/r \rceil$ for all $k \in N$. If $k \leq n - r$, (EP3) implies that $\rho_k = \pi_r - \pi_{k+r} \leq 1 \leq \lceil n/k \rceil$. If $k > n - r$, then as (π, ρ, π_o) is an extreme point of T, at least one of (SA1) and (SA3) must hold with equality, in which case $\rho_k \leq \pi_i$, for some $i \in N$. Thus $\rho_k \leq \lceil n/r \rceil$ ∎

2.2.4 Pairwise subadditivity conditions

Next we give an example that demonstrates that using pairwise subadditivity conditions instead of the relaxed subadditivity conditions in the description of the coefficient polyhedron T leads to extreme points that do not give facet-defining inequalities for $K(n,r)$. To generate this example, we used PORTA [27] developed by Thomas Christof and Andreas Löbel at ZIB.

Example 2.2.20. *Consider $K(3,2)$ and denote an extreme point of T as $p = (\pi_1, \pi_2, \pi_3, \rho_1, \rho_2, \rho_3)$. We do not include π_o in the description of p since $\pi_o = 1$ for all points in T. It can be checked that T has the following two extreme points: $p_1 = (1/2, 1, 0, 1, 1/2, 0)$ and $p_2 = (1/2, 1, 3/2, -1/2, -1, 0)$. Point p_2 corresponds to the nonnegativity constraint for y_n.*

Let T' be a restriction of T obtained by replacing (SA3) *with the missing pairwise subadditivity conditions* (SA0) *and* (SA1'). *It can be checked that in addition to p_1 and p_2, the set T' has the following extreme point: $p_3 = (1/2, 1, 3/2, -1/2, 1/2, 0)$.*

Note that $p_2 \leq p_3$ and $p_2 \neq p_3$. Therefore $p_3^T(x,y) \geq 1$ is strictly implied by $p_2^T(x,y) \geq 1$. Therefore T' indeed has extreme points that do not correspond to facets of $K(3,2)$.

One could think that by using (SA2') and/or (SA2") it would be possible to obtain a characterization of either nontrivial facets or all facets except nonnegativity constraints. However this is not the case, since using (SA2') and/or (SA2") in addition to (SA0) and (SA1'), we get counterexamples similar to the above one.

2.3 Lifting facets of $P(n,r)$ when $n > r > 0$

Lifting is a general principle for constructing valid (facet-defining) inequalities for higher dimensional sets using valid (facet-defining) inequalities for lower dimensional sets. Starting with the early work of Gomory [55], the lifting approach was generalized by Padberg [88], Wolsey [101], Balas and Zemel [18] and Gu et. al [67], among others.

As $P(n,r)$ is an $n-1$ dimensional face of $K(n,r)$ obtained by setting n variables to their lower bounds, any facet-defining inequality for $P(n,r)$

$$\sum_{i=1}^{n-1} \bar{\pi}_i x_i \geq 1 \tag{21}$$

can be lifted to obtain one or more facet-defining inequalities of the form

$$\sum_{i=1}^{n-1} \bar{\pi}_i x_i + \pi'_n x_n + \sum_{i=1}^{n-1} \rho'_i y_i \geq 1 \tag{22}$$

for $K(n,r)$. We call inequality (22) a *lifted* inequality. Throughout this section we assume that $n > r > 0$, as this assumption holds true for $P(n,r)$.

For any valid inequality $\pi x + \rho y \geq \beta$ for $K(n,r)$, if $\pi_i = 0$ for some $i \in N$, then Lemma 2.2.3 implies that $\rho \geq 0$. This, in turn, implies that $\pi \geq 0$. Therefore, a (trivial) facet of $P(n,r)$ defined by a nonnegativity inequality can only yield a conic combination of nonnegativity inequalities for $K(n,r)$ when lifted. Consequently, we only consider nontrivial facets of $P(n,r)$ for lifting.

2.3.1 The restricted coefficient polyhedron $T^{\bar{\pi}}$

We start by reviewing the result of Gomory [55], previously presented in Section 2.1.1, that gives a complete characterization of the nontrivial facets (i.e., excluding the nonnegativity inequalities) of $P(n,r)$. In this description facet-defining inequalities are normalized so that y_n has a coefficient of zero and the righthand side is 1.

Theorem 2.1.1 (Gomory [55]). *The inequality $\bar{\pi} x \geq 1$ defines a nontrivial facet of $P(n,r)$, for $n > r > 0$, if and only if $\bar{\pi} \in \mathbb{R}^{n-1}$ is an extreme point of*

$$Q = \begin{cases} \pi_i + \pi_j \geq \pi_{(i+j)\,mod n} & \forall i,j \in \{1,\ldots,n-1\}, \\ \pi_i + \pi_j = \pi_r & \forall i,j \text{ such that } r = (i+j)\,mod n, \\ \pi_j \geq 0 & \forall j \in \{1,\ldots,n-1\}, \\ \pi_r = 1. & \end{cases}$$

Clearly nontrivial facets of $P(n,r)$ would give nontrivial facets of $K(n,r)$ when lifted. Using Gomory's characterization above, the lifting of nontrivial facets of $P(n,r)$ can be seen as a way of extending an extreme point $\bar{\pi}$ of Q to obtain an extreme point $(\bar{\pi}, \pi'_n, \rho', 0)$ of T.

Let $p = (\bar{\pi}, \pi'_n, \rho', 0)$ be an extreme point of T. Then, p also has to be an extreme point of the lower dimensional polyhedron

$$T^{\bar{\pi}} \;=\; T \cap \Big\{ \pi_i = \bar{\pi}_i, \;\forall i \in \{1, \ldots, n-1\} \Big\}.$$

Let $L = \{n - r + 1, \ldots, n - 1\}$.

Lemma 2.3.1. *If* (21) *defines a nontrivial facet of* $P(n, r)$*, then* $T^{\bar{\pi}} \neq \emptyset$ *and has the form*

$$T^{\bar{\pi}} = \begin{cases} \tau \;\geq\; \pi_n \;\geq\; 0 & \\ \rho_k \;\geq\; l_k & \forall\, k \in L \\ \rho_k + \pi_n \;\geq\; t_k & \forall\, k \in L \\ \rho_k - \pi_n \;\geq\; f_k & \forall\, k \in L \\ \pi_n + \rho_{n-r} \;=\; 1 & \\ \rho_n \;=\; 0 & \\ \rho_k \;=\; \bar{\pi}_{n-k} & \forall k \in \{1, \ldots, n - r - 1\} \\ \pi_i \;=\; \bar{\pi}_i & \forall i \in \{1, \ldots, n - 1\} \end{cases}$$

where numbers l_k, t_k, f_k *and* τ *can be computed easily using* $\bar{\pi}$*.*

Proof. First note that $\bar{\pi} \in Q$ and therefore $\bar{\pi}$ satisfies inequality (SA2) as well as equations (EP1) and (EP2). In addition, as $\bar{\pi}_i + \bar{\pi}_j = 1$ for all i, j such that $r = (i + j) \bmod n$, equality (EP3) can be rewritten as $\rho_i = \pi_{n-i}$ for all $1 \leq i \leq n - r$. Further, as $\bar{\pi}$ is subadditive (in the modular sense), inequalities (SA1) and (SA3) are satisfied for all $k \in \{1, \ldots, n - r - 1\}$. Therefore, setting

$$\pi_n = 0 \;\;\text{and}\;\;\; \rho_k = \begin{cases} \bar{\pi}_{n-k} & \text{if } k \in \{1, \ldots, n - r - 1\}, \\ 1 & \text{otherwise,} \end{cases}$$

produces a feasible point for $T^{\bar{\pi}}$, establishing that the set is not empty.

We next show that $T^{\bar{\pi}}$ has the proposed form, and also compute the values of l_k, t_k, f_k and τ.

Inequality (SA1)**:** If $i = n$, (SA1) becomes $\pi_n + \rho_k \geq \bar{\pi}_{n-k}$. If $i \neq n$, it becomes $\rho_k \geq \bar{\pi}_{i-k} - \bar{\pi}_i$, and therefore $\rho_k \geq l_k^1 = \max_{n>i>k}\{\bar{\pi}_{i-k} - \bar{\pi}_i\}$.

Inequality (SA2)**:** The only relevant case is $i + j = n$ when (SA2) becomes $\pi_n \leq \bar{\pi}_i + \bar{\pi}_{n-i}$. When combined, these inequalities simply become $\pi_n \leq \tau^1 = \min_{n>i>0}\{\bar{\pi}_i + \bar{\pi}_{n-i}\}$.

Inequality (SA3)**:** Without loss of generality assume $i \geq j$. We consider 3 cases.

Case 1, $k = n$: In this case the inequality reduces to $\pi_i + \pi_j \geq \pi_{i+j-n}$ which is satisfied by $\bar{\pi}$ when $i, j < n$. For $i = n$, this inequality simply becomes $\pi_n \geq 0$.

Case 2, $k < n$ and $i + j - k = n$: In this case the inequality becomes $\rho_k - \pi_n \geq -\pi_i - \pi_j$. If $i, j < n$ these inequalities can be combined to obtain $\rho_k - \pi_n \geq f_k^1 = \max_{1\leq i,j<n,\ k=i+j-n}\big\{ - \bar{\pi}_i - \bar{\pi}_j\big\}$. If $i = n$, then $j = k$ and the inequality becomes $\rho_k \geq -\bar{\pi}_k$.

Case 3, $k < n$ and $i+j-k < n$: If $i, j < n$ the inequality becomes $\rho_k \geq \pi_{i+j-k} - \pi_i - \pi_j$. These inequalities can be combined to obtain $\rho_k \geq l_k^2 = \max_{1\leq i,j<n:\ k<i+j<n+k}\big\{\bar{\pi}_{i+j-k} - \bar{\pi}_i - \bar{\pi}_j\big\}$. If $i = n$ then $j < n$, so the inequality becomes $\pi_n + \rho_k \geq \pi_{n+j-k} - \pi_j$, implying $\pi_n + \rho_k \geq t_k^1 = \max_{k>j}\big\{\bar{\pi}_{n+j-k} - \bar{\pi}_j\big\}$.

Therefore, combining these observations, it is easy to see that $T^{\bar{\pi}}$ has form given in Lemma 2.3.1 where l_k, t_k, f_k and τ are computed as follows:

$$
\begin{aligned}
l_k &= \max\big\{l_k^1, l_k^2, -\bar{\pi}_k\big\}, \\
t_k &= \max\big\{t_k^1, \bar{\pi}_{n-k}\big\} \;=\; \bar{\pi}_{n-k}, \\
f_k &= f_k^1, \\
\tau &= \min\big\{\tau^1,\ 1 - l_{n-r},\ (1 - f_{n-r})/2\big\}.
\end{aligned}
$$

The second equality in the description of t_k stating that $t_k = \bar{\pi}_{n-k}$ comes from the fact that $\bar{\pi}$ is sub-additive and therefore $\bar{\pi}_{n-k} + \bar{\pi}_j \geq \bar{\pi}_{n+j-k}$ for all $j < k$. The $1 - l_{n-r}$ and $(1 - f_{n-r})/2$ terms in the last equation come from using the bounds on ρ_{n-r} together with the equations and inequalities of $T^{\bar{\pi}}$ to obtain implied bounds for

π_n. ∎

We next make a simple observation that will help us show that $T^{\bar{\pi}}$ has a polynomial number of extreme points.

Lemma 2.3.2. *If $p = (\bar{\pi}, \pi'_n, \rho', 0)$ is an extreme point of $T^{\bar{\pi}}$, then*

$$\rho'_k = \max\left\{l_k, t_k - \pi'_n, f_k + \pi'_n\right\}$$

for all $k \in L$.

Proof. Assume that the claim does not hold for some $k \in L$ and let $\theta = \max\left\{l_k, t_k - \pi'_n, f_k + \pi'_n\right\}$. As $p \in T^{\bar{\pi}}$, $\rho'_k \geq \theta$ and therefore $\epsilon = \rho'_k - \theta > 0$. In this case, two distinct points in $T^{\bar{\pi}}$ can be generated by increasing and decreasing the associated coordinate of p by ϵ, establishing that p is not an extreme point, a contradiction. ∎

We next characterize the set of possible values π'_n can take at an extreme point of $T^{\bar{\pi}}$.

Lemma 2.3.3. *Let $p = (\bar{\pi}, \pi'_n, \rho', 0)$ be an extreme point of $T^{\bar{\pi}}$, if $\pi'_n \notin \{0, \tau\}$, then*

$$\pi'_n \ \in \ \Lambda \ := \ \left(\bigcup_{k \in L_1} \left\{t_k - l_k, l_k - f_k\right\}\right) \bigcup \left(\bigcup_{k \in L_2} \left\{(t_k - f_k)/2\right\}\right)$$

where $L_1 = \{k \in L \,:\, t_k + f_k < 2l_k\}$ and $L_2 = L \setminus L_1$.

Proof. Note that the description of $T^{\bar{\pi}}$ consists of $3(r-1)$ inequalities that involve ρ_k variables and upper and lower bound inequalities for π'_n. Being an extreme point, p has to satisfy r of these inequalities as equality. Therefore, if $\pi'_n \notin \{0, \tau\}$ then there exists an index $k \in L$ for which at least two of the following inequalities

$$\rho_k \ \geq \ l_k \qquad \text{(a)}$$

$$\rho_k + \pi_n \ \geq \ t_k \qquad \text{(b)}$$

$$\rho_k - \pi_n \ \geq \ f_k \qquad \text{(c)}$$

hold as equality. Clearly, this uniquely determines the value of π'_n and therefore

$$\pi'_n \in \Lambda^+ = \bigcup_{k \in L} \Big\{ t_k - l_k, l_k - f_k, (t_k - f_k)/2 \Big\}.$$

Furthermore, for any fixed $k \in L$, adding inequalities (b) and (c) gives $2\rho_k \geq t_k + f_k$. Therefore if $t_k + f_k > 2l_k$, then inequality (a) is implied by inequalities (b) and (c) and it cannot hold as equality. Similarly, if $t_k + f_k < 2l_k$, then inequalities (b) and (c) cannot hold simultaneously. Finally, if $t_k + f_k = 2l_k$, then it is easy to see that $t_k - l_k = l_k - f_k = (t_k - f_k)/2$. Therefore letting

$$L_1 = \{k \in S : t_k + f_k < 2l_k\}, \quad L_2 = L \setminus L_1$$

proves the claim. ∎

Combining the previous lemmas, we have the following result:

Theorem 2.3.4. *Given a nontrivial facet-defining inequality* (21) *for* $P(n, r)$*, there are at most* $2r$ *lifted inequalities that define facets of* $K(n, r)$*.*

Proof. The set L has $r - 1$ members and therefore together with 0 and τ, there are at most $2r$ possible values for π'_n in a facet-defining lifted inequality (22). As the value of π'_n uniquely determines the remaining coefficients in the lifted inequality, by Lemma 2.3.2, the claim follows. ∎

In general, determining all possible lifted inequalities is a hard task. However, the above results show that obtaining all possible facet-defining inequalities lifted from a facet of $P(n, r)$ is straightforward and can be performed in polynomial time. In Figure 1, we display two facets of $K(16, 13)$ obtained by lifting the same facet of $P(16, 13)$. The facet coefficient of each variable is a function of its coefficient in $K(n, r)$, that is, variable x_i will have a facet coefficient $f(i)$ and variable y_i will have a facet coefficient $f(-i)$. We marked all of these coefficients as discs in the figure. Also, note that the displayed functions are obtained by interpolating the facet coefficients; we explain their significance in Section 2.8.

Note that the second facet has the same coefficient values for $x_i, i = 1, \ldots, n-1$ as the first, and a larger coefficient for x_n, and therefore it has coefficient values for the y variables which are less than (sometimes strictly less than) the corresponding coefficients for the first facet. To make comparison easier, we plot the first facet in dashed lines behind the second facet. Later, in Figure 2, we display facets of $K(16, 13)$ which cannot be obtained by lifting.

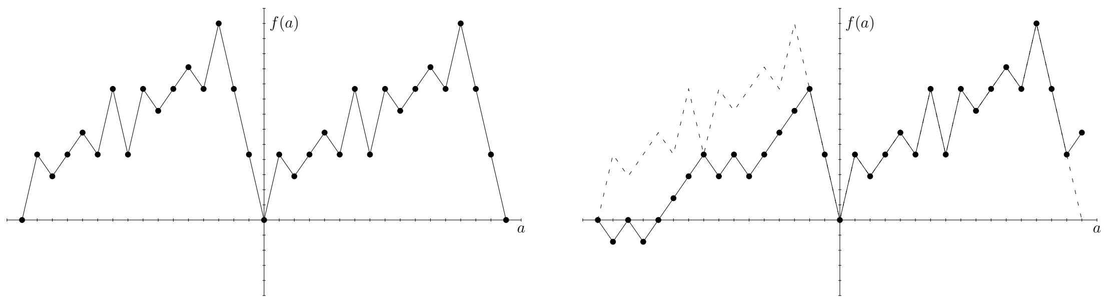

Figure 1: Example of two facets of $K(16, 13)$ obtained by lifting

2.3.2 Sequential lifting

Sequential lifting is a procedure that introduces the missing variables one at a time to obtain the lifted inequality. Depending on the order in which missing variables are lifted, one obtains different inequalities. We note that not all lifted facets can be obtained by sequential lifting, see [85].

In Lemma 2.3.1 we have established that (regardless of the lifting sequence) $\rho'_k = \pi_{n-k}$ for $k \in K' = \{1, \ldots, n-r-1\}$. Furthermore, in Lemma 2.3.2 we established that the coefficient of variable x_n determines the coefficients of the remaining variables. Therefore, given a lifting sequence, if K'' denotes the set of indices of y_k variables that are lifted after either one of x_n or y_{n-r}, then the lifted inequality does not depend on the order in which variables y_k for $k \in K'$ or $k \in K''$ are lifted. We next present a result adapted from Wolsey [101].

Lemma 2.3.5 (Wolsey [101]). *Given a facet-defining inequality* (21) *for* $P(n, r)$ *and a lifting sequence for the variables* x_n *and* y_i *for* $i = 1, \ldots, n-1$, *the sequential lifting procedure produces a facet-defining inequality for* $K(n, r)$.

Furthermore, when a variable is lifted, it is assigned the smallest value among all coefficients for that variable in lifted inequalities having the same coefficients for previously lifted variables.

Therefore, given a nontrivial facet-defining inequality $\bar{\pi}x \geq 1$ for $P(n, r)$, the lifting coefficient of the variable currently being lifted can simply be computed by solving a linear program that minimizes the coefficient of that variable subject to the constraint that $T^{\bar{\pi}}$ has a point consistent with the coefficients of the variables that have already been lifted.

We conclude this section by showing that the simple mapping discussed in the Introduction (see inequality (14)) leads to a particular lifted inequality.

Lemma 2.3.6. *If variable* x_n *is lifted before all* y_k *for* $k \in \{n-r, \ldots, n-1\}$, *then independent of the rest of the lifting sequence the lifted inequality is*

$$\sum_{i=1}^{n-1} \bar{\pi}_i x_i + \sum_{i=1}^{n-1} \bar{\pi}_{n-i} y_i \geq 1.$$

Proof. By Lemma 2.3.5, we know that variable x_n will be assigned the smallest possible coefficient in the lifted inequality. As $\pi_n \geq 0$ in the description of $T^{\bar{\pi}}$ and as $T^{\bar{\pi}}$ does contain a point with $\pi_n = 0$ (as described in the proof of Lemma 2.3.1), we conclude that $\pi_n = 0$ in the lifted inequality.

Therefore, by Lemma 2.3.2, $\rho'_k = \min\{l_k, \bar{\pi}_{n-k}, f_k\}$ and we need to show that $\bar{\pi}_{n-k} \geq l_k, t_k$ for all $k \in \{1, \ldots, n-1\}$. First, observe that $0 \geq t_k$ and therefore $\bar{\pi}_{n-k} \geq t_k$ for all $k \in \{1, \ldots, n-1\}$. Finally, recall that $\bar{\pi}$ is subadditive (in the modular sense), and therefore $\bar{\pi}_{n-k} + \bar{\pi}_i \geq \bar{\pi}_{i-k}$ for all $n > i > k$ and $\bar{\pi}_{n-k} + \bar{\pi}_i + \bar{\pi}_j \geq \bar{\pi}_{i+j-k}$ for all $n > i, j$ and $n + k > i + j > k$. ∎

The first facet in Figure 1 has the form given in the previous lemma; it is obtained from a facet of $P(n,r)$ by first lifting x_n.

2.4 *Mixed integer rounding inequalities for* $K(n,r)$ *for* $r > 0$

In this section we study MIR inequalities in the context of $K(n,r)$. Our analysis also provides an example that shows that lifting facets of $P(n,r)$ cannot give all facets of $K(n,r)$. Throughout, we will use the notation $\hat{x} := x - \lfloor x \rfloor$ and $(x)^+ = \max\{x, 0\}$. Recall that, for a general single row system of the form: $\{w \in \mathbb{Z}_+^p : \sum_{i=1}^p a_i w_i = b\}$ where $\hat{b} > 0$, the MIR inequality is:

$$\sum_{i=1}^{p} \left(\lfloor a_i \rfloor + \min\left(\hat{a}_i / \hat{b}, 1 \right) \right) w_i \geq \lceil b \rceil .$$

We define the $\frac{1}{t}$-MIR (for $t \in \mathbb{Z}_+$) to be the MIR inequality obtained from the following equivalent representation of $K(n,r)$:

$$\left\{ (x,y) \in \mathbb{Z}_+^n \times \mathbb{Z}_+^n : \sum_{i=1}^{n} (i/t) x_i - \sum_{i=1}^{n} (i/t) y_i = r/t \right\}.$$

Lemma 2.4.1. *Given $t \in \mathbb{Z}$ such that $2 \leq t \leq n$, the $\frac{1}{t}$-MIR inequality*

$$\sum_{i=1}^{n} \left(\left\lfloor \frac{i}{t} \right\rfloor + \min\left(\frac{i \mod t}{r \mod t}, 1 \right) \right) x_i + \sum_{i=1}^{n} \left(- \left\lceil \frac{i}{t} \right\rceil + \min\left(\frac{(t-i) \mod t}{r \mod t}, 1 \right) \right) y_i \geq \left\lceil \frac{r}{t} \right\rceil$$

is facet-defining for $K(n,r)$ provided that $r/t \notin \mathbb{Z}$.

Proof. Let $\pi x + \rho y \geq \pi_o$ denote the $\frac{1}{t}$-MIR inequality and let F denote the set of points that are on the face defined by this inequality. Also let q^i denote $i \bmod t$. Using this definition $i = q^i + \lfloor i/t \rfloor t$.

For $i \in N \setminus \{1, t\}$, consider the point

$$w^i = e_i + (\lfloor r/t \rfloor - \lfloor i/t \rfloor)^+ e_t + (\lfloor i/t \rfloor - \lfloor r/t \rfloor)^+ f_t + (q^r - q^i)^+ e_1 + (q^i - q^r)^+ f_1$$

and observe that $w^i \in K(n,r)$. Moreover,

$$(\pi,\rho)^T w^i = (\lfloor i/t \rfloor + \min\{q^i/q^r, 1\}) + (\lfloor r/t \rfloor - \lfloor i/t \rfloor) + \frac{(q^r - q^i)^+}{q^r} = \lfloor r/t \rfloor + 1 = \pi_o$$

and therefore $w^i \in F$. Similarly, let $-i = -\lceil i/t \rceil + m^i$, with $0 \le m^i < t$ and consider the point

$$z^i = f_i + (\lfloor r/t \rfloor + \lceil i/t \rceil)e_t + (q^r - m^i)^+ e_1 + (m^i - q^r)^+ f_1$$

for $i \in N \setminus \{1,t\}$. Clearly $x^i \in K(n,r)$. Furthermore,

$$(\pi,\rho)^T z^i = (-\lceil i/t \rceil + \min\{m/q^r, 1\}) + (\lfloor r/t \rfloor + \lceil i/t \rceil) + \frac{(q^r - m)^+}{q^r} = \lfloor r/t \rfloor + 1 = \pi_o$$

and therefore $z^i \in F$.

Additionally the following three points are also in $K(n,r) \cap F$: $u^1 = \lfloor r/t \rfloor e_t + q^r e_1$, $u^2 = (\lfloor r/t \rfloor + 1)e_t + (t - q^r)f_1$, $u^3 = (\lfloor r/t \rfloor + 1)e_t + f_t + q^r e_1$. Therefore, $\{u^i\}_{i=1}^3 \cup \{w^i\}_{i \in N \setminus \{1,t\}} \cup \{z^i\}_{i \in N \setminus \{1,t\}}$ is a set of $2n-1$ affinely independent points in F. ∎

We next show that $\frac{1}{t}$-MIR inequalities are not facet-defining unless they satisfy the conditions of Theorem 2.4.1. First, observe that the inequality is not defined if t divides r. Next, we show that the $1/n$-MIR inequality dominates all $\frac{1}{t}$-MIR inequalities with $t > n$.

Lemma 2.4.2. *If $t > n$, then the $\frac{1}{t}$-MIR inequality is not facet-defining for $K(n,r)$.*

Proof. When $t > n$, the $\frac{1}{t}$-MIR inequality becomes

$$\sum_{i \in N} \min\{i/r, 1\} x_i - \sum_{i: i > t-r} \left(1 - \frac{t-i}{r}\right) y_i \ge 1$$

and is dominated by the $1/n$-MIR inequality:

$$\sum_{i \in N} \min\{i/r, 1\} x_i - \sum_{i: i > n-r} \left(1 - \frac{n-i}{r}\right) y_i \ge 1.$$

∎

We conclude this section by showing that $\frac{1}{t}$-MIR inequalities give facets that cannot be obtained by lifting facets of $P(n,r)$.

Theorem 2.4.3. *For $n \geq 9$ and $n-2 \geq r > 0$, there are facet-defining inequalities for $K(n,r)$ that cannot be obtained by lifting facet-defining inequalities for $P(n,r)$.*

Proof. When $0 < r \leq n-4$, consider the facet induced by the $\frac{1}{n-2}$-MIR inequality $\pi x + \rho y \geq \pi_o$ where

$$\rho_n = -2 + \min\left(\frac{n-4}{r}, 1\right) = -1.$$

We subtract $\frac{1}{n}$ times $\sum_{i=1}^{n} ix_i - \sum_{i=1}^{n} iy_i = r$ from the inequality to obtain $\pi' x + \rho' y \geq \pi'_o$ where $\rho'_n = 0$ and therefore the inequality satisfies the normalization condition (NC1). Notice that

$$\begin{aligned} \pi'_{r+1} + \pi'_{n-1} &= \left(1 - \frac{r+1}{n}\right) + \left(1 + \frac{1}{r} - \frac{n-1}{n}\right) \\ &= 1 - \frac{r}{n} + \frac{1}{r} \end{aligned}$$

whereas $\pi'_r = 1 - r/n < \pi'_{r+1} + \pi'_{n-1}$. This proves the claim for $0 < r \leq n-4$ as all facet-defining inequalities for $P(n,r)$ must satisfy $\pi'_{r+1} + \pi'_{n-1} = \pi'_r$.

For $r \in \{n-3, n-2\}$, the $\frac{1}{r-1}$-MIR inequality provides such an example. ∎

For $r = n-1$, all points in T automatically satisfy all equations in Q. Therefore, any given facet-defining inequality of $K(n,r)$ can be obtained by lifting a point in Q. However, this point is not necessarily an extreme point of Q.

2.5 Polyhedral analysis of $K(n,0)$

Observe that $LK(n,0)$, the linear relaxation of $K(n,0)$, is a pointed cone (as it is contained in the nonnegative orthant) and has a single extreme point $(x,y) = (0,0)$. Therefore $LK(n,0)$ equals its integer hull, i.e., $LK(n,0) = K(n,0)$. In Lemma 2.2.3, we characterized the extreme rays of $K(n,r)$ and thereby showed that the recession cone of $K(n,r)$ is generated by the vectors $\{r_{ij}\}$. But the recession cone of $K(n,r)$ for some $r > 0$ is just $K(n,0)$. Therefore, $LK(n,0)$ is generated by the vectors $\{r_{ij}\}$, and the next result follows.

Theorem 2.5.1. *The inequality $\pi x + \rho y \geq \pi_o$ is facet-defining for $K(n,0)$ if and only if (π, ρ, π_o) is a minimal face of*

$$T_o = \begin{cases} j\pi_i + i\rho_j \geq 0 & ,\forall i,j \in N, \\ \pi_o = 0. \end{cases}$$

In his work on MCGP, Gomory also studied $P(n,0)$, the convex hull of non-zero integral solutions that satisfy $\sum_{i=1}^{n-1} ix_i - ny_n = 0$, and showed that an analog of Theorem 2.1.1 also holds for $r = 0$. For the sake of completeness, we now consider a similar modification of $K(n,0)$ and study the set:

$$\bar{K}(n,0) = conv\left\{(x,y) \in \mathbb{Z}^n_+ \times \mathbb{Z}^n_+ : \sum_{i=1}^{n} ix_i - \sum_{i=1}^{n} iy_i = 0, (x,y) \neq 0\right\}.$$

For this case it will be more convenient to consider all nonnegativity inequalities (including $y_n \geq 0$) as trivial. We will next prove that all nontrivial facet-defining inequalities for $\bar{K}(n,0)$ are given by the extreme points of $\bar{T}_o$ defined below.

Definition 2.5.2. *We define $\bar{T}_o \subseteq \mathbb{R}^{2n+1}$ as the set of points that satisfy the following linear equalities and inequalities:*

$$\pi_i + \rho_j \geq \pi_{i-j}, \qquad \forall i,j \in N, \quad i > j, \tag{SA1}$$

$$\pi_i + \rho_j \geq \rho_{j-i}, \qquad \forall i,j \in N, \quad i < j, \tag{SA1'}$$

$$\pi_i + \rho_i = \pi_o, \qquad \forall i \in N, \tag{EP1-R0}$$

$$\pi_o = 1, \tag{N1-R0}$$

$$\rho_n = 0. \tag{N2-R0}$$

It is easy to see that the conditions (SA1), (SA1'), and (EP1-R0) are together equivalent to the conditions (SA2), (SA2'), (SA2") and (EP1-R0). For example, replacing π_i by $\pi_o - \rho_i$ and π_{i-j} by $\pi_o - \rho_{i-j}$ in (SA1), we get (SA2'). Therefore, a point in $\bar{T}_o$ satisfies all the pairwise subadditivity conditions given in the previous section.

Lemma 2.5.3. *If $(\pi, \rho, \pi_o) \in \bar{T}_o$ then $\pi x + \rho y \geq \pi_o$ is a valid inequality for $\bar{K}(n, 0)$.*

Proof. Suppose $\pi x + \rho y \geq \pi_o$ is not valid for $\bar{K}(n, 0)$. Then, let $(x^*, y^*) \in \bar{K}(n, 0)$ be the integer point in $\bar{K}(n, 0)$ with smallest L_1 norm such that $\pi x^* + \rho y^* < \pi_o$. Note that any point in $\bar{K}(n, 0)$ has L_1 norm 2 or more.

If $||(x^*, y^*)||_1 = 2$, then $(x^*, y^*) = e_i + f_i$ for some $i \in N$, but by (EP1-R0), $\pi x^* + \rho y^* = \pi_o$, which is a contradiction. So we may assume that $||(x^*, y^*)||_1 > 2$.

As $(x^*, y^*) \in \bar{K}(n, 0)$, there exists $i, j \in N$ such that $x_i^* > 0$ and $y_j^* > 0$. Let

$$(x', y') = (x^*, y^*) - e_i - f_j + \begin{cases} f_{j-i} & \text{if } i < j, \\ 0 & \text{if } i = j, \\ e_{i-j} & \text{if } i > j. \end{cases}$$

Clearly, (x', y') is an integer point in $\bar{K}(n, 0)$ and $||(x', y')||_1 \leq ||(x^*, y^*)||_1 - 1$. Furthermore, as (π, ρ, π_o) satisfies (SA1), (SA1') and (EP1-R0), we also have $\pi x' + \rho y' \leq \pi x^* + \rho x^* < \pi_o$, which contradicts the choice of (x^*, y^*). ∎

Theorem 2.5.4. *Consider an inequality $\pi x + \rho y \geq \pi_o$ with $\rho_n = 0$. It defines a nontrivial facet of $\bar{K}(n, 0)$ if and only if it the following conditions hold: (i) $\pi_o > 0$, and, (ii) $(\pi, \rho, \pi_o)/\pi_o$ is an extreme point of $\bar{T}_o$.*

Proof. ($\Rightarrow$):

Let $\pi x + \rho y \geq \pi_o$ define a nontrivial facet of $\bar{K}(n, 0)$. We first show that (π, ρ, π_o) satisfies (SA1), (SA1') and (EP1-R0), and can be assumed to satisfy (N1-R0) and (N2-R0).

(SA1) - (SA1'): Let i, j be indices such that $i, j \in N$ and $i > j$. Let $z = (x^*, y^*)$ be an integral point lying on the above facet such that $x_{i-j}^* > 0$. As $z + (e_i + f_j - e_{i-j})$ belongs to $\bar{K}(n, 0)$, (SA1) is true. The proof of (SA1') is similar.

