Cutting Planes for Mixed Integer Programming

by
ANDREA QUALIZZA

Submitted in partial fulfillment
of the requirements for the degree of
DOCTOR OF PHILOSOPHY
in
Industrial Administration
(Algorithms, Combinatorics and Optimization)

September 2011

Dissertation Committee:
Egon Balas (Chair)
Gérard Cornuéjols
François Margot
Shlomo Ta'asan

Tepper School of Business
Carnegie Mellon University

Abstract

My work focuses on cutting planes technology in Mixed Integer Programming. I explore novel classes of valid linear inequalities to strengthen linear relaxations of both Linear and Nonlinear Mixed Integer problems. My dissertation consists of three chapters that investigate theoretically and computationally the families of cuts considered.

The first chapter is based on joint work with Prof. Pietro Belotti and Prof. François Margot. We study linear relaxations of Quadratically Constrained Quadratic Programs. The proposed relaxations are models with both semidefinite constraints (PSD) and linear constraints given by the Reformulation Linearization Technique (RLT). It is known from the literature (Anstreicher, 2007) that PSD and RLT used together yield better bounds than each technique used separately. We adopt a linear outer-approximation of the PSD cone, and we use exclusively linear programming tools to enforce the PSD condition via a cutting plane approach in the lifted space containing the $Y_{ij} = x_i x_j$ variables. We study new classes of valid linear inequalities and we test their effectiveness empirically. These include sparse PSD cuts and cuts derived from principal minors. Computational results based on instances from the GLOBALLib and Boxed Constrained Quadratic Programs show that this approach yields better bounds than using solely the standard PSD cuts on top of the RLT inequalities. The C++ code developed for this study has been included in Coin-OR as part of the Couenne project (exact solver for MINLPs).

In the second chapter I present a work closely related to the recent developments in the area of "cuts from multiple rows of the simplex tableau" (Andersen et al., 2007). This chapter is based on joint work with Prof. Egon Balas. We generate intersection cuts from lattice-free convex sets as lift-and-project cuts from multiple-term disjunctions. We use the concept of "Disjunctive Hull" defined for a Mixed Integer Program at a fractional vertex v of its linear relaxation P as the convex hull of points in P satisfying all basic disjunctions that cut off v but no integer point. We examine the relationship between the Disjunctive Hull and the Integer Hull and we give procedures to generate inequalities for the Integer Hull derived from the Cut Generating Linear Program associated to the Disjunctive Hull. Strengthening techniques based on coefficient modularization and monoidal strengthening are also discussed. In this chapter we also analyze the case of 0-1 programming which has not been covered in the literature. Our framework applies to this setting with minor changes and produces valid families of cuts for the 0-1 case but invalid for general integer programs. These cuts include the triangle, quadrilateral, split cuts for the MIP case, and cuts from cones and truncated cones for the 0-1 setting. Moreover we run an experiment in which we separate two families of non facet defining cuts: we consider cuts from fixed shape triangles

of type 1 and conic disjunctions having the apex at one vertex of the unit cube and two extreme rays, each containing an additional vertex of the cube. In practice, in order to match the strength of the non facet defining inequalites, a large number of facet defining inequalites is typically needed. Triangles of type 1 produce a consistent improvement over the Gomory cuts, as shown in experiments on the MIPLIB 3 0-1 instances.

In the third chapter I present a theoretical result on strengthening valid inequalities for Mixed Integer Linear Programs. This chapter is also based on joint work with Prof. Egon Balas. There are two distinct strengthening methods for disjunctive cuts; they differ in the way they modularize the coefficients associated to integer constrained variables. We introduce a new variant of one of these methods, the monoidal cut strengthening procedure (Balas and Jeroslow, 1980), based on the paradox that sometimes weakening a disjunction helps the strengthening procedure and results in sharper cuts. We first derive a general result that applies to cuts from disjunctions with any number of terms. It defines the coefficients of the cut in a way that takes advantage of the option of adding new terms to the disjunction. We then specialize this result to the case of split cuts, in particular Gomory Mixed Integer cuts, and to intersection cuts from multiple rows of a simplex tableau. In both instances we give conditions for the new cuts to have stronger coefficients than the cuts obtained by either of the two currently known strengthening procedures.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Egon Balas, for his continuous guidance and encouragement throughout my studies. He has been a great source of inspiration for me and I really feel honored being advised by such a great person.

I am indebted to Professor François Margot for his help and advice on the work included in the first chapter and for his insightful comments to improve the second and third chapters. Special thanks to Professor Gérard Cornuéjols for the valuable suggestions that greatly enriched my work. I also thank Professor Shlomo Ta'asan for agreeing to be the outside reader for my dissertation. I am grateful to Professor Pietro Belotti and Anureet Saxena for their invaluable contributions to the first chapter of this dissertation.

I would never been able to get to this point without my former advisor Professor Paolo Serafini at the University of Udine who was the first to introduce me to research and constantly believed in me ever since. I also thank Professor Giuseppe Lancia, Professor Alberto Policriti and Professor Franca Rinaldi that together with Professor Serafini put a great effort in training me and motivating me to embark on this enterprise.

I thank my colleagues and friends Alessandro Condotta, Amin Sayedi, Amitabh Basu, Andre Cire, Carla Michini, Daniele Catanzaro, David Bergman, Diego Falsini, Ishani Aggarwal, Luca Parolini, Marco Molinaro, Marco Vincenzi, Negar Soheili, Qihang Lin, Samid Hoda, Selvaprabu Nadarajah, Siriporn Liangthong and Yangfang Zhou. I shared with them wonderful times and they provided me with moral support and helped me go through difficult moments. I owe much to Amin, I have learned a lot from him and have truly enjoyed our conversations.

Thank you to my girlfriend Jiaxu who stuck with me during the long months of the preparation of this dissertation. I thank her for her patience, encouragement and love with all my heart.

I also would like to express my appreciation to Lawrence Rapp for making life easier for me at Carnegie Mellon. I gratefully acknowledge the financial support of the William Larimer Mellon fellowship.

Last but not least, I thank my parents Laura and Gino and my brother Matteo for their constant support since I was very young. This journey would not have been possible without them.

Andrea Qualizza

Contents

List of Figures

List of Tables

Chapter 1

Linear Programming Relaxations for QCQPs

The work presented in this chapter has been published as "Linear Programming Relaxations of Quadratically Constrained Quadratic Programs" by A. Qualizza, P. Belotti and F. Margot in IMA Volume Series, Springer, 2010.

1.1 Introduction

Many combinatorial problems have Linear Programming (LP) relaxations that are commonly used for their solution through branch-and-cut algorithms. Some of them also have stronger relaxations involving positive semidefinite (PSD) constraints. In general, stronger relaxations should be preferred when solving a problem, thus using these PSD relaxations is tempting. However, they come with the drawback of requiring a Semidefinite Programming (SDP) solver, creating practical difficulties for an efficient implementation within a branch-and-cut algorithm. Indeed, a major weakness of current SDP solvers compared to LP solvers is their lack of efficient warm starting mechanisms. Another weakness is solving problems involving a mix of PSD constraints and a large number of linear inequalities, as these linear inequalities put a heavy toll on the linear algebra steps required during the solution process.

In this chapter, we investigate LP relaxations of PSD constraints with the aim of capturing most of the strength of the PSD relaxation, while still being able to use an LP solver. The LP relaxation we obtain is an outer-approximation of the PSD cone, with the typical convergence difficulties when aiming to solve problems to optimality. We thus do not cast this work as an efficient way to solve PSD problems, but we aim at finding practical ways to approximate PSD constraints with linear ones.

We restrict exclusively to Quadratically Constrained Quadratic Programming ($QCQP$). A QCQP problem with variables $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ is a problem of the form

$$\begin{array}{lll} \max & x^T Q_0 x + a_0^T x + b_0^T y & \\ \text{s.t.} & & \\ & x^T Q_k x + a_k^T x + b_k^T y \ \leq c_k & \text{for } \ k = 1, 2, \ldots, p \\ & l_{x_i} \leq x_i \leq u_{x_i} & \text{for } \ i = 1, 2, \ldots, n \\ & l_{y_j} \leq y_j \leq u_{y_j} & \text{for } \ j = 1, 2, \ldots, m \end{array} \tag{1.1}$$

where, for $k = 0, 1, 2, \ldots, p$, Q_k is a rational symmetric $n \times n$-matrix, a_k is a rational n-vector, b_k is a rational m-vector, and $c_k \in \mathbb{Q}$. Moreover, the lower and upper bounds l_{x_i}, u_{x_i} for $i = 1, \ldots, n$, and l_{y_j}, u_{y_j} for $j = 1, \ldots, m$ are all finite. If Q_0 is negative semidefinite and Q_k is positive semidefinite for each $k = 1, 2, \ldots, p$, problem (1.1) is convex and thus easy to solve. Otherwise, the problem is NP-hard [22].

An alternative *lifted* formulation for (1.1) is obtained by replacing each quadratic term $x_i x_j$ with a new variable X_{ij}. Let $X = xx^T$ be the matrix with entry X_{ij} corresponding to the quadratic term $x_i x_j$. For square matrices A and B of the same dimension, let $A \bullet B$ denote the *Frobenius inner product* of A and B, i.e., the trace of $A^T B$. Problem (1.1) is then equivalent to

$$\begin{array}{lll} \max & Q_0 \bullet X + a_0^T x + b_0^T y & \\ \text{s.t.} & & \\ & Q_k \bullet X + a_k^T x + b_k^T y \ \leq c_k & \text{for } \ k = 1, 2, \ldots, p \\ & l_{x_i} \leq x_i \leq u_{x_i} & \text{for } \ i = 1, 2, \ldots, n \\ & l_{y_j} \leq y_j \leq u_{y_j} & \text{for } \ j = 1, 2, \ldots, m \\ & X = xx^T. & \end{array} \tag{1.2}$$

The difficulty in solving problem (1.2) lies in the non-convex constraint $X = xx^T$. A relaxation, dubbed PSD, that is possible to solve relatively efficiently is obtained by relaxing this constraint to the requirement that $X - xx^T$ be positive semidefinite, i.e., $X - xx^T \succeq 0$. An alternative relaxation of (1.1), dubbed RLT, is obtained by the Reformulation Linearization Technique [52], using products of pairs of original constraints and bounds and replacing nonlinear terms with new variables.

Anstreicher [3] compares the PSD and RLT relaxations on a set of quadratic problems with box constraints, i.e., QCQP problems with $p = 0$ and with all the variables bounded between 0 and 1. He shows that the PSD relaxations of these instances are fairly good and that combining the PSD and RLT relaxations yields significantly tighter relaxations than either of the PSD or RLT relaxations. The drawback of combining the two relaxations is that current SDP solvers have difficulties to handle the large number of linear constraints of the RLT.

Our aim is to solve relaxations of (1.1) using exclusively linear programming tools. The RLT is readily applicable for our purposes, while the PSD technique requires a cutting plane approach as described in Section 1.2.

In Section 1.3 we consider several families of valid cuts. The focus is essentially on capturing the strength of the positive semidefinite condition using standard cuts [53], and some sparse versions of these.

We analyze empirically the strength of the considered cuts on instances taken from GLOBALLib [35] and quadratic programs with box constraints described in more details in the next section. Implementation and computational results are presented in Section 1.4. Finally, Section 1.5 summarizes the results and gives possible directions for future research.

1.2 Relaxations of QCQP problems

A typical approach to get bounds on the optimal value of a QCQP is to solve a convex relaxation. Since our aim is to work with linear relaxations, the first step is to linearize (1.2) by relaxing the last constraint to $X = X^T$. We thus get the Extended formulation

$$\begin{array}{lll} \max & Q_0 \bullet X + a_0^T x + b_0^T y & \\ \text{s.t.} & & \\ & Q_k \bullet X + a_k^T x + b_k^T y \ \leq c_k & \text{for } \ k = 1, 2, \ldots, p \\ & l_{x_i} \leq x_i \leq u_{x_i} & \text{for } \ i = 1, 2, \ldots, n \\ & l_{y_j} \leq y_j \leq u_{y_j} & \text{for } \ j = 1, 2, \ldots, m \\ & X = X^T. & \end{array} \tag{1.3}$$

(1.3) is a Linear Program with $n(n+3)/2+m$ variables and the same number of constraints as (1.1). Note that the optimal value of (1.3) is usually a weak upper bound for (1.1), as no constraint links the values of the x and X variables. Two main approaches for doing that have been proposed and are based on relaxations of the last constraint of (1.2), namely

$$X - xx^T = 0. \tag{1.4}$$

They are known as the Positive Semidefinite (PSD) relaxation and the Reformulation Linearization Technique (RLT) relaxation.

1.2.1 PSD Relaxation

As $X - xx^T = 0$ implies $X - xx^T \succcurlyeq 0$, using this last constraint yields a convex relaxation of (1.1). This is the approach used in [53, 55, 56, 58], among others.

Moreover, using Schur's complement

$$X - xx^T \succcurlyeq 0 \qquad \Leftrightarrow \qquad \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succcurlyeq 0,$$

and defining

$$\tilde{Q}_k = \begin{pmatrix} -c_k & a_k^T/2 \\ a_k/2 & Q_k \end{pmatrix}, \qquad \tilde{X} = \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix},$$

we can write the PSD relaxation of (1.1) in the compact form

$$\begin{array}{lll} \max & \tilde{Q}_0 \bullet \tilde{X} + b_0^T y & \\ \text{s.t.} & & \\ & \tilde{Q} \bullet \tilde{X} + b_k^T y \leq 0 & k = 1, 2, \ldots, p \\ & l_{x_i} \leq x_i \leq u_{x_i} & i = 1, 2, \ldots, n \\ & l_{y_j} \leq y_j \leq u_{y_j} & j = 1, 2, \ldots, m \\ & \tilde{X} \succcurlyeq 0. & \end{array} \tag{1.5}$$

This is a positive semidefinite problem with linear constraints. It can thus be solved in polynomial time using interior point algorithms. (1.5) is tighter than usual linear relaxations for problems such as the Maximum Cut, Stable Set, and Quadratic Assignment problems [60]. All these problems can be formulated as QCQPs. For convenience of notation we refer to the relaxation (1.5) as **PSD**.

1.2.2 RLT Relaxation

The Reformulation Linearization Technique [52] can be used to produce a relaxation of (1.1). It adds linear inequalities to (1.3). These inequalities are derived from the variable bounds and constraints of the original problem as follows: multiply together two original constraints or bounds and replace each product term $x_i x_j$ with the variable X_{ij}. For instance, let $x_i, x_j,\ i, j \in \{1, 2, \ldots, n\}$ be two variables from (1.1). By taking into account only the four original bounds $x_i - l_{x_i} \geq 0$, $x_i - u_{x_i} \leq 0$, $x_j - l_{x_j} \geq 0$, $x_j - u_{x_j} \leq 0$, we get the *RLT* inequalities

$$
\begin{array}{ll}
X_{ij} - l_{x_i} x_j - l_{x_j} x_i & \geq -l_{x_i} l_{x_j}, \\
X_{ij} - u_{x_i} x_j - u_{x_j} x_i & \geq -u_{x_i} u_{x_j}, \\
X_{ij} - l_{x_i} x_j - u_{x_j} x_i & \leq -l_{x_i} u_{x_j}, \\
X_{ij} - u_{x_i} x_j - l_{x_j} x_i & \leq -u_{x_i} l_{x_j}\ .
\end{array}
\tag{1.6}
$$

Anstreicher [3] observes that, for Quadratic Programs with box constraints, the *PSD* and *RLT* constraints together yield much better bounds than those obtained from the **PSD** or **RLT** relaxations. In this work, we want to capture the strength of both techniques and generate a Linear Programming relaxation of (1.1).

Notice that the four inequalities above, introduced by McCormick [44], constitute the convex envelope of the set $\{(x_i, x_j, X_{ij}) \in \mathbb{R}^3 : l_{x_i} \leq x_i \leq u_{x_i}, l_{x_j} \leq x_j \leq u_{x_j}, X_{ij} = x_i x_j\}$ as proven by Al-Khayyal and Falk [1], i.e., they are the tightest relaxation for the single term X_{ij}.

1.3 Our Framework

While the *RLT* constraints are linear in the variables in the formulation (1.3) and therefore can be added directly to (1.3), this is not the case for the *PSD* constraint. We use a linear outer-approximation of the **PSD** relaxation and a cutting plane framework, adding a linear inequality separating the current solution from the *PSD* cone.

The initial relaxation we use and the various cuts generated by our separation procedure are described in more details in the next sections.

1.3.1 Initial Relaxation

Our initial relaxation is the formulation (1.3) together with the $O(n^2)$ *RLT* constraints derived from the bounds on the variables $x_i,\ i = 1, 2, \ldots, n$. We did not include the *RLT* constraints derived from the problem constraints due to their large number and the fact that we want to avoid the introduction of extra variables for the multivariate terms that occur when quadratic constraints are multiplied together.

The bounds $[L_{ij}, U_{ij}]$ for the extended variables X_{ij} are computed as follows:

$$L_{ij} = \min\{l_{x_i}l_{x_j};\ l_{x_i}u_{x_j};\ u_{x_i}l_{x_j};\ u_{x_i}u_{x_j}\}, \quad \forall i = 1, 2, \ldots, n;\ j = i, \ldots, n$$
$$U_{ij} = \max\{l_{x_i}l_{x_j};\ l_{x_i}u_{x_j};\ u_{x_i}l_{x_j};\ u_{x_i}u_{x_j}\}, \quad \forall i = 1, 2, \ldots, n;\ j = i, \ldots, n.$$

In addition, equality (1.4) implies $X_{ii} \geq x_i^2$. We therefore also make sure that $L_{ii} \geq 0$.

1.3.2 PSD Cuts

We use the equivalence that a matrix is positive semidefinite if and only if

$$v^T \tilde{X} v \geq 0 \quad \text{for all} \quad v \in \mathbb{R}^{n+1} . \tag{1.7}$$

We can reformulate **PSD** as the semi-infinite Linear Program

$$\begin{array}{lll}
\max & \tilde{Q}_0 \bullet \tilde{X} + b_0^T y & \\
\text{s.t.} & & \\
& \tilde{Q} \bullet \tilde{X} + b_k^T y \quad \leq c_k & \text{for} \quad k = 1, 2, \ldots, p \\
& l_{x_i} \leq x_i \leq u_{x_i} & \text{for} \quad i = 1, 2, \ldots, n \\
& l_{y_j} \leq y_j \leq u_{y_j} & \text{for} \quad j = 1, 2, \ldots, m \\
& v^T \tilde{X} v \quad \geq 0 & \text{for all} \quad v \in \mathbb{R}^{n+1}.
\end{array} \tag{1.8}$$

A practical way to use (1.8) is to adopt a cutting plane approach to separate constraints (1.7) as done in [53].

Let $\tilde{X}^*$ be an arbitrary point in the space of the $\tilde{X}$ variables. The spectral decomposition of $\tilde{X}^*$ is used to decide if $\tilde{X}^*$ is in the PSD cone or not. Let the eigenvalues and corresponding orthonormal eigenvectors of $\tilde{X}^*$ be λ_k and v_k for $k = 1, 2, \ldots, n$, and assume without loss of generality that $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ and let $t \in \{0, \ldots, n\}$ such that $\lambda_t < 0 \leq \lambda_{t+1}$. If $t = 0$, then all the eigenvalues are non negative and $\tilde{X}^*$ is positive semidefinite. Otherwise, $v_k^T \tilde{X}^* v_k = v_k^T \lambda_k v_k = \lambda_k < 0$ for $k = 1, \ldots, t$. Hence, the valid cut

$$v_k^T \tilde{X} v_k \geq 0 \tag{1.9}$$

is violated by $\tilde{X}^*$. Cuts of the form (1.9) are called PSDCUTs in the remainder of the chapter.

The above procedure has two major weaknesses: First, only one cut is obtained from eigenvector v_k for $k = 1, \ldots, t$, while computing the spectral decomposition requires a non trivial investment in cpu time, and second, the cuts are usually very dense, i.e. almost all entries in vv^T are nonzero. Dense cuts are frowned upon when used in a cutting plane approach, as they might slow down considerably the reoptimization of the linear relaxation.

To address these weaknesses, we describe in the next section a heuristic to generate several sparser cuts from each of the vectors v_k for $k = 1, \ldots, t$.

Sparsify$(v, \tilde{X}, pct_{NZ}, pct_{VIOL})$

1 $min_{VIOL} \leftarrow -v^T \tilde{X} v \cdot pct_{VIOL}$
2 $max_{NZ} \leftarrow \lfloor length[v] \cdot pct_{NZ} \rfloor$
3 $w \leftarrow v$
4 $perm \leftarrow$ random permutation of 1 **to** $length[w]$
5 **for** $j \leftarrow 1$ **to** $length[w]$
6   **do**
7         $z \leftarrow w$, $z[perm[j]] \leftarrow 0$
8   **if** $-z^T \tilde{X} z > min_{VIOL}$
9   **then** $w \leftarrow z$
10 **if** number of non-zeroes in $w < max_{NZ}$
11   **then** output w

Figure 1.1: Sparsification procedure for the PSD cuts

1.3.3 Sparsification of PSD cuts

A simple idea to get sparse cuts is to start with vector $w = v_k$, for $k = 1, \ldots, t$, and iteratively set to zero some component of w, provided that $w^T \tilde{X}^* w$ remains sufficiently negative. If the entries are considered in random order, several cuts can be obtained from a single eigenvector v_k. For example, consider the *Sparsify* procedure in Figure 1.1, taking as parameters an initial vector v, a matrix $\tilde{X}$, and two numbers between 0 and 1, pct_{NZ} and pct_{VIOL}, that control the maximum percentage of nonzero entries in the final vector and the minimum violation requested for the corresponding cut, respectively. In the procedure, parameter $length[v]$ identifies the size of vector v.

It is possible to implement this procedure to run in $O(n^2)$ if $length[v] = n + 1$: Compute and update a vector m such that

$$m_j = \sum_{i=1}^{n+1} w_j w_i \tilde{X}_{ij} \quad \text{for} \quad j = 1, \ldots, n+1 \ .$$

Its initial computation takes $O(n^2)$ and its update, after a single entry of w is set to 0, takes $O(n)$. The vector m can be used to compute the left hand side of the test in step 8 in constant time given the value of the violation d for the inequality generated by the current vector w: Setting the entry $\ell = perm[j]$ of w to zero reduces the violation by $2m_\ell - w_\ell^2 \tilde{X}_{\ell\ell}$ and thus the violation of the resulting vector is $(d - 2m_\ell + w_\ell^2 \tilde{X}_{\ell\ell})$.

A slight modification of the procedure is used to obtain several cuts from the same eigenvector: Change the loop condition in step 5 to consider the entries in *perm* in cyclical order, from all possible starting points s in $\{1, 2 \ldots, length[w]\}$, with the additional condition that entry $s - 1$ is not set to 0 when starting from s to guarantee that we do not generate always the same cut. From our experiments, this simple idea produces collections of sparse and well-diversified cuts. This is referred to as SPARSE1 in the remainder of the chapter.

We also consider the following variant of the procedure given in Figure 1.1. Given a vector w, let $\tilde{X}_{[w]}$ be the principal minor of $\tilde{X}$ induced by the indices of the nonzero entries in w. Replace step 7 with

7. $z \leftarrow \bar{w}$ where $\bar{w}$ is an eigenvector corresponding to the most negative eigenvalue of a spectral decomposition of $\tilde{X}_{[w]}$, $z[perm[j]] \leftarrow 0$.

This is referred to as SPARSE2 in the remainder, and we call the cuts generated by SPARSE1 or SPARSE2 described above *Sparse PSD cuts*.

Once sparse PSD cuts are generated, for each vector w generated, we can also add all PSD cuts given by the eigenvectors corresponding to negative eigenvalues of a spectral decomposition of $\tilde{X}_{[w]}$. These cuts are valid and sparse. They are called *Minor PSD cuts* and denoted by MINOR in the following.

An experiment to determine good values for the parameters pct_{NZ} and pct_{VIOL} was performed on the 38 GLOBALLIB instances and 51 BoxQP instances described in Section 1.4.1. It is run by selecting two sets of three values in $[0, 1]$, $\{V_{LOW}, V_{MID}, V_{UP}\}$ for pct_{VIOL} and $\{N_{LOW}, N_{MID}, N_{UP}\}$ for pct_{NZ}. The nine possible combinations of these parameter values are used and the best of the nine (V_{best}, N_{best}) is selected. We then center and reduce the possible ranges around V_{best} and N_{best}, respectively, and repeat the operation. The procedure is stopped when the best candidate parameters are (V_{MID}, N_{MID}) and the size of the ranges satisfy $|V_{UP} - V_{LOW}| \leq 0.2$ and $|N_{UP} - N_{LOW}| \leq 0.1$.

In order to select the best value of the parameters, we compare the bounds obtained by both algorithms after $1, 2, 5, 10, 20$, and 30 seconds of computation. At each of these times, we count the number of times each algorithm outperforms the other by at least 1% and the winner is the algorithm with the largest number of wins over the 6 clocked times. It is worth noting that typically the majority of the comparisons end up as ties, implying that the results are not extremely sensitive to the selected values for the parameters.

For SPARSE1, the best parameter values are $pct_{VIOL} = 0.6$ and $pct_{NZ} = 0.2$. For SPARSE2, they are $pct_{VIOL} = 0.6$ and $pct_{NZ} = 0.4$. These values are used in all experiments using either SPARSE1 or SPARSE2 in the remainder of the chapter.

1.4 Computational Results

In the implementation, we have used the Open Solver Interface (Osi-0.97.1) from COIN-OR [23] to create and modify the LPs and to interface with the LP solvers ILOG Cplex-11.1. To compute eigenvalues and eigenvectors, we use the `dsyevx` function provided by the LAPACK library version 3.1.1. We also include a cut management procedure to reduce the number of constraints in the outer approximation LP. This procedure, applied at the end of each iteration, removes the cuts that are not satisfied with equality by the optimal solution. Note however that the constraints from the of the initial relaxations are never removed, only constraints from added cutting planes are possibly removed.

The machine used for the tests is a 64 bit 2.66GHz AMD processor, 64GB of RAM memory, and Linux kernel 2.6.29. Tolerances on the accuracy of the primal and dual solutions of the LP solver and LAPACK calls are set to 10^{-8}.

The set of instances used for most experiments consists of 51 BoxQP instances with at most 50 variables and the 38 GLOBALLib instances as described in Section 1.4.1.

For an instance $\mathcal{I}$ and a given relaxation of it, we define the *gap closed* by the relaxation as

$$100 \cdot \frac{RLT - BND}{RLT - OPT}, \tag{1.10}$$

where BND and RLT are the optimal value for the given relaxation and the initial relaxation respectively, and OPT is either the optimal value of $\mathcal{I}$ or the best known value for a feasible solution. The OPT values are taken from [49].

1.4.1 Instances

Tests are performed on a subset of instances from GLOBALLib [35] and on Box Constrained Quadratic Programs (BoxQPs) [59]. GLOBALLib contains 413 continuous global optimization problems of various sizes and types, such as BoxQPs, problems with complementarity constraints, and general QCQPs. Following [49], we select 160 instances from GLOBALLib having at most 50 variables and that can easily be formulated as (1.1). The conversion of a non-linear expression into a quadratic expression, when possible, is performed by adding new variables and constraints to the problem. Additionally, bounds on the variables are derived using linear programming techniques and these bound are included in the formulation. From these 160 instances in AMPL format, we substitute each bilinear term $x_i x_j$ by the new variable X_{ij} as described for the formulation (1.2). We build two collections of linearized instances in MPS format, one with the original precision on the coefficients and right hand side, and the second with 8-digit precision. In our experiments we used the latter.

As observed in [49], using together the **SDP** and **RLT** relaxations yields stronger bounds than those given by the **RLT** relaxation only for 38 out of 160 GLOBALLib instances. Hence, we focus on these 38 instances to test the effectiveness of the PSD Cuts and their sparse versions.

The BoxQP collection contains 90 instances with a number of variables ranging from 20 to 100. Due to time limit constraints and the number of experiments to run, we consider only instances with a number of variables between 20 to 50, for a total of 51 BoxQP problems.

The converted GLOBALLib and BoxQP instances are available in MPS format from [47].

1.4.2 Effectiveness of each class of cuts

We first compare the effectiveness of the various classes of cuts when used in combination with the standard PSDCUTs. For these tests, at most 1,000 cutting iterations are performed, at most 600 seconds are used, and operations are stopped if tailing off is detected. More precisely, let z_t be the optimal value of the linear relaxation at iteration t. The operations are halted if $t \geq 50$ and $z_t \geq (1 - 0.0001) \cdot z_{t-50}$. A cut purging procedure is used to remove cuts that are not tight at iteration t if the condition $z_t \geq (1 - 0.0001) \cdot z_{t-1}$ is satisfied. On average in each iteration the algorithm generates $\frac{n^2}{2}$ cuts, of which only $\frac{n}{2}$ are are kept by the cut purging procedure and the rest are discarded.

In order to compare two different cutting plane algorithms, we compare the closed gaps values first after a fixed number of iterations, and second at several given times, for all

QCQP instances at avail. Comparisons at fixed iterations indicate the quality of the cuts, irrespective of the time used to generate them. Comparisons at given times are useful if only limited time is available for running the cutting plane algorithms and a good approximation of the PSD cone is sought. The closed gaps obtained at a given point are deemed different only if their difference is at least g% of the initial gap. We report comparisons for $g = 1$ and $g = 5$. Comparisons at one point is possible only if both algorithms reach that point. The number of problems for which this does not happen – because, at a given time, either result was not available or one of the two algorithms had already stopped, or because either algorithm had terminated in fewer iterations – is listed in the "inc." (incomparable) columns in the tables. For the remaining problems, we report the percentage of problems for which one algorithm is better than the other and the percentage of problems were they are tied. Finally, we also report the average improvement in gap closed for the second algorithm over the first algorithm in the column labeled "impr.".

Tests are first performed to decide which combination of the SPARSE1, SPARSE2 and MINOR cuts perform best on average. Based on Tables 1.1 and 1.2 below, we conclude that using MINOR is useful both in terms of iteration and time, and that the algorithm using PSDCUT+SPARSE2+MINOR (abbreviated *S2M* in the remainder) dominates the algorithm using PSDCUT+SPARSE1+MINOR (abbreviated *S1M*) both in terms of iteration and time. Table 1.1 gives the comparison between S1M and S2M at different iterations. S2M dominates clearly S1M in the very first iteration and after 200 iterations, while after the first few iterations S1M also manages to obtain good bounds. Table 1.2 gives the comparison between these two algorithms at different times. For comparisons with $g = 1$, S1M is better than S2M only in at most 2.25% of the problems, while the converse varies between roughly 50% (at early times) and 8% (for late times). For $g = 5$, S2M still dominates S1M in most cases.

Sparse cuts yield better bounds than using solely the standard PSD cuts. The observed improvement is around 3% and 5% respectively for SPARSE1 and SPARSE2. When we are using the MINOR cuts, this value gets to 6% and 8% respectively for each type of sparsification algorithm used. Table 1.3 compares PSDCUT (abbreviated by *S*) with S2M. The table shows that the sparse cuts generated by the sparsification procedures and minor PSD cuts yield better bounds than the standard cutting plane algorithm at fixed iterations. Comparisons performed at fixed times, on the other hand, show that considering the whole set of instances we do not get any improvement in the first 60 to 120 seconds of computation (see Table 1.4). Indeed S2M initially performs worse than the standard cutting plane algorithm, but after 60 to 120 seconds, it produces better bounds on average. In Section 1.6 detailed computational results are given in Tables 1.5 and 1.6 where for each instance we compare the duality gap closed by S and S2M at several iterations and times. In the latter table the instances solved in less than 1 seconds are not shown. The initial duality gap is obtained as in (1.10) as $RLT - OPT$. We then let S2M run with no time limit until the value s obtained does not improve by at least 0.01% over ten consecutive iterations. This value s is an upper bound on the value of the **PSD+RLT** relaxation. The column "bound" in the tables gives the value of $RLT - s$ as a percentage of the gap $RLT - OPT$, i.e. an approximation of the percentage of the gap closed by the **PSD+RLT** relaxation. The columns labeled S and S2M in the tables give the gap closed by the corresponding algorithms at different iterations.

Note that although S2M relies on numerous spectral decomposition computations, most of its running time is spent in generating cuts and reoptimization of the LP. For example, on the BoxQP instances with a time limit of 300 seconds, the average percentage of CPU time spent for obtaining spectral decompositions is below 21 for instances of size 30, below 15 for instances of size 40 and below 7 for instances of size 50.

1.5 Conclusions

We studied linearizations of the PSD cone based on spectral decompositions. Sparsification of eigenvectors corresponding to negative eigenvalues is shown to produce useful cuts in practice, in particular when the minor cuts are used. The goal of capturing most of the strength of a PSD relaxation through linear inequalities is achieved, although tailing off occurs relatively quickly. As an illustration of typical behavior of a PSD solver and our linear outer-approximation scheme, consider the two instances, spar020-100-1 and spar030-060-1, with respectively 20 and 30 variables. We use the SDP solver `SeDuMi` and S2M, keeping track at each iteration of the bound achieved and the time spent. Figure 1.2 and Figure 1.3 compare the bounds obtained by the two solvers at a given time. For the small size instance spar020-100-1, we note that S2M converges to the bound value more than twenty times faster than `SeDuMi`. In the medium size instance spar030-060-1 we note that S2M closes a large gap in the first ten to twenty iterations, and then tailing off occurs. To compute the exact bound, `SeDuMi` requires 408 seconds while S2M requires 2,442 seconds to reach the same precision. Nevertheless, for our purposes, most of the benefits of the PSD constraints are captured in the early iterations.

Two additional improvements are possible. The first one is to use a cut separation procedure for the RLT inequalities, avoiding their inclusion in the initial LP and managing them as other cutting planes. This could potentially speed up the reoptimization of the LP. Another possibility is to use a mix of the S and S2M algorithms, using the former in the early iterations and then switching to the latter.

Figure 1.2: Instance spar020-100-1

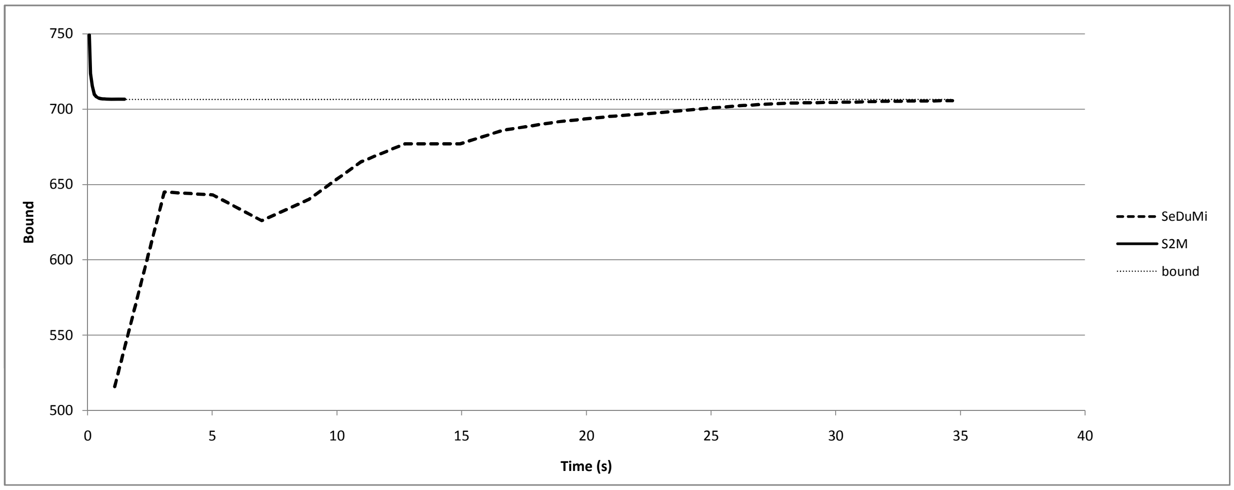


Figure 1.3: Instance spar030-060-1

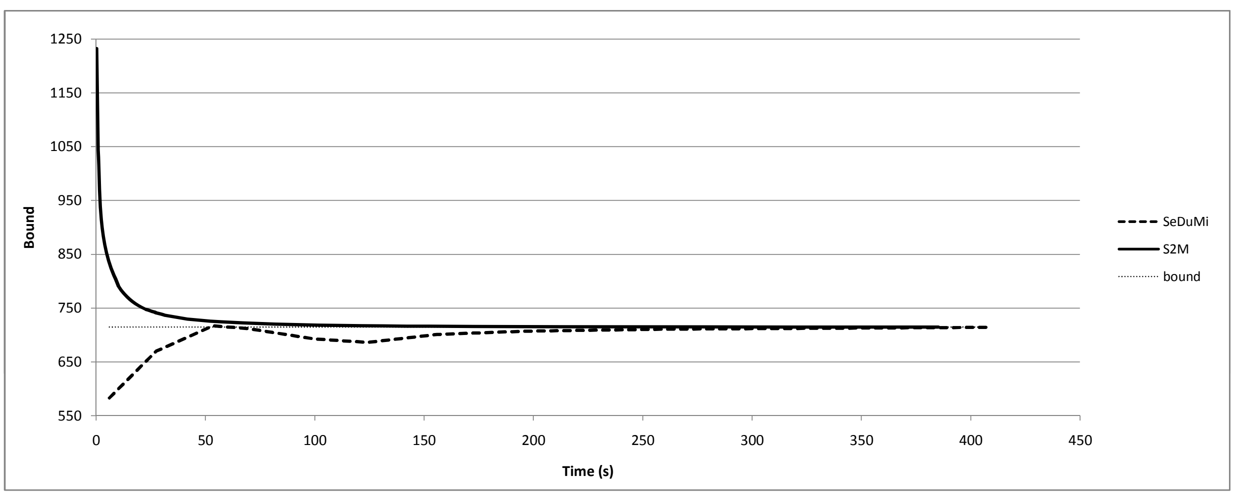


Table 1.1: Comparison of S1M with S2M at several iterations.

	g = 1			g = 5				
Iteration	S1M	S2M	Tie	S1M	S2M	Tie	inc.	impr.
1	7.87	39.33	52.80	1.12	19.1	79.78	0.00	3.21
2	17.98	28.09	53.93	0.00	10.11	89.89	0.00	2.05
3	17.98	19.10	62.92	1.12	7.87	91.01	0.00	1.50
5	12.36	14.61	73.03	3.37	5.62	91.01	0.00	1.77
10	10.11	13.48	76.41	0.00	5.62	94.38	0.00	1.42
15	4.49	13.48	82.03	1.12	6.74	92.14	0.00	1.12
20	1.12	10.11	78.66	1.12	6.74	82.02	10.11	1.02
30	1.12	8.99	79.78	1.12	5.62	83.15	10.11	0.79
50	2.25	6.74	80.90	1.12	4.49	84.28	10.11	0.47
100	0.00	4.49	28.09	0.00	2.25	30.33	67.42	1.88
200	0.00	3.37	15.73	0.00	2.25	16.85	80.90	2.51
300	0.00	2.25	12.36	0.00	2.25	12.36	85.39	3.30
500	0.00	2.25	7.87	0.00	2.25	7.87	89.88	3.85
1000	0.00	2.25	3.37	0.00	2.25	3.37	94.38	7.43

Table 1.2: Comparison of S1M with S2M at several times.

	g = 1			g = 5				
Time	S1M	S2M	Tie	S1M	S2M	Tie	inc.	impr.
0.5	3.37	52.81	12.36	0.00	43.82	24.72	31.46	2.77
1	0.00	51.68	14.61	0.00	40.45	25.84	33.71	4.35
2	0.00	47.19	15.73	0.00	39.33	23.59	37.08	5.89
3	1.12	44.94	14.61	0.00	34.83	25.84	39.33	5.11
5	1.12	43.82	15.73	0.00	38.20	22.47	39.33	6.07
10	1.12	41.58	16.85	0.00	24.72	34.83	40.45	4.97
15	2.25	37.08	16.85	1.12	21.35	33.71	43.82	3.64
20	1.12	35.96	16.85	1.12	17.98	34.83	46.07	3.49
30	1.12	28.09	22.48	1.12	16.86	33.71	48.31	2.99
60	1.12	20.23	28.09	0.00	12.36	37.08	50.56	2.62
120	0.00	15.73	32.58	0.00	10.11	38.20	51.69	1.73
180	0.00	13.49	32.58	0.00	5.62	40.45	53.93	1.19
300	0.00	11.24	31.46	0.00	3.37	39.33	57.30	0.92
600	0.00	7.86	24.72	0.00	0.00	32.58	67.42	0.72

Table 1.3: Comparison of S with S2M at several iterations.

Iteration	g = 1			g = 5				
	S	S2M	Tie	S	S2M	Tie	inc.	impr.
1	0.00	76.40	23.60	0.00	61.80	38.20	0.00	10.47
2	0.00	84.27	15.73	0.00	55.06	44.94	0.00	10.26
3	0.00	83.15	16.85	0.00	48.31	51.69	0.00	10.38
5	0.00	80.90	19.10	0.00	40.45	59.55	0.00	10.09
10	1.12	71.91	26.97	0.00	41.57	58.43	0.00	8.87
15	1.12	60.67	38.21	1.12	35.96	62.92	0.00	7.49
20	1.12	53.93	40.45	1.12	29.21	65.17	4.50	6.22
30	1.12	34.83	53.93	0.00	16.85	73.03	10.12	5.04
50	1.12	25.84	62.92	0.00	13.48	76.40	10.12	3.75
100	1.12	8.99	21.35	0.00	5.62	25.84	68.54	5.57
200	0.00	5.62	8.99	0.00	3.37	11.24	85.39	7.66
300	0.00	3.37	7.87	0.00	3.37	7.87	88.76	8.86
500	0.00	3.37	5.62	0.00	3.37	5.62	91.01	8.72
1000	0.00	2.25	0.00	0.00	2.25	0.00	97.75	26.00

Table 1.4: Comparison of S with S2M at several times.

Time	g = 1			g = 5				
	S	S2M	Tie	S	S2M	Tie	inc.	impr.
0.5	41.57	17.98	5.62	41.57	17.98	5.62	34.83	-9.42
1	41.57	14.61	5.62	39.33	13.48	8.99	38.20	-8.66
2	42.70	10.11	6.74	29.21	8.99	21.35	40.45	-8.73
3	41.57	8.99	8.99	31.46	6.74	21.35	40.45	-8.78
5	35.96	7.87	15.72	33.71	5.62	20.22	40.45	-7.87
10	34.84	7.87	13.48	30.34	4.50	21.35	43.81	-5.95
15	37.07	5.62	11.24	22.47	2.25	29.21	46.07	-5.48
20	37.07	5.62	8.99	17.98	1.12	32.58	48.32	-4.99
30	30.34	5.62	15.72	11.24	1.12	39.32	48.32	-3.9
60	11.24	12.36	25.84	11.24	2.25	35.95	50.56	-1.15
120	8.99	12.36	24.72	2.25	2.25	41.57	53.93	0.48
180	2.25	14.61	29.21	0.00	4.50	41.57	53.93	1.09
300	0.00	15.73	26.97	0.00	6.74	35.96	57.30	1.60
600	0.00	14.61	13.48	0.00	5.62	22.47	71.91	2.73

1.6 Appendix

Table 1.5: Duality gap closed after 2, 10 and 50 iterations.

Instance	$\lvert x \rvert$	$\lvert y \rvert$	bound	iter. 2		iter. 10		iter. 50	
				S	S2M	S	S2M	S	S2M
circle	3	0	45.79	0.00	0.00	10.97	41.31	45.77	45.79
dispatch	3	1	100.00	25.59	27.92	37.25	35.76	95.90	92.17
ex2_1_10	20	0	22.05	3.93	8.65	15.93	21.05	22.05	22.05
ex3_1_2	5	0	49.75	49.75	49.75	49.75	49.75	49.75	49.75
ex4_1_1	3	0	100.00	99.81	99.84	100.00	100.00	100.00	100.00
ex4_1_3	3	0	56.40	0.00	0.00	51.19	51.19	56.40	56.40
ex4_1_4	3	0	100.00	22.33	42.78	98.98	99.98	100.00	100.00
ex4_1_6	3	0	100.00	69.44	69.87	92.62	99.94	100.00	100.00
ex4_1_7	3	0	100.00	18.00	48.17	96.86	99.90	100.00	100.00
ex4_1_8	3	0	100.00	56.90	81.93	99.76	99.93	100.00	100.00
ex8_1_4	4	0	100.00	94.91	95.19	99.98	100.00	100.00	100.00
ex8_1_5	5	0	68.26	32.32	39.17	59.01	66.76	68.00	68.25
ex8_1_7	9	0	77.43	3.04	33.75	33.13	53.44	64.03	75.38
ex8_4_1	21	1	91.81	4.45	21.80	18.60	45.08	38.07	69.83
ex9_2_1	10	0	54.52	0.00	42.55	0.01	50.13	0.01	51.90
ex9_2_2	10	0	70.37	0.00	14.08	2.34	51.97	7.12	69.41
ex9_2_4	6	2	99.87	0.00	24.84	25.24	99.85	86.37	99.87
ex9_2_6	16	0	99.88	3.50	99.42	23.09	99.86	62.32	99.88
ex9_2_7	10	0	42.30	0.00	4.59	0.00	27.34	3.14	34.91
himmel11	5	4	49.75	49.75	49.75	49.75	49.75	49.75	49.75
hydro	12	19	52.06	0.00	20.87	21.95	29.03	26.04	31.39
mathopt1	4	0	100.00	95.76	100.00	99.96	100.00	100.00	100.00
mathopt2	3	0	100.00	99.84	99.93	100.00	100.00	100.00	100.00
meanvar	7	1	100.00	0.00	0.00	78.35	95.84	100.00	100.00
nemhaus	5	0	53.97	26.00	26.41	48.49	50.16	53.87	53.96
prob06	2	0	100.00	90.61	92.39	98.39	98.39	98.39	98.39
prob09	3	1	100.00	0.00	99.00	61.14	99.96	99.64	100.00
process	9	3	8.00	0.00	4.25	0.00	4.98	0.00	5.73
qp1	50	0	100.00	79.59	89.09	93.89	99.77	98.93	100.00
qp2	50	0	100.00	55.94	70.99	82.42	93.92	93.04	99.35
rbrock	3	0	100.00	97.48	100.00	99.96	100.00	100.00	100.00
st_e10	3	1	100.00	56.90	81.93	99.76	99.93	100.00	100.00
st_e18	2	0	100.00	0.00	0.00	98.72	98.72	100.00	100.00
st_e19	3	1	93.51	5.14	15.93	29.97	60.10	93.40	93.50
st_e25	4	0	87.55	55.80	55.80	87.02	87.01	87.23	87.23
st_e28	5	4	49.75	49.75	49.75	49.75	49.75	49.75	49.75

Continued on Next Page...