(EP1-R0): Let $\gamma = (\pi, \rho)$. Let $z^1 = (x^1, y^1)$ and $z^2 = (x^2, y^2)$ be integral points lying on the facet such that $x_i^1 > 0$ and $y_i^2 > 0$. Then $z = z^1 + z^2 - e_i - f_i \in \bar{K}(n, 0)$, and therefore $\gamma z = \gamma z^1 + \gamma z^2 - \pi_i - \rho_i = 2\pi_o - \pi_i - \rho_i \geq \pi_o \Rightarrow \pi_i + \rho_i \leq \pi_o$. But as $e_i + f_i \in \bar{K}(n,0)$, $\pi_i + \rho_i \geq \pi_o$ and the result follows.

(N1-R0): Assume $\pi_o < 0$, and let (x^*, y^*) be an integral point in $\bar{K}(n,0)$ satisfying $\pi x^* + \rho y^* = \pi_o$. As $\alpha(x^*, y^*) \in \bar{K}(n,0)$ for any positive integer α, whereas $\pi\alpha x^* + \rho\alpha y^* = \alpha\pi_o < \pi_o$, we obtain a contradiction to the fact that points in $\bar{K}(n,0)$ satisfy $\pi x + \rho y \geq \pi_o$.

If $\pi_o = 0$, then (EP1-R0) implies that $\rho_i = -\pi_i$ for all $i \in N$. This fact, along with (SA1) and (SA1') implies that $\pi_i = i\pi_1$ and $\rho_i = -i\pi_1$ for all $i \in N$. But then $\pi x + \rho y \geq \pi_o$ is the same as $\pi_1(\sum_{i=1}^n ix_i - \sum_{i=1}^n iy_i) \geq 0$, and therefore cannot define a proper face of $\bar{K}(n,0)$. Therefore, for any nontrivial facet, $\pi_o > 0$ and can be assumed to be 1 by scaling.

We can assume, by subtracting appropriate multiples of $\sum_{i=1}^n ix_i - \sum_{i=1}^n iy_i = 0$ from $\pi x + \rho y \geq \pi_o$, that (N2-R0) holds.

Therefore (π, ρ, π_o) can be assumed to be contained in $\bar{T}_o$. If it is not an extreme point of $\bar{T}_o$, it can be written as a convex combination of two distinct points of $\bar{T}_o$, different from itself, each of which defines a valid inequality for $\bar{K}(n,0)$ (by Lemma 2.5.3). As the normalization conditions (N1-R0) and (N2-R0) mean that each nontrivial facet-defining inequality corresponds to a unique point in $\bar{T}_o$, this implies that (π, ρ, π_o) is an extreme point of $\bar{T}_o$.

$(\Leftarrow)$:

Let $\mathcal{F} = \left\{(\pi^k, \rho^k, \pi_o^k)\right\}_{k=1}^M$ be the set of all nontrivial facets of $\bar{K}(n,0)$ such that $\rho_n^k = 0$ and $\pi_o^k = 1$. Let $(\pi, \rho, 1)$ be an extreme point of $\bar{T}_o$. By Lemma 2.5.3, $(\pi, \rho, 1)$ defines a valid inequality for $\bar{K}(n,0)$, and therefore there exist numbers λ^k and α

such that

$$\begin{aligned}
\alpha i + \textstyle\sum_{k=1}^{M} \lambda_k \pi_i^k &\leq \pi_i, \qquad \forall i \in N\\
-\alpha i + \textstyle\sum_{k=1}^{M} \lambda_k \rho_i^k &\leq \rho_i, \qquad \forall i \in N\\
\textstyle\sum_{k=1}^{M} \lambda_k &\geq 1,\\
\lambda \geq 0, \qquad & \alpha \text{ free.}
\end{aligned}$$

(EP1-R0) implies that for all $i \in N$, $1 = \pi_i + \rho_i \geq \sum_{k=1}^{M} \lambda_k(\pi_i^k + \rho_i^k)$ and since all nontrivial facets also satisfy (EP1-R0), we can conclude that $\sum_{k=1}^{M} \lambda_k = 1$. Therefore, $1 = \pi_i + \rho_i \geq \sum_{k=1}^{M} \lambda_k(\pi_i^k + \rho_i^k) = 1$ and hence $\pi_i = \sum_{k=1}^{M} \lambda_k \pi_i^k$ and $\rho_i = \sum_{k=1}^{M} \lambda_k \rho_i^k$. In other words, $(\pi, \rho, 1)$ can be expressed as a convex combination of the elements of $\mathcal{F}$, each of which is contained in $\bar{T}_o$. This is possible only if $(\pi, \rho, 1)$ is itself an element of $\mathcal{F}$, i.e., it defines a nontrivial facet of $\bar{K}(n, 0)$. ■

As $P(n, 0)$ is a lower dimensional face of $\bar{K}(n, 0)$, it is possible to lift facet-defining inequalities of $P(n, 0)$ to obtain facet-defining inequalities for $\bar{K}(n, 0)$. From the description of $\bar{T}_o$, any nontrivial facet-defining inequality $\pi x + \rho y \geq \pi_o$ of $\bar{K}(n, 0)$ satisfies $\rho_i = 1 - \pi_i$. Thus, if $\sum_{i=1}^{n-1} \bar{\pi}_i x_i \geq 1$ defines a nontrivial facet of $P(n, 0)$, there is a unique way to lift this inequality to obtain a facet-defining inequality for $K(n, 0)$, namely:

$$x_n + \sum_{i=1}^{n-1} \bar{\pi}_i x_i + \sum_{i=1}^{n-1} (1 - \bar{\pi}_i) y_i \geq 1.$$

Next, we show that when $n = 3$ not all facet-defining inequalities for $\bar{K}(n, 0)$ can be obtained via lifting. Remember that, after normalization, coefficients of all facet-defining inequalities for $P(n, 0)$ are between 0 and 1.

Example 2.5.5. *Consider $\bar{K}(3, 0)$ and note that all $(x, y) \in \bar{K}(3, 0)$ satisfy $\sum_{i=1}^{3} x_i \geq 1$. Adding $\sum_{i=1}^{3} i x_i - \sum_{i=1}^{3} i y_i = 0$ to this inequality yields*

$$2x_1 + 3x_2 + 4x_3 - y_1 - 2y_2 - 3y_3 \geq 1. \tag{23}$$

Dividing (23) by 2 and writing the MIR inequality gives $x_1 + 2x_2 + 2x_3 - y_2 - y_3 \geq 1$, *which becomes*

$$\frac{2}{3}x_1 + \frac{4}{3}x_2 + x_3 + \frac{1}{3}y_1 - \frac{1}{3}y_2 \geq 1 \tag{24}$$

after normalization. Denoting feasible points $p \in K(3,0)$ *as* $p = (x_1, x_2, x_3, y_1, y_2, y_3)$, *note that the following 5 affinely independent points satisfy* (24) *as equality:* $p_1 = (1,0,0,1,0,0)$, $p_2 = (0,1,0,0,1,0)$, $p_3 = (0,0,1,0,0,1)$, $p_4 = (2,0,0,0,1,0)$ *and* $p_5 = (0,0,1,1,1,0)$ *and therefore* (24) *defines a facet of* $\bar{K}(3,0)$.

Notice that (24) *cannot be obtained via lifting as the coefficient of* x_2 *is greater than 1.*

Lemma 2.5.6. *For any* $n \geq 3$, $\bar{K}(n,0)$ *has at least one facet that cannot be obtained by lifting a facet of* $P(n,0)$ *.*

Proof. For $n = 3$ the example above proves the claim so we consider $n \geq 4$. Let

$$\pi_i = \begin{cases} \frac{i+2}{n} & \text{if } i < n \\ 1 & \text{if } i = n \end{cases} \qquad \rho_i = \begin{cases} \frac{n-i-2}{n} & \text{if } i < n \\ 0 & \text{if } i = n \end{cases} \tag{25}$$

and notice that $\pi_{n-1} > 1$. We next show that $p = (\pi, \rho, 1) \in \bar{T}_o$. Clearly p satisfies equations (EP1-R0), (N1-R0) and (N2-R0). Using $(i+2)/n \geq \pi_i \geq i/n$ and $(n-i)/n \geq \rho_i \geq (n-i-2)/n$ for all $i \in N$, note that

$$\pi_i + \rho_j - \pi_{i-j} \geq i/n + (n-j-2)/n - (i-j+2)/n \geq 1 - 4/n \geq 0$$

and therefore p satisfies inequality (SA1). Finally, if we have that $i, j \in N$ and $j > i$, then $j - i < n$ which implies that $\rho_{j-i} = (n-j+i-2)/n$ and hence:

$$\pi_i + \rho_j - \rho_{j-i} \geq i/n + (n-j-2)/n - (n-j+i-2)/n \geq 0$$

Therefore p satisfies inequality (SA1') as well and it is indeed contained in T_o. As T_o is a bounded polyhedron, it must therefore have an extreme point $(\bar{\pi}, \bar{\rho}, 1)$ that has $\bar{\pi}_{n-1} > 1$. Clearly, the inequality corresponding to this extreme point cannot be obtained by lifting a facet of $P(n,0)$. ∎

2.6 Separating over $K(n,r)$

Let $LK(n,r)$ be the linear relaxation of $K(n,r)$. We define the separation problem over $K(n,r)$ as follows: given $(x^*, y^*) \in LK(n,r)$, either verify that $(x^*, y^*) \in K(n,r)$ or find a violated valid inequality for $K(n,r)$. Note that the condition that $(x^*, y^*) \in LK(n,r)$ is easy to satisfy.

From Theorems 2.2.6, 2.5.1 and 2.5.4, it follows that any point $(x^*, y^*) \in LK(n,r)$, with $0 < r \leq n$, can be separated from $K(n,r)$ by minimizing $\pi x^* + \rho y^*$ over T. Therefore, in this case the separation problem can be solved in polynomial-time using an LP with $O(n)$ variables and $O(n^3)$ constraints. Similarly, by Theorem 2.5.4, one can separate a point from $\bar{K}(n,0)$ using an LP with $O(n)$ variables and $O(n^2)$ constraints.

We next show that the separation problem over $K(n,r)$ can also be solved for any $r > 0$ using an LP with $O(n^2)$ constraints.

Theorem 2.6.1. *Given* $(x^*, y^*) \in LK(n,r)$*, where* $r > 0$*, the separation problem over* $K(n,r)$ *can be solved in time polynomial in* $\max\{n,r\}$ *using an LP with* $O(\max\{n,r\})$ *variables and* $O(\max\{n,r\}^2)$ *constraints.*

Proof. First we consider the case $r \leq n$. Let T' be a restriction of T obtained by replacing the relaxed subadditivity conditions (SA1)-(SA3) with the pairwise subadditivity conditions (SA0), (SA1), (SA1'), (SA2), and (SA2'). Due to Lemma 2.2.7, $T' \subseteq T$ and therefore if $(\pi, \rho) \in T'$, then $\pi x + \rho y \geq 1$ is a valid inequality for $K(n,r)$. In addition, by Lemma 2.2.9, if $\pi x + \rho y \geq 1$ defines a nontrivial facet of $K(n,r)$, then $(\pi, \rho) \in T'$.

Let (π', ρ') be an optimal solution of

$$z^* = \min\{\pi x^* + \rho y^* : (\pi, \rho) \in T' \}.$$

If $z^* < 1$ then $\pi' x + \rho' y \geq 1$ is a violated valid inequality. If, on the other hand, $z^* \geq 1$, then (x^*, y^*) satisfies all facet-defining inequalities and therefore belongs to $K(n,r)$.

Next, assume that $r > n$. Define $(x', y') \in \mathbb{R}^r \times \mathbb{R}^r$ such that $x'_i = x^*_i; y'_i = y^*_i$, for $i = 1, \ldots, n$ and $x'_i = y'_i = 0$, for $i = n+1, \ldots, r$. As $(x^*, y^*) \in K(n, r) \iff (x', y') \in K(r, r)$, separation can be done using $K(r, r)$. If a violated inequality is found, then it can simply be turned into a valid inequality for $K(n, r)$ by dropping the extra coefficients. ■

2.7 *Mixed-integer extension*

Consider the mixed-integer extension of $K(n, r)$:

$$K'(n,r) = conv\left\{(v_+, v_-, x, y) \in \mathbb{R}^2_+ \times \mathbb{Z}^{2n}_+ : v_+ - v_- + \sum_{i=1}^{n} ix_i - \sum_{i=1}^{n} iy_i = r\right\}$$

where $n, r \in \mathbb{Z}$ and $n \geq r > 0$. As in the case of the mixed-integer extension of MCGP studied by Gomory and Johnson [57], the facets of $K'(n, r)$ can easily be derived from the facets of $K(n, r)$ when r is an integer. To prove such a result, we introduce a few definitions, and also state some easy results without proof.

The dimension of $K'(n, r)$ is $2n + 1$, that is, one less than the number of variables. The inequalities $x_i \geq 0$ and $y_i \geq 0$, for $i = 1, \ldots, n$, and $v_+ \geq 0$ and $v_- \geq 0$ define facets of $K'(n, r)$. We refer to the facets above – other than $y_n \geq 0$ – as trivial facets, and refer to the remaining facets of $K'(n, r)$ as nontrivial. Finally, note that the recession cone of $K'(n, r)$ contains the vectors $je_i + if_j$, for all i, j satisfying $1 \leq i, j \leq n$. For $K'(n, r)$, let e_+ and e_- be the unit vectors in $\mathbb{R}^{2n+2}$ with ones in, respectively, the v_+ component, and the v_- component, and zeros elsewhere. For a vector χ in the $K'(n, r)$ space, define its restriction to the $K(n, r)$ space by removing the v_+ and v_- components, and denote it by χ_{re}.

Proposition 2.7.1. *All nontrivial facet-defining inequalities for $K'(n, r)$ have the form*

$$\pi_1 v_+ + \rho_1 v_- + \sum_{i=1}^{n} \pi_i x_i + \sum_{i=1}^{n} \rho_i y_i \geq \pi_o. \tag{26}$$

Furthermore, inequality (26) *is facet-defining if and only if $\pi x + \rho y \geq \pi_o$ defines a nontrivial facet of $K(n, r)$.*

Proof. Let $\pi x + \rho y \geq \pi_o$ define a nontrivial facet of $K(n,r)$. We first show that the inequality (26) is valid for $K'(n,r)$. Assume (26) is violated by some integral point $\chi \in K'(n,r)$ (the x and y components of χ are integral). Then the left-hand-side of (26) evaluated at χ equals a number z less than π_o. Let $v'_+ = e_+^T\chi$ and $v'_- = e_-^T\chi$. The property (R1) in Observation 2.2.8 implies that $\pi_1 + \rho_1 \geq 0$. Therefore, if $\min\{v'_+, v'_-\} = \epsilon > 0$, then (26) is also violated by the point $\chi - \epsilon(e_+ + e_-) \in K'(n,r)$. We can thus assume that χ satisfies $\min\{v'_+, v'_-\} = 0$. But $\min\{v'_+, v'_-\} = 0$ combined with the integrality of χ implies that v'_+ and v'_- are both integers. Therefore $\chi' = \chi_{re} + v'_+e_1 + v'_-f_1$ is an integral point contained in $K(n,r)$, and $(\pi,\rho)^T\chi' = z < \pi_o$, which contradicts the fact that $\pi x + \rho y \geq \pi_o$ is satisfied by all points in $K(n,r)$.

To see that (26) defines a facet of $K'(n,r)$, let $\chi^1, \ldots \chi^{2n-1}$ be affinely independent integral points in $K(n,r)$ which satisfy $(\pi,\rho)^T\chi^i = \pi_o$. As the facet defined by $\pi x + \rho y \geq \pi_o$ does not equal the facet defined by either $x_1 \geq 0$ or $y_1 \geq 0$, there are indices j,k such that $e_1^T\chi^j = s > 0$ and $f_1^T\chi^k = t > 0$. Define $2n+1$ affinely independent points in $\mathbb{R}^{2n+2}$ as follows:

$$\psi^i = (0,0,\chi^i) \text{ for } i = 1,\ldots,2n-1;$$

$$\psi^+ = \psi^j + se_+ - se_1;\ \psi^- = \psi^k + te_- - tf_1.$$

These points satisfy (26) at equality, and therefore (26) defines a facet of $K'(n,r)$.

We now show that every nontrivial facet of $K'(n,r)$ has the form in (26). Assume $\eta^T(v_+, v_-, x, y) \geq \eta_o$ defines a nontrivial facet F of $K'(n,r)$. Let $\eta = (\alpha_+, \alpha_-, \pi, \rho)$, where $\alpha_+, \alpha_- \in \mathbb{R}$, and $\pi, \rho \in \mathbb{R}^n$. There exists a point $\chi \in K'(n,r)$ lying on the above facet such that $\chi^Te_1 > 0$. As $\chi - e_1 + e_+ \in K'(n,r)$, we conclude that $\alpha_+ \geq \pi_1$. We can similarly conclude that $\alpha_- \geq \rho_1$ and therefore $\alpha_+ + \alpha_- \geq \pi_1 + \rho_1 \geq 0$. The last inequality is implied by the fact that $e_1 + f_1$ is contained in the recession cone of $K'(n,r)$. If $\alpha_+ + \alpha_- = 0$, then clearly $\alpha_+ = \pi_1$ and $\alpha_- = \rho_1$. Assume $\alpha_+ + \alpha_- > 0$. As F is not the same as the facet $v_+ \geq 0$, there exists an integral point $\chi = (v'_+, v'_-, x', y') \in K'(n,r)$ lying on F such that $v'_+ > 0$. If $v'_- > 0$, let $\min\{v'_+, v'_-\} = \epsilon > 0$. Then $\chi^1 = \chi - \epsilon(e_+ + e_-) \in K'(n,r)$, but $\eta^T\chi^1 = \eta_o - \epsilon(\alpha_+ + \alpha_-) < \eta_o$. This contradicts the fact that (η, η_o) defines a valid inequality for $K'(n,r)$. We can therefore assume that $v'_- = 0$ and $v'_+ = t$, for some positive integer t. Define χ^2 as $\chi - te_+ + te_1$. As $\chi^2 \in K'(n,r)$, it follows that $\eta^T\chi^2 \geq \eta_o \Rightarrow \alpha_+ \leq \pi_1$. We can conclude that $\alpha_+ = \pi_1$; a similar argument shows that $\alpha_- = \rho_1$.

Finally, we show that if (26) defines a facet of $K'(n,r)$, then the inequality $\pi x + \rho y \geq \pi_o$ defines a facet of $K(n,r)$. Firstly, this defines a valid inequality for $K(n,r)$ as any point in $K(n,r)$ can be mapped to a point in $K'(n,r)$ by appending zeros in the v_+ and v_- components. If it does not define a facet, then $(\pi, \rho) \geq \sum_i \lambda_i(\pi^i, \rho^i)$ and $\pi_o \leq \sum_i \lambda_i \pi_o^i$ for some nontrivial facet-defining inequalities $\pi^i x + \rho^i y \geq \pi_o^i$ of $K(n,r)$, and some numbers $\lambda_i \geq 0$. But that would imply that $(\pi_1, \rho_1, \pi, \rho) \geq \sum_i \lambda_i(\pi_1^i, \rho_1^i, \pi^i, \rho^i)$. By the first part of the proof, the inequalities $\pi_1^i v_+ + \rho_1^i v_- + \pi^i x + \rho^i y \geq \pi_o^i$ define facets of $K'(n,r)$, and this contradicts the assumption that (26) defines a facet of $K'(n,r)$. ∎

2.8 Using $K(n,r)$ to generate valid inequalities for MIP

Gomory and Johnson used facets of $P(n,r)$ to derive valid inequalities for knapsack sets. In particular, they derived *subadditive functions* from facet coefficients via *interpolation*. We show here how to derive valid inequalities for knapsack sets from facets of $K(n,r)$ via interpolation. Clearly, such inequalities can also be used as valid inequalities for general MIPs by using a knapsack set that is a relaxation of the original MIP. Our main result is presented at the end of the section in Theorem 2.8.5 and the rest of the section is dedicated to developing auxiliary steps for it. For a real number v, we define $\hat{v}$ as $v - \lfloor v \rfloor$.

Definition 2.8.1. *Given a facet-defining inequality $\pi x + \rho y \geq \pi_o$ for $K(n,r)$, let*

$f^z : \mathbb{Z} \cap [-n, n] \to \mathbb{R}$ *be defined as*

$$f^z(s) = \begin{cases} \pi_s & \text{if } s > 0, \\ 0 & \text{if } s = 0, \\ \rho_{-s} & \text{if } s < 0. \end{cases}$$

We say that $f : [-n, n] \to \mathbb{R}$ *is a* facet-interpolated function *derived from* (π, ρ, π_o) *if*

$$f(v) = (1 - \hat{v})f^z(\lfloor v \rfloor) + \hat{v}f^z(\lceil v \rceil).$$

The function f, as defined above, equals $f^z(v)$ when v is an integer, and therefore satisfies:

$$f(v) = (1 - \hat{v})f(\lfloor v \rfloor) + \hat{v}f(\lceil v \rceil). \tag{27}$$

In the next result, we show that continuous functions arising via interpolation from facets of $K(n, r)$ satisfy continuous analogues of the pairwise subadditivity conditions.

Proposition 2.8.2. *Let* f *be a facet-interpolated function derived from a facet of* $K(n, r)$ *that is not a nonnegativity constraint. Then*

$$f(u) + f(v) \geq f(u + v) \text{ if } u, v, u + v \in [-n, n].$$

Proof. The proposition is true when u and v are integers; the condition $f(u)+f(v) \geq f(u+v)$ translates to one of (SA0), (SA1), (SA2), (SA1'), (SA2') or (SA2"). Assume u is not an integer. As $u + v \in [-n, n]$, clearly $\lfloor u + v \rfloor$ and $\lceil u + v \rceil$ also belong to $[-n, n]$.

Case 1: $\hat{u} + \hat{v} \leq 1$. Then $\lfloor u + v \rfloor = \lfloor u \rfloor + \lfloor v \rfloor$ and $\lceil u + v \rceil = \lceil u \rceil + \lfloor v \rfloor$. We can rewrite the expression for $f(u)$ in (27) as

$$f(u) = (1 - \hat{u} - \hat{v})f(\lfloor u \rfloor) + \hat{u}f(\lceil u \rceil) + \hat{v}f(\lfloor u \rfloor). \tag{28}$$

Similarly,

$$f(v) = (1 - \hat{u} - \hat{v})f(\lfloor v \rfloor) + \hat{v}f(\lceil v \rceil) + \hat{u}f(\lfloor v \rfloor). \tag{29}$$

Adding the right-most terms in the above expressions, and using the fact that the proposition is true when u and v are integers, we obtain

$$f(u) + f(v) \geq (1 - \hat{u} - \hat{v}) f(\lfloor u + v \rfloor) + \hat{u} f(\lceil u + v \rceil) + \hat{v} f(\lfloor u \rfloor + \lceil v \rceil). \tag{30}$$

If v is an integer, then $\hat{v} = 0$ and the right-hand side of (30) equals $f(u+v)$. If v is not an integer, then $\lceil u + v \rceil = \lfloor u \rfloor + \lceil v \rceil$, and again the right-hand side of (30) equals $f(u+v)$.

Case 2: $\hat{u} + \hat{v} > 1$. Then $\lfloor u + v \rfloor = \lfloor u \rfloor + \lceil v \rceil = \lceil u \rceil + \lfloor v \rfloor$ and $\lceil u + v \rceil = \lceil u \rceil + \lceil v \rceil$. We can expand $\hat{u} f(\lceil u \rceil)$ in (27) as $(\hat{u} + \hat{v} - 1) f(\lceil u \rceil) + (1 - \hat{v}) f(\lceil u \rceil)$. We can similarly expand $\hat{v} f(\lceil v \rceil)$. When we add the expressions for $f(u)$ and $f(v)$ in (27) after writing the expanded terms, we get

$$f(u) + f(v) \geq (\hat{u} + \hat{v} - 1) f(\lceil u + v \rceil) + (2 - \hat{u} - \hat{v}) f(\lfloor u + v \rfloor).$$

The right-hand side of the inequality above equals $f(u+v)$. ∎

We say that functions satisfying the property in Proposition 2.8.2 are *subadditive over the interval* $[-n, n]$. We will see how to generate valid inequalities for knapsack sets from such functions in Proposition 2.8.4. Also, we can obtain valid inequalities using slightly more restricted functions: we say that f is a *restricted subadditive* function if $f(u) + f(v) \geq f(u+v)$ for $u \in [-n, n]$, and $v, u+v \in [0, n]$. In the next result, we show that facet-interpolated functions satisfy the continuous analogue of condition (SA3).

Proposition 2.8.3. *Let f be a facet-interpolated function derived from a nontrivial facet of $K(n, r)$. Then*

$$f(u) + f(v) + f(w) \geq f(u + v + w) \textit{ if } u \in [-n, n], \textit{ and } v, w, u + v + w \in [0, n].$$

Proof. (sketch) The proposition is true when u, v and w are integers; the condition $f(u) + f(v) + f(w) \geq f(u + v + w)$ translates to (SA3). As in the proof of (2.8.2), we assume either that $\hat{u}+\hat{v}+\hat{w}$ is contained in $(0,1]$ or $(1,2]$ or $(2,3)$. In the first case, we expand $(1-\hat{u})f(\lfloor u \rfloor)$ as $(1-\hat{u}-\hat{v}-\hat{w})f(\lfloor u \rfloor)+(\hat{v}+\hat{w})f(\lfloor u \rfloor)$, and proceed similarly for the terms involving $f(\lfloor v \rfloor)$ and $f(\lfloor w \rfloor)$. In the third case, we expand $\hat{u}f(\lceil u \rceil)$ as $(\hat{u}+\hat{v}+\hat{w}-2)f(\lceil u \rceil)+(2-\hat{v}-\hat{w})f(\lceil u \rceil)$, and proceed similarly for the terms involving $f(\lceil v \rceil)$ and $f(\lceil w \rceil)$. The second case has a number of sub-cases. For example, in expanding the terms in the definition of $f(u)$ in (27), we need to consider the value of $\hat{v}+\hat{w}$ with respect to 1. If $\hat{v}+\hat{w} \leq 1$, then we write $\hat{u}f(\lceil u \rceil)$ as $(\hat{u}+\hat{v}+\hat{w}-1)f(\lceil u \rceil)+(1-\hat{v}-\hat{w})f(\lceil u \rceil)$. On the other hand, if $\hat{u}+\hat{v}>1$, we expand ■

It is well-known that subadditive functions yield valid inequalities for knapsack sets; the point we emphasize in the next result is that one does not need subadditivity over the entire real line.

Proposition 2.8.4. *Consider the set $K=\{w \in \mathbb{Z}^p : \sum_{i=1}^p a_i w_i = b\}$, where the coefficients a_i and b are rational numbers. Let t be a number such that $ta_i, tb \in [-n,n]$ and $tb>0$. If a function f is (i) subadditive over the interval $[-n,n]$ or (ii) satisfies restricted subadditivity and the condition in Proposition 2.8.3, then*

$$\sum_{i=1}^{p} f(ta_i)w_i \geq f(tb)$$

is a valid inequality for K.

Proof. We can scale the coefficients a_i and b in the constraint defining K by a rational number $\lambda>0$ so that they become integers contained in the interval $[-m,m]$, with $m=\lambda\max\{|a_i|,|b|\}$. Define the function $g:[-m,m]\to\mathbb{R}$ by $g(w)=f((t/\lambda)w)$. In case (i), g is subadditive over the domain $[-m,m]$. Therefore the vector $\tilde{g}=(g(-m),g(-m+1),\ldots,g(1),\ldots,g(m))$ satisfies (SA0), (SA1), (SA2) and (SA1') with respect to $K(m,b)$, and (by Lemma 2.2.7 and Lemma 2.2.13)

$$\sum_{i=1}^{p} g(\lambda a_i)w_i \geq g(\lambda b)$$

is a valid inequality for K. In case (ii), $\tilde{g}$ satisfies (SA1), (SA2) and (SA3) with respect to $K(m,b)$ and by Lemma 2.2.13 the inequality above is valid for K. ∎

Note that Proposition 2.8.4 implies that facet-interpolated functions derived from facets of $K(n,r)$ can be used to generate valid inequalities for K (and consequently for general MIPs). Figure 2 presents examples of such functions obtained from a facet of $K(n,r)$. Note that the first example displays a feature not observed in facets of $P(n,r)$, namely negative cut coefficients.

We can now give the mixed-integer extension of the previous result.

Theorem 2.8.5. *Let f be a facet-interpolated function derived from a nontrivial facet of $K(n,r)$. Consider the set*

$$Q = \left\{(s,w) \in \mathbb{R}^q_+ \times \mathbb{Z}^p_+ : \sum_{i=1}^{q} c_i s_i + \sum_{i=1}^{p} a_i w_i = b\right\},$$

where the coefficients of the knapsack constraint defining Q are rational numbers. Let t be such that $ta_i, tb \in [-n,n]$ and $tb > 0$. Then the inequality

$$f(1)\sum_{i=1}^{q}(tc_i)^+ s_i + f(-1)\sum_{i=1}^{q}(-tc_i)^+ s_i + \sum_{i=1}^{p} f(ta_i) w_i \geq f(tb),$$

where $(\alpha)^+ = \max(\alpha, 0)$, is valid for Q.

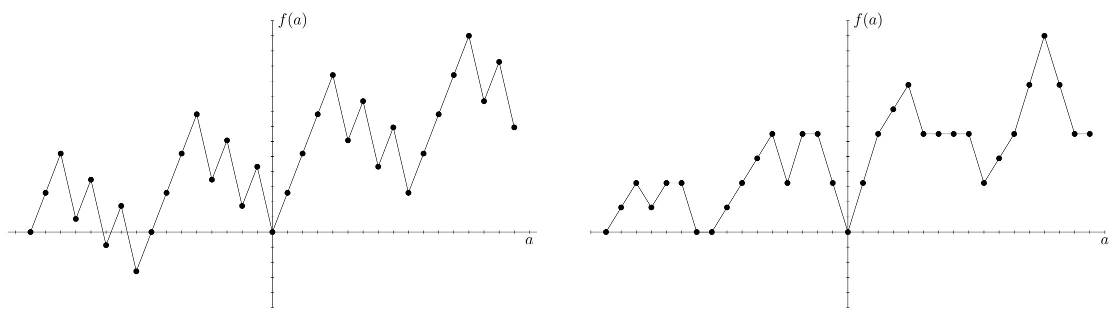

Figure 2: Example of two facet-interpolated functions derived from facets of $K(16,13)$

2.9 *Final remarks*

We defined and studied the Master Equality Polyhedron, a generalization of the Master Cyclic Group Polyhedron and presented an explicit characterization of the polar of its nontrivial facet-defining inequalities. We showed that facets of MEP can be used to obtain valid inequalities for a general MIP that cannot be obtained from facets of MCGP. In addition, for mixed-integer knapsack sets with rational data and nonnegative variables without upper bounds, our results yield a pseudo-polynomial time algorithm to separate and therefore optimize over their convex hull. This can be done by scaling their data and aggregating variables to fit into the MEP framework. Our characterization of MEP can also be used to find violated Homogeneous Extended Capacity Cuts efficiently. These cuts were proposed in [98] for solving Capacitated Minimum Spanning Tree problems and Capacitated Vehicle Routing problems.

An interesting topic for further study is the derivation of "interesting" classes of facets for MEP, i.e., facets which cannot be lifted from facets of MCGP and are not defined by rank one mixed-integer rounding inequalities.

CHAPTER III

LIFTING

3.1 Introduction

In Section 1.4 we introduced *lifting* as a technique to obtain valid inequalities for a polyhedron based on lower dimensional restrictions of it. However, in that section, we introduced the case where the lower dimensional restrictions were obtained by fixing some of the variables to a certain value. In this chapter we consider lifting in a slightly more general form, where the lower dimensional restrictions are obtained by intersecting the original polyhedron with any arbitrary affine subspace.

Specifically, consider the nonempty mixed integer set

$$P = \{x \in \mathbb{R}^n : Ax \leq d,\ x_i \in \mathbb{Z}\ \forall i \in I\},$$

where $A \in \mathbb{Q}^{m\times n}$, $d \in \mathbb{Q}^n$, and $I \subseteq \{1,\dots,n\}$. Consider a set of affine functions $g_l(x) = (v^l x - w^l)$ for $l = 1,\dots,L$, where $v^l \in \mathbb{Q}^n$ and $w^l \in \mathbb{Q}$ for all l. Also, for notation purposes, given an affine function $h(x) = ux - u_o$, let $h^o(x) := ux$ be the linear homogeneous function corresponding to h.

Let

$$P(g_1,\dots,g_L) = \{x \in \mathbb{R}^n : Ax \leq d;\ x_i \in \mathbb{Z},\ \ \forall i \in I;\ \ g_l(x) = 0,\ \ \forall l = 1,\dots,L\}$$

be the nonempty restriction of P obtained by fixing $g_l(x) = 0$ for all $l \in \{1,\dots,L\}$. Also, assume that $f(x) \geq 0$ is a valid inequality for $P(g_1,\dots,g_L)$, where $f(x)$ is an affine function of x defined by $f(x) := \pi x - \pi_o$. Define *lifting* as obtaining a set of scalars $\{\lambda_l\}_{l=1}^L$ such that $f(x) - \sum_{l=1}^{L} \lambda_l g_l(x) \geq 0$ is a valid inequality for P. Defining the set

$$\Lambda(P) := \left\{\lambda \in \mathbb{R}^L : f(x) - \sum_{l=1}^{L} \lambda_l g_l(x) \geq 0,\ \forall x \in P\right\}, \tag{31}$$

we can say that lifting consists in finding feasible points in $\Lambda(P)$. Obviously we will only be interested in feasible points in $\Lambda(P)$ that are "good" in a certain sense that will be defined depending on the context.