Table 1.5 – Continued

Instance	$\|x\|$	$\|y\|$	bound	iter. 2 S	iter. 2 S2M	iter. 10 S	iter. 10 S2M	iter. 50 S	iter. 50 S2M
st_iqpbk1	8	0	97.99	71.99	76.69	97.20	97.95	97.99	97.99
st_iqpbk2	8	0	97.93	70.55	75.16	94.93	97.52	97.93	97.93
spar020-100-1	20	0	100.00	91.15	94.64	99.77	99.99	100.00	100.00
spar020-100-2	20	0	99.70	90.12	92.64	98.17	99.32	99.66	99.69
spar020-100-3	20	0	100.00	96.96	98.51	100.00	100.00	100.00	100.00
spar030-060-1	30	0	98.87	43.53	53.64	79.61	87.39	93.90	97.14
spar030-060-2	30	0	100.00	80.74	89.73	99.89	100.00	100.00	100.00
spar030-060-3	30	0	99.40	67.43	71.94	91.48	95.68	98.75	99.26
spar030-070-1	30	0	97.99	49.05	54.94	76.54	86.51	91.15	95.68
spar030-070-2	30	0	100.00	81.19	85.82	99.26	99.99	100.00	100.00
spar030-070-3	30	0	99.98	85.97	87.43	98.44	99.52	99.92	99.97
spar030-080-1	30	0	98.99	64.44	70.99	87.32	92.11	96.23	98.01
spar030-080-2	30	0	100.00	92.78	95.45	100.00	100.00	100.00	100.00
spar030-080-3	30	0	100.00	92.71	94.18	99.99	100.00	100.00	100.00
spar030-090-1	30	0	100.00	80.37	86.35	97.27	99.30	100.00	100.00
spar030-090-2	30	0	100.00	86.09	89.26	98.13	99.65	100.00	100.00
spar030-090-3	30	0	100.00	90.65	91.56	99.97	100.00	100.00	100.00
spar030-100-1	30	0	100.00	77.28	83.25	95.20	98.30	99.85	100.00
spar030-100-2	30	0	99.96	76.78	81.65	93.44	96.84	98.70	99.72
spar030-100-3	30	0	99.85	86.82	88.74	97.45	98.75	99.75	99.83
spar040-030-1	40	0	100.00	25.60	41.96	73.59	84.72	99.13	100.00
spar040-030-2	40	0	100.00	30.93	53.39	79.34	95.62	99.46	100.00
spar040-030-3	40	0	100.00	9.21	31.38	66.46	86.62	98.53	100.00
spar040-040-1	40	0	96.74	23.62	29.03	63.04	75.93	85.93	93.29
spar040-040-2	40	0	100.00	33.17	48.87	89.08	97.94	100.00	100.00
spar040-040-3	40	0	99.18	21.77	30.31	70.44	80.96	91.37	96.69
spar040-050-1	40	0	99.42	35.62	44.87	73.11	84.05	92.81	97.21
spar040-050-2	40	0	99.48	36.79	47.68	82.38	91.27	97.26	98.93
spar040-050-3	40	0	100.00	41.91	51.72	84.04	90.70	96.88	99.34
spar040-060-1	40	0	98.09	46.22	52.89	81.65	87.28	92.39	95.97
spar040-060-2	40	0	100.00	63.02	72.87	94.09	97.66	99.78	100.00
spar040-060-3	40	0	100.00	78.09	87.91	99.30	99.99	100.00	100.00
spar040-070-1	40	0	100.00	64.02	71.33	93.92	97.35	99.77	100.00
spar040-070-2	40	0	100.00	67.49	76.78	95.12	97.97	99.97	100.00
spar040-070-3	40	0	100.00	70.13	79.43	95.65	97.99	99.75	100.00
spar040-080-1	40	0	100.00	63.06	69.40	91.09	95.44	99.00	99.97
spar040-080-2	40	0	100.00	71.42	79.77	94.98	97.62	99.92	100.00
spar040-080-3	40	0	99.99	83.93	88.65	97.76	98.86	99.81	99.95
spar040-090-1	40	0	100.00	75.73	79.96	95.34	97.43	99.46	99.91
spar040-090-2	40	0	99.97	76.39	80.97	95.16	96.72	99.20	99.81
spar040-090-3	40	0	100.00	84.90	87.04	98.33	99.52	100.00	100.00
spar040-100-1	40	0	100.00	87.64	90.43	98.27	99.35	99.98	100.00
spar040-100-2	40	0	99.87	79.78	83.02	94.58	96.76	98.74	99.50
spar040-100-3	40	0	98.70	72.69	78.31	90.83	93.03	95.84	97.36

Continued on Next Page...

Table 1.5 – Continued

Instance	$\|x\|$	$\|y\|$	bound	iter. 2		iter. 10		iter. 50	
				S	S2M	S	S2M	S	S2M
spar050-030-1	50	0	100.00	3.11	17.60	58.23	79.98	-	-
spar050-030-2	50	0	99.27	1.35	16.67	51.11	70.58	-	-
spar050-030-3	50	0	99.29	0.08	13.63	50.19	67.46	-	-
spar050-040-1	50	0	100.00	23.13	30.86	72.10	81.73	-	-
spar050-040-2	50	0	99.39	21.89	34.45	71.24	81.63	-	-
spar050-040-3	50	0	100.00	27.18	37.42	83.96	91.70	-	-
spar050-050-1	50	0	93.02	25.24	33.77	61.42	68.75	-	-
spar050-050-2	50	0	98.74	32.10	41.26	77.48	83.48	-	-
spar050-050-3	50	0	98.84	38.57	44.67	80.97	85.36	-	-
Average	-	-	-	48.75	59.00	75.53	84.39	85.85	89.60

Table 1.6: Duality gap closed after 1, 60, 150, 200 and 600 seconds.

		1 s		60 s		180 s		300 s		600 s	
Instance	bound	S	S2M	S	S2M	S	S2M	S	S2M	S	S2M
ex4_1_4	100.00	-	100.00	-	-	-	-	-	-	-	-
ex8_1_4	100.00	-	100.00	-	-	-	-	-	-	-	-
ex8_1_7	77.43	77.43	77.37	-	-	-	-	-	-	-	-
ex8_4_1	91.81	28.14	36.24	61.60	90.43	-	-	-	-	-	-
ex9_2_2	70.37	-	70.35	-	-	-	-	-	-	-	-
ex9_2_6	99.88	96.28	-	-	-	-	-	-	-	-	-
hydro	52.06	26.43	31.46	-	-	-	-	-	-	-	-
mathopt2	100.00	-	100.00	-	-	-	-	-	-	-	-
process	8.00	-	7.66	-	-	-	-	-	-	-	-
qp1	100.00	79.99	80.28	98.22	99.52	99.73	99.96	99.92	99.98	99.99	100.00
qp2	100.00	55.82	55.27	91.74	95.56	95.86	98.69	97.41	99.66	98.80	100.00
spar020-100-1	100.00	100.00	100.00	-	-	-	-	-	-	-	-
spar020-100-2	99.70	99.67	99.61	-	-	-	-	-	-	-	-
spar020-100-3	100.00	-	100.00	-	-	-	-	-	-	-	-
spar030-060-1	98.87	69.98	58.72	96.53	97.61	98.45	98.70	98.68	98.82	-	-
spar030-060-2	100.00	96.52	91.05	-	-	-	-	-	-	-	-
spar030-060-3	99.40	82.99	76.15	99.27	99.32	99.38	99.39	99.39	99.40	99.40	99.40
spar030-070-1	97.99	69.81	60.36	94.50	96.38	97.29	97.73	97.70	97.91	-	97.98
spar030-070-2	100.00	96.05	87.93	-	-	-	-	-	-	-	-
spar030-070-3	99.98	96.26	90.42	99.98	99.98	99.98	99.98	-	99.98	-	-
spar030-080-1	98.99	83.36	74.42	97.80	98.11	98.74	98.88	98.89	98.96	-	98.99
spar030-080-2	100.00	99.83	96.70	-	-	-	-	-	-	-	-
spar030-080-3	100.00	99.88	95.87	-	-	-	-	-	-	-	-
spar030-090-1	100.00	92.86	87.69	-	-	-	-	-	-	-	-
spar030-090-2	100.00	93.80	88.46	-	100.00	-	-	-	-	-	-
spar030-090-3	100.00	97.78	91.35	-	-	-	-	-	-	-	-
spar030-100-1	100.00	91.04	84.34	100.00	100.00	-	-	-	-	-	-
spar030-100-2	99.96	90.21	83.14	99.56	99.75	99.91	99.95	99.95	99.96	-	99.96
spar030-100-3	99.85	94.26	89.55	99.84	99.84	99.85	99.85	99.85	99.85	99.85	99.85
spar040-030-1	100.00	28.97	40.51	89.30	84.19	99.06	99.98	99.98	100.00	-	100.00
spar040-030-2	100.00	31.97	48.01	94.01	96.39	99.58	99.98	99.99	100.00	-	-
spar040-030-3	100.00	9.20	27.59	81.66	85.43	97.25	99.86	99.81	100.00	100.00	-
spar040-040-1	96.74	19.38	22.90	70.35	75.45	80.73	88.63	85.34	92.29	90.79	94.74
spar040-040-2	100.00	24.51	29.87	98.63	98.60	100.00	100.00	-	-	-	-
spar040-040-3	99.18	20.88	21.31	78.28	79.31	86.02	91.22	89.52	95.04	94.07	97.71
spar040-050-1	99.42	28.96	21.27	80.18	84.01	88.70	94.62	92.75	96.71	96.53	98.32
spar040-050-2	99.48	29.52	16.91	91.33	91.42	97.01	97.97	98.26	98.87	-	99.31
spar040-050-3	100.00	28.67	19.81	90.03	90.72	95.68	97.51	97.49	99.08	98.92	99.89
spar040-060-1	98.09	37.16	17.10	86.26	87.13	90.18	93.50	92.25	95.32	95.05	96.84
spar040-060-2	100.00	39.57	22.83	98.09	98.22	99.90	99.96	100.00	100.00	100.00	-
spar040-060-3	100.00	52.41	30.57	100.00	99.99	-	-	-	-	-	-

Continued on Next Page...

Table 1.6 – Continued

Instance	bound	1 s		60 s		180 s		300 s		600 s	
		S	S2M	S	S2M	S	S2M	S	S2M	S	S2M
spar040-070-1	100.00	50.01	21.79	97.74	97.78	99.80	99.87	99.97	99.99	100.00	100.00
spar040-070-2	100.00	47.57	25.19	98.81	98.46	99.99	99.99	100.00	100.00	-	-
spar040-070-3	100.00	47.22	21.95	98.96	98.70	99.88	99.92	99.98	100.00	100.00	100.00
spar040-080-1	100.00	51.66	28.00	95.13	95.38	98.29	99.05	99.09	99.74	99.77	99.99
spar040-080-2	100.00	52.24	25.94	98.71	98.31	99.95	99.97	100.00	100.00	-	-
spar040-080-3	99.99	56.05	26.98	99.54	99.25	99.89	99.88	99.94	99.95	99.97	99.98
spar040-090-1	100.00	59.71	28.17	98.10	97.86	99.43	99.61	99.70	99.86	99.90	99.99
spar040-090-2	99.97	59.14	29.82	97.83	97.70	99.34	99.58	99.68	99.81	99.86	99.93
spar040-090-3	100.00	63.07	34.62	99.94	99.85	100.00	100.00	-	-	-	-
spar040-100-1	100.00	69.47	28.24	99.66	99.47	99.99	99.99	100.00	100.00	-	-
spar040-100-2	99.87	65.27	26.07	97.34	96.87	98.60	98.98	99.02	99.39	99.44	99.69
spar040-100-3	98.70	61.40	29.61	93.01	93.17	94.91	96.02	95.81	97.00	96.84	97.77
spar050-030-1	100.00	0.37	3.63	54.46	37.52	70.10	73.34	76.87	84.75	86.23	96.33
spar050-030-2	99.27	0.08	2.79	44.68	38.62	59.58	64.94	67.79	74.98	77.02	86.58
spar050-030-3	99.29	0.00	2.75	44.32	32.31	57.13	59.07	62.54	68.99	71.18	82.86
spar050-040-1	100.00	3.76	1.77	69.97	56.87	77.15	78.30	80.31	84.30	84.90	91.79
spar050-040-2	99.39	2.08	2.84	68.64	58.47	77.72	77.61	81.54	83.63	86.40	90.94
spar050-040-3	100.00	1.76	2.31	79.44	65.71	89.73	87.74	92.67	93.00	95.99	97.69
spar050-050-1	93.02	4.91	1.84	60.64	53.28	65.52	66.42	66.81	70.38	68.45	74.76
spar050-050-2	98.74	6.18	3.39	76.56	68.33	82.34	82.21	84.94	86.52	-	91.34
spar050-050-3	98.84	6.12	2.82	79.38	69.23	84.95	83.23	86.99	86.98	89.77	91.57
Average	-	51.45	42.96	87.50	86.38	92.14	93.22	93.18	94.77	93.16	95.86

Chapter 2

Cuts from multiple rows of the simplex tableau

2.1 Introduction

Since 2007 in the context of Integer Linear Programming a new class of valid inequalities derived from multiple rows of the simplex tableau are being investigated. Andersen, Louveaux, Weismantel and Wolsey study in [2] the geometrical structure of 2 rows of the simplex tableau corresponding to basic integer constrained variables and they show that there exist valid cuts which can only be derived using information from the 2 rows simultaneously. Several papers further investigate the family of cuts from multiple rows. Cornuejols and Margot in [26] present a complete characterization of the facets of a mixed integer linear program with two integer variables and two constraints. Borozan and Cornuejols [20] investigate the relation between minimal valid inequalities and maximal lattice free convex sets. In [13] Basu et al. study the strength of cuts derived from 2 rows using the Goeamans framework. Dey and Wolsey in [31] study lifting and strengthening this type of cuts using integrality of non-basic variables.

The work in this chapter develops and discusses the approach presented by Balas in 2009 in the meetings [7, 8].

As shown in [2], multiple row cuts belong to the class of Intersection Cuts [5] which are Disjunctive Cuts [4]. In this chapter we present a different perspective of the structure given by multiple rows of the simplex tableau and we show how Disjunctive Programming can be used to generate valid cuts in this context.

In the rest of this Section we review the tools offered by Disjunctive Programming and the connection with Intersection cuts. In Section 2.2 we introduce the concept of Disjunctive Hull associated to 2 rows of a simplex tableau and we examine the relation between the Disjunctive Hull and the Integer Hull. We then consider the case of 2 rows of the simplex tableau in Section 2.3 and we focus on the 0-1 case in Section 2.4. We discuss efficient procedures based on the Lift-and-Project framework to generate our class of cuts and we illustrate them computationally in Section 2.5. Cut strengthening techniques that exploit the integrality of the non basic variables are also presented.

2.1.1 Intersection cuts and disjunctive programming

Suppose a Mixed Integer Program is given in the form of q rows of the simplex tableau

$$x = \bar{x} + \sum_{j \in J} r_j s_j, \quad x \in \mathbb{Z}_+^q,\ s \in \mathbb{R}_+^n \tag{2.1.1}$$

where $\bar{x}$ is a basic feasible solution to LP, the linear programming relaxation of a MIP, and we are interested in generating an inequality that cuts off $\bar{x}$ but no feasible integer point.

Theorem 2.1.1. *Balas [5]. Let $T \subseteq \mathbb{R}^q$ be a closed convex set such that $\bar{x} \in \operatorname{int} T$ and $\operatorname{int} T$ contains no feasible integer point. For $j \in J$, let $s_j^* := \max\{s_j : \bar{x} + r_j s_j \in T\}$. Then the inequality $\alpha s \geq 1$, where $\alpha_j = \frac{1}{s_j^*}$, $j \in J$, cuts off $\bar{x}$ but no feasible integer point.*

The inequality $\alpha s \geq 1$ is known as an *intersection cut*. Theorem 2.1.1 is illustrated by

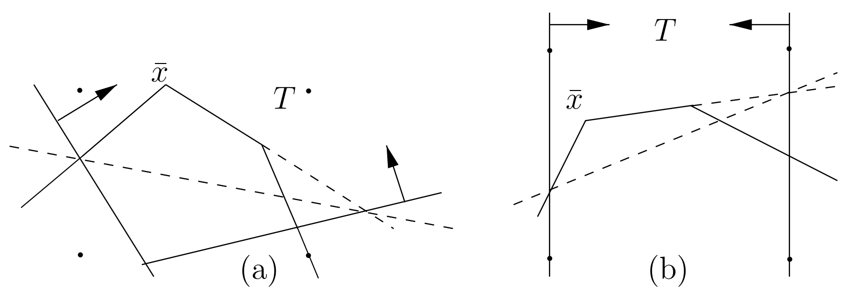


Figure 2.1: Two intersection cuts

Figure 2.1. In both cases (a) and (b) the convex set T consists of the intersection of two halfspaces, but in (b) the two halfspaces are defined by hyperplanes parallel to one of the coordinate axes, and so their intersection defines an infinite strip. The intersection cut from this latter set T is the Gomory Mixed Integer cut (GMI).

This particular class of intersection cuts, the GMI cuts, has played a crucial role in making mixed integer programs practically solvable. These cuts are derived from a convex set of the form $\lfloor \bar{x}_i \rfloor \leq x_i \leq \lceil \bar{x}_i \rceil$, where $x_i = \bar{x}_i + \sum_{j \in J} r_j^i s_j$ is one of the rows of an optimal simplex tableau. More generally, cuts obtained from a convex set of the form $\pi_0 \leq \pi x \leq \pi_0 + 1$, where (π, π_0) is an integer vector, are known in the literature as split cuts. It is then natural to ask the question whether intersection cuts derived simultaneously from several rows of a simplex tableau have some properties that distinguish them from split cuts [24]. It was this question that has led to the investigation of intersection cuts from maximal lattice-free convex sets by [2, 26] and others.

We propose a different approach to the same problem, which promises some computational advantages. The approach is that of Disjunctive Programming, a natural outgrowth of the study of intersection cuts. To see the connection, consider an intersection cut from a polyhedral set with the required properties, of the form $T := \{x : d^i x \leq d_0^i,\ i = 1, \ldots, m\}$. Clearly, the requirement that $\operatorname{int} T$ should contain no feasible integer point, can be rephrased as the requirement that every feasible integer point should satisfy at least one of the weak complements of the inequalities defining T, i.e. should satisfy the disjunction

$$\bigvee_{i=1}^{m} (d^i x \geq d_0^i). \tag{2.1.2}$$

Therefore an intersection cut from T can be viewed as a disjunctive cut from (2.1.2). While these two cuts are the same, the disjunctive point of view opens up new perspectives. Thus, suppose that in addition to (2.1.2), all feasible solutions have to satisfy the inequalities $Ax \geq b$. Then one way to proceed is to generate all valid cutting planes from (2.1.2) and append these to $Ax \geq b$. The resulting system will be

$$P := \left\{ x \in \mathbb{R}^n : (Ax \geq b) \cap \ \text{conv} \left(\bigvee_{i=1}^{m} \left(d^i x \geq d_0^i \right) \right) \right\}.$$

But another way to proceed is to introduce $Ax \geq b$ into each term of the disjunction (2.1.2), i.e. replace (2.1.2) with

$$\bigvee_{i=1}^{m} \begin{pmatrix} Ax \geq b \\ d^i x \geq d_0^i \end{pmatrix}, \tag{2.1.3}$$

and take the convex hull of this union of polyhedra:

$$Q := \ \text{conv} \left(\bigvee_{i=1}^{m} \begin{pmatrix} Ax \geq b \\ d^i x \geq d_0^i \end{pmatrix} \right)$$

Now it is not hard to see that $Q \subseteq P$, and in fact Q is in most cases a much tighter constraint set than P. We illustrate the difference on a 2-term disjunction. Given an arbitrary Mixed Integer Program, let (π, π_0) be an integer vector with a component π_j for every integer-constrained variable. Then the disjunctive cut derived from

$$\pi x \leq \pi_0 \ \vee \ \pi x \geq \pi_0 + 1 \tag{2.1.4}$$

is equivalent to the intersection cut derived from the convex set

$$\pi_0 \leq \pi x \leq \pi_0 + 1,$$

illustrated in Figure 2.1. On the other hand, the disjunction

$$\begin{pmatrix} Ax & \geq & b \\ \pi x & \leq & \pi_0 \end{pmatrix} \quad \vee \quad \begin{pmatrix} Ax & \geq & b \\ \pi x & \geq & \pi_0 + 1 \end{pmatrix} \tag{2.1.5}$$

gives rise to an entire family of cuts, whose members are determined by the multipliers u, v associated with $Ax \geq b$ in the two terms of this more general disjunction

$$(\pi - uA)x \leq \pi_0 - ub \ \vee \ (\pi + vA)x \geq \pi_0 + vb + 1 \tag{2.1.6}$$

Cuts derived from a disjunction of the form (2.1.4) are called split cuts, a term that reflects the fact that (2.1.4) splits the space into two disjoint half-spaces. Cook, Kannan and Schrijver [24] who coined this term also extended it to the much larger family of cuts derived from disjunctions of the form (2.1.6). Since we sometimes need to distinguish between these two classes, we will call the first one *pure split cuts*, and the second one *composite split cuts*.

Disjunctive sets of the form (2.1.3) or (2.1.5) represent unions of polyhedra, and the study of optimization over unions of polyhedra is known as Disjunctive Programming. Its two basic results are a compact representation of the convex hull of a union of polyhedra in a higher dimensional space, and the sequential convexifiability of facial disjunctive sets [6, 4]. The application of disjunctive programming to mixed 0-1 programs has become known as the lift-and-project method [9]. Here we apply this approach to the study of intersection cuts from multiple rows of the simplex tableau.

2.2 Integer and Disjunctive Hulls

Consider again a system defined by q rows of the simplex tableau, this time without the integrality constraints:

$$x = f + \sum_{j \in J} r_j s_j,\ s_j \geq 0,\ j \in J, \tag{2.2.1}$$

where $f,\ r_j \in \mathbb{R}^q,\ j \in J := \{1, \ldots, n\}$, and assume $0 < f_i < 1,\ i \in Q := \{1, \ldots, q\}$. This assumption can be made without loss of generality since we replace $x'_i = x_i - \lfloor f_i \rfloor$ and $f'_i = f_i - \lfloor f_i \rfloor,\ i \in \{1, 2\}$, and we have that $x'_i, f'_i,\ i \in Q$ satisfy the assumption. The set

$$P_L := \{(x, s) \in \mathbb{R}^q \times \mathbb{R}^n : (x, s) \text{ satisfies } (2.2.1)\}$$

is the polyhedral cone with apex at $(x, s) = (f, 0)$ defined by the constraints that are tight for this particular basic solution. Imposing the integrality constraints on the basic components we get the mixed integer set

$$P_I := \{(x, s) \in P_L : x_i \text{ integer}, i \in Q\},$$

obtained from the original mixed integer feasible set by removing the remaining constraints of the latter. Its convex hull, $\operatorname{conv} P_I$, is Gomory's corner polyhedron [37, 38], or the *Integer Hull* of the MIP over the cone P_L. The main objective of the papers mentioned in the introduction was to study the structure of P_I for small q, with a view of characterizing the facets of $\operatorname{conv} P_I$ and minimal valid inequalities for P_I.

Without loss of generality we can assume $0 \leq f_i \leq 1, i \in Q$ in (2.2.1). Consider now the following disjunctive relaxation of P_I, obtained by replacing the integrality constraints on x_i with the simple disjunctions $x_i \leq 0 \ \vee \ x_i \geq 1,\ i \in Q$:

$$P_D := \{(x, s) \in P_L : x_i \leq 0 \ \vee \ x_i \geq 1,\ i \in Q\}.$$

Like P_I, P_D is a nonconvex set. Its convex hull, $\operatorname{conv} P_D$, which we call the *Simple Disjunctive Hull*, is a weaker relaxation of P_I than $\operatorname{conv} P_I$, i.e. $\operatorname{conv} P_D \supseteq \operatorname{conv} P_I$, but it is easier to handle, since it is the convex hull of the union of 2^q polyhedra. Thus one can apply disjunctive programming and lift-and-project techniques to generate facets of $\operatorname{conv} P_D$ at a computational cost that for small q seems acceptable. In this context, the crucial question is of course, how much weaker is the relaxation $\operatorname{conv} P_D$ than $\operatorname{conv} P_I$? We will pose this question in a more specific form that will enable us to give it a practically useful answer: When is it that a facet defining inequality for $\operatorname{conv} P_D$ is also facet defining for $\operatorname{conv} P_I$? In other words, which facets of the simple Disjunctive Hull are also facets of the Integer Hull? Before addressing this question, however, we will take a side-step, by introducing a third kind of hull. If we strengthen the disjunctive relaxation of P_I by replacing the inequalities in the disjunctions $x_i \leq 0 \ \vee \ x_i \geq 1$, $i \in Q$, with equations, we get the set

$$P_D^{=} := \{(x, s) \in P_L : x_i = 0 \ \vee \ x_i = 1, \ i \in Q\}, \tag{2.2.2}$$

whose convex hull, $\operatorname{conv} P_D^{=}$, we call *the 0-1 Disjunctive Hull*. For a general mixed integer program, the 0-1 Disjunctive Hull is not a valid relaxation, in that it may cut off nonbinary feasible integer points. Indeed, we have

$$\operatorname{conv} P_D \supseteq \operatorname{conv} P_I \supseteq \operatorname{conv} P_D^{=},$$

where both inclusions are strict and are valid in the context of Mixed Integer 0-1 programs only, since all the non 0-1 integer points that it cuts off are infeasible. Hence $\operatorname{conv} P_D^{=}$ is equivalent to the convex hull of $P_I \cap \{(x, 0) : 0 \leq x_i \leq 1, \ i \in Q\}$, or the Integer Hull of P_I reinforced with the bounds on the x_i. Furthermore, as we will see later on, finding facets of $\operatorname{conv} P_D^{=}$ requires roughly the same computational effort as finding facets of $\operatorname{conv} P_D$.

The upshot of this is that for the important class of Mixed Integer 0-1 Programs, all facet defining inequalities of $\operatorname{conv} P_D^{=}$ are facet defining for the Integer Hull. Furthermore, from the sequential convexification theorem of Disjunctive Programming, all such inequalities are of split rank $\leq q$, i.e. they can be obtained by applying a split cut generating procedure at most q times recursively. This is an important fact which should be kept in mind when comparing the strength of intersection cuts from q rows with that of split cuts.

2.2.1 Properties of the Simple Disjunctive Hull

The set P_D defined in Section 2.2 is the collection of those points $(x, s) \in \mathbb{R}^q \times \mathbb{R}^n$ satisfying (2.2.1) and $x_i \leq 0 \ \vee \ x_i \geq 1$, $i \in Q$. Put in disjunctive normal form, this last constraint set becomes

$$\begin{pmatrix} x_1 \leq 0 \\ x_2 \leq 0 \\ \vdots \\ x_q \leq 0 \end{pmatrix} \vee \begin{pmatrix} x_1 \geq 1 \\ x_2 \leq 0 \\ \vdots \\ x_q \leq 0 \end{pmatrix} \vee \cdots \vee \begin{pmatrix} x_1 \geq 1 \\ x_2 \geq 1 \\ \vdots \\ x_q \geq 1 \end{pmatrix} \tag{2.2.3}$$

Each term of (2.2.3) defines an orthant-cone with apex at a vertex of the q-dimensional unit cube. These 2^q orthant-cones are illustrated for $q = 2$ in Figure 2.2.

Using (2.2.1) to eliminate the x-components and denoting by r^i the i-th row of the $q \times n$

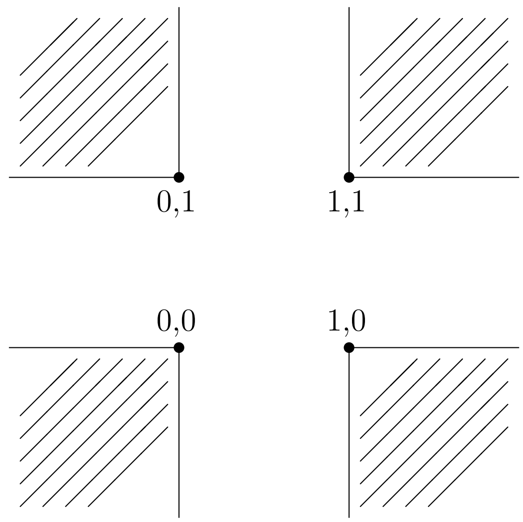


Figure 2.2: Orthant-cones for the case $q = 2$

matrix $R = (r_j)_{j=1}^n$, (2.2.3) can be represented in $\mathbb{R}^n$ as $s \geq 0$ and

$$\begin{pmatrix} -r^1 s & \geq & f_1 \\ -r^2 s & \geq & f_2 \\ & \vdots & \\ -r^q s & \geq & f_q \end{pmatrix} \vee \begin{pmatrix} r^1 s & \geq & 1 - f_1 \\ -r^2 s & \geq & f_2 \\ & \vdots & \\ -r^q s & \geq & f_q \end{pmatrix} \vee \cdots \vee \begin{pmatrix} r^1 s & \geq & 1 - f_1 \\ r^2 s & \geq & 1 - f_2 \\ & \vdots & \\ r^q s & \geq & 1 - f_q \end{pmatrix} \tag{2.2.4}$$

If $P_i^{(n)} \subseteq \mathbb{R}^n$ denotes the polyhedron defined by the i-th term of this disjunction plus the constraints $s \geq 0$, then P_D can be defined in n-space as $P_D^{(n)} = \cup_{i=1}^t P_i^{(n)}$ where $t = 2^q$. Furthermore, we have the following:

Theorem 2.2.1. conv $P_D^{(n)}$ *is the set of those* $s \in \mathbb{R}^n$ *satisfying* $s \geq 0$ *and all the inequalities* $\alpha s \geq 1$ *whose coefficient vectors* $\alpha \in \mathbb{R}^n$ *satisfy the system*

$$\begin{array}{llllll}
\alpha + r^1 u_{11} + \cdots + r^q u_{1q} & & & & \geq 0 \\
\alpha & & - r^1 u_{21} + \cdots + r^q u_{2q} & & & \geq 0 \\
\vdots & & & \ddots & & \vdots \\
\alpha & & & & - r^1 u_{t1} - \cdots - r^q u_{tq} & \geq 0 \\
& f_1 u_{11} + \cdots + f_q u_{1q} & & & & \geq 1 \\
& & (1-f_1) u_{21} + \cdots + f_2 u_{2q} & & & \geq 1 \\
& & & \ddots & & \vdots \\
& & & & (1-f_1) u_{t1} + \cdots + (1-f_2) u_{tq} & \geq 1
\end{array} \tag{2.2.5}$$

for some $u_{ik} \geq 0$, $i = 1, \ldots, t = 2^q, k = 1, \ldots, q$.

Proof. Applying the basic theorem of Disjunctive Programming to conv $P_D^{(n)}$ we introduce auxiliary variables $s^i \in \mathbb{R}^n$, $z_i \in \mathbb{R}$, $i = 1, \ldots, t = 2^q$, and obtain the higher-dimensional representation

$$
\begin{array}{rrrrrrrrrrl}
s & -s^1 & -s^2 & \ldots & -s^t & & & & & = & 0 \\
& -r^1 s^1 & & & & -f_1 z_1 & & & & \geq & 0 \\
& -r^2 s^1 & & & & -f_2 z_1 & & & & \geq & 0 \\
& \vdots & & & & \vdots & & & & \vdots & \\
& -r^q s^1 & & & & -f_q z_1 & & & & \geq & 0 \\
& & r^1 s^2 & & & & -(1-f_1) z_2 & & & \geq & 0 \\
& & -r^2 s^2 & & & & -f_2 z_2 & & & \geq & 0 \\
& & \vdots & & & & \vdots & & & \vdots & \\
& & -r^q s^2 & & & & -f_q z_2 & & & \geq & 0 \\
& & & \ddots & & & & \ddots & & \vdots & \\
& & & & -r^1 s^t & & & & -(1-f_1) z_t & \geq & 0 \\
& & & & -r^2 s^t & & & & -(1-f_2) z_t & \geq & 0 \\
& & & & \vdots & & & & \vdots & \vdots & \\
& & & & -r^q s^t & & & & -(1-f_q) z_t & \geq & 0 \\
& & & & & z_1 & +z_2 & +\cdots & +z_t & = & 1 \\
\multicolumn{11}{c}{s^i \geq 0,\ i = 1, \ldots, t; \quad z_i \geq 0,\ i = 1, \ldots, t}
\end{array}
\tag{2.2.6}
$$

Projecting this system onto the s-space with multipliers α; $u_{11}, \ldots, u_{1q}$; $u_{21}, \ldots, u_{2q}$; $\ldots$; $u_{t1}, \ldots, u_{tq}$, we obtain

$$
\begin{array}{lllllllll}
\alpha & + \ r^1 u_{11} & + \cdots + & r^q u_{1q} & & & & \geq & 0 \\
\vdots & & & & \ddots & & & \vdots & \\
\alpha & & & & -r^1 u_{t1} & - \cdots - & -r^q u_{tq} & \geq & 0 \\
& -\beta + f_1 u_{11} & + \cdots + & f_q u_{1q} & & & & \geq & 0 \\
& \vdots & & & \ddots & & & \vdots & \\
& -\beta & & & +(1-f_1) u_t f_1 & + \cdots + & (1-f_q) u_{tq} & \geq & 0 \\
\multicolumn{9}{c}{u_{ik} \geq 0,\ i = 1, \ldots, t,\ k = 1, \ldots, q}
\end{array}
\tag{2.2.7}
$$

Applying the normalization $\beta = 1$ (clearly $\beta = -1$ does not yield any cuts since it makes (2.2.7) unbounded) we obtain the representation given in the theorem. $\square$

In order to restate the system (2.2.5) in a more concise form, for each $i \in \{1, \ldots, t\}$ we partition the index set $Q := \{1, \ldots, q\}$ into

$$Q_i^+ := \{k \in Q : u_{ik} \text{ has coefficient vector } r_k\}$$
$$Q_i^- := \{k \in Q : u_{ik} \text{ has coefficient vector } -r_k\},$$

with $Q_i^+ \cup Q_i^- = Q$, $i = 1, \ldots, t = 2^k$. Then (2.2.5) can be restated as

$$\begin{array}{llllll} \alpha & + \sum\left(r^k u_{ik} : k \in Q_i^+\right) & - & \sum\left(r^k u_{ik} : k \in Q_i^-\right) & \geq & 0 \\ & \sum(f_k u_{ik} : k \in Q_i^+) & + & \sum((1-f_k) u_{ik} : k \in Q_i^-) & \geq & 1, \quad i = 1, \ldots, t \\ & & & u_{ik} \geq 0, \; i = 1, \ldots, t = 2^q, \; k = 1, \ldots, q & & \end{array} \tag{2.2.5$'$}$$

The system (2.2.5) has several interesting properties described in the next few propositions.

Proposition 2.2.2. *For any $p \in \mathbb{R}^n$, $p > 0$, all optimal basic solutions to the cut generating linear program*

$$\min\{p\alpha : (\alpha, u) \textit{ satisfies (2.2.5)}\} \tag{CGLP$_Q$}$$

are of the form

$$\alpha_j = \max\{\alpha_j^1, \ldots, \alpha_j^t\}, \tag{2.2.8}$$

where

$$\alpha_j^i := -\sum(r_j^k u_{ik} : k \in Q_i^+) + \sum(r_j^k u_{ik} : k \in Q_i^-), \tag{2.2.9}$$

$i = 1, \ldots, t = 2^q$, with the u_{ik} satisfying (2.2.5$'$).

Proof. The constraints of (2.2.5) require

$$\alpha_j \geq \alpha_j^i, \quad i = 1, \ldots, t, \; j = 1, \ldots, n$$

Suppose there is an optimal solution to (CGLP)$_Q$ such that $\alpha_{j_*} > \max\{\alpha_{j_*}^i : i = 1, \ldots, t\}$ for some $j_* \in \{1, \ldots, n\}$. Then setting α_{j_*} equal to the maximum on the righthand side, and leaving α_j unchanged for all $j \neq j_*$ yields a better solution, contrary to the assumption. □

Proposition 2.2.3. *In any valid inequality $\alpha s \geq 1$ for* conv $P_D^{(n)}$*, $\alpha_j \geq 0$, $j = 1, \ldots, n$.*

Proof. From (2.2.8), $\alpha_j \geq \alpha_j^i$ for all $i = 1, \ldots, 2^q$, and in view of the presence of all sign patterns of $r_j^k u_{ik}$ in the expressions (2.2.9), there is always an index $i \in \{i, \ldots, 2^q\}$ with $\alpha_j^i \geq 0$. □

Proposition 2.2.4. *For any basic solution (α, u) to (CGLP)$_Q$ that satisfies as strict inequality some of the nonhomogeneous constraints of (2.2.5), there exists a basic solution $(\bar{\alpha}, u)$, with $\bar{\alpha} = \alpha$, that satisfies at equality all the nonhomogeneous constraints of (CGLP)$_Q$.*

Proof. Let (α, u) be a basic solution to (CGLP)$_Q$ that satisfies as strict inequality some of the nonhomogeneous constraints of (2.2.5). W.l.o.g., assume that

$$f_1 u_{11} + \cdots + f_q u_{1q} - \theta = 1$$

is one of those constraints with the surplus variable θ positive in the solution (α, u). We will show that there exists a solution $(\bar{\alpha}, \bar{u})$, with $\bar{\alpha} = \alpha$ and $\bar{u}_{ik} = u_{ik}$ for all $i \neq 1$ and all k, such that

$$f_1\bar{u}_{11} + \cdots + f_q\bar{u}_{1q} = 1.$$

Applying this argument recursively then proves the Proposition.

Fix all variables of $(\text{CGLP})_Q$ except for $u_{11}, \ldots, u_{1q}$, at their values in the current solution. The fixing includes all the surplus variables except those in the $n+1$ rows containing $u_{11}, \ldots, u_{1q}$. This leaves the following constraint set in the free variables:

$$\begin{array}{rcll} -r_j^1 u_{11} - \cdots - r_j^q u_{1q} + t_j & = & \bar{\alpha}_j & \qquad j = 1, \ldots, n \\ f_1 u_{11} + \cdots + f_q u_{1q} - \theta & = & 1 & \\ \multicolumn{4}{c}{u_{11}, \ldots, u_{1q} \geq 0,\ t_j \geq 0,\ j = 1, \ldots, n,\ \theta \geq 0} \end{array} \tag{2.2.10}$$

Here θ, t_j represent the surplus variables of the respective constraints. We claim that this system has a solution with $\theta = 0$. To see this, consider the linear program

$$\min\{\theta : u_{ik}, t_j \text{ and } \theta \text{ satisfy (2.2.10)}\}$$

and its dual,

$$\max \lambda_0 + \sum_{j=1}^{n} \bar{\alpha}_j \lambda_j$$

subject to

$$\begin{array}{rcl} f_1\lambda_0 - \sum_{j=1}^{n} r_j^1 \lambda_j & \leq & 0 \\ \vdots & & \vdots \\ f_q\lambda_0 - \sum_{j=1}^{n} r_j^q \lambda_j & \leq & 0 \\ -\lambda_0 \qquad\qquad & \leq & 1 \\ \lambda_j & \leq & 0, \quad j = 1, \ldots, n \end{array}$$

Since $\bar{\alpha}_j \geq 0$, $j = 1, \ldots, n$, it is not hard to see that the dual linear program has an optimal solution $\lambda_0 = 0$, $\lambda_j = 0$, $j = 1, \ldots, n$ and hence the primal has an optimal solution with $\theta = 0$. □

The obvious and important consequence of Proposition 2.2.4 is that for all practical purposes we can replace all 2^q nonhomogeneous inequalities in the constraint set (2.2.5) of $(\text{CGLP})_Q$ with equations. In view of Proposition 2.2.2, it then follows that we may restrict our attention to basic feasible solutions that satisfy at equality $n + 2^q$ out of the $n \times 2^q + 2^q$ inequalities of (2.2.5) other than the nonnegativity constraints.

At this point we introduce the characterization of $\operatorname{conv} P_D^{=}$, the 0-1 Disjunctive Hull defined by (2.2.2), closely related to that of $\operatorname{conv} P_D$. Just as in the case of P_D, we denote by $P_D^{=(n)}$ the union of polyhedra in $\mathbb{R}^n$ representing the disjunction (2.2.4) in which all the inequalities have been replaced by equations. The following Theorem is the analog of

Theorem 2.2.1 for this case.

Theorem 2.2.5. $\operatorname{conv} P_D^{=(n)}$ *is the set of those* $s \in \mathbb{R}^n$ *satisfying* $s \geq 0$ *and all inequalities* $\alpha s \geq \beta$ *whose coefficients satisfy the system*

$$
\begin{array}{llll}
\alpha + r^1 u_{11} + \cdots + r^1 u_{1q} & & \geq & 0 \\
\vdots & \ddots & \vdots & \\
\alpha & -r^1 u_{t1} - \cdots - r^q u_{tq} & \geq & 0 \\
\quad -\beta + f_1 u_{11} + \cdots + f_q u_{1q} & & = & 0 \\
\vdots & \ddots & \vdots & \\
\quad -\beta & +(1-f_1) u_{t1} + \cdots + (1-f_q) u_{tq} & = & 0
\end{array}
\tag{2.2.11}
$$

for some u_{ik}, $i = 1, \ldots, t = 2^q$, $k = 1, \ldots, q$.

Proof. The proof of Theorem 2.2.1 goes through with the following modifications. Since the inequalities in the disjunctions (2.2.3) and (2.2.4) are all replaced with equations, the inequalities in the system (2.2.6), other than the nonnegativity constraints, also become equations. As a consequence, the variables u_{ik} of the projected system (2.2.7) become unrestricted in sign. The remaining difference between (2.2.11) and (2.2.5) is the fact that in (2.2.11) the last 2^q constraints are equations rather than inequalities. This is due to the fact that Proposition 2.2.4 applies here too. In other words, if we denote by (2.2.11′) the system obtained from (2.2.11) by replacing the equations containing β with inequalities $\geq$, then for any basic solution (α, u) to $(\text{CGLP})_Q$ that satisfies as strict inequalities some of the constraints (2.2.11′) containing β, there exists a basic solution $(\bar{\alpha}, u)$, with $\bar{\alpha} = \alpha$, that satisfies at equality all the constraints containing β. The proof is essentially the same as that of Proposition 2.2.4.

Thus the two basic differences between the systems describing $\operatorname{conv} P_D^{(n)}$ and $\operatorname{conv} P_D^{=(n)}$ are that (a) the latter also contains inequalities of the form $\alpha x \leq 1$ (corresponding to $\beta < 0$), and (b) the coefficients α_j of the latter can be of any sign. □

We now return to the simple Disjunctive Hull, $\operatorname{conv} P_D$, and describe its vertices.

Proposition 2.2.6. *Every vertex of* $\operatorname{conv} P_D^{(n)}$ *is a vertex of some* $P_i^{(n)}$, $i \in \{1, \ldots, 2^q\}$.

Proof. Let v be a vertex of $\operatorname{conv} P_D^{(n)}$. If $v \in P_i^{(n)}$ for some $i \in \{1, \ldots, t = 2^q\}$, then v must be a vertex of $P_i^{(n)}$, or else it could be expressed as a convex combination of points in $P_i^{(n)}$, hence of $P_D^{(n)}$. On the other hand, if $v \notin \cup P_i^{(n)}$ but $v \in \operatorname{conv} P_i^{(n)}$, then v is a convex combination of points in $\cup_{i=1}^t P_i^{(n)}$, hence of $\operatorname{conv} P_D^{(n)}$, a contradiction. □

Next we describe the vertices of $P_i^{(n)}$, $i \in \{1, \ldots, 2^q\}$. We will call a vertex of $\operatorname{conv} P_D^{(n)}$ (of $P_i^{(n)}$) integer if it defines an integer x through (2.2.1); in other words if $f_i + r^i s$ is integer for $i = 1, \ldots, q$. All other vertices will be called fractional.

For any particular $i_* \in \{1, \ldots, 2^q\}$,

$$P_{i_*}^{(n)} := \{s \in \mathbb{R}_+^n : r^h s \leq -f_h,\ h \in Q_{i_*},\ r^h s \geq 1 - f_h,\ h \in Q \setminus Q_{i_*}\}$$

where $(Q_{i_*}, Q \setminus Q_{i_*})$ is the partition of Q that defines i_*.

Proposition 2.2.7. $P_{i_*}^{(n)}$ *can have three kinds of vertices, distinguished by the corresponding* x*-vectors that belong to one of these types:*

(a) 0-1 vertices: $x_h = 0$, $h \in Q_{i_*}$ *and* $x_h = 1$, $h \in Q \setminus Q_{i_*}$.

(b) non-binary integer vertices: $x_h \in \mathbb{Z}_-$, $h \in Q_{i_*}$, $x_h \in \mathbb{Z}_+$, $h \in Q \setminus Q_{i_*}$ *(here* $\mathbb{Z}_-$ *and* $\mathbb{Z}_+$ *stand for the nonpositive and nonnegative integers respectively).*

(c) fractional vertices: $x_h \leq 0$, $h \in Q_{i_*}$, $x_h \geq 1$, $h \in Q \setminus Q_{i_*}$, *with at least one inequality strict.*

Proof. The three cases become exhaustive if the following fourth one is added: (d) fractional vertices with $0 < x_h < 1$ for some $h \in Q$. But this case clearly violates at least one of the constraints defining $P_{i_*}^{(n)}$. □

Note that $P_{i_*}^{(n)}$ can have several distinct vertices with the same associated x-vector, corresponding to basic solutions with the same x-component. Note also that if a component x_h of a vertex is fractional, then $x_h < 0$ or $x_h > 1$. Finally, note that $P_{i_*}^{(n)}$ may contain nonbinary integer components that are not vertices.

The next theorem characterizes the facets of the Simple Disjunctive Hull.

Theorem 2.2.8. *The inequality* $\bar{\alpha} s \geq 1$ *defines a facet of* $\operatorname{conv} P_D^{(n)}$ *if and only if there exists an objective function of the linear program (CGLP)*$_Q$ *of Proposition 2.2.2 with* $p > 0$ *such that all optimal solutions* (α, u) *have* $\alpha = \bar{\alpha}$.

Proof outline. This is a special case of Theorem 4.6 of [4]. The inequality $\bar{\alpha} x \geq 1$ defines a facet of $\operatorname{conv} P_D^{(n)}$ if and only if $\bar{\alpha}$ is a vertex of the polar of $\operatorname{conv} P_D^{(n)}$, which is the projection of (2.2.5) onto the α-space. But $\bar{\alpha}$ is a vertex of this polar if and only if there exists an objective function vector $p > 0$ such that $p\alpha$ attains its unique minimum at $\bar{\alpha}$. □

If the system (2.2.1) defining P_L is of full row rank q, then the dimension of $\operatorname{conv} P_D$ is n, since there are $q + n$ variables and q independent equations. The dimension of $\operatorname{conv} P_D^{(n)}$ is also n, so the facets of $\operatorname{conv} P_D^{(n)}$ are of dimension $n - 1$.

From a computational standpoint, the most important feature of (CGLP)$_Q$ is that the facets of the n-dimensional $\operatorname{conv} P_D^{(n)}$ can be generated by solving a smaller CGLP in a subspace of at most $t = 2^q$ variables s_j, and lifting the resulting inequality into the full space. The idea of generating cuts in a subspace of the original higher dimensional cut generating linear program and then lifting them to the full space goes back to [9, 12], where lift-and-project cuts were generated from a 2-term disjunction by working in the subspace of the fractional variables of the LP solution. Here we are working with a 2^q-term disjunction, and are considering a different subspace, suggested by the structure of the problem at hand, but the lifting procedure is essentially the same as the one used in [9, 10].

Since our cuts are derived from a disjunction with 2^q terms, if we want to create a subproblem in which all terms are represented, we need 2^q out of the n variables α_j of our (CGLP)$_Q$. Furthermore, the 2^q vectors r_j corresponding to these α_j have to span the subspace $\mathbb{R}^q$ of the x-variables. Solving the (CGLP)$_Q$ in this subspace yields 2^q values α_j and $q \times 2^q$ associated multipliers u_{ik}, $i = 1, \dots, 2^q$, $k = 1, \dots, q$; and these multipliers can then be used to compute the remaining components of α. The significance of this is that the computational cost of generating facets of conv P_D grows only linearly with n. Of course this cost grows exponentially with q, but the approach discussed here is being considered for small q (q in the single digits).

The choice of the subspace is intimately related to one of the central questions that we are pursuing, that of deciding which facets of the Disjunctive Hull are also facets of the Integer Hull. The best way to address this question and that of the subspace to be chosen, is to first interpret the inequalities defining the Disjunctive Hull as intersection cuts.

2.2.2 Cuts from the q-dimensional parametric cross-polytope

Consider the q-dimensional unit cube centered at $\left(\frac{1}{2},\ldots,\frac{1}{2}\right)$, $K_q := \{x \in \mathbb{R}^q : -\frac{1}{2} \leq x_j \leq \frac{1}{2},\ j \in Q\}$. Its polar, $K_q^o := \{x \in \mathbb{R}^q : xy \leq 1,\ \forall x \in K\}$, is known to be the q-dimensional octahedron or cross-polytope; which, when scaled so as to circumscribe the unit cube, is the outer polar of K_q:

$$K_q^* = \{x \in \mathbb{R}^q : |x| \leq \tfrac{1}{2}q\},$$

where $|x| = \sum(|x_j| : j = 1,\ldots,q\}$. Equivalently, $|x| \leq \frac{1}{2}q$ can be written as the system

$$\begin{array}{rcccccl} -x_1 & - & \cdots & - & x_q & \leq & \frac{1}{2}q \\ x_1 & - & \cdots & - & x_q & \leq & \frac{1}{2}q \\ & & & & & \vdots & \\ x_1 & + & \cdots & + & x_q & \leq & \frac{1}{2}q \end{array} \tag{2.2.12}$$

of $t = 2^q$ inequalities in q variables.

Moving the center of the coordinate system to $(0,\ldots,0)$ from $(\frac{1}{2},\cdots,\frac{1}{2})$ changes the righthand side coefficient of the i-th inequality from $\frac{1}{2}q$ to the sum of positive coefficients on the lefthand side of the inequality.

Next we introduce the parameters v_{ik}, $i = 1,\ldots,t = 2^q$, $k = 1,\ldots,q$, to obtain the system

$$\begin{array}{rcccccl} -v_{11}x_1 & - & \cdots & - & v_{1q}x_q & \leq & 0 \\ v_{21}x_1 & - & \cdots & - & v_{2q}x_q & \leq & v_{21} \\ -v_3x_1 & + & \cdots & - & v_{3q}x_q & \leq & v_{31} \\ \vdots & & & & & \vdots & \\ v_{t1}x_1 & + & \cdots & + & v_{tq}x_q & \leq & v_{t1} + \ldots + v_{tq} \end{array} \tag{2.2.13}$$
$$v_{ik} \geq 0,\ i = 1,\ldots,t = 2^q,\ k = 1,\ldots,q.$$

Note that the constraints of (2.2.13) are of the form

$$\sum_{k \in \widetilde{Q}_i^+} v_{ik}x_k - \sum_{k \in \widetilde{Q}_i^-} v_{ik}x_k \leq \sum_{k \in \widetilde{Q}_i^+} v_{ik},$$

where $\widetilde{Q}_i^+$ and $\widetilde{Q}_i^-$ are the sets of indices for which the coefficient of x_k is $+v_{ik}$ and $-v_{ik}$, respectively. Note also that all inequalities that have the same number of coefficients with the plus sign have the same righthand side, equal to the sum of these coefficients.

The system (2.2.13) is homogeneous, so every one of its inequalities can be normalized. Since we are looking for a connection with the system (2.2.5) defining $(\mathrm{CGLP})_Q$, we will use the normalization given by this system and Proposition 2.2.4, i.e.

$$\begin{array}{rcccrcl} f_1 v_{11} & + & \cdots & + & f_q v_{1q} & = & 1 \\ (1-f_1)v_{21} & + & \cdots & + & f_q v_{2q} & = & 1 \\ \cdots & & & & \cdots & & \\ (1-f_1)v_{t1} & + & \cdots & + & (1-f_q)v_{tq} & = & 1 \end{array} \tag{2.2.14}$$

Note that these normalization constraints are of the general form

$$\sum_{h \in \widetilde{Q}_i^+} (1-f_k)v_{ik} + \sum_{h \in \widetilde{Q}_i^-} f_k v_{ik} = 1.$$

Let $\widetilde{K}^*(v)$ denote the parametric cross-polytope defined by (2.2.13). It is not hard to see that for any fixed set of v_{ik}, (2.2.13) defines a convex polyhedron in x-space that contains in its boundary all $x \in \mathbb{R}^q$ such that $x_k \in \{0,1\}$, $k \in Q$, hence is suitable for generating intersection cuts. Furthermore, letting $\widetilde{K}^{*(n)}(v)$ be the expression for $\widetilde{K}^*(v)$ in the space of the s-variables, obtained by substituting $f + Rs$ for x into (2.2.13), we have

Theorem 2.2.9. *For any values of the parameters v_{ik} satisfying (2.2.13), the intersection cut $\widetilde{\alpha}s \geq 1$ from $\widetilde{K}^{*(n)}(v)$ has coefficients $\widetilde{\alpha}_j = \frac{1}{s_j^*}$, where*

$$s_j^* = \max\{s_j : f + r_j s_j \in K^{*(n)}(v)\}. \tag{2.2.15}$$

Proof. This is a special case of Theorem 2.1.1. $\square$

In order to compare the intersection cut $\widetilde{\alpha}s \geq 1$ with the cut $\alpha s \geq 1$ from the q-term disjunction (2.2.4), we have to restate (2.2.15) in terms of the system of inequalities defining $\widetilde{K}^{*(n)}(v)$. This means that $f + r_j s_j^*$ has to be expressed as the intersection point of the ray $f + r_j s_j$, $s_j \geq 0$, with the first facet of $K^{*(n)}(v)$ encountered. This yields

$$s_j^* = \min\{s_j^1, \ldots, s_j^t\}, \tag{2.2.16}$$

where the s_j^i are obtained by substituting $f_k + \sum_{h=1}^n r_j^k s_h$ for x_k, $k = 1, \ldots, q$ into the i-th inequality of (2.2.13), and setting $s_h = 0$ for all $h \neq j$:

$$s_j^i = \max\left\{ s_j : \left(\sum_{k \in \widetilde{Q}_i^+} v_{ik} r_j^k - \sum_{k \in \widetilde{Q}_i^-} v_{ik} r_j^k \right) s_j \leq \sum_{k \in \widetilde{Q}_i^+} v_{ik}(1-f_k) + \sum_{k \in \widetilde{Q}_i^-} v_{ik} f_k \right\},$$

$i = 1, \ldots, t = 2^q$.

Clearly, this maximum is bounded whenever the coefficient of s_j is positive, in which case, if we normalize by setting $\sum_{k\in\widetilde{Q}_i^+} v_{ik}(1-f_k) + \sum_{k\in\widetilde{Q}_i^-} v_{ik} f_k = 1$, we obtain

$$s_j^i = \left(\sum_{k\in\widetilde{Q}_i^+} v_{ik} r_j^k - \sum_{k\in\widetilde{Q}_i^-} v_{ik} r_j^k \right)^{-1} . \tag{2.2.17}$$

Comparing (2.2.16) and (2.2.17) to the expressions (2.2.8) and (2.2.9) for the coefficient α_j of the lift-and-project cut $\alpha s \geq 1$ of section 4, we find that setting $v_{ik} = u_{ik}$ for all i, k, as well as $\widetilde{Q}_i^+ = Q_i^-$ and $\widetilde{Q}_i^- = Q_i^+$, we obtain $\widetilde{\alpha}_j = \alpha_j$.

This proves

Corollary 2.2.10. *The intersection cut $\widetilde{\alpha}s \geq 1$ from the parametric octahedron $\widetilde{K}^{*(n)}(v)$ is the same as the lift-and-project cut $\alpha s \geq 1$ corresponding to the (CGLP)$_Q$ solution (α, u), with $v_{ik} = u_{ik}$, $i = 1, \ldots, t$, $k = 1, \ldots, q$.*

2.2.3 Facets of the Disjunctive Hull and the Integer Hull

Consider again the disjunctive relaxation of P_I

$$P_D = \{(x,s) \in \mathbb{R}^q \times \mathbb{R}^n : x = f + Rs,\ s \geq 0,\ x_i \leq 0 \ \vee\ x_i \geq 1,\ i \in Q\}$$

introduced in section 3, where $x, f \in \mathbb{R}^q$, $R \in \mathbb{R}^{q\times n}$, and $Q := \{1, \ldots, q\}$. For $i = 1, \ldots, t = 2^q$, let p^i be the vertex of K_q, the q-dimensional unit cube, defined by $p_k^i = 0$, $i \in Q_i^+$, $p_k^i = 1$, $i \in Q_i^-$.

Next we give a sufficient condition for an inequality $\alpha s \geq 1$ valid for P_D to define a facet of $\operatorname{conv} P_I$, which for small q leads to an efficient procedure for generating inequalities that are facet defining for $\operatorname{conv} P_I$.

The dimension of $P_I^{(n)}$ being $n \geq 2^q$, $\alpha s \geq 1$ defines a facet of $\operatorname{conv} P_I^{(n)}$ if there exists a subspace $\mathbb{R}^{2^q}$ of $\mathbb{R}^n$ such that the restriction of $\alpha s \geq 1$ to this subspace defines a facet of $\operatorname{conv} P_I^{(2^q)}$. If this is the case, then the inequality in question can be lifted to the full space to yield a facet of $\operatorname{conv} P_I^{(n)}$ by using the u-components of the solution (α, u) to the CGLP in the subspace to compute the missing coefficients α_j.

Theorem 2.2.11. *Let $\alpha s \geq 1$ be a valid inequality for P_D corresponding to a basic solution (α, u) of (CGLP)$_Q$, and let p^i, $i = 1, \ldots, 2^q$, be the vertices of K_q. Suppose for each p^i, $i = 1, \ldots, 2^q$, there exists a subset $J_i \subset J$ containing the indices of q linearly independent rays $r^{j_1}, \ldots, r^{j_q}$, and a vector $\lambda \in \mathbb{R}_+^q$, satisfying*

$$p^i - f = \sum_{j=j_1}^{j_q} \tfrac{1}{\alpha_j} r_j \lambda_j, \quad \sum_{j=j_1}^{j_q} \lambda_j = 1. \tag{2.2.18}$$

Then the inequality $\sum_{j\in J} \alpha_j s_j \geq 1$ defines a facet of $\operatorname{conv} P_I^{(|J|)}$*, and its lifting based on the u-components of the solution (α, u) defines a facet of* $\operatorname{conv} P_I^{(n)}$.

Proof. Suppose the subset of 2^q rays indexed by J satisfies the requirements of the Theorem. Then for every $i = 1, \dots, 2^q$, the vertex p^i of K^q satisfies

$$p^i = \sum_{j=j_1}^{j_q} (f - \tfrac{1}{\alpha_j} r_j)\lambda_j, \quad \sum_{j=j_1}^{j_q} \lambda_j = 1$$

for some $\lambda_j \geq 0$, $j = j_1, \dots, j_q$, i.e. p^i can be expressed as a convex combination of the q points $f + \frac{1}{\alpha_j} r_j$, $j = j_1, \dots, j_q$. But $f + \frac{1}{\alpha_j} r_j = f + r_j s_j^*$ is the intersection point of the ray $f + r_j s_j$ with $\operatorname{bd} \widetilde{K}_q^*$, hence each of these points satisfies $\alpha s = 1$ and consequently so does p^i. Since $\sum_{j \in J} \alpha_j s_j \geq 1$ is satisfied at equality by 2^q integer points of $\operatorname{conv} P_I^{(|J|)}$, it defines a facet of the latter. Furthermore, lifting the remaining coefficients α_j of the inequality by using the u-components of (α, u) yields a facet defining inequality for $\operatorname{conv} P_I^{(n)}$. $\square$

The sufficient condition of Theorem 2.2.11 is not necessary. There are two kinds of situations not satisfying the above condition, in which a valid inequality $\alpha s \geq 1$ for P_D may define a facet of $\operatorname{conv} P_I$. The first one involves an inequality $\alpha s \geq 1$ such that (2.2.18) is not satisfied for all 2^q vertices of K^q, but $\operatorname{conv} P_D$ has 2^q vertices whose x-components p^i satisfy (2.2.18), i.e. $\operatorname{conv} P_D$ has multiple vertices with the same x-component. The second situation involves facet defining split cuts.

2.3 The two-row case

We now restrict our attention to the case $q = 2$, i.e. we consider two rows from a simplex tableau of a MIP problem with the variables x_1, x_2 and $s_j, j \in J$:

$$P_L = \{(x,s) \in \mathbb{R}^{2+|J|} : \begin{array}{ll} x_1 & = f_1 + \sum_{j \in J} r_j^1 s_j \\ x_2 & = f_2 + \sum_{j \in J} r_j^2 s_j \\ s_j \geq 0 & j \in J \; \}. \end{array} \tag{2.3.1}$$

where x_1, x_2 are basic variables required to be integers and $s_j, j \in J$ are non-basic. Let $P_I = \{(x,s) \in \mathbb{Z}^2 \times \mathbb{R}^{|J|} : (x,s) \in P_L\}$. We can assume that f_1, f_2 are fractional values such that $0 < f_1 < 1$ and $0 < f_2 < 1$.

In the literature the column vectors $r_j, j \in J$ are called **rays**. In addition we give the following definition:

Definition We say that a ray r_j in (2.3.1) **hits** an orthant-cone $Q_i, i \in \{1, \dots, 4\}$ if there exists $\lambda_0 > 0$ such that $f + \lambda r_j \in Q_i$ for all $\lambda \geq \lambda_0$.

For the case of 2 rows we can express the Disjunctive Relaxation as follows. Denote by s the vector of non-basic variables $s_j, j \in J$ and by r^1, r^2 the row vectors of their corresponding coefficients in (2.3.1). For this case (2.2.4) becomes

$$\begin{pmatrix} -r^1 s \geq f_1 \\ -r^2 s \geq f_2 \end{pmatrix} \vee \begin{pmatrix} r^1 s \geq 1 - f_1 \\ -r^2 s \geq f_2 \end{pmatrix} \vee \begin{pmatrix} r^1 s \geq 1 - f_1 \\ r^2 s \geq 1 - f_2 \end{pmatrix} \vee \begin{pmatrix} -r^1 s \geq f_1 \\ r^2 s \geq 1 - f_2 \end{pmatrix} \tag{2.3.2}$$

for $s \geq 0$.
As shown in [4, 9] we can reformulate (2.3.2) as the constraint set

$$\begin{array}{llll} s & -s^1 - s^2 - s^3 - s^4 & & = 0 \quad (\alpha) \\ & & s_0^1 + s_0^2 + s_0^3 + s_0^4 & = 1 \quad (\beta) \\ & -r^1 s^1 & -f_1 s_0^1 & \geq 0 \quad (v_1) \\ & -r^2 s^1 & -f_2 s_0^1 & \geq 0 \quad (w_1) \\ & +r^1 s^2 & -(1-f_1) s_0^2 & \geq 0 \quad (v_2) \\ & -r^2 s^2 & -f_2 s_0^2 & \geq 0 \quad (w_2) \\ & +r^1 s^3 & -(1-f_1) s_0^3 & \geq 0 \quad (v_3) \\ & +r^2 s^3 & -(1-f_2) s_0^3 & \geq 0 \quad (w_3) \\ & -r^1 s^4 & -f_1 s_0^4 & \geq 0 \quad (v_4) \\ & +r^2 s^4 & -(1-f_2) s_0^4 & \geq 0 \quad (w_4) \\ \multicolumn{4}{l}{s^i \geq 0, \quad i \in \{1 \dots 4\}} \\ \multicolumn{4}{l}{s_0^i \geq 0, \quad i \in \{1 \dots 4\}.} \end{array} \tag{2.3.3}$$

The Disjunctive Hull is the projection of (2.3.3) onto the space of the s variables. The Lift-and-Project framework allows us to generate valid cuts for the Disjunctive Hull. A valid cut $\alpha s - \beta \geq 0$ for the Disjunctive Hull expressed in the space of the s variables only is given by solving the Cut Generation Linear Program for an appropriate objective function with $c_\beta \geq 0, c_j \geq 0, j \in J$.

$$\begin{array}{llll} \multicolumn{4}{l}{\min \sum_j c_j \alpha_j + c_\beta \beta} \\ \alpha & +r^1 v_1 & +r^2 w_1 & \geq 0 \\ \alpha & -r^1 v_2 & +r^2 w_2 & \geq 0 \\ \alpha & -r^1 v_3 & -r^2 w_3 & \geq 0 \\ \alpha & +r^1 v_4 & -r^2 w_4 & \geq 0 \\ -\beta & +f_1 v_1 & +f_2 w_1 & \geq 0 \\ -\beta & +(1-f_1) v_2 & +f_2 w_2 & \geq 0 \\ -\beta & +(1-f_1) v_3 & +(1-f_2) w_3 & \geq 0 \\ -\beta & +f_1 v_4 & +(1-f_2) w_4 & \geq 0 \\ \multicolumn{4}{l}{v_i, w_i \geq 0 \quad i \in \{1 \dots 4\}.} \end{array} \tag{2.3.4}$$

Applying normalization $\beta = 1$, by Proposition 2.2.4 (2.3.4) becomes

$$\begin{array}{llll} \multicolumn{4}{l}{\min \sum_j c_j \alpha_j} \\ \alpha & +r^1 v_1 & +r^2 w_1 & \geq 0 \\ \alpha & -r^1 v_2 & +r^2 w_2 & \geq 0 \\ \alpha & -r^1 v_3 & -r^2 w_3 & \geq 0 \\ \alpha & +r^1 v_4 & -r^2 w_4 & \geq 0 \\ & +f_1 v_1 & +f_2 w_1 & = 1 \\ & +(1-f_1) v_2 & +f_2 w_2 & = 1 \\ & +(1-f_1) v_3 & +(1-f_2) w_3 & = 1 \\ & +f_1 v_4 & +(1-f_2) w_4 & = 1 \\ \multicolumn{4}{l}{v_i, w_i \geq 0 \quad i \in \{1 \dots 4\}.} \end{array} \tag{2.3.5}$$

By Proposition 2.2.2 the cuts generated by (2.3.5) have the form $\alpha s \geq 1$ where

$$\alpha_j = \max\{\alpha_j^1, \alpha_j^2, \alpha_j^3, \alpha_j^4\}$$

where

$$\begin{array}{lll} \alpha_j^1 & = -r_j^1 v_1 & -r_j^2 w_1 \\ \alpha_j^2 & = +r_j^1 v_2 & -r_j^2 w_2 \\ \alpha_j^3 & = +r_j^1 v_3 & +r_j^2 w_3 \\ \alpha_j^4 & = -r_j^1 v_4 & +r_j^2 w_4. \end{array} \tag{2.3.6}$$

2.3.1 Geometric interpretation of the CGLP

A cut produced by the CGLP can be viewed as an intersection cut given by a convex set that does not contain any feasible integer point in its interior. This convex set is uniquely determined by the values of v, w.

Definition For given v, w, we call the polyhedron

$$P_{\text{octa}}(v, w) = \{(x_1, x_2) \in \mathbb{R}^2 : \begin{array}{l} -v_1 x_1 - w_1 x_2 \leq 0 \ ; \\ +v_2 x_1 - w_2 x_2 \leq v_2 \ ; \\ +v_3 x_1 + w_3 x_2 \leq v_3 + w_3 \ ; \\ -v_4 x_1 + w_4 x_2 \leq w_4 \end{array} \}$$

the (v, w)**-parametric octahedron**.

If $v_i = 0$ or $w_i = 0$ for some $i \in \{1, \ldots, 4\}$ the i-th facet of P_{octa} is parallel to one of the coordinate axes. If $v_i, w_i > 0$ then the i-th facet of P_{octa} is *tilted* (note that since we use the normalization $\beta = 1$, v_i and w_i cannot both be 0).

Varying the parameters v, w, the (v, w)**-parametric octahedron** produces different configurations according to the non-zero components of v, w.. In the rest of the section we refer to these configurations using the short reference indicated in parenthesis.

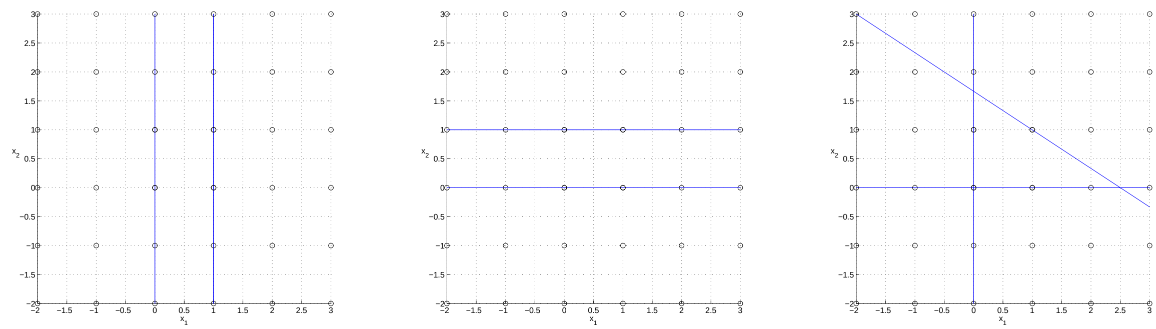
- (S) If exactly 4 components of (v, w) are positive, the parametric octahedron is the split $\{x \in \mathbb{R}^2 : 0 \leq x_1 \leq 1\}$ if $v_i > 0, i = 1, \ldots, 4$; or $x_2 \leq 0 \vee x_2 \geq 1$ if $w_i > 0, i = 1, \ldots, 4$. Figure 2.3(a) illustrates the case with $v_i > 0, w_i = 0, i \in \{1, \ldots, 4\}$,Figure 2.3(b) illustrate the case with $v_i = 0, w_i > 0, i \in \{1, \ldots, 4\}$.

- (T_A) If exactly 5 components of (v, w) are positive, the parametric octahedron is a triangle with 1 tilted face. Figure 2.3(c) illustrate the case with $v_1, w_2, v_3, w_3, v_4 > 0; w_1, v_2, w_4 = 0$. When $v_i = w_i$ for some $i \in \{1, \ldots, 4\}$ the parametric octahedron defines a triangle with vertices $(0, 0); (2, 0); (0, 2)$ or symmetric configurations, then it corresponds to a *Triangle of Type 1* as in [2]. In the general case it corresponds to a *Triangle of Type 2* as in [2].

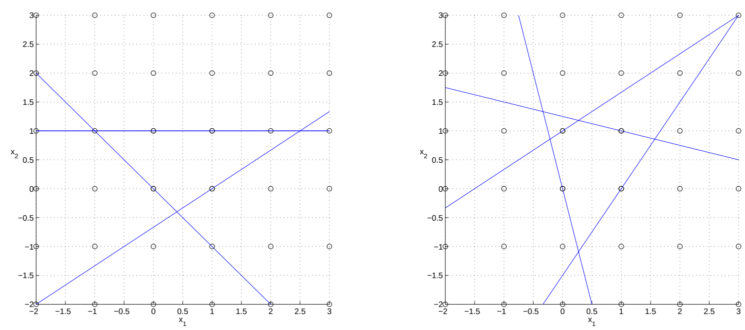
- (T_B) If exactly 6 components of (v, w) are positive, the parametric octahedron is a triangle with 2 tilted faces. Figure 2.3(d) illustrate the case with $v_1, w_1, v_2, w_2, w_3, w_4 > 0; v_3, v_4 = 0$. In [2] this configuration corresponds to a *Triangle of Type 2*.

- (Q) If all 8 components of (v, w) are positive, the parametric octahedron is a quadrilateral. See Figure 2.3(e).

The case with 7 components of (v, w) positive does not correspond to a maximal parametric octahedron, therefore we do not need to consider it. Suppose all the components are positive except for v_1 which is 0. The facet of P_{octa} corresponding to $(0, 0)$ is horizontal and goes through the point $(1, 0)$. Is not hard to see that setting $v_2 = 0$ we enlarge the set defined by the parametric octahedron.



(a) 4 non-zeros - vertical strip (b) 4 non-zeros - horizontal strip (c) 5 non-zeros



(d) 6 non-zeros (e) 8 non-zeros

Figure 2.3: Configurations of the parametric octahedron for the MIP case

For a cut $\sum_{j \in J} \alpha_j s_j \geq 1$ Andersen et al. introduce in [2] the set

$$L_\alpha = \left\{ x \in \mathbb{R}^2 : (x, s) \in P_L \wedge \sum_{j \in J} \alpha_j s_j \leq 1 \right\}. \tag{2.3.7}$$

Clearly, $L_\alpha \subseteq P_{\text{octa}}(v, w)$, and the inclusion is sometimes strict.

2.3.2 Disjunctive Hull facets for the Andersen et al. example

In [2], Andersen et al. considered the two rows instance

$$\begin{array}{lllllll} x_1 & = \frac{1}{4} & +2s_1 & +1s_2 & -3s_3 & & +1s_5 \\ x_2 & = \frac{1}{2} & +1s_1 & +1s_2 & +2s_3 & -1s_4 & -2s_5. \end{array} \tag{2.3.8}$$

We present the complete description of the Disjunctive Hull for (2.3.8). In order to do so we generated the CGLP of (2.3.8) using the normalization constraint $\beta = 1$ and we considered all feasible bases. Every feasible base could yield a facet necessary to the description of the Disjunctive Hull. From a practical point of view, the method we followed is not applicable for real world instances, since the number of bases grows exponentially with $|J|$, but it is suitable for this experiment since the number of rays is of managable size. We explore later in this chapter alternative methodologies to generate cuts derived from the CGLP that are cheaper than fully enumerate all the bases. The CGLP produces 5 different facets. For each of these we show the configuration of the parametric octahedron that yields the corresponding cut in terms of the v, w variables:

1. Cut (T_B): $2s_1 + 2s_2 + 4s_3 + s_4 + \frac{12}{7}s_5 \geq 1$
   $v_1 = 2;\ v_2 = \frac{8}{7};\ v_3 = 0;\ v_4 = 0$
   $w_1 = 1;\ w_2 = \frac{2}{7};\ w_3 = 2;\ w_4 = 2$
2. Cut (T_B): $\frac{8}{3}s_1 + \frac{4}{3}s_2 + \frac{44}{9}s_3 + \frac{8}{9}s_4 + \frac{4}{3}s_5 \geq 1$
   $v_1 = \frac{20}{9};\ v_2 = \frac{4}{3};\ v_3 = \frac{4}{3};\ v_4 = \frac{4}{9}$
   $w_1 = \frac{8}{9};\ w_2 = 0;\ w_3 = 0;\ w_4 = \frac{16}{9}$
3. Cut (T_A): $\frac{8}{3}s_1 + 2s_2 + 4s_3 + s_4 + \frac{4}{3}s_5 \geq 1$
   $v_1 = 2;\ v_2 = \frac{4}{3};\ v_3 = 0;\ v_4 = 0$
   $w_1 = 1;\ w_2 = 0;\ w_3 = 2;\ w_4 = 2$
4. Cut (S): $\frac{8}{3}s_1 + \frac{4}{3}s_2 + 12s_3 + \frac{4}{3}s_5 \geq 1$
   $v_1 = 4;\ v_2 = \frac{4}{3};\ v_3 = \frac{4}{3};\ v_4 = 4$
   $w_1 = 0;\ w_2 = 0;\ w_3 = 0;\ w_4 = 0$
5. Cut (T_B): $2s_1 + 2s_2 + \frac{68}{7}s_3 + \frac{2}{7}s_4 + \frac{12}{7}s_5 \geq 1$
   $v_1 = \frac{24}{7};\ v_2 = \frac{8}{7};\ v_3 = 0;\ v_4 = 0$
   $w_1 = \frac{2}{7};\ w_2 = \frac{2}{7};\ w_3 = 2;\ w_4 = 2$

For each of the 5 cuts, we give its intersection cut representation in the space of x_1, x_2 variables in Figure 2.4. Cut 1 is the facet of the Integer Hull presented in [2].

Using the software CDD+ by Fukuda [34] we determined the extreme points and extreme directions of P_D for the considered example. We used as input the model given by the two original simplex rows together with the 5 generated cuts

$$
\begin{array}{llllllll}
x_1 & & -2s_1 & -1s_2 & +3s_3 & & -1s_5 & = \frac{1}{4} \\
 & x_2 & -1s_1 & -1s_2 & -2s_3 & +1s_4 & +2s_5 & = \frac{1}{2} \\
 & & 2s_1 & +2s_2 & +4s_3 & +1s_4 & +\frac{12}{7}s_5 & \geq 1 \\
 & & \frac{8}{3}s_1 & +2s_2 & +4s_3 & +1s_4 & +\frac{4}{3}s_5 & \geq 1 \\
 & & \frac{8}{3}s_1 & +\frac{4}{3}s_2 & +\frac{44}{9}s_3 & +\frac{8}{9}s_4 & +\frac{4}{3}s_5 & \geq 1 \\
 & & \frac{8}{3}s_1 & +\frac{4}{3}s_2 & +12s_3 & & +\frac{4}{3}s_5 & \geq 1 \\
 & & 2s_1 & +2s_2 & +\frac{68}{7}s_3 & +\frac{2}{7}s_4 & +\frac{12}{7}s_5 & \geq 1.
\end{array}
$$

CDD+ produced the following results:

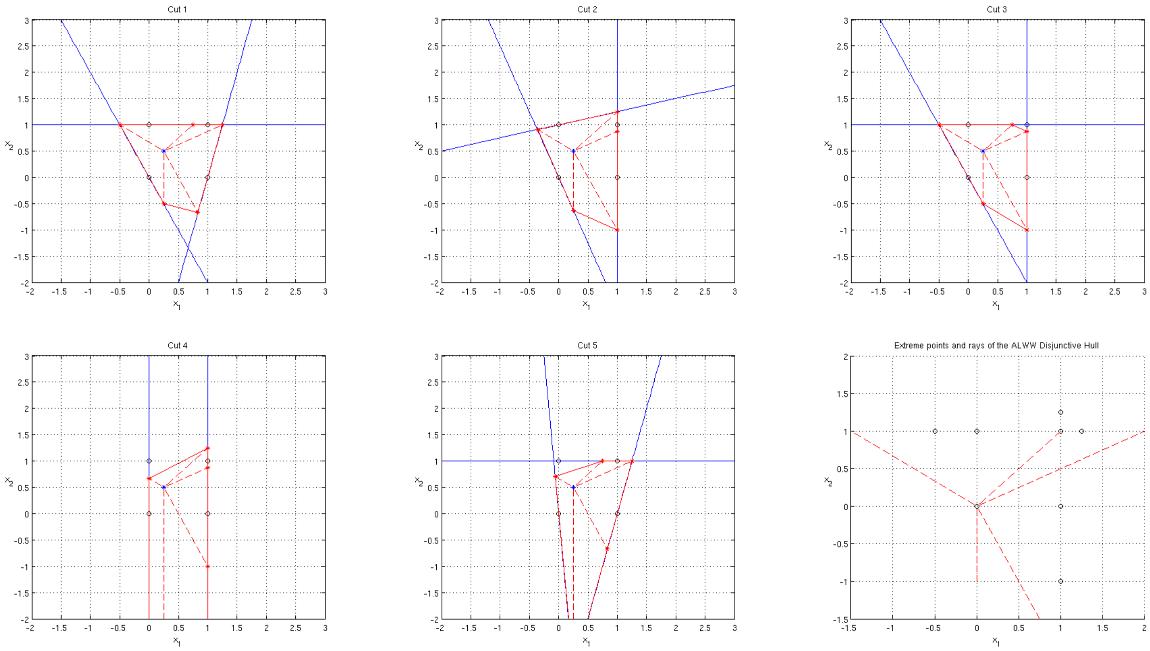


Figure 2.4: The Disjunctive Hull facets for the Andersen et al. example

Extreme points:

x_1	x_2	s_1	s_2	s_3	s_4	s_5
0	0	0	0	$\frac{1}{12}$	$\frac{2}{3}$	0
$-\frac{1}{2}$	1	0	0	$\frac{1}{4}$	0	0
0	1	0	$\frac{1}{5}$	$\frac{3}{20}$	0	0
1	-1	0	0	0	0	$\frac{3}{4}$
1	$\frac{5}{4}$	0	$\frac{3}{4}$	0	0	0
1	1	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0
1	0	$\frac{3}{8}$	0	0	$\frac{7}{8}$	0
$\frac{5}{4}$	1	$\frac{1}{2}$	0	0	0	0
1	0	$\frac{1}{5}$	0	0	0	$\frac{7}{20}$

Extreme directions:

x_1	x_2	s_1	s_2	s_3	s_4	s_5
2	1	1	0	0	0	0
1	1	0	1	0	0	0
1	-2	0	0	0	0	1
-3	2	0	0	1	0	0
0	-1	0	0	0	1	0

The extreme points and extreme rays, restricted to the space of x_1, x_2 variables, are shown in the sixth graph in Figure 2.4. Of the 5 facets of P_D, 3 are facets for P_I: cuts 1,2 and 4. Note that only cuts 1 and 2 are of the type we are after since cut 4 is a split cut and can be derived using only the tableau row corresponding to the variable x_2. Cut 3 and 5 are facets of the Disjunctive Hull by Theorem 2.2.8 using the objective coefficients $(17, 65, 73, 65, 45)$

and $(75, 1, 5, 67, 60, 1)$ respectively.

2.3.3 Facets of the Disjunctive Hull that are facets of the Integer Hull

In this section we explore the connection between the facets of P_D and facets of P_I and we illustrate procedures derived from the Lift-and-Project framework to derive facets for P_I.

Definition The K**-vertex ray** associated to a vertex t of K is the ray going from f to t. The four K-vertex rays are denoted as follows:

- q_1 is the K-vertex ray corresponding to $(0, 0)$, i.e. $q_1 = \binom{-f_1}{-f_2}$
- q_2 is the K-vertex ray corresponding to $(1, 0)$, i.e. $q_2 = \binom{1-f_1}{-f_2}$
- q_3 is the K-vertex ray corresponding to $(1, 1)$, i.e. $q_3 = \binom{1-f_1}{1-f_2}$
- q_4 is the K-vertex ray corresponding to $(0, 1)$, i.e. $q_4 = \binom{-f_1}{1-f_2}$

Definition The K**-vertex cone** associated to a vertex t of K is the set cone(r_{t-}, r_{t+}) that satisfies the following properties:

- $r_{t-}, r_{t+} \in \{r_j : j \in J\}$
- $q_t \in \text{cone}(r_{t-}, r_{t+})$
- $r_j \notin \text{cone}(r_{t-}, r_{t+})\ \forall r_j, j \in J : r_j \neq r_{t-}, r_{t+}$

If there exists a ray $r_k \in \{r_j, j \in J\}$ such that $t = f + \theta r_k$ for some $\theta > 0$ then the K-vertex cone associated to t contains the ray r_k only, i.e. $r_{t-} = r_{t+} = r_k$.

By Theorem 2.2.8 all the relevant inequalities of the Disjunctive Hull can be derived from at most 2^q (=4 in this case) rays and lifting the resulting inequality into the full space. Note that basic feasible solutions to the CGLP can sometimes be obtained even from 3 or 2 rays. In these cases the solutions we get are primal degenerate, i.e. some basic variables are equal to 0 as shown in the following example

Example

$$\begin{array}{lllll} x_1 & = \frac{1}{4} & +2s_1 & -3s_2 & +1s_3 \\ x_2 & = \frac{1}{2} & -1s_1 & -2s_2 & +2s_3 \\ s_j & \geq 0 & j \in J & & \\ x_1, x_2 \in \mathbb{Z} & & & & \end{array} \tag{2.3.9}$$

The CGLP (2.3.5) associated to (2.3.9) contains 16 constraints and 23 variables (of which 12 slack variables t^i_j associated to the α constraint corresponding to the orthant-cone i and the ray j). The following is a basic feasible solution to (2.3.5). Only basic variables are shown, the other variables are equal to 0:

$\alpha_1 = 2; \alpha_2 = 4; \alpha_3 = \frac{12}{7}; v_4 = 2; v_3 = \frac{8}{7}; w_1 = 2; w_2 = 2; w_4 = 1; w_3 = \frac{2}{7}; t^1_1 = 0; t^1_2 = 0;$
$t^1_3 = \frac{40}{7}; t^2_3 = \frac{40}{7}; t^3_1 = 7; t^3_3 = \frac{12}{7}; t^4_2 = 8.$

Suppose now that the variable s_3 and the associated ray is dropped. The new CGLP contains 12 constraints and 19 variables (of which 8 slack variables t^i_j defined as before). The following is a basic feasible solution to (2.3.5). Only basic variables are shown, the other variables are equal to 0:

$$\alpha_1 = 2; \alpha_2 = 4; w_1 = 2; w_2 = 2; w_3 = 2; w_4 = 2; t^1_1 = 0; t^1_2 = 0; t^3_1 = 4; t^3_2 = 8; t^4_1 = 4; t^4_2 = 8.$$

The next Proposition (that follows from Theorem 2.2.11 for $q = 2$) gives a sufficient condition for a facet of P_D to be a facet of P_I that we use later when we present our procedure to generate facets of P_I.

Proposition 2.3.1. *A facet $\alpha s \geq 1$ of P_D that is not a split cut and is generated by the CGLP from a MIP instance with 4 rays, is a facet of P_I if the following condition is satisfied by the values of α:*

$$\exists \lambda_t, 0 \leq \lambda_t \leq 1 : t = f + \lambda_t \frac{r_{t-}}{\alpha_{t-}} + (1 - \lambda_t) \frac{r_{t+}}{\alpha_{t+}}, \ \forall t \in \{(0,0), (1,0), (1,1), (0,1)\} \quad (2.3.10)$$

The condition is equivalent to require that each vertex t of K must lie on the segment between the points given by the intersection of the parametric octahedron boundary and the rays r_{t-}, r_{t+}.

Proof. If the values α satisfy the condition (2.3.10), then the inequality $\alpha s \geq 1$ holds at equality for all four vertices of K which are four affinely independent in the space (x, s) with x integral. □

Let P'_I be the Integer Hull given by dropping all but 4 non-basic variables of P_I. Proposition 2.3.1 gives a condition for instances of P'_I, in 2.3.4 we discuss how a facet for the P_I restricted to 4 rays is lifted to a facet for P_I.

In the following we are going to discuss triangle and quadrilateral facets that can be obtained using the Disjunctive Hull approach. Our discussion does not exhaustively consider every type of facet for P_I that is also a facet for P_D, instead we focus on two subclasses of those that can be obtained without the computational effort required to solve the CGLP. The two subclasses are:

1. *Non-degenerate Quadrilateral facets*, are obtained from quadrilateral parametric octahedra,

2. *Degenerate Quadrilateral facets*, are obtained from triangular parametric octahedra.

Non-degenerate quadrilateral facets

For these facets the set L_α and the parametric octahedron defined by the values of v, w coincide. This is equivalent to the following: each vertex of the quadrilateral lies on two adjacent facets of the parametric octahedron. In this type of configuration every edge of K is intersected by at least one ray. Given four rays r_A, r_B, r_C, r_D intersecting respectively the edges $\{(0,0),(1,0)\}$; $\{(1,0),(1,1)\}$; $\{(1,1),(0,1)\}$ and $\{(0,1),(0,0)\}$ the following inequalities of the CGLP must hold at equality:

$$\begin{array}{llll}
\alpha_A & +r_A^1 v_1 & +r_A^2 w_1 & \geq 0 \\
\alpha_A & -r_A^1 v_2 & +r_A^2 w_2 & \geq 0 \\
\alpha_B & -r_B^1 v_2 & +r_B^2 w_2 & \geq 0 \\
\alpha_B & -r_B^1 v_3 & -r_B^2 w_3 & \geq 0 \\
\alpha_C & -r_C^1 v_3 & -r_C^2 w_3 & \geq 0 \\
\alpha_C & +r_C^1 v_4 & -r_C^2 w_4 & \geq 0 \\
\alpha_D & +r_D^1 v_4 & -r_D^2 w_4 & \geq 0 \\
\alpha_D & +r_D^1 v_1 & +r_D^2 w_1 & \geq 0
\end{array}$$

The system has 8 inequalities and 4 equalities given by the normalization. For the parametric octahedron to be a quadrilateral, all v, w have to be positive. Hence all 8 inequalities have to hold at equality. Therefore we have the following system of 12 equations in 12 variables (α, v, w):

$$\begin{array}{llll}
\alpha_A & +r_A^1 v_1 & +r_A^2 w_1 & = 0 \\
\alpha_A & -r_A^1 v_2 & +r_A^2 w_2 & = 0 \\
\alpha_B & -r_B^1 v_2 & +r_B^2 w_2 & = 0 \\
\alpha_B & -r_B^1 v_3 & -r_B^2 w_3 & = 0 \\
\alpha_C & -r_C^1 v_3 & -r_C^2 w_3 & = 0 \\
\alpha_C & +r_C^1 v_4 & -r_C^2 w_4 & = 0 \\
\alpha_D & +r_D^1 v_4 & -r_D^2 w_4 & = 0 \\
\alpha_D & +r_D^1 v_1 & +r_D^2 w_1 & = 0 \\
f_1 v_1 & +f_2 w_1 & = 1 & \\
(1-f_1) v_2 & +f_2 w_2 & = 1 & \\
(1-f_1) v_3 & +(1-f_2) w_3 & = 1 & \\
f_1 v_4 & +(1-f_2) w_4 & = 1. &
\end{array} \tag{2.3.11}$$

This system imposes the condition that each of the 4 rays must intersect one vertex of the parametric octahedron. If such a parametric octahedron exists, i.e. the solution to (2.3.11) satisfies $\alpha, v, w \geq 0$. In Figure 2.5 an example of a facet from a Quadrilateral parametric octahedron is shown.

Cornuejols and Margot in [26] study intersection cuts obtained from two rows of the simplex tableau and give conditions for the cuts to be facets of the Integer Hull Quadrilateral cuts. The following is their Theorem 9:

Theorem 2.3.2. *(Cornuejols and Margot, 2009) Consider a maximal lattice-free quadrilateral with vertices p_j, integral point q_j on edge $p_j p_{j+1}$ (indices taken modulo 4) an corner rays $r_j, j = 1, \dots, 4$. The corresponding quadrilateral inequality defines a facet of the Integer Hull if and only if there is no $t \in \mathbb{R}_+$ such that the integral point k_j divides the edge joining p_j to p_{j+1} in a ratio t for odd j and in a ratio $1/t$ for even j, i.e.*

$$\frac{||k_j - p_j||}{||k_j - p_{j+1}||} \begin{cases} t \text{ for } j = 1, 3 \\ 1/t \text{ for } j = 2, 4 \end{cases} \tag{2.3.12}$$

As stated in [26], the ratio condition (2.3.12) is equivalent to the following condition on

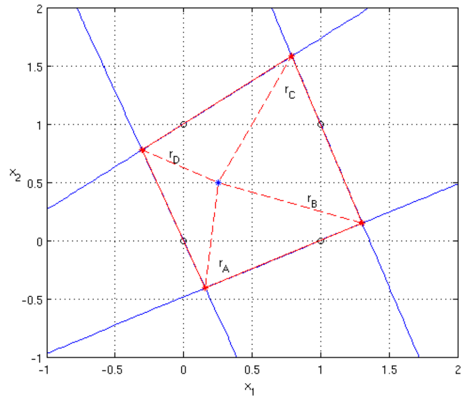


Figure 2.5: Facet from a Quadrilateral P_{octa}

$p_j, k_j, j = 1, \ldots, 4$:

$$
\begin{array}{lllll}
+\beta p_1 & & & +(1-\beta)p_4 & = k_1 \\
\beta p_1 & +(1-\beta)p_2 & & & = k_2 \\
 & +(1-\beta)p_2 & +\beta p_3 & & = k_3 \\
 & & +\beta p_3 & +(1-\beta)p_4 & = k_4
\end{array}
\tag{2.3.13}
$$

for some $\beta : 0 < \beta < 1$.

For the type of quadrilaterals we generate with the Disjunctive Hull framework the integral points $k_j, j = 1, \ldots, 4$ are the vertices of the unit cube K indexed as

$$
\begin{array}{ll}
k_1 & = (0,0). \\
k_2 & = (1,0); \\
k_3 & = (1,1); \\
k_4 & = (0,1);
\end{array}
\tag{2.3.14}
$$

Suppose r_A, r_B, r_C, r_D are such that they produce a quadrilateral that satisfies (2.3.13). By Theorem 2.3.2, the quadrilateral cut is not a facet of the integer hull P_I. We now link together the quadrilateral cuts obtained from the Disjunctive Hull and the previous result.

Proposition 2.3.3. *A set of 4 rays for which the ratio condition (2.3.12) is satisfied, does not yield a basic solution to the CGLP corresponding to a quadrilateral cut.*

Proof. After eliminating $\alpha_A, \alpha_B, \alpha_C, \alpha_D$ from (2.3.11) we obtain the system of 8 equations in 8 variables v, w

$$\begin{array}{lll} +r_A^1 v_1 + r_A^1 v_2 & +r_A^2 w_1 - r_A^2 w_2 & = 0 \\ -r_B^1 v_2 + r_B^1 v_3 & +r_B^2 w_2 + r_B^2 w_3 & = 0 \\ -r_C^1 v_3 - r_C^1 v_4 & -r_C^2 w_3 + r_C^2 w_4 & = 0 \\ -r_D^1 v_1 + r_D^1 v_4 & -r_D^2 w_1 - r_D^2 w_4 & = 0 \\ f_1 v_1 & +f_2 w_1 & = 1 \\ (1-f_1) v_2 & +f_2 w_2 & = 1 \\ (1-f_1) v_3 & +(1-f_2) w_3 & = 1 \\ f_1 v_4 & +(1-f_2) w_4 & = 1. \end{array} \tag{2.3.15}$$

Assuming unit length rays we have

$$p_j = f + r_j, j = 1, \ldots, 4. \tag{2.3.16}$$

Substituting (2.3.14) and (2.3.16) in (2.3.13) we get that when the ratio condition holds, the rays must satisfy

$$\begin{array}{l} r_B^1 = \frac{f_1 - 1 + \beta r_A^1}{\beta - 1} \\ r_B^2 = \frac{f_2 + \beta r_A^2}{\beta - 1} \\ r_C^1 = r_A^1 \\ r_C^2 = \frac{1 + \beta r_A^2}{\beta} \\ r_D^1 = \frac{f_1 + \beta r_A^1}{\beta - 1} \\ r_D^2 = r_B^2. \end{array} \tag{2.3.17}$$

After substituting the expressions for r_j's, the coefficient matrix of (2.3.15) becomes

$$\begin{pmatrix} r_A^1 & r_A^1 & 0 & 0 & r_A^2 & -r_A^2 & 0 & 0 \\ 0 & -\frac{f_1-1+\beta r_A^1}{\beta-1} & \frac{f_1-1+\beta r_A^1}{\beta-1} & 0 & 0 & \frac{f_2+\beta r_A^2}{\beta-1} & \frac{f_2+\beta r_A^2}{\beta-1} & 0 \\ 0 & 0 & -r_A^1 & -r_A^1 & 0 & 0 & -\frac{1+\beta r_A^2}{\beta} & \frac{1+\beta r_A^2}{\beta} \\ -\frac{f_1+\beta r_A^1}{\beta-1} & 0 & 0 & \frac{f_1+\beta r_A^1}{\beta-1} & -\frac{f_2+\beta r_A^2}{\beta-1} & 0 & 0 & -\frac{f_2+\beta r_A^2}{\beta-1} \\ f_1 & 0 & 0 & 0 & f_2 & 0 & 0 & 0 \\ 0 & 1-f_1 & 0 & 0 & 0 & f_2 & 0 & 0 \\ 0 & 0 & 1-f_1 & 0 & 0 & 0 & 1-f_2 & 0 \\ 0 & 0 & 0 & f_1 & 0 & 0 & 0 & 1-f_2 \end{pmatrix} \tag{2.3.18}$$

The matrix (2.3.18) is singular since, for example, the last row can be obtained from the rows $j = 1, \ldots, 7$ using multipliers μ_j where $\mu_1 = \beta; \mu_2 = \beta - 1; \mu_3 = \beta; \mu_4 = \beta - 1; \mu_5 = 1; \mu_6 = -1; \mu_7 = 1$. Since the coefficient matrix is singular, it cannot be a basic solution to the CGLP associated with the 4 rays that yields the non-facet quadrilateral cuts. □

Degenerate quadrilateral facets

For these facets the set L_α and the parametric octahedron defined by the values of v, w do not coincide. The configuration of the rays r_A, r_B, r_C, r_D differs from before in that now one edge of K is not intersected by any of the four rays. Assume that this edge is $\{(0,0),(1,0)\}$ and let r_A, r_B be the rays defining the K-vertex cone for $(0,0)$ and $(1,0)$. The reasoning

that follows can be easily extended to the other 3 edges of K by symmetry.
One way to ensure that we produce a facet for P_I we enforce the following conditions from the Proposition 2.3.1:

$$\begin{array}{l}\exists\lambda_{(0,0)}, 0 \leq \lambda_{(0,0)} \leq 1 : (0,0) = f + \lambda_{(0,0)}\frac{r_A}{\alpha_A} + (1-\lambda_{(0,0)})\frac{r_B}{\alpha_B}\\ \exists\lambda_{(1,0)}, 0 \leq \lambda_{(1,0)} \leq 1 : (1,0) = f + \lambda_{(1,0)}\frac{r_A}{\alpha_A} + (1-\lambda_{(1,0)})\frac{r_B}{\alpha_B}.\end{array} \tag{2.3.19}$$

This condition can only be satisfied if the two facets of the parametric octahedron corresponding to the vertices $(0,0)$ and $(1,0)$ coincide, thus making the parametric octahedron a triangle of type 1 or 2. For a triangle of type 1 to yield a facet for P_I one of the rays r_A or r_B must intersect either $(0,0)$ or $(1,0)$. This is a special case and is therefore not considered here, since it is less recurring. We assume that the parametric octahedron defines a triangle of type 2.