Note that, by setting $g_l(x) = x_{i_l} - y_{i_l}$, the above problem includes the case where the restrictions are obtained by setting variables x_{i_l} to given values y_{i_l}.

The main focus of this chapter is the case where $L = 1$, which we call single-lifting. Section 3.3 is dedicated to the case where $L > 1$, but is only meant to start a discussion of the more general case, giving basic properties and relations to known problems.

3.2 Single-lifting

In this section we consider the case where $L = 1$, in which case, we denote $g_1(x)$ as $g(x)$. This is an important case, since sequential lifting can be seen as multiple applications of single-lifting over increasingly higher dimensional polyhedra.

In the case of single-lifting, the set $\Lambda(P)$ defined in (31) becomes

$$\{\lambda : f(x) - \lambda g(x) \geq 0,\ \forall x \in P\}$$

and since we want only nondominated valid inequalities, we want to get λ as large as possible, that is, we want to find

$$\max\{\lambda : f(x) - \lambda g(x) \geq 0,\ \forall x \in P\} \tag{32}$$

where $g(x) \geq 0$ for all $x \in P$. We call this the *lifting problem.* The following theorem, which is adapted from a theorem of Wolsey [101], shows that the assumption that $g(x) \geq 0$ for all $x \in P$ is not restrictive, since otherwise we can just partition P into $P \cap \{x : g(x) \geq 0\}$ and $P \cap \{x : g(x) \leq 0\}$ and obtain the lifting coefficient from the lifting coefficients for each side of the partition of P.

Theorem 3.2.1 (Wolsey [101]). *$f(x) - \lambda' g(x) \geq 0$ is a valid inequality for P if and* *only if* $\underline{\lambda} \leq \lambda' \leq \bar{\lambda}$ *where*

$$\underline{\lambda} = \min\{\lambda : f(x) - \lambda g(x) \geq 0 \textit{ is valid for } P \cap \{x : g(x) \leq 0\}\},$$

$$\bar{\lambda} = \max\{\lambda : f(x) - \lambda g(x) \geq 0 \textit{ is valid for } P \cap \{x : g(x) \geq 0\}\}.$$

Moreover, if $\lambda' = \underline{\lambda} > -\infty$ *or* $\lambda' = \bar{\lambda} < +\infty$ *and* $f(x) \geq 0$ *defines a* k*-dimensional face of* $conv(P(g))$*, then* $f(x) - \lambda' g(x) \geq 0$ *defines an at least* $k+1$*-dimensional face of* $conv(P)$*.*

Proof. To see that the problems are well-defined, note that $f(x) - \lambda g(x) \geq 0$ is valid for $P \cap \{x : g(x) \leq 0\}$ if and only if it is valid for $Q := conv(P \cap \{x : g(x) \leq 0\})$, which is a polyhedron. Thus $f(x) - \lambda g(x) \geq 0$ is valid for $P \cap \{x : g(x) \leq 0\}$ if and only if

$$f(x^k) - \lambda g(x^k) \geq 0 \tag{33}$$

for all extreme points x^k of Q and

$$f^o(r^j) - \lambda g^o(r^j) \geq 0 \tag{34}$$

for all extreme rays r^j of Q.

Thus, the problems are equivalent to linear programs and, therefore, it is well-defined to put a minimum or maximum there, instead of sup or inf.

$\underline{(\Rightarrow)}$

If $\lambda' < \underline{\lambda}$, then there exists $x \in P \cap \{x : g(x) \leq 0\} \subseteq P$ such that $f(x) - \lambda' g(x) < 0$, since, otherwise, $\underline{\lambda}$ would not be minimal.

Likewise, if $\lambda' > \bar{\lambda}$, then there exists $x \in P \cap \{x : g(x) \geq 0\} \subseteq P$ such that $f(x) - \lambda g(x) < 0$, since, otherwise, $\bar{\lambda}$ would not be maximal.

$\underline{(\Leftarrow)}$

Suppose there exists $x \in P$ such that $f(x) - \lambda' g(x) < 0$. Clearly, $g(x) \neq 0$ since we are assuming $f(x) \geq 0$ is valid for $P \cap \{x : g(x) = 0\}$.

If $g(x) < 0$, then this implies that $\lambda' < \underline{\lambda}$. This is true, since if $\lambda' \geq \underline{\lambda}$, then $0 > f(x) - \lambda' g(x) \geq f(x) - \underline{\lambda} g(x) \geq 0$.

Likewise, if $g(x) > 0$, then this implies that $\lambda' > \bar{\lambda}$. This is true, since if $\lambda' \leq \underline{\lambda}$, then $0 > f(x) - \lambda' g(x) \geq f(x) - \bar{\lambda} g(x) \geq 0$.

This concludes the proof of the first part of the theorem.

Now, to prove the dimension of the resulting inequality, consider the case when $\lambda = \underline{\lambda} > -\infty$. Note that since $\underline{\lambda} > -\infty$, there must be either one inequality from (33) with $g(x^k) < 0$ or one inequality from (34) with $g^o(r^j) < 0$ that is satisfied at equality at $\underline{\lambda}$. If such an inequality is from (33), then let $x' = x^k$, otherwise, let $x' = x'' + \beta r^j$ with x'' a point in $P \cap \{x : g(x) = 0\}$ that satisfies $f(x'') = 0$ and β an appropriate number such that $x' \in P \cap \{x : g(x) \leq 0\}$. Notice that x'' exists because $f(x)$ defines a k-dimensional face of $conv(P(g))$. It is easy to see that, since $g(x') < 0$, then x' is affinely independent from any point in $P \cap \{x : g(x) = 0\}$ and that $f(x') - \underline{\lambda} g(x') = 0$. Thus, we have one more affinely independent point satisfying such a constraint at equality, giving us a face of dimension at least $k + 1$. The proof is analogous for $\lambda = \bar{\lambda} < +\infty$. ∎

Note that no assumption is made on $g(x)$ other than that it is an affine function of x and that $g(x) \geq 0$ for all $x \in P$. Therefore, if we can solve the lifting problem as defined in (32), we can solve the lifting problem for lifting a general constraint, as well as lifting variables that are continuous or integer, bounded or unbounded. As mentioned in Section 1.4, general lifting algorithms have been developed only in the case of lifting bounded integer variables.

In Section 3.2.1, we study basic properties of the solution of problem (32) and in Sections 3.2.2 and 3.2.3 present algorithms to solve it based on an optimization oracle. Sections 3.2.4 and 3.2.5 show the relationship between problem (32) and tilting and fractional programming. These connections are important because they allow the use of results from one problem to be applied to the others. Moreover, they allow us to see all three problems in a same common perspective and, therefore, one only needs to focus on problem (32).

We do not present computational results in this chapter. The reason is that the context in which we apply the ideas of this chapter is the subject of Chapter 4 and, therefore, we refrain from presenting the computational experiments until that chapter so that the description of the experiments becomes clearer.

3.2.1 Solving the single-lifting problem

This section is devoted to solving problem (32):

$$\max\{\lambda : f(x) - \lambda g(x) \geq 0,\ \forall x \in P\},$$

where $g(x) \geq 0$ for all $x \in P$ and $P \neq \emptyset$. Notice that we do not assume that $P(g)$ is nonempty or that $f(x) \geq 0$ is valid for $P(g)$. We begin by characterizing feasibility and unboundedness by means of the following two propositions.

Proposition 3.2.2. *Problem* (32) *is feasible if and only if*

$$g(x) = 0 \textit{ implies } f(x) \geq 0,\ \forall x \in P \tag{35}$$

and

$$g^o(r) = 0 \textit{ implies } f^o(r) \geq 0,\ \forall r \in \textit{cone}(P). \tag{36}$$

Proof. It is easy to see that (35) and (36) are necessary conditions, so we focus on sufficiency. We assume conditions (35) and (36) and show that there exists $\lambda < 0$ such that $f(x) - \lambda g(x) \geq 0$ for all $x \in P$.

If condition (35) holds, it follows that for each $x \in P$ we have either $f(x) \geq 0$ or $g(x) > 0$, that is, $f(x) < 0$ implies $g(x) > 0$ for each $x \in P$. Likewise, from (36) it follows that $f^o(r) < 0$ implies $g^o(r) > 0$ for every ray r of $\textit{cone}(P)$.

Let $\{x^1, \ldots, x^p\}$ and $\{r^1, \ldots, r^q\}$ correspond to the set of all extreme points and rays of P. It suffices to show that there exists a value $\lambda < 0$ such that $f(x^i) - \lambda g(x^i) \geq 0$ for all $i \in 1, \ldots, p$ and $f^o(r^j) - \lambda g^o(r^j) \geq 0$ for all $j \in 1, \ldots, q$.

Let λ be a negative value such that

$$\lambda < \frac{f^o(r^j)}{g^o(r^j)}, \quad \text{for all } j \in 1, \ldots, q \text{ such that } f^o(r^j) < 0,$$

and such that

$$\lambda < \frac{f(x^i)}{g(x^i)}, \quad \text{for all } i \in 1, \ldots, p \text{ such that } f(x^i) < 0.$$

Choosing λ in this way it is clear that all extreme points (and rays) such that $f(x^i) < 0$ (and $f^o(r^j) < 0$) satisfy the required condition.

Finally, given that $\lambda < 0$, we have that $f(x^i) \geq 0$ implies $f(x^i) - \lambda g(x^i) \geq 0$. Likewise, $f^o(r^j) \geq 0$ implies $f^o(r^j) - \lambda g^o(r^j) \geq 0$. Thus the conditions hold. ∎

It is worth noting that if there exists $x \in P$ such that $g(x) = 0$, then (35) implies (36). Let us now focus on determining when (32) is unbounded.

Proposition 3.2.3. *Problem* (32) *is unbounded if and only if*

$$g(x) = 0 \text{ and } f(x) \geq 0 \text{ for all } x \in P. \tag{37}$$

Proof. It is easy to see that condition (37) is sufficient for unboundedness, so we focus on necessity. Suppose that this condition does not hold. We have that either (a) there exists $x' \in P$ such that $f(x') < 0$ and $g(x') = 0$, or (b) there exists $x' \in P$ such that $g(x') > 0$. From Proposition 3.2.2 we have that in the first case the problem is infeasible. In the latter case we have that every feasible solution λ satisfies $\lambda \leq -f(x')/g(x')$. ∎

From the previous two propositions it is clear that by solving at most two mixed integer programming problems one can determine if problem (32) is feasible or unbounded. Algorithm 1 shows how this can be achieved. Moreover, notice that, in the case of lifting, we have the assumptions that $P(g) \neq \emptyset$ and $f(x) \geq 0$ is valid for $P(g)$ and therefore conditions (35) and (36) are always true. In addition, condition (37) is true if and only if P is contained in the affine subspace defined by $g(x) = 0$. Therefore, the lifting problem is always feasible and it is only unbounded if $P = P \cap \{x : g(x) = 0\}$.

Algorithm 1: Phase I algorithm

Input: A non-empty mixed integer polyhedron P, and affine functions $f(x)$ and $g(x)$ as defined above.

Output: A variable STATUS indicating if $\max\{\lambda : f(x) - \lambda g(x) \geq 0\}$ is a feasible, infeasible or unbounded problem. In case the problem is feasible it also outputs a solution $x^o \in P$ such that $g(x^o) > 0$.

1 STATUS $\leftarrow$ unknown
2 $q \leftarrow \min\{f(x) : g(x) = 0,\ x \in P\}$
3 **if** $q < 0$ **then**
4   STATUS $\leftarrow$ infeasible
5 /* Case where $g(x) > 0$ for all $x \in P$ */
6 **else if** $q = +\infty$ **then**
7     $q \leftarrow \min\{f^o(r) : g^o(r) = 0,\ r \in cone(P)\}$
8   **if** $q = -\infty$ **then**
9   STATUS $\leftarrow$ infeasible
10 **if** STATUS $=$ *unknown* **then**
11     $z \leftarrow \max\{g(x) : x \in P\}$
12   **if** $z = 0$ **then**
13   STATUS $\leftarrow$ unbounded
14   **else**
15   STATUS $\leftarrow$ feasible
16   **if** $z = +\infty$ **then**
17   Let r^g be the ray with $g^o(r^g) > 0$ obtained from step 11
18   Scale ray r^g so that it becomes integer
19             $x^f \leftarrow \arg\min\{f(x) : g(x) = 0,\ x \in P\}$
20             $x^o \leftarrow x^f + r^g$
21   **else**
22             $x^o \leftarrow \text{argmax}\{g(x) : x \in P\}$

3.2.2 The Newton-Rhapson algorithm

Once it has been established that problem (32) is feasible and bounded, it is possible to solve it using a variant of the Newton-Rhapson algorithm for finding roots of a function. For the motivation of this, consider the function

$$z(\lambda) = \min\{f(x) - \lambda g(x) : x \in P\}.$$

It is easy to see that, for any valid λ, $z(\lambda) \geq 0$. Moreover, if $z(\lambda) > 0$, then there exists $\lambda' > \lambda$ that is also feasible for problem (32). Therefore, the optimal solution λ^* of problem (32) should satisfy the condition $z(\lambda^*) = 0$ (this statement is not necessarily true if P is not compact, but since we are just motivating the method, we will not worry about that case).

To solve this equation, we start with a guess $\lambda_1 \geq \lambda^*$ such that $f(x^o) - \lambda_1 g(x^o) = 0$ for some $x^o \in P$ and thus $z(\lambda_1) \leq 0$. At iteration $i \geq 1$ assume $0 > z(\lambda_i) > -\infty$. Let x^i be the corresponding minimizer for z, that is, $z(\lambda_i) = f(x^i) - \lambda_i g(x^i)$. We have that $-g(x^i)$ is a subgradient for z in λ_i. Thus, if $g(x^i) \neq 0$, a valid Newton-Raphson iteration would be $\lambda_{i+1} = \lambda_i + \frac{z(\lambda_i)}{g(x^i)} = \frac{f(x^i)}{g(x^i)}$. Note that at all iterations, $f(x^i) - \lambda_{i+1} g(x^i) = 0$ and therefore $z(\lambda_i) \leq 0$ for all i and if we find that $z(\lambda_i) = 0$ at iteration i, then $\lambda_i = \lambda^*$.

Another intuitive way to look at the procedure is the following: Note that if $z(\lambda_i) < 0$ then $g(x^i) > 0$, since otherwise the problem would be infeasible. Since $f(x^i) - \lambda_{i+1} g(x^i) = 0$ and $g(x^i) > 0$, we know that $\lambda^* \leq \lambda_{i+1}$. The idea is to make a sequence of reasonable guesses that are decreasing monotonically and are always an upper bound on λ^*. The guesses are reasonable since, at least for some point $x \in P$, $f(x) - \lambda_i g(x) = 0$, so it is possible that $z(\lambda_i) = 0$. For each guess, we use the optimization oracle for P to check if λ_i is indeed equal to λ^*.

The procedure is described in detail in Algorithm 2, where we also consider the case that $z(\lambda_i)$ is not finite in Steps 3 - 8. A proof of correctness and convergence is given in Theorem 3.2.4. It is important to notice that Algorithm 2 is, in essence, the same as Dinkelbach's [43] algorithm for solving fractional programming problems and Applegate et al. [5] and Espinoza [47]'s algorithm for tilting, but it has not been previously applied to lifting.

Algorithm 2: Main algorithm (Phase II)

Input: A mixed integer polyhedron P, and affine functions $f(x) = \pi x - \pi_o,\ g(x) = vx - w$ such that $g(x) \geq 0, \forall x \in P$ and such that $\max\{\lambda : f(x) - \lambda g(x) \geq 0,\ \forall x \in P\}$ is neither infeasible nor unbounded. A feasible solution $x^o \in P$ such that $g(x^o) > 0$.

Output: The solution λ^* to $\max\{\lambda : f(x) - \lambda g(x) \geq 0,\ \forall x \in P\}$.

1 $\lambda_1 \leftarrow \frac{f(x^o)}{g(x^o)}$
2 $i \leftarrow 1$
3 $z_1 \leftarrow \min\{f(x) - \lambda_1 g(x) : x \in P_{LP}\}$
4 **while** $z_i = -\infty$ **do**
5   Let r^i be an extreme ray of P_{LP} proving the unboundedness of z_i
6     $\lambda_{i+1} \leftarrow \frac{f^o(r^i)}{g^o(r^i)}$
7     $i \leftarrow i + 1$
8     $z_i \leftarrow \min\{f(x) - \lambda_i g(x) : x \in P_{LP}\}$
9 $z_i \leftarrow \min\{f(x) - \lambda_i g(x) : x \in P\}$
10 **while** $z_i < 0$ **do**
11     $x^i \leftarrow \arg\min\{f(x) - \lambda_i g(x) : x \in P\}$
12     $\lambda_{i+1} \leftarrow \frac{f(x^i)}{g(x^i)}$
13     $i \leftarrow i + 1$
14     $z_i \leftarrow \min\{f(x) - \lambda_i g(x) : x \in P\}$
15 $\lambda^* \leftarrow \lambda_i$

Theorem 3.2.4. *Algorithm 2 satisfies the following properties:*

1. *Steps 6 and 12 are well-defined. That is, there are no divisions by zero.*

2. *The sequence $\{\lambda_i\}$ is monotone decreasing.*

3. *Step 11 is well-defined. That is, there exists an optimal solution x^i.*

4. *The algorithm terminates in a finite number of steps.*

5. *Upon completion, the value λ^* corresponds to the optimal solution value.*

Proof. 1. If $g^o(r_i) = 0$ in Step 6 then $f^o(r_i) < 0$. Likewise, if $g(x^i) = 0$ in Step 12 then $f(x^i) < 0$. Both of these conditions contradict the feasibility of the problem, as shown in Proposition 3.2.2.

2. In Step 6 we have that $g^o(r^i) > 0$ and $f^o(r^i) - \lambda_i g^o(r^i) < 0$. Thus, $\lambda_i > \frac{f^o(r^i)}{g^o(r^i)} = \lambda_{i+1}$. Likewise, in step 12 we have that $g(x^i) > 0$ and $f(x^i) - \lambda_i g(x^i) < 0$. Thus, $\lambda_i > \frac{f(x^i)}{g(x^i)} = \lambda_{i+1}$.

3. Let k represent the value of index i at Step 9. Since $z_k > -\infty$ we have $f^o(r) - \lambda_k g^o(r) \geq 0$ for any ray r of P. Since $g^o(r) \geq 0$ for every ray r of P, and since the sequence λ_i is monotone decreasing, we also have that $f^o(r) - \lambda_i g^o(r) \geq 0$ for all $i > k$. Moreover, if $g^o(r) > 0$ then $f^o(r) - \lambda_i g^o(r) > 0$ (we will use this in the next proof). Thus the problem is finite at Step 11 and a solution exists.

4. Observe at Step 6 that ray r^i satisfies $f^o(r^i) - \lambda_i g^o(r^i) < 0$ and $f^o(r^i) - \lambda_j g^o(r^i) \geq 0$ for $j > i$ and so, due to the finiteness of the number of extreme rays, the algorithm reaches step 9 after a finite number of steps.

   Again let k be the value of i once the algorithm reaches step 9. Assume now we are at an iteration $i > k$. At Step 12, point x^i satisfies $f(x^i) - \lambda_i g(x^i) < 0$. Now let K_i be the set of extreme points $\bar{x}^k$ of $conv(P)$ such that $f(\bar{x}^k) - \lambda_i g(\bar{x}^k) = z(\lambda_i)$. Clearly, x^i can be written as a convex combination of the points in K_i and a conic combination of a set of extreme rays R_i of P. But we know that if any $r \in R_i$ has $g^o(r) > 0$, then $f^o(r) - \lambda_i g^o(r) > 0$, which contradicts the optimality of x^i. Therefore $g^o(r) = 0$ which implies $f^o(r) = 0$ for all $r \in R_i$.

   But now note that $f(x^i) - \lambda_j g(x^i) \geq 0$ for all $j > i$. Since $f^o(r) - \lambda_j g^o(r) = 0$ for all $r \in R_i$, this implies that at least one extreme point $\bar{x}^i \in K_i$ also satisfies $f(\bar{x}^i) - \lambda_i g(\bar{x}^i) < 0$ and $f(\bar{x}^i) - \lambda_j g(\bar{x}^i) \geq 0$ for $j > i$. Finite termination follows from the finiteness of the number of extreme points of P.

5. Assume that the algorithm terminates on the i-th interation. Since it is clear that λ^* is feasible it suffices to show that every $\bar{\lambda} > \lambda^* = \lambda_i$ is infeasible. First, assume that λ^* was obtained at Step 6 of the algorithm. This implies $\lambda^* = \frac{f^o(r^{i-1})}{g^o(r^{i-1})}$ for some ray r^{i-1} of P. In this case $f^o(r^{i-1}) - \lambda^* g^o(r^{i-1}) = 0$. Since $g^o(r^{i-1}) > 0$ it follows that $f^o(r^{i-1}) - \bar{\lambda} g^o(r^{i-1}) < 0$. Let $\bar{x} = x^o + \alpha r^{i-1}$. It is easy to see that one can choose $\alpha > 0$ so that $\bar{x} \in P$ and $f(\bar{x}) - \bar{\lambda} g(\bar{x}) < 0$. Next, assume that λ^* was obtained at Step 1 or Step 12 of the algorithm. This implies $\lambda^* = \frac{f(x^{i-1})}{g(x^{i-1})}$ for some solution $x^{i-1} \in P$, and in turn that $f(x^{i-1}) - \lambda^* g(x^{i-1}) = 0$. Given $g(x^{i-1}) > 0$ it follows that $f(x^{i-1}) - \bar{\lambda} g(x^{i-1}) < 0$. In both cases $\bar{\lambda}$ is infeasible.

∎

Corollary 3.2.5. *If problem* (32) *is feasible and there exists a finite optimal solution* λ^* *then either*

1. *There exists an extreme point* x^* *of* P *such that* $g(x^*) > 0$ *and* $\lambda^* = \frac{f(x^*)}{g(x^*)}$,
2. *There exists an extreme ray* r^* *of* P *such that* $g^o(r^*) > 0$ *and* $\lambda^* = \frac{f^o(r^*)}{g^o(r^*)}$.

Proof. Follows from the proof of part 4 of Theorem 3.2.4. ∎

Observe that in general, given an optimal solution λ^* to problem (32), there need not exist a value $x^* \in P$ such that $f(x^*) - \lambda^* g(x^*) = 0$. The following proposition describes a simple sufficient condition by which to ensure that such a solution exists.

Proposition 3.2.6. *Assume that problem* (32) *admits an optimal solution* λ^*. *If there exists* $\overline{x} \in P$ *such that* $g(\overline{x}) = 0$ *and* $f(\overline{x}) = 0$, *then there exists* $x^* \in P$ *such that* $g(x^*) > 0$ *and* $\lambda^* = \frac{f(x^*)}{g(x^*)}$.

Proof. From Algorithm 2 it is easy to see that the optimal solution λ^* is such that $\lambda^* = \frac{f(\hat{x})}{g(\hat{x})}$ for some point $\hat{x}$ of P satisying $g(\hat{x}) > 0$, or such that $\lambda^* = \frac{f^o(\hat{r})}{g^o(\hat{r})}$ for some ray $\hat{r}$ of P satisfying $g^o(\hat{r}) > 0$. In the first case, simply define $x^* = \hat{x}$. In the latter, we may assume that $\hat{r}$ is integral and let $x^* = \bar{x} + \hat{r}$. Observe that in either case, the conditions hold. ∎

A natural concern regarding Algorithm 2 is the number of mixed-integer programming problems that could be called as sub-routines in Step 11. Proposition 3.2.7, which is an adaptation of a result from Schaible [94] in the context of fractional programming, gives some indication as to the number of iterations required to terminate.

Proposition 3.2.7. *Consider an instance of problem* (32)*. Suppose that iterates x^i and x^{i+1} have been obtained from Step 12 of Algorithm 2. If λ^{i+1} is not optimal, then, $g(x^i) > g(x^{i+1})$.*

Proof. We first prove that $g(x^i) \geq g(x^{i+1})$. For this, observe that the following conditions hold:

$$\begin{aligned} f(x^i) - \lambda^{i+1} g(x^i) &= 0, &\quad (38) \\ f(x^{i+1}) - \lambda^{i+1} g(x^{i+1}) &\leq 0, &\quad (39) \\ f(x^{i+1}) - \lambda^{i} g(x^{i+1}) &\geq f(x^i) - \lambda^i g(x^i). &\quad (40) \end{aligned}$$

Inequality (39) holds since x^{i+1} is the optimal solution to $z(\lambda^{i+1})$ and $f(x^i) - \lambda^{i+1} g(x^i) = 0$. Inequality (40) holds since x^i is the optimal solution to $z(\lambda^i)$. Thus, (39) - (38) imply that

$$f(x^{i+1}) - f(x^i) \leq \lambda^{i+1}(g(x^{i+1}) - g(x^i)).$$

Observe that (40) is equivalent to

$$f(x^{i+1}) - f(x^i) \geq \lambda^{i}(g(x^{i+1}) - g(x^i)).$$

From these last two inequalities we have that

$$(\lambda^{i+1} - \lambda^i)(g(x^{i+1}) - g(x^i)) \geq 0.$$

Since from Theorem 3.2.4 we know that $\lambda^i > \lambda^{i+1}$, the result follows.

We now prove that $g(x^i) \neq g(x^{i+1})$. For this, define

$$P_i = \{x \in P : g(x) = g(x^i)\}.$$

Observe that

$$x^i = \arg\min\{f(x) - \lambda^i g(x) : x \in P_i\} = \arg\min\{f(x) : x \in P_i\}.$$

From this, $f(x) - \lambda^{i+1}g(x) \geq 0$ for all $x \in P_i$. In fact, if $x \in P_i$ we have $f(x) - \lambda^{i+1}g(x) = f(x) - \lambda^{i+1}g(x^i) \geq f(x^i) - \lambda^{i+1}g(x^i) = 0$.

However, since λ^{i+1} is not optimal, we have $f(x^{i+1}) - \lambda^{i+1}g(x^{i+1}) < 0$. Thus, x^{i+1} is not in P_i. ∎

Note that, in the case that $g(x)$ assumes a finite number of values, Proposition 3.2.7 gives an immediate bound on the number of MIP solves that Algorithm 2 performs. Also, note that in the case where we are lifting a bounded integer variable that can assume values from $0, 1, \ldots, K$, for example, Wolsey [101] and Easton and Gutierrez [46] had already shown that at most $O(K)$ MIPs need to be solved in order to lift. Even though the worst-case complexity has not changed, Proposition 3.2.7 shows that it is reasonable to expect that less than K MIPs will actually need to be solved.

3.2.3 The one-tree algorithm

An alternative to Algorithm 2 was proposed by Easton and Gutierrez [46] for lifting bounded integer variables. Rather than enumerating a full branch-and-bound tree each time Step 11 of Algorithm 2 is executed, they instead propose an algorithm that uses a single branch-and-bound tree to solve (32). We call this algorithm the *one-tree* algorithm. We extend the one-tree algorithm for lifting generic constraints without any bound restrictions and prove its correctness.

Algorithm 3: The one-tree algorithm

Input: A mixed integer polyhedron P, and affine functions $f(x) = \pi x - \pi_o,\ g(x) = vx - w$ such that $g(x) \geq 0, \forall x \in P$ and such that $\max\{\lambda : f(x) - \lambda g(x) \geq 0,\ \forall x \in P\}$ is neither infeasible nor unbounded. A feasible solution $x^o \in P$ such that $g(x^o) > 0$.

Output: The solution λ^* to $\max\{\lambda : f(x) - \lambda g(x) \geq 0,\ \forall x \in P\}$.

$\lambda_o \leftarrow \frac{f(x^o)}{g(x^o)}$
$i \leftarrow 0$
$z^* \leftarrow \min\{f(x) - \lambda_i g(x) : x \in P_{LP}\}$
while $z^* = -\infty$ **do**
    Let r be an extreme ray of P_{LP} proving the unboundedness of z^*
    $\lambda_{i+1} \leftarrow \frac{f^o(r)}{g^o(r)}$
    $i \leftarrow i + 1$
    $z^* \leftarrow \min\{f(x) - \lambda_i g(x) : x \in P_{LP}\}$
$\mathcal{Q} \leftarrow \{P\}$
while $\mathcal{Q} \neq \emptyset$ **do**
    Choose $Q \in \mathcal{Q}$
    $(x^*, z^*) \leftarrow \arg\min\{z : z \geq f(x) - \lambda_i g(x),\ x \in Q_{LP}\}$
    if *the problem is infeasible* **then**
        $\mathcal{Q} \leftarrow \mathcal{Q} - \{Q\}$
    else if $z^* \geq 0$ **then**
        $\mathcal{Q} \leftarrow \mathcal{Q} - \{Q\}$
    else if $x^* \in P$ **then**
        $\lambda_{i+1} \leftarrow \frac{f(x^*)}{g(x^*)}$
        $i \leftarrow i + 1$
    else
        Partition Q into Q^1 and Q^2 such that $x^* \notin Q^1 \cup Q^2$.
        $\mathcal{Q} \leftarrow \mathcal{Q} - \{Q\} + \{Q^1, Q^2\}$
$\lambda^* \leftarrow \lambda$

Let us first make the following observation that will help us prove the correctness of Algorithm 3.

Observation 3.2.8. *Let h be an affine function from $\mathbb{R}^n \to \mathbb{R}$ and Q be a nonempty polyhedron such that* $\min\{h(x) : x \in Q\}$ *is bounded. Then*

$$(x^*, z^*) = \arg\min\{z : z \geq h(x),\ x \in Q\} \tag{41}$$

if and only if $x^* = \arg\min\{h(x) : x \in Q\}$ *and* $z^* = h(x^*)$.

Proof. $\underline{(\Rightarrow)}$

Since (x^*, z^*) is optimal to (41), then $z^* = h(x^*)$. If $x' \in Q$ is such that $h(x') < h(x^*)$, let $z' = h(x')$ and note that (x', z') is feasible for (41), which contradicts the optimality of (x^*, z^*).

$\underline{(\Leftarrow)}$

Note that (x^*, z^*) is feasible for (41). Now since $x^* = \arg\min\{h(x) : x \in Q\}$, any (x', z') feasible for (41) satisfies $z' \geq h(x') \geq h(x^*) = z^*$, so $(x^*, z^*) = \arg\min\{z : z \geq h(x),\ x \in Q\}$. ∎

We now prove the correctness of Algorithm 3.

Theorem 3.2.9. *Upon completion, Algorithm 3 returns the correct value of λ^*.*

Proof. First, note that Steps 3-8 are the same as Steps 3-8 in Algorithm 2 and therefore when Step 9 is reached, the problem $\min\{f(x) - \lambda g(x) : x \in P\}$ is bounded for all $\lambda \leq \lambda_i$.

In addition, if Step 18 is never reached, then the while loop beginning in Step 10 just describes a regular branch-and-bound algorithm for solving $\min\{f(x) - \lambda_i g(x) : x \in P\}$ for some fixed λ_i. Since Step 18 is never reached, the optimal solution is greater than or equal to zero, thus, $f(x) - \lambda_i g(x) \geq 0$ for all $x \in P$, so λ^* is feasible and either $f(x^o) - \lambda^* g(x^o) = 0$ or there exists a ray $r \in cone(P)$ such that $g^o(r) > 0$

and $f^o(r) - \lambda^* g^o(r) = 0$. Thus, optimality follows from the same arguments used in the proof of Theorem 3.2.4.

Now assume Step 18 is reached. First, note that the step is well-defined, since $g(x^*)$ cannot be zero because, otherwise, we have a point x^* with $f(x^*) < 0$ and $g(x^*) = 0$, which makes the problem infeasible. Notice that since λ_i is updated in Step 18, there exists $x^* \in P$ such that $g(x^*) > 0$ and $f(x^*) - \lambda^* g(x^*) = 0$, and, therefore, λ^* is an upper bound on any feasible λ.

All that remains to show is that λ^* is feasible. Suppose, by contradiction, that there exists $\bar{x} \in P$ such that $f(\bar{x}) - \lambda^* g(\bar{x}) < 0$. First, note that the sequence of values $\{\lambda_i\}$ is monotone decreasing, as in Algorithm 2. Let $\{Q\}$ be the leaf node of the branch-and-bound tree such that $\bar{x} \in Q$. This means that Q has to be pruned at line 16, and therefore, at that point, we know that $f(\bar{x}) - \lambda_i g(\bar{x}) \geq 0$, which is a contradiction, since $\lambda^* \leq \lambda_i$ and $g(\bar{x}) \geq 0$. ∎

One interesting point to notice is that we do not prune by feasibility, since otherwise we could obtain an incorrect solution λ^* in the end. Another important point is that this adaptation of the algorithm of Easton and Gutierrez [46] can be implemented using a standard commercial MIP solver and some callback functions. In constrast, Easton and Gutierrez's algorithm needs a custom implementation of a MIP solver which allows the user to modify the objective function throughout the branch-and-bound tree. Algorithm 3 gets around this fact by using the auxiliary variable z. In fact, notice that every time we update λ_i, we don't need to change the previous LP, we can just use the cut $z \geq f(x) - \lambda_i g(x)$ as a globally valid cut to the problem. This allows all the technology that is already implemented in commercial MIP solvers (like cutting planes, branching rules, heuristics, etc.) to be reused to implement it.