We have that the facets of P_{octa} going through $(0,0)$ and $(1,0)$ are determined by solving the system of 6 equations in the variables $\alpha_A, \alpha_B, v_1, w_1, v_2, w_2$.

$$\begin{array}{llll}\alpha_A & +r_A^1 v_1 & +r_A^2 w_1 & = 0\\ \alpha_A & -r_A^1 v_2 & +r_A^2 w_2 & = 0\\ \alpha_B & +r_B^1 v_1 & +r_B^2 w_1 & = 0\\ \alpha_B & -r_B^1 v_2 & +r_B^2 w_2 & = 0\\ f_1 v_1 & +f_2 w_1 & = 1 & \\ (1-f_1)v_2 & +f_2 w_2 & = 1 & \end{array} \tag{2.3.20}$$

Let $\bar{\alpha}_A, \bar{\alpha}_B, \bar{v}_1, \bar{w}_1, \bar{v}_2, \bar{w}_2$ denote the solution to (2.3.20).

We need to consider 3 different cases, according to the positions of the rays r_C, r_D. Let v_A, v_B, v_C, v_D denote the intersection points of the rays r_A, r_B, r_C, r_D with the parametric octahedron.

1. The rays r_C and r_D both intersect the edge $\{(1,1),(0,1)\}$. In this case the points we can have the following possibile configurations

   (a) v_C lies on the parametric octahedron facet corresponding to $(0,1)$ and v_D lies on the facet corresponding to $(1,1)$ (an example is given in Figure 2.6(a));

   (b) both v_C and v_D lie on the parametric octahedron facet corresponding to $(0,1)$ (an example is given in Figure 2.6(b));

   (c) both v_C and v_D lie on the parametric octahedron facet corresponding to $(1,1)$ (an example is given in Figure 2.6(c)).

   We determine the values for v_3, w_3 for which the facet corresponding to $(1,1)$ intersects r_B in v_B, and the values for v_4, w_4 for which the facet corresponding to $(0,1)$ intersects r_A in v_A by solving the system of linear equations in 4 variables

$$\begin{array}{lll}-r_B^1 v_3 & -r_B^2 w_3 & = -\bar{\alpha}_B\\ +r_A^1 v_4 & -r_A^2 w_4 & = -\bar{\alpha}_A\\ (1-f_1)v_3 & +(1-f_2)w_3 & = 1\\ f_1 v_4 & +(1-f_2)w_4 & = 1.\end{array}$$

Let $\bar{v}_3, \bar{w}_3, \bar{v}_4, \bar{w}_4$ be the solution to the system above. To determine the value of α_C and α_D we just need to apply a similar procedure to that described at 2.3.4 for Lifting. The values for α_C and α_D are computed as follows

$$\begin{aligned} \alpha_C = &\quad \max\{ \quad r_C^1 \bar{v}_3 + r_C^2 \bar{w}_3 \quad ; \quad -r_C^1 \bar{v}_4 + r_C^2 \bar{w}_4 \quad \}, \\ \alpha_D = &\quad \max\{ \quad r_D^1 \bar{v}_3 + r_D^2 \bar{w}_3 \quad ; \quad -r_D^1 \bar{v}_4 + r_D^2 \bar{w}_4 \quad \}. \end{aligned}$$

In all three configurations we obtain a facet for P_I. For case (a) the facet is generated from the four rays and can be easily seen that all the four vertices of K can be expressed as convex combination of the points v_A, v_B, v_C, v_D. For the case (b) we can still exhibit four integral points but this time not all four vertices of K are covered. In this circumstance we have that the point $(0, 1)$ can be expressed as a convex combination of v_A and v_C and also as a convex combination of the points v_A and v_D. Similarly for case (c) the vertex of the cube $(1, 1)$ can be obtained by two convex combinations, the first is by the points v_B and v_D and the second is other by the points v_B and v_D.

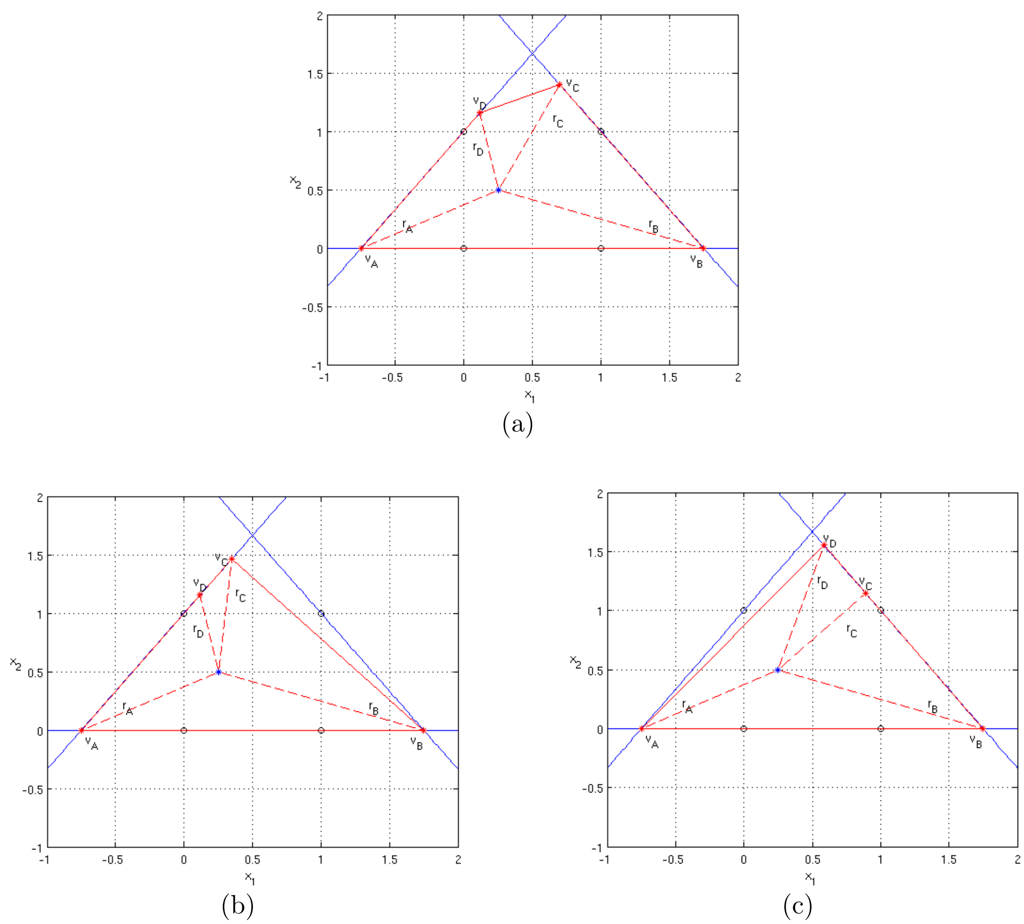


Figure 2.6: Degenerate Quadrilateral configurations for case 1

2. The ray r_C intersects the edge $\{(1, 1), (0, 1)\}$ and r_D intersects the edge $\{(0, 1), (0, 0)\}$.

In this case the points we can have the following possibile configurations

(a) v_D lies on the parametric octahedron facet corresponding to $(0, 1)$ (an example is given in Figure 2.7(a));

(b) v_D lies on the parametric octahedron facet corresponding to $(0, 0)$ (an example is given in Figure 2.7(b)).

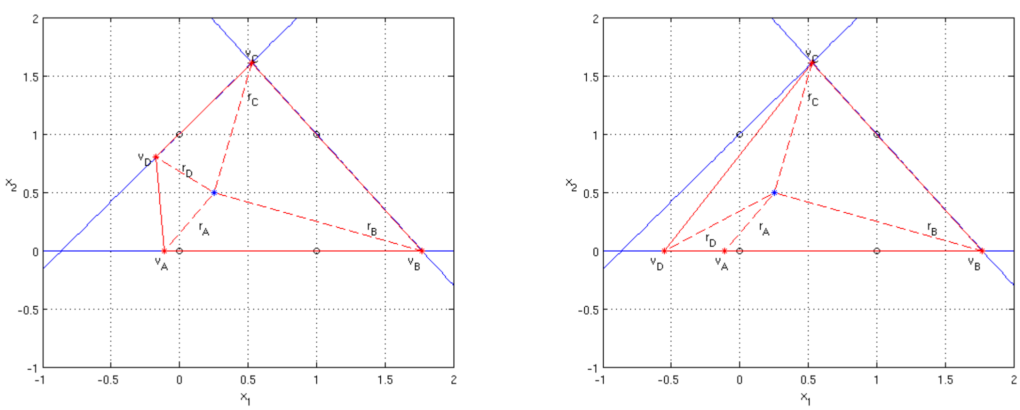
We determine the values for v_3, w_3 for which the facet corresponding to $(1, 1)$ intersects r_B in v_B, and the values for v_4, w_4 for which the facet corresponding to $(0, 1)$ intersects both r_C and the facet corresponding to $(1, 1)$ in v_C by solving the system of linear equations in 5 variables

$$\begin{array}{lll} -r_B^1 v_3 & -r_B^2 w_3 & = -\bar{\alpha}_B \\ \alpha_C - r_C^1 v_3 & -r_C^2 w_3 & = 0 \\ \alpha_C + r_C^1 v_4 & -r_C^2 w_4 & = 0 \\ (1-f_1)v_3 & +(1-f_2)w_3 & = 1 \\ f_1 v_4 & +(1-f_2)w_4 & = 1. \end{array}$$

Let $\bar{\alpha}_C, \bar{v}_3, \bar{w}_3, \bar{v}_4, \bar{w}_4$ be the solution to the system above. To determine the value of α_D we just need to apply a similar procedure to that described at 2.3.4 for Lifting. The value for α_D is computed as follows

$$\alpha_D = \max\{ \; -r_D^1\bar{v}_3 - r_D^2\bar{w}_3 \;\; ; \;\; -r_D^1\bar{v}_4 + r_D^2\bar{w}_4 \; \}.$$

Following the same reasoning used for case 1, we can show that the considered configurations for case 2 yield facets of P_I.



(a) (b)

Figure 2.7: Degenerate Quadrilateral configurations for case 2

3. The ray r_C intersects the edge $\{(1, 0), (1, 1)\}$ and r_D intersects the edge $\{(1, 1), (0, 1)\}$ (symmetric of case 2). In this case we can hae the following possible configurations

(a) v_C lies on the parametric octahedron facet corresponding to $(1,1)$ (an example is given in Figure 2.8(a));

(b) v_C lies on the parametric octahedron facet corresponding to $(1,0)$ (an example is given in Figure 2.8(b)).

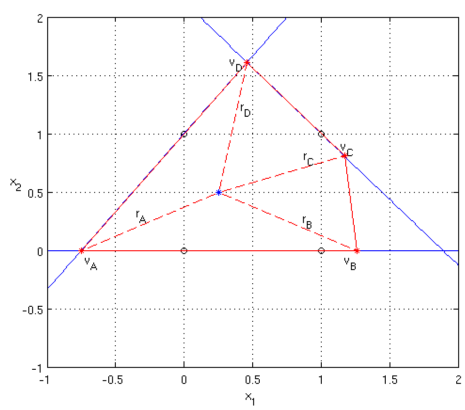
We determine the values for v_4, w_4 for which the facet corresponding to $(0,1)$ intersects r_A in v_A, and the values for v_3, w_3 for which the facet corresponding to $(1,1)$ intersects both r_D and the facet corresponding to $(0,1)$ in v_D by solving the system of linear equations in 5 variables

$$\begin{array}{lll} r_B^1 v_4 & -r_B^2 w_4 & = -\bar{\alpha}_B \\ \alpha_D - r_D^1 v_3 & -r_D^2 w_3 & = 0 \\ \alpha_D + r_D^1 v_4 & -r_D^2 w_4 & = 0 \\ (1-f_1)v_3 & +(1-f_2)w_3 & = 1 \\ f_1 v_4 & +(1-f_2)w_4 & = 1. \end{array}$$

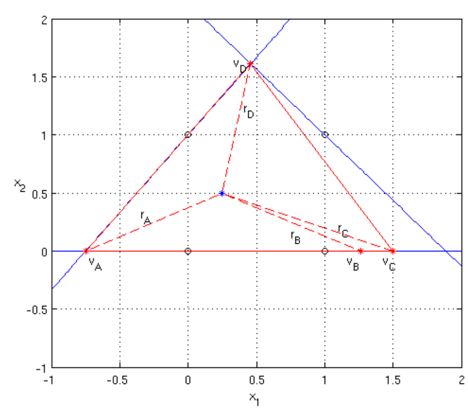
Let $\bar{\alpha}_D, \bar{v}_3, \bar{w}_3, \bar{v}_4, \bar{w}_4$ be the solution to the system above. To determine the value of α_C we just need to apply a similar procedure to that described at 2.3.4 for Lifting. The value for α_C is computed as follows

$$\alpha_C = \max\{ \quad r_C^1 \bar{v}_2 - r_C^2 \bar{w}_2 \quad ; \quad r_C^1 \bar{v}_3 + r_C^2 \bar{w}_3 \quad \}.$$

Following the same reasoning used for case 1, we can show that the considered configurations for case 3 yield facets of P_I.



(a)



(b)

Figure 2.8: Degenerate Quadrilateral configurations for case 3

As for the case of Non-Degenerate cuts, the Proposition 2.3.1 ensures that the cut we generate after lifting the remaining rays is a facet for P_I.

There exist other types of Degenerate Quadrilateral cuts that cannot be derived by the systems above. Previously we restricted the choice of the 4 rays by requiring r_A, r_B to satisfy (2.3.19). But facets for P_I might also be derived when r_A, r_B satisfy the following more general condition:

$$\begin{array}{l} \exists\lambda_1, 0 \leq \lambda_1 \leq 1 : (y_1, y_2) = f + \lambda_1 \frac{r_A}{\alpha_A} + (1-\lambda_1)\frac{r_B}{\alpha_B} \\ \exists\lambda_2, 0 \leq \lambda_2 \leq 1 : (y_1', y_2') = f + \lambda_2 \frac{r_A}{\alpha_A} + (1-\lambda_2)\frac{r_B}{\alpha_B}. \end{array} \tag{2.3.21}$$

where $(y_1, y_2), (y_1', y_2') \in \mathbb{Z}^2$ are two adjacent points in the lattice $\mathbb{Z}^2$, i.e. $(y_1, y_2) \in \mathbb{Z}^2$ and either $y_1' = y_1 \wedge y_2' = y_2 + 1$ or $y_1' = y_1 + 1 \wedge y_2' = y_2$. These points might be general integer points, different than the vertices of K.
In this chapter we do not study further this more general class of Degenerate Quadrilateral facets.

Outline of the procedures to generate facets of P_I

In this section we give the pseudo-code for two procedures that can be used to produce valid facets of the Integer Hull. The procedures are parametrized with the rays $r_j, j \in J$ and the fractional solution $f = (f_1, f_2)$.

The procedures will select 4 rays among the $|J|$ available that satisfy some specified criteria and build a system of linear equations on (α, v, w) variables that is then solved. If the solution is feasible, i.e. $\alpha, v, w \geq 0$, then a facet of the Integer Hull is obtained. Lifting of the remaining rays is then applied to it.
To simplify notation let CGLP(r_j, i) with $j \in J$, $i \in \{1 \dots 4\}$ identify the inequality in the MIP CGLP corresponding to ray r_j and the facet of the parametric octahedron going through the i-th vertex of K and let NORM(i) with $i \in \{1 \dots 4\}$ be the normalization constraint associated with the i-th facet of the parametric octahedron of the MIP CGLP.

In the procedure *GenerateNonDegenerateQuadrilateralFacet* we describe how to obtain a Non-degenerate Quadrilateral facet and in the procedure *GenerateDegenerateQuadrilateralFacet* we describe how to obtain a Degenerate Quadrilateral facet.

Every time we run one of the two procedures and a cut is generated, we can remove one of the chosen rays from the set $\{r_j, j \in J\}$ so that we ensure that the same cut will not be generated in future calls. This does not allow to consider all the possibilities, but it suffices for an heuristic procedure which aims to generate a constant number of facets. Moreover randomization in the ray selection can also be added.

GenerateNonDegenerateQuadrilateralFacet$(\{r_j, j \in J\}\ ,\ f)$

$r_A, r_B, r_C, r_D \leftarrow NULL$
$i \leftarrow 1;$
for $i \leftarrow 1$ **up to** $|J|$
        do
            if $(r_A == NULL) \wedge (r_i \in \text{int}(\text{cone}(q_1, q_2)))$
                then $r_A \leftarrow r_i$
            if $(r_B == NULL) \wedge (r_i \in \text{int}(\text{cone}(q_2, q_3)))$
                then $r_B \leftarrow r_i$
            if $(r_C == NULL) \wedge (r_i \in \text{int}(\text{cone}(q_3, q_4)))$
                then $r_C \leftarrow r_i$
            if $(r_D == NULL) \wedge (r_i \in \text{int}(\text{cone}(q_4, q_1)))$
                then $r_D \leftarrow r_i$
if $(r_A == NULL) \vee (r_B == NULL) \vee (r_C == NULL) \vee (r_D == NULL)$
    then error ("There are not enough rays satisfying the required conditions")
    else Solve the system in the variables $(\alpha_A, \alpha_B, \alpha_C, \alpha_D, v, w)$ and the equations:
            CGLP$(r_A, 1)$; CGLP$(r_A, 2)$
            CGLP$(r_B, 2)$; CGLP$(r_B, 3)$
            CGLP$(r_C, 3)$; CGLP$(r_C, 4)$
            CGLP$(r_D, 4)$; CGLP$(r_D, 1)$
            NORM(1);NORM(2);NORM(3);NORM(4);
            if system has a solution
                then if the solution satisfies $(v, w) \geq 0$
                        then return (α, v, w)
                        else error ("The chosen rays do not produce a cut")
                else error ("System of equation does not have a solution")

GenerateDegenerateQuadrilateralFacet$(\{r_j, j \in J\}\ ,\ f)$

1 $r_A, r_B, r_C, r_D \leftarrow NULL$
2 $y \leftarrow$ a random number from $\{1 \ldots 4\}$
3 $q'_1 \leftarrow q_y$
4 $q'_2 \leftarrow q_{y+1}$
5 $q'_3 \leftarrow q_{y+2}$
6 $q'_4 \leftarrow q_{y+3}$
7 $r_A, r_B \leftarrow$ two rays that satisfy the conditions:
  1) $q'_1, q'_2 \in \mathrm{int}(\mathrm{cone}(r_A, r_B))$
  2) $q'_3, q'_4 \notin \mathrm{int}(\mathrm{cone}(r_A, r_B))$
  3) the sign patterns of r_A and r_B differ in exactly one component
8 $r_C, r_D \leftarrow$ two rays that satisfy one of the following sets of conditions:
  a) a$_1$) $r_C, r_D \in \mathrm{int}(\mathrm{cone}(q'_3, q'_4))$
  b) $\left\{ \begin{array}{l} \text{b}_1)\ r_C \in \mathrm{int}(\mathrm{cone}(q'_3, q'_4)) \\ \text{b}_2)\ r_D \in \mathrm{int}(\mathrm{cone}(q'_4, q'_1)) \end{array} \right.$
  c) $\left\{ \begin{array}{l} \text{c}_1)\ r_C \in \mathrm{int}(\mathrm{cone}(q'_2, q'_3)) \\ \text{c}_2)\ r_D \in \mathrm{int}(\mathrm{cone}(q'_3, q'_4)) \end{array} \right.$

9 **if** $(r_A == NULL) \vee (r_B == NULL) \vee (r_C == NULL) \vee (r_D == NULL)$
10   **then error** ("There are not enough rays satisfying the required conditions")
11 **if** at step 8 the chosen rays satisfy conditions in a)
12   **then** Solve the system in the 10 variables $(\alpha_A, \alpha_B, v, w)$ with the 10 equations:
  CGLP$(r_A, 1')$; CGLP$(r_A, 2')$; CGLP$(r_A, 4')$;
  CGLP$(r_B, 1')$; CGLP$(r_B, 2')$; CGLP$(r_B, 3')$;
  NORM(1);NORM(2);NORM(3);NORM(4);
  and determine the value of α_C, α_D by lifting.
13 **if** at step 8 the chosen rays satisfy conditions in b)
14   **then** Solve the system in the 11 variables $(\alpha_A, \alpha_B, \alpha_C, v, w)$ with the 11 equations:
  CGLP$(r_A, 1')$; CGLP$(r_A, 2')$;
  CGLP$(r_B, 1')$; CGLP$(r_B, 2')$; CGLP$(r_B, 3')$
  CGLP$(r_C, 3')$; CGLP$(r_C, 4')$;
  NORM(1);NORM(2);NORM(3);NORM(4);
  and determine the value of α_D by lifting.
15 **if** at step 8 the chosen rays satisfy conditions in c)
16   **then** Solve the system in the 11 variables $(\alpha_A, \alpha_B, \alpha_D, v, w)$ with the 11 equations:
  CGLP$(r_A, 1')$; CGLP$(r_A, 2')$; CGLP$(r_A, 4')$
  CGLP$(r_B, 1')$; CGLP$(r_B, 2')$;
  CGLP$(r_D, 3')$; CGLP$(r_D, 4')$;
  NORM(1);NORM(2);NORM(3);NORM(4);
  and determine the value of α_C by lifting.
17 **return** (α, v, w)

2.3.4 Lifting

The procedures given in 2.3.3 select rays that yield feasible configurations of the parametric octahedron that can be used to derive facets for P_I. Together with the values for v, w defining the parametric octahedron we also compute the coefficients of the facet for the 4 chosen rays. Lifting must be applied to determine the coefficients of the facet generated by the parametric octahedron for the remaining rays.

Definition For a triangle or quadrilateral facet $\alpha s \geq 1$ of P_I we call the rays required to produce it **generating rays** and the remaining rays **lifted rays**.

Definition Let $r_k, k \in T$ be a collection of generating rays for a facet $\alpha s \geq 1$ and let $\bar{v}, \bar{w}$ be the values for the associated parametric octahedron $P_{\text{octa}}(\bar{v}, \bar{w})$. Let $F_i, i \in \{1 \dots 4\}$ identify the facet of $P_{\text{octa}}(\bar{v}, \bar{w})$ corresponding to the i-th vertex of K. We say that the parametric octahedron is **flexible** if for some $i \in \{1 \dots 4\}$ there exists $v'_i, w'_i \geq 0$ with $v'_i \neq \bar{v}_i, w'_i \neq \bar{w}_i$ and satisfying the i-th β-constraint, the parametric octahedron with the modified parameters produces the same coefficients for $r_k, k \in T$, otherwise we say that is **rigid**.

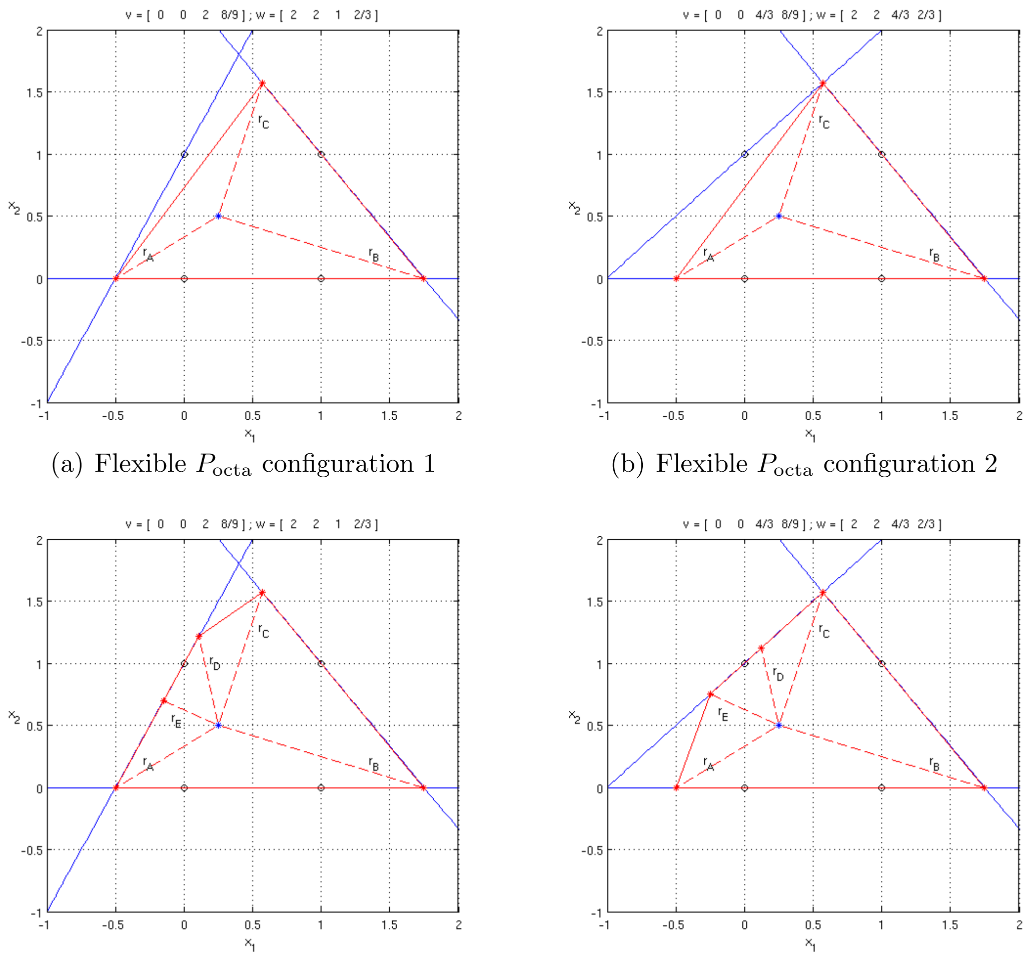
Simply put, a parametric octahedron is flexible when one of its facets can be tilted so that the cut produced in the space of its generating rays remains the same. In Figures 2.9(a) and 2.9(b) we present a cut that exhibits this property. The parametric octahedron facet corresponding to the vertex $(0, 1)$ can be tilted as shown, and the two different parametric octahedrons yield the same cut. Two rigid parametric configurations are shown in Figures 2.9(c) and 2.9(d), for these cases no facet of the parametric octahedron can be tilted without modifying the α coefficients. The values for the v, w parameters are shown on top of each figure.

For the case of a flexible parametric octahedron the sequence used in lifting matters, while it does not for the case of a rigid parametric octahedron. Consider the rays r_A, r_B, r_C and the flexible parametric octahedrons they define in Figures 2.9(a), 2.9(b). In Figures 2.9(c), 2.9(d) two different cuts are obtained depending on the order used to lift rays r_D and r_E. In Figure 2.9(c) the ray r_D is lifted first and r_E last, while in Figure 2.9(d) the ray r_D is lifted last and r_E first. Note that after r_D or r_E had been lifted the parametric octahedrons become rigid.

Given $\bar{v}, \bar{w}$ defining a rigid parametric octahedron, the lifted coefficient α_j for a ray r_j is computed as follows: let $\bar{a}_j$ be the maximum value such that the point $p = f + \bar{a}_j r_j$ belongs to $P_{\text{octa}}(\bar{v}, \bar{w})$, and set $\bar{\alpha}_j = \frac{1}{\bar{a}_j}$. By Proposition 2.2.2 The coefficient α_j for the j-th ray is given by

$$\begin{aligned} \bar{\alpha}_j = \max\{ \quad & \bar{\alpha}_j^1 = -r_j^1 \bar{v}_1 - r_j^2 \bar{w}_1; \\ & \bar{\alpha}_j^2 = +r_j^1 \bar{v}_2 - r_j^2 \bar{w}_2; \\ & \bar{\alpha}_j^3 = +r_j^1 \bar{v}_3 + r_j^2 \bar{w}_3; \\ & \bar{\alpha}_j^4 = -r_j^1 \bar{v}_4 + r_j^2 \bar{w}_4 \}. \end{aligned} \tag{2.3.22}$$

Proposition 2.3.4. *A basic solution to the CGLP for a 2-row instance with n rays can be lifted to a basic solution to the same instance amended with an extra ray r_{n+1} in which the new variable α_{n+1} is given by (2.3.22) and all the other variables keep the same values.*



(a) Flexible P_{octa} configuration 1

(b) Flexible P_{octa} configuration 2

(c) Rigid P_{octa} configuration 1

(d) Rigid P_{octa} configuration 2

Figure 2.9: Flexible and rigid parametric octahedron configurations

Proof. Adding the new ray amends the CGLP built from the original 2-row instance with 4 new constraints, one per orthant-cone (with relative slack variables $t^i_{n+1}, i \in \{1, \dots, 4\}$), and a new α_{n+1} variable unrestricted in sign. Let $k \in \{1, \dots, 4\}$ be an index for which $\bar{\alpha}_{n+1}$ attains its max in (2.3.22). The new basis is constructed from the old basis plus the variable α_{n+1} and the 3 slack variables associated to the new constraints of all orthant cones but the k-th one. Letting α_{n+1} be equal to the expression (2.3.22), and $t^i_{n+1} = \alpha_{n+1} - \alpha^i_{n+1}, i \in \{1, \dots, 4\} \setminus k$, yields a basic feasible solution. $\square$

Theorem 2.2.11 guarantees that using (2.3.22) the lifted inequality from a facet of the integer hull is still a facet of the integer hull with the added ray.

2.3.5 Strengthening

Given a facet $\alpha s \geq 1$ computed using the Disjunctive Hull approach, if some non-basic variable s_j is required to be integral in the original problem formulation, then the cut could be strengthened. In this section we describe a technique that can be used to strengthen a cut based on a modification of the CGLP for the 4-term disjunction.

Let J_1 be the set of indices of those non-basic variables subject to integrality constraints in the original formulation and let $J_2 = J \setminus J_1$ be the set of the remaining non-basic variables. From now on we will work with the following definition for P_I:

$$P_I = \{(x,s) : \begin{array}{l} x_1 \ = f_1 + \sum_{j\in J_1} r^1_j s_j \ + \sum_{j\in J_2} r^1_j s_j \\ x_2 \ = f_2 + \sum_{j\in J_1} r^2_j s_j \ + \sum_{j\in J_2} r^2_j s_j \\ s_j \geq 0 \quad j \in J_1 \cup J_2 \\ x_1, x_2 \in \mathbb{Z} \\ s_j \in \mathbb{Z} \quad j \in J_1 \\ s_j \in \mathbb{R} \quad j \in J_2 \ \}. \end{array} \tag{2.3.23}$$

We have the following

Proposition 2.3.5. *If the disjunction*

$$\begin{pmatrix} -r^1 s \geq f_1 \\ -r^2 s \geq f_2 \end{pmatrix} \vee \begin{pmatrix} r^1 s \geq 1 - f_1 \\ -r^2 s \geq f_2 \end{pmatrix} \vee \begin{pmatrix} r^1 s \geq 1 - f_1 \\ r^2 s \geq 1 - f_2 \end{pmatrix} \vee \begin{pmatrix} -r^1 s \geq f_1 \\ r^2 s \geq 1 - f_2 \end{pmatrix} \tag{2.3.24}$$

is valid for P_I, $s, t \geq 0$ and $s \in \mathbb{Z}^{|J_1|}$ then so is the disjunction

$$\begin{pmatrix} -\sum_{j\in J_1}(r^1_j - m^1_j)s_j - \sum_{j\in J_2} r^1_j s_j \geq f_1 \\ -\sum_{j\in J_1}(r^2_j - m^2_j)s_j - \sum_{j\in J_2} r^2_j s_j \geq f_2 \end{pmatrix} \vee \begin{pmatrix} \sum_{j\in J_1}(r^1_j - m^1_j)s_j + \sum_{j\in J_2} r^1_j s_j \geq 1 - f_1 \\ -\sum_{j\in J_1}(r^2_j - m^2_j)s_j - \sum_{j\in J_2} r^2_j s_j \geq f_2 \end{pmatrix} \vee$$

$$\begin{pmatrix} \sum_{j\in J_1}(r^1_j - m^1_j)s_j + \sum_{j\in J_2} r^1_j s_j \geq 1 - f_1 \\ \sum_{j\in J_1}(r^2_j - m^2_j)s_j + \sum_{j\in J_2} r^2_j s_j \geq 1 - f_2 \end{pmatrix} \vee \begin{pmatrix} -\sum_{j\in J_1}(r^1_j - m^1_j)s_j - \sum_{j\in J_2} r^1_j s_j \geq f_1 \\ \sum_{j\in J_1}(r^2_j - m^2_j)s_j + \sum_{j\in J_2} r^2_j s_j \geq 1 - f_2 \end{pmatrix} \tag{2.3.25}$$

for any $m_j^1, m_j^2 \in \mathbb{Z},\ j \in J_1$.

Proof. Assume for the sake of contradiction that $\bar{m}_j^1, \bar{m}_j^2 \in \mathbb{Z},\ j \in J_1$ violate the disjunction (2.3.25). This implies that there exists a solution $(x, s) \in P_I$ with $x \in \mathbb{Z}^2$ such that for some $i \in \{1, 2\}$ the condition

$$\left(-\sum_{j\in J_1}(r_j^i - \bar{m}_j^i)s_j - \sum_{j\in J_2} r_j^i s_j < f_i\right) \wedge \left(\sum_{j\in J_1}(r_j^i - \bar{m}_j^i)s_j + \sum_{j\in J_2} r_j^i s_j < 1 - f_i\right) \quad (2.3.26)$$

holds. We can rewrite (2.3.26) as

$$\sum_{j\in J_1} r_j^i s_j + \sum_{j\in J_2} r_j^i s_j + f_i - 1 < \sum_{j\in J_1} \bar{m}_j^i s_j < \sum_{j\in J_1} r_j^i s_j + \sum_{j\in J_2} r_j^i s_j + f_i$$

i.e.

$$x_i - 1 < \sum_{j\in J_1} \bar{m}_j^i s_j < x_i. \quad (2.3.27)$$

But $\sum_{j\in J_1} \bar{m}_j^i s_j$ is integral and by (2.3.27) is required to be strictly between two consecutive integral numbers $x_i - 1$ and x_i. Hence, we reached a contradiction. □

Theorem 2.3.6. *Given* $(\bar{v}, \bar{w}) \geq 0$ *defining a parametric octahedron, the cut* $\alpha s \geq 1$ *can be strengthened to* $\bar{\alpha} s \geq 1$ *with coefficients* $\bar{\alpha}_j, j \in J_1$ *given by the Mixed Integer Program*

$$\begin{array}{lllll}
\min & \alpha_j & & & \\
 & \alpha_j & -\bar{v}_1 m_j^1 - \bar{w}_1 m_j^2 & \geq & -r_j^1 \bar{v}_1 - r_j^2 \bar{w}_1 \\
 & \alpha_j & +\bar{v}_2 m_j^1 - \bar{w}_2 m_j^2 & \geq & +r_j^1 \bar{v}_2 - r_j^2 \bar{w}_2 \\
 & \alpha_j & +\bar{v}_3 m_j^1 + \bar{w}_3 m_j^2 & \geq & +r_j^1 \bar{v}_3 + r_j^2 \bar{w}_3 \\
 & \alpha_j & -\bar{v}_4 m_j^1 + \bar{w}_4 m_j^2 & \geq & -r_j^1 \bar{v}_4 + r_j^2 \bar{w}_4 \\
 & m_j^1, m_j^2 \in \mathbb{Z}. & & &
\end{array} \quad (2.3.28)$$

The coefficients associated with non-basic continuous variables remain the same, i.e. $\bar{\alpha}_j = \alpha_j, j \in J$ *as given in (2.2.9).*

Proof. Validity of $\bar{\alpha} s \geq 1$ follows from Proposition 2.3.5.

Solving at optimality (2.3.28) is an easy task. Note that the polyhedron associated to the linear relaxation of (2.3.28) is a translated cone having apex $(\alpha_j, m_j^1, m_j^2) = (0, r_j^1, r_j^2)$, which is also the optimal solution to the linear relaxation. As proven in Theorem 2.3.9, the optimal solution to (2.3.28) is attained at one of the following four possible combinations of values for (m_j^1, m_j^2):

$$\left\{(\lfloor r_j^1 \rfloor, \lfloor r_j^2 \rfloor); (\lfloor r_j^1 \rfloor, \lceil r_j^2 \rceil); (\lceil r_j^1 \rceil, \lfloor r_j^2 \rfloor); (\lceil r_j^1 \rfloor, \lceil r_j^2 \rceil)\right\}.$$

Equivalently, the coefficients $\bar{\alpha}_j, j \in J_1$ of $\bar{\alpha} s \geq 1$ are computed as $\bar{\alpha}_j = \max\{\bar{\alpha}_j^1; \bar{\alpha}_j^2; \bar{\alpha}_j^3; \bar{\alpha}_j^4\}$

where we denote

$$\begin{array}{lcll}\bar\alpha_1 & = & -(r^1-m^1)\bar v_1 & -(r^2-m^2)\bar w_1\\ \bar\alpha_2 & = & +(r^1-m^1)\bar v_2 & -(r^2-m^2)\bar w_2\\ \bar\alpha_3 & = & +(r^1-m^1)\bar v_3 & +(r^2-m^2)\bar w_3\\ \bar\alpha_4 & = & -(r^1-m^1)\bar v_4 & +(r^2-m^2)\bar w_4\\ m^1, m^2 \in \mathbb{Z}. & & & \end{array} \tag{2.3.29}$$

In the rest of this section we omit the index j from r_j and α_j.

Lemma 2.3.7. *The values $\bar\alpha_k, k \in \{1,\dots,4\}$, can be expressed as the convex combinations*

$$\begin{array}{ll}\bar\alpha_1 = \lambda_1\left(-(r^1-m^1)\frac{1}{f_1}\right) & +(1-\lambda_1)\left(-(r^2-m^2)\frac{1}{f_2}\right)\\ \bar\alpha_2 = \lambda_2\left((r^1-m^1)\frac{1}{1-f_1}\right) & +(1-\lambda_2)\left(-(r^2-m^2)\frac{1}{f_2}\right)\\ \bar\alpha_3 = \lambda_3\left((r^1-m^1)\frac{1}{1-f_1}\right) & +(1-\lambda_3)\left((r^2-m^2)\frac{1}{1-f_2}\right)\\ \bar\alpha_4 = \lambda_4\left(-(r^1-m^1)\frac{1}{f_1}\right) & +(1-\lambda_4)\left((r^2-m^2)\frac{1}{1-f_2}\right)\end{array}$$

with $\lambda_1 = \bar v_1 f_1$; $\lambda_2 = \bar v_2(1-f_1)$; $\lambda_3 = \bar v_3(1-f_1)$; $\lambda_4 = \bar v_4 f_1$.

Proof. Let $\lambda_1 = \bar v_1 f_1$. From $f_1\bar v_1 + f_2\bar w_1 = 1$ and $\bar v_1, \bar w_1 \geq 0$ we get $\bar v_1 = \lambda_1\frac{1}{f_1}$ and $\bar w_1 = (1-\lambda_1)\frac{1}{f_2}$ with $0 \leq \lambda_1 \leq 1$. Substituting the terms $\bar v_1$ and $\bar w_1$ in the $\bar\alpha_1$ expression in (2.3.29) we get the first equation.
The other cases are similar.

Definition Let $\bar r^i = r^i - m^i, m^i \in \mathbb{Z}, i \in \{1,2\}$. We say that $\bar r = (\bar r^1, \bar r^2)$ is a *modularization* of the ray $r = (r^1, r^2)$. Moreover we say that $\bar r$ is a *standard modularization* of r if

$$(m^1, m^2) \in \left\{(\lfloor r^1\rfloor, \lfloor r^2\rfloor); (\lfloor r^1\rfloor, \lceil r^2\rceil); (\lceil r^1\rceil, \lfloor r^2\rfloor); (\lceil r^1\rceil, \lceil r^2\rceil)\right\}. \tag{2.3.30}$$

Lemma 2.3.8. *There exists a standard modularization $\bar r$ of the ray r such that*

$$0 \leq f_i + \bar r^i \leq 1, i \in \{1,2\} \tag{2.3.31}$$

i.e. the point $(f+\bar r)$ belongs to K.

Proof. If $f_i + r^i - \lfloor r^i\rfloor \leq 1$ then set let $m^i = \lfloor r^i\rfloor$, note that the condition $f_i + r^i - \lfloor r^i\rfloor \geq 0$ follows since $0 \leq f_i \leq 1$ and $r^i - \lfloor r^i\rfloor \geq 0$. Otherwise ($f_i + r^i - \lfloor r^i\rfloor > 1$) let $m^i = \lceil r^i\rceil$ and from $f_i \leq 1$ and $r^i - \lfloor r^i\rfloor \leq 1$ we get $0 \leq f_i + r^i - \lfloor r^i\rfloor - 1 = f_i + r^i - \lceil r^i\rceil \leq 1$.

Lemma 2.3.9. *The optimal solution to the problem obtained from (2.3.28) by adding the restriction (2.3.30) satisfies* $0 \leq \bar\alpha \leq 1$.

Proof. For any m^1, m^2 at least one $\bar\alpha_k, k \in \{1,\dots,4\}$ is non-negative since $\bar v, \bar w \geq 0$ and therefore $\bar\alpha \geq 0$. We prove the second inequality we exhibit a solution to (2.3.28) such that $\bar\alpha \leq 1$. Let $(\bar m_1, \bar m_2)$ be a pair in (2.3.30) corresponding to a standard modularization $\bar r$ of r that satisfies (2.3.31). By Lemma 2.3.8 such pair exists. We distinguish 4 cases:

- Case $(\bar m_1, \bar m_2) = (\lfloor r^1\rfloor, \lfloor r^2\rfloor)$. Note that $\bar r^i = r^i - \bar m_i \geq 0, i \in \{1,2\}$. We have the following upper bounds on the values $\bar\alpha_k, k \in \{1,\dots,4\}$:

$\bar{\alpha}_1 = -\bar{r}^1\bar{v}_1 - \bar{r}^2\bar{w}_1 \leq 0$.

$\bar{\alpha}_2 = +\bar{r}^1\bar{v}_2 - \bar{r}^2\bar{w}_2 \leq \bar{r}^1\frac{1}{1-f_1}$. From (2.3.31) we get $\bar{r}^1\frac{1}{1-f_1} \leq (1-f_1)\frac{1}{1-f_1} = 1$ therefore $\bar{\alpha}_2 \leq 1$.

$\bar{\alpha}_3 = \lambda_3\bar{r}^1\frac{1}{1-f_1} + (1-\lambda_3)\bar{r}^2\frac{1}{1-f_2}$ for $0 \leq \lambda_3 \leq 1$ by Lemma 2.3.7. From (2.3.31), $\bar{r}^i\frac{1}{1-f_i} \leq (1-f_i)\frac{1}{1-f_i} = 1, i \in \{1,2\}$ holds, therefore $\bar{\alpha}_3 \leq 1$.

$\bar{\alpha}_4 = -\bar{r}^1\bar{v}_4 + \bar{r}^2\bar{w}_4 \leq \bar{r}^2\frac{1}{1-f_2}$. From (2.3.31) we get $\bar{r}^2\frac{1}{1-f_2} \leq (1-f_2)\frac{1}{1-f_2} = 1$ therefore $\bar{\alpha}_4 \leq 1$.

- Case $(\bar{m}_1, \bar{m}_2) = (\lfloor r^1 \rfloor, \lceil r^2 \rceil)$. Note that $\bar{r}^1 = r^1 - \bar{m}_1 \geq 0$ and $\bar{r}^2 = r^2 - \bar{m}_2 \leq 0$. We have the following upper bounds on the values $\bar{\alpha}_k, k \in \{1, \ldots, 4\}$:

$\bar{\alpha}_1 = -\bar{r}^1\bar{v}_1 - \bar{r}^2\bar{w}_1 \leq -\bar{r}^2\frac{1}{f_2}$. From (2.3.31) we get $-\bar{r}^2\frac{1}{f_2} \leq f_2\frac{1}{f_2} = 1$ therefore $\bar{\alpha}_1 \leq 1$.

$\bar{\alpha}_2 = \lambda_2(\bar{r}^1\frac{1}{1-f_1}) + (1-\lambda_2)(-\bar{r}^2\frac{1}{f_2})$ for $0 \leq \lambda_2 \leq 1$ by Lemma 2.3.7. From (2.3.31), $\bar{r}^1\frac{1}{1-f_1} \leq (1-f_1)\frac{1}{1-f_1} = 1$ and $-\bar{r}^2\frac{1}{f_2} \leq f_2\frac{1}{f_2} = 1$ hold, therefore $\bar{\alpha}_2 \leq 1$.

$\bar{\alpha}_3 = +\bar{r}^1\bar{v}_3 + \bar{r}^2\bar{w}_3 \leq \bar{r}^1\frac{1}{1-f_1}$. From (2.3.31) we get $\bar{r}^1\frac{1}{1-f_1} \leq (1-f_1)\frac{1}{1-f_1} = 1$ therefore $\bar{\alpha}_3 \leq 1$.

$\bar{\alpha}_4 = -\bar{r}^1\bar{v}_4 + \bar{r}^2\bar{w}_4 \leq 0$.

- Case $(\bar{m}_1, \bar{m}_2) = (\lceil r^1 \rceil, \lfloor r^2 \rfloor)$. Note that $\bar{r}^1 = r^1 - \bar{m}_1 \leq 0$ and $\bar{r}^2 = r^2 - \bar{m}_2 \geq 0$. We have the following upper bounds on the values $\bar{\alpha}_k, k \in \{1, \ldots, 4\}$:

$\bar{\alpha}_1 = -\bar{r}^1\bar{v}_1 - \bar{r}^2\bar{w}_1 \leq -\bar{r}^1\frac{1}{f_1}$. From (2.3.31) we get $-\bar{r}^1\frac{1}{f_1} \leq f_1\frac{1}{f_1} = 1$ therefore $\bar{\alpha}_1 \leq 1$.

$\bar{\alpha}_2 = +\bar{r}^1\bar{v}_2 - \bar{r}^2\bar{w}_2 \leq 0$.

$\bar{\alpha}_3 = +\bar{r}^1\bar{v}_3 + \bar{r}^2\bar{w}_3 \leq \bar{r}^2\frac{1}{1-f_2}$. From (2.3.31) we get $\bar{r}^2\frac{1}{1-f_2} \leq (1-f_2)\frac{1}{1-f_2} = 1$ therefore $\bar{\alpha}_3 \leq 1$.

$\bar{\alpha}_4 = \lambda_4(-\bar{r}^1\frac{1}{f_1}) + (1-\lambda_4)(\bar{r}^2\frac{1}{1-f_2})$ for $0 \leq \lambda_4 \leq 1$ by Lemma 2.3.7. From (2.3.31), $-\bar{r}^1\frac{1}{f_1} \leq f_1\frac{1}{f_1} = 1$ and $\bar{r}^2\frac{1}{1-f_2} \leq (1-f_2)\frac{1}{1-f_2} = 1$ hold, therefore $\bar{\alpha}_4 \leq 1$.

- Case $(\bar{m}_1, \bar{m}_2) = (\lceil r^1 \rceil, \lceil r^2 \rceil)$. Note that $\bar{r}^i = r^i - \bar{m}_i \leq 0, i \in \{1,2\}$. We have the following upper bounds on the values $\bar{\alpha}_k, k \in \{1, \ldots, 4\}$:

$\bar{\alpha}_1 = \lambda_1(-\bar{r}^1\frac{1}{f_1}) + (1-\lambda_1)(-\bar{r}^2\frac{1}{f_2})$ for $0 \leq \lambda_1 \leq 1$ by Lemma 2.3.7. From (2.3.31), $-\bar{r}^1\frac{1}{f_1} \leq (f_1)\frac{1}{f_1} = 1$ and $-\bar{r}^2\frac{1}{f_2} \leq (f_2)\frac{1}{f_2} = 1$ hold, therefore $\bar{\alpha}_1 \leq 1$.

$\bar{\alpha}_2 = +\bar{r}^1\bar{v}_2 - \bar{r}^2\bar{w}_2 \leq -\bar{r}^2\frac{1}{f_2}$. From (2.3.31) we get $-\bar{r}^2\frac{1}{f_2} \leq (f_2)\frac{1}{f_2} = 1$ therefore $\bar{\alpha}_2 \leq 1$.

$\bar{\alpha}_3 = +\bar{r}^1\bar{v}_3 + \bar{r}^2\bar{w}_3 \leq 0$.

$\bar{\alpha}_4 = -\bar{r}^1\bar{v}_4 + \bar{r}^2\bar{w}_4 \leq -\bar{r}^1\frac{1}{f_1}$. From (2.3.31) we get $-\bar{r}^1\frac{1}{f_1} \leq (f_1)\frac{1}{f_1} = 1$ therefore $\bar{\alpha}_4 \leq 1$.

Since $\bar{\alpha}_k \leq 1, k \in \{1, \ldots, 4\}$ we get $\alpha = \max\{\bar{\alpha}_1, \bar{\alpha}_2, \bar{\alpha}_3, \bar{\alpha}_4\} \leq 1$.

Theorem 2.3.10. *The optimal solution to (2.3.28) is attained for at least one of the four pairs of values for (m^1, m^2) in (2.3.30)*

Proof. Let $(\alpha^*, \alpha_1^*, \alpha_2^*, \alpha_3^*, \alpha_4^*, m^{1*}, m^{2*})$ be an optimal solution to the problem obtained from (2.3.28) by adding the restriction (2.3.30). By Lemma 2.3.9, $\alpha^* \leq 1$. Assume that the optimal solution is such that $m^{1*} = \lfloor r^1 \rfloor$, $m^{2*} = \lfloor r^2 \rfloor$, the proof for the other cases is analogous. Increasing the current value of m^{1*} by one unit brings us to case $m^{1*} = \lceil r^1 \rceil$, $m^{2*} = \lfloor r^2 \rfloor$. Similarly if only m^{2*} is increased or both m^{1*} and m^{2*} are increased. So we are left to prove that any value $m^{1\prime} < m^{1*}$ or $m^{2\prime} < m^{2*}$ yields a coefficient α' with $\alpha' \geq \alpha^*$.
Let $m^{i\prime} = m^{i*} - d_i, d_i \geq 0, d_i \in \mathbb{Z}$, i.e. the value m^{i*} decreased by d_i units, furthermore denote $\bar{r}^i = r^i - m^{i*}$; $\bar{r}^{i\prime} = r^i - m^{i\prime} = \bar{r}^i + d_i$ for $i \in \{1, 2\}$ and let α'_k be the new value α_k for $k \in \{1, \ldots, 4\}$ in (2.3.29) computed using $m^{i\prime}$ instead of $m^i, i \in \{1, 2\}$. Note that $0 \leq \bar{r}^i \leq 1$ since we assumed $m^{i*} = \lfloor r^i \rfloor, i \in \{1, 2\}$.

If $d_1 \geq 1 \wedge d_2 \geq 1$. From the definitions we have $\bar{r}^{1\prime} = \bar{r}^1 + d_1 \geq 1 \wedge \bar{r}^{2\prime} = \bar{r}^2 + d_2 \geq 1$. By Lemma 2.3.7 $\alpha'_3 = \lambda_3 \left(\bar{r}^{1\prime} \frac{1}{1-f_1}\right) + (1 - \lambda_3) \left(\bar{r}^{2\prime} \frac{1}{1-f_2}\right)$ and since $\bar{r}^{i\prime} \frac{1}{1-f_i} \geq 1, i \in \{1, 2\}$, α'_3 is the convex combination of two terms greater or equal to 1 and therefore $\alpha' \geq \alpha'_3 \geq 1 \geq \alpha^*$.

In the rest of the proof we can exclude the following occurrences:

- if $\alpha^* = \alpha_1^*$ then $\alpha^* = 0$ since α_1^* is the sum of two non positive terms. By Lemma 2.3.9 α^* is optimal.

- if $\alpha^* = \alpha_3^*$ then $\alpha' \geq \alpha'_3 = \alpha_3^* + d_1 \bar{v}_3 + d_2 \bar{w}_3 \geq \alpha_3^* = \alpha^*$.

therefore we only need to consider the cases when $\alpha^* = \alpha_2^*$ or $\alpha^* = \alpha_4^*$.

If $d_1 \geq 1 \wedge d_2 = 0$ then in the case $\alpha^* = \alpha_2^*$ we have $\alpha' \geq \alpha'_2 = \alpha_2^* + d_1 \bar{v}_2 \geq \alpha_2^* = \alpha^*$. In the case $\alpha^* = \alpha_4^*$ note that $\bar{r}^{1\prime} \frac{1}{1-f_1} \geq 1$ holds since $\bar{r}^{1\prime} = \bar{r}^1 + d_1 \geq 1$. If also $\bar{r}^{2\prime} \frac{1}{1-f_2} \geq 1$ holds, since α'_3 is a convex combination of two terms greater or equal to ,1 we get $\alpha' \geq \alpha'_3 \geq 1 \geq \alpha^*$ by Lemma 2.3.9. Otherwise, $0 \leq \bar{r}^{2\prime} \frac{1}{1-f_2} < 1$, we have $\alpha^* = \alpha_4^* = -\bar{r}^1 \bar{v}_4 + \bar{r}^2 \bar{w}_4 \leq \bar{r}^{2\prime} \bar{w}_4 \leq \bar{r}^{2\prime} \frac{1}{1-f_2} \leq \lambda_3 \left(\bar{r}^{1\prime} \frac{1}{1-f_1}\right) + (1 - \lambda_3) \left(\bar{r}^{2\prime} \frac{1}{1-f_2}\right)$ for any $0 \leq \lambda_3 \leq 1$. By Lemma 2.3.7 we have $\alpha'_3 = \lambda_3 \left(\bar{r}^{1\prime} \frac{1}{1-f_1}\right) + (1 - \lambda_3) \left(\bar{r}^{2\prime} \frac{1}{1-f_2}\right)$, therefore $\alpha' \geq \alpha'_3 \geq \alpha_4^* = \alpha^*$.

If $d_1 = 0 \wedge d_2 \geq 1$ the proof is analogous.

Example In the following numerical instance we show how strengthening is applied to the cuts generated by the CGLP. Consider the instance

$$\begin{array}{llllll} x_1 & = \frac{1}{3} & +\frac{4}{3} s_1 & +\frac{13}{2} s_2 & -\frac{9}{4} s_3 & -\frac{4}{3} s_4 \\ x_2 & = \frac{1}{3} & +\frac{7}{2} s_1 & -\frac{7}{3} s_2 & -\frac{7}{6} s_3 & +\frac{5}{4} s_4 \\ s_j & \geq 0 & j \in J & & & \\ s_j & \in \mathbb{Z} & j \in J & & & \end{array} \tag{2.3.32}$$

where $J = \{1 \ldots 4\}$.

We first relax the integrality constraints $s_j \in \mathbb{Z} j \in J$ and then enumerating all possible bases of the CGLP (2.3.4) normalized with $\beta = 1$, we determine the facets of the Disjunctive Hull:

$$\begin{array}{lrrrrl} \text{Cut 1}: & 2.6745\ s_1 & +7\ s_2 & +3.5\ s_3 & +2.4667\ s_4 & \geq 1 \\ \text{Cut 2}: & 5.25\ s_1 & +7\ s_2 & +3.5\ s_3 & +1.875\ s_4 & \geq 1 \\ \text{Cut 3}: & 2\ s_1 & +9.75\ s_2 & +3.5\ s_3 & +2.6216\ s_4 & \geq 1 \end{array} \tag{2.3.33}$$

The values of v, w associated with the 3 cuts are:

$$\begin{array}{l} \text{Cut } 1: v = [\ 0 \quad 0 \quad 1.1887 \quad 0.8353\ ]; \quad w = [\ 3 \quad 3 \quad 0.3113 \quad 1.0824\ ] \\ \text{Cut } 2: v = [\ 0 \quad 0 \quad 0 \quad 0\ ]; \quad w = [\ 3 \quad 3 \quad 1.5 \quad 1.5\ ] \\ \text{Cut } 3: v = [\ 0 \quad 1.5 \quad 1.5 \quad 1.0541\ ]; \quad w = [\ 3 \quad 0 \quad 0 \quad 0.9730\ ] \end{array}$$

We now apply the strengthening procedure to Cut 1.
The values for v, w for Cut 1 are $v = [0 \ \ 0 \ \ 1.1887 \ \ 0.8353]; w = [3 \ \ 3 \ \ 0.3113 \ \ 1.0824]$.

- For the ray $(r_1^1, r_1^2) = (\frac{4}{3}, \frac{7}{2})$:
  - choosing $(m_1^1, m_1^2) = (\lfloor r_1^1 \rfloor, \lfloor r_1^2 \rfloor) = (1, 3)$ yields $\bar{\alpha}_1 = 0.5519$
  - choosing $(m_1^1, m_1^2) = (\lfloor r_1^1 \rfloor, \lceil r_1^2 \rceil) = (1, 4)$ yields $\bar{\alpha}_1 = 1.5$
  - choosing $(m_1^1, m_1^2) = (\lceil r_1^1 \rceil, \lfloor r_1^2 \rfloor) = (2, 3)$ yields $\bar{\alpha}_1 = 1.0981$
  - choosing $(m_1^1, m_1^2) = (\lceil r_1^1 \rceil, \lceil r_1^2 \rceil) = (2, 4)$ yields $\bar{\alpha}_1 = 1.5$

  The minimum is attained for $(m_1^1, m_1^2) = (\lfloor r_1^1 \rfloor, \lfloor r_1^2 \rfloor) = (1, 3)$ that produces $\bar{\alpha}_1 = 0.5519$.

- For the ray $(r_2^1, r_2^2) = (\frac{13}{2}, -\frac{7}{3})$:
  - choosing $(m_2^1, m_2^2) = (\lfloor r_2^1 \rfloor, \lfloor r_2^2 \rfloor) = (6, -3)$ yields $\bar{\alpha}_2 = 0.8019$
  - choosing $(m_2^1, m_2^2) = (\lfloor r_2^1 \rfloor, \lceil r_2^2 \rceil) = (6, -2)$ yields $\bar{\alpha}_2 = 1$
  - choosing $(m_2^1, m_2^2) = (\lceil r_2^1 \rceil, \lfloor r_2^2 \rfloor) = (7, -3)$ yields $\bar{\alpha}_2 = 1.1393$
  - choosing $(m_2^1, m_2^2) = (\lceil r_2^1 \rceil, \lceil r_2^2 \rceil) = (7, -2)$ yields $\bar{\alpha}_2 = 1$

  The minimum is attained for $(m_2^1, m_2^2) = (\lfloor r_2^1 \rfloor, \lfloor r_2^2 \rfloor) = (6, -3)$ that produces $\bar{\alpha}_2 = 0.8019$.

- For the ray $(r_3^1, r_3^2) = (-\frac{9}{4}, -\frac{7}{6})$:
  - choosing $(m_3^1, m_3^2) = (\lfloor r_3^1 \rfloor, \lfloor r_3^2 \rfloor) = (-3, -2)$ yields $\bar{\alpha}_3 = 1.1509$
  - choosing $(m_3^1, m_3^2) = (\lfloor r_3^1 \rfloor, \lceil r_3^2 \rceil) = (-3, -1)$ yields $\bar{\alpha}_3 = 0.8396$
  - choosing $(m_3^1, m_3^2) = (\lceil r_3^1 \rceil, \lfloor r_3^2 \rfloor) = (-2, -2)$ yields $\bar{\alpha}_3 = 1.1108$
  - choosing $(m_3^1, m_3^2) = (\lceil r_3^1 \rceil, \lceil r_3^2 \rceil) = (-2, -1)$ yields $\bar{\alpha}_3 = 0.5$

  The minimum is attained for $(m_3^1, m_3^2) = (\lceil r_3^1 \rceil, \lceil r_3^2 \rceil) = (-2, -1)$ that produces $\bar{\alpha}_3 = 0.5$.

- For the ray $(r_4^1, r_4^2) = (-\frac{4}{3}, \frac{5}{4})$:
  - choosing $(m_4^1, m_4^2) = (\lfloor r_4^1 \rfloor, \lfloor r_4^2 \rfloor) = (-2, 1)$ yields $\bar{\alpha}_4 = 0.8703$
  - choosing $(m_4^1, m_4^2) = (\lfloor r_4^1 \rfloor, \lceil r_4^2 \rceil) = (-2, 2)$ yields $\bar{\alpha}_4 = 2.25$
  - choosing $(m_4^1, m_4^2) = (\lceil r_4^1 \rceil, \lfloor r_4^2 \rfloor) = (-1, 1)$ yields $\bar{\alpha}_4 = 0.549$
  - choosing $(m_4^1, m_4^2) = (\lceil r_4^1 \rceil, \lceil r_4^2 \rceil) = (-1, 2)$ yields $\bar{\alpha}_4 = 2.25$

The minimum is attained for $(m_4^1, m_4^2) = (\lceil r_4^1 \rceil, \lfloor r_4^2 \rfloor) = (-1, 1)$ that produces $\bar{\alpha}_4 = 0.549$.

So we obtained the strengthened cut

$$0.5519\ s_1 + 0.8019\ s_2 + 0.5\ s_3 + 0.549\ s_4 \geq 1.$$

After applying the same procedure to the cuts 2 and 3 we get the following strengthened cuts:

$$\begin{array}{lrrrrl} \text{Strengthened cut 1 :} & 0.5519\ s_1 & +0.8019\ s_2 & +0.5\ s_3 & +0.549\ s_4 & \geq 1 \\ \text{Strengthened cut 2 :} & 0.75\ s_1 & +1\ s_2 & +0.5\ s_3 & +0.375\ s_4 & \geq 1 \\ \text{Strengthened cut 3 :} & 0.5\ s_1 & +0.75\ s_2 & +0.5\ s_3 & +0.5946\ s_4 & \geq 1 \end{array} \tag{2.3.34}$$

We can compare the cuts in (2.3.34) with the Mixed Integer Gomory (GMI) cuts that are obtained from x_1, x_2 :

$$\begin{array}{lrrrrl} \text{GMI from } x_1 : & 0.5\ s_1 & +0.75\ s_2 & +0.75\ s_3 & +1\ s_4 & \geq 1 \\ \text{GMI from } x_2 : & 0.75\ s_1 & +1\ s_2 & +0.5\ s_3 & +0.375\ s_4 & \geq 1. \end{array} \tag{2.3.35}$$

Notice that the GMI cut derived from x_1 is dominated by the Strengthened cut 3 in (2.3.34) and the GMI cut derived from x_2 is the same as the Strengthened cut 2 in (2.3.34).

If we modularize the instance before generating the cuts it is possible to get different and/or stronger inequalities. We now show this on the instance (2.3.32). We can modularize (2.3.32) using the optimal choice of the m's to obtain Cut 1 in (2.3.34) as follows:

$$\begin{array}{llllll} x_1 & = \frac{1}{3} & +(\frac{4}{3} - \lfloor \frac{4}{3} \rfloor)s_1 & +(\frac{13}{2} - \lfloor \frac{13}{2} \rfloor)s_2 & +(-\frac{9}{4} - \lceil -\frac{9}{4} \rceil)s_3 & +(-\frac{4}{3} - \lceil -\frac{4}{3} \rceil)s_4 \\ x_2 & = \frac{1}{3} & +(\frac{7}{2} - \lfloor \frac{7}{2} \rfloor)s_1 & +(-\frac{7}{3} - \lfloor -\frac{7}{3} \rfloor)s_2 & +(-\frac{7}{6} - \lceil -\frac{7}{6} \rceil)s_3 & +(\frac{5}{4} - \lfloor \frac{5}{4} \rfloor)s_4 \\ s_j & \geq 0 & j \in J & & & \\ s_j & \in \mathbb{Z} & j \in J & & & \end{array}$$

i.e.

$$\begin{array}{llllll} x_1 & = \frac{1}{3} & +\frac{1}{3}s_1 & +\frac{1}{2}s_2 & -\frac{1}{4}s_3 & -\frac{1}{3}s_4 \\ x_2 & = \frac{1}{3} & +\frac{1}{2}s_1 & +\frac{2}{3}s_2 & -\frac{1}{6}s_3 & +\frac{1}{4}s_4 \\ s_j & \geq 0 & j \in J & & & \\ s_j & \in \mathbb{Z} & j \in J. & & & \end{array} \tag{2.3.36}$$

As we did before, we relax the integrality constraints $s_j \in \mathbb{Z} j \in J$ in (2.3.36) and then enumerating all possible bases of the CGLP in (2.3.4) we determine all the facets of the Disjunctive Hull:

$$\begin{array}{lrrrrl} \text{Cut 1 :} & 0.5\ s_1 & +0.75\ s_2 & +0.5\ s_3 & +0.4643\ s_4 & \geq 1 \\ \text{Cut 2 :} & 0.5725\ s_1 & +0.75\ s_2 & +0.5\ s_3 & +0.4375\ s_4 & \geq 1 \\ \text{Cut 3 :} & 0.75\ s_1 & +1\ s_2 & +0.5\ s_3 & +0.375\ s_4 & \geq 1 \end{array} \tag{2.3.37}$$

The values of v, w associated with the 3 cuts are:

$$\begin{array}{l}
\text{Cut 1}: v = [\ 0 \ \ 0 \ \ 1.5 \ \ 0.4286\ ]; \quad w = [\ 3 \ \ 3 \ \ 0 \ \ 1.2857\ ] \\
\text{Cut 2}: v = [\ 0 \ \ 0 \ \ 1.5 \ \ 0.3\ ]; \quad w = [\ 3 \ \ 3 \ \ 0 \ \ 1.35\ ] \\
\text{Cut 3}: v = [\ 0 \ \ 0 \ \ 0 \ \ 0\ ]; \quad w = [\ 3 \ \ 3 \ \ 1.5 \ \ 1.5\ ]
\end{array}$$

Comparing the cuts in (2.3.37) with the strengthened cuts in (2.3.34) we observe that Cut 1 in (2.3.37) strictly dominates the Strengthened Cut 1 in (2.3.34); Cut 3 in (2.3.37) is the same as the Strengthened cut 2 in (2.3.34) and Cut 2 is a new cut that is incomparable to the cuts in (2.3.34).

The cuts produces here are different subce they depend on the values of v, w of the CGLP which in turns depend on the instance itself. Modularizing the rays, changes the instance, therefore it is no surprise that new combination of v, w are produced and consequently different cuts. Note, though, that if one were to use a "fixed" parametric octahedron, strengthening after the intersection cut is generated or using modularization with the same values of m would yield the same cut.

The following is a different way to modularize (2.3.32):

$$\begin{array}{lllll}
x_1 & = \frac{1}{3} & +(\frac{4}{3} - \lfloor \frac{4}{3} \rfloor)s_1 & +(\frac{13}{2} - \lfloor \frac{13}{2} \rfloor)s_2 & +(-\frac{9}{4} - \lceil -\frac{9}{4} \rceil)s_3 \quad +(-\frac{4}{3} - \lceil -\frac{4}{3} \rceil)s_4 \\
x_2 & = \frac{1}{3} & +(\frac{7}{2} - \lfloor \frac{7}{2} \rfloor)s_1 & +(-\frac{7}{3} - \lceil -\frac{7}{3} \rceil)s_2 & +(-\frac{7}{6} - \lceil -\frac{7}{6} \rceil)s_3 \quad +(\frac{5}{4} - \lfloor \frac{5}{4} \rfloor)s_4 \\
s_j & \geq 0 & j \in J & & \\
s_j & \in \mathbb{Z} & j \in J & &
\end{array}$$

i.e.

$$\begin{array}{llllll}
x_1 & = \frac{1}{3} & +\frac{1}{3}s_1 & +\frac{1}{2}s_2 & -\frac{1}{4}s_3 & -\frac{1}{3}s_4 \\
x_2 & = \frac{1}{3} & +\frac{1}{2}s_1 & -\frac{1}{3}s_2 & -\frac{1}{6}s_3 & +\frac{1}{4}s_4 \\
s_j & \geq 0 & j \in J & & & \\
s_j & \in \mathbb{Z} & j \in J. & & &
\end{array} \tag{2.3.38}$$

The following are all the facets of the Disjunctive Hull:

$$\begin{array}{llllll}
\text{Cut } 1': & 0.5\ s_1 & +0.75\ s_2 & +0.525\ s_3 & +0.4643\ s_4 & \geq 1 \\
\text{Cut } 2': & 0.5\ s_1 & +1\ s_2 & +0.5\ s_3 & +0.4643\ s_4 & \geq 1 \\
\text{Cut } 3': & 0.75\ s_1 & +0.5\ s_2 & +0.5\ s_3 & +0.375\ s_4 & \geq 1
\end{array} \tag{2.3.39}$$

The values of v, w associated with the 3 cuts are:

$$\begin{array}{l}
\text{Cut } 1': v = [\ 0.3 \ \ 1.5 \ \ 1.5 \ \ 0.4286\ ]; \quad w = [\ 2.7 \ \ 0 \ \ 0 \ \ 1.2857\ ] \\
\text{Cut } 2': v = [\ 0 \ \ 1.5 \ \ 1.5 \ \ 0.4286\ ]; \quad w = [\ 3 \ \ 0 \ \ 0 \ \ 1.2857\ ] \\
\text{Cut } 3': v = [\ 0 \ \ 2 \ \ 2.0769 \ \ 0\ ]; \quad w = [\ 3 \ \ 0 \ \ 0.1154 \ \ 1.5\ ]
\end{array}$$

Cut $1'$ is weaker than Cut 1 of the previously modularized instance and Cut $3'$ is stronger than Cut 3. Cut $2'$ can be further strengthened as $0.5\ s_1 + 0.75\ s_2 + 0.5\ s_3 + 0.4643\ s_4 \geq 1$ and it becomes the same as Cut 1.

2.3.6 Initial experiments with Integer Hull facets

As of today, four computational studies, [14, 30, 29, 33], for cuts from multiple rows have been published and procedures to derive them automatically are still under investigation. The experiments by Dash et al. in [29] show that their cross cuts [28] add a non-trivial improvement on top of the gap closed by the split closure. The other experiments suggest that among all families of cuts considered, split cuts are the most useful ones. Espinoza shows in his experiments that there might be some potential in using cuts from multiple rows. On average they provide a 31% speedup on reaching optimality when using the additional cuts with CPLEX.

We considered a collection of 18 small MIP instances to test the strength of triangles of type 2 and quadrilateral cuts obtained by the procedures given in 2.3.3. Our instances are taken from the MIPLIB 1 repository and we restrict our attention to problems with no more than 100 integer variables and less than 300 constraints. Moreover we discarded all those instances for which 1 round of GMI cuts does not improve the duality gap by more than 1%. We use the procedures described in Section 2.3 to separate Disjunctive Hull facets that are also facets for the Integer Hull. First we solve the LP relaxation and we consider every pair of rows associated to some variable subject to integrality constraints with a fractional value in the LP solution. For each pair of rows we eliminate duplicate rays and we apply the procedures *GenerateNonDegenerateQuadrilateralFacet* and *GenerateDegenerateQuadrilateralFacet* to generate facets of the Integer Hull. Every time a subset of 3 or 4 rays is chosen, we eliminate the first ray in order to avoid to generate the same cut again. Cuts are collected in a pool and cut validating procedures are applied before computing the duality gap closed by the new cuts. The cut validating procedure discards duplicated cuts and cuts which have a normalized violation less than 10^{-7} and a maximum cut dynamics of 10^{13} (i.e. max ratio between the largest and the smallest coefficient in the normalized cut). Note that the procedures are not exhaustive, i.e. only a subset of facets of the Integer Hull are generated. The purpose of this experiment is to assess the impact of the Integer Hull facets we consider compared to the standard GMI cuts (separated using the cut generator *CglGomory* part of CGL/CoinOR). Our implementation is written in C++ and is based on the CoinOR framework. We use CLP version 1.11 and CGL version 0.55. Our results take into consideration the duality gap as a measure of stregth for the cuts we separate. Given a Mixed Integer Program having optimal value v^{IP} and a Linear Programming relaxation optimal value v^{LP}, we define the duality gap for a relaxation R having optimal solution v^R as

$$\text{DualityGap\%} = \frac{v^R - v^{LP}}{v^{IP} - v^{LP}} \cdot 100. \tag{2.3.40}$$

In Table 2.1 we compare GMI cuts and Integer Hull facets. The duality gap closed by a family of cut is indicated by a trailing % in the column name, while the number of cuts separated per family is indicated by a trailing #.

The cuts generated via the procedure *GenerateNonDegenerateQuadrilateralFacet* are denoted by Q^d and those generated via *GenerateDegenerateQuadrilateralFacet* are denoted by T^d. For some instances we are able to generate only few cuts, and in 2 cases (sample2 and egout) our procedures do not generate any cut, this is due to the fact that our procedure are not exhaustive and not all facets of the Integer Hull are generated by our procedures.

Table 2.1: Preliminary experiments comparing GMI and Disjunctive Hull facets

instance	**GMI%**	**GMI#**	T^d%	Q^d%	T^d+Q^d%	T^d+Q^d#	**GMI+T^d+Q^d%**
stein09*	28.57	9	0.00	28.57	28.57	38	28.57
flugpl	11.74	10	13.25	2.81	**13.27**	68	**13.37**
stein15*	16.64	15	18.06	17.62	**18.06**	2671	**18.06**
sample2	5.82	12	0.00	0.00	0.00	0	5.82
bm23	16.81	4	20.20	19.83	**20.20**	4008	**20.20**
stein27*	8.30	27	8.85	8.79	**8.85**	17763	**8.85**
p0033	12.60	5	0.01	4.05	4.05	9	12.60
pipex	31.49	6	28.63	31.49	**32.08**	607	**32.08**
mod013	4.40	5	15.06	8.42	**16.04**	60	**16.04**
egout	21.84	16	0.00	0.00	0.00	0	21.84
bell5	14.43	22	3.48	0.00	3.48	23	**14.50**
misc02	3.68	8	4.66	4.66	**4.66**	277	**4.66**
bell4	23.37	43	2.26	0.50	2.64	20	23.37
bell3a	58.07	11	23.15	33.35	34.69	85	**58.20**
bell3b	41.11	31	5.39	35.17	35.17	21	41.11
misc05	23.50	12	2.66	23.65	**23.65**	1586	**23.65**
misc01	1.68	12	1.68	1.68	1.68	20188	1.68
lseu	55.83	12	6.04	19.32	19.32	1627	55.83
Average	**21.10**		**8.52**	**13.33**	**14.80**		**22.25**

* original formulations from Fulkerson, Nemhauser and Trotter, 1974. They have non-zero GMI gap.

The results show that the families of cuts separated have a very marginal impact. Our cuts applied together with GMI cuts improve the standard GMI duality gap only for 10 instances out of the 18 considered. On average the improvement over GMI cuts is only 1.15%.

2.3.7 Triangles of Type 1

Following the first round of experiments presented in 2.3.6 we abandoned the idea of separating facets of the Integer Hull and we generated cuts derived from fixed configurations of the parametric octahedron. As fixed shape we initially considered Triangles of Type 1, later in the computational section, we amended these with conical cuts.

There had been several theoretical studies that considered Triangles of Type 1: Basu et al. in [13] show that splits+Triangles of Type 1 might yield arbitrarily bad approximations of P_I, in [16] they consider different measures of strength and they show that Gomory cuts are as useful as Triangle cuts.; He et al. in [39] show that in terms of "volume" cut off, the split cuts are more likely to be better than cuts from Triangles of Type 1; Del Pia et al. in [27] relate the strength of the cuts with the lattice width of the underlying convex set used to produce them. Closer the set is to a split, less likely is the associated cut to improve over the split closure.

In Table 2.2 we report computational results on the same set of instances as in Table 2.1 but this time we separated cuts that derive from fixed Triangles of Type 1.

Every pair of rows is used to generate up to 4 different cuts as there are 4 triangles of type 1 that contain the unit cube:

- Triangle of Type 1 having vertices $(0, 0); (0, 2); (2, 0)$;
- Triangle of Type 1 having vertices $(1, 0); (1, 2); (-1, 0)$;
- Triangle of Type 1 having vertices $(1, 1); (-1, 1); (1, -1)$;
- Triangle of Type 1 having vertices $(0, 1); (2, 1); (0, -1)$.

The cuts separated from fixed shapes such as the 4 triangles above are in general not facets for the Integer Hull of the 2-row relaxations. For this set of experiments, the results show that these type of cuts perform much better than the facets of the Integer Hull considered in Table 2.1. On average the cuts from Triangles of type 1 improve by 6.15% the duality gap on the 18 instances.

This larger improvement of Triangles of Type 1 versus the Integer Hull facets suggested the following question:how well do the 4 cuts from triangles of type 1 approximate the disjunctive hull P_D? We address this question in 2.3.7. Furthermore we carried out experiments on a larger scale with cuts from Triangles of Type 1 and we discuss them in Section 2.5.

Splits+Triangles of Type 1 yield good empirical approximations of P_D

The goal is to see how the Gomory cuts combined with the Triangles of Type 1 compare with the entire collection of facets of the Disjunctive Hull. Note that both Gomory cuts and Triangle of Type 1 cuts correspond to nonbasic solutions that are combinations of basic ones, some of which might be facet defining for the Disjunctive Hull. We considered instances with

Table 2.2: Preliminary experiments comparing GMI and cuts from Triangles of type 1

instance	**GMI%**	**GMI#**	$T_1^{\triangle}$**%**	$T_1^{\triangle}$**#**	**GMI+**$T_1^{\triangle}$**%**	**GMI+**$T_1^{\triangle}$**#**
stein09*	28.57	9	28.57	114	28.57	123
flugpl	11.74	10	13.11	180	**13.11**	190
stein15*	16.64	15	16.88	403	**17.23**	418
sample2	5.82	12	5.86	264	**5.86**	276
bm23	16.81	4	19.21	60	**19.36**	64
stein27*	8.30	27	8.33	1375	**8.53**	1402
p0033	12.60	5	57.04	98	**57.04**	103
pipex	31.49	6	31.87	148	**31.87**	154
mod013	4.40	5	18.78	140	**18.78**	145
egout	21.84	16	61.11	3162	**61.11**	3178
bell5	14.43	22	15.92	1214	**16.14**	1236
misc02	3.68	8	4.26	207	**4.26**	215
bell4	23.37	43	23.62	4133	**23.62**	4176
bell3a	58.07	11	64.11	1990	**64.11**	2001
bell3b	41.11	31	41.44	2747	**41.44**	2778
misc05	23.50	12	23.65	553	**23.65**	565
misc01	1.68	12	1.68	556	1.68	568
lseu	55.83	12	55.19	289	55.83	301
Average	**21.10**		**27.26**		**27.35**	

* original formulations from Fulkerson, Nemhauser and Trotter, 1974. They have non-zero GMI gap.

12 variables. Due to the size of the associated CGLPs, it is impractical to use the double description method via CDD [34] to enumerate all the Disjunctive Hull facets. Instead we resorted on a different technique to obtain them. The procedure *GenerateDisjunctiveHullFacets* shown below, generates basic solutions many of which correspond to Disjunctive Hull facets.

GenerateDisjunctiveHullFacets

Inizialize cut collection $C \leftarrow$ empty
repeat
    Generate random CGLP objective function coefficients $\bar{c}_j, j \in J$
    Optimize CGLP with $\min \sum_j \bar{c}_j \alpha_j$ and get optimal solution $(\bar{\alpha}, \bar{v}, \bar{w})$
    $C \leftarrow C \cup$ "$\bar{\alpha}s \geq 1$" (if not present already)
  until no cuts are added to C in the past 100 iterations
return C

We applied the procedure *GenerateDisjunctiveHullFacets* to a collection of 100 randomly generated MIP instances with 2 constraints, 2 basic integer variables and 10 non-basic continuous variables. For each instance we computed the gap given by adding Disjunctive Hull cuts, Gomory cuts only and Gomory + Triangles of Type 1 cuts. Average results are given in Table 2.3.

Table 2.3: Average gap closed by Disjunctive Hull cuts on 100 random instances

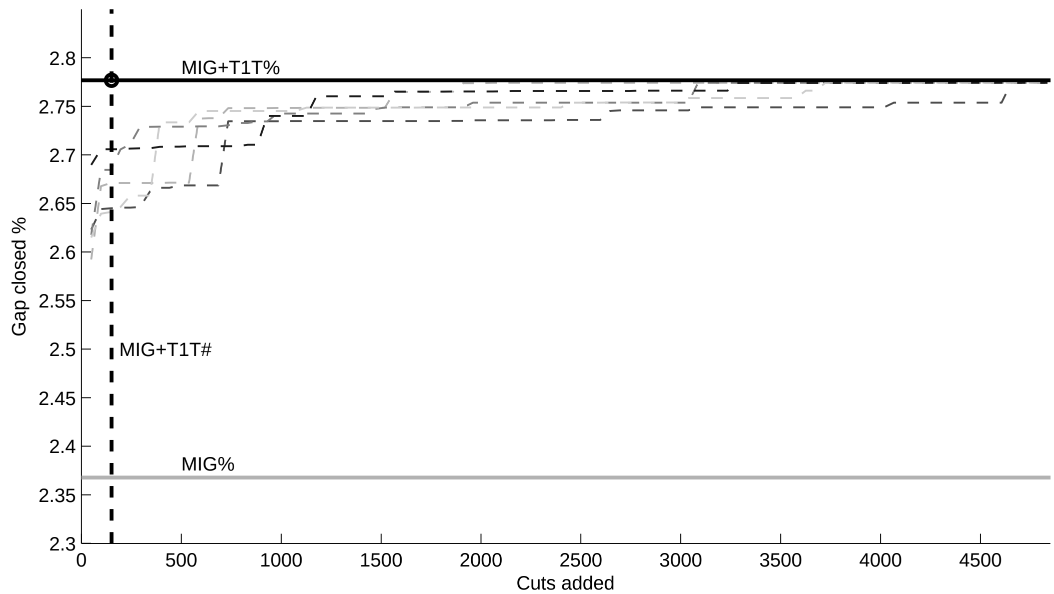
G (2)	G + T1T (6)	DH (10)	DH (20)	DH (50)	DH (100)	DH (150)	DH (200)
49.17	65.02	29.68	53.52	57.23	59.51	62.94	65.69

Gomory and Triangles of Type 1 cuts seem to capture most of the strength of the Disjunctive Hull approach with very little effort since at most 6 cuts (2 Gomory and 4 Triangles) must be separated. On the other hand there exists a very large number of Disjunctive Hull facets and is computationally challenging to generate them and add them to the linear relaxation.

As an example on a real instance, we also run the procedure *GenerateDisjunctiveHullFacets* on every pair of rows of the instance *mas76* from MIPLIB 3. We stopped the computation after 2 hours during which we were able to generate 4836 unique Disjunctive Hull facets. We computed the gap closed by adding them on top of the LP relaxation. The 5 dashed lines in Figure 2.10 illustrate how the gap closed increases with the number of Disjunctive Hull cuts added (each line corresponds to a random ordering of the 4836 cuts). We observe that, in order to reach the gap closed by Gomory+Triangles of Type 1 cuts, a very large number of Disjunctive Hull facets is needed (the vertical black dashed line indicated the number of Gomory+Triangle of Type 1 cuts generated for the instance and the solid black line corresponds to the gap closed by the same cuts).

In our experiments we did not use integrality of the non-basic variables, for Gomory, Triangle of Type 1 and Disjunctive Hull cuts. (This explains the different gap values reported in Figure 2.10 and the tables in the computational results section).

Figure 2.10: Gap closed by Disjunctive Hull facets on the instance mas76



2.4 The 0-1 Disjunctive Hull

The work by Dash et al. [28] has a similar flavor to the one we presented so far. They introduce cross and crooked cross disjunctions and cuts derived from those. They give theoretical results that relate the closures of their cuts with the split closure and they show that any 2 dimensional lattice free cut can be obtained as a crooked cross cut. Cross cuts are a generalization of the MIP Disjunctive Hull cuts, which are the subject of our investigation. When restricted to the 0-1 case though, the family of cuts obtainable via the Disjunctive Hull approach are new and they were not yet considered in the literature.

We now consider the 0-1 Disjunctive Hull $P_D^=$ for $q = 2$, i.e. we work with $P_{01} = \{(x, s) \in \{0, 1\}^2 \times \mathbb{R}^{|J|} : (x, s) \in P_L\}$ where P_L is given in (2.3.1). The CGLP that produce the facets for $P_D^=$ is the Linear Program (2.2.5). In addition to the cuts obtainable for the MIP CGLP given in Section 2.3.1, when v, w are unrestricted in sign some additional parametric octahedra configurations are possible:

- Unbounded parametric octahedra.
- Triangles with 3 tilted faces.

Note that for the 0-1 case, triangles with 3 tilted faces can contain in the interior integer lattice points not belonging to the vertices of K. Moreover, since we can have unbounded parametric octahedra, there exist facets of the Disjunctive Hull with negative coefficients.
As we did in 2.3.1, we give a geometrical classification of the types of CGLP basis that correspond to Disjunctive Hull facets for the 0-1 case. Let $k_1 \in \{1, \ldots, 4\}$ be the index of any vertex of K. We denote by k_2, k_3, k_4 the indices of the vertices of K that follow k_1 in

counter-clockwise order. The following configurations are symmetrical and exhaustive when considering every value for $k_1 \in \{1, \ldots, 4\}$ and swapping the roles of x_1 and x_2 (i.e. swapping v with w).

- (T_C) $v_{k_1} > 0, w_{k_1} < 0$; $v_{k_2}, w_{k_2} > 0$; $v_{k_3} > 0, w_{k_3} > 0$ and $v_{k_4} > 0, w_{k_4} > 0$, the parametric octahedron is a triangle of type 3 with all vertices outside the cube K. The face corresponding to k_4 is inactive. See Figure 2.11(a).

- (T_D) $v_{k_1} > 0, w_{k_1} < 0$; $v_{k_2}, w_{k_2} > 0$; $v_{k_3} < 0, w_{k_3} > 0$ and $v_{k_4} > 0, w_{k_4} > 0$, the parametric octahedron is a triangle of type 3 with one vertex inside the cube K. The face corresponding to k_4 is inactive. See Figure 2.11(b).

- (C_A) $v_{k_1}, w_{k_1} > 0$; $v_{k_2}, v_{k_3} > 0$;$w_{k_2} = w_{k_3} = 0$ and $v_{k_4} > 0, w_{k_4} < 0$, the parametric octahedron is a cone with a vertical face (horizontal, when v and w are swapped). See Figure 2.11(c).

- (C_B) $v_{k_1} < 0, w_{k_1} > 0$; $v_{k_2}, w_{k_2} > 0$; $v_{k_3} > 0, w_{k_3} < 0$ and $v_{k_4} > 0, w_{k_4} > 0$, the parametric octahedron is a cone with the two faces in general position. See Figure 2.11(d).

- (C_T) $v_{k_1} > 0, w_{k_1} < 0$; $v_{k_2}, w_{k_2} > 0$; $v_{k_3} < 0, w_{k_3} > 0$ and $v_{k_4} < 0, w_{k_4} > 0$, the parametric octahedron is a truncated cone (in this case the parametric octahedron face through the vertex k_4 induces the truncation). See Figure 2.11(e).

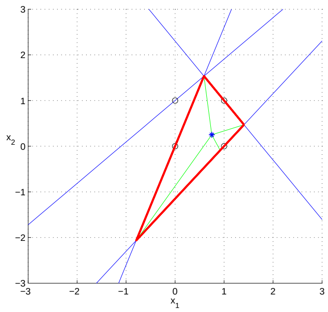
- (S_T) $v_{k_1} < 0, w_{k_1} > 0$; $v_{k_3} > 0, w_{k_3} < 0$, the parametric octahedron is a truncated split. One or two of the faces might truncate the split. See Figure 2.11(f).

- (Q_G) $v_{k_1}, w_{k_1} > 0$; $v_{k_2}, w_{k_2} > 0$;$v_{k_3} < 0, w_{k_3} > 0$ and $v_{k_4} < 0, w_{k_4} > 0$, the parametric octahedron is a general quadrilateral, not maximal (0-1)-free. See Figure 2.11(g).

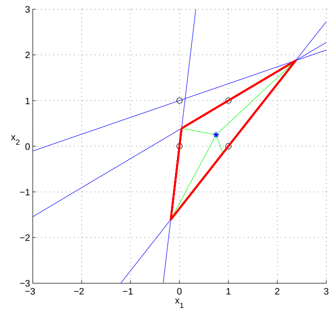
Example Consider the Andersen et al. instance amended with the condition $x_i \in \{0, 1\}, i \in \{1, 2\}$:

$$\begin{array}{llllllll} x_1 & = \frac{1}{4} & +2s_1 & +1s_2 & -3s_3 & & +1s_5 \\ x_2 & = \frac{1}{2} & +1s_1 & +1s_2 & +2s_3 & -1s_4 & -2s_5 \\ s \geq 0 \\ x_1, x_2 \in \{0, 1\}. \end{array} \tag{2.4.1}$$

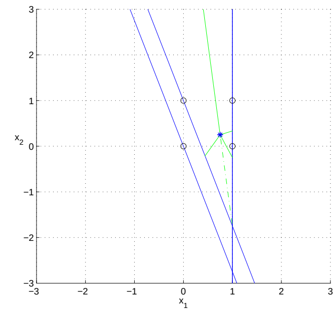
The CGLP for (2.4.1) contains 38 variables (α, β, v, w and the slacks for the $4 \times 5 = 20$ α-constraints and 4 β-constraints) and 25 constraints ($4 \times 5 = 20$ α-constraints, 4 β-constraints and the normalization constraint $\beta = 1$). We can eliminate the variable β and its normalization constraint getting a CGLP with 37 variables and 24 constraints. Of the 37 variables, 13 are unrestricted in sign and must be in the base. That leaves 11 variables that can be part of the base and must be chosen from 37-13=24 variables. Using grey code enumeration, we enumerated all the possible $\binom{24}{11} = 2,496,144$ bases of the 0-1 CGLP associated to (2.4.1). We discarded the infeasible bases and removed all the bases that yield duplicate cuts. The following 35 cuts are all the facets of the 0-1 Disjunctive Hull:



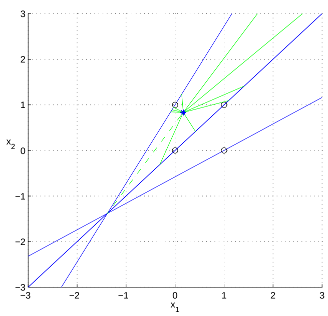
(a) triangle of type 3 with all vertices outside K



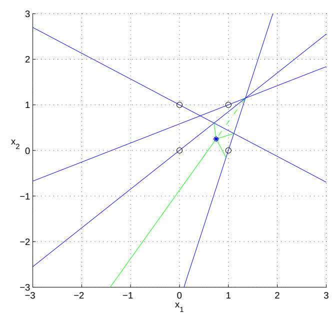
(b) triangle of type 3 with one vertex inside K



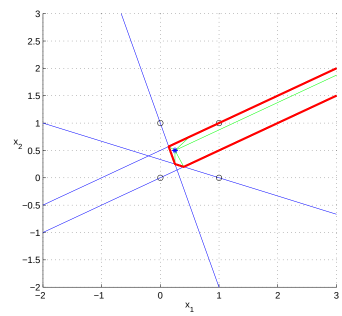
(c) cone with vertical face



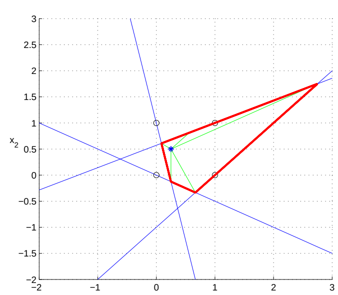
(d) cone with two faces in general position



(e) truncated cone



(f) tilted split truncated



(g) general quadrilateral not maximal (0-1)-free

Figure 2.11: Additional configurations of the parametric octahedron for the 0-1 case

1. Cut (type S): $2.667s_1 + 1.333s_2 + 12s_3 + 0s_4 + 1.333s_5 \geq 1$
   $v_1 = 4;\ v_2 = 1.333;\ v_3 = 1.333;\ v_4 = 4$
   $w_1 = 0;\ w_2 = 0;\ w_3 = 0;\ w_4 = 0$

2. Cut (type T_B): $2.667s_1 + 1.333s_2 + 4.889s_3 + 0.8889s_4 + 1.333s_5 \geq 1$
   $v_1 = 2.222;\ v_2 = 1.333;\ v_3 = 1.333;\ v_4 = 0.4444$
   $w_1 = 0.8889;\ w_2 = 0;\ w_3 = 0;\ w_4 = 1.778$

3. Cut (type T_B): $2s_1 + 2s_2 + 4s_3 + 1s_4 + 1.714s_5 \geq 1$
   $v_1 = 2;\ v_2 = 1.143;\ v_3 = 0;\ v_4 = 0$
   $w_1 = 1;\ w_2 = 0.2857;\ w_3 = 2;\ w_4 = 2$

4. Cut (type T_B): $2s_1 + 2s_2 + 9.714s_3 + 0.2857s_4 + 1.714s_5 \geq 1$
   $v_1 = 3.429;\ v_2 = 1.143;\ v_3 = 0;\ v_4 = 0$
   $w_1 = 0.2857;\ w_2 = 0.2857;\ w_3 = 2;\ w_4 = 2$

5. Cut (type T_C): $3s_1 + 1s_2 + 5.333s_3 + 2s_4 + 4s_5 \geq 1$
   $v_1 = 0;\ v_2 = 0;\ v_3 = 2;\ v_4 = 0.6667$
   $w_1 = 2;\ w_2 = 2;\ w_3 = -1;\ w_4 = 1.667$

6. Cut (type T_C): $2.947s_1 + 1.053s_2 + 5.263s_3 + 0.8421s_4 + 3.579s_5 \geq 1$
   $v_1 = 2.316;\ v_2 = 0.7719;\ v_3 = 1.895;\ v_4 = 0.6316$
   $w_1 = 0.8421;\ w_2 = 0.8421;\ w_3 = -0.8421;\ w_4 = 1.684$

7. Cut (type T_C): $2.667s_1 + 2.4s_2 + 3.467s_3 + 1.067s_4 + 1.333s_5 \geq 1$
   $v_1 = 1.867;\ v_2 = 1.333;\ v_3 = 1.333;\ v_4 = -0.2667$
   $w_1 = 1.067;\ w_2 = 0;\ w_3 = 0;\ w_4 = 2.133$

8. Cut (type T_C): $1.63s_1 + 2.37s_2 + 8.444s_3 + 0.4444s_4 + 1.926s_5 \geq 1$
   $v_1 = 3.111;\ v_2 = 1.037;\ v_3 = -0.7407;\ v_4 = 2.222$
   $w_1 = 0.4444;\ w_2 = 0.4444;\ w_3 = 3.111;\ w_4 = 0.8889$

9. Cut (type T_C): $3.6s_1 + 2s_2 + 4s_3 + 1s_4 + 0.8s_5 \geq 1$
   $v_1 = 2;\ v_2 = 1.6;\ v_3 = 0;\ v_4 = 0$
   $w_1 = 1;\ w_2 = -0.4;\ w_3 = 2;\ w_4 = 2$

10. Cut (type T_D): $1.111s_1 + 2.889s_2 + 14.67s_3 + 0.6667s_4 + 2.222s_5 \geq 1$
    $v_1 = 2.667;\ v_2 = 0.8889;\ v_3 = -1.778;\ v_4 = 5.333$
    $w_1 = 0.6667;\ w_2 = 0.6667;\ w_3 = 4.667;\ w_4 = -0.6667$