Intuitively, what this algorithm does is explore a branch-and-bound tree at the same time as it guesses values λ_i for λ^*. The tree is explored in such a way that if our guess is incorrect, whenever we update our guess λ_i, all the work that has already been done is still guaranteed to be correct and therefore is not wasted.

In the next two sections, we present the connection that problem (32) has with tilting and fractional programming. As we have seen in some of the theorems and propositions up to this point, such connections between lifting, tilting and fractional programming allow theoretical results from one area to be applied to the others as well.

3.2.4 Connection to tilting

Given a mixed-integer set P, tilting is a procedure which takes a valid inequality for P with a certificate of high dimension and transforms it into another valid inequality with a certificate of even higher dimension. This procedure is used to obtain a facet-defining inequality for $conv(P)$ from an inequality $g(x) \geq 0$ that is valid and defines a nonempty face of $conv(P)$. It has been used in Applegate et al. [5] in the context of the Traveling Salesman Problem (TSP) and in Espinoza [47] for general MIPs.

More specifically, given a linear inequality $g(x) \geq 0$ valid for P and a nonempty set P_o of affinely independent points contained in P such that $g(x) = 0$ for all $x \in P_o$, the *tilting* problem consists in finding an inequality $h(x) \geq 0$ valid for P and a point x' such that:

- $h(x) = 0$ for all $x \in P_o$,
- $h(x') = 0$,
- $P_o \cup \{x'\}$ is a set of affinely independent points.

An obvious assumption that is necessary for the tilting problem is that $|P_o| < \dim(P)$. Moreover, the tilting problem assumes that we are given as additional inputs a linear inequality $f(x) \geq 0$ (not necessarily valid for P) and a point $\bar{x}$ such that $g(\bar{x}) > 0$, $f(x) = 0$ for all $x \in P_o$ and $f(\bar{x}) = 0$. We will refer to the tilting problem as $TILT(f, g, \bar{x}, P_o)$.

We now discuss how solving problem (32), namely

$$\max\{\lambda : f(x) - \lambda g(x) \geq 0, \forall x \in P\},$$

also solves the tilting problem.

First, note that if phase I (Algorithm 1) detects that problem (32) is infeasible, by Proposition 3.2.2 this means that there exists a point $\hat{x} \in P$ such that $g(\hat{x}) = 0$ and $f(\hat{x}) < 0$ or there exists a ray $\hat{r}$ such that $g^o(\hat{r}) = 0$ and $f^o(\hat{r}) < 0$. Now notice that if the latter condition is true, since there exists $x^o \in P_o$ such that $g(x^o) = 0$, we can pick $\alpha \in \mathbb{R}_+$ large enough such that $\hat{x} = x^o + \alpha\hat{r} \in P$, $g(\hat{x}) = 0$ and $f(\hat{x}) < 0$. But then $f(\hat{x}) < 0$ implies $\hat{x}$ must be affinely independent of all points in P_o. Thus, we can simply return $h(x) = g(x)$ and $x' = \hat{x}$.

Second, note that the condition of the tilting problem that there exists $\bar{x}$ such that $f(\bar{x}) = 0$ and $g(\bar{x}) > 0$ implies that the problem is not unbounded.

Therefore, we may assume problem (32) is feasible and bounded. So, let λ^* be the optimal solution for problem (32). Proposition 3.2.6 implies that there exists $x^* \in P$ such that $g(x^*) > 0$ and $\lambda^* = \frac{f(x^*)}{g(x^*)}$. Therefore, we may return $h(x) = f(x) - \lambda^* g(x)$ and $x' = x^*$, since $g(x^*) > 0$ implies that x^* is affinely independent of all points in P_o.

As mentioned in Section 1.4, the tilting algorithm used in Applegate et al. [5] and Espinoza [47] is essentially the same as the Newton-Rhapson algorithm presented in this thesis.

We next review how we can use the tilting procedure to obtain facet-defining inequalities for $conv(P)$.

3.2.4.1 Using tilting to obtain facet-defining inequalities

Assume that we have a valid inequality for P that we know defines a non-empty face of $conv(P)$ and that separates a point x^* from P. We could, in principle, use such an inequality as a cut to strengthen our LP relaxation of P. Ideally, however, we would like to add to the LP relaxation the strongest possible valid inequality for P separating x^*. The tilting problem can be helpful in this context, since it can be used to obtain a facet-defining inequality for $conv(P)$ from the original inequality defining a non-empty face of $conv(P)$.

In order to see how, we need the following proposition, which characterizes facet-defining inequalities of $conv(P)$ and is given without proof in [47].

Proposition 3.2.10 (Espinoza [47]). *$vx \geq w$ is a facet-defining inequality for $conv(P)$ if and only if the following system of linear equations has a unique solution, the zero vector:*

$$\begin{array}{rcll} \pi p_i & = & 0 & \forall p_i \in P_o^{\perp} \\ \pi_o - \pi x_i & = & 0 & \forall x_i \in P_o \\ \pi_o - \pi \bar{x} & = & 0, & \end{array} \tag{42}$$

where P_o is a maximal set of affinely independent points in P that satisfy $vx = w$, $P_o^{\perp}$ is a basis for the space orthogonal to P, $\bar{x}$ is a point in P such that $v\bar{x} > w$ and $(\pi, \pi_o) \in \mathbb{R}^n \times \mathbb{R}$ is the set of variables of the above system.

Proof. Assume that P has dimension k.

First we prove that if $vx \geq w$ is a facet then 0 is the unique solution to the above LP.

Let y_o be an arbitrary point in P. Define E as a $|P_o^{\perp}| \times n$ matrix whose rows are the elements of $P_o^{\perp}$, and $\eta = Ey_o$.

From Nemhauser and Wolsey [85] (p. 91) we know that the following holds:

Let $F := \{x \in P : vx = w\}$. We have that F is a facet of $conv(P)$ if and only if $\lambda x = \lambda_o, \forall x \in F$ implies that $(\lambda, \lambda_o) = (uE + \alpha v, u\eta + \alpha w)$ for some $u \in \mathbb{R}^{|P_o^{\perp}|}$ and $\alpha \in \mathbb{R}$.

Since P_o is a maximal set of affinely independent points in F, we have that $\pi x = \pi_o$ for all $x \in F$. Thus, from Nemhauser and Wolsey's proposition, we can write

$(\pi, \pi_o) = (uE + \alpha v, u\eta + \alpha w)$.

Now, since $\pi\bar{x} = \pi_o$ we have $uE\bar{x} + \alpha v\bar{x} = u\eta + \alpha w$ and since $\bar{x} \in P$, we have that $E\bar{x} = \eta$ and thus $\alpha v\bar{x} = \alpha w$. But, since $v\bar{x} > w$, this implies that $\alpha = 0$.

Therefore $(\pi, \pi_o) = (uE, u\eta)$. Note that the equations $\pi p_i = 0$ for all $p_i \in P_o^{\perp}$ is the same as $\pi E^T = 0 \iff (uE)E^T = 0$. Since $P_o^{\perp}$ is a basis, EE^T is a square invertible matrix. Therefore, this system has a unique solution $u = 0$ and hence $(\pi, \pi_o) = (0, 0)$.

Let us now prove that if $vx \geq w$ is not a facet, then there is a nonzero solution to the LP.

To see that, note that $|P_o^{\perp}| = n - k$, and since $vx \geq w$ is not a facet, $|P_o| < k$. Therefore, the system of linear equations has $n + 1$ unknowns and at most n equations. Therefore it is an underdefined system and therefore has an infinite number of solutions. ∎

Proposition 3.2.10 can be used together with the tilting procedure in the following way: We start with a valid inequality $g(x) \geq 0$ defining a non-empty face of $conv(P)$. We are also given a set P_o of affinely independent points such that $g(x) = 0$ for all $x \in P_o$ (this is not necessarily a maximal set), a set $P_o^{\perp}$ of points in the space orthogonal to P (not necessarily a basis for it) and a point $\bar{x} \in P$ such that $g(\bar{x}) > 0$. At each step, we try to either increase the dimension of $P_o^{\perp}$ or P_o at the same time maintaining a valid inequality that is satisfied by the points in P_o at equality.

To do so, we try to find a nonzero solution of (42). This can be accomplished for instance by formulating an LP that maximizes the L_1 norm of (π, π_o) subject to the equations in (42) and that the L_∞ norm is bounded by 1. If the only solution is zero, we stop, since we know that $g(x) \geq 0$ defines a facet of $conv(P)$. If there is a nonzero solution (π, π_o), then $f(x) = \pi x - \pi_o$ is the appropriate input for the tilting procedure, which will return a new valid inequality with one more affinely independent point satisfying it at equality. However, the inequality that the tilting procedure returns might actually define an affine subspace where P lies in, in other words, if the tilting procedure returns $h(x) \geq 0$ as a valid inequality, it may be the case where $P \cap \{x : h(x) = 0\} = P$. The following proposition shows that this will happen only if $\lambda^+ = \lambda^- = 0$, where λ^+ and λ^- are the optimal solutions to $TILT(f, g, \bar{x}, P_o)$ and $TILT(-f, g, \bar{x}, P_o)$ respectively.

Proposition 3.2.11. *Let λ^+ and λ^- be the optimal solutions to $TILT(f, g, \bar{x}, P_o)$ and $TILT(-f, g, \bar{x}, P_o)$ respectively. Then $\lambda^+ = \lambda^- = 0$ if and only if $P \cap \{x : f(x) = 0\} = P$.*

Proof. We know that $\lambda^+ = \max\{\lambda : f(x) - \lambda g(x) \geq 0, \ \forall x \in P\}$ and $\lambda^- = \max\{\lambda : -f(x) - \lambda g(x) \geq 0, \ \forall x \in P\}$. Also, by assumption, the tilting procedure receives as an input a point $\bar{x} \in P$ such that $g(\bar{x}) > 0$.

So, if $P \cap \{x : f(x) = 0\} = P$, then $\lambda^+ \leq \frac{f(\bar{x})}{g(\bar{x})} = 0$. Moreover $\lambda = 0$ is feasible for $\max\{\lambda : f(x) - \lambda g(x) \geq 0, \ \forall x \in P\}$, thus $\lambda^+ = 0$. The proof that $\lambda^- = 0$ is analogous.

Finally, if $\lambda^+ = \lambda^- = 0$, then $f(x) - \lambda^+ g(x) = f(x) = \pi x - \pi_o \geq 0$ and $-f(x) - \lambda^- g(x) = -f(x) = -\pi x + \pi_o \geq 0$ are valid for P. Thus $f(x) = 0$ for all $x \in P$. ∎

Therefore, if $\lambda^+ = \lambda^- = 0$, Proposition 3.2.11 implies that the inequality $f(x)$ returned by the tilting procedure gives us a new point $\pi \in P_o^{\perp}$. Indeed, $f(x) = 0$ for all $x \in P$ implies that $\pi \in P_o^{\perp}$, and since (π, π_o) was obtained as a solution to (42) we have that π is orthogonal to all elements currently in $P_o^{\perp}$.

Now if either λ^+ or λ^- is nonzero, recall that our main goal is to obtain a facet-defining inequality of $conv(P)$ that separates a point x^* from P. In that context, we want to make sure that if the inequality $g(x) \geq 0$ separates a point x^* from P and λ^+ or λ^- are not zero, then at least one of the inequalities $f(x) - \lambda^+ g(x) \geq 0$ or $-f(x) - \lambda^- g(x) \geq 0$ obtained by tilting also separates x^* from P. The following proposition guarantess that this happens.

Proposition 3.2.12 (Espinoza [47])**.** *If λ^+ or λ^- are nonzero and $g(x^*) < 0$, then either $f(x^*) - \lambda^+ g(x^*) < 0$ or $-f(x^*) - \lambda^- g(x^*) < 0$.*

Proof. Note that since we have $\bar{x}$ such that $f(\bar{x}) = 0$ and $g(\bar{x}) > 0$, and since $f(\bar{x}) - \lambda^+ g(\bar{x}) \geq 0$ and $-f(\bar{x}) - \lambda^- g(\bar{x}) \geq 0$, we must have $\lambda^+ \leq 0$ and $\lambda^- \leq 0$. But since they cannot be both zero, we have that $\lambda^+ + \lambda^- < 0$.

Moreover, if we add the two inequalities $f(x) - \lambda^+ g(x) \geq 0$ to $-f(x) - \lambda^- g(x) \geq 0$ we get $-(\lambda^+ + \lambda^-) g(x) \geq 0$. Therefore since $\lambda^+ + \lambda^- < 0$, we have that $f(x) - \lambda^+ g(x) \geq 0$ and $-f(x) - \lambda^- g(x) \geq 0$ implies $g(x) \geq 0$. This implies that $\{x : f(x) - \lambda^+ g(x) \geq 0 \text{ and } -f(x) - \lambda^- g(x) \geq 0\} \subseteq \{x : g(x) \geq 0\}$, so at least one of the inequalities is violated by x^*. ■

These are the main ingredients of using the tilting procedure to get a facet-defining inequality of $conv(P)$ from an inequality defining a non-empty face of $conv(P)$. For more details on this, see [5, 47].

3.2.5 Connection to fractional programming

In this section we present a connection between problem (32) and fractional programming. It is worth noting that a connection between tilting and fractional programming has already been mentioned in Applegate et al. [6] and that Dinkelbach [43] already uses (32) in his algorithm for fractional programming (though for a bounded feasible region). Moreover, Wolsey [101] states the lifting problem for integer variables as also solving a fractional programming problem, though the method proposed to solve it is to enumerate all possible integer values of the denominator and solve all resulting MIPs. Nevertheless, we present such a connection here for completeness and to make it easier to understand the applicability (or lack thereof) of results from one area (lifting, tilting and fractional programming) to the others.

Fractional programming is an important problem with several applications and has received much attention in the research literature. The term usually refers to optimization of ratios or sum of ratios of functions (linear or nonlinear), but it also includes optimization of the largest/smallest ratios. The constraint set can be linear, nonlinear, continuous or integer. In the context of this thesis, we will focus on the case where we want to optimize a single ratio between two affine functions and the feasible space being a mixed-integer set defined by linear inequalities. Specifically, the mixed-integer linear fractional programming consists of solving

$$\min \left\{ \frac{f(x)}{g(x)} : x \in P \right\} \tag{43}$$

where $f(x)$ and $g(x)$ are affine functions of x, P is a mixed-integer set defined by linear inequalities and $g(x) > 0$ for all $x \in P$. For simplicity, we will call this problem the fractional programming problem.

The assumption that $g(x) > 0$ for all $x \in P$ is to make sure that $\frac{f(x)}{g(x)}$ is well-defined for all $x \in P$. Notice that if no assumptions were made on P, then there may not be a point attaining the smallest ratio. Indeed, consider the following very simple example:

$$\inf \left\{ \frac{1}{1+x} : x \geq 0 \right\}.$$

It is easy to see that the infimum of 0 is not attained for any $x \geq 0$ and thus problem (43) would not be well-defined. For this reason, it is common in the literature on fractional programming to assume that P is bounded (we will discuss this a bit further in a while).

Perhaps the best-known algorithm for tackling the purely linear case, that is, the case where there are no integrality requirements, is Dinkelbach's [44] algorithm, which applies to a wide range of fractional programming problems even when f, g and the description of P involve nonlinear constraints. Another well-known approach for such a case is that of Charnes and Cooper [26], which consists in defining a standard linear program over a new set of variables in such a way that the solution to this linear program can be translated to the optimal solution to problem (43). A third methodology consists in using simplex-like algorithms which iteratively explore extreme-point solutions (see, for example Martos [81, 82]). There are also several other methods used, which we will refrain from exhaustively listing, for example interior-point methods [4]. In the mixed-integer linear fractional programming case, the basic approach is to do branch-and-bound, as in the standard MIP case. For a comprehensive survey of these and other methods for fractional programming see Bajalinov [14], Stancu-Minasian [96] and Schaible [95].

Curiously, it seems that the case where P is bounded is the most studied case in the fractional programming literature. Indeed, the textbook of Stancu-Minasian [96] always makes this assumption. The textbook of Bajalinov [14] reserves most of its attention to this case, briefly mentioning what can happen in the unbounded case as well as some methods which work for either case.

Now notice that, even if we define the fractional programming problem as finding $\inf \left\{ \frac{f(x)}{g(x)} : x \in P \right\}$, we can rephrase it as finding a maximum λ such that $\lambda \leq \frac{f(x)}{g(x)}$ for all $x \in P$, and this problem is equivalent to problem (32). Therefore, solving problem (32) allows us to solve the fractional programming problem with an unbounded feasible region and Algorithm 2 can be interpreted as an extension of Dinkelbach's algorithm for such a case.

We do not know what to expect in terms of performance of the algorithms presented in this chapter in comparison with the algorithms specific for fractional programming. On one hand, one could argue that Algorithms 1 and 2 have to solve several MIPs and that Algorithm 3 explores an "unusual" branch-and-bound tree which may be much larger than any branch-and-bound tree explored in fractional programming. However, it seems that a great deal of the effort in fractional programming is to extend what is known for the MIP case (simplex, interior-point, branch-and-bound, etc.) and even though a great deal of work has been done over the years in fractional programming, we believe that MIP technology is still more developed than that used in fractional programming, especially in the implementation details and code optimization. Therefore, since all the algorithms presented in this chapter use MIP algorithms as subroutines, or can be implemented with a standard MIP solver, it may be the case that in practice they still may outperform the algorithms specific for fractional programming. Unfortunately, we were not able to implement or obtain any implementation of the fractional programming algorithms. In any case, our main focus was to solve the lifting problem for which most fractional programming algorithms may not be applied directly, since the assumption that $g(x) > 0$ for all $x \in P$ does not hold and therefore we did not pursue this line of research any further.

3.3 Multilifting

Consider now the case where we have $L > 1$ inequalities $g_l(x) \geq 0$ and we want to lift an inequality $f(x) \geq 0$ valid for $P(g_1, \ldots, g_L)$ into an inequality $f(x) - \sum_{l=1}^{L} \lambda_l g_l(x) \geq 0$ that is valid for P.

One possible way to obtain the λ coefficients is to sequentially lift each inequality $g_l(x) \geq 0$ one at a time, which was the focus of the previous section. Another possible way is to use super-additive lifting functions leading to sequence-independent lifting (for example [10, 11, 66, 67, 102]). A third way, proposed by Easton and Gutierrez [46], is to predetermine that λ is going to be a multiple of a fixed vector $\bar{\lambda} \in \mathbb{R}^L$, that is, $\lambda = \alpha\bar{\lambda}$ for some $\alpha \in \mathbb{R}$, consequently reducing the problem to the single-lifting case. All of these methods, however, cannot guarantee that all possible lifting coefficients are considered. In what follows, we study the general problem of obtaining the lifting coefficients λ.

We are interested in algorithms to find "good" values of $\lambda \in \mathbb{R}^L$. In the single-lifting case the "good" value of λ was the maximal value. In the multilifting case, we evaluate λ by means of a given cost vector $z \in \mathbb{R}^L_+$ such that $z \neq 0$. So we define the *multilifting problem* as

$$\max \{z\lambda : \lambda \in \Lambda(P)\} \tag{44}$$

where $\Lambda(P)$ is defined as

$$\Lambda(P) := \left\{ \lambda \in \mathbb{R}^L : f(x) - \sum_{l=1}^{L} \lambda_l g_l(x) \geq 0, \forall x \in P \right\}$$

and $g_l(x) \geq 0$ for all $x \in P$ and for all $l = 1, \ldots, L$. Moreover, we will assume throughout this section that $P(g_1, \ldots, g_L) \neq \emptyset$.

Multilifting is much more general than single-lifting and, as expected, more complex as well. This section is dedicated to start a discussion on the problem by understanding when it is feasible, bounded and what is the relationship between solving (44) and the other methods for obtaining valid lifting coefficients $\lambda \in \mathbb{R}^L$.

We start by analyzing when (44) is feasible. As in the single-lifting case, this is not a difficult task, as shown in the following proposition.

Proposition 3.3.1. *The set $\Lambda(P)$ is nonempty if and only if*

$$g_l(x) = 0, \ \forall l = 1, \ldots, L \text{ implies } f(x) \geq 0, \quad \forall x \in P \tag{45}$$

Proof. It is clear that if $\Lambda(P)$ is nonempty, then (45) has to hold.

Therefore we focus on proving the reverse implication. We will do so by induction on L.

The base case $L = 1$ is true due to Proposition 3.2.2.

Now assume that the statement is true for $L - 1$. We wish to prove that it is also true for L.

Notice that (45) is equivalent to

$$g_L(x) = 0 \text{ implies } f(x) \geq 0, \quad \forall x \in P(g_1, \ldots, g_{L-1}) \tag{46}$$

Moreover, since $P(g_1, \ldots, g_L) \neq \emptyset$, there exists a point x' in $P(g_1, \ldots, g_{L-1})$ such that $g_L(x') = 0$. Therefore (46) implies that

$$g_L^o(r) = 0 \text{ implies } f^o(r) \geq 0, \quad \forall r \in cone(P(g_1, \ldots, g_{L-1})).$$

But by Proposition 3.2.2, this implies that there exists λ_L such that $h(x) = f(x) - \lambda_L g_L(x) \geq 0$ is valid for $P(g_1, \ldots, g_{L-1})$. This means that there exists λ_L such that

$$g_l(x) = 0, \ \forall l = 1, \ldots, L-1 \text{ implies } h(x) \geq 0, \quad \forall x \in P.$$

By induction, there exists $\lambda_1, \ldots, \lambda_{L-1}$ such that $h(x) - \sum\limits_{l=1}^{L-1} \lambda_l g_l(x) \geq 0$ is valid for P, therefore $\Lambda(P)$ is not empty. ∎

Notice that, as in the case of single-lifting, it is also true that the multilifting problem is always feasible. Unfortunately, unboundedness is not as easy to characterize as in the single-lifting case. However, the following proposition relates the single-lifting coefficient with a particular problem in multilifting and helps us deal better with unboundedness.

Proposition 3.3.2. *Let $f(x) - \sum\limits_{l=1}^{k-1} \lambda_l^* g_l(x) \geq 0$ be a valid inequality for $P(g_k, \ldots, g_L)$ and consider the following two problems, which are assumed to be feasible,*

$$\max \left\{ \lambda_k : f(x) - \sum_{l=1}^{k-1} \lambda_l^* g_l(x) - \lambda_k g_k(x) \geq 0, \ \textit{for all } x \in P(g_{k+1}, \ldots, g_L) \right\} \tag{47}$$

$$\max \left\{ \lambda_k : \lambda \in \Lambda(P); \ \lambda_l = \lambda_l^*, \forall l = 1, \ldots, k-1 \right\}. \tag{48}$$

Then we have that (47) = (48).

Proof. Pick any point $\lambda' \in \Lambda(P)$ feasible for (48). By definition $f(x) - \sum\limits_{l=1}^{L} \lambda_l' g_l(x) \geq 0$ is valid for $x \in P$. But then, $f(x) - \sum\limits_{l=1}^{k} \lambda_l' g_l(x) = f(x) - \sum\limits_{l=1}^{k-1} \lambda_l^* g_l(x) - \lambda_k' g_k(x) \geq 0$ is valid for all $x \in P(g_{k+1}, \ldots, g_L)$. Therefore λ'_l is also feasible for (47) and hence (48) $\leq$ (47).

Conversely, let λ^*_k be the optimal solution to (47). Since the multilifting problem is always feasible, we can lift $f(x) - \sum_{l=1}^{k} \lambda^*_l g_l(x) \geq 0$ to a valid inequality for P, therefore there exists a point in $\Lambda(P)$ such that $\lambda_l = \lambda^*_l$ for all $l = 1, \ldots, k$. Thus (47)$\leq$ (48). ∎

Proposition 3.3.2 allows us to state the following corollary, giving a necessary condition for unboundedness.

Corollary 3.3.3. *If* (44) *is unbounded, then there exists a single lifting problem*

$$\max\{\lambda_k : f(x) - \lambda_k g_k(x) \geq 0, \text{ for all } x \in P(g_1, \ldots, g_{k-1}, g_{k+1}, \ldots, g_L)\}$$

that is also unbounded.

Proof. If (44) is unbounded, then there must exist $k \in 1, \ldots, L$ such that $\max\{\lambda_k : \lambda \in \Lambda(P)\} = \infty$. Thus, the result follows from Proposition 3.3.2. ∎

Notice that the reverse is not necessarily true. In fact, even if the single-lifting problem

$$\max\{\lambda_k : f(x) - \lambda_k g_k(x) \geq 0, \text{ for all } x \in P(g_1, \ldots, g_{k-1}, g_{k+1}, \ldots, g_L)\}$$

is unbounded, the multilifting problem can still be bounded if $z_k = 0$, that is, the objective function coefficient of λ_k in (44) is zero. However, Corollary 3.3.3 and Proposition 3.2.3 imply that as long as we guarantee that for each $l = 1, \ldots, L$ there exists a point $x^l \in P$ such that $g_l(x^l) > 0$, then (44) is always bounded for any objective function $z \in \mathbb{R}^L_+$.

Also notice that, in addition to helping us obtain necessary conditions for the unboundedness of (44), Proposition 3.3.2 establishes a relationship between the coefficients obtained by sequential lifting and the ones obtained by multilifting. As mentioned in the beginning of this section, besides sequential lifting, there were two other general approaches for obtaining the lifting coefficients λ. One is the Easton and Gutierrez [46] approach which, as mentioned before, reduces the multilifting problem to one single-lifting problem. The other approach is sequence-independent lifting. Since we know the relationship between multilifting and the first two approaches, we would like to end this section by understanding the relationship between sequence-independent lifting and (44). In order to understand this relationship, we make a series of observations about $\Lambda(P)$ that will be helpful later.

Observation 3.3.4. *Consider two mixed integer sets P and Q:*

1. *Consider $\lambda^1, \lambda^2 \in \mathbb{R}^L$. If $\lambda^1 \in \Lambda(P)$ and $\lambda^2 \leq \lambda^1$ then $\lambda^2 \in \Lambda(P)$.*

2. *The set $\Lambda(P)$ is a polyhedron.*

3. *If $\Lambda(P)$ is nonempty, then it is full-dimensional.*

4. *If $P \subseteq Q$ then $\Lambda(Q) \subseteq \Lambda(P)$.*

5. *Any inequality $\rho\lambda \leq \rho_o$ valid for all $\lambda \in \Lambda(P)$ satisfies $\rho \geq 0$.*

Proof. Observation 1 follows from the definition of $\Lambda(P)$ and it has Observation 3 as a consequence. Observation 2 follows from using the decomposition of $conv(P)$ into extreme points and rays. Observation 4 also follows from the definition of $\Lambda(P)$. Finally, by Observation 1, the unit vector $-e_i$ is a ray of $\Lambda(P)$. Therefore any inequality $\rho\lambda \geq \rho_o$ valid for all $\lambda \in \Lambda(P)$ satisfies $\rho(-e_i) \leq 0 \Rightarrow \rho_i \geq 0$ for all $i = 1, \ldots, L$. Thus Observation 5 is true. ∎

We now define some additional notation. Let $f(x) \geq 0$ be an inequality valid for $P(g_1, \ldots, g_L)$ and let $\alpha \in S_L$ be a permutation of $\{1, \ldots, L\}$. We denote by λ^α the set of coefficients λ obtained by sequentially lifting $f(x) \geq 0$ in the order $(\alpha(1), \ldots, \alpha(L))$, that is, $\lambda^\alpha_{\alpha(k)}$ is the optimal solution to

$$\max \left\{ \lambda_{\alpha(k)} : f(x) - \sum_{l=1}^{k-1} \lambda^\alpha_{\alpha(l)} g_{\alpha(l)}(x) - \lambda_{\alpha(k)} g_{\alpha(k)}(x) \geq 0, \forall x \in P(g_{\alpha(k+1)}, \ldots, g_{\alpha(L)}) \right\},$$

for all $k = 1, \dots, L$.

Now we need to establish what points in $\Lambda(P)$ can be obtained by performing sequential lifting. The following proposition allows us to do so.

Proposition 3.3.5. *Let $\alpha \in S_L$. Then λ^α is an extreme point of $\Lambda(P)$.*

Proof. We will prove this for α being the identity, since for all others it is the exact same proof, except that the notation becomes more complicated.

Suppose there exists $\lambda', \lambda'' \in \Lambda(P)$ such that λ^α is a convex combination of λ' and λ''.

Let k be the smallest index such that $\lambda_k^\alpha \neq \lambda'_k$ and $\lambda_k^\alpha \neq \lambda''_k$. We may assume without loss of generality that $\lambda_k^\alpha < \lambda'_k$.

But notice that since $\lambda'_l = \lambda_l^\alpha$ for all $l < k$, we have that λ'_k is feasible for (48), which contradicts the maximality of λ_k^α. Therefore, we have that there does not exist such an index k, which implies that $\lambda' = \lambda'' = \lambda^\alpha$ and hence λ^α is an extreme point of $\Lambda(P)$ ∎

We are now able to derive a relationship between sequence-independent lifting and the set $\Lambda(P)$.

Proposition 3.3.6. *Suppose that (44) is always bounded for any $z \in \mathbb{R}^L_+$. Then the following conditions are equivalent:*

1. *$\Lambda(P)$ has a unique extreme point.*

2. *$\Lambda(P) = \{\lambda \in \mathbb{R}^L : \lambda \leq u\}$ for some vector $u \in \mathbb{R}^L$.*

3. *$\lambda^\alpha = \lambda^{\alpha'}$ for any $\alpha, \alpha' \in S_L$.*

Proof. First, since (44) is always bounded for any $z \in \mathbb{R}^L_+$, let $u_l = \max\{\lambda_l : \lambda \in \Lambda(P)\}$ for all $l \in \{1, \dots, L\}$.

(3) ⇒ (1)

For all $l = 1, \ldots, L$, let $\alpha^l \in S_L$ be such that $\alpha^l(1) = l$, that is, the first constraint to be lifted sequentially is $g_l(x) \geq 0$. Proposition (47) implies that $\lambda_l^{\alpha^l} = u_l$ for all $l = 1, \ldots, L$. Now let $\lambda^* = \lambda^\alpha$ for all $\alpha \in S_L$ and note that $\lambda^{\alpha^l} = \lambda^*$ for all $l = 1, \ldots, L$, which implies that $\lambda^* = u$. Since all points $\lambda \in \Lambda(P)$ satisfy $\lambda \leq u = \lambda^*$, it follows that λ^* is the unique extreme point of $\Lambda(P)$.

(1) $\Rightarrow$ (2)

It is easy to see that $\lambda_l \leq u_l$ defines a facet of $\Lambda(P)$, since we can just pick any point $\bar{\lambda}$ satisfying $\bar{\lambda}_l = u_l$ and the affinely independent points $\bar{\lambda} - e_i$ in $\Lambda(P)$ for all $i = 1, \ldots, L$ such that $i \neq l$. This gives us a total of L affinely independent points satisfying $\lambda_l \leq u_l$ at equality.

Now notice that any facet has at least one extreme point satisfying it at equality. Thus, the unique extreme point of $\Lambda(P)$ is u. Now pick any facet-defining inequality $\rho x \leq \rho_o$ of $\Lambda(P)$. Since u is the unique extreme point we have that $\rho_o = \rho u$. But from Observation 3.3.4 we know $\rho \geq 0$. Thus $\rho x \leq \rho u$ is implied by $x \leq u$. So, the only facet-defining inequalities of $\Lambda(P)$ are $x \leq u$.

(1) $\Rightarrow$ (3)

Follows trivially from Proposition 3.3.5.

(2) $\Rightarrow$ (1)

Trivial. ∎

Proposition 3.3.6 characterizes when sequence independent lifting occurs in terms of the polyhedron $\Lambda(P)$ and suggests that we can try to pursue sequence independent lifting by restricting ourselves to subsets of $\Lambda(P)$ that are guaranteed to have a unique extreme point. Obtaining such sets is not necessarily an easy task, but we believe that such a relation is important to be able to start thinking about the subject in the context of multilifting.

3.4 Final remarks

So far, we were able to prove some properties of the multilifting problem and its relationship to other approaches for obtaining valid lifting coefficients $\lambda \in \mathbb{R}^L$.

With respect to actually solving the multilifting problem, the only way we know how to solve it is to use the following trivial algorithm: Let $EP(P)$ represent the set of all extreme points of $conv(P)$. Likewise, let $ER(P)$ represent the set of all extreme rays. It is easy to see that problem (44) is equivalent to the linear programming problem:

$$\max \sum_{l=1}^{L} z_l \lambda_l$$
$$f(x^k) - \sum_{l=1}^{L} \lambda_l g_l(x^k) \geq 0 \quad \forall x^k \in EP(P)$$
$$f^o(r^j) - \sum_{l=1}^{L} \lambda_l g_l^o(r^j) \geq 0 \quad \forall r^j \in ER(P).$$

This problem can be solved by a dynamic cut-generation algorithm making use of a mixed integer programming oracle for optimizing over P that would find inequalities that are violated.

It would be interesting to understand better the multilifting problem and to develop algorithms to solve it exactly or approximately within a reasonable time.