11. Cut (type T_D): $0.4s_1 + 3.6s_2 + 23.2s_3 + 2.8s_4 + 7.2s_5 \geq 1$
    $v_1 = -1.6;\ v_2 = 1.6;\ v_3 = -3.2;\ v_4 = 9.6$
    $w_1 = 2.8;\ w_2 = 2.8;\ w_3 = 6.8;\ w_4 = -2.8$

12. Cut (type T_D): $3.034s_1 + 2.621s_2 + 3.172s_3 + 1.103s_4 + 4.276s_5 \geq 1$
    $v_1 = 1.793;\ v_2 = 0.5977;\ v_3 = 2.069;\ v_4 = -0.4138$
    $w_1 = 1.103;\ w_2 = 1.103;\ w_3 = -1.103;\ w_4 = 2.207$

13. Cut (type T_D): $4.364s_1 + 2.545s_2 + 3.273s_3 + 1.091s_4 + 0.3636s_5 \geq 1$
$v_1 = 1.818;\ v_2 = 1.818;\ v_3 = 1.818;\ v_4 = -0.3636$
$w_1 = 1.091;\ w_2 = -0.7273;\ w_3 = 0.7273;\ w_4 = 2.182$

14. Cut (type T_D): $3.765s_1 + 3.059s_2 + 2.588s_3 + 1.176s_4 + 0.7059s_5 \geq 1$
$v_1 = 1.647;\ v_2 = 1.647;\ v_3 = 0.7059;\ v_4 = -0.7059$
$w_1 = 1.176;\ w_2 = -0.4706;\ w_3 = 2.353;\ w_4 = 2.353$

15. Cut (type C_A): $12s_1 + 8s_2 - 4s_3 + 2s_4 + 4s_5 \geq 1$
$v_1 = 0;\ v_2 = 0;\ v_3 = 4;\ v_4 = -4$
$w_1 = 2;\ w_2 = 2;\ w_3 = 4;\ w_4 = 4$

16. Cut (type C_A): $12s_1 + 8s_2 + 12s_3 + 0s_4 - 4s_5 \geq 1$
$v_1 = 4;\ v_2 = 4;\ v_3 = 4;\ v_4 = 4$
$w_1 = 0;\ w_2 = -4;\ w_3 = 4;\ w_4 = 0$

17. Cut (type C_B): $32s_1 + 20s_2 - 20s_3 + 4s_4 + 12s_5 \geq 1$
$v_1 = -4;\ v_2 = 4;\ v_3 = 4;\ v_4 = -12$
$w_1 = 4;\ w_2 = 4;\ w_3 = -4;\ w_4 = 8$

18. Cut (type C_B): $12s_1 + 8s_2 + 44s_3 - 4s_4 - 4s_5 \geq 1$
$v_1 = 12;\ v_2 = 4;\ v_3 = 4;\ v_4 = -4$
$w_1 = -4;\ w_2 = -4;\ w_3 = 4;\ w_4 = 4$

19. Cut (type C_B): $12s_1 + 8s_2 + 28s_3 - 2s_4 - 4s_5 \geq 1$
$v_1 = 8;\ v_2 = 4;\ v_3 = 0;\ v_4 = -4$
$w_1 = -2;\ w_2 = -4;\ w_3 = 2;\ w_4 = 4$

20. Cut (type C_T): $0.6667s_1 + 3.333s_2 + 20s_3 + 2s_4 + 4s_5 \geq 1$
$v_1 = 0;\ v_2 = 0;\ v_3 = -2.667;\ v_4 = 8$
$w_1 = 2;\ w_2 = 2;\ w_3 = 6;\ w_4 = -2$

21. Cut (type C_T): $3s_1 + 2.6s_2 + 3.2s_3 + 2s_4 + 4s_5 \geq 1$
$v_1 = 0;\ v_2 = 0;\ v_3 = 2;\ v_4 = -0.4$
$w_1 = 2;\ w_2 = 2;\ w_3 = -1;\ w_4 = 2.2$

22. Cut (type C_T): $-2s_1 + 6s_2 + 52s_3 + 2s_4 + 4s_5 \geq 1$
$v_1 = 0;\ v_2 = 0;\ v_3 = -8;\ v_4 = 8$
$w_1 = 2;\ w_2 = 2;\ w_3 = 14;\ w_4 = -2$

23. Cut (type C_T): $4s_1 + 2.286s_2 + 3.619s_3 + 4s_4 + 12s_5 \geq 1$
$v_1 = -4;\ v_2 = 1.714;\ v_3 = 4;\ v_4 = -0.1905$
$w_1 = 4;\ w_2 = -0.5714;\ w_3 = -4;\ w_4 = 2.095$

24. Cut (type C_T): $3.515s_1 + 2.909s_2 + 14.91s_3 - 0.3636s_4 + 0.8485s_5 \geq 1$
$v_1 = 4.727;\ v_2 = 1.576;\ v_3 = -1.818;\ v_4 = -0.6061$
$w_1 = -0.3636;\ w_2 = -0.3636;\ w_3 = 4.727;\ w_4 = 2.303$

25. Cut (type C_T): $4.533s_1 + 2.667s_2 + 12s_3 + 0s_4 + 0.2667s_5 \geq 1$
    $v_1 = 4;\ v_2 = 1.867;\ v_3 = -1.333;\ v_4 = 4$
    $w_1 = 0;\ w_2 = -0.8;\ w_3 = 4;\ w_4 = 0$

26. Cut (type C_T): $5.778s_1 + 3.556s_2 + 22.67s_3 - 1.333s_4 - 0.4444s_5 \geq 1$
    $v_1 = 6.667;\ v_2 = 2.222;\ v_3 = -3.111;\ v_4 = 1.333$
    $w_1 = -1.333;\ w_2 = -1.333;\ w_3 = 6.667;\ w_4 = 1.333$

27. Cut (type C_T): $8s_1 - 4s_2 + 12s_3 + 16s_4 + 44s_5 \geq 1$
    $v_1 = 4;\ v_2 = -9.333;\ v_3 = 12;\ v_4 = 4$
    $w_1 = 0;\ w_2 = 16;\ w_3 = -16;\ w_4 = 0$

28. Cut (type S_T): $4s_1 + 0s_2 + 6.667s_3 + 4s_4 + 12s_5 \geq 1$
    $v_1 = -4;\ v_2 = -1.333;\ v_3 = 4;\ v_4 = 1.333$
    $w_1 = 4;\ w_2 = 4;\ w_3 = -4;\ w_4 = 1.333$

29. Cut (type S_T): $4s_1 + 3.2s_2 + 2.4s_3 + 4s_4 + 12s_5 \geq 1$
    $v_1 = -4;\ v_2 = -1.333;\ v_3 = 4;\ v_4 = -0.8$
    $w_1 = 4;\ w_2 = 4;\ w_3 = -4;\ w_4 = 2.4$

30. Cut (type S_T): $5s_1 + 3s_2 + 16s_3 + 1s_4 + 0s_5 \geq 1$
    $v_1 = 2;\ v_2 = 2;\ v_3 = -2;\ v_4 = 6$
    $w_1 = 1;\ w_2 = -1;\ w_3 = 5;\ w_4 = -1$

31. Cut (type S_T): $0s_1 + 4s_2 + 28s_3 + 4s_4 + 6.667s_5 \geq 1$
    $v_1 = -1.333;\ v_2 = -1.333;\ v_3 = -4;\ v_4 = 12$
    $w_1 = 2.667;\ w_2 = 4;\ w_3 = 8;\ w_4 = -4$

32. Cut (type Q_G): $0.8s_1 + 3.2s_2 + 18.4s_3 + 1.6s_4 + 2.4s_5 \geq 1$
    $v_1 = 0.8;\ v_2 = 0.8;\ v_3 = -2.4;\ v_4 = 7.2$
    $w_1 = 1.6;\ w_2 = 0.8;\ w_3 = 5.6;\ w_4 = -1.6$

33. Cut (type Q_G): $2.811s_1 + 1.189s_2 + 5.081s_3 + 0.8649s_4 + 2.486s_5 \geq 1$
    $v_1 = 2.27;\ v_2 = 0.7568;\ v_3 = 1.622;\ v_4 = 0.5405$
    $w_1 = 0.8649;\ w_2 = 0.8649;\ w_3 = -0.4324;\ w_4 = 1.73$

34. Cut (type Q_G): $2.847s_1 + 2.508s_2 + 3.322s_3 + 1.085s_4 + 2.78s_5 \geq 1$
    $v_1 = 1.831;\ v_2 = 0.6102;\ v_3 = 1.695;\ v_4 = -0.339$
    $w_1 = 1.085;\ w_2 = 1.085;\ w_3 = -0.5424;\ w_4 = 2.169$

35. Cut (type Q_G): $2.975s_1 + 1.554s_2 + 4.595s_3 + 0.9256s_4 + 3.802s_5 \geq 1$
    $v_1 = 2.149;\ v_2 = 1.421;\ v_3 = 1.95;\ v_4 = 0.2975$
    $w_1 = 0.9256;\ w_2 = -0.1322;\ w_3 = -0.9256;\ w_4 = 1.851$

For each cut, we give its representation in the space of x_1, x_2 variables in Figure 2.12. For all the cases except for cones and splits, the set L_α is also shown. The number of cuts generated by the 0-1 CGLP (35) largerly exceeds the number of cuts generated by the MIP CGLP (5). As a side experiment, we solved 1000 linear programs with the two rows in 2.4.1, and random objective function coefficients chosen from the interval [0; 100]. Adding the 35 0-1 Disjunctive Hull facets we close 100% of the gap, (this should not surprise since $P_I = P_D^{=}$ for (2.4.1)). Instead, if we use the 5 MIP Disjunctive Hull facets shown in section 2.3.2, the average gap closed is 77%.

2.4.1 Basic solutions and cut dominance

Observation 2.4.1. *There exist optimal basic solutions to the CGLP (both for the MIP and 0-1 case) which correspond to dominated cuts.*

Example Consider the following MIP 2-row instance:

$$\begin{array}{llllll} x_1 & = \frac{1}{4} & +2s_1 & -3s_2 & & +4s_4 \\ x_2 & = \frac{1}{2} & +1s_1 & +2s_2 & -1s_3 & +1s_4 \\ s \geq 0 & & & & & \\ x_1, x_2 \in \mathbb{Z}. & & & & & \end{array}$$

These are two basic feasible solutions to the MIP CGLP

$$\begin{array}{lllll} \text{(a)} & v_1 = 4 & v_2 = \frac{8}{11} & v_3 = 0 & v_4 = 4 \\ & w_1 = 0 & w_2 = \frac{10}{11} & w_3 = 2 & w_4 = 0 \\ & \multicolumn{4}{l}{\text{producing the cut: } 2s_1 + 12s_2 + \frac{10}{11}s_3 + 2s_4 \geq 1} \\ \text{(b)} & v_1 = \frac{24}{11} & v_2 = \frac{8}{11} & v_3 = 0 & v_4 = \frac{8}{55} \\ & w_1 = \frac{10}{11} & w_2 = \frac{10}{11} & w_3 = 2 & w_4 = \frac{118}{55} \\ & \multicolumn{4}{l}{\text{producing the cut: } 2s_1 + \frac{52}{11}s_2 + \frac{10}{11}s_3 + 2s_4 \geq 1} \end{array}$$

Both cuts correspond to basic feasible solutions, but the cut (b) strictly dominates the cut (a). The two cuts are illustrated graphically in Figure 2.13.

On this matter, the following open question arises: "Is it possible to characterize the basic feasible solutions to the CGLP that yield non-dominated cuts?". As of now, we do not have an answer to this question and it will be subject of our future investigations.

Observation 2.4.2. *There exist optimal basic solutions to the CGLP whose associated parametric octahedra do not define maximal 0-1 free cuts.*

This is the case, for example, for the cuts 32-35 of the Andersen et al. 0-1 instance.

2.4.2 Strengthening

Modularization based strengthening applies to cuts derived from the 0-1 Disjunctive Hull only when the parametric octahedron is a lattice free convex set. Moreover, even when this is so, Theorem 2.3.10 does not hold as the following Observation and example highlight.

Observation 2.4.3. *If we allow unconstrained $\bar{v}, \bar{w}$ then the optimal solution to (2.3.28) might not correspond to (m^1, m^2) restricted to (2.3.30).*

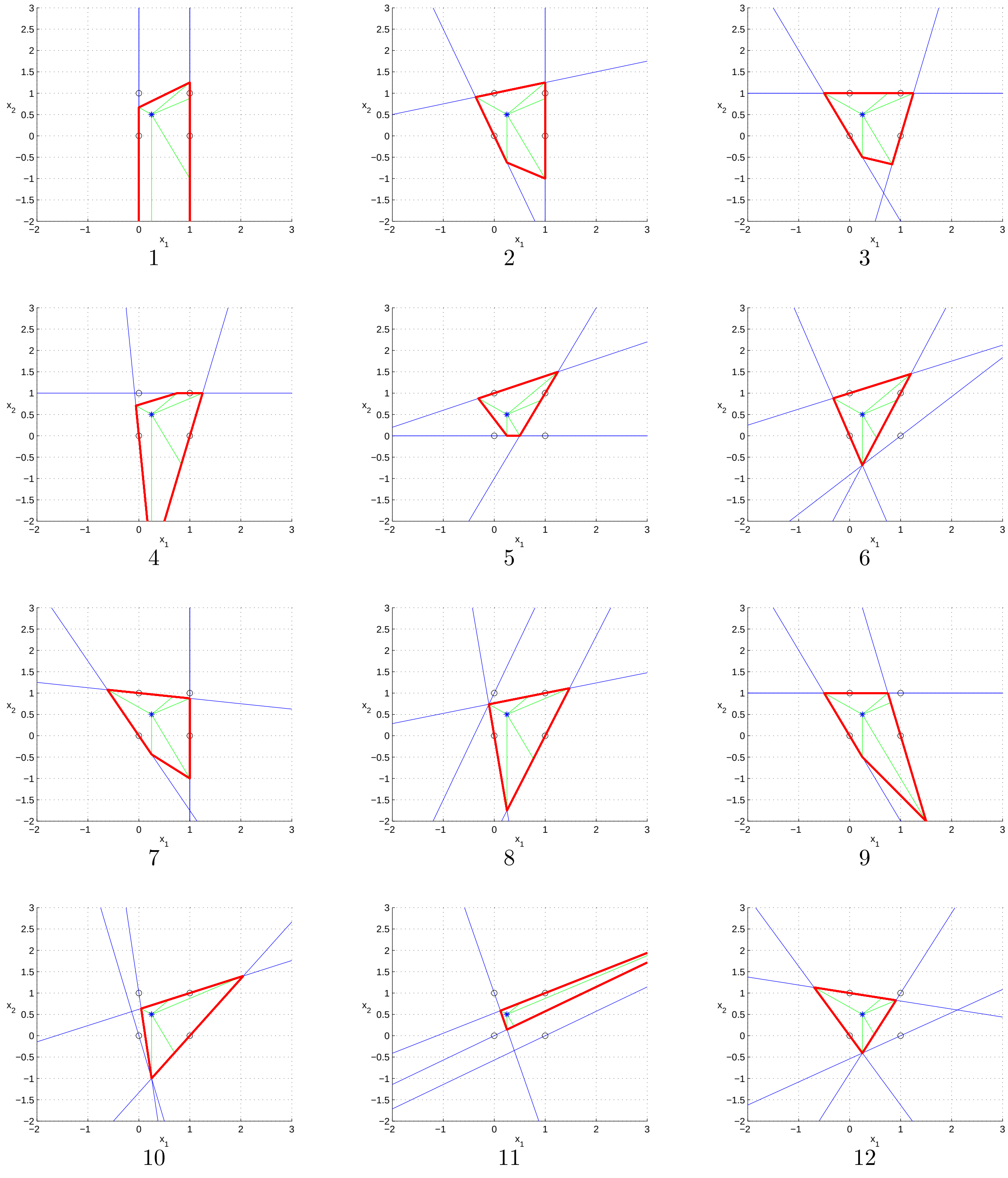


Figure 2.12: 0-1 Disjunctive Hull facets of the Andersen et al. example

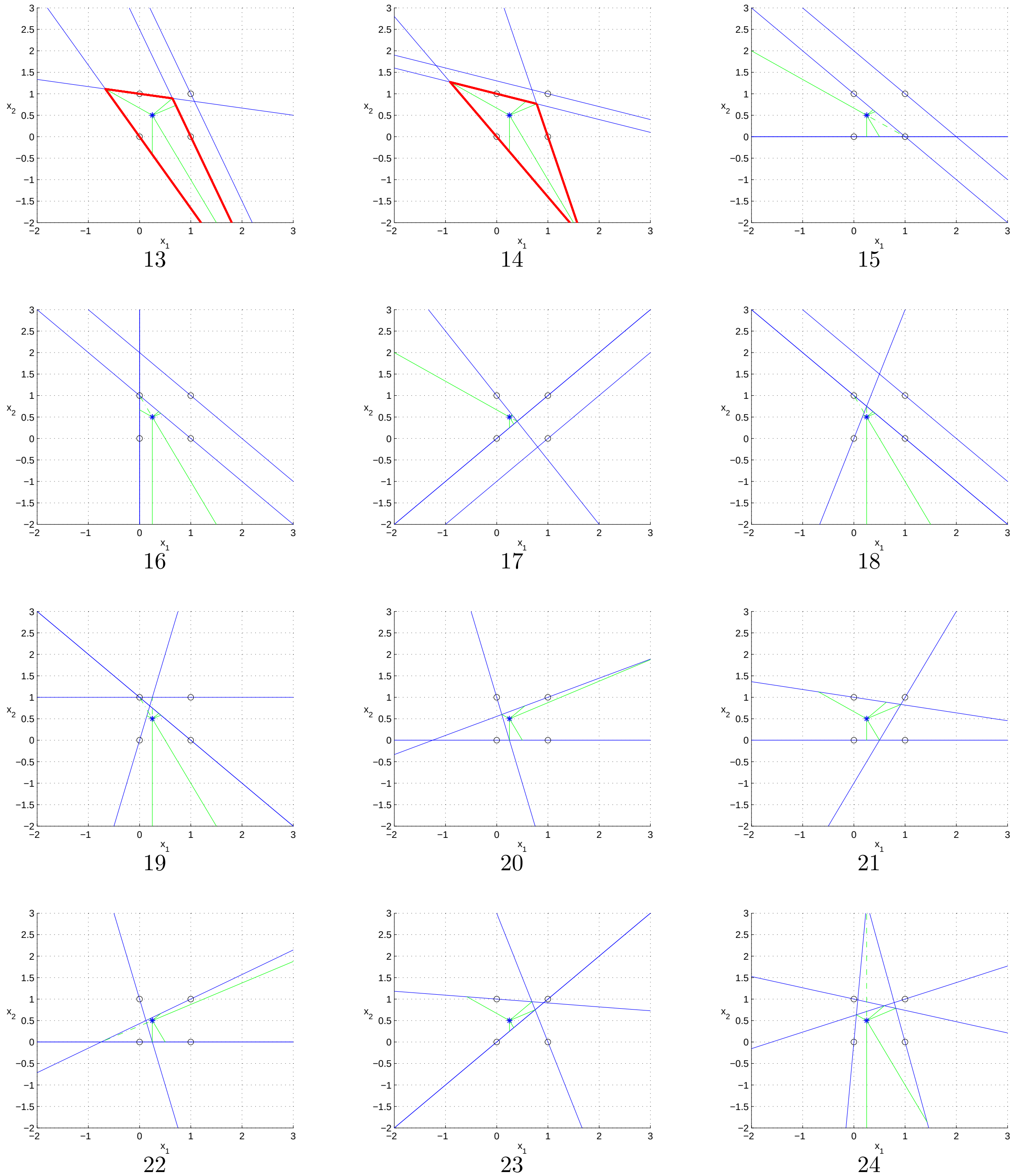


Figure 2.12: 0-1 Disjunctive Hull facets of the Andersen et al. example - continuation

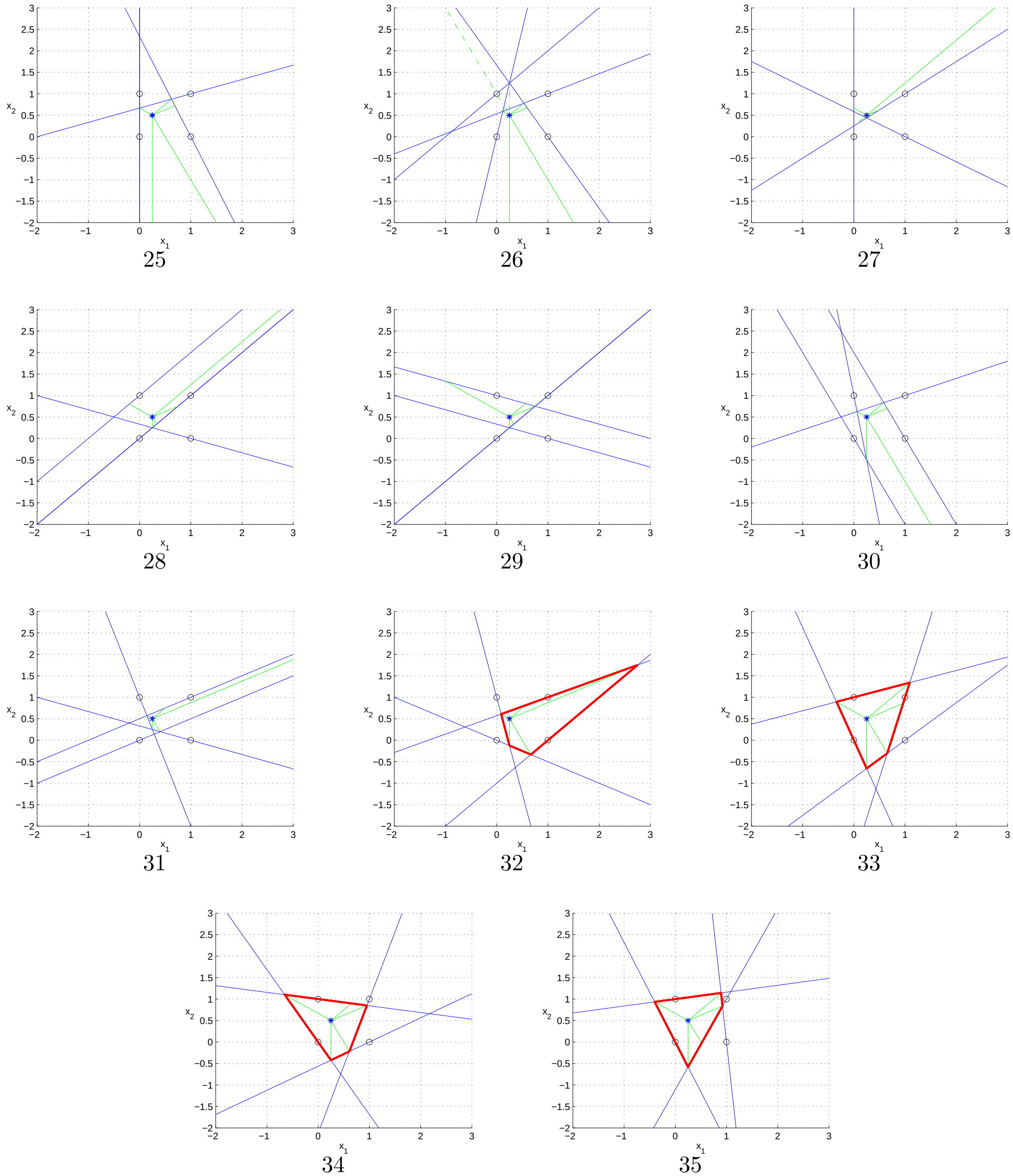
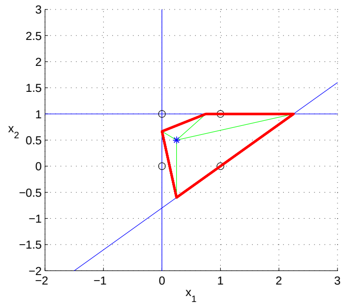
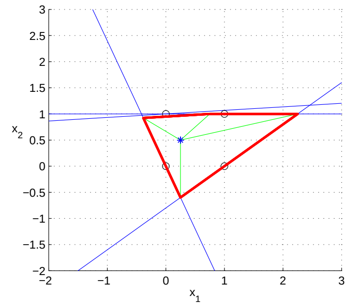


Figure 2.12: 0-1 Disjunctive Hull facets of the Andersen et al. example - continuation



(a)



(b)

Figure 2.13: Dominated and Non-dominated cuts from basic feasible CGLP solutions

What is stated in the Observation 2.4.3 can be seen in the following instance: $(f_1, f_2) = (0.3; 0.1)$, $\bar{v} = (3.2; 1.2; 6.0; 0.5)$, $\bar{w} = (0.4; 1.6; -3.55; 0.94)$ and the ray $r = (3.3; 2.1)$. The coefficient corresponding to r, computed using (2.3.29) subject to the restriction (2.3.30) is 1.44 and the optimal solution has $m_1^* = \lfloor 3.2 \rfloor = 3; m_2^* = \lfloor 2.1 \rfloor = 2$. If we allow any value $m^1, m^2 \in \mathbb{Z}$ then the optimal coefficient is 0.884 and that corresponds to the values $m_1^* = 3; m_2^* = 1$.

A different strengthening technique applies to the 0-1 case, the Monoidal Strengthening introduced by Balas and Jeroslow in [11]. We illustrate this technique on 8 conic cuts that are used in the computational experiments given later. Consider the following 8 disjunctions that are valid for P_{01}:

1. $(-x_2 \geq 0) \vee (-x_1 + x_2 \geq 0)$
2. $(-x_1 \geq 0) \vee (+x_1 - x_2 \geq 0)$
3. $(+x_2 \geq 1) \vee (-x_1 - x_2 \geq -1)$
4. $(-x_1 \geq 0) \vee (+x_1 + x_2 \geq 1)$
5. $(+x_2 \geq 1) \vee (+x_1 - x_2 \geq 0)$
6. $(+x_1 \geq 1) \vee (-x_1 + x_2 \geq 0)$
7. $(-x_2 \geq 0) \vee (+x_1 + x_2 \geq 1)$
8. $(+x_1 \geq 1) \vee (-x_1 - x_2 \geq -1)$

Substituting $x_i = f_i + \sum_{j \in J} r_j^i s_j$ for $i \in \{1, 2\}$ we get

1. $\left(\sum_{j \in J}(-r_j^2)s_j \geq f_2\right) \vee \left(\sum_{j \in J}(-r_j^1 + r_j^2)s_j \geq f_1 - f_2\right)$
2. $\left(\sum_{j \in J}(-r_j^1)s_j \geq f_1\right) \vee \left(\sum_{j \in J}(r_j^1 - r_j^2)s_j \geq f_2 - f_1\right)$
3. $\left(\sum_{j \in J}(+r_j^2)s_j \geq 1 - f_2\right) \vee \left(\sum_{j \in J}(-r_j^1 - r_j^2)s_j \geq f_1 + f_2 - 1\right)$

4. $\left(\sum_{j\in J}(-r_j^1)s_j \geq f_1\right) \vee \left(\sum_{j\in J}(r_j^1+r_j^2)s_j \geq 1-f_1-f_2\right)$

5. $\left(\sum_{j\in J}(+r_j^2)s_j \geq 1-f_2\right) \vee \left(\sum_{j\in J}(r_j^1-r_j^2)s_j \geq f_2-f_1\right)$

6. $\left(\sum_{j\in J}(+r_j^1)s_j \geq 1-f_1\right) \vee \left(\sum_{j\in J}(-r_j^1-r_j^2)s_j \geq f_1+f_2-1\right)$

7. $\left(\sum_{j\in J}(-r_j^2)s_j \geq f_2\right) \vee \left(\sum_{j\in J}(r_j^1+r_j^2)s_j \geq 1-f_1-f_2\right)$

8. $\left(\sum_{j\in J}(+r_j^1)s_j \geq 1-f_1\right) \vee \left(\sum_{j\in J}(-r_j^1+r_j^2)s_j \geq f_1-f_2\right)$

Using Disjunctive Programming we can obtain from each disjunction a disjunctive cut of the form $\alpha s \geq 1$ whose coefficients $\alpha_j, j \in J$ as given by the expressions

1. $\max\{\frac{-r_j^2}{f_2}, \frac{-r_j^1+r_j^2}{f_1-f_2}\}$
2. $\max\{\frac{-r_j^1}{f_1}, \frac{r_j^1-r_j^2}{f_2-f_1}\}$
3. $\max\{\frac{r_j^2}{1-f_2}, \frac{-r_j^1-r_j^2}{f_1+f_2-1}\}$.
4. $\max\{\frac{-r_j^1}{f_1}, \frac{r_j^1+r_j^2}{1-f_1-f_2}\}$
5. $\max\{\frac{r_j^2}{1-f_2}, \frac{r_j^1-r_j^2}{f_2-f_1}\}$
6. $\max\{\frac{r_j^1}{1-f_1}, \frac{-r_j^1+r_j^2}{f_1-f_2}\}$
7. $\max\{\frac{-r_j^2}{f_2}, \frac{r_j^1+r_j^2}{1-f_1-f_2}\}$
8. $\max\{\frac{r_j^1}{1-f_1}, \frac{-r_j^1-r_j^2}{f_1+f_2-1}\}$

Note that for any solution with x_1, x_2 fractional at most 4 of the 8 disjunctions are violated, therefore at most 4 cuts can be obtained. Now we assume that the non-basic variables $s_j, j \in J_1 \subseteq J$ are subject to integrality constraints. We can then use monoidal strengthening to strengthen the coefficients of the disjunctive cut $\alpha s \geq 1$.

Proposition 2.4.4. *Consider the monoid $M = \{m \in \mathbb{Z}^2 : m^1 + m^2 \geq 0\}$. The following disjunctions are satisfied whenever the original disjunctions are*

1. $\left(\sum_{j\in J}(-r_j^2+m_j^1)s_j \geq f_2\right) \vee \left(\sum_{j\in J}(-r_j^1+r_j^2+m_j^2)s_j \geq f_1-f_2\right)$
2. $\left(\sum_{j\in J}(-r_j^1+m_j^1)s_j \geq f_1\right) \vee \left(\sum_{j\in J}(r_j^1-r_j^2+m_j^2)s_j \geq f_2-f_1\right)$
3. $\left(\sum_{j\in J}(+r_j^2+m_j^1)s_j \geq 1-f_2\right) \vee \left(\sum_{j\in J}(-r_j^1-r_j^2+m_j^2)s_j \geq f_1+f_2-1\right)$
4. $\left(\sum_{j\in J}(-r_j^1+m_j^1)s_j \geq f_1\right) \vee \left(\sum_{j\in J}(r_j^1+r_j^2+m_j^2)s_j \geq 1-f_1-f_2\right)$

5. $\left(\sum_{j\in J}(+r_j^2+m_j^1)s_j \geq 1-f_2\right) \vee \left(\sum_{j\in J}(r_j^1-r_j^2+m_j^2)s_j \geq f_2-f_1\right)$

6. $\left(\sum_{j\in J}(+r_j^1+m_j^1)s_j \geq 1-f_1\right) \vee \left(\sum_{j\in J}(-r_j^1+r_j^2+m_j^2)s_j \geq f_1-f_2\right)$

7. $\left(\sum_{j\in J}(-r_j^2+m_j^1)s_j \geq f_2\right) \vee \left(\sum_{j\in J}(r_j^1+r_j^2+m_j^2)s_j \geq 1-f_1-f_2\right)$

8. $\left(\sum_{j\in J}(+r_j^1+m_j^1)s_j \geq 1-f_1\right) \vee \left(\sum_{j\in J}(-r_j^1-r_j^2+m_j^2)s_j \geq f_1+f_2-1\right)$

Proof. Skipped, follows from [11].

We can strengthen the coefficient $\alpha_j, j \in J_1$ as follows

1. $\min_{m_j\in M} \max\{\frac{-r_j^2+m_j^1}{f_2}, \frac{-r_j^1+r_j^2+m_j^2}{f_1-f_2}\}$
2. $\min_{m_j\in M} \max\{\frac{-r_j^1+m_j^1}{f_1}, \frac{r_j^1-r_j^2+m_j^2}{f_2-f_1}\}$
3. $\min_{m_j\in M} \max\{\frac{r_j^2+m_j^1}{1-f_2}, \frac{-r_j^1-r_j^2+m_j^2}{f_1+f_2-1}\}$.
4. $\min_{m_j\in M} \max\{\frac{-r_j^1+m_j^1}{f_1}, \frac{r_j^1+r_j^2+m_j^2}{1-f_1-f_2}\}$
5. $\min_{m_j\in M} \max\{\frac{r_j^2+m_j^1}{1-f_2}, \frac{r_j^1-r_j^2+m_j^2}{f_2-f_1}\}$
6. $\min_{m_j\in M} \max\{\frac{-r_j^2+m_j^1}{f_2}, \frac{r_j^1+r_j^2+m_j^2}{1-f_1-f_2}\}$
7. $\min_{m_j\in M} \max\{\frac{r_j^1+m_j^1}{1-f_1}, \frac{-r_j^1+r_j^2+m_j^2}{f_1-f_2}\}$
8. $\min_{m_j\in M} \max\{\frac{r_j^1+m_j^1}{1-f_1}, \frac{-r_j^1-r_j^2+m_j^2}{f_1+f_2-1}\}$

2.5 Computational Experiments

In this section we present computational experiments with cuts derived from from fixed configurations of the parametric octahedron. We assess the strength of the cuts by analyzing the gap closed on instances from the MIPLIB3_C_V2 [42] when used in combination with standard Gomory cuts. MIPLIB3_C_V2 is a collection of 68 instances by Margot which are slight variations of the standard MIPLIB3 [45] and for which the validity of a provided feasible solution can be checked in finite precision arithmetic. We restricted the collection of instances to a subset of 41 instances. The considered instances are such that they contain at least 2 binary variables currently fractional in the basis of the root relaxation and the cut generation procedure on each round takes less than 3600 seconds.

We separated the following two families of cuts

- Cuts from 4 Triangles of type 1 whose vertices are expressed in terms of x_1, x_2 coordinates (shown in Figure 2.14):

- $(0,0); (2,0); (0,2)$
- $(-1,0); (1,0); (1,2)$
- $(0,-1); (2,1); (0,1)$
- $(-1,1); (1,1); (1,-1)$

- Cuts from 8 cones (shown in Figure 2.15):
  - apex at $(0,0)$ and rays $(1,0),(1,1)$
  - apex at $(0,0)$ and rays $(0,1),(1,1)$
  - apex at $(0,1)$ and rays $(-1,0),(-1,1)$
  - apex at $(0,1)$ and rays $(0,1),(-1,1)$
  - apex at $(1,1)$ and rays $(-1,0),(-1,-1)$
  - apex at $(1,1)$ and rays $(0,-1),(-1,-1)$
  - apex at $(0,1)$ and rays $(1,0),(1,-1)$
  - apex at $(0,1)$ and rays $(-1,0),(1,-1)$

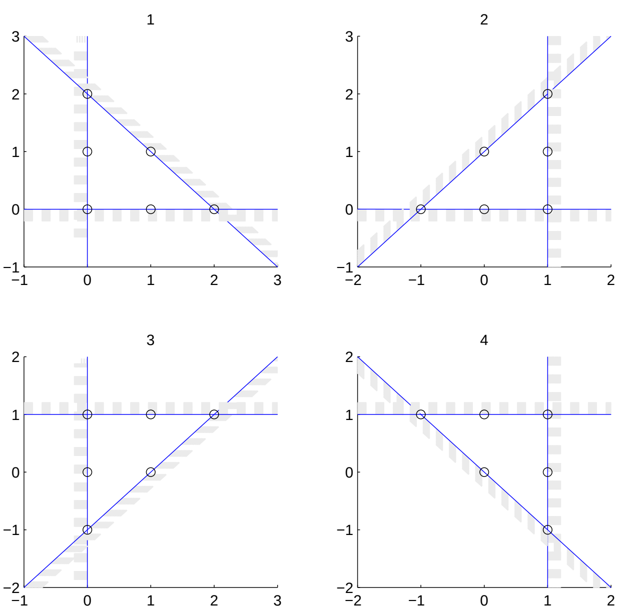


Figure 2.14: Fixed shape Triangles of type 1

We tested our cuts from Triangles of Type 1 and fixed-shape cones using the procedure described in procedure *SeparateFixedParametricOctahedra*. We considered 5 rounds of cuts, in each round we separated Gomory cuts and depending on the parameter flags `SeparateTriangles`, `SeparateCones` we separate respectively cuts from the Triangles of Type 1 and cuts from the fixed-shape cones described above. Strengthening of cuts from

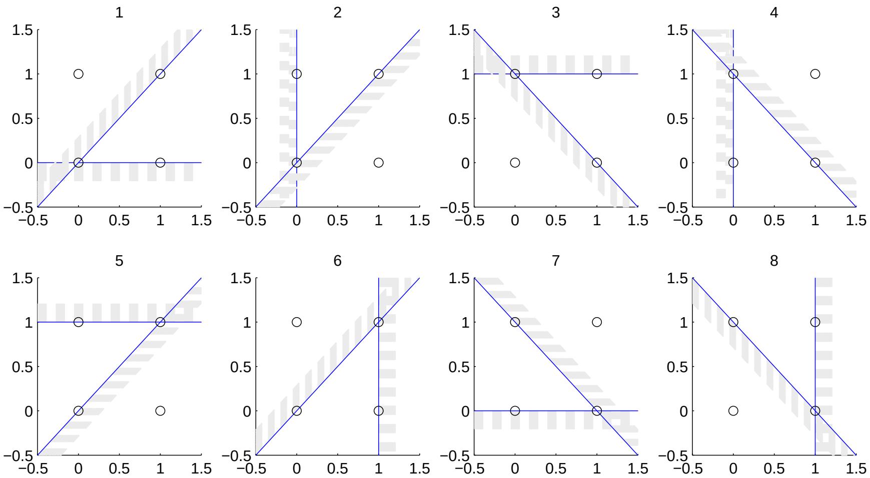


Figure 2.15: Fixed shape cones

Triangles of Type 1 is performed via modularization of the rays $r_j, j \in J_1$ as shown in Theorem 2.3.10 whereas for the cuts from cones we use monoidal cut strengthening as illustrated in 2.4.2. If the flag `StrengthenCuts` is set to true, the cuts are strengthened using integrality of the non-basic variables. In each round, after reoptimizing the relaxation with the new cuts, we remove the inequalities that are not tight for the current solution and are not part of initial the root relaxation.

The Gomory cut generator we used is *GomoryTwo* which is developed by Margot and available at [15]. We used the following configuration (the descriptions of the parameters is taken from [41]):

- `LUB` $= 10^4$: If the absolute value of the upper bound on a variable is larger than that, it is considered large.

- `EPS_COEFF` $= 10^{-5}$: Any cut coefficient smaller than that for a variable that does not have a large upper bound is replaced by zero.

- `EPS_COEFF_LUB` $= 10^{-13}$: Similar to `EPS_COEFF` for variables having a large upper bound.

- `MAXDYN` $= 10^8$: A cut is discarded if none of the variables with nonzero coefficient have a large upper bound and its dynamism is larger that this value.

- `MAXDYN_LUB` $= 10^{13}$: Similar to `MAXDYN`, but for cuts where some of the variables with nonzero coefficients have a large upper bound.

- `AWAY` $= 0.05$: Lower bound on the absolute value of $\min\{f; 1-f\}$.

- `MINVIOL` $= 10^{-4}$: If the violation of the cut by the current optimal solution of the LP relaxation is lower than this number, the cut is discarded.

The cut generators we implemented to separate Triangles of Type 1 and the fixed Cones are used in conjunction with a cut validator that was configured to use the same values of the parameters above. We ran 6 different experiments, one for each combination of the 2 cut generators with or without strengthening enabled. Detailed results by instance are given for each experiment for the first round of cuts in Tables 2.4, 2.5 and 2.6 for the strengthened version of the cuts. In Tables 2.7, 2.8 and 2.9 we give the detailed results for the unstrengthened versions of the cuts. We denote by "G" the Mixed Integer Gomory cuts, by "T" the cuts from Triangles of Type 1 and by "C" the cuts derived from the fixed cones. In each table the columns "X%" show the gap closed by the combination of cut generators given in "X", the columns "X add#" and "X del#" denote the number of cuts added and removed after the reoptimization phase. The column "impr.%" indicates the percentage improvement produced by adding the combination of cuts "X" on top of the Mixed Integer Gomory cuts. The number of Triangular and Conic cuts is much larger than that of Gomory cuts. Let n be the number of basic variables subject to integrality constraints. In this setting up to n Gomory cuts are generated (one for each variable with a current fractional value). For the other two families of cuts we consider up to $n(n-1)/2$ pairs of variables and for each pair we separate up to 4 Triangular cuts and 4 Conic cuts. The times to generate Gomory cuts is, for the same reason, much lower than the time to generate the other two families of cuts. Aggregate results for 5 rounds of cuts are given in Table 2.10.

The results show that the added families of cuts have a significant impact on the root relaxations when used in combination of Gomory cuts. On average, using the two families of strengthened cuts yields an improvement of 71.31% over the standard Gomory cuts. Moreover, as seen in Table 2.10, the same behaviour applies for additional rounds of cuts, i.e. the improvement does not appear to decrease with iterative rounds. It actually increments substantially to 84.23% in the third round of cuts.

We also ran the same experiments using a different Gomory cut generator, *CglGomory* part of the CGL package of Coin-OR [23]. This cut generator is less aggressive on average than *GomoryTwo*. Table 2.11 summarizes the average gap closed and average improvement of the Triangle and Conic strengthened cuts added on top of the Gomory cuts. The improvement of the new cuts over the Gomory cuts computed via *CoinCglGomory* is slightly higher than with the cuts computed using the *GomoryTwo* implementation (aggregate results by round given in Table 2.10). For brevity we omit the comparisons for the other different combinations of cuts and strengthening. The outcome for those shows a similar behavior.

As an alternative way to assess how the cuts we consider perform in comparison to Gomory cuts [43], we can determine the minimum number of rounds of Gomory cuts needed to close at least the same amount of gap closed by the Gomory, Triangles of Type 1 and the Conic cuts. Using *CglGomoryTwo* we need an average of 2.2 rounds of Gomory cuts to get at least the same gap as the families of cuts we consider. We limited our computation to 30 rounds of Gomory cuts. For 6 instances out of 41 the Gomory gap never reached the gap closed by Triangles and Cones. For the computation of the average, we discarded those 6 instances.

SeparateFixedParametricOctahedra(r, f)

Solve LP relaxation P
for $k \leftarrow 1$ **up to** 5
    do
        Inizialize cut collection $C \leftarrow$ empty
        for each x_i in basis, subject to integrality constraint and fractional in the current basic solution
            do
                Compute Gomory cut G_i
                $C \leftarrow C \cup G_i$
        for each pair x_i, x_j in basis, subject to integrality constraints and with at least one fractional in the current basic solution
            do
                if SeparateTriangles==true
                  then separate the cuts $T^1_{ij}, \ldots, T^4_{ij}$ from each of the 4 triangles of type 1 that contain the fractional solution in their interior
                      if StrengthenCuts==true
                        then strengthen the cuts $T^1_{ij}, \ldots, T^4_{ij}$ via modularization
                      $C \leftarrow C \cup T^1_{ij}, \ldots, T^4_{ij}$
                if SeparateCones==true
                  then separate the cuts $K^1_{ij}, \ldots, K^8_{ij}$ from each of the 8 cones that contain the fractional solution in their interior
                      if StrengthenCuts==true
                        then strengthen the cuts $K^1_{ij}, \ldots, K^8_{ij}$ via monoidal cut strengthening
                      $C \leftarrow C \cup K_1, \ldots, K^8_{ij}$
        Resolve P and get new solution $\bar{x}^k$ with value $\overline{opt}^k$
        Remove from P the cuts in C that are not tight at $\bar{x}^k$

For the computations we used a single core of a Linux 2.6.32-32 computer with an Intel Core 2 Duo 2.66GHz with 4GB of memory. We used Cbc version 2.4.0 and Clp version 1.11 from the optimization suite Coin-OR [23].

2.5.1 Cut validity and statistical testing

Recently in [41] Margot proposed a method for testing new cut generators that goes beyond the comparison in terms of strength and gap closed of the new families of cuts versus some standard references, e.g. Mixed Integer Gomory or Reduce-and-split cuts. Margot developed a code (available in [15]) that can be used to assess the validity of the new cut generators keeping track of failures that might occur such as generation of invalid cuts. It also performs statistical testing to compare the strength of the cut generators using the nonparametric *Quade* test within the statistical software `R` version 2.10.1 (2009-12-14). The code can be used with any cut generator that conforms to the CGL package of Coin-OR [23]. The statistical testing code on all 6 combinations of cuts/strengthening shows that the improvement given by the cuts considered after one round at depth 0 of the branch-and-bound tree is statistically significant with a 99% confidence level. As for the validity of the cuts, we tested the combination of cuts from Triangles of Type 1 and fixed Cones with strengthening enabled on the 41 instances considered. On the instances *dcmulti,qiu* the GomoryTwo cut generator had a failure of type 2, i.e. during the random diving towards a feasible solution an invalid cut was generated. When the additional two cut generators were considered, in addition to the previous two instances, a failure of type 2 also occurred for the instance *pp08a*.