CHAPTER IV

BENCHMARKING MIXED-INTEGER KNAPSACK CUTS

4.1 Introduction

Consider positive integers n, m and let $d \in \mathbb{Q}^m,\ A \in \mathbb{Q}^{m\times n},\ l \in \{\mathbb{Q} \cup \{-\infty\}\}^n$ and $u \in \{\mathbb{Q} \cup \{+\infty\}\}^n$. Let $I \subseteq N := \{1, \ldots, n\}$ and consider the mixed integer set

$$P = \{x \in \mathbb{R}^n : Ax \leq d,\ l \leq x \leq u,\ x_i \in \mathbb{Z},\ \forall i \in I\}.$$

We say that a mixed integer knapsack set of the form

$$K = \{x \in \mathbb{R}^n : ax \leq b,\ l \leq x \leq u,\ x_i \in \mathbb{Z},\ \forall i \in I\}$$

with $b \in \mathbb{Q},\ a \in \mathbb{Q}^n$ is *implied* by P if (a, b) is a non-negative linear combination of rows obtained from (A, d). Observe that if K is implied by P, then $P \subseteq K$. Hence, any inequality which is valid for K is also valid for P. Recall that we denote by *mixed-integer knapsack cuts* (or *knapsack cuts* for short) the inequalities that are valid for some implied knapsack K.

As pointed out in Section 1.2, deriving strong knapsack cuts is of great practical importance to Mixed Integer Programming (MIP). In fact, most cutting planes known for general mixed integer programming are knapsack cuts. For example, Gomory Mixed Integer (GMI) cuts [56, 84] are knapsack cuts derived from the tableaus of linear programming relaxations, and Lifted Cover Inequalities [32, 64] are knapsack cuts derived from the original rows of P. Other classes of knapsack cuts include mixed-integer-rounding (MIR) cuts and their variations [31, 78, 85], split cuts [29], and group cuts [37, 57] — to name but a few.

In practice, the most successful classes of knapsack cuts for MIP are the GMI/MIR cuts [21]. Dash and Günlük [38] have recently done an empirical study to better understand this practical success. They show that, in a significant number of benchmark MIP instances, after adding GMI cuts for all the optimal tableau rows and reoptimizing the continuous relaxation, there are no more violated group cuts. Their results are quite surprising, and already suggest how strong are GMI cuts within the context of group cuts. Goycoolea [62] extended the results of Dash and Günlük by showing that in most cases, after adding GMI cuts for all the optimal tableau rows and re-optimizing the continuous relaxation, there are still violated mixed-integer knapsack cuts.

Though the results in Goycoolea [62] may seem to indicate that GMI cuts are not that strong in the context of knapsack cuts, it may still be the case that, even though there are violated knapsack cuts, these cuts do not significantly improve the performance already obtained by adding only GMI cuts. Therefore, a natural question to answer is: In the cases where there are some violated knapsack cuts, do these cuts give any significant improvement in the continuous relaxation bound?

A slightly different way to pose this question is: How much of the duality gap can we close using only knapsack cuts derived from a given family of implied knapsack sets? Answering such a question gives rise to an empirical methodology for evaluating sub-classes of knapsack cuts using the objective function of the continuous relaxation as a measure of quality. Formally, consider P as defined above, $c \in \mathbb{Q}^n$, and $\mathcal{C}$ a set of valid inequalities for P. Define

$$z^*(\mathcal{C}) = \min\{cx : Ax \leq d,\ l \leq x \leq u,\ \pi x \leq \pi_o\ \forall(\pi, \pi_o) \in \mathcal{C}\}.$$

Observe that the value $z^*(\mathcal{C})$ defines a benchmark by which to evaluate classes of cuts that are subsets of $\mathcal{C}$. In other words, if there is a subset $\mathcal{C}'$ of $\mathcal{C}$ such that $z^*(\mathcal{C}')$ is very close to $z^*(\mathcal{C})$, then $\mathcal{C}'$ must contain most of the "important" cuts in $\mathcal{C}$, so we can empirically say that $\mathcal{C}'$ is a "good" family of cuts. This idea will be applied in our context in the following way: Given a family of implied knapsack sets $\mathcal{K}$, let $\mathcal{C}^{\mathcal{K}}$ and $\mathcal{M}^{\mathcal{K}}$ represent, respectively, the set of all knapsack cuts and MIR cuts which can be derived from sets $K \in \mathcal{K}$. Since $\mathcal{M}^{\mathcal{K}} \subseteq \mathcal{C}^{\mathcal{K}}$ it is easy to see that $z^*(\mathcal{C}^{\mathcal{K}}) \geq z^*(\mathcal{M}^{\mathcal{K}})$ and that the proximity of these two values gives an indication of the strength of MIR inequalities derived from that particular family $\mathcal{K}$.

We perform experiments, computing the values $z^*(\mathcal{C}^{\mathcal{F}})$ and $z^*(\mathcal{C}^{\mathcal{T}})$, where $\mathcal{F}$ is the set of original formulation rows and $\mathcal{T}$ is the set of first tableau rows. Note that MIR cuts obtained from rows of a simplex tableau are simply GMI cuts. In this study we are able to obtain results for a large subset of MIPLIB 3.0 [19] and MIPLIB 2003 [2] instances, including general mixed integer problems.

There have been several related papers computing the value of the continuous relaxation using all cuts in a certain class of knapsack cuts. Boyd [23], Yan and Boyd [103] and Kaparis and Letchford [74] compute $z^*(\mathcal{C}^{\mathcal{F}})$, where $\mathcal{F}$ is the set of original formulation rows. They perform these tests on a subset of pure and mixed 0-1 instances in MIPLIB 3.0. Balas and Saxena [17] and Dash et al. [38] compute $z^*(\mathcal{M}^{\mathcal{A}})$ for all MIPLIB 3.0 problems, where $\mathcal{A}$ is the set of all implied knapsack polyhedra. Those results generalize the results of Fischetti and Lodi [49] and Bonami et al. [22] which consider Chvátal-Gomory cuts and projected Chvátal-Gomory cuts respectively. Finally, Fischetti and Lodi [48] compute $z^*(\mathcal{C}^{\mathcal{A}})$ but only for pure 0-1 problems. We summarize these results and compare them to ours in Table 1.

Finally, note that, since our method establishes benchmarks for knapsack cuts, we use rational arithmetic to ensure that the results obtained are accurate. Indeed, Boyd [23] already cites numerical difficulties in P2756, a pure 0-1 instance, and as a consequence, he gets different bounds than us and Kaparis and Letchford [74]. We expect that more numerical difficulties would arise in mixed integer instances with general bounds and considering other implied knapsack sets besides the formulation rows.

Table 1: Computational studies of classes of cuts

Paper	Instance set	Implied Knapsack Set ($\mathcal{K}$)	Class of Cuts
Boyd [23]	Pure 0-1	Formulation	Knapsack
Yan and Boyd [103]	Mixed 0-1	Formulation	Knapsack
Fischetti and Lodi [49]	Pure integer	All	CG
Bonami et al. [22]	General MIP	All	Pro-CG
Fischetti and Lodi [48]	Pure 0-1	All	Knapsack
Balas and Saxena [17]	General MIP	All	MIR
Dash et al. [38]	General MIP	All	MIR
Kaparis and Letchford [74]	Pure 0-1	Formulation	Knapsack
This thesis	General MIP	Formulation and Tableaus	Knapsack

In Section 4.2 we discuss how to solve the basic problem of separating over a single mixed integer knapsack set. In Section 4.3 we describe the computational improvements that need to be made to the basic separation problem in order to make it more computationally tractable. Computational results are presented in Section 4.4, while final remarks are given in Section 4.5.

4.2 *Identifying a violated knapsack cut*

Consider $x^* \in \mathbb{Q}^n$ and a feasible mixed integer knapsack polyhedron $K := \{x \in \mathbb{R}^n : ax \leq b,\ l \leq x \leq u,\ x_i \in \mathbb{Z},\ \forall i \in I\}$. In this section we will present a basic algorithm to resolve the following questions: Is $x^* \in conv(K)$? If not, can we find an inequality $\pi x \leq \pi_o$ which is valid for K, and such that $\pi x^* > \pi_o$?

Throughout this section we assume that K has no free variables, since it is easy to substitute a free variable by two non-negative variables. Under that assumption and, by possibly complementing variables, we will also assume without loss of generality that $l_i > -\infty$ for all $i = 1, \ldots, n$. Further, we assume that the bound constraints are tight, that is, for every finite bound there exists a feasible solution which meets that bound at equality. We may assume this condition because if the bounds are not tight it is trivial to strengthen and make them so. Finally, we assume that $l_i < u_i$ for all $i \in 1, \ldots, n$, as otherwise the problem is either infeasible or, in case of equality, the variable can be ignored and treated as a constant.

Let $\{x^1, x^2, \ldots, x^q\}$ and $\{r^1, r^2, \ldots, r^t\}$ represent the extreme points and rays of $conv(K)$. The following proposition, which follows from the work of Applegate et. al [5], suggests a natural algorithm for addressing our concern.

Proposition 4.2.1. *Consider the following linear programming (LP) problem with variables $u, v, \pi \in \mathbb{R}^n$, and $\pi_o \in \mathbb{R}$:*

$$
\begin{array}{llllr}
LP_1 : \min & \sum_{i=1}^{n}(u_i + v_i) & & & \\
s.t. & \pi x^k - \pi_o & \leq 0 & \forall k = 1 \ldots q & \text{(C1)} \\
 & \pi r^k & \leq 0 & \forall k = 1 \ldots t & \text{(C2)} \\
 & \pi x^* - \pi_o & = 1 & & \text{(C3)} \\
 & \pi + u - v & = 0 & & \text{(C4)} \\
 & u \geq 0, v \geq 0. & & &
\end{array}
$$

If this problem is infeasible, then $x^* \in conv(K)$, and thus there exists no knapsack cut violated by x^*. Otherwise, this problem admits an optimal solution (u, v, π, π_o) such that inequality $\pi x \leq \pi_o$ is a valid knapsack cut maximizing

$$\frac{\pi x^* - \pi_o}{||\pi||_1}$$

That is, the hyperplane defined by (π, π_o) maximizes the L_1 distance to x^*.

Because LP_1 has an exponential number of rows, we use a dynamic constraint generation algorithm to solve the problem. We begin with constraints (C3)-(C4) and a subset of constraints (C1)-(C2). We refer to this problem with a reduced number of constraints as the *master problem*. The dynamic constraint generation algorithm requires solving the problem

$$K(\pi) := \max\{\pi x : x \in K\} \tag{49}$$

at each iteration. We call $K(\pi)$ the *oracle problem* and in order to solve it we use a specialized mixed-integer knapsack solver. Details of this solver are described in Goycoolea [62].

If the oracle problem is unbounded at any given iteration, then there exists an extreme ray r^j of $conv(K)$ such that $\pi r^j > 0$. That is, there exists a violated constraint in (C2) which can be added. If this problem is not unbounded, then there exists an optimal solution corresponding to an extreme point x^k of $conv(K)$. If $\pi x^k > \pi_o$ then this extreme point results in a violated constraint of (C1) which can be added. Otherwise, it means that all constraints of the problem are satisfied. Because there can only be a finite number of extreme points and rays for a set K, this dynamic constraint generation algorithm is assured to converge in a finite number of iterations.

Notice that it is not hard to ensure that a mixed-integer knapsack solver will return an extreme ray of $conv(K)$ if the oracle sub-problem is unbounded. In fact, the solver described in Goycoolea [62] does so. However, when the oracle sub-problem has a finite optimum, it is much harder to ensure that the optimal solution returned by it is an extreme point of $conv(K)$. Despite this fact, we can still ensure that the dynamic constraint generation algorithm converges finitely. In fact, it is easy to see that if we use any finite set of points in K containing the extreme points of $conv(K)$, Proposition 4.2.1 still remains valid and the dynamic cut generation converges finitely. Espinoza [47] proposes that this can be achieved if the solution x^* returned by $K(\pi)$ satisfies the following two conditions:

1. $|x_i^*| \leq L$ for some constant L, for all $i \in I$;
2. x^* is an extreme point of $K_C(x^*) = K \cap \{x \in \mathbb{R}^n : x_i = x_i^*, \forall i \in I\}$.

In the remainder of this section we present a simple post-processing algorithm which, given an optimal solution to $K(\pi)$, returns another (or possibly the same) optimal solution satisfying the conditions proposed by Espinoza.

The post-processing procedure is described in Algorithm 4, and consists of two phases: the first phase (lines 1–6) ensures that the integer variables have bounded L_∞ norm by using domination, and the second phase (line 7) ensures that the solution satisfies condition 2.

In order to show that Algorithm 4 returns a solution that satisfies both conditions proposed by Espinoza, we review the concept of domination and a related proposition. Consider indices $i, j \in \{1, \ldots, n\}$, and non-zero integers k_i, k_j. If $a_i k_i + a_j k_j \geq 0$ and $\pi_i k_i + \pi_j k_j < 0$ we say that (i, j, k_i, k_j) defines an integer cost-domination tuple. If $i < j$, $k_i > 0$, $a_i k_i + a_j k_j \geq 0$ and $\pi_i k_i + \pi_j k_j = 0$ we say that (i, j, k_i, k_j) defines an integer lexicographic-domination tuple.

Proposition 4.2.2 (Goycoolea [62])**.** *Consider an integer cost/lexicographic-domination tuple (i, j, k_i, k_j) and let x be a feasible MIKP solution. Define $\delta \in \mathbb{Z}^n$ such that $\delta_i = k_i$, $\delta_j = k_j$ and $\delta_q = 0$ for all $q \in \{1, \ldots, n\} \setminus \{i, j\}$. If*

$$l_i + k_i \leq x_i \leq u_i + k_i \text{ and } l_j + k_j \leq x_j \leq u_j + k_j \tag{50}$$

Then x is cost/lexicographically-dominated by $x - \delta$. Moreover, we say that x violates the domination tuple (i, j, k_i, k_j). ∎

In Goycoolea [62] it is also shown that minimal integer-domination tuples exist, in other words, for every (i, j), there exists a minimal domination tuple (i, j, k_i^o, k_j^o) such that for every integer domination tuple (i, j, k_i, k_j), $|k_i| \geq |k_i^o|$ and $|k_j| \geq |k_j^o|$.

Since $\mathcal{L}$ is the set of all possible minimal integer lexicographic-domination tuples, phase 1 ensures that no tuple in $\mathcal{L}$ is violated by the returned optimal solution after the first phase. In the following proposition, we show that this fact implies that the integer variables have bounded L_∞ norm.

Proposition 4.2.3. *Consider a MIKP instance* $\max\{\pi x : x \in K\}$*, which has been preprocessed according to the preprocessing algorithm proposed in Goycoolea [62]. Let*

Algorithm 4: The Post-Processing Algorithm

Input: An instance of the mixed integer knapsack problem, an optimal solution x^* to this instance, and the set $\mathcal{L}$ of all minimal integer lexicographic-domination tuples.

1 **while** $\exists(i,j,k_i,k_j) \in \mathcal{L}$ *that is violated by* x^* **do**
2 $\quad \delta_i \longleftarrow k_i$
3 $\quad \delta_j \longleftarrow k_j$
4 $\quad \delta_q \longleftarrow 0, \forall q \neq i,j$
5 $\quad$ **while** $x^* - \delta$ *is feasible* **do**
6 $\quad\quad x^* \longleftarrow x^* - \delta$
7 $x^* \longleftarrow$ optimal solution to $\max\{cx : x \in K, x_i = x_i^*, \forall i \in I\}$

x^* correspond to an optimal solution of this instance which has been post-processed as indicated in lines 1–6 of Algorithm 4. There exists a constant L depending only on a, b, l and u, such that $|x_i^*| \leq L$ for all $i \in N$.

Proof. Consider an instance of MIKP and an optimal solution x^* obtained from the post-processing algorithm. Let $M_1 = \max\{(u_i - l_i) : u_i \neq \infty \text{ and } i \in I\}$. Let $M_2 = \max\{\max\{|k_i|, |k_j|\} : (i,j,k_i,k_j) \in \mathcal{L}\}$. Let $M = \max\{M_1, M_2\}$.

Observe, first, that at most one variable x_i can be such that $x_i > l_i + M$. In fact, suppose that there are two variables x_i, x_j such that $x_i > l_i + M$ and $x_j > l_j + M$. We know that there exists a minimal domination pair (i,j,k_i,k_j). Define $\delta = k_i \frac{\pi_i}{a_i} + k_j \frac{\pi_j}{a_j} \in \mathbb{Z}^n$. Since necessarily $u_i = u_j = \infty$ and $M \geq |k_i|$, $M \geq |k_j|$, it follows that x^* satisfies $l_i + k_i \leq x_i^* \leq u_i + k_i$ and $l_j + k_j \leq x_j^* \leq u_j + k_j$ and thus violates (i,j,k_i,k_j). So, from Proposition 4.2.2 we know that x^* is dominated by $x^* - \delta$. Given that x^* is optimal, we know that x^* cannot be cost-dominated. Thus $(i,j,k_i,k_j) \in \mathcal{L}$, which contradicts the fact that after post-processing there is no violated tuple in $\mathcal{L}$.

Now let x_i be the single variable such that $x_i^* > l_i + M$. If $a_i > 0$, then $x_i^* \leq \frac{b - \sum_{j \neq i} a_j x_j^*}{a_i}$. Since for all $j \neq i$, we have that $x_j^* \leq l_i + M$, it is clear that x_i^* can be bounded by a constant α_i depending only on a, b, l, u and M. If $a_i < 0$, then let

$\alpha_i = \max\left\{0, \frac{b - \sum_{j \neq i} a_j x_j^*}{a_i}\right\}$. If $x_i^* > \alpha_i$, then we can define a new solution x' by letting $x'_j = x^*_j, \forall j \neq i$ and $x'_i = \alpha_i$. Clearly x' is feasible. Goycoolea's [62] preprocessing algorithm implies that $a_i < 0 \iff \pi_i < 0$, thus since $a_i < 0$, we have that $\pi_i < 0$ and therefore $\pi x' < \pi x^*$. Since x^* is optimal we obtain a contradiction. By letting $L = \max\{M, \alpha_i\}$ we obtain our result. ∎

It is worth to note that the preprocessing algorithm of Goycoolea [62] is very simple to implement and only makes sure that the mixed-integer knapsack optimization problem is not trivial to solve, is bounded, and has no variables that can be immediately fixed at bounds.

It is easy to see how step 7 ensures condition 2, since $K_C(x^*)$ is a polyhedron and the optimal solution to the LP in step 7 will be an extreme point of $K_C(x^*)$. Therefore, we have shown that, after applying Algorithm 4 all conditions proposed by Espinoza are satisified.

Note that in the loop of lines 1–6, we are always getting a lexicographically smaller solution, so it is easy to see that the loop terminates after a finite number of iterations.

As a final remark, note that, in practice, it is faster to solve the dual of LP_1, since it has a fixed number of rows. This requires implementing this procedure as a column generation algorithm rather than as a a dynamic constraint generation algorithm. Also, observe that if this algorithm finds a violated knapsack cut, then this cut will correspond to a non-empty face of $conv(K)$. This follows follows from the fact that the optimal solution to LP_1 maximizes the L_1 distance from x^* to the hyperplane defined by (π, π_o). For more details, see Espinoza [47].

4.3 *Improving the performance*

The dynamic cut generation algorithm, as presented in Section 4.2, is sufficient to solve LP_1 and allows us to obtain separating knapsack cuts, or show that none exist. Unfortunately, the algorithm as presented is prohibitively slow. In Sections 4.3.1 -

4.3.5 we present several steps which help speed up the procedure.

4.3.1 Determine if the problem admits a trivial solution.

We first try to identify cases where there is a fast way to determine if $x^* \in conv(K)$ or produce a cut separating x^* from $conv(K)$. In order to achieve this we test for the following conditions.

1. **Is** $x^* \in conv(K)$**?**

   For this, define $N_K = \{i \in 1, \ldots, n : a_i \neq 0\}$. We know that $x^* \in conv(K)$ if any of the following conditions are met:

   - $a_i \in \{-1, 1\}$ for all $i \in N_K$ and $b \in \mathbb{Z}$,
   - $N_K \cap I = \emptyset$,
   - $x_i^* \in \mathbb{Z}$ for all $i \in I \cap N_K$.

   In fact, the first two conditions imply that $conv(K)$ is defined by the knapsack and bound constraints, and the third condition implies that x^* is feasible for K.

2. **Is** $x^* \notin conv(K)$**?**

   For this, attempt to find a violated MIR inequality. In our implementation we use the MIR separation heuristic described in [62]. If this heuristic is succesful we can terminate the algorithm early and utilize the violated MIR inequality as a separating knapsack cut. Observe that one could also attempt to find violated inequalities from among other classes of cuts which are easy to separate, such as lifted cover inequalities.

4.3.2 Eliminate variables from LP_1.

Say that a knapsack cut for K is *trivial* if it is implied by the linear programming relaxation of K. A proof of the following result concerning non-trivial knapsack cuts can be found in Atamtürk [10].

Proposition 4.3.1. *Every non-trivial facet-defining knapsack cut* $\pi x \leq \pi_o$ *of* $conv(K)$ *satisfies the following properties:*

(i) If $a_i > 0$*, then* $\pi_i \geq 0$*.*

(ii) If $a_i < 0$*, then* $\pi_i \leq 0$*.*

(iii) $\pi_i = 0$ *for all* $i \notin I$ *such that* $a_i > 0$ *and* $u_i = +\infty$*.*

(iv) $\pi_i = 0$ *for all* $i \notin I$ *such that* $a_i < 0$ *and* $l_i = -\infty$*.*

(v) There exists a constant $\alpha > 0$ *such that* $\pi_i = \alpha a_i$ *for all* $i \notin I$ *such that* $a_i > 0$ *and* $l_i = -\infty$*, and for all* $i \notin I$ *such that* $a_i < 0$ *and* $u_i = +\infty$*.*

The following result concerning violated and non-trivial knapsack cuts is a simple generalization of a remark made in Boyd [23].

Proposition 4.3.2. *Consider* $x^* \notin conv(K)$*. Let* $H^+ = \{i \in 1, \ldots, n : a_i > 0,\ x_i^* = l_i\}$ *and* $H^- = \{i \in 1, \ldots, n : a_i < 0,\ x_i^* = u_i\}$*. If there does not exist a trivial inequality separating* x^* *from* $conv(K)$*, then there exists a separating knapsack cut* $\pi x \leq \pi_o$ *such that* $\pi_i = 0,\ \forall i \in H^+ \cup H^-$*.*

Proof. Since $x^* \notin conv(K)$, then by assumption there exists a nontrivial facet-defining inequality $\hat{\pi} x \leq \hat{\pi}_o$ for $conv(K)$ violated by x^*. Define $\pi_o = \hat{\pi}_o - \sum_{i \in H^+} \hat{\pi}_i l_i - \sum_{i \in H^-} \hat{\pi}_i u_i$ and

$$\pi_i = \begin{cases} 0 & \text{if } i \in H^+ \cup H^- \\ \hat{\pi}_i & \text{otherwise.} \end{cases}$$

Let $x \in conv(K)$. Consider $i \in H^+$. From Proposition 4.3.1 we know that $a_i > 0$ implies $\hat{\pi}_i \geq 0$, and, thus, $\hat{\pi}_i x_i \geq \hat{\pi}_i l_i$. Likewise, consider $i \in H^-$. From Proposition 4.3.1 we know that $a_i < 0$ implies $\hat{\pi}_i \leq 0$, and, thus, $\hat{\pi}_i x_i \geq \hat{\pi}_i u_i$. Hence, $\hat{\pi}_o \geq$

$\hat{\pi}x \geq \pi x + \sum_{i \in H^+} \hat{\pi}_i l_i + \sum_{i \in H^-} \hat{\pi}_i u_i$, and, so, $\pi x \leq \pi_o$. On the other hand, $\hat{\pi}_o < \hat{\pi}x^* = \pi x^* + \sum_{i \in H^+} \hat{\pi}_i l_i + \sum_{i \in H^-} \hat{\pi}_i u_i$. Thus, $\pi x^* > \pi_o$, and we conclude the result. ■

From Proposition 4.3.1 and Proposition 4.3.2 we can see that variables π_i with $i \in 1, \ldots, n$ can be assumed to have value zero whenever any of the following conditions are met:

- $i \notin I,\ a_i > 0$ and $u_i = +\infty$,
- $i \notin I,\ a_i < 0$ and $l_i = -\infty$,
- $a_i > 0$ and $x_i^* = l_i$,
- $a_i < 0$ and $x_i^* = u_i$.

In these cases the corresponding variables can simply be eliminated from LP_1. Further, from Proposition 4.3.1 it can be seen that if we add a non-negative continuous variable α to LP_1, we can replace all variables π_i satisfying the conditions of (v) by the corresponding terms αa_i. This can effectively reduce the variable space by eliminating all remaining continuous unbounded variables. Finally, observe that Proposition 4.3.1 allows us to add non-negativity bounds on all varibales π_i such that $a_i > 0$, and non-positivity bounds on all variables π_i such that $a_i < 0$.

4.3.3 Fix variables and lift back again

Given a mixed integer knapsack set K and a point x^* we are concerned with determining if $x^* \in conv(K)$, or if we can find a separating hyperplane.

To help improve the performance, we can use a problem of smaller dimension to determine if $x^* \in conv(K)$. For that purpose, define $U = \{i \in 1, \ldots, n : x_i^* = u_i\}$ and $L = \{i \in 1, \ldots, n : x_i^* = l_i\}$ and let $K(L, U) = K \cap \{x \in \mathbb{R}^n : x_i = l_i\ \forall i \in L\} \cap \{x \in \mathbb{R}^n : x_i = u_i\ \forall i \in U\}$. It is easy to see that $x^* \in conv(K)$ if and only if $x^* \in conv(K(L, U))$. However, this latter problem is much easier to solve since we can treat all variables x_i with $i \in L \cup U$ as constants, thus obtaining a problem of smaller dimension. In practice, this means we can apply the dynamic cut generation methodology on the smaller knapsack constraint

$$\sum_{i \notin L \cup U} a_i x_i \leq b - \sum_{i \in L} a_i l_i - \sum_{i \in U} a_i u_i.$$

If solving this problem reveals that $x^* \in conv(K(L,U))$ then we conclude that $x^* \in conv(K)$. We call the problem of determining if $x^* \in conv(K(L,U))$ as the *membership* problem.

Using the membership problem can greatly help us speed up the separation process in case $x^* \in conv(K)$. However, if $x^* \notin conv(K(L,U))$ the membership problem will return an inequality $\sum_{i \notin L \cup U} \pi_i x_i \leq \pi_o$ separating x^* from $conv(K(L,U))$ but which is not necessarily valid for K.

To obtain a valid inequality for $conv(K)$ that separates x^* from it, we could just run the whole separation process in the full space once we determine that $x^* \notin conv(K)$. However, this approach seems to waste the effort in determining that $x^* \notin conv(K)$ and obtaining an inequality $\sum_{i \notin L \cup U} \pi_i x_i \leq \pi_o$ separating x^* from $K(L,U)$. Another alternative is to apply lifting to that constraint to obtain coefficients π_i for all $i \in L \cup U$ such that $\sum_{i \notin L \cup U} \pi_i x_i + \sum_{i \in L} \pi_i (x_i - l_i) + \sum_{i \in U} \pi_i (x_i - u_i) \leq \pi_o$ is valid for K.

In our study we used the sequential lifting procedure described in Chapter 3 which allow us to lift both continuous and general integer variables. The functions we use to obtain the lower dimension restriction of K are just the functions defined by the bounds, that is, $x_i - l_i$ for all $i \in L$ and $u_i - x_i$ for all $i \in U$. Also, the sequence in which we lifted the variables is completely arbitrary (from lowest to highest index).

4.3.4 Early termination rules

Recall that the dynamic cut generation algorithm described for solving LP_1 iterated as follows: At the beginning of each iteration it started with a solution (π, π_o) satisfying $\pi x^* = \pi_o + 1$. Then, the algorithm solved the oracle sub-problem $\max\{\pi x : x \in K\}$

to optimality. The stopping condition of the dynamic cut generation algorithm was that the optimal answer to this sub-problem should be finite, and of value less than or equal to π_o. In what follows we present two different ways in which this scheme can be modified in such a way as to still guarantee convergence.

- **Method 1:** If, while solving $\max\{\pi x : x \in K\}$, we encounter a solution $\hat{x}$ that is feasible for K and such that $\pi\hat{x} > \pi_o$, we can stop the execution of the oracle sub-problem and add the constraint $\pi\hat{x} \leq \pi_o$ to LP_1 rather than continuing to solve the oracle to optimality.

- **Method 2:** Let v^* be the optimal value of $\max\{\pi x : x \in K\}$. If $v^* > \pi_o$, then $\pi x \leq \pi_o$ is clearly not valid for $conv(K)$. However, if $\pi_o < v^* < \pi_o + 1$, then inequality $\pi x \leq v^*$ is both valid for $conv(K)$ and violated by x^*. In this case we may terminate the dynamic cut generation algorithm and conclude that $x^* \notin conv(K)$, providing the inequality (π, v^*) as a separating cut.

Observe that both methods cannot be used simultaneously, given that the latter requires solving the sub-problem to optimality.

4.3.5 Dealing with rational arithmetic

As mentioned in Section 4.1, we used rational arithmetic to be able to obtain accurate bounds for our benchmarks. However, as noted in Applegate et al. [7], doing all computations in rational arithmetic can be orders of magnitude slower than doing all computations in floating point arithmetic. We therefore used an approach similar to the one proposed in Dhiflaoui et al. [42], Koch [75] and Applegate et al. [7], which is to use floating point arithmetic at first, gradually increasing the number of bits used and only using rational arithmetic when strictly necessary. We call this the *increasing precision* approach.

An easy way to make sure that the separation problem returns a correct answer, is to ensure that the optimal solution to the master problem and the solution to the oracle problem $K(\pi) = \max\{\pi x : x \in K\}$ are both performed in exact arithmetic. However, in either case, we can use floating point arithmetic to speed up the computations.

Notice first that we solve the oracle problem $K(\pi)$ in order to verify if any constraint (C1) or (C2) are violated by the current candidate cut $\pi x \leq \pi_o$ obtained by the master problem. If we have a solution $\hat{x} \in K$ such that $\pi\hat{x} > \pi_o$, then $\hat{x}$ does not need to be the optimal solution to $K(\pi)$. In fact, we can just add the constraint $\pi x \leq \pi_o$ to (C1) and iterate. Therefore, we solve $K(\pi)$ using floating point arithmetic, and convert the solution x' it returns in floating point to a solution $\hat{x}$ in rational arithmetic and check if $\pi\hat{x} > \pi_o$. The only cases in which we need to either switch to a higher precision or solve $K(\pi)$ in exact arithmetic are the cases where the solution x' returned cannot be converted to a feasible solution $\hat{x} \in K$ or when the converted solution $\hat{x}$ satisfies $\pi\hat{x} \leq \pi_o$. In first case, we have no feasible solution and in the latter case, we need to make sure that it is indeed the case that $\pi x \leq \pi_o$ for all $x \in K$, otherwise we could generate an invalid cut.

With respect to solving the master problem, notice that any feasible solution to LP_1 will give us a cut that is valid for K and is violated by x^*. The objective function just serves as a guide so that the cut returned by LP_1 is "good" in a certain way (the distance from the hyperplane defined by (π, π_o) to x^* is maximized). However, if we end up with an inequality that is feasible but not optimal by using floating point arithmetic, we still get an inequality valid for K that is violated by x^*. Even though the solution obtained by "solving" LP_1 in floating point arithmetic may not be the true optimal, it is reasonable to expect that it is a "good" solution. Moreover, (π, π_o) does not even need to be feasible for LP_1, in the sense that it may violate constraint (C3) by a small amount as long as $\pi x^* > \pi_o$. Therefore, as long as we are sure that the solution to $K(\pi)$ is correct, that is, we are guaranteed that our current candidate cut $\pi x \leq \pi_o$ is indeed satisfied by all points in K, we will, in principle, not need to ever solve LP_1 in exact arithmetic. Due to this observation, we try to solve the master problem in floating point arithmetic and convert its floating point solution (π', π'_o) to a rational solution $(\hat{\pi}, \hat{\pi}_o)$ and call the oracle problem $K(\hat{\pi})$ to check if any constraints in (C1) or (C2) is violated.

The only problem that we may face is the case that $K(\hat{\pi})$ returns a point x^k (or a ray r^j) that violates constraint (C1) (or (C2)) but the violation is small enough that after adding the corresponding cut to the master problem and resolving it, the optimal solution to the master problem does not change, which would then lead to cycling. This can happen if we are using floating point arithmetic, since if the cuts are only violated by a small amount, floating point arithmetic solvers will consider that they are not violated at all and will conclude that the optimal solution does not change. In that case, we choose to start solving the master problem in a higher precision, or decide to solve it in rational arithmetic if our floating point representation is already using more than 1024 bits.

Notice that this approach is better than using an increasing precision exact LP solver (like the one from Applegate et al.[7]) each time we solve the master problem. This is true, since such solvers will always check the optimality of the basis in rational arithmetic each time it is invoked and that step is very expensive. We on the other hand do not need an optimal solution for LP_1 and therefore we can afford not to do exact solves unless it is needed due to the abovementioned cycling problems.