Table 2.4: MIGs + Type 1 Triangles + Conic cuts, Strengthening, 1 round

Instance	G%	G add#	G del#	GTC%	GTC add#	GTC del#	impr.%
air03	100	5	1	100	12272	12196	0.00
cap6000	41.65	2	1	41.65	1357	1355	0.00
danoint	0.26	24	14	0.26	24	14	0.00
dcmulti	45.40	48	13	46.17	708	669	1.70
egout	21.45	15	0	61.13	3505	3472	184.99
enigma	100	1	0	100	336	335	0.00
fiber	51.08	20	8	61.41	23163	23100	20.22
fixnet3	6.62	6	0	56.19	2925	2827	748.79
fixnet4	4.79	6	0	13.02	2830	2732	171.82
fixnet6	3.98	6	0	13.03	2502	2388	227.39
khb05250	73.59	18	0	84.21	1434	1404	14.43
l152lav	10.87	12	2	24.87	15693	15610	128.79
lseu	55.19	12	7	55.19	569	563	0.00
markshare1	0	6	3	0	124	110	0.00
markshare2	0	7	3	0	173	147	0.00
mas74	6.52	9	0	7.50	511	490	15.03
mas76	6.36	9	0	7.66	433	411	20.44
misc03	8.62	14	12	8.62	2488	2483	0.00
misc06	26.09	9	0	26.09	73	5	0.00
misc07	0	21	19	0	4198	4193	0.00
mod008	20.10	4	0	20.27	96	90	0.85
mod010	100	5	1	100	14783	14641	0.00
mod011	27.87	15	3	32.75	867	178	17.51
modglob	13.32	16	2	16.26	137	41	22.07
p0033	12.60	5	1	57.04	174	168	352.70
p0201	16.89	18	11	19.31	1931	1927	14.33
p0282	3.47	24	17	6.20	2837	2829	78.67
p0548	3.06	19	2	18.53	5584	5520	505.56
p2756	0.21	7	1	0.56	908	885	166.67
pk1	0	15	3	0	34	20	0.00
pp08a	54.30	50	0	65.30	7822	7730	20.26
pp08aCUTS	33.13	43	0	41.25	2591	2527	24.51
qiu	0.33	36	25	0.33	36	25	0.00
rentacar	0	2	0	0	7	5	0.00
rgn	3.15	17	10	3.15	1341	1331	0.00
set1ch	30.36	125	1	44.47	27465	27206	46.48
stein27	0	21	18	0	1257	1254	0.00
stein45	0	35	30	0	3584	3576	0.00
swath	5.50	13	0	9.10	7538	7515	65.45
vpm1	20.73	12	0	23.94	1073	1047	15.48
vpm2	10.14	27	3	16.35	4431	4380	61.24
Average	22.38	18.51	5.15	28.82	3897.90	3839	71.35

Table 2.5: MIGs + Type 1 Triangles, Strengthening, 1 round

Instance	G%	G add#	G del#	GT%	GT add#	GT del#	impr.%
air03	100	5	1	100	5437	5361	0.00
cap6000	41.65	2	1	41.65	361	359	0.00
danoint	0.26	24	14	0.26	24	14	0.00
dcmulti	45.40	48	13	45.88	225	190	1.06
egout	21.45	15	0	60.76	2038	2007	183.26
enigma	100	1	0	100	138	137	0.00
fiber	51.08	20	8	60.22	9919	9880	17.89
fixnet3	6.62	6	0	47.01	1571	1479	610.12
fixnet4	4.79	6	0	12.49	1524	1427	160.75
fixnet6	3.98	6	0	12.03	1352	1240	202.26
khb05250	73.59	18	0	84.21	721	693	14.43
l152lav	10.87	12	2	11.09	6970	6885	2.02
lseu	55.19	12	7	55.19	291	286	0.00
markshare1	0	6	3	0	66	56	0.00
markshare2	0	7	3	0	91	69	0.00
mas74	6.52	9	0	7.44	261	239	14.11
mas76	6.36	9	0	7.12	225	201	11.95
misc03	8.62	14	12	8.62	1185	1182	0.00
misc06	26.09	9	0	26.09	29	0	0.00
misc07	0	21	19	0	1915	1912	0.00
mod008	20.10	4	0	20.11	48	43	0.05
mod010	100	5	1	100	6089	5947	0.00
mod011	27.87	15	3	32.47	439	87	16.51
modglob	13.32	16	2	15.76	107	13	18.32
p0033	12.60	5	1	57.04	85	80	352.70
p0201	16.89	18	11	19.31	1233	1228	14.33
p0282	3.47	24	17	5.38	1416	1409	55.04
p0548	3.06	19	2	17.37	2958	2914	467.65
p2756	0.21	7	1	0.56	546	524	166.67
pk1	0	15	3	0	25	11	0.00
pp08a	54.30	50	0	64.89	3859	3771	19.50
pp08aCUTS	33.13	43	0	41	1136	1073	23.75
qiu	0.33	36	25	0.33	36	25	0.00
rentacar	0	2	0	0	4	2	0.00
rgn	3.15	17	10	3.15	697	689	0.00
set1ch	30.36	125	1	44.26	12086	11822	45.78
stein27	0	21	18	0	846	843	0.00
stein45	0	35	30	0	2395	2389	0.00
swath	5.50	13	0	9.10	3678	3655	65.45
vpm1	20.73	12	0	21.94	480	458	5.84
vpm2	10.14	27	3	15.42	2160	2118	52.07
Average	22.38	18.51	5.15	28.00	1821.12	1773.61	61.50

Table 2.6: MIGs + Conic cuts, Strengthening, 1 round

Instance	G%	G add#	G del#	GC%	GC add#	GC del#	impr.%
air03	100	5	1	100	6840	6764	0.00
cap6000	41.65	2	1	41.65	998	997	0.00
danoint	0.26	24	14	0.26	24	14	0.00
dcmulti	45.40	48	13	46	531	496	1.32
egout	21.45	15	0	35.15	1482	1456	63.87
enigma	100	1	0	100	199	198	0.00
fiber	51.08	20	8	52.03	13264	13178	1.86
fixnet3	6.62	6	0	49.30	1360	1266	644.71
fixnet4	4.79	6	0	10.41	1312	1211	117.33
fixnet6	3.98	6	0	11.51	1156	1051	189.20
khb05250	73.59	18	0	77.62	731	704	5.48
l152lav	10.87	12	2	24.87	8735	8650	128.79
lseu	55.19	12	7	55.19	290	284	0.00
markshare1	0	6	3	0	64	55	0.00
markshare2	0	7	3	0	89	73	0.00
mas74	6.52	9	0	7.15	259	236	9.66
mas76	6.36	9	0	7.62	217	190	19.81
misc03	8.62	14	12	8.62	1317	1312	0.00
misc06	26.09	9	0	26.09	53	1	0.00
misc07	0	21	19	0	2304	2298	0.00
mod008	20.10	4	0	20.27	52	46	0.85
mod010	100	5	1	100	8699	8557	0.00
mod011	27.87	15	3	28.60	443	60	2.62
modglob	13.32	16	2	13.94	46	2	4.65
p0033	12.60	5	1	24.59	94	89	95.16
p0201	16.89	18	11	16.89	716	708	0.00
p0282	3.47	24	17	5.40	1445	1436	55.62
p0548	3.06	19	2	6.53	2645	2599	113.40
p2756	0.21	7	1	0.21	369	346	0.00
pk1	0	15	3	0	24	11	0.00
pp08a	54.30	50	0	57.01	4013	3933	4.99
pp08aCUTS	33.13	43	0	34.32	1498	1424	3.59
qiu	0.33	36	25	0.33	36	25	0.00
rentacar	0	2	0	0	5	3	0.00
rgn	3.15	17	10	3.15	661	651	0.00
set1ch	30.36	125	1	37.43	15504	15243	23.29
stein27	0	21	18	0	432	428	0.00
stein45	0	35	30	0	1224	1212	0.00
swath	5.50	13	0	7.14	3873	3848	29.82
vpm1	20.73	12	0	22.73	605	577	9.65
vpm2	10.14	27	3	14.23	2298	2244	40.34
Average	22.38	18.51	5.15	25.52	2095.29	2045.76	38.20

Table 2.7: MIGs + Type 1 Triangles + Conic cuts, No strengthening, 1 round

Instance	G%	G add#	G del#	GTC%	GTC add#	GTC del#	impr.%
air03	100	5	1	100	13107	13031	0.00
cap6000	41.65	2	1	41.65	1442	1440	0.00
danoint	0.26	24	14	0.26	24	14	0.00
dcmulti	45.40	48	13	45.95	708	668	1.21
egout	21.45	15	0	61.13	3520	3485	184.99
enigma	100	1	0	100	343	342	0.00
fiber	51.08	20	8	52	26050	25968	1.80
fixnet3	6.62	6	0	56.19	2925	2827	748.79
fixnet4	4.79	6	0	10.40	2830	2733	117.12
fixnet6	3.98	6	0	11.27	2502	2378	183.17
khb05250	73.59	18	0	84.21	1434	1404	14.43
l152lav	10.87	12	2	13.23	16168	16079	21.71
lseu	55.19	12	7	55.19	584	579	0.00
markshare1	0	6	3	0	124	118	0.00
markshare2	0	7	3	0	173	164	0.00
mas74	6.52	9	0	6.67	511	497	2.30
mas76	6.36	9	0	6.36	433	421	0.00
misc03	8.62	14	12	8.62	2534	2529	0.00
misc06	26.09	9	0	26.09	73	5	0.00
misc07	0	21	19	0	4310	4305	0.00
mod008	20.10	4	0	20.10	96	90	0.00
mod010	100	5	1	100	15812	15670	0.00
mod011	27.87	15	3	28.89	867	103	3.66
modglob	13.32	16	2	13.93	137	40	4.58
p0033	12.60	5	1	12.82	182	176	1.75
p0201	16.89	18	11	16.89	2107	2099	0.00
p0282	3.47	24	17	6.19	2878	2870	78.39
p0548	3.06	19	2	18.10	5650	5586	491.50
p2756	0.21	7	1	0.36	908	877	71.43
pk1	0	15	3	0	34	22	0.00
pp08a	54.30	50	0	64.36	7822	7733	18.53
pp08aCUTS	33.13	43	0	37.89	2591	2523	14.37
qiu	0.33	36	25	0.33	36	25	0.00
rentacar	0	2	0	0	7	5	0.00
rgn	3.15	17	10	3.15	1341	1333	0.00
set1ch	30.36	125	1	44.37	27465	27196	46.15
stein27	0	21	18	0	1257	1254	0.00
stein45	0	35	30	0	3584	3577	0.00
swath	5.50	13	0	5.65	7790	7757	2.73
vpm1	20.73	12	0	20.73	1073	1045	0.00
vpm2	10.14	27	3	15.80	4431	4361	55.82
Average	22.38	18.51	5.15	26.56	4045.44	3983.63	50.35

Table 2.8: MIGs + Type 1 Triangles, No strengthening, 1 round

Instance	G%	G add#	G del#	GT%	GT add#	GT del#	impr.%
air03	100	5	1	100	5486	5410	0.00
cap6000	41.65	2	1	41.65	497	495	0.00
danoint	0.26	24	14	0.26	24	14	0.00
dcmulti	45.40	48	13	45.88	225	190	1.06
egout	21.45	15	0	60.76	2038	2006	183.26
enigma	100	1	0	100	145	144	0.00
fiber	51.08	20	8	51.82	10057	9994	1.45
fixnet3	6.62	6	0	47.01	1571	1479	610.12
fixnet4	4.79	6	0	8.80	1524	1431	83.72
fixnet6	3.98	6	0	10.46	1352	1238	162.81
khb05250	73.59	18	0	84.21	721	693	14.43
l152lav	10.87	12	2	10.87	7406	7316	0.00
lseu	55.19	12	7	55.19	297	292	0.00
markshare1	0	6	3	0	66	62	0.00
markshare2	0	7	3	0	91	85	0.00
mas74	6.52	9	0	6.52	261	249	0.00
mas76	6.36	9	0	6.36	225	213	0.00
misc03	8.62	14	12	8.62	1223	1221	0.00
misc06	26.09	9	0	26.09	29	0	0.00
misc07	0	21	19	0	2008	2006	0.00
mod008	20.10	4	0	20.10	48	42	0.00
mod010	100	5	1	100	6984	6842	0.00
mod011	27.87	15	3	28.67	439	53	2.87
modglob	13.32	16	2	13.33	107	11	0.08
p0033	12.60	5	1	12.72	85	79	0.95
p0201	16.89	18	11	16.89	1310	1303	0.00
p0282	3.47	24	17	5.38	1416	1410	55.04
p0548	3.06	19	2	16.67	2958	2913	444.77
p2756	0.21	7	1	0.36	546	515	71.43
pk1	0	15	3	0	25	13	0.00
pp08a	54.30	50	0	64.32	3859	3773	18.45
pp08aCUTS	33.13	43	0	37.82	1136	1071	14.16
qiu	0.33	36	25	0.33	36	25	0.00
rentacar	0	2	0	0	4	2	0.00
rgn	3.15	17	10	3.15	697	690	0.00
set1ch	30.36	125	1	44.20	12086	11822	45.59
stein27	0	21	18	0	846	843	0.00
stein45	0	35	30	0	2395	2389	0.00
swath	5.50	13	0	5.64	3744	3711	2.55
vpm1	20.73	12	0	20.73	480	459	0.00
vpm2	10.14	27	3	14.60	2160	2116	43.98
Average	22.38	18.51	5.15	26.08	1868.46	1820	42.85

Table 2.9: MIGs + Conic cuts, No strengthening, 1 round

Instance	G%	G add#	G del#	GC%	GC add#	GC del#	impr.%
air03	100	5	1	100	7626	7550	0.00
cap6000	41.65	2	1	41.65	947	945	0.00
danoint	0.26	24	14	0.26	24	14	0.00
dcmulti	45.40	48	13	45.79	531	496	0.86
egout	21.45	15	0	35.03	1497	1471	63.31
enigma	100	1	0	100	199	198	0.00
fiber	51.08	20	8	51.29	16013	15934	0.41
fixnet3	6.62	6	0	49.30	1360	1266	644.71
fixnet4	4.79	6	0	10.41	1312	1214	117.33
fixnet6	3.98	6	0	9.95	1156	1040	150.00
khb05250	73.59	18	0	77.62	731	705	5.48
l152lav	10.87	12	2	13.23	8774	8684	21.71
lseu	55.19	12	7	55.19	299	294	0.00
markshare1	0	6	3	0	64	59	0.00
markshare2	0	7	3	0	89	83	0.00
mas74	6.52	9	0	6.67	259	245	2.30
mas76	6.36	9	0	6.36	217	205	0.00
misc03	8.62	14	12	8.62	1325	1320	0.00
misc06	26.09	9	0	26.09	53	1	0.00
misc07	0	21	19	0	2323	2318	0.00
mod008	20.10	4	0	20.10	52	46	0.00
mod010	100	5	1	100	8833	8691	0.00
mod011	27.87	15	3	27.92	443	60	0.18
modglob	13.32	16	2	13.92	46	2	4.50
p0033	12.60	5	1	12.82	102	96	1.75
p0201	16.89	18	11	16.89	815	807	0.00
p0282	3.47	24	17	5.40	1486	1477	55.62
p0548	3.06	19	2	6.37	2711	2666	108.17
p2756	0.21	7	1	0.21	369	358	0.00
pk1	0	15	3	0	24	12	0.00
pp08a	54.30	50	0	56.84	4013	3933	4.68
pp08aCUTS	33.13	43	0	33.48	1498	1426	1.06
qiu	0.33	36	25	0.33	36	25	0.00
rentacar	0	2	0	0	5	3	0.00
rgn	3.15	17	10	3.15	661	653	0.00
set1ch	30.36	125	1	37.37	15504	15243	23.09
stein27	0	21	18	0	432	428	0.00
stein45	0	35	30	0	1224	1212	0.00
swath	5.50	13	0	5.59	4059	4027	1.64
vpm1	20.73	12	0	20.73	605	578	0.00
vpm2	10.14	27	3	14.01	2298	2251	38.17
Average	22.38	18.51	5.15	24.70	2195.49	2147.22	30.36

Table 2.10: MIGs + Type 1 Triangles + Conic cuts, Strengthening, 5 rounds

round	G%	GTC%	impr.%
1	22.38	28.82	71.35
2	28.22	33.98	84.08
3	31.07	36.68	84.23
4	32.83	37.78	78.76
5	34.61	38.86	77.28

Table 2.11: CoinCglGomory + Type 1 Triangles + Conic cuts, Strengthening, 5 rounds

round	G%	GTC%	impr.%
1	19.49	29.06	71.46
2	24.94	34.01	93.23
3	27.87	36.70	89.57
4	29.57	37.95	86.19
5	30.48	38.78	85.04

Chapter 3

Monoidal Cut Strengthening Revisited

3.1 Introduction

Consider the q-term disjunction

$$\bigvee_{i \in Q} \left(\sum_{j \in J} a_{ij} x_j \geq a_{i0} \right), \tag{3.1.1}$$

where $x_j \geq 0, j \in J$ and $a_{i0} > 0, i \in Q$. For convenience of notation let $q = |Q|$. It is well known [6] that (3.1.1) yields a disjunctive cut of the form $\beta x \geq 1$, with

$$\beta_j = \max_{i \in Q} \left\{ \frac{a_{ij}}{a_{i0}} \right\}. \tag{3.1.2}$$

If $x_j \in \mathbb{Z}, j \in J_1 \subseteq J$, then (3.1.2) can be strengthened. There exist two distinct strengthening procedures for disjunctive cuts. The first is based on the standard modularization of the coefficients a_{ij} and is used in [31] among others; in the rest of the chapter we refer to this technique as the "standard strengthening". The second is the monoidal strengthening of Balas and Jeroslow [11]. In the literature [28, 31, 32], Monoidal cut strengthening is treated as equivalent to "standard strengthening" and indeed if one uses the set of integer points as monoid in Theorem 1 of [11] the strengthened coefficients are the same. But Monoidal cut strengthening is more general as changing the monoid adopted results in different cut strengthenings. In this chapter we introduce a variation of [11] which yields different cuts, that sometimes dominate the cuts resulting from both of the above procedures.

3.1.1 Monoidal Cut Strengthening

Monoidal cut strengthening can be outlined as follows. If valid lower bounds $b_i \geq 0$ for the quantities $\sum_{j \in J} a_{ij} x_j, i \in Q$, are known, then monoidal cut strengthening can be applied to obtain a stronger cut $\bar{\beta}x \geq 1$, with

$$\bar{\beta}_j := \begin{cases} \min_{m_j \in M} \max_{i \in Q} \left\{ \frac{a_{ij} + (a_{i0} - b_i) m_j^i}{a_{i0}} \right\} & j \in J_1 \\ \beta_j & j \in J \setminus J_1 \end{cases} \tag{3.1.3}$$

where M is the monoid $M := \{m \in \mathbb{Z}^q : \sum_{i \in Q} m^i \geq 0\}$. The validity of $\bar{\beta}x \geq 1$ follows from the following

Proposition 3.1.1. *Balas, Jeroslow [11]. Any x satisfying $x_j \geq 0, j \in J, x_j \in \mathbb{Z}, j \in J_1$ and (3.1.1) such that $\sum_{j \in J} a_{ij} x_j \geq b_i, i \in Q$, also satisfies the disjunction*

$$\bigvee_{i \in Q} \left(\sum_{j \in J_1} \left(a_{ij} + (a_{i0} - b_i) m_j^i \right) x_j + \sum_{j \in J \setminus J_1} a_{ij} x_j \geq a_{i0} \right). \tag{3.1.4}$$

Proof. We need to consider 3 cases based on the value of $\sum_{j \in J_1} m_j^i x_j, i \in Q$.

- If $\sum_{j \in J_1} m_j^i x_j = 0, \forall i \in Q$ then the disjunctions (3.1.1) and (3.1.4) are equivalent.
- If $\sum_{j \in J_1} m_j^i x_j \geq 1$, for some $i \in Q$ the i-th term of (3.1.4) becomes

  $$\textstyle\sum_{j \in J} a_{ij} x_j \geq a_{i0} - (a_{ij} - b_i) \sum_{j \in J_1} m_j^i x_j$$

  which is satisfied since $\sum_{j \in J} a_{ij} x_j \geq b_i$ holds by assumption.
- If $\sum_{j \in J_1} m_j^i x_j \leq -1$ for some $i \in Q$
  then $\exists i' \in Q : \sum_{j \in J_1} m_j^{i'} x_j \geq 1$ since the monoidal condition $\sum_{i \in Q} m_j^i \geq 0$ and $x_j \geq 0, j \in J_1$ hold.
  The previous case applies to i'.

□

Applying the formula (3.1.2) to the disjunction (3.1.4) substituted for (3.1.1) yields $\bar{\beta}x \geq 1$ with coefficients (3.1.3). A glance at expression (3.1.3) suggests that the role of the integers $m_j^i, i \in Q$, in strengthening $\bar{\beta}_j$ consists in reducing the value of the largest term in brackets "at the cost" of increasing the values of several smaller terms, this limit being enforced by the condition $\sum_{i \in Q} m_j^i \geq 0$. The more terms there are, the lesser the amount by which the value of each term has to be increased in order to offset a given decrease in the value of the largest term. This suggests that from the point of view of monoidal strengthening, there may be an advantage in weakening a disjunction by adding extra terms to it. While a weaker disjunction can only yield a weaker (unstrengthened) cut, applying to such a cut the monoidal strengthening procedure may result in a stronger cut than the one obtained by applying the same strengthening procedure to the cut from the original disjunction.

In this chapter we characterize the family of cuts obtainable through this technique (Section 3.2). We then apply the results to the special case of simple split disjunctions (Section 3.2.1), and to a class of intersection cuts from two rows of the simplex tableau (Section 3.2.2). In both instances we specify conditions under which the new cuts have smaller coefficients than the cuts obtained by both the standard and the monoidal strengthening procedures. In Section 3.2.3 we briefly discuss the case of cuts obtained by strictly weakening a disjunction.

3.2 Lopsided cuts

The next Theorem introduces a new class of valid cuts derived from disjunctions equivalent to (3.1.1) but with additional redundant terms.

Theorem 3.2.1. *For each $k \in Q$, the cut $\tilde{\beta}^k x \geq 1$, with*

$$\tilde{\beta}^k_j := \begin{cases} \min\left\{ \frac{a_{kj}+a_{k0}-b_k}{a_{k0}}, \min_{\substack{m_j \in M \\ m_j^k \geq 0}} \max_{i \in Q} \left\{ \frac{a_{ij}+(a_{i0}-b_i)m_j^i}{a_{i0}} \right\} \right\} & j \in J_1 \\ \beta_j & j \in J \setminus J_1 \end{cases} \tag{3.2.1}$$

is valid.

Proof. Consider the disjunction

$$\bigvee_{i \in Q} \left(\sum_{j \in J} a_{ij}x_j \geq a_{i0} \right) \vee \underbrace{\left(\sum_{j \in J} a_{kj}x_j \geq a_{k0} \right) \vee \cdots \vee \left(\sum_{j \in J} a_{kj}x_j \geq a_{k0} \right)}_{r \text{ terms}}. \tag{3.2.2}$$

(3.2.2) contains $(q+r)$ terms of which $(r+1)$ are copies of the k-th term of (3.1.1). Adding new terms to a given disjunction in general weakens the latter, hence is a legitimate operation. If the new terms are just replicas of an existing term, then the operation leaves the solution set of the disjunction unchanged. The number r of replicated terms does not affect this reasoning, and will be specified later. By monoidal strengthening applied to (3.2.2) and Proposition 3.1.1 a cut $\gamma^k x \geq 1$ with coefficients

$$\gamma^k_j(m_j) := \begin{cases} \max\left\{ \max_{i \in Q} \left\{ \frac{a_{ij}+(a_{i0}-b_i)m_j^i}{a_{i0}} \right\}, \frac{a_{kj}+(a_{k0}-b_k)m_j^{q+1}}{a_{k0}}, \ldots, \frac{a_{kj}+(a_{k0}-b_k)m_j^{q+r}}{a_{k0}} \right\} & j \in J_1 \\ \beta_j & j \in J \setminus J_1 \end{cases} \tag{3.2.3}$$

is valid for any $m_j \in M' = \{m \in \mathbb{Z}^{q+r} : \sum_{i=1}^{q+r} m^i \geq 0\}, j \in J_1$. Note that for $j \in J \setminus J_1$ the coefficients $\gamma^k_j = \tilde{\beta}^k_j = \beta_j$ for $j \in J \setminus J_1$ are not affected by the strengthening. For $j \in J_1$ we will show that there exists $m_j \in M'$ such that $\gamma^k_j(m_j) = \tilde{\beta}^k_j$. We consider two cases: one where we set $m_j^i = 1, i \in \{k, q+1, q+2, \ldots, q+r\}$ and the other where we allow $m_j^i = 0, i \in \{k, q+1, q+2, \ldots, q+r\}$ as this might yield better coefficients. Let

$$\bar{m}^i_{1j} := \begin{cases} \bar{t}^i_j = \max\left\{ t \in \mathbb{Z} : \frac{a_{ij}+(a_{i0}-b_i)t}{a_{i0}} \leq \frac{a_{kj}+a_{k0}-b_k}{a_{k0}} \right\} & i \in Q \setminus \{k\} \\ 1 & i \in \{k\} \cup \{q+1, \ldots, q+r\}. \end{cases}$$

Replacing the values m_j with $\bar{m}_{1j}$ in (3.2.3) all the ratios corresponding to the terms different than k are reduced to a value less than or equal to the smallest ratio $\frac{a_{kj}+a_{k0}-b_k}{a_{k0}}$, therefore $\gamma_j^k(\bar{m}_{1j}) = \frac{a_{kj}+a_{k0}-b_k}{a_{k0}}$. The monoidal condition $\sum_{i=1}^{q+r} \bar{m}_{1j}^i = \sum_{i=1,i\neq q}^{q} \bar{t}_j^i + r + 1 \geq 0$ is satisfied if we choose r to be $r = \max_{j\in J_1} \sum_{i\in Q\setminus\{k\}} (-\bar{t}_j^i) - 1$. Let $\bar{m}_{2j}$ be the vector in M' that minimizes the expression (3.2.3) subject to the extra condition $m_j^i \geq 0, i \in \{k\} \cup \{q+1, \ldots, q+r\}$. The coefficient $\gamma_j^k(\bar{m}_{2j})$ cannot be greater than the coefficient obtained by (3.1.3) if we require $m_j^k \geq 0$ i.e.

$$\gamma_j^k(\bar{m}_{2j}) \quad \leq \min_{\substack{m_j \in M \\ m_j^k \geq 0}} \max_{i\in Q} \left\{ \frac{a_{ij}+(a_{i0}-b_i)m_j^i}{a_{i0}} \right\}.$$

Therefore $\tilde{\beta}_j^k \geq \min\{\gamma_j^k(\bar{m}_{1j}), \gamma_j^k(\bar{m}_{2j})\}$ for $j \in J_1$. The validity of $\tilde{\beta}_j^k$ follows from the validity of $\gamma_j^k(m_j), \forall m_j \in M'$.

□

Theorem 3.2.1 can also be proved using Remark 3.1 of [11] that states that Proposition 3.1.1 remains valid if we use monoids having the more general form

$$M(\mu) = \left\{ m \in \mathbb{Z}^q : \sum_{i\in Q} \mu_i m^i \geq 0 \right\},$$

where $\mu_i > 0, i \in Q$. It can be shown that the cut 3.2.1 can be obtained by applying the general monoidal cut strengthening on (3.1.1) using the monoid $M(\bar{\mu})$ where $\bar{\mu}$ is

$$\bar{\mu}_i := \begin{cases} 1 & i \in Q \setminus \{k\} \\ r+1 & i = k. \end{cases}$$

and r is defined as before.

We call $\tilde{\beta}^k x \geq 1$ the *k-th Lopsided cut* associated with the disjunction (3.1.1). The upshot of Theorem 3.2.1 is that given a q-term disjunction (3.1.1) where each term is unique, there exist q Lopsided cuts $\tilde{\beta}^k x \geq 1, k \in Q$ that are in general different from $\bar{\beta} x \geq 1$.

To compute the coefficients (3.2.1) we need to determine

$$\min_{\substack{m_j \in M \\ m_j^k \geq 0}} \max_{i\in Q} \left\{ \frac{a_{ij} + (a_{i0} - b_i)m_j^i}{a_{i0}} \right\}. \tag{3.2.4}$$

We can solve (3.2.4) using Algorithm 1 of [11] with the expression $\lambda_r(\alpha_r + m_r) = \max_{i\in Q} \lambda_i(\alpha_i + m_i)$ replaced by

$$\lambda_r(\alpha_r + m_r) = \begin{cases} \max_{i\in Q} \lambda_i(\alpha_i + m_i) & \text{if } m_k \geq 1 \\ \max_{\substack{i\in Q \\ i\neq k}} \lambda_i(\alpha_i + m_i) & \text{o.w.} \end{cases} \tag{3.2.5}$$

The expression (3.2.5) guarantees that $m_k \geq 0$ since $r \in Q$ is the index of the element of m that is decremented by 1 unit in the current iteration of the algorithm. The proof of correctness of the modified algorithm remains essentially unchanged. The algorithm has complexity $O(|Q| \cdot \max_{i\in Q}\{\bar{m}^i\})$ where $\bar{m}$ is the vector that minimizes (3.2.4) .

Corollary 3.2.2. *If $\tilde{\beta}^k_j < \bar{\beta}_j$ then $\tilde{\beta}^k_j = \frac{a_{kj}+a_{k0}-b_k}{a_{k0}}$ for $j \in J_1$.*

Proof. Notice that

$$\tilde{\beta}^k_j < \bar{\beta}_j = \min_{m_j\in M} \max_{i\in Q} \left\{ \frac{a_{ij} + (a_{i0} - b_i)m^i_j}{a_{i0}} \right\} \leq \min_{\substack{m_j\in M \\ m^k_j \geq 0}} \max_{i\in Q} \left\{ \frac{a_{ij} + (a_{i0} - b_i)m^i_j}{a_{i0}} \right\}. \tag{3.2.6}$$

From (3.2.1) and (3.2.6) we have $\tilde{\beta}^k_j = \frac{a_{kj}+a_{k0}-b_k}{a_{k0}}$. □

The next Corollary gives a weaker version of the lopsided cuts that does not require optimizing over a monoid.

Corollary 3.2.3. *For each $k \in Q$, the cut $\delta^k x \geq 1$, with*

$$\delta^k_j := \begin{cases} \min\left\{\frac{a_{kj}+a_{k0}-b_k}{a_{k0}}, \max_{i\in Q}\left\{\frac{a_{ij}}{a_{i0}}\right\}\right\} & j \in J_1 \\ \beta_j & j \in J \setminus J_1 \end{cases} \tag{3.2.7}$$

is valid.

Proof. Note that

$$\min_{\substack{m_j\in M \\ m^k_j \geq 0}} \max_{i\in Q} \left\{ \frac{a_{ij} + (a_{i0} - b_i)m^i_j}{a_{i0}} \right\} \leq \max_{i\in Q} \left\{ \frac{a_{ij}}{a_{i0}} \right\}$$

for $j \in J_1$, therefore $\tilde{\beta}^k_j \leq \delta^k_j$ and thus $\delta^k x \geq 1$ is valid. □

3.2.1 Simple split disjunction

Now consider a Mixed Integer Program, and let

$$\begin{array}{l} y = a_0 - \sum_{j\in J} a_j x_j \\ x_j \geq 0, j \in J \\ x_j \in \mathbb{Z}, j \in J_1 \subseteq J \end{array} \tag{3.2.8}$$

be a row of the simplex tableau associated with a basic solution to its linear relaxation, where $y \in \{0, 1\}$ and $0 < a_0 < 1$. The Gomory Mixed Integer (GMI) cut from (3.2.8) can be derived as a disjunctive cut from $(y \leq 0) \vee (y \geq 1)$, or

$$\left(\sum_{j\in J} a_j x_j \geq a_0 \right) \vee \left(\sum_{j\in J} (-a_j) x_j \geq 1 - a_0 \right) \tag{3.2.9}$$

as $\alpha x \geq 1$, with

$$\alpha_j := \max\left\{ \frac{a_j}{a_0}, \frac{-a_j}{1 - a_0} \right\}, j \in J, \tag{3.2.10}$$

which can be strengthened to $\bar{\alpha}x \geq 1$ by using the integrality of $x_j, j \in J_1$, with

$$\bar{\alpha}_j := \begin{cases} \min\left\{\frac{a_j-\lfloor a_j\rfloor}{a_0}, \frac{-a_j+\lceil a_j\rceil}{1-a_0}\right\} & j \in J_1 \\ \alpha_j & j \in J \setminus J_1. \end{cases} \tag{3.2.11}$$

As (3.2.9) and (3.2.10) are a special case of (3.1.1) and (3.1.2), the coefficients $\bar{\alpha}_j$ of (3.2.11) can be obtained by monoidal cut strengthening. Indeed, as in this case $b_1 = a_0 - 1, b_2 = -a_0$, we have $a_0 - b_1 = 1, 1 - a_0 - b_2 = 1$, and (3.2.11) becomes (3.2.12)

$$\bar{\beta}_j := \begin{cases} \min_{(m_j^1, m_j^2)\in M} \max\left\{\frac{a_j+m_j^1}{a_0}, \frac{-a_j+m_j^2}{1-a_0}\right\} & j \in J_1 \\ \alpha_j & j \in J \setminus J_1 \end{cases} \tag{3.2.12}$$

which is a special case of (3.1.3). It is not hard to see that the minimum in the expression for $\bar{\beta}_j, j \in J_1$, is attained for the smaller of $\frac{a_j-\lfloor a_j\rfloor}{a_0}$ and $\frac{-a_j+\lceil a_j\rceil}{1-a_0}$.

Theorem 3.2.4. *$\alpha^+x \geq 1$ and $\alpha^-x \geq 1$ are valid cuts, with*

$$\alpha_j^+ := \begin{cases} \frac{-a_j+1}{1-a_0} & j \in J_1^+ := \{j \in J_1 : a_j > 1\} \\ \min\left\{\frac{a_j-\lfloor a_j\rfloor}{a_0}, \frac{-a_j+\lceil a_j\rceil}{1-a_0}\right\} & j \in J_1^> := \{j \in J_1 : a_0 - 1 \leq a_j \leq 1\} \\ \max\left\{\frac{a_j}{a_0}, \frac{-a_j}{1-a_0}\right\} & j \in (J \setminus J_1) \cup \{j \in J_1 : a_j < a_0 - 1\} \end{cases} \tag{3.2.13}$$

which we call Right Lopsided cut *and*

$$\alpha_j^- := \begin{cases} \frac{a_j+1}{a_0} & j \in J_1^- := \{j \in J_1 : a_j < -1\} \\ \min\left\{\frac{a_j-\lfloor a_j\rfloor}{a_0}, \frac{-a_j+\lceil a_j\rceil}{1-a_0}\right\} & j \in J_1^< := \{j \in J_1 : -1 \leq a_j \leq a_0\} \\ \max\left\{\frac{a_j}{a_0}, \frac{-a_j}{1-a_0}\right\} & j \in (J \setminus J_1) \cup \{j \in J_1 : a_j > a_0\} \end{cases} \tag{3.2.14}$$

which we call Left Lopsided cut*.*

Proof. We give a proof of the validity of the cut $\alpha^+x \geq 1$. The proof of validity of $\alpha^-x \geq 1$ is analogous. Applying Theorem 3.2.1 to (3.2.9) with $k = 2$ we get $\tilde{\beta}_j^2 = \alpha_j^+$ for $j \in J \setminus J_1$ and $\tilde{\beta}_j^2 = \min\{A, B\}$ for $j \in J_1$ where

$$\begin{aligned} A &= \frac{-a_j+1}{1-a_0} \\ B &= \min_{\substack{m_j \in M \\ m_j^2 \geq 0}} \max\left\{\frac{a_j+m_j^1}{a_0}, \frac{-a_j+m_j^2}{1-a_0}\right\} = \min_{\substack{m_j^2 \in \mathbb{Z} \\ m_j^2 \geq 0}} \max\left\{\frac{a_j-m_j^2}{a_0}, \frac{-a_j+m_j^2}{1-a_0}\right\} \end{aligned}$$

- If $a_j > 1$ then $A < 0$, $B \geq 0$ and $\tilde{\beta}_j^2 = A = \alpha_j^+$;
- if $a_0 \leq a_j \leq 1$ then $\tilde{\beta}_j^2 = A = B = \frac{-a_j+\lceil a_j\rceil}{1-a_0} = \alpha_j^+$;
- if $0 \leq a_j < a_0$ then $A > 1, B \leq 1$ and $\tilde{\beta}_j^2 = B = \frac{a_j-\lfloor a_j\rfloor}{a_0} = \alpha_j^+$;
- if $a_j < 0$ then $A > B = \frac{-a_j}{1-a_0}$ therefore $\tilde{\beta}_j^2 = \max\left\{\frac{a_j}{a_0}, \frac{-a_j}{1-a_0}\right\} = \alpha_j^+$.

$\square$

Corollary 3.2.5. *If $a_j \geq a_0 - 1, j \in J_1$ and $J_1^+ \neq \emptyset$, the cut $\alpha^+ x \geq 1$ strictly dominates the GMI cut. If $a_j \leq a_0, j \in J_1$ and $J_1^- \neq \emptyset$, the cut $\alpha^- x \geq 1$ strictly dominates the GMI cut.*

Example Consider the following row of a simplex tableau

$$y = 0.2 - 1.5x_1 + 0.3x_2 + 0.4x_3 + 0.6x_4 - 4.3x_5 - 0.1x_6. \tag{3.2.15}$$

subject to the additional constraints $y \in \{0, 1\}$ and $x_j \in \mathbb{Z}, j \in J_1 = \{1, \dots, 6\}$. The point $\bar{y} = 0; \bar{x}_1 = \bar{x}_2 = \bar{x}_3 = \bar{x}_4 = 1; \bar{x}_5 = \bar{x}_6 = 0$ is a feasible integer solution. The GMI cut obtained from this row is

$$0.625x_1 + 0.375x_2 + 0.5x_3 + 0.75x_4 + 0.875x_5 + 0.5x_6 \geq 1. \tag{3.2.16}$$

Applying Theorem 3.2.4 we obtain a cut $\alpha^+ x \geq 1$ that strictly dominates (3.2.16). For (3.2.15) the index sets $J_1^+, J_1^>$ are respectively $J_1^+ = \{1, 5\}$ and $J_1^> = \{2, 3, 4, 6\}$. Therefore we have $\alpha_1^+ = \frac{-a_j+1}{1-a_0} = \frac{-0.5}{0.8} = -0.625$ and $\alpha_5^+ = \frac{-a_j+1}{1-a_0} = \frac{-3.3}{0.8} = -4.125$ and the remaining coefficients α_j^+ for $j \in J_1^+$ are the same as in (3.2.16). The cut $\alpha^+ x \geq 1$ is then

$$-0.625x_1 + 0.375x_2 + 0.5x_3 + 0.75x_4 - 4.125x_5 + 0.5x_6 \geq 1. \tag{3.2.17}$$

Note that (3.2.17) is tight for the solution $(\bar{y}, \bar{x}_1, \dots, \bar{x}_6)$ while the GMI cut (3.2.16) has a slack of 1.25.

In Figure 1 we illustrate graphically the value of the cut coefficients for the GMI cut and the two lopsided cuts given in (3.2.11), (3.2.13) and (3.2.14) for an arbitrary tableau row with $a_0 = 0.3$. The cut coefficients are shown on the vertical axis as a function of the tableau row coefficient values $(-a_j)$ shown on the horizontal axis.

As the GMI cut can be derived in different ways [46, 11], the same holds also for the two cuts in Theorem (3.2.4). Indeed, it can be shown that the cut $\alpha^+ x \geq 1$ can be obtained by dividing the source row in (3.2.8) by a large number and then deriving a GMI cut, and a similar procedure yields the cut $\alpha^- x \geq 1$. However, in the case of a more general disjunction we do not know of any alternative method for deriving the cut of Theorem 3.2.1 or Corollary 3.2.3.

In Table 3.1 we compare the GMI relaxation (denoted by G) and the relaxation where both GMI cuts and lopsided cuts are generated (denoted by G+L). In both cases the cuts were applied for only 1 round. To measure the strength of the relaxations we consider the duality gap closed which is computed as

$$\text{Gap} = 100 \frac{C_{opt} - LP_{opt}}{IP_{opt} - LP_{opt}} \tag{3.2.18}$$

where IP_{opt}, LP_{opt} and C_{opt} are respectively the value of the optimal integer solution, the value of the linear relaxation and the value of the relaxation currently analyzed. The columns $G_{\#}$ and $G + L_{\#}$ indicate the number of cuts generated, $G_{\%}$ and $G + L_{\%}$ indicate the gap computed according to formula (3.2.18). The column imp shows the difference between $G_{\%}$

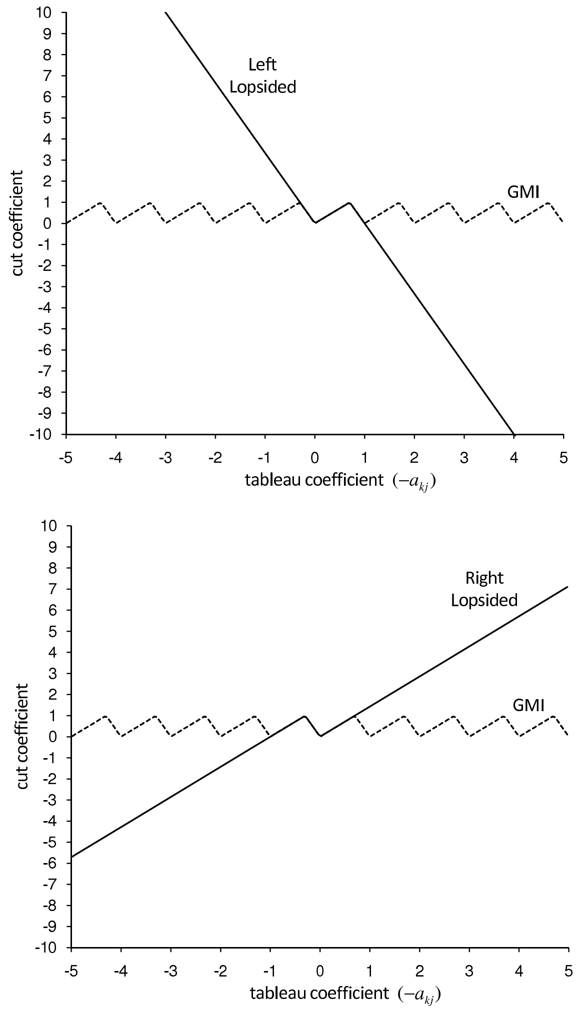


Figure 3.1: Coefficient values for the GMI and the lopsided cuts for the case $a_0 = 0.3$.

and $G+L_{\%}$, and the column imp$_{\%}$ indicates the percentage improvement produced by adding the lopsided cuts on top of the GMI relaxation. For our experiments we considered the entire collection of MIPLIB3_C_V2 [42] that contain some 0-1 variables. In Table 3.1 we only show those instances for which the percentage improvement given by the lopsided cuts exceeds 1%.

Although in many problems the lopsided cuts produce no improvement, in some cases the impact is substantial. As we did in Chapter 2, for the computational results presented here we also followed the approach used in [14]. We performed statistical tests to compare the relaxation strength produced by Gomory cuts and Gomory+Lopsided cuts. Experiments show that the improvement given by the Lopsided cuts after one round of cuts at depth 0 of the branch-and-bound tree is statistically significant with a 95% confidence level. At depths 8, 16 and 24 the improvement is not statistically significant. This shows that the marginal improvement produced by the Lopsided cuts on the root relaxation is lost at further branch-and-bound depths. For the computations we used a single core of a Linux 2.6.32-32 computer with an Intel Core 2 Duo 2.66GHz with 4GB of memory.

Table 3.1: Computational results with lopsided cuts on the simple split disjunction

Instance	$G_{\#}$	$G_{\%}$	$G+L_{\#}$	$G+L_{\%}$	$\text{imp}_{\%}$
aflow40b	27	10.60	79	10.76	1.51
air04	156	8.44	582	8.53	1.07
blend2	5	15.98	11	16.17	1.19
dcmulti	45	47.69	53	48.36	1.40
gesa2	52	25.10	66	26.25	4.58
harp2	27	22.05	72	22.56	2.31
l152lav	9	12.80	119	15.18	18.59
mas76	9	6.36	25	6.53	2.67
mkc	126	1.83	330	4.25	132.24
modglob	16	13.32	29	14.05	5.48
vpm2	21	10.79	36	11.25	4.26

Connection to K-cuts by Cornuejols et al.

Cornuejols et al. in [25] showed that valid cuts for (3.2.8) that are different from the GMI cut can be obtained by applying the Gomory Mixed Integer procedure to positive integer multiples of the source row. We will now give a derivation of Lopsided cuts for the Simple Split Disjunction which uses a similar approach except that instead of multiply the source row by a positive integer we divide it by a positive integer. This variation has not been considered in the literature. When the source row coefficients are divided by a given integer constant M, the coefficient associated to the basic variable becomes fractional and and therefore the Gomory Mixed Integer procedure will associate a non-zero cut coefficient α_y to it. This does not occur if we consider a multiple of the source row.

Let $f_j = a_j - \lfloor a_j \rfloor, j \in J_1$ and $f_0 = a_0 - \lfloor a_0 \rfloor = a_0$. The GMI cut from (3.2.8) is

$$\sum_{f_j \le f_0, j \in J_1} \frac{f_j}{f_0} x_j + \sum_{f_j > f_0, j \in J_1} \frac{1 - f_j}{1 - f_0} x_j + \sum_{a_j \le 0, j \in J \setminus J_1} \frac{-a_j}{1 - f_0} x_j + \sum_{a_j > 0, j \in J \setminus J_1} \frac{a_j}{f_0} x_j \ge 1 \quad (3.2.19)$$

Now, suppose we divide (3.2.8) by a number $M \in \mathbb{Z}$, with

$$M > \max_{j \in J} \{|a_j|\} \quad (3.2.20)$$

Thus we have the scaled row

$$\frac{1}{M} y + \sum_{j \in J_1} \frac{a_j}{M} x_j + \sum_{j \in J \setminus J_1} \frac{a_j}{M} x_j = \frac{a_0}{M} \quad (3.2.21)$$

Let $\bar{f}_j = \frac{a_j}{M} - \lfloor \frac{a_j}{M} \rfloor, j \in J_1$ and $\bar{f}_0 = \frac{a_0}{M} - \lfloor \frac{a_0}{M} \rfloor$. Note that by (3.2.20)

$$\bar{f}_j := \begin{cases} \frac{a_j}{M} & a_j \ge 0 \\ \frac{M + a_j}{M} & a_j < 0 \end{cases} \quad (3.2.22)$$

for $j \in J_1$. The GMI cut from (3.2.21) is

$$\frac{M-1}{M-a_0}y + \sum_{\bar{f}_j \le \bar{f}_0, j \in J_1} \frac{\bar{f}_j}{\bar{f}_0}x_j + \sum_{\bar{f}_j > \bar{f}_0, j \in J_1} \frac{1-\bar{f}_j}{1-\bar{f}_0}x_j + \sum_{a_j \le 0, j \in J \setminus J_1} \frac{-a_j}{M(1-\bar{f}_0)}x_j + \sum_{a_j > 0, j \in J \setminus J_1} \frac{a_j}{M\bar{f}_0}x_j \ge 1 \quad (3.2.23)$$

Note that in our case $f_y > f_0$ since ($\frac{1}{M} > \frac{a_0}{M}$), therefore the coefficient for y in (3.2.23) is $\frac{1-f_y}{1-f_0} = \frac{1-\frac{1}{M}}{1-\frac{a_0}{M}} = \frac{M-1}{M-a_0}$. For y general integer we can have $b > 1$, and therefore we should also consider the case $\frac{1}{M} < \frac{a_0}{M}$ and the coefficient for y in (3.2.23) would be $\frac{1}{a_0}$. In this note we do not explore this more general case.
We now compare (3.2.23) with (3.2.19). Let's first rewrite (3.2.23) as

$$\frac{M-1}{M-a_0}y + \sum_{\bar{f}_j \le \bar{f}_0, a_j \ge 0, j \in J_1} \frac{\bar{f}_j}{\bar{f}_0}x_j + \sum_{\bar{f}_j \le \bar{f}_0, a_j < 0, j \in J_1} \frac{\bar{f}_j}{\bar{f}_0}x_j + \sum_{\bar{f}_j > \bar{f}_0, a_j \ge 0, j \in J_1} \frac{1-\bar{f}_j}{1-\bar{f}_0}x_j + \sum_{\bar{f}_j > \bar{f}_0, a_j < 0, j \in J_1} \frac{1-\bar{f}_j}{1-\bar{f}_0}x_j + \sum_{a_j \le 0, j \in J \setminus J_1} \frac{-a_j}{M(1-\bar{f}_0)}x_j + \sum_{a_j > 0, j \in J \setminus J_1} \frac{a_j}{M\bar{f}_0}x_j \ge 1$$

then we substitute the values for $\bar{f}_0, \bar{f}_j$

$$\frac{M-1}{M-a_0}y + \sum_{\bar{f}_j \le \bar{f}_0, a_j \ge 0, j \in J_1} \frac{a_j}{M}\frac{M}{a_0}x_j + \sum_{\bar{f}_j \le \bar{f}_0, a_j < 0, j \in J_1} \frac{M+a_j}{M}\frac{M}{a_0}x_j + \sum_{\bar{f}_j > \bar{f}_0, a_j \ge 0, j \in J_1} \frac{M-a_j}{M}\frac{M}{M-a_0}x_j + \sum_{\bar{f}_j > \bar{f}_0, a_j < 0, j \in J_1} \frac{-a_j}{M}\frac{M}{M-a_0}x_j + \sum_{a_j \le 0, j \in J \setminus J_1} \frac{-a_j}{M(1-\frac{a_0}{M})}x_j + \sum_{a_j > 0, j \in J \setminus J_1} \frac{a_j}{M\frac{a_0}{M}}x_j \ge 1$$

we simplify terms and remove the second summation since it contains no term (this follows since (3.2.20) holds)

$$\frac{M-1}{M-a_0}y + \sum_{\bar{f}_j \le \bar{f}_0, a_j \ge 0, j \in J_1} \frac{a_j}{a_0}x_j + \sum_{\bar{f}_j > \bar{f}_0, a_j \ge 0, j \in J_1} \frac{M-a_j}{M-a_0}x_j + \sum_{\bar{f}_j > \bar{f}_0, a_j < 0, j \in J_1} \frac{-a_j}{M-a_0}x_j + \sum_{a_j \le 0, j \in J \setminus J_1} \frac{-a_j}{M-b}x_j + \sum_{a_j > 0, j \in J \setminus J_1} \frac{a_j}{a_0}x_j \ge 1$$

and now we eliminate the variable y using (3.2.8)

$$\sum_{\bar{f}_j \le \bar{f}_0, a_j \ge 0, j \in J_1} \left(\frac{a_j}{a_0} - \frac{M-1}{M-a_0} a_j \right) x_j + \sum_{\bar{f}_j > \bar{f}_0, a_j \ge 0, j \in J_1} \left(\frac{M-a_j}{M-a_0} - \frac{M-1}{M-a_0} a_j \right) x_j +$$
$$\sum_{\bar{f}_j > \bar{f}_0, a_j < 0, j \in J_1} \left(\frac{-a_j}{M-a_0} - \frac{M-1}{M-a_0} a_j \right) x_j + \sum_{a_j \le 0, j \in J \setminus J_1} \left(\frac{-a_j}{M-a_0} - \frac{M-1}{M-a_0} a_j \right) x_j +$$
$$\sum_{a_j > 0, j \in J \setminus J_1} \left(\frac{a_j}{a_0} - \frac{M-1}{M-a_0} a_j \right) x_j \ge 1 - \frac{M-1}{M-a_0} a_0$$

Simplifying we get

$$\sum_{\bar{f}_j \le \bar{f}_0, a_j \ge 0, j \in J_1} \frac{a_j(1-a_0)M)}{a_0(M-a_0)} x_j + \sum_{\bar{f}_j > \bar{f}_0, a_j \ge 0, j \in J_1} \frac{(1-a_j)M}{M-a_0} x_j +$$
$$\sum_{\bar{f}_j > \bar{f}_0, a_j < 0, j \in J_1} \frac{-a_j M}{M-a_0} x_j + \sum_{a_j \le 0, j \in J \setminus J_1} \frac{-a_j M}{M-a_0} x_j +$$
$$\sum_{a_j > 0, j \in J \setminus J_1} \frac{a_j(1-a_0)M}{b(M-a_0)} x_j \ge \frac{(1-a_0)M}{M-a_0}$$

We scale the cut to bring the right hand side to 1 by dividing the inequality by $\frac{M-a_0}{(1-a_0)M}$

$$\sum_{\bar{f}_j \le \bar{f}_0, a_j \ge 0, j \in J_1} \frac{a_j}{a_0} x_j + \sum_{\bar{f}_j > \bar{f}_0, a_j \ge 0, j \in J_1} \frac{1-a_j}{1-a_0} x_j + \sum_{\bar{f}_j > \bar{f}_0, a_j < 0, j \in J_1} \frac{-a_j}{1-a_0} x_j +$$
$$\sum_{a_j \le 0, j \in J \setminus J_1} \frac{-a_j}{1-a_0} x_j + \sum_{a_j > 0, j \in J \setminus J_1} \frac{a_j}{a_0} x_j \ge 1$$

The conditions on the summations can be simplified to get the cut

$$\sum_{0 \le a_j \le a_0, j \in J_1} \frac{a_j}{a_0} x_j + \sum_{a_j > a_0, j \in J_1} \frac{1-a_j}{1-a_0} x_j + \sum_{a_j < 0, j \in J_1} \frac{-a_j}{1-a_0} x_j +$$
$$\sum_{a_j \le 0, j \in J \setminus J_1} \frac{-a_j}{1-a_0} x_j + \sum_{a_j > 0, j \in J \setminus J_1} \frac{a_j}{a_0} x_j \ge 1$$

which is equivalent to the Right Lopsided cut $\alpha^+ x \ge 1$ in (3.2.13). The Left Lopsided cut can be obtained similarly by applying the same procedure to the complement of y.

3.2.2 Multiple term disjunctions

We now consider intersection cuts derived from 2 rows of the simplex tableau associated with a basic solution of a Mixed Integer Linear Program. Let

$$\begin{array}{l} y_1 = a_{10} - \sum_{j\in J} a_{1j}x_j \\ y_2 = a_{20} - \sum_{j\in J} a_{2j}x_j \\ x_j \geq 0, j \in J \\ x_j \in \mathbb{Z}, j \in J_1 \subseteq J \end{array} \tag{3.2.24}$$

be two rows of the simplex tableau associated with a basic solution to a linear relaxation of a Mixed Integer Program, where $x_j, j \in J$ are nonbasic variables, $y_h \in \{0,1\}$ are basic variables with $0 < a_{h0} < 1, h \in \{1,2\}$.

Given a closed convex set S that contains the point $(a_{10}, a_{20}, 0, \ldots, 0)$ in its interior but no feasible integer point, we can generate the intersection cut [5, 2] $\alpha^{ic}x \geq 1$ with coefficients

$$\alpha_j^{ic} \quad := \min_{\mu>0} \left\{ \tfrac{1}{\mu} : (a_{10}, a_{20}) + \mu(-a_{1j}, -a_{2j}) \in S \right\} \tag{3.2.25}$$

for $j \in J$. When $J_1 \neq \emptyset$ the standard strengthening used, for instance, in [31], yields a stronger cut $\alpha^{str}x \geq 1$ where

$$\alpha_j^{str} := \min_{\substack{\mu>0 \\ p_{1j}, p_{2j} \in \mathbb{Z}}} \left\{ \frac{1}{\mu} : (a_{10}, a_{20}) + \mu(-a_{1j} + p_{1j}, -a_{2j} + p_{2j}) \in S \right\}. \tag{3.2.26}$$

If S is polyhedral, i.e. its representation in terms of the basic variables is

$$S = \{(y_1, y_2) \in \mathbb{R}^2 : d_{1i}y_1 + d_{2i}y_2 \leq e_i, i \in Q\}, \tag{3.2.27}$$

then the cut $\alpha^{ic}x \geq 1$ is a disjunctive cut derived from the disjunction $\vee_{i\in Q}(d_{1i}y_1 + d_{2i}y_2 \geq e_i)$, or

$$\bigvee_{i\in Q} \left(\sum_{j\in J} (-d_{1i}a_{1j} - d_{2i}a_{2j})x_j \geq e_i - d_{1i}a_{10} - d_{2i}a_{20} \right). \tag{3.2.28}$$

Furthermore, if lower bounds are known for the left hand sides of the terms of (3.2.28), then monoidal strengthening [11] can be used to obtain another strengthened cut $\bar{\beta}x \geq 1$, generally different from (3.2.26).

We now consider the lopsided cuts derived from Theorem 3.2.1 applied to the disjunction (3.2.28). We will show that the lopsided cuts are different from the cut $\alpha^{str}x \geq 1$ and the monoidal strengthened cut $\bar{\beta}x \geq 1$. If certain conditions hold, the lopsided cuts are stronger than both. In the rest of the section we fix S to be the lattice free triangle with vertices $(0,2); (2,0); (0,0)$, i.e.

$$S = \left\{ (y_1, y_2) \in \mathbb{R}^2 : y_1 \geq 0; y_2 \geq 0; y_1 + y_2 \leq 2 \right\}. \tag{3.2.29}$$

The point (a_{10}, a_{20}) is in the interior of S and S does not contain in its interior any feasible integer point. The intersection cut $\alpha^{ic}x \geq 1$ from S can be derived from the disjunction

$$\left(\sum_{j\in J} a_{1j}x_j \geq a_{10}\right) \vee \left(\sum_{j\in J} a_{2j}x_j \geq a_{20}\right) \vee \left(\sum_{j\in J}(-a_{1j}-a_{2j})x_j \geq 2-a_{10}-a_{20}\right). \quad (3.2.30)$$

The cut $\alpha^{ic}x \geq 1$ has coefficients

$$\alpha_j^{ic} = \max\left\{\frac{a_{1j}}{a_{10}}, \frac{a_{2j}}{a_{20}}, \frac{-a_{1j}-a_{2j}}{2-a_{10}-a_{20}}\right\}, j \in J$$

Similarly for $\alpha^{str}x \geq 1$ we have

$$\alpha_j^{str} := \begin{cases} \min_{p_{1j},p_{2j}\in\mathbb{Z}} \max\left\{\frac{a_{1j}+p_{1j}}{a_{10}}, \frac{a_{2j}+p_{2j}}{a_{20}}, \frac{-a_{1j}-a_{2j}-p_{1j}-p_{2j}}{2-a_{10}-a_{20}}\right\} & j \in J_1 \\ \alpha_j^{ic} & j \in J \setminus J_1 \end{cases}$$

The values $b_1 = a_{10} - 1; = b_2 = a_{20} - 1; b_3 = -a_{10} - a_{20}$ are valid lower bounds for the left hand sides of the three terms of (3.2.30). Therefore the monoidal strengthened cut $\bar{\beta}_j x \geq 1$ has coefficients

$$\bar{\beta}_j := \begin{cases} \min_{m_j\in M} \max\left\{\frac{a_{1j}+m_j^1}{a_{10}}, \frac{a_{2j}+m_j^2}{a_{20}}, \frac{-a_{1j}-a_{2j}+2m_j^3}{2-a_{10}-a_{20}}\right\} & j \in J_1 \\ \alpha_j^{ic} & j \in J \setminus J_1 \end{cases}$$

where $M = \{m \in \mathbb{Z}^3 : \sum_{i=1}^3 m^i \geq 0\}$. Theorem 3.2.1 yields the cuts $\tilde{\beta}^k x \geq 1, k \in \{1,2,3\}$ where

$$\tilde{\beta}_j^k := \begin{cases} \min\left\{\frac{a_{1j}+1}{a_{10}}, \min_{\substack{m_j\in M\\ m_1\geq 0}} \max\left\{\frac{a_{1j}+m_j^1}{a_{10}}, \frac{a_{2j}+m_j^2}{a_{20}}, \frac{-a_{1j}-a_{2j}+2m_j^3}{2-a_{10}-a_{20}}\right\}\right\} & k = 1, j \in J_1 \\ \min\left\{\frac{a_{2j}+1}{a_{20}}, \min_{\substack{m_j\in M\\ m_2\geq 0}} \max\left\{\frac{a_{1j}+m_j^1}{a_{10}}, \frac{a_{2j}+m_j^2}{a_{20}}, \frac{-a_{1j}-a_{2j}+2m_j^3}{2-a_{10}-a_{20}}\right\}\right\} & k = 2, j \in J_1 \\ \min\left\{\frac{-a_{1j}-a_{2j}+1}{2-a_{10}-a_{20}}, \min_{\substack{m_j\in M\\ m_3\geq 0}} \max\left\{\frac{a_{1j}+m_j^1}{a_{10}}, \frac{a_{2j}+m_j^2}{a_{20}}, \frac{-a_{1j}-a_{2j}+2m_j^3}{2-a_{10}-a_{20}}\right\}\right\} & k = 3, j \in J_1 \\ \alpha_j^{ic} & j \in J \setminus J_1 \end{cases}$$

The next three Propositions give sufficient conditions for a coefficient $\tilde{\beta}_j^k, k \in \{1,2,3\}$ to be strictly less than $\bar{\beta}_j, j \in J_1$.

Proposition 3.2.6. *If $a_{1j} + a_{2j} \geq 2$ for some $j \in J_1$ then $\tilde{\beta}_j^3 \leq 0 < \bar{\beta}_j$ for $j \in J_1$.*

Proof. Since $a_{1j} + a_{2j} \geq 2$ we have that $\tilde{\beta}_j^3 \leq \frac{-a_{1j}-a_{2j}+2}{2-a_{10}-a_{20}} \leq 0$. Now assume by contradiction that $\bar{\beta}_j \leq 0$. This implies that the following conditions hold

$$\left\{ \begin{array}{l} a_{1j} + m_1 \leq 0 \\ a_{2j} + m_2 \leq 0 \\ -a_{1j} - a_{2j} + 2m_3 \leq 0 \\ m_1 + m_2 + m_3 \geq 0 \\ a_{1j} + a_{2j} \geq 2 \\ m_i \in \mathbb{Z}, i \in \{1,2,3\} \end{array} \right\} \tag{3.2.31}$$

The set defined as $\{z = (a_{1j}, a_{2j}, m_1, m_2, m_3) \in \mathbb{R}^5 : Az \leq b\}$ where A and b are

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ -1 & -1 & 0 & 0 & 2 \\ 0 & 0 & -1 & -1 & -1 \\ -1 & -1 & 0 & 0 & 0 \end{pmatrix} \quad b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -2 \end{pmatrix} \tag{3.2.32}$$

is a relaxation of (3.2.31). Let $c = (2, 2, 1, 2, 1)$. Since $cA = 0$, $c \geq 0$, $cb = -2 < 0$ by Farkas Lemma (3.2.32) is infeasible. Thus (3.2.31) is also infeasible and we reached a contradiction, therefore $\bar{\beta}_j > 0$. □

Proposition 3.2.7. *If* $(2 + a_{20})a_{1j} - a_{10}a_{2j} < -2 - a_{10} - a_{20}$ *then* $\tilde{\beta}^1_j < \bar{\beta}_j$ *for* $j \in J_1$.

Proof. Assume by contradiction that $\tilde{\beta}^1_k \geq \bar{\beta}_j$. Then the following conditions must be satisfied

$$\left\{ \begin{array}{ll} \frac{a_{1j}+m_1}{a_{10}} & \leq \frac{a_{1j}+1}{a_{10}} \\ \frac{a_{2j}+m_2}{a_{20}} & \leq \frac{a_{1j}+1}{a_{10}} \\ \frac{-a_{1j}-a_{2j}+2m_3}{2-a_{10}-a_{20}} & \leq \frac{a_{1j}+1}{a_{10}} \\ m_1 + m_2 + m_3 & \geq 0 \\ (2 + a_{20})a_{1j} - a_{10}a_{2j} & < -2 - a_{10} - a_{20} \\ m_i \in \mathbb{Z}, i \in \{1,2,3\} & \end{array} \right\} \tag{3.2.33}$$

since $\tilde{\beta}^1_j \leq \frac{a_{1j}+1}{a_{10}}$. The set defined as $\{z = (a_{1j}, a_{2j}, m_1, m_2, m_3) \in \mathbb{R}^5 : Az \leq b\}$ where A and b are

$$A = \begin{pmatrix} 0 & 0 & \frac{1}{a_{10}} & 0 & 0 \\ -\frac{1}{a_{10}} & \frac{1}{a_{20}} & 0 & \frac{1}{a_{20}} & 0 \\ \frac{a_{20}-2}{a_{10}(2-a_{10}-a_{20})} & -\frac{1}{2-a_{10}-a_{20}} & 0 & 0 & \frac{2}{2-a_{10}-a_{20}} \\ 0 & 0 & -1 & -1 & -1 \\ (2+a_{20}) & -a_{10} & 0 & 0 & 0 \end{pmatrix} \quad b = \begin{pmatrix} \frac{1}{a_{10}} \\ \frac{1}{a_{10}} \\ \frac{1}{a_{10}} \\ 0 \\ -2 - a_{10} - a_{20} - \epsilon \end{pmatrix} \tag{3.2.34}$$

is a relaxation of (3.2.33) for any $\epsilon > 0$. Let $c = (2a_{10}^2(2-a_{20}), 2a_{10}a_{20}(2-a_{20}), a_{10}(2-a_{20})(2-a_{10}-a_{20}), 2a_{10}(2-a_{20}), 2+a_{20})$. Since $cA = 0$, $c \geq 0$ $cb = -\epsilon(2+a_{20}) < 0$, by Farkas Lemma (3.2.34) is infeasible. Thus, (3.2.33) is infeasible and we reached a contradiction, therefore $\tilde{\beta}^1_j < \bar{\beta}_j$. □

Proposition 3.2.8. *If* $-a_{20}a_{1j} + (2 + a_{10})a_{2j} < -2 - a_{10} - a_{20}$ *then* $\tilde{\beta}^2_j < \bar{\beta}_j$ *for* $j \in J_1$.

Proof. Analogous to the proof of Lemma 3.2.7. $\square$

Similarly, we can derive some simple sufficient conditions for $\tilde{\beta}^k_j$ to strictly dominate the coefficient α^{str}_j that follow immediately from the definition of the lopsided cuts. Noting that $\alpha^{str}_j \geq 0, j \in J_1$ we have the following

Proposition 3.2.9. *If* $(k = 1 \wedge a_{1j} < -1)$ *or* $(k = 2 \wedge a_{2j} < -1)$ *or* $(k = 3 \wedge -a_{1j} - a_{2j} < -2)$ *then* $\tilde{\beta}^k_j < 0 \leq \alpha^{str}_j$.

In the next example we show that applying monoidal cut strengthening to a disjunction with redundant terms yields a stronger cut than the cut derived using the standard strengthening or the monoidal strengthening.

Example Consider the 2-row relaxation

$$P = \Big\{ \begin{array}{l} (y,x) \in \mathbb{R}^8 : \\ y_1 = \;\; 0.25 \;\; -0.15x_1 \;\; +0.6x_2 \;\; -0.4x_3 \;\; -1.2x_4 \;\; -2.9x_5 \;\; +0.8x_6 \\ y_2 = \;\; 0.5 \;\; +1.15x_1 \;\; -0.1x_2 \;\; -0.2x_3 \;\; -1.6x_4 \;\; +0.5x_5 \;\; -2.5x_6 \\ x_j \geq 0, j \in J = \{1,\ldots,6\} \\ x_j \in \mathbb{Z}, j \in J_1 = \{4,5,6\} \\ y_k \in \{0,1\}, k \in \{1,2\} \;\Big\}. \end{array} \tag{3.2.35}$$

The current basic solution to the relaxation (3.2.35) is $(\bar{y}_1, \bar{y}_2, \bar{x}) = (0.25, 0.5, \bar{0})$ and it violates the integrality conditions on y_1, y_2. We can derive an intersection cut $\alpha^{ic}x \geq 1$ from S defined in (3.2.29) since $(\bar{y}_1, \bar{y}_2)$ is in the interior of S. The coefficients α^{ic}_j are

$$\alpha^{ic}_j = \min_{\mu>0}\Big\{ \tfrac{1}{\mu} : \;\; \begin{array}{l} a_{10} + \mu(-a_{1j}) \geq 0; a_{20} + \mu(-a_{2j}) \geq 0; \\ a_{10} + a_{20} + \mu(a_{1j} + a_{2j}) \leq 2 \Big\} \end{array} \tag{3.2.36}$$

Applying (3.2.36) to the instance (3.2.35) we compute the coefficients

$$\begin{array}{ll}
\alpha^{ic}_1 & = \min_{\mu>0}\{\tfrac{1}{\mu} : 0.25 + \mu(-0.15) \geq 0; \;\; 0.5 + \mu(1.15) \geq 0; \;\; 0.75 + \mu(1) \leq 2\} \\
 & = \tfrac{1}{1.25} = 0.8 \\
\alpha^{ic}_2 & = \min_{\mu>0}\{\tfrac{1}{\mu} : 0.25 + \mu(0.6) \geq 0; \;\; 0.5 + \mu(-0.1) \geq 0; \;\; 0.75 + \mu(0.5) \leq 2\} \\
 & = \tfrac{1}{2.5} = 0.4 \\
\alpha^{ic}_3 & = \min_{\mu>0}\{\tfrac{1}{\mu} : 0.25 + \mu(-0.4) \geq 0; \;\; 0.5 + \mu(-0.2) \geq 0; \;\; 0.75 + \mu(-0.6) \leq 2\} \\
 & = \tfrac{1}{0.625} = 1.6 \\
\alpha^{ic}_4 & = \min_{\mu>0}\{\tfrac{1}{\mu} : 0.25 + \mu(-1.2) \geq 0; \;\; 0.5 + \mu(-1.6) \geq 0; \;\; 0.75 + \mu(-2.8) \leq 2\} \\
 & = \tfrac{1}{0.2083} = 4.8 \\
\alpha^{ic}_5 & = \min_{\mu>0}\{\tfrac{1}{\mu} : 0.25 + \mu(-2.9) \geq 0; \;\; 0.5 + \mu(0.5) \geq 0; \;\; 0.75 + \mu(-2.4) \leq 2\} \\
 & = \tfrac{1}{0.0862} = 11.6 \\
\alpha^{ic}_6 & = \min_{\mu>0}\{\tfrac{1}{\mu} : 0.25 + \mu(0.8) \geq 0; \;\; 0.5 + \mu(-2.5) \geq 0; \;\; 0.75 + \mu(-1.7) \leq 2\} \\
 & = \tfrac{1}{0.2} = 5
\end{array}$$

Therefore the intersection cut is

$$0.8x_1 + 0.4x_2 + 1.6x_3 + 4.8x_4 + 11.6x_5 + 5x_6 \geq 1. \tag{3.2.37}$$

By applying standard strengthening on the integer variables x_4, x_5, x_6 we get the coefficients

$$\begin{aligned}
\alpha_4^{str} &= \min_{\substack{\mu>0\\ p_{1j},p_{2j}\in\mathbb{Z}}} \{\tfrac{1}{\mu}: && 0.25 + \mu(-1.2 + p_1) \geq 0; 0.5 + \mu(-1.6 + p_2) \geq 0;\\
& && 0.75 + \mu(-2.8p_1 + p_2) \leq 2\}\\
&= \min_{\mu>0}\{\tfrac{1}{\mu}: && 0.25 + \mu(-1.2 + 1) \geq 0; 0.5 + \mu(-1.6 + 2) \geq 0;\\
& && 0.75 + \mu(-2.8 + 1 + 2) \leq 2\big\}\\
&= \tfrac{1}{1.25} = 0.8.\\
\alpha_5^{str} &= \min_{\substack{\mu>0\\ p_{1j},p_{2j}\in\mathbb{Z}}} \{\tfrac{1}{\mu}: && 0.25 + \mu(-2.9 + p_1) \geq 0; 0.5 + \mu(+0.5 + p_2) \geq 0;\\
& && 0.75 + \mu(-2.4 + p_1 + p_2) \leq 2\}\\
&= \min_{\mu>0}\{\tfrac{1}{\mu}: && 0.25 + \mu(-2.9 + 3) \geq 0; 0.5 + \mu(+0.5 + 0) \geq 0;\\
& && 0.75 + \mu(-2.4 + 3 + 0) \leq 2\}\\
&= \tfrac{1}{2.0833} = 0.48.\\
\alpha_6^{str} &= \min_{\substack{\mu>0\\ p_{1j},p_{2j}\in\mathbb{Z}}} \{\tfrac{1}{\mu}: && 0.25 + \mu(0.8 + p_1) \geq 0; 0.5 + \mu(-2.5 + p_2) \geq 0;\\
& && 0.75 + \mu(-1.7 + p_1 + p_2) \leq 2\}\\
&= \min_{\mu>0}\{\tfrac{1}{\mu}: && 0.25 + \mu(0.8 - 1) \geq 0; 0.5 + \mu(-2.5 + 3) \geq 0;\\
& && 0.75 + \mu(-1.7 - 1 + 3) \leq 2\}\\
&= \tfrac{1}{1.25} = 0.8.
\end{aligned}$$

Therefore the standard strengthened cut is

$$0.8x_1 + 0.4x_2 + 1.6x_3 + 0.8x_4 + 0.48x_5 + 0.8x_6 \geq 1. \tag{3.2.38}$$

Monoidal strengthening applied to the disjunction (3.2.30) and the instance (3.2.35) yields the coefficients

$$\begin{aligned}
\bar{\beta}_4^3 &= \min_{m_j\in M} \max\left\{\tfrac{1.2+m_j^1}{0.25}, \tfrac{1.6+m_j^2}{0.5}, \tfrac{-1.2-1.6+2m_j^3}{2-0.25-0.5}\right\}\\
&= \max\left\{\tfrac{1.2-1}{0.25}, \tfrac{1.6-1}{0.5}, \tfrac{-1.2-1.6+2\times2}{2-0.25-0.5}\right\} = 1.2\\
\bar{\beta}_5^3 &= \min_{m_j\in M} \max\left\{\tfrac{2.9+m_j^1}{0.25}, \tfrac{-0.5+m_j^2}{0.5}, \tfrac{-2.9+0.5+2m_j^3}{2-0.25-0.5}\right\}\\
&= \max\left\{\tfrac{2.9-3}{0.25}, \tfrac{-0.5+1}{0.5}, \tfrac{-2.9+0.5+2\times2}{2-0.25-0.5}\right\} = 1.28\\
\bar{\beta}_6^3 &= \min_{m_j\in M} \max\left\{\tfrac{-0.8+m_j^1}{0.25}, \tfrac{2.5+m_j^2}{0.5}, \tfrac{0.8-2.9+2m_j^3}{2-0.25-0.5}\right\}\\
&= \max\left\{\tfrac{-0.8+1}{0.25}, \tfrac{2.5-2}{0.5}, \tfrac{0.8-2.9+2\times1}{2-0.25-0.5}\right\} = 1
\end{aligned}$$

and therefore the monoidal strengthened cut is

$$0.8x_1 + 0.4x_2 + 1.6x_3 + 1.2x_4 + 1.28x_5 + x_6 \geq 1. \tag{3.2.39}$$

In this case, (3.2.39) is weaker than (3.2.38).

If we apply Theorem 3.2.1 to the disjunction (3.2.30) with $k = 3$, we obtain the lopsided cut $\tilde{\beta}^3 x \geq 1$ where

$$\tilde{\beta}^3_j := \begin{cases} \min\left\{\frac{-a_{1j}-a_{2j}+2}{2-a_{10}-a_{20}}, \min_{\substack{m_j\in M\\ m_j^3\geq 0}} \max\left\{\frac{a_{1j}+m_j^1}{a_{10}}, \frac{a_{2j}+m_j^2}{a_{20}}, \frac{-a_{1j}-a_{2j}+2m_j^3}{2-a_{10}-a_{20}}\right\}\right\} & j\in J_1 \\ \alpha^{ic}_j & j \in J\setminus J_1 \end{cases}$$

Computing these coefficients for the instance (3.2.35) we get

$$\begin{aligned} \tilde{\beta}^3_4 &= \min\left\{\tfrac{-1.2-1.6+2\times 1}{2-0.25-0.5}, \max\left\{\tfrac{1.2-1}{0.25}, \tfrac{1.6-1}{0.5}, \tfrac{-1.2-1.6+2\times 2}{2-0.25-0.5}\right\}\right\} = -0.64 \\ \tilde{\beta}^3_5 &= \min\left\{\tfrac{-2.9+0.5+2\times 1}{2-0.25-0.5}, \max\left\{\tfrac{2.9-3}{0.25}, \tfrac{-0.5+1}{0.5}, \tfrac{-2.9+0.5+2\times 2}{2-0.25-0.5}\right\}\right\} = -0.32 \\ \tilde{\beta}^3_6 &= \min\left\{\tfrac{0.8-2.5+2\times 1}{2-0.25-0.5}, \max\left\{\tfrac{-0.8+1}{0.25}, \tfrac{2.5-2}{0.5}, \tfrac{0.8-2.5+2\times 1}{2-0.25-0.5}\right\}\right\} = 0.24. \end{aligned}$$

Therefore the cut $\tilde{\beta}^3 x \geq 1$ is

$$0.8x_1 + 0.4x_2 + 1.6x_3 - 0.64x_4 - 0.32x_5 + 0.24x_6 \geq 1. \tag{3.2.40}$$

The lopsided cut (3.2.40) strictly dominates both the standard strengthened cut (3.2.38) and the monoidal stregthened cut (3.2.39).

3.2.3 Strictly weaker disjunctions

So far we only considered cuts derived from disjunctions with redundant terms. The next example shows that stronger cuts can also be obtained from disjunctions that are strictly weaker.

Example Consider the 2-row relaxation

$$P = \left\{ \begin{array}{llllllll} (y,x)\in\mathbb{R}^9 : & & & & & & & \\ y_1 = 0.25 & +0.58x_1 & +0.1x_2 & -0.48x_3 & +0.5x_4 & -1.5x_5 & +2.5x_6 & -7.2x_7 \\ y_2 = 0.5 & +0.29x_1 & -0.6x_2 & +0.32x_3 & +5.8x_4 & -0.1x_5 & +3.6x_6 & +2.6x_7 \\ x_j \geq 0, j\in J=\{1,\dots,7\} & & & & & & & \\ x_j \in \mathbb{Z}, j \in J_1 = \{4,\dots,7\} & & & & & & & \\ y_k\in\{0,1\}, k\in\{1,2\} \end{array} \right\}. \tag{3.2.41}$$

The point $(\bar{y}_1,\bar{y}_2,\bar{x}) = (0.25, 0.5, \bar{0})$ satisfies the two equations in P but does not satisfy the integrality conditions. We generate an intersection cut $\alpha^{ic}x \geq 1$ from the set S where

$$S = \left\{(y_1,y_2)\in\mathbb{R}^2 : 2y_1+y_2\geq 0;\ 4y_1-y_2\leq 4;\ y_2\leq 1\right\}.$$

The set S is a triangle with vertices $q_1 = (-\frac{1}{2},1); q_2 = (\frac{5}{4},1); q_3 = (\frac{2}{3},-\frac{4}{3})$. The intersection cut $\alpha^{ic}x\geq 1$ is

$$0.58x_1 + 0.4x_2 + 0.64x_3 + 11.6x_4 + 3.1x_5 + 7.2x_6 + 11.8x_7 \geq 1. \tag{3.2.42}$$

The coefficients of (3.2.42) are computed as follows:

$$
\begin{array}{lllll}
\alpha_1^{ic} & = \min_{\mu>0}\{\frac{1}{\mu} : 1+\mu(1.45)\geq 0; & 0.5+\mu(+2.03)\leq 4; & 0.5+\mu(0.29)\leq 1\} \\
& = \frac{1}{1.7241} = 0.58 \\
\alpha_2^{ic} & = \min_{\mu>0}\{\frac{1}{\mu} : 1+\mu(-0.4)\geq 0; & 0.5+\mu(1)\leq 4; & 0.5+\mu(-0.6)\leq 1\} \\
& = \frac{1}{2.5} = 0.4 \\
\alpha_3^{ic} & = \min_{\mu>0}\{\frac{1}{\mu} : 1+\mu(-0.64)\geq 0; & 0.5+\mu(-2.24)\leq 4; & 0.5+\mu(0.32)\leq 1\} \\
& = \frac{1}{1.5625} = 0.64 \\
\alpha_4^{ic} & = \min_{\mu>0}\{\frac{1}{\mu} : 1+\mu(6.8)\geq 0; & 0.5+\mu(-3.8)\leq 4; & 0.5+\mu(5.8)\leq 1\} \\
& = \frac{1}{0.0862} = 11.6 \\
\alpha_5^{ic} & = \min_{\mu>0}\{\frac{1}{\mu} : 1+\mu(-3.1)\geq 0; & 0.5+\mu(-5.9)\leq 4; & 0.5+\mu(-0.1)\leq 1\} \\
& = \frac{1}{0.3226} = 3.1 \\
\alpha_6^{ic} & = \min_{\mu>0}\{\frac{1}{\mu} : 1+\mu(8.6)\geq 0; & 0.5+\mu(6.4)\leq 4; & 0.5+\mu(3.6)\leq 1\} \\
& = \frac{1}{0.1389} = 7.2 \\
\alpha_7^{ic} & = \min_{\mu>0}\{\frac{1}{\mu} : 1+\mu(-11.8)\geq 0; & 0.5+\mu(-31.4)\leq 4; & 0.5+\mu(2.8)\leq 1\} \\
& = \frac{1}{0.0847} = 11.8.
\end{array}
$$

Applying standard strengthening on the coefficients associated to the variables x_4, x_5, x_6, x_7 we get

$$
\begin{array}{lll}
\alpha_4^{str} & = \min_{\substack{\mu>0 \\ p_1,p_2\in\mathbb{Z}}}\{\frac{1}{\mu} : & 1+\mu(6.8-2p_1-p_2)\geq 0; 0.5+\mu(-3.8-4p_1+p_2)\leq 4; \\
& & 0.5+\mu(5.8-p_2)\leq 1\} \\
& = \min_{\mu>0}\{\frac{1}{\mu} : & 1+\mu(6.8+0-6)\geq 0; 0.5+\mu(-3.8+0+6)\leq 4; \\
& & 0.5+\mu(5.8-6)\leq 1\} = \frac{1}{1.5909} = 0.6286 \\
\alpha_5^{str} & = \min_{\substack{\mu>0 \\ p_1,p_2\in\mathbb{Z}}}\{\frac{1}{\mu} : & 1+\mu(-3.1-2p_1-p_2)\geq 0; 0.5+\mu(-5.9-4p_1+p_2)\leq 4; \\
& & 0.5+\mu(-0.1-p_2)\leq 1\} \\
& = \min_{\mu>0}\{\frac{1}{\mu} : & 1+\mu(-3.1+4+0)\geq 0; 0.5+\mu(-5.9+8+0)\leq 4; \\
& & 0.5+\mu(-0.1+0)\leq 1\} = \frac{1}{1.6667} = 0.6 \\
\alpha_6^{str} & = \min_{\substack{\mu>0 \\ p_1,p_2\in\mathbb{Z}}}\{\frac{1}{\mu} : & 1+\mu(8.6-2p_1-p_2)\geq 0; 0.5+\mu(6.4-4p_1+p_2)\leq 4; \\
& & 0.5+\mu(3.6-p_2)\leq 1\} \\
& = \min_{\mu>0}\{\frac{1}{\mu} : & 1+\mu(8.6-4-4)\geq 0; 0.5+\mu(6.4-8+4)\leq 4; \\
& & 0.5+\mu(3.6-4)\leq 1\} = \frac{1}{1.4583} = 0.6857 \\
\alpha_7^{str} & = \min_{\substack{\mu>0 \\ p_1,p_2\in\mathbb{Z}}}\{\frac{1}{\mu} : & 1+\mu(-11.8-2p_1-p_2)\geq 0; 0.5+\mu(-31.4-4p_1+p_2)\leq 4; \\
& & 0.5+\mu(2.6-p_2)\leq 1\} \\
& = \min_{\mu>0}\{\frac{1}{\mu} : & 1+\mu(-11.8+14-3)\geq 0; 0.5+\mu(-31.4+28+3)\leq 4; \\
& & 0.5+\mu(2.6-3)\leq 1\} = \frac{1}{1.25} = 0.8.
\end{array}
$$

Thus, the standard strengthened cut $\alpha^{str}x \geq 1$ is

$$0.58x_1 + 0.4x_2 + 0.64x_3 + 0.6286x_4 + 0.6x_5 + 0.6857x_6 + 0.8x_7 \geq 1. \tag{3.2.43}$$

The cut (3.2.42) can be obtained from the disjunction

$$(2y_1 + y_2 \le 0) \vee (4y_1 - y_2 \ge 4) \vee (y_2 \ge 1)\,. \tag{3.2.44}$$

Applying monoidal strengthening to (3.2.44) we get the cut $\bar{\beta}x \ge 1$:

$$0.58x_1 + 0.4x_2 + 0.64x_3 + 2.2x_4 + 0.1x_5 + 1.8286x_6 + 1.0286x_7 \ge 1. \tag{3.2.45}$$

Consider now the disjunction

$$(-2y_1 - y_2 \ge 0) \vee (4y_1 - y_2 \ge 4) \vee (y_2 \ge 1) \vee (-y_2 \ge 1) \tag{3.2.46}$$

obtained from (3.2.44) by adding the term $(y_2 \le -1)$. The disjunction (3.2.46) is strictly weaker than (3.2.44) since $(y_1, y_2) = (0.7, -1)$ does not satisfy (3.2.44) but satisfies (3.2.46). In the nonbasic space, we can rewrite (3.2.46) as

$$\begin{array}{lll} (\sum_{j\in J_1}(2a_{1j} + a_{2j})x_j + \sum_{j\in J\setminus J_1}(2a_{1j} + a_{2j})x_j & \ge 2a_{10} + a_{20}) & \vee \\ (\sum_{j\in J_1}(-4a_{1j} + a_{2j})x_j + \sum_{j\in J\setminus J_1}(-4a_{1j} + a_{2j})x_j & \ge 4 - 4a_{10} + a_{20}) & \vee \\ (\sum_{j\in J_1}(-a_{2j})x_j + \sum_{j\in J\setminus J_1}(-a_{2j})x_j & \ge 1 - a_{20}) & \vee \\ (\sum_{j\in J_1}(a_{2j})x_j + \sum_{j\in J\setminus J_1}(a_{2j})x_j & \ge 1 + a_{20}) & \end{array} \tag{3.2.47}$$

Applying monoidal cut strengthening to (3.2.47) we derive the cut

$$0.58x_1 + 0.4x_2 + 0.64x_3 + 0.3429x_4 + 0.1x_5 + 0.4x_6 - 0.2x_7 \ge 1. \tag{3.2.48}$$

which strictly dominates both cuts (3.2.43) and (3.2.45). Moreover, it can be shown that (3.2.48) is not dominated by any combination of the cuts (3.2.43), (3.2.45), the 2 GMI cuts derived from (3.2.41) and the 3 lopsided cuts derived from the disjunction (3.2.44).

The detailed computation of the cut (3.2.48) is hereby included. The quantities

$$\begin{array}{ll} b_1 & = 2a_{10} + a_{20} - 3 \\ b_2 & = -4a_{10} + a_{20} - 1 \\ b_3 & = -a_{20} \\ b_4 & = -a_{20} - 1 \end{array}$$

are valid lower bounds for the left hand side term of each disjunct in (3.2.47). Therefore by the monoidal strengthening proposition we have that the disjunction

$$\begin{array}{lll} (\sum_{j\in J_1}(2a_{1j} + a_{2j} + 3m_j^1)x_j + \sum_{j\in J\setminus J_1}(2a_{1j} + a_{2j})x_j & \ge 2a_{10} + a_{20}) & \vee \\ (\sum_{j\in J_1}(-4a_{1j} + a_{2j} + 5m_j^2)x_j + \sum_{j\in J\setminus J_1}(-4a_{1j} + a_{2j})x_j & \ge 4 - 4a_{10} + a_{20}) & \vee \\ (\sum_{j\in J_1}(-a_{2j} + m_j^3)x_j + \sum_{j\in J\setminus J_1}(-a_{2j})x_j & \ge 1 - a_{20}) & \vee \\ (\sum_{j\in J_1}(a_{2j} + 2m_j^4)x_j + \sum_{j\in J\setminus J_1}(a_{2j})x_j & \ge 1 + a_{20}) & \end{array}$$

is valid for (3.2.41) for any $m \in M$, where $M = \{m \in \mathbb{Z}^4 : \sum_{k=1}^4 m_i \ge 0\}$. The monoidal strengthened cut $\bar{\beta}'x \geq 1$ from (3.2.47) has coefficients

$$\begin{aligned}
\bar{\beta}'_4 &= \min_{m\in M} \max\left\{\tfrac{-6.8+3m_4^1}{1}, \tfrac{-3.8+5m_4^2}{3.5}, \tfrac{5.8+m_4^3}{0.5}, \tfrac{-5.8+2m_4^4}{1.5}\right\}\\
&= \max\left\{\tfrac{-6.8+3(2)}{1}, \tfrac{-3.8+5(1)}{3.5}, \tfrac{5.8+1(-6)}{0.5}, \tfrac{-5.8+2(3)}{1.5}\right\} = 0.3429\\
\bar{\beta}'_5 &= \min_{m\in M} \max\left\{\tfrac{3.1+3m_5^1}{1}, \tfrac{-5.9+5m_5^2}{3.5}, \tfrac{-0.1+m_5^3}{0.5}, \tfrac{0.1+2m_5^4}{1.5}\right\}\\
&= \max\left\{\tfrac{3.1+3(-1)}{1}, \tfrac{-5.9+5(1)}{3.5}, \tfrac{-0.1+1(0)}{0.5}, \tfrac{0.1+2(0)}{1.5}\right\} = 0.1\\
\bar{\beta}'_6 &= \min_{m\in M} \max\left\{\tfrac{-8.6+3m_6^1}{1}, \tfrac{6.4+5m_6^2}{3.5}, \tfrac{3.6+m_6^3}{0.5}, \tfrac{-3.6+2m_6^4}{1.5}\right\}\\
&= \max\left\{\tfrac{-8.6+3(3)}{1}, \tfrac{6.4+5(-1)}{3.5}, \tfrac{3.6+1(-4)}{0.5}, \tfrac{-3.6+2(2)}{1.5}\right\} = 0.4\\
\bar{\beta}'_7 &= \min_{m\in M} \max\left\{\tfrac{11.8+3m_7^1}{1}, \tfrac{-31.4+5m_7^2}{3.5}, \tfrac{2.6+m_7^3}{0.5}, \tfrac{-2.6+2m_7^4}{1.5}\right\}\\
&= \max\left\{\tfrac{11.8+3(-4)}{1}, \tfrac{-31.4+5(6)}{3.5}, \tfrac{2.6+1(-3)}{0.5}, \tfrac{-2.6+2(1)}{1.5}\right\} = -0.2.
\end{aligned}$$

Bibliography

[1] Al-Khayyal, F. A., Falk, J. E., Jointly constrained biconvex programming. Mathematics of Operations Research 8, 1983.

[2] Andersen, K., Louveaux, Q., Weismantel, R., Wolsey, L.A., Cutting planes from two rows of a simplex tableau, Proceedings 12th Conference on Integer Programming and Combinatorial Optimization, Springer, 2007.

[3] Anstreicher, K. M., Semidefinite Programming versus the Reformulation Linearization Technique for Nonconvex Quadratically Constrained Quadratic Programming, Journal of Global Optimization 43, 2008.

[4] Balas,E., Disjunctive programming: properties of the convex hull of feasible points. Discrete Applied Mathematics 89, 1998.

[5] Balas, E., Intersection cuts - a new type of cutting planes for integer programming, Operations Research 19, 1939, 1971.

[6] Balas, E., Disjunctive Programming, Annals of Discrete Mathematics 5, 1979.

[7] Balas, E., Intersection cuts from maximal lattice-free convex sets and lift-and-project cuts from multiple-term disjunctions. American Mathematical Society Spring Western Section Meeting, San Francisco, April 25-27, 2009.

[8] Balas, E., Multiple-term disjunctive cuts and intersection cuts from multiple rows of the simplex tableau, 20-th International Symposium on Mathematical Programming, Chicago, August 23-28, 2009.

[9] Balas, E., Ceria, S., Cornuéjols, G., A lift-and-project cutting plane algorithm for mixed 0-1 programs, Mathematical Programming 58, 1993.

[10] Balas, E., Ceria, S., Cornuéjols, G., Mixed 0-1 Programming by Lift-and-Project in a Branch-and-Cut Framework, Management Science 42, 1996.

[11] Balas, E.,Jeroslow, R. G., Strengthening cuts for mixed integer programs, European Journal of Operational Research, 1980.

[12] Balas, E., Perregaard, M., A precise correspondence between lift-and-project cuts, simple disjunctive cuts, and mixed integer gomory cuts for 0-1 programming, Mathematical Programming 94, 2003.

[13] Basu, A., Bonami, P., Cornuejols, G., Margot, F., On the Relative Strength of Split, Triangle and Quadrilateral Cuts, SODA 2009.

[14] Basu, A., Bonami, P., Cornuejols, G., Margot, F., Experiments with Two-Row Cuts from Degenerate Tableaux, INFORMS Journal on Computing, 2010.

[15] Basu, A., Bonami, P., Cornuejols, G., Margot, F., Source code for the computational experiments of the paper "Experiments with Two-Row Cuts from Degenerate Tableaux", INFORMS Journal on Computing, 2010.

[16] Basu, A., Cornuéjols, G., Molinaro, M., A Probabilistic Analysis of the Strength of the Split and Triangle Closures, IPCO 2011.

[17] Bazaraa, M. S., Sherali, H. D., Shetty, C. M., Nonlinear Programming: Theory and Algorithms, Wiley, 2006.

[18] Benson, S. J., Ye, Y., DSDP5: Software For Semidefinite Programming. Available at `http://www-unix.mcs.anl.gov/DSDP`

[19] Borchers, B., CSDP, A C Library for Semidefinite Programming, Optimization Methods and Software 11(1), 1999.

[20] Borozan, V., Cornuéjols, G., Minimal valid inequalities for integer constraints. Mathematics of Operations Resarch 34, 2009.

[21] Boyd, S., Vandenberghe, L., Convex Optimization. Cambridge University Press, 2004.

[22] Burer, S., Letchford, A., On Non-Convex Quadratic Programming with Box Constraints, SIAM Journal of Optimization 20, 2009.

[23] COmputational INfrastructure for Operations Research (COIN-OR). `http://www.coin-or.org`

[24] Cook, W., Kannan, R., Schrijver,A., Chvatal Closures for Mixed Integer Programming Problems, Mathematical Programming 47, 1990.

[25] Cornuéjols, G., Li, Y., Vandenbussche, D.: K-cuts: a variation of Gomory mixed integer cuts from the LP tableau. INFORMS Journal on Computing 15, 2003.

[26] Cornuéjols, G., Margot, F., On the Facets of Mixed Integer Programs with Two Integer Variables and Two Constraints, Mathematical Programming 120, 2009.

[27] Del Pia, A., Wagner, C., Weismantel, R, A probabilistic comparison of the strength of split, triangle, and quadrilateral cuts. Operations Research Letters 39(4), 2011.

[28] Dash, S., Dey, S., Gunluk, O., Two dimensional lattice-free cuts and asymmetric disjunctions for mixed-integer polyhedra, to appear in Mathematical Programming, Optimization Online, January 2011. Available at `http://www.optimization-online.org/DB_HTML/2010/03/2582.html`

[29] Dash, S., Dey, S., Gunluk, O., Computational Experiments with Cross and Crooked Cross Cuts, Optimization Online, June 2011. Available at
`http://www.optimization-online.org/DB_HTML/2011/06/3071.html`

[30] Dey, S., Lodi, A., Tramontani, A., Wolsey, L.A. , On the Practical Strength of Two-Row Tableau Cuts, DEIS University of Bologna Technical Report OR-11-1, 2011.

[31] Dey, S., Wolsey, L.A., Two Row Mixed-Integer Cuts Via Lifting, Mathematical Programming 124, 2010.

[32] Dey, S., Wolsey, L.A., Composite lifting of group inequalities and an application to two-row mixing inequalities, Discrete Optimization 7, 2010.

[33] Espinoza, D. G., Computing with Multi-row Gomory Cuts IPCO 2008, A. Lodi et al. (Eds), Lecture Notes in Computer Science 5035, Springer, 2008.

[34] Fukuda, K., CDD+ 0.77. Available at
`http://www.ifor.math.ethz.ch/~fukuda/cdd_home/`

[35] GamsWorld Global Optimization library.
`http://www.gamsworld.org/global/globallib/globalstat.htm`

[36] Goemans, M. X., Worst-case Comparison of Valid Inequalities for the TSP, Mathematical Programming 69, 1995.

[37] Gomory, R. E., Johnson, E., T-space and cutting planes, Mathematical Programming Series B, 96, 2003.

[38] Gomory, R. E., Johnson, E., Evans, L., Corner Polyhedra and their connection with cutting planes Mathematical Programming Series B, 96, 2003.

[39] He, Q., Ahmed, S., Nemhauser, G.L., A Probabilistic Comparison of Split and Type 1 Triangle Cuts for Two-Row Mixed-Integer Programs. SIAM Journal on Optimization 21(3), 2011.

[40] Lovász, L., Schrijver, A. , Cones of matrices and set-functions and 0-1 optimization. SIAM Journal on Optimization, May 1991.

[41] Margot, F., Testing Cut Generators for Mixed-Integer Linear Programming, Mathematical Programming Computation 1, 2009.

[42] Margot, F, MIPLIB3_C_V2.
`http://wpweb2.tepper.cmu.edu/fmargot/MPS/miplib3_c_v2.tar.gz`

[43] Margot, F., Personal communication, August 2011.

[44] McCormick, G.P., Nonlinear programming: theory, algorithms and applications. John Wiley & sons, 1983.

[45] MIPLIB 3. Available at `http://www.caam.rice.edu/ bixby/miplib/miplib3.html`

[46] Nemhauser, G. L., Wolsey, L. A., Integer and Combinatorial Optimization, Wiley, 1999.

[47] Qualizza, A., Repository of linear extended formulations of GLOBALLib and BoxQP instances. Available at `http://www.andrew.cmu.edu/user/aqualizz/research/MIQCP`

[48] Qualizza, A., Belotti, P., Margot, F., Linear Programming Relaxations of Quadratically Constrained Quadratic Programs, IMA Volume Series, Springer, 2011.

[49] Saxena, A., Bonami, P., Lee, J., Convex Relaxations of Non-Convex Mixed Integer Quadratically Constrained Programs: Extended Formulations