4.4 Computational experiments

In this section we describe the results of our computational experiments. All implementations were developed in the "C" and "C++" programming languages, using the Linux operating system (v2.4.27) on Intel Xeon dual-processor computers (2GB of RAM, at 2.66GHz). As mentioned before, generating cuts which are invalid is a real point of concern, so we use rational arithmetic to avoid any errors in computations. Specifically, we used the exact LP solver of Applegate et al. [7] for solving the master problem and the GNU Multiple Precision (GMP) Arithmetic library [63] for implementing all other computations.

In order to test our results we use the MIPLIB 3.0 [19] and MIPLIB 2003 [2] data sets (92 problems in total). Section 4.4.1 is dedicated to tests of the effectiveness of each of the improvements presented in Section 4.3. In Section 4.4.3 we present the benchmarks we obtained for MIK cuts and use it to evaluate the practical performance of MIR inequalities.

Throughout our experiments we computed the values $z^*(\mathcal{C}^{\mathcal{K}})$, that is, the value of the optimal solution to the LP relaxation of a mixed-integer program (P) after adding all possible mixed-integer knapsack cuts derived from mixed-integer knapsacks in $\mathcal{K}$. In these experiments we only consider the sets $\mathcal{K} = \mathcal{F}$, i.e., the family of knapsack sets induced by the original formulation rows, and $\mathcal{K} = \mathcal{T}$, i.e., the family of knapsack sets induced by the tableau rows of the optimal LP relaxation of the original MIP (without any cuts). Since we wish to compute $z^*(\mathcal{C}^{\mathcal{F}})$ and $z^*(\mathcal{C}^{\mathcal{T}})$ for all problems in MIPLIB 3.0 and MIPLIB 2003, we get a total of 184 instances.

4.4.1 Testing the improvements

We have presented several computational improvements needed to make the computation of $z^*(\mathcal{C}^{\mathcal{K}})$ practical. This section presents the computational experiments that were performed in order to test the efficacy of each feature and to present the best set of parameters that we have found.

We use the time it takes to compute $z^*(\mathcal{C}^{\mathcal{K}})$ for each of the instances as a measure of the quality of a given choice of parameters. We believe that this measure is good since it establishes a common goal for all methods and evaluates how fast can that goal be reached. This allows a fair comparison between methods that compromise the quality of the cuts obtained for gains in the time it takes to generate one cut. We also imposed the time limit of 3 hours for our tests. If $z^*(\mathcal{C}^{\mathcal{K}})$ has not been computed within the time limit, it is considered that the solution time was $+\infty$.

The only feature we applied directly without experimenting first was the test presented in Section 4.3.1 to determine if $x^* \in conv(K)$, since these are very computationally inexpensive and will clearly benefit our algorithm. Therefore, all the experiments performed assume that such a feature is always in use.

We tested several different combinations of the parameters presented throughout this chapter. To make the presentation clearer, we present in Table 2 a list of the different configurations used and the corresponding parameter settings.

Table 2: Parameter settings.

Parameter setting	Use MIR	Use Proposition 4.3.1	Use Proposition 4.3.2	Use membership mode	Use lifting	Use early termination rule 1	Use early termination rule 2	Increasing precision oracle	Increasing precision master
alloff	–	–	–	–	–	–	–	–	–
onlymir	Yes	–	–	–	–	–	–	–	–
onlyprop1	–	Yes	–	–	–	–	–	–	–
onlyprop2	–	–	Yes	–	–	–	–	–	–
onlyprop12	–	Yes	Yes	–	–	–	–	–	–
onlye1	–	–	–	–	–	Yes	–	–	–
onlye2	–	–	–	–	–	–	Yes	–	–
onlyor	–	–	–	–	–	–	–	Yes	–
onlymas	–	–	–	–	–	–	–	–	Yes
memb	–	–	–	Yes	–	–	–	–	–
lift	–	–	–	Yes	Yes	–	–	–	–
membincmas	–	–	–	Yes	–	–	–	–	Yes
best	Yes	Yes	Yes	Yes	Yes	–	–	–	Yes

We first tested the effect of each of the proposed improvements individually, to assess which of them seem promising when compared to doing the straightforward appraoch of solving the separation problem without any improvements (**alloff** parameter setting). Since there are many features to be tested, we will present them grouped as they were presented throughout this chapter.

Figure 3 presents the results of using MIR inequalities as a quick heuristic to determine if there is a MIK cut separating x^* from $conv(K)$. Since all our results on this section are presented using performance profiles (including Figure 3) we will briefly explain how to read performance profiles. Performance profiles are used to compare two or more solution strategies according to some metric. In our case, the metric used is the time required to compute $z^*(\mathcal{C}^{\mathcal{K}})$. Each point (x, y) of the curves tells us that in y percent of the instances, this particular strategy was at most x times slower than the fastest solver for that instance (for a more detailed explanation of performance profiles, see [45]). For instance, by looking at the point where $x = 1$ we see that **onlymir** is faster than **alloff** in about 26% of the instances and **alloff** is faster in about 16% of the instances. By looking at the point where $x = 10$, we can see that **onlymir** is at most 10 times slower than **alloff** in 40% of the instances, while **alloff** is at most 10 times slower in a little over 35% of the instances. From the figure it is also clear that **onlymir** was not able to compute $z^*(\mathcal{C}^{\mathcal{K}})$ in 60% of the instances, since the plot for **onlymir** never goes above 40%. Likewise, **alloff** was not able to compute $z^*(\mathcal{C}^{\mathcal{K}})$ in about 64% of the instances.

In general, a quick way to interpret performance profiles is to look at the lines. If one line is always above the other, then the corresponding strategy is considered to perform better. Therefore, the experiments show that using MIR as a quick heuristic to determine if there is a cut separating x^* from $conv(K)$ is good in practice. Notice that it was expected that MIR would make each round of cuts faster, since the heuristic we use to separate them is extremely fast. However, it could be the case that the MIR cuts were too weak, in which case one could need many more rounds of cuts to reach $z^*(\mathcal{C}^\mathcal{K})$, which could eventually make the overall running time slower. The experiments show that this does not happen.

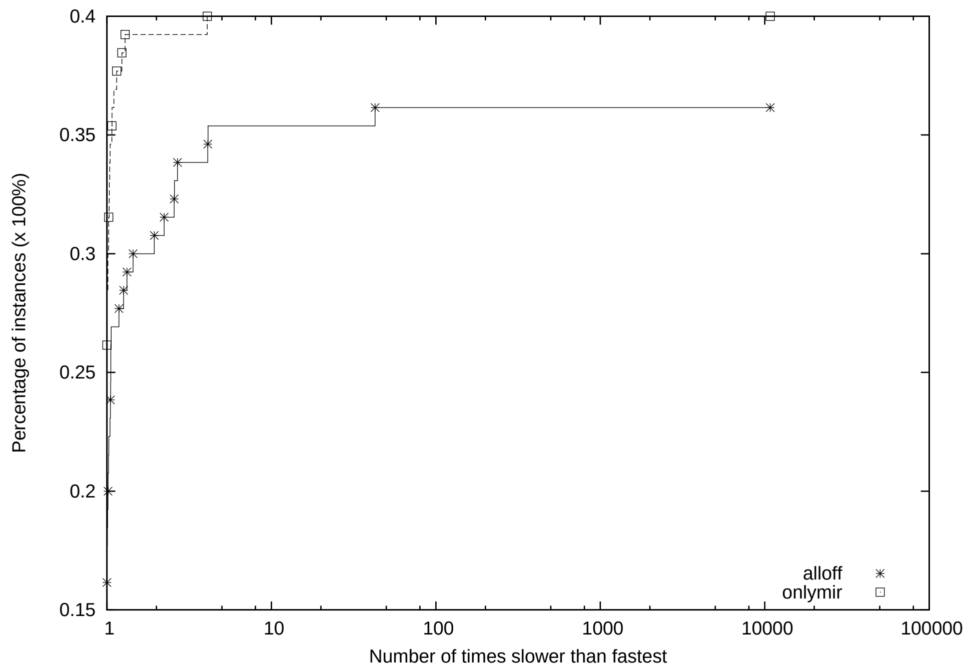

Figure 3: Performance profile of using MIR heuristic.

Figure 4 shows how much Propositions 4.3.1 and 4.3.2 help compared to the **alloff** setting. It is clear that using Proposition 4.3.1 helps reduce the overall running time. The results for Proposition 4.3.2 are not that significant. Indeed there seems to be a slight gain in performance by using Proposition 4.3.2. Even though the gain may be small, it is certainly present and we found that testing in other parameter settings, the use of Proposition 4.3.2 does not seem to cause any loss in performance. Therefore we decided it was a good idea to use it.

Figure 5 shows the results of using early termination rules 1 and 2, presented in Section 4.3.4. Notice that the scale of the horizontal axis has changed. The maximum value displayed is now 2.5. We choose to do so since the plots did not change for the values of the horizontal axis greater than 2.5. The results are somewhat surprising:

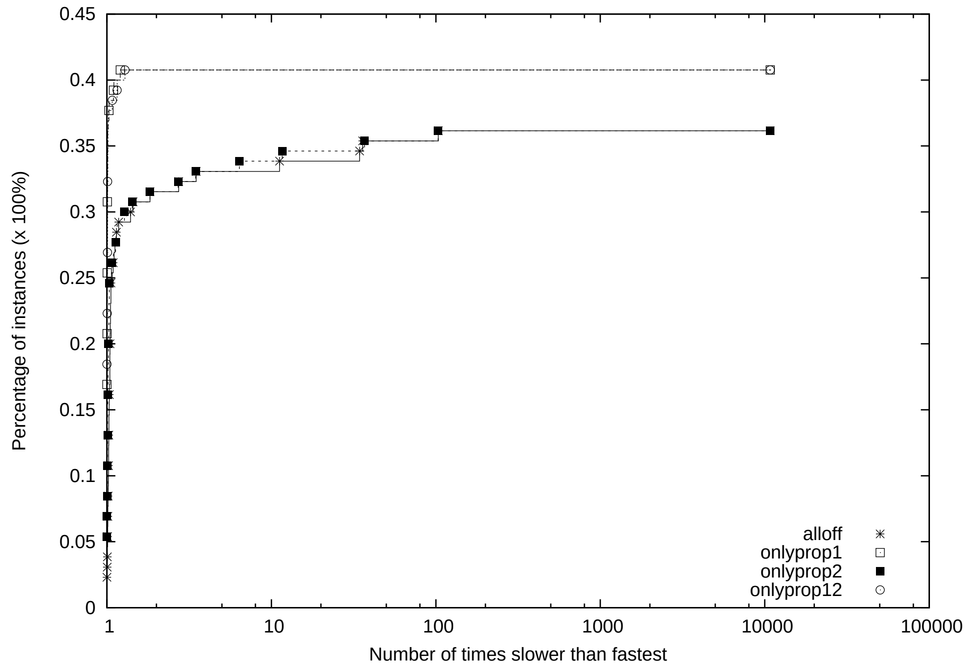

Figure 4: Performance profile of using Propositions 4.3.1 and 4.3.2.

it seems no strategy is clearly better than the others. On one hand, using either early termination rule seems to make the procedure faster in some instances, but it has a tailing off effect, making the procedure slower in some harder instances. In general, though, it seems that using early termination rule 2 is better than using rule 1. These experiments show that the tradeoff between making each round of cuts faster by possibly compromising the quality of the cuts does not necessarily make the whole process converge faster. In the end, from these experiments, we concluded that it was better not to use either of the early termination rules. Our reasoning was that it was better to have a method that solves all instances not taking much more time than the faster solver instead of a method that solves some of the instances faster, but some much slower.

In Figure 6 we present the results for testing the strategies for dealing with rational arithmetic presented in Section 4.3.5. Recall that we proposed in that section that we could solve both the master problem in increasing precision or the oracle problem

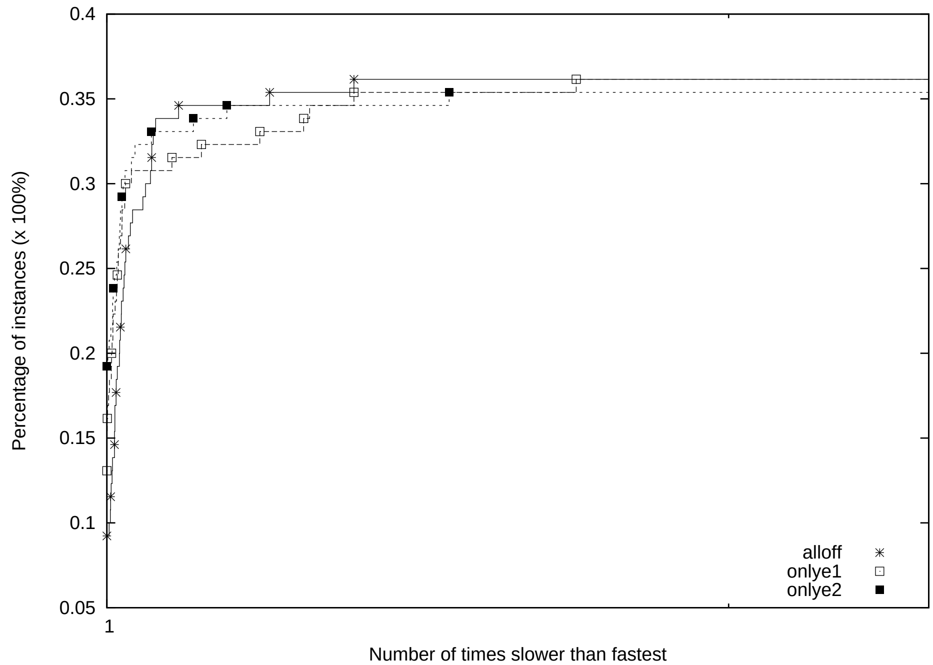

Figure 5: Performance profile of the early termination rules.

$K(\pi)$ in increasing precision. Again, the results are surprising. It is clear that solving the master problem in increasing precision is a good strategy. However, note that using an increasing precision oracle actually makes the performance worse. We believe this is due to the fact that more often than not the effort of solving the oracle in floating point arithmetic is wasted, since its solution is either not optimal or not feasible.

We leave the analysis of the strategies involving lifting and membership mode (presented in Section 4.3.3) to the next section, since those results are relevant for Chapter 3 and deserve a more thorough analysis. To finalize this subsection, we present in Figure 7 the performance profile of the best strategy we found according to our experiments. It is worth noting that we tried small variations of this strategy by turning on/off some individual parameters, but strategy **best** still outperformed all others. It is clear that there is a substantial gain in performance obtained by combining all techniques presented in this chapter. Indeed we believe it would be

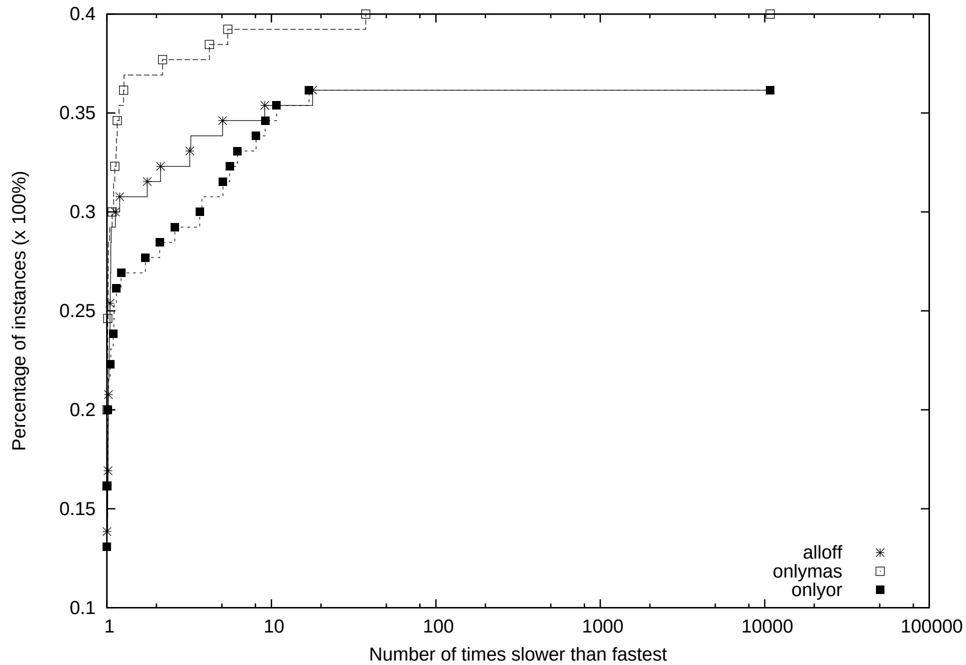

Figure 6: Performance profile of the different increasing precision rules.

extremely difficult to compute $z^*(\mathcal{C}^{\mathcal{K}})$ for most instances without these improvements, even if given large amounts of computing time. Algorithm 5 summarizes the final algorithm using the best combination of parameters found by our experiments.

4.4.2 Testing the effectiveness of lifting

Even though the tests presented in this subsection are similar to the ones presented so far, we decided to devote a separate subsection to lifting since these tests are relevant to both the current chapter and Chapter 3. We will first focus on the **memb**, **lift** and **alloff** strategies.

Recall that the strategy **memb** is to use $K(L,U)$ to determine if $x^* \in conv(K)$ and, in case $x^* \notin conv(K)$ then obtain a valid inequality for K violated by x^* solving the separation problem LP_1 on the whole set K. Strategy **lift** instead uses the inequality separating x^* from $conv(K(L,U))$ as a starting point and sequentially lifts the coefficients of variables in L and U to obtain a valid inequality for K. Figure 8 shows that, as expected, using **memb** is better than not. Also, it is easy to see that

Algorithm 5: KNAPSACK_SEPARATOR(K, x^*)

Input: A mixed integer knapsack set K, and a vector x^*.
Output: An answer to the question "Is $x^* \in conv(K)$?". In case that the answer is FALSE, the algorithm also returns a separating cut.

1. Test if x^* and K verify any of the sufficient conditions described in Section 4.3.1 for showing that $x^* \in conv(K)$. If the answer is affirmative then STOP and report that $x^* \in conv(K)$.
2. Run the MIR separation heuristic using K and x^* as input. If this heuristic finds a violated MIR cut then STOP, report that $x^* \notin conv(K)$, and return the MIR inequality which was found.
3. Simplify the problem by fixing variables which are at their bounds, as indicated in Section 4.3.3, and reduce to a smaller dimensional knapsack separation problem.
4. Formulate problem LP_1 in the reduced variable space without adding any of the constraints in (C1)-(C2).
5. Apply Propositions 4.3.1 and 4.3.2 to simplify LP_1 by eliminating variables from consideration and adding bounds.
6. Solve LP_1 using the dynamic row generation algorithm, using increasing precision for the master and rational arithmetic for the oracle.
7. If solving LP_1 reveals that $x^* \in conv(K)$, then STOP and report that $x^* \in conv(K)$.
8. Let $(\hat{\pi}, \hat{\pi}_o)$ represent the optimal solution to LP_1.
9. Lift cut $\hat{\pi}x \leq \hat{\pi}_o$ to obtain a cut $\pi x \leq \pi_o$ which is valid for K.
10. STOP and return the cut (π, π_o).

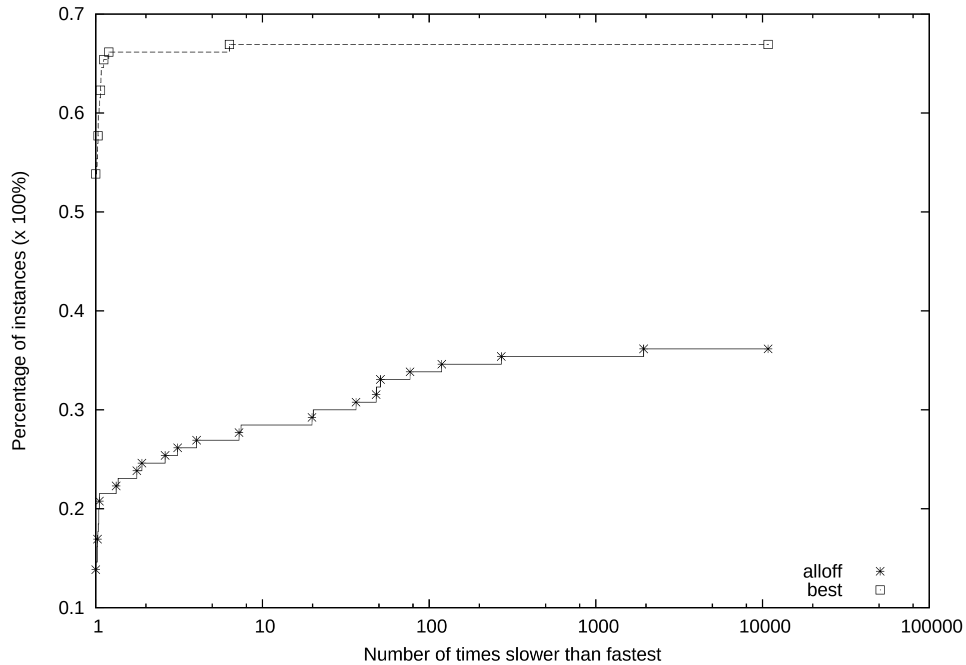

Figure 7: Performance profile of the best configuration.

strategy **lift** is better than **memb** or **alloff**. This result is relevant not only in the context of separating mixed-integer knapsack cuts. It also shows that, if you are given an optimization oracle that enables you to optimize over a mixed-integer set K and possibly some restrictions of it obtained by fixing certain variables at some values, it is better to use the lifting approach than to use the straightforward separation approach. This shows that the algorithms developed in Chapter 3 produce good results in practice, at least in this context.

To try to understand a little more why **lift** performs better than **memb**, we computed some statistics to try to understand where most of the computing time is spent. For those experiments, we only considered instances for which there was some gain in bound by using MIK cuts. The reason for doing so is that for all other instances we expect very few (if any) knapsack cuts to be generated and therefore **lift** or **memb** would have little or no difference. This leaves us with a total of 59 instances. Results are presented in Table 3. Row *Total time* represents the total

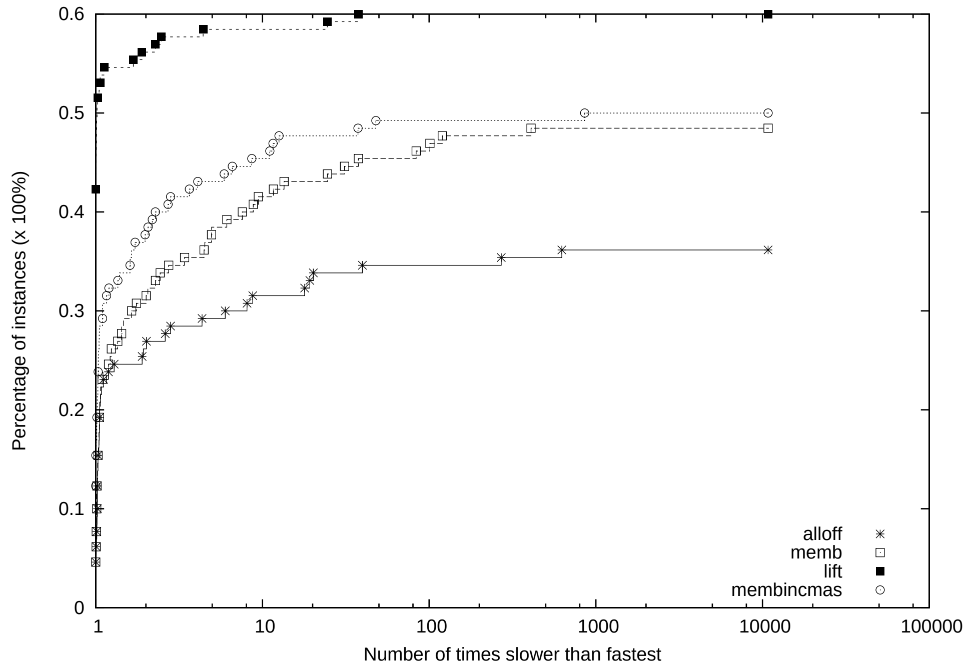

Figure 8: Performance profile of different lifting strategies.

amount of time used to compute $z^*(\mathcal{C}^{\mathcal{K}})$ for all 59 instances (adding up all times). If an instance was not completed within 3 hours, the amount of time considered for calculating *Total time* was 10800 seconds (3 hours). Rows *Oracle time* and *Master time* represent the total time added for all instances that was spent in oracle and master calls respectively (in seconds and in percentage of the *Total time*). Rows *Average oracle call time* and *Average master call time* represent the average time, in seconds, it takes for each oracle and master call to be completed, respectively.

The results in Table 3 are surprising. We expected the oracle calls in **lift** to take less time in average than the oracle calls in **memb**, since after establishing that $x^* \notin conv(K)$ all oracle calls in **memb** are performed in the full variable space, while the oracle calls in **lift** are not — all variables not yet lifted are considered to be fixed.

Also surprising is the fact that almost 94% of the time in **memb** is spent in solving the master problem. In fact, such a startling statistic raises the question if the only reason why **lift** outperforms **memb** is that we are not efficient enough in solving the master problem. To try to answer to such a question, we ran **membincmas**, which is **memb** but now using the increasing precision strategy for solving the master problem, which would make it be solved much faster. Notice that this can be considered an unfair advantage to **membincmas** if compared to **lift**, since the first uses floating point arithmetic in increasing precision and the latter uses only rational arithmetic, which is known to be orders of magnitude slower. However, as can be seen in Figure 8, the **lift** strategy still outperforms **membincmas**, which indicates that it is indeed a successful strategy.

Table 3: Statistics for **lift** vs. **memb**.

		lift	**memb**
Total time	(s)	254761	427739
Oracle time	(s)	105722	1102
	%	41%	0.3%
Master time	(s)	120983	399964
	%	47%	94%
Average oracle call time	(s)	0.06	0.003
Average master call time	(s)	0.39	0.95

Finally we conclude by noting that one of the possible reasons why **lift** performs better than **memb** and **membincmas** is that using sequential lifting, we have a guarantee that the inequality that we obtain at the end of the process has a high dimension. On the other hand, doing the separation using LP_1 straight away does not have such guarantees. Thus, **lift** should produce better inequalities than **memb/membincmas**, converging faster to $z^*(\mathcal{C}^{\mathcal{K}})$.

4.4.3 Benchmarking the MIR inequalities

We now describe our experiments computing the values $z^*(\mathcal{C}^{\mathcal{K}})$ and $z^*(\mathcal{M}^{\mathcal{K}})$ and our use of these values to benchmark the performance of MIR inequalities, as detailed in Section 4.1.

As a first step, we implemented the MIR separation heuristic described in Goycoolea [62]. This heuristic takes as input a mixed integer knapsack polyhedron K, a vector x^*, and attempts to find an MIR inequality which is valid for K and violated by x^*. In order to do this, the heuristic scales the knapsack set and complements variables in different ways, attempting to obtain an MIR inequality each time by applying a rounding procedure. We henceforth refer to this heuristic as MIR_HEURISTIC(K, x^*). Given a mixed integer programming problem and a set of implied mixed integer knapsack polyedra, we can use this heuristic to compute an approximation of $z^*(\mathcal{M}^{\mathcal{K}})$ as shown in Algorithm 6. This is only an approximation of the value $z^*(\mathcal{M}^{\mathcal{K}})$, because there could potentially be violated MIR inequalities for a polyhedron $K \in \mathcal{K}$ which MIR_HEURISTIC fails to find. A better approximation can probably be obtained by using the approch of Balas and Saxena [17] or the approach of Dash et al. [38] as a subroutine instead of using MIR_HEURISTIC.

Algorithm 6: Computing an approximation of $z^*(\mathcal{M}^{\mathcal{K}})$

Input: A vector $c \in \mathbb{R}^n$, a mixed integer polyhedron P, and a set of implied mixed integer knapsack polyhedra $\mathcal{K}$.

1 Let R be the linear programming relaxation of P;
2 Let x^* be the optimal solution of $\max\{cx : x \in R\}$;
3 Let $\mathcal{C} = \emptyset$;
4 **foreach** $K \in \mathcal{K}$ **do**
5   **if** MIR_HEURISTIC(K, x^*) *finds a violated cut* (π, π_o) **then**
6   Add cut (π, π_o) to $\mathcal{C}$;
7 **if** $\mathcal{C} \neq \emptyset$ **then**
8   Add cuts in $\mathcal{C}$ to R;
9   Resolve $\max\{cx : x \in R\}$ letting x^* correspond to the optimal solution;
10   Go to line 3;
11 **else**
12   Stop.

As a second step, we implemented the KNAPSACK_SEPARATOR algorithm described in Section 4.2 (for a summary see Algorithm 5). Observe from Algorithm 5 that KNAPSACK_SEPARATOR needs an MIR generator as a subroutine. We use MIR_HEURISTIC for this purpose. Given a mixed integer programming problem and a set of implied mixed integer knapsack polyedra we can use KNAPSACK_SEPARATOR to compute $z^*(\mathcal{C}^{\mathcal{K}})$ as shown in Algorithm 7.

Algorithm 7: Computing $z^*(\mathcal{C}^{\mathcal{K}})$

Input: A vector $c \in \mathbb{R}^n$, a mixed integer polyhedron P, and a set of implied mixed integer knapsack polyhedra $\mathcal{K}$.

1 Let R be the linear programming relaxation of P;
2 Let x^* be the optimal solution of $\max\{cx : x \in R\}$;
3 Let $\mathcal{C} = \emptyset$;
4 **foreach** $K \in \mathcal{K}$ **do**
5   **if** KNAPSACK_SEPARATOR(K, x^*) *finds a violated cut* (π, π_o) **then**
6   Add cut (π, π_o) to $\mathcal{C}$;
7 **if** $\mathcal{C} \neq \emptyset$ **then**
8   Add cuts in $\mathcal{C}$ to R;
9   Resolve $\max\{cx : x \in R\}$ letting x^* correspond to the optimal solution;
10   Go to line 3;
11 **else**
12   Stop.

The following proposition and the fact that KNAPSACK_SEPARATOR returns a nonempty face of $conv(K)$ ensure the finiteness of Algorithm 7:

Proposition 4.4.1. *If each cut added in line 6 of Algorithm 7 defines a nonempty face of $conv(K)$, then Algorithm 7 terminates in a finite number of iterations.*

Proof. Follows immediately, since for every $K \in \mathcal{K}$, $conv(K)$ has only a finite number of nonempty faces. ∎

Tables 4.4.3.1 and 4.4.3.2 present the results for $\mathcal{K} = \mathcal{F}$ and $\mathcal{K} = \mathcal{T}$ respectively. For each problem instance let z^*_{UB} represent the value of the optimal (or best known) solution and z^*_{LP} the LP relaxation value. Also, let $z^{\mathcal{K}}_M$ be the bound obtained by adding only MIR inequalities according to Algorithm 6. For each set $\mathcal{K}$ and each instance we compute the following performance measures:

ORIG-GAP: Performance of the original LP formulation. That is, the value of the LP relaxation gap

$$\frac{z^*_{UB} - z^*_{LP}}{|z^*_{UB}|}.$$

MIR: Performance of MIR separation heuristic. That is, how much of the LP gap was closed by the heuristic

$$\frac{z^{\mathcal{K}}_{M} - z^*_{LP}}{z^*_{UB} - z^*_{LP}}.$$

KNAP: Performance of the knapsack cuts. That is, how much of the LP gap was closed by the knapsack cuts

$$\frac{z^*(\mathcal{C}^{\mathcal{K}}) - z^*_{LP}}{z^*_{UB} - z^*_{LP}}.$$

MIR-REL: Relative performance of MIR separation heuristic. That is, how much of the LP gap closed by the knapsack cuts was closed by the MIR cuts

$$\frac{z^{\mathcal{K}}_{M} - z^*_{LP}}{z^*(\mathcal{C}^{\mathcal{K}}) - z^*_{LP}}$$

4.4.3.1 Knapsack cuts derived from formulation rows

We now discuss the computational results obtained for $\mathcal{K} = \mathcal{F}$. Of the 92 instances in MIPLIB 3.0 and MIPLIB2003, we were able to compute the values $z^*(\mathcal{C}^{\mathcal{K}})$ for 83 of them. The 9 problems which we were unable to solve were atlanta-ip, cap6000, dano3mip, harp2, momentum3, markshare1, markshare2, mkc and stp3d. Of the 83 problems solved, 4 of them were such that ORIG-GAP was equal to 0.0 (disctom, dsbmip, enigma, and noswot), and 53 of them were such that KNAP and MIR were both equal to 0.0. In Table 4.4.3.1 we present the results for the remaining 26 problems. That is, the only ones in which knapsack cuts improved the bound.

Table 4: Benchmarks for Formulation Closure

Instance	ORIG-GAP	MIR	KNAP	MIR-REL
arki001	0.02%	12.99%	13.12%	99.01 %
fiber	61.55%	91.07%	93.82%	97.06 %
gen	0.16%	99.78%	99.78%	100 %
gesa2	1.18%	69.96%	71.03%	98.48 %
gesa3	0.56%	47.80%	49.33%	96.90 %
gt2	36.41%	92.56%	94.52%	97.93 %
l152lav	1.39%	0.01%	1.36%	0.41 %
lseu	25.48%	67.91%	76.09%	89.25 %
mitre	0.36%	89.14%	100.00%	89.14 %
mod008	5.23%	71.23%	89.21%	79.84 %
mod010	0.24%	18.34%	18.34%	100 %
msc98-ip	1.61%	0.16%	0.16%	100%
net12	91.94%	1.39%	1.39%	100%
nsrand-ipx	4.53%	9.93%	9.93%	100%
p0033	18.40%	76.33%	87.42%	87.31 %
p0201	9.72%	33.78%	33.78%	100 %
p0282	31.56%	94.08%	98.59%	95.42 %
p0548	96.37%	53.69%	84.34%	62.76 %
p2756	13.93%	44.46%	86.35%	51.49 %
qnet1	10.95%	50.48%	89.06%	56.68 %
qnet1_o	24.54%	84.32%	95.12%	88.65 %
rgn	40.63%	57.49	57.49%	100 %
roll3000	13.91%	55.28%	55.28%	100%
sp97ar	1.38%	1.70%	1.70%	100%
timtab1	96.25%	21.13%	21.13%	100%
timtab2	93.49%	12.79%	12.79%	100%

First, note that knapsack cuts alone can considerably close the remaining LP gap in some problems (column KNAP). In fact, in 11 problems out of the 26 problems in which knapsack cuts improved the gap, over 84% of the gap was closed, and in 15 out of 26 problems, over 50 % of the gap was closed. On average, the GAP closed by the knapsack cuts among these 26 instances is around 55%. Note, however, that in 53 instances knapsack cuts do nothing to improve the gap. If we consider the average GAP closed including these 53 instances, the average drops to 18.24%.

Second, consider the column MIR in which we can get an idea of how well the mixed integer rounding cut closure compares to the knapsack cut closure. Observe that of the 26 problems, in 21 of them, by using the MIR cuts alone, we close over 87% of the GAP closed by the knapsack cuts. This indicates that MIR inequalities are a very important subset of knapsack inequalities, at least for the instances considered. A natural question is the following: How much could we improve the value of MIR-PERF if we used an exact MIR separation algorithm as opposed to a heuristic? In an attempt to answer this question we fine-tuned the settings of the MIR heuristic for the problems p0033, p0548 and qnet1. In these, we managed to improve the value of MIR-PERF from 87.31% to 100%, from 62.76% to 63.66% and from 56.68% to 77.27% respectively. This indicates that the true value of MIR-REL might be much closer to 100% than suggested by the table.

4.4.3.2 Knapsack cuts derived from tableau rows

In order to obtain $\mathcal{T}$ for a given problem instance, we first solve the linear programming relaxation of the corresponding problem and let $\mathcal{T}$ be the optimal tableau rows in the augmented variable space consisting of both structural and slack variables. When attempting to generate a cut from a tableau row our algorithms all worked in this augmented space. Whenever a cut was generated from a tableau row, we would first substitute out slack variables before adding it back to the LP. Note that slack variables were always assumed to be non-negative and continuous.

We now discuss the computational results obtained for $\mathcal{K} = \mathcal{T}$. Of the 92 instances in MIPLIB 3.0 and MIPLIB2003, we were able to compute the values $z^*(\mathcal{C}^{\mathcal{T}})$ for 65 of them. Of these 65 problems, 4 of them were such that ORIG-GAP was equal to 0.0 (disctom, dsbmip, enigma, and noswot), and 9 of them were such that KNAP and MIR were both equal to 0.0 (glass, liu, msc98-ip, mzzv11, mzzv42z, net12, rd-rplusc-21, stein27 and stein45). In Table 4.4.3.1 we present the results for the remaining 52 problems. That is, the only ones in which knapsack cuts improved the bound.

First, it is important to remark that separating knapsack cuts from tableau rows

Table 5: Benchmarks for Tableau Closure

Instance	ORIG-GAP	MIR	KNAP	MIR-REL
air03	0.38%	100.00%	100.00%	100%
a1c1s1	91.33%	18.75%	18.75%	100%
aflow30a	15.10%	12.02%	12.02%	100%
aflow40b	13.90%	7.45%	7.45%	100%
bell3a	1.80%	60.15%	60.15%	100%
bell5	3.99%	14.53%	14.68%	98.94%
blend2	8.99%	20.63%	20.63%	100%
danoint	4.62%	1.74%	1.74%	100%
dcmulti	2.24%	50.46%	50.49%	99.94%
egout	73.67%	55.33%	55.33%	100%
fiber	61.55%	75.89%	77.27%	98.21%
fixnet6	69.85%	11.08%	11.08%	100%
flugpl	2.86%	11.74%	11.74%	100%
gen	0.16%	61.67%	61.97%	99.52%
gesa2	1.18%	28.13%	28.13%	99.98%
gesa2_o	1.18%	29.55%	29.65%	99.67%
gesa3	0.56%	45.76%	45.83%	99.85%
gesa3_o	0.56%	49.96%	49.99%	99.94%
gt2	36.41%	84.56%	90.02%	93.93%
khb05250	10.31%	75.14%	75.14%	100%
lseu	25.48%	61.21%	61.21%	100%
manna81	1.01%	30.08%	30.08%	100%
misc03	43.15%	7.24%	7.24%	100%
misc06	0.07%	26.98%	26.98%	100%
misc07	49.64%	0.72%	0.72%	100%
mod008	5.23%	22.57%	22.59%	99.92%
mod011	13.86%	22.17%	22.17%	100%
modglob	1.49%	18.05%	18.05%	100%
momentum2	41.32%	40.13%	40.13%	100%
nsrand-ipx	4.53%	23.34%	23.34%	100%
nw04	3.27%	22.37%	22.37%	100%
opt1217	25.13%	23.09%	23.09%	100%
p0033	18.40%	74.71%	74.71%	100%
p0201	9.72%	34.36%	34.36%	100%
p0282	31.56%	9.21%	12.48%	73.79%
p0548	96.37%	70.97%	92.74%	76.52%
p2756	13.93%	22.38%	45.42%	49.28%
pp08a	62.61%	50.97%	50.97%	100%
pp08aCUTS	25.43%	32.30%	32.54%	99.26%
protfold	35.35%	3.79%	3.79%	100%
qiu	601.15%	3.47%	3.47%	100%
rentacar	5.11%	27.42%	27.42%	100%
rgn	40.63%	9.78%	9.78%	100%
roll3000	13.91%	0.16%	0.16%	100%
set1ch	41.31%	39.18%	39.18%	100%
sp97ar	1.38%	4.23%	4.23%	100%
swath	28.44%	33.22%	33.58%	98.91%
timtab1	96.25%	23.59%	23.59%	100%
timtab2	93.49%	11.91%	11.91%	100%
tr12-30	89.12%	28.47%	28.47%	100%
vpm1	22.92%	47.27%	49.09%	96.30%
vpm2	28.08%	19.17%	19.39%	98.85%

is considerably more difficult than separating knapsack cuts from original formulation rows. This is due to several reasons: Tableau rows are typically much more dense, coefficients tend to be numerically very bad, and rows tend to have many continuous variables. This added difficulty is reflected in the fact that out of 92 instances, after several days of runs we managed to solve 65 instances to completion, as opposed to the 83 which we solved when considering formulation rows.

Second, it is interesting to note that the value KNAP is very erratic, uniformly ranging in values from 100% to 0.0%. In contrast to the case of formulation rows, only 9 instances are such that KNAP-PERF is 0.0%.

The last, and perhaps most startling, observation is that the MIR-REL is very often close to 100%. If this result were true in general, it would be very surprising. However, because there are still 27 instances which have not been solved one must be very careful. Because of the way in which we computed these numbers, it could be the case that those instances with MIR-REL close to 100% are easier for our methodology to solve. It is very reasonable to expect that instances with MIR-REL well below 100% are more difficult to solve as they require more iterations of the knapsack separation algorithm as opposed to iterations of the MIR separation heuristic.

4.5 Final remarks

One of the goals of this study has been to assess the overall effectiveness of MIR inequalities relative to knapsack cuts. The motivation being the empirical observation that though much research has been conducted studying inequalities derived from single row systems, no such class of inequalities has been able to systematically improve upon the performance of MIRs. In this regard, the results we present are surprising.

We observe that in several test problems, the bound obtained by using just MIR inequalities is very similar in value (if not equal) to the bound obtained using all possible knapsack cuts. Though it is important to note that this observation is limited in the number of test problems considered, it does help explain the lack of success in generating other cuts from tableau and formulation rows, and, suggests that for further bound improvements we might have to consider new row aggregation schemes, or cuts derived from multiple row systems.

Notice, however, that our benchmarks only take into account the bound improvement which can be obtained after several rounds of knapsack cuts. In that sense, it is still possible to obtain classes of knapsack cuts that perform better than MIR/GMI cuts even in the cases where the relative gap closed by the MIR/GMI inequalities is 100%. A more successful class of knapsack cuts would, for example, be able to obtain the same bound as MIR/GMI much faster, with a fewer number of cuts and a fewer number of rounds of cutting planes.

Another remark is that we put great care into ensuring that the generated cuts are valid and that the procedure runs correctly, but this makes the methodology very slow and also makes all the improvements presented in this chapter mandatory to perform these experiments in practice. Even with all the improvements, some of the computed KNAP values took as much as 5 days to obtain. Some of the unsolved instances have been ran for over a week without a final answer being reported. Part of the difficulty arises from the fact that rational arithmetic is being employed. In average, we have observed that performing rational arithmetic computations takes orders of magnitude longer than floating-point computations.

As a final remark, we note that, recently, knapsack cuts have been used in practice to help solve some combinatorial optimization problems [13]. Following up on this, it would be interesting to see if we can also use our knapsack cut generation procedure in practice to help solve other mixed-integer programming problems.

CHAPTER V

NUMERICALLY ACCURATE GOMORY MIXED-INTEGER CUTS

5.1 Introduction

As mentioned in Chapter 1, MIP is a very powerful and useful tool in operations research, which has evolved immensely in the past 50 years. For example, according to Bixby et al. [21], in the eight-year span between 1996 and 2004 alone, there was a speedup of almost two orders of magnitude (almost 100 times faster) in the solution of a set of benchmark instances.

This evolution has turned MIP into a practical tool with widespread use to solve many practical problems. In fact, nowadays, MIP is often used as a subroutine in complex systems to solve real-world problems and even as a subroutine inside MIP software itself.

In spite of all this progress in the solution of MIPs, one major flaw of most MIP solvers available to date is that usually there is no guarantee that they will return the correct optimal solution. This happens because most MIP solvers use floating-point arithmetic computations which have intrinsic arithmetic errors, and, therefore, several crucial decisions that guarantee the correctness of MIP algorithms are also subject to the same errors. This issue is usually addressed by using tolerances that try to mitigate the effect of these imprecise computations. For instance, if the solver finds a number in floating-point that is equal to 3.0000001, it may consider it as having the value of 3, regarding the fractional part of 0.0000001 as an error.

While this approach has been successful in making MIP solvers less prone to numerical instability and more reliable in returning "good" feasible solutions, there is still no guarantee that the solution that is returned by most MIP solvers is the true optimal solution.

One may argue that, in most applications, the data is already imprecise and therefore one is not really interested in the true optimal solution, but in a solution of “good quality”. However, there are contexts in which the true optimal solution is required. First, consider that MIPs are starting to be used as subroutines, even inside MIP software, which means errors in the solution of MIPs can start propagating and becoming more significant and troublesome. Also, there are applications like combinatorial auctions, in which obtaining the incorrect optimal solution would be highly undesirable. Finally, if MIP is made to be reliable in the sense that it will always return the correct optimal solution, it can be used to help answer some theoretical questions, similar to the use of linear programming (for example, the proof of the Kepler conjecture by Hales [68]).

Producing correct solutions to linear programming problems has been studied in several papers [7, 52, 73, 42, 75]. Doing the same for MIPs, however, is more complicated since there are many more components involved in the solution of MIPs and one would need to make sure all of them are implemented without error. One possible approach is to implement an MIP solver doing most of its operations using rational arithmetic. This has been tried by William Cook and Daniel Espinoza (unpublished), but it was not competitive with floating-point based codes since rational arithmetic is orders of magnitude slower than floating-point. Therefore, ideally, one would like to take advantage of the speed of floating-point arithmetic, without suffering from its numerical inaccuracies.

In this chapter, we address this issue under the point of view of one of the most important components in MIP solvers [21], namely cutting-planes, more specifically Gomory Mixed-Integer (GMI) cuts. These are the most effective cutting planes used in solving MIPs, but their validity is purely based on the correctness of algebraic calculations and, therefore, if generated using floating-point arithmetic, they may not be valid. Indeed, Margot [79] presents computational experiments that show that GMI cuts can be invalid even in small binary instances.

We call a cutting-plane generation process *safe* in a specified model of arithmetic if it is guaranteed to produce only inequalities that are valid for all feasible solutions to a given MIP problem. We present a simple and easy-to-implement way to generate safe GMI cuts in floating-point arithmetic. As discussed above, this gives us the advantage of being fast without having to compromise on the correctness of the cuts. Another benefit of this procedure is that MIP solvers tend not to add multiple rounds of Gomory cuts, for numerical stability and to avoid the propagation of error when generating possibly invalid cuts based on already invalid ones. This measure, while necessary if no guarantee of validity is present, reduces the potential gain that one can have in reducing the duality gap by adding those cuts to the LP relaxation. Indeed, our experiments show that this gain can be significant.

It is worth mentioning that Neumaier and Shcherbina [87] have proposed a way to generate safe cuts, using a combination of interval arithmetic and directed rounding. However, since interval arithmetic is not readily available in most computer languages, their approach is not straightforward to implement. In contrast, our approach is based on standard IEEE floating-point operations [71] and can be easily implemented in any programming language that complies with those standards.

In Section 5.2 we define the MIR procedure for generating a cut from a mixed-integer knapsack set. General properties of floating-point arithmetic are discussed in Section 5.3, together with simple procedures for safe row aggregation and safe substitution of slack variables. In Section 5.4 we present the safe MIR procedure and in Section 5.5 we extend this to a safe procedure for the more general complemented-MIR cuts. In addition to our discussion of cutting planes, we also present the Neumaier-Shcherbina method for obtaining valid dual bounds for bounded LP problems in

Section 5.6. We present this method to illustrate how the rounding operations can be used to help in making other components of the solution of MIPs error-free. In Section 5.7 we present the computational experiments we performed using the safe MIR cuts. Final remarks are presented in Section 5.8

5.2 MIR inequalities

As mentioned in the previous section, this chapter is devoted to generating safe GMI cuts. However, since GMI cuts are simply an application of the mixed-integer rounding (MIR) procedure to the tableau rows, we will make all our developments for MIR cuts and they will be immediately applicable to GMI cuts.

Let n be a positive integer and let N denote the set $\{1, \ldots, n\}$. Consider $a \in \mathbb{Q}^n$, $b \in \mathbb{Q}$, and a partition (I, C) of N. A single-row relaxation of an MIP model can take the form

$$K = \{x \in \mathbb{R}^n : \sum_{j=1}^{n} a_j x_j \geq b,\ x \geq 0,\ x_j \in \mathbb{Z}\ \forall j \in I\}. \tag{51}$$

The defining inequality of K can, for example, be one of the original MIP constraints or, more generally, a nonnegative linear combination of the original constraints.

Let $S = \{j \in I : \hat{a}_j \leq \hat{b}\}$. The *MIR inequality*

$$\sum_{j \in S} (\hat{a}_j + \hat{b} \lfloor a_j \rfloor) x_j + \sum_{j \in I \setminus S} (\hat{b} + \hat{b} \lfloor a_j \rfloor) x_j + \sum_{j \in C} \max\{a_j, 0\} x_j \geq \hat{b} \lceil b \rceil \tag{52}$$

is valid for K and can therefore be considered for use as a cutting plane to improve the LP relaxation of the original MIP model.

Defining $f(q_1, q_2) := \min\{\hat{q_1}, \hat{q_2}\} + \hat{q_2} \lfloor q_1 \rfloor$ and $h(q) := \max\{q, 0\}$, the MIR inequality (52) can be written as

$$\sum_{i \in I} f(a_i, b) x_i + \sum_{i \in C} h(a_i) x_i \geq \hat{b} \lceil b \rceil. \tag{53}$$

This compact notation will be convenient in our discussion of accurate versions of the inequality.

For each $j \in C$, let $l_j \in \mathbb{Q}$ and $u_j \in \mathbb{Q} \cup \{+\infty\}$. For each $j \in I$, let $l_j \in \mathbb{Z}$ and $u_j \in \mathbb{Z} \cup \{+\infty\}$. Assume $0 \leq l_j \leq u_j$ for each $j \in N$ and consider the set

$$K_B = \{x \in \mathbb{R}^n : \sum_{i \in N} a_j x_j \geq b,\ l_j \leq x_j \leq u_j\ \forall j \in N \text{ and } x_j \in \mathbb{Z}\ \forall j \in I\}. \tag{54}$$

The lower and upper bounds on the variables lead to a slightly more general form of the MIR inequality, called the *complemented-MIR* (c-MIR) inequalities [78]. The name derives from the fact that the c-MIR inequalities are obtained by complementing variables.

Let U, L be disjoint subsets of N such that every $j \in U$ satisfies $u_j < \infty$. By substituting variables x_j with $j \in U$ by $u_j - x_j$, and variables x_j with $j \in L$ by $x_j - l_j$, applying the MIR procedure, and substituting back, we obtain the following c-MIR inequality for K_B:

$$-\sum_{U \cap I} f(-a_j, r) x_j + \sum_{I \setminus U} f(a_j, r) x_j - \sum_{U \cap C} h(-a_j) x_j + \sum_{C \setminus U} h(a_j) x_j \geq R, \tag{55}$$

where

$$r = b - \sum_{j \in U} a_j u_j - \sum_{j \in L} a_j l_j$$

and

$$R = \hat{r} \lceil r \rceil - \sum_{U \cap I} f(-a_j, r) u_j + \sum_{L \cap I} f(a_j, r) l_j - \sum_{U \cap C} h(-a_j) u_j + \sum_{L \cap C} h(a_j) l_j.$$

Marchand and Wolsey [78] have demonstrated the effectiveness of c-MIR cuts in practical computations with MIP test instances.

If the defining inequality of K_B is taken as a row of an optimal simplex tableau for the LP relaxation of an MIP instance, then the c-MIR cut is a form of the GMI cut.

5.3 Floating-point arithmetic

The validity of MIR and c-MIR inequalities, for the sets K and K_B, relies on the correctness of arithmetic calculations, which cannot be guaranteed when floating-point arithmetic is employed. Moreover, if the defining inequality for K or K_B is obtained by aggregating several of the original constraints from an MIP instance, then caution must be taken to ensure that the set itself is a valid relaxation. To discuss an approach for dealing with these issues, we give a brief description of the floating-point-arithmetic platform.

5.3.1 The model of floating-point arithmetic

A floating-point number consists of a sign, exponent, and a significand. Standard IEEE double precision floating-point arithmetic allots 11 bits for the exponent and 52 bits for the significand [53]. In our discussion we assume only that the number of bits assigned to the exponent and significand are fixed to some known values.

With the above assumption, let $\mathbb{M} \subseteq \mathbb{Q}$ be the set of floating-point representable reals. Note that $\mathbb{M}$ is finite. If $q \in \mathbb{M}$, then both $-q$ and $\hat{q}$ are members of M. Also, if $a, b \in \mathbb{M}$, then $\min\{a, b\}$ and $\max\{a, b\}$ are members of M.

The set $\mathbb{M}$ does not have much structure; it is not a monoid since associativity for addition does not hold.

5.3.2 Approximating real functions

Given a function $f : \mathbb{R}^n \to \mathbb{R}$, we say that a function $f^{up} : \mathbb{M}^n \to \mathbb{M}$ upper approximates f in $\mathbb{M}$ if

$$f^{up}(x) \geq f(x) \ \text{ if } x \in \mathbb{M}^n$$

and we say that a function $f^{dn} : \mathbb{M}^n \to \mathbb{M}$ lower approximates f in $\mathbb{M}$ if

$$f^{dn}(x) \leq f(x) \ \text{ if } x \in \mathbb{M}^n.$$

Note that the basic arithmetic operations $+$ and $*$ are functions from $\mathbb{R}^2 \to \mathbb{R}$. Let $+^{up} : \mathbb{M}^2 \to \mathbb{M}$ and $+^{dn} : \mathbb{M}^2 \to \mathbb{M}$ be upper and lower approximations of $+$, and let $*^{up} : \mathbb{M}^2 \to \mathbb{M}$ and $*^{dn} : \mathbb{M}^2 \to \mathbb{M}$ be upper and lower approximations of $*$. (In the C programming language [72], IEEE floating-point rounding conventions can be set to produce upper and lower approximations using the `fesetround` function.

Upper approximations are obtained by first calling the function with the argument `FE_UPWARD` and then carrying out the usual arithmetic operation. Similarly, lower approximations are obtained using the `FE_DOWNWARD` argument.)

To simplify notation, we write

$$\overline{a+b} := +^{up}(a,b) \quad \text{and} \quad \underline{a+b} := +^{dn}(a,b)$$

for addition and similarly for subtraction and multiplication.

Let $n \geq 3$ be an integer and let $a_i \in \mathbb{M}$, $\pi_i \in \mathbb{M}$ for $i = 1, \ldots, n$. We define upper and lower approximations for $\sum_{i=1}^{n} \pi_i a_i$ by

$$\overline{\sum_{i=1}^{n} \pi_i a_i} := \overline{\left(\overline{\sum_{i=1}^{n-1} \pi_i a_i} \right) + \overline{\pi_n a_n}}$$

and

$$\underline{\sum_{i=1}^{n} \pi_i a_i} := \underline{\left(\underline{\sum_{i=1}^{n-1} \pi_i a_i} \right) + \underline{\pi_n a_n}}$$

respectively. It is important to notice that any ordering of $a_1, \ldots, a_n$ will yield an upper/lower approximation of $\sum_{i=1}^{n} \pi_i a_i$, but different orders may yield different results. Thus, the operation is not commutative.

5.3.3 Safe row aggregation

Consider a polyhedron

$$P = \{x \in \mathbb{R}^n : Ax \geq d, l \leq x \leq u\}$$

with $A \in \mathbb{M}^{m \times n}$, $d \in \mathbb{M}^m$, $u \in \{\mathbb{M} \cup \{+\infty\}\}^n$, $l \in \mathbb{M}^n$, and $0 \leq l \leq u$. Given a set of multipliers $\lambda \in \mathbb{M}^m$, with $\lambda \geq 0$, the inequality $\lambda^T Ax \geq \lambda^T d$ is satisfied by all $x \in P$, but this may not be true in the presence of rounding errors. Nonetheless, since $x \geq 0$, we have that

$$\sum_{j=1}^{n} \overline{\left(\sum_{i=1}^{m} \lambda_i a_{ij} \right)} x_j \geq \underline{\sum_{i=1}^{m} \lambda_i d_i}$$

is always a valid inequality. This process permits the safe aggregation of systems of inequalities.

The same concept can be applied to equality systems $Ax = d$, but in this case the multipliers λ need not be restricted to nonnegative values.

5.3.4 Safe substitution of slack variables

Suppose $P \subseteq \mathbb{R}^{n+1}$ is a polyhedron such that every point $(x, s) \in P$ satisfies the equation

$$\sum_{j=1}^{n} v_j x_j + \beta s = v_o \tag{56}$$

with $\beta \in \{-1, +1\}$. Consider an inequality

$$\sum_{j=1}^{n} \pi_j x_j + \rho s \geq \pi_o \tag{57}$$

that is valid for P. We wish to eliminate s from (57), while maintaining its validity.

Note that adding $-\rho\beta$ times (56) to (57) gives the inequality

$$\sum_{j=1}^{n} (\pi_j - \rho\beta v_j) x_j + (\rho - \rho)s \geq \pi_o - \rho\beta v_o.$$

Assuming all coefficients in (56) and (57) are in $\mathbb{M}$, this suggests that we apply safe row aggregation using $\lambda = -\rho\beta \in \mathbb{M}$ for (56) and $\lambda = 1 \in \mathbb{M}$ for (57). The coefficient of the s variable in the resulting inequality is

$$\overline{\overline{1\rho} + \overline{(-\rho\beta)\beta}} = \overline{\rho + \overline{-\rho\beta^2}} = \overline{\rho + \overline{-\rho}} = \overline{\rho - \rho} = 0$$

and thus we have safely removed s from (57).

5.4 Safe MIR inequalities

For the single-row relaxation (51) in Section 5.2, we described the MIR inequality

$$\sum_{i \in I} f(a_i, b) x_i + \sum_{i \in C} h(a_i) x_i \geq \hat{b} \lceil b \rceil$$

where $f(q_1, q_2) := \min\{\hat{q_1}, \hat{q_2}\} + \hat{q_2}\lfloor q_1 \rfloor$ and $h(q) := \max\{q, 0\}$. To obtain a safe version of this cutting plane in floating-point arithmetic, define the upper approximation $f^{up} : \mathbb{M} \times \mathbb{M} \to \mathbb{M}$ of f as

$$f^{up}(q_1, q_2) := \overline{\left(\min\{\hat{q_1}, \hat{q_2}\} + \overline{\hat{q_2}\lfloor q_1 \rfloor}\right)}$$

and define $h^{up}(q) := h(q)$ (no rounding is needed).

Since $x \geq 0$, we have

$$\sum_{i \in I} f^{up}(a_i, b)x_i + \sum_{i \in C} h^{up}(a_i)x_i \geq \sum_{i \in I} f(a_i, b)x_i + \sum_{i \in C} h(a_i)x_i \geq \hat{b}\lceil b \rceil \geq \underline{\hat{b}\lceil b \rceil}.$$

Therefore

$$\sum_{i \in I} f^{up}(a_i, b)x_i + \sum_{i \in C} h^{up}(a_i)x_i \geq \underline{\hat{b}\lceil b \rceil} \tag{58}$$

is a safe MIR inequality.

5.5 Safe c-MIR inequalities

Consider again the set

$$K_B = \{x \in \mathbb{R}^n : \sum_{i \in N} a_j x_j \geq b,\ l_j \leq x_j \leq u_j\ \forall j \in N \text{ and } x_j \in \mathbb{Z}\ \forall j \in I\}$$

defined in Section 5.2.

Let sets U, S, L form a partition of $N = \{1, \ldots, n\}$, such that every $j \in U$ satisfies $u_j < \infty$ and every $j \in S$ satisfies $l_j \geq 0$. For each $j \in U$, define $x_j^u = u_j - x_j$. Also, define $x_j^l = x_j - l_j$ for each $j \in L$ and $x_j^l = x_j$ for each $j \in S$. Observe that x_j^u and x_j^l are both bounded below by zero. Furthermore, if $j \in I$ and x_j takes an integer value, so will x_j^u or x_j^l, depending on if $j \in U$ or $j \in L \cup S$.

By substituting variables x_j with $j \in U$ by $u_j - x_j^u$, variables x_j with $j \in L$ by $x_j^l + l_j$ and denoting $x_j^l = x_j$ for $j \in S$, we obtain the inequality

$$\sum_{j \in U}(-a_j)x_j^u + \sum_{j \notin U} a_j x_j^l \geq b - \sum_{j \in U} a_j u_j - \sum_{j \in L} a_j l_j. \tag{59}$$

Now note that regardless of the sign of a_j, u_j and l_j, we have $\overline{a_j u_j} \geq a_j u_j$ and $\overline{a_j l_j} \geq a_j l_j$. It follows that

$$\overline{\sum_{j\in U} a_j u_j} \geq \sum_{j\in U} \overline{a_j u_j} \geq \sum_{j\in U} a_j u_j$$

and

$$\overline{\sum_{j\in L} a_j l_j} \geq \sum_{j\in L} \overline{a_j l_j} \geq \sum_{j\in L} a_j l_j$$

and thus we have the valid inequality

$$\sum_{j\in U}(-a_j)x_j^u + \sum_{j\notin U} a_j x_j^l \geq \underline{b - \overline{\sum_{j\in U} a_j u_j} - \overline{\sum_{j\in L} a_j l_j}}. \tag{60}$$

Observe that $x_j^u, x_j^l \geq 0$ for all $j \in N$, hence we can apply the MIR procedure to obtain a valid inequality for (60) subject to the corresponding nonnegativity constraints.

Let $r = \underline{b - \overline{\sum_{j\in U} a_j u_j} - \overline{\sum_{j\in L} a_j l_j}} \in \mathbb{M}$. If we apply (58) we obtain

$$\sum_{U\cap I} f^{up}(-a_j, r)x_j^u + \sum_{I\setminus U} f^{up}(a_j, r)x_j^l + \sum_{U\cap C} h^{up}(-a_j)x_j^u + \sum_{C\setminus U} h^{up}(a_j)x_j^l \geq \underline{\hat{r}\lceil r\rceil}.$$

If we substitute back the variables we get the following valid inequality in terms of our original variables:

$$\begin{aligned}\sum_{S\cap I} f^{up}(a_j, r)x_j + \sum_{U\cap I} f^{up}(-a_j, r)(u_j - x_j) + \sum_{L\cap I} f^{up}(a_j, r)(x_j - l_j)\\ + \sum_{S\cap C} h^{up}(a_j)x_j + \sum_{U\cap C} h^{up}(-a_j)(u_j - x_j) + \sum_{L\cap C} h^{up}(a_j)(x_j - l_j) \geq \underline{\hat{r}\lceil r\rceil}.\end{aligned}$$

Rearranging the terms we obtain

$$\begin{aligned}&\sum_{U\cap I} f^{up}(-a_j, r) * (-1)x_j + \sum_{I\setminus U} f^{up}(a_j, r)x_j\\ &\qquad + \sum_{U\cap C} h^{up}(-a_j) * (-1)x_j + \sum_{C\setminus U} h^{up}(a_j)x_j \geq \underline{\hat{r}\lceil r\rceil} +\\ &\qquad\quad \sum_{U\cap I} f^{up}(-a_j, r) * (-1)u_j + \sum_{L\cap I} f^{up}(a_j, r)l_j\\ &\qquad\qquad + \sum_{U\cap C} h^{up}(-a_j) * (-1)u_j + \sum_{L\cap C} h^{up}(a_j)l_j.\end{aligned}$$

Now regardless of the signs of u_j, l_j, we have

$$\underline{\hat{r}\lceil r\rceil} + \sum_{U\cap I} f^{up}(-a_j, r) * (-1)u_j + \sum_{L\cap I} f^{up}(a_j, r)l_j + \sum_{U\cap C} h^{up}(-a_j) * (-1)u_j + \sum_{L\cap C} h^{up}(a_j)l_j \geq R$$

where

$$R = \underline{\hat{r}\lceil r\rceil} + \underline{\sum_{U\cap I}(f^{up}(-a_j, r) * (-1))u_j} + \underline{\sum_{L\cap I} f^{up}(a_j, r)l_j} + \underline{\sum_{U\cap C}(h^{up}(-a_j) * (-1))u_j} + \underline{\sum_{L\cap C} h^{up}(a_j)l_j}.$$

Thus

$$\sum_{U\cap I} f^{up}(-a_j, r) * (-1)x_j + \sum_{I\setminus U} f^{up}(a_j, r)x_j + \sum_{U\cap C} h^{up}(-a_j) * (-1)x_j + \sum_{C\setminus U} h^{up}(a_j)x_j \geq \underline{R}$$

is a valid inequality for K_B with all coefficients in $\mathbb{M}$.

5.6 Safe dual bounds

In this section we describe the technique of Neumaier and Shcherbina [87] for computing valid dual LP bounds using floating-point arithmetic.

Consider the bounded LP

$$\min\{c^T x : Ax \geq b,\ 0 \leq l \leq x \leq u\} \tag{61}$$

and its dual

$$\max\{b^T y + l^T w - u^T z : A^T y + w - z = c,\ y, w, z \geq 0\} \tag{62}$$

where we assume all components of u are finite.

If (61) is a relaxation of an MIP instance, then an optimal solution provides a bound on the MIP objective value. The problem we face is that because of intrinsic errors in floating-point arithmetic, obtaining an optimal solution to (61) may be difficult. Note, however, that any feasible solution to the dual (62) will also give a bound via the LP weak duality theorem.

Consider any assignment of (potentially infeasible) values y^* to the dual variables y. If $A^T y^* = c$, then $b^T y^*$ is a lower bound for (61). However, ensuring that $A^T y^* = c$ may itself be difficult in floating-point arithmetic. We get around this problem by taking advantage of the form of (62).

For $j \in \{1, \ldots, n\}$, the jth constraint of (62) is $\sum_{i=1}^{m} y_i a_{ij} + w_j - z_j = c_j$. Given y^*, we have $w_j - z_j = c_j - \sum_{i=1}^{m} y_i^* a_{ij}$. Set $z_j^* = \max\{0, \overline{\sum_{i=1}^{m} y_i^* a_{ij} - c_j}\}$. Clearly, $z_j^* \in \mathbb{M}$ and

$$z_j^* \geq \overline{\sum_{i=1}^{m} y_i^* a_{ij} - c_j} \geq \sum_{i=1}^{m} y_i^* a_{ij} - c_j.$$

Thus, we can set $w_j^* = z_j^* - (\sum_{i=1}^{m} y_i^* a_{ij} - c_j) \geq 0$.

It is difficult to guarantee that $w_j^* \in \mathbb{M}$, but since $l \geq 0$ we have

$$\sum_{i=1}^{m} b_i y_i^* + \sum_{j=1}^{n} l_j w_j^* - \sum_{j=1}^{n} u_j z_j^* \geq \sum_{i=1}^{m} b_i y_i^* + \sum_{j=1}^{n} (-u_j) z_j^* \geq \underline{\sum_{i=1}^{m} b_i y_i^* + \sum_{j=1}^{n} (-u_j) z_j^*}$$

To obtain a good lower bound, the values y^* can be taken as the optimal dual solution returned by a floating-point LP solver. Such values y^* are likely to be only slightly infeasible. Note that the bound we obtain via the safe procedure may be less than the floating-point solver's claimed objective value, due to possible increases in z^* and to dropping the w^* variables.

This approach is similar to the bounding technique adopted in the TSP code of Applegate et al. [6]. In their work, the dual values y^* are truncated to a fixed precision, appropriate fixed-precision values are assigned to w and z, and the dual bound is computed in the same fixed precision with overflow checking enabled. An advantage of the method described above is that the bound can be computed entirely in floating-point arithmetic.

5.7 *Computational study*

In our study we consider two scenarios where safe MIR procedures can potentially be useful. The first set of tests is concerned with the use of safe cuts in cases that demand accurate bounds. We use the TSP as a case study, applying multiple rounds of safe Gomory cuts to improve LP relaxations generated by TSP-specific methods.

The second tests consider the repeated application of Gomory cuts for general MIP instances. Generating Gomory cuts based on previous cuts can quickly lead to inaccurate results with standard MIP software. Our tests aim to give an indication of the possible improvements in LP bounds that can be obtained by adopting the safe methods, where multiple rounds of cuts can be added without loss of accuracy.

Note that as an alternative to our safe MIR approach, the computations could be carried out entirely in exact rational arithmetic, using the LP solver of Applegate et al. [7] and rational versions of the MIR procedure. On large instances such an approach is considerably more time-consuming, however, due to the complexity of rational computations on the difficult LP instances that are created. Moreover, the approach adopted here demonstrates the feasibility of including safe MIR methods in standard floating-point-based software.

5.7.1 Selection of Gomory cuts

Given an MIP instance, our cut-generation process begins by adding a slack variable to each row of the model to obtain equality constraints. If a row is such that all of the participating variables are integer, all of the left-hand-side coefficients are integer, and the right-hand-side is integer, then the corresponding slack variable is defined to be integer.

The resulting LP relaxation is solved with the simplex algorithm, producing a basic optimal solution x^*. We improve the relaxation by adding rounds of Gomory cuts, where each round consists of the following steps.

1. *Variable complementation.* Define the index sets $U = \{j : x_j^* = u_j, 1 \leq j \leq n\}$ and $L = \{j : x_j^* = l_j, 1 \leq j \leq n\}$.

2. *Rank the fractional variables.* The integer variables x_j that take on a fractional value x_j^* in the current basis are ranked in non-decreasing value of $|x_j^* - 0.5|$.

3. *Row selection.* Select the 500 highest-ranking variables and safely compute the corresponding tableau rows.

4. *Compute the cuts.* Process the selected rows in the ranked order. For each row safely compute the c-MIR inequality using the sets U and L as defined above, after scaling by $K = 1$, $K = 2$, and $K = 3$, as proposed in Cornuéjols et al. [31]. Stop this procedure if 500 (sufficiently) violated cuts are generated before all selected variables are processed.

5. *Add the cuts.* Add the computed cuts to the LP and resolve.

6. *Remove cuts.* Remove from the LP all previously added cuts that are no longer violated.

This is a straightforward implementation of Gomory cuts. For a discussion of general cut selection see Achterberg [1] and Goycoolea [62].

5.7.2 TSPLIB results

Early cutting-plane research on the TSP by Martin [80], Miliotis [83], and others adopted general-purpose MIP codes for improving LP relaxations. In later work, these methods were replaced by algorithms for generating TSP-specific cutting planes. With the safe MIR procedures it is possible to combine these ideas, employing general cuts to further improve relaxations obtained by problem-specific methods.

Table 6: MIP Relaxations of the TSP

Name	Variables	TSP Optimal	LP Value	+Cuts	Gap Closed
pcb3038	6976	137694	137684.25	137684.52	2.51%
fnl4461	10129	182566	182558.55	182559.97	19.08%
rl5915	24939	565530	565484.03	565487.91	8.44%
rl5934	25285	556045	555994.47	556001.02	12.97%
pla7397	27209	23260728	23258946.65	23259208.71	14.71%
rl11849	70259	923288	923208.71	923210.01	1.53%
usa13509	129837	19982859	19981199.08	19981229.43	1.83%
brd14051	83221	469385	469353.80	469353.91	0.35%
d15112	110072	1573084	1572966.32	1572967.08	0.64%
d18512	141025	645238	645194.86	645195.17	0.71%
pla33810	67683 (+)	660048945	66001233.03	66001901.19	1.40%
pla85900	167870 (+)	142382641	142296659.63	142299299.63	3.07%

We consider MIP relaxations obtained by long runs of the Concorde TSP solver (Applegate et al. [6]). In Table 5.7.2 we give statistics for the relaxations for all TSPLIB instances having at least 3,000 cities, with the exception of fl3795. In the case of fl3795, the Concorde LP bound establishes the optimality of the tour. Except for pla33810 and pla85900, the set of variables in each case is obtained by reduced-cost fixing based on the Concorde relaxation (the variables set to 0 are removed from the LP), thus any LP bound obtained with general MIP cuts is a valid bound for the original TSP. In the cases of the two largest instances, pla33810 and pla85900, only a subset of the variables is given due to the large number that remain after reduced-cost fixing.

The relaxations were found with `concorde -mC48 -Z3`, the strongest recommended setting of the Concorde separation routines. This setting uses repeated local cuts, up to size 48, as well as the domino-parity inequalities (see Applegate et al. [6]).

The optimal value of the Concorde LP for each instance is reported in the "LP Value" column of Table 5.7.2. In the "+Cuts" column we report the improved lower bounds obtained by multiple rounds of safe Gomory cuts, after applying the safe-LP bounding technique from Section 5.6. The improvements range from 0.35% of the

Table 7: Valid MIP Relaxation for pla85900

Name	Variables	TSP Optimal	LP Value	+Cuts	Gap Closed
pla85900f	300969	142382641	142381453.65	142381460.98	0.62%

Concorde LP optimality gap for brd14051, up to 19.09% of the gap for fnl4461. It is interesting, however, that the very strong LP relaxations obtained by Concorde could be improved with general-purpose MIP cuts, suggesting that this may be a technique worth considering for problem classes that have not received the intense study given to the TSP.

To further illustrate the use of of the safe MIR procedure, we constructed a valid MIP relaxation for the 85900-city TSP instance pla85900, using the full set of variables that were not eliminated by reduced-cost fixing. This much stronger relaxation was obtained from the very long computation described in Applegate at al. [6]. The result after adding Gomory cuts to this instance is reported in Table 5.7.2, showing a 0.62% decrease in the gap. Although this is a modest improvement, it is possible that the LP solution is sufficiently altered to permit the further use of Concorde's TSP-specific cutting planes in an iterative fashion. (We have not tested this idea.)

5.7.3 MIPLIB results

To test the effectiveness of repeated rounds of Gomory cuts in general, we constructed a test set consisting of the union of the following instances.

- The full set of 65 instances in the MIPLIB 3.0 collection [19].
- The 27 instances in the MIPLIB 2003 collection [2] that are not included in MIPLIB 3.0.
- The 13 TSPLIB-MIP instances described in Section 5.7.2

We excluded four of these instances due to the presence of variables that may assume negative values (that are not handled in our current implementation), leaving a total of 101 instances.

To illustrate the impact of multiple rounds of safe cutting planes, we recorded the lower bounds obtained after 1, 2, 4, 8, 16, and 128 rounds of our cutting-plane procedure; if the optimal LP basis does not change after a round of cutting planes, then the particular run is terminated. In these tests, three of the instances were excluded since they each have zero integrality gap, leaving a total of 98 examples. The results are displayed as curves in Figure 9. The points (x, y) in each curve represent how many instances y closed at least x fraction of the starting integrality gap, after adding the indicated number of rounds of cuts. For example, after one

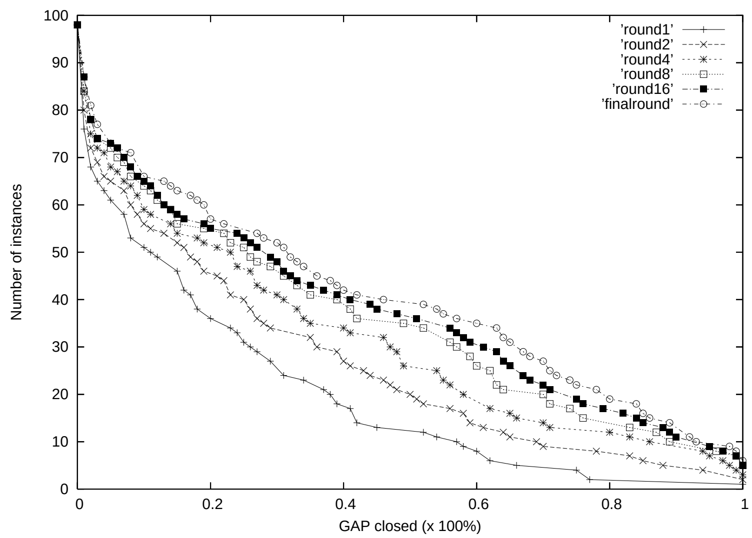

Figure 9: GAP closed after multiple rounds of safe Gomory cuts.

round of cuts 50% of the gap was closed in 12 instances, whereas four rounds of cuts closed 50% of the gap in 26 instances.

A similar chart is given in Figure 10, comparing the final bounds obtained with 128 rounds of safe cuts versus 128 rounds of the standard unsafe version of Gomory cuts. The two curves indicate that no significant loss in the bounds is incurred with the use of the accurate cutting planes.

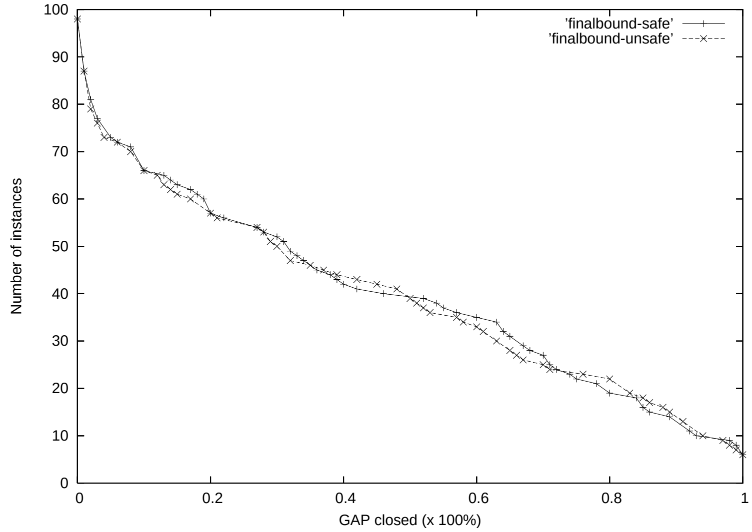

Figure 10: GAP closed by adding at most 128 rounds of Gomory cuts (safe vs. unsafe).

5.7.4 Running time

We now analyze the impact on running time by using safe GMI cuts. Figure 11 shows the performance profile comparing the average time it takes to generate the GMI cuts per round. It serves the purpose of trying to assess the impact that changing the rounding mode frequently has in the running time. By analyzing the performance profile, it can be seen that the average time per round spent generating cuts is worse in the safe version of the MIR. This result is expected, since the operation of changing the rounding mode back and forth has a negative effect on the running time. However, note that in only one instance the safe procedure takes more than twice the amount of time that the unsafe does. Moreover, a relevant information that is not present in the performance profiles is that the average time to generate cuts per round in that instance is 0.12 seconds. Therefore, the loss in time, even though is noticeable, is not large and should be relatively small compared to total running times to solve MIPs.

Notice that this comparison of times takes into account merely the generation of cuts. However, the cuts that are added can have an influence in the total running

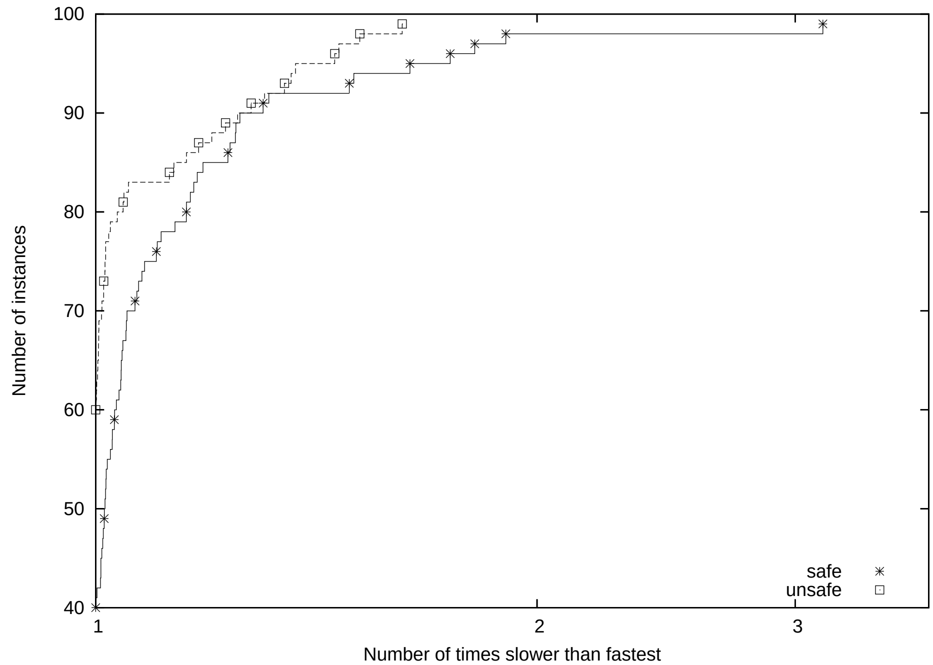

Figure 11: Performance profile of average time (per round) to generate cuts.

time even if the time to generate cuts is almost identical. As an example of how this situation may occur, consider that if we add different cuts, the LP relaxation may become harder to solve using, for example, safe cuts due to density of cuts or maybe due to numerical difficulties. However, we chose not to compare these times since these may depend on several other factors, including the LP solver used. Moreover, those results would not take into account, for instance, the fact that if we are using unsafe cuts then we can use tolerances to make the cut sparser, making the LP easier to solve. These are issues that one needs to consider carefully when implementing safe cuts.

5.8 Final remarks

The results obtained by using safe GMI cuts are encouraging: with a small loss in terms of running time, we were able to generate several rounds of safe cuts, which in turn allowed us to significantly close the duality gap in comparison with adding only one round of GMI cuts.

It would be interesting to derive ways to use the techniques presented in this chapter to generate other classes of safe cuts that are used in practice.

In addition, there is still much work to be done in order to obtain a numerically accurate and fast MIP solver. Generating numerically accurate GMI cuts is only the first effort in that direction. For example, it would be of great interest if one could generate numerically accurate dual bounds using floating-point arithmetic even for instances with unbounded variables.

REFERENCES

[1] Achterberg, T., *Constraint Integer Programming*. PhD thesis, Technischen Universtität Berlin, Berlin, Germany, 2007.

[2] Achterberg, T., Koch, T., and Martin, A., "MIPLIB 2003," *Operations Research Letters*, vol. 34, no. 4, pp. 1–12, 2006. See `http://miplib.zib.de`.

[3] Andersen, K., Cornuejols, G., and Li, Y., "Reduce-and-split cuts: Improving the performance of mixed-integer Gomory cuts," *Management Science*, vol. 51, no. 11, pp. 1720–1732, 2005.

[4] Anstreicher, K., "A monotonic projective algorithm for fractional linear programming," *Algorithmica*, vol. 1, no. 4, pp. 483–498, 1986.

[5] Applegate, D., Bixby, R. E., Chvátal, V., and Cook, W., "TSP cuts which do not conform to the template paradigm," in *Computational Combinatorial Optimization, Optimal or Provably Near-Optimal Solutions* (Jünger, M. and Naddef, D., eds.), (London, UK), pp. 261–304, Springer-Verlag GmbH, 2001.

[6] Applegate, D., Bixby, R., Chvátal, V., and Cook, W. J., *The Traveling Salesman Problem: A Computational Study*. Princeton Series in Applied Mathematics, Princeton University Press, 2006.

[7] Applegate, D., Cook, W., Dash, S., and Espinoza, D., "Exact solutions to linear programming problems," *Operations Research Letters*, vol. 35, pp. 693–699, 2007.

[8] Araoz, J., *Polyhedral Neopolarities*. PhD thesis, University of Waterloo, Department of Computer Sciences, 1974.

[9] Araoz, J., Evans, L., Gomory, R. E., and Johnson, E. L., "Cyclic group and knapsack facets," *Mathematical Programming*, vol. 96, no. 2, pp. 377–408, 2003.

[10] Atamtürk, A., "On the facets of the mixed–integer knapsack polyhedron," *Mathematical Programming*, vol. 98, pp. 145–175, 2003.

[11] Atamtürk, A., "Sequence independent lifting for mixed–integer programming," *Operations Research*, vol. 52, pp. 487–490, 2004.

[12] Atamtürk, A. and Günlük, O., "Mingling: Mixed-integer rounding with bounds," *Technical report available at Optimization Online*, November 2007.

[13] Avella, P., Boccia, M., and Vasilyev, I., "A computational study of exact knapsack separation for the generalized assignment problem," *Technical report available at Optimization Online*, July 2006.

[14] Bajalinov, E. B., *Linear-Fractional Programming: Theory, Methods, Applications and Software.* Applied Optimization, Kluwer Academic Publishers, 2003.

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

[16] Balas, E., Ceria, S., Cornuejols, G., and Natraj, N., "Gomory cuts revisited," *Operations Research Letters*, vol. 19, pp. 1–9, 1996.

[17] Balas, E. and Saxena, A., "Optimizing over the split closure," *Mathematical Programming*, vol. To appear.

[18] Balas, E. and Zemel, E., "Facets of knapsack polytope from minimal covers," *SIAM Journal on Applied Mathematics*, vol. 34, no. 1, pp. 119–148, 1978.

[19] Bixby, R. E., Ceria, S., McZeal, C. M., and Savelsbergh, M. W. P., "An updated mixed integer programming library: MIPLIB 3.0," *Optima*, pp. 12–15, June 1998.

[20] Bixby, R. E., Fenelon, M., Gu, Z., Rothberg, E., and Wunderling, R., "MIP: Theory and practice - closing the gap," in *Proceedings of the 19th IFIP TC7 Conference on System Modelling and Optimization* (Powell, M. J. D. and Scholtes, S., eds.), (Deventer, The Netherlands), pp. 19–50, Kluwer, B.V., 2000.

[21] Bixby, R., Gu, Z., Rothberg, E., and Wunderling, R., "Mixed integer programming: a progress report," in *The sharpest cut: the impact of Manfred Padberg and his work. MPS/SIAM Series on Optimization* (Grötschel, M., ed.), pp. 309–326, 2004.

[22] Bonami, P., Cornuéjols, G., Dash, S., Fischetti, M., and Lodi, A., "Projected Chvátal Gomory cuts for mixed integer linear programs," *Mathematical Programming*, vol. 113, pp. 241–257, 2008.

[23] Boyd, A. E., "Fenchel cutting planes for integer programs," *Operations Research*, vol. 42, pp. 53–64, 1992.

[24] Caprara, A. and Fischetti, F., "0-1/2 Chvátal-Gomory cuts," *Mathematical Programming*, vol. 74, pp. 221–236, 1996.

[25] Caprara, A., Fischetti, M., and Letchford, A. N., "On the separation of maximally violated mod- k cuts," *Mathematical Programming*, vol. 87, no. 1, pp. 37–56, 2000.

[26] Charnes, A. and Cooper, W., "Programming with linear fractional functions," *Naval Research Logistics Quarterly*, vol. 9, pp. 181–186, 1962.

[27] Christof, T. and Löbel, A., "PORTA - POlyhedron Representation Transformation Algorithm." Available on-line at `http://www.zib.de/Optimization/Software/Porta`, 2002.

[28] COIN-OR, "Computation infrastructure for operations research." Available on-line at http://www.coin-or.org.

[29] Cook, W., Kannan, R., and Schrijver, A., "Chvátal closures for mixed integer programming problems," *Mathematical Programming*, vol. 47, pp. 155–174, 1990.

[30] Cornuéjols, G., "Revival of the Gomory cuts in the 1990's," *Annals of Operations Research*, vol. 149, no. 1, pp. 63–66, 2007.

[31] Cornuéjols, G., Li, Y., and Vanderbussche, D., "K-cuts: A variation of Gomory mixed integer cuts from the LP tableau," *INFORMS Journal on Computing*, vol. 15, no. 4, pp. 385–396, 2003.

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

[33] Dantzig, G. and Ramser, R., "The truck dispatching problem," *Management Science*, vol. 6, pp. 80–91, 1959.

[34] Dantzig, G., Fulkerson, R., and Johnson, S., "Solution of a large-scale traveling salesman problem," *Operations Research*, vol. 2, pp. 393–410, 1954.

[35] Dash, S. and Günlük, O., "Valid inequalities based on simple mixed-integer sets," *Mathematical Programming*, vol. 105, no. 1, pp. 29–53, 2006.

[36] Dash, S. and Günlük, O., "Valid inequalities based on the interpolation procedure," *Mathematical Programming*, vol. 106, no. 1, pp. 111–136, 2006.

[37] Dash, S. and Günlük, O., "On the strength of Gomory mixed-integer cuts as group cuts," *Mathematical Programming*, Available online first. DOI: 10.1007/s10107-007-0179-4, 2007.

[38] Dash, S., Günlük, O., and Lodi, A., "On the MIR closure of polyhedra," in *IPCO, Lecture notes in computer science, M. Fischetti and D. Williamson (Eds.)*, vol. 4513, pp. 337–351, 2007.

[39] de Farias, I., Johnson, E., and Nemhauser, G., "A generalized assignment problem with special ordered sets: A polyhedral approach," *Mathematical Programming*, vol. 89, pp. 187–203, 2000.

[40] de Farias, I., Johnson, E., and Nemhauser, G., "Facets of the complementarity knapsack polytope," *Mathematics of Operations Research*, vol. 27, pp. 210–226, 2002.

[41] de Farias, I. and Nemhauser, G., "A family of inequalities for the generalized assignment polytope," *Operations Research Letters*, vol. 29, pp. 49–51, 2001.

[42] Dhiflaoui, M., Funke, S., Kwappik, C., Mehlhorn, K., Seel, M., Schömer, E., Schulte, R., and Weber, D., "Certifying and repairing solutions to large lps how good are lp-solvers?," in *SODA 2003, Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 255–256, 2003.

[43] Dinkelbach, W., "Die maximierung eines quotienten zweier linearer funktionen unter linearen nebenbedingungen," *Wahrscheinlichkeitstheorie*, vol. 1, pp. 141–145, 1962.

[44] Dinkelbach, W., "On nonlinear fractional programming," *Management Science*, vol. 13, pp. 492–492, 1967.

[45] Dolan, E. and Moré, J., "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2001.

[46] Easton, T. and Gutierrez, T., "Sequential and simultaneous uplifting of general integer variables," *Submitted to Mathematical Programming*, 2007.

[47] Espinoza, D. G., *On Linear Programming, Integer Programming and Cutting Planes*. PhD thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, March 2006.

[48] Fischetti, M. and Lodi, A., "On the knapsack closure of 0-1 integer linear problems." Presentation at 10th International Workshop on Combinatorial Optimization, Aussois (2006). Available at `http://www-id.imag.fr/IWCO2006/slides/Fischetti.pdf`.

[49] Fischetti, M. and Lodi, A., "Optimizing over the first Chvátal closure," *Mathematical Programming B*, vol. 110, no. 1, pp. 3–20, 2007.

[50] Fischetti, M. and Monaci, M., "How tight is the corner relaxation?," *Discrete Optimization*, 2008. To appear.

[51] Fischetti, M. and Saturni, C., "Mixed-integer cuts from cyclic groups," *Mathematical Programming A*, vol. 109, no. 1, pp. 27–53, 2007.

[52] Gärtner, B., "Exact arithmetic at low cost - a case study in linear programming," *Computatioinal Geometry*, vol. 13, pp. 121–139, 1999.

[53] Goldberg, D., "What every computer scientist should know about floating-point arithmetic," *ACM Computing Surveys (CSUR)*, vol. 23, pp. 5–48, 1991.

[54] Gomory, R. E., "Outline of an algorithm for integer solutions to linear programs," *Bulletin of the American Mathematical Society*, vol. 64, pp. 275–278, 1958.

[55] Gomory, R. E., "Some polyhedra related to combinatorial problems," *Journal of Linear Algebra and its Applications*, vol. 2, pp. 451–558, 1969.

[56] Gomory, R. E., "Early integer programming (reprinted)," *Operations Research*, vol. 50, pp. 78–81, Jan 2002.

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

[58] Gomory, R. E. and Johnson, E., "Some continuous functions related to corner polyhedra II," *Mathematical Programming*, vol. 3, pp. 359–389, 1972.

[59] Gomory, R. and Johnson, E., "T-space and cutting planes," *Mathematical Programming*, vol. 96, pp. 341–375, May 2003.

[60] Gomory, R., Johnson, E., and Evans, L., "Corner polyhedra and their connection with cutting planes," *Mathematical Programming*, vol. 96, pp. 321–339, May 2003.

[61] Gouveia, L., "A $2n$ constraint formulation for the capacitated minimal spanning tree problem," *Operations Research*, vol. 43, no. 1, pp. 130–141, 1995.

[62] Goycoolea, M., *Cutting Planes for Large Mixed Integer Programming Models*. PhD thesis, Georgia Institute of Technology, 2006.

[63] Granlund, T., "The GNU multiple precision arithmetic library." Available on-line at `http://www.swox.com/gmp/`.

[64] Gu, Z., Nemhauser, G. L., and Savelsbergh, M. W. P., "Lifted cover inequalities for 0-1 integer programs: Computation," *INFORMS Journal on Computing*, vol. 10, pp. 427–437, 1998.

[65] Gu, Z., Nemhauser, G. L., and Savelsbergh, M. W. P., "Lifted cover inequalities for 0-1 integer programs: Complexity," *INFORMS Journal on Computing*, vol. 11, pp. 117–123, 1999.

[66] Gu, Z., Nemhauser, G. L., and Savelsbergh, M. W. P., "Lifted flow cover inequalities for 0-1 integer programs," *Mahtematical Programming*, vol. 85, pp. 437–467, 1999.

[67] Gu, Z., Nemhauser, G. L., and Savelsbergh, M. W. P., "Sequence independent lifting in mixed integer programming," *Journal Of Combinatorial Optimization*, vol. 4, no. 1, pp. 109–129, 2000.

[68] Hales, T. C., "A proof of the Kepler conjecture," *Annals of Mathematics*, vol. 162, pp. 1065–1185, 2005.

[69] Hammer, P., Johnson, E., and Peled, U., "Facet of regular 0-1 polytopes," *Mathematical Programming*, vol. 8, no. 2, pp. 179–206, 1975.

[70] Hong, S., *A Linear Programming Approach for the Traveling Salesman Problem.* PhD thesis, Johns Hopkins University, 1972.

[71] IEEE, "ANSI/IEEE Standard 754-1985, standard for binary floating point arithmetic." Available on-line at `http://standards.ieee.org/`, 1985.

[72] ISO/IEC, "Programming languages–C. ISO/IEC 9899:1999," 1999.

[73] Jansson, C., "Rigorous lower and upper bounds in linear programming," *SIAM Journal on Optimization*, vol. 14, pp. 914–935, 2004.

[74] Kaparis, K. and Letchford, A., "Separation algorithms for 0-1 knapsack polytopes," *Technical report available at Optimization Online*, June 2007.

[75] Koch, T., "The final NETLIB-lp results," *Operations Research Letters*, vol. 32, pp. 138–142, 2003.

[76] Letchford, A. and Lodi, A., "Strengthening Chvátal-Gomory cuts and Gomory fractional cuts," *Operations Research Letters*, vol. 30, no. 2, pp. 74–82, 2002.

[77] Louveaux, Q. and Wolsey, L., "Lifting, superadditivity, mixed integer rounding and single node flow sets revisited," *4OR: A Quarterly Journal of Operations Research*, vol. 1, pp. 173–207, 2003.

[78] Marchand, H. and Wolsey, L., "Aggregation and mixed integer rounding to solve MIPs," *Operations Research*, vol. 49, pp. 363–371, 2001.

[79] Margot, F., "Testing cut generators for mixed-integer linear programming," Tepper Working Paper 2007 E-43, Carnegie Mellon University, Pittsburgh, USA, 2007.

[80] Martin, G. T., "Solving the traveling salesman problem by integer linear programming," technical report, C-E-I-R, New York, USA, 1966.

[81] Martos, B., "Hyperbolic programming," *Publ. Math. Inst., Hungarian Academy of Sciences, ser. B*, vol. 5, pp. 386–406, 1960.

[82] Martos, B., "Hyperbolic programming," *Naval Research Logistics Quarterly*, vol. 11, pp. 135–155, 1964.

[83] Miliotis, P., "Using cutting planes to solve the symmetric travelling salesman problem," *Mathematical Programming*, vol. 15, pp. 177–188, 1978.

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

[85] Nemhauser, G. L. and Wolsey, L. A., *Integer and Combinatorial Optimization*. Discrete Mathematics and Optimization, Wiley-Interscience, 1999.

[86] Nemhauser, G. and Jr., L. T., "Properties of vertex packing and independence systems polyhedra," *Mathematical programming*, vol. 6, pp. 48–61, 1974.

[87] Neumaier, A. and Shcherbina, O., "Safe bounds in linear and mixed-integer lineear programming," *Mathematical Programming*, vol. 99, pp. 283–296, 2004.

[88] Padberg, M. W., "On the facial structure of set packing polyhedra," *Mathematical programming*, vol. 5, pp. 199–215, 1973.

[89] Padberg, M. W. and Rinaldi, G., "A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems," *SIAM Review*, vol. 33, pp. 60–100, 1991.

[90] Padberg, M., "A note on zero-one programming," *Operations Research*, vol. 23, pp. 833–837, 1975.

[91] Pollatschek, M., *Algorithms for Finite Weighed Graphs*. PhD thesis, Faculty of Industrial and Management Engineering, Technion-Israel Institute of Technology, 1970.

[92] Richard, J., de Farias, I., and Nemhauser, G., "Lifted inequalities for 0-1 mixed integer programming: Basic theory and algorithms," *Mathematical Programming*, vol. 98, pp. 89–113, 2003.

[93] Richard, J., de Farias, I., and Nemhauser, G., "Lifted inequalities for 0-1 mixed integer programming: Superlinear lifting," *Mathematical Programming*, vol. 98, pp. 115–143, 2003.

[94] Schaible, S., "Fractional programming II: on Dinkelbach's algorithm," *Management science*, vol. 22, no. 8, pp. 868–873, 1976.

[95] Schaible, S. and Shi, J., *Nonlinear analysis and convex analysis*, ch. Recent developments in fractional programming: single ratio and max-min case, pp. 493–506. Yokohama Publishers, Yokohama, 2004.

[96] Stancu-Minasian, I. M., *Fractional Programming: Theory, Methods and Applications*. Springer, 1 ed., 1997.

[97] Uchoa, E., "Robust branch-and-cut-and-price for the CMST problem and extended capacity cuts." Presentation in the MIP 2005 Workshop, Minneapolis (2005). Available at `http://www.ima.umn.edu/matter/W7.25-29.05/activities/Uchoa-Eduardo/cmst-ecc-IMA.pdf`.

[98] Uchoa, E., Fukasawa, R., Lysgaard, J., Pessoa, A., de Aragão, M. P., and Andrade, D., "Robust branch-cut-and-price for the capacitated minimum spanning tree problem over a large extended formulation," *Mathematical Programming*, vol. 112, no. 2, pp. 443–472, 2008.

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

[100] Wolsey, L., "Faces for a linear inequality in 0-1 variables," *Mathematical Programming*, vol. 8, no. 2, pp. 165–178, 1975.

[101] Wolsey, L., "Facets and strong valid inequalities for integer programs," *Operations Research*, vol. 24, no. 2, pp. 367–372, 1976.

[102] Wolsey, L., "Valid inequalities and superadditivity for 0-1 integer programs," *Mathematics of Operations Research*, vol. 2, no. 1, pp. 66–77, 1977.

[103] Yan, X. Q. and Boyd, E. A., "Cutting planes for mixed-integer knapsack polyhedra," *Mathematical Programming*, vol. 81, no. 2, pp. 257–262, 1998.

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

VITA

Ricardo Fukasawa was born in Rio de Janeiro, Brazil on August 31st, 1977. In 1995 he enrolled in the Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), from which he received the degree of Electrical Engineer with emphasis on Decision Support Systems. After graduating from PUC-Rio in 2000, he started his Masters in the same university, in the same major. In 2002, he obtained his M.Sc. after defending his dissertation entitled: "Solution of railroad logistics problems using integer programming", which he developed while working with Dr. Oscar Porto, Dr. Marcus Poggi de Aragão and Dr. Eduardo Uchoa. After getting his M.Sc., he went to work for GAPSO Inc., where he was one of the people responsible for developing softwares for logistics using integer programming. In August 2003, he joined the School of Industrial and Systems Engineering at the Georgia Institute of Technology, majoring in Algorithms, Combinatorics and Optimization, where he completed his doctoral research on cutting planes for mixed-integer programming problems derived from single-row relaxations under the supervision of Dr. William J. Cook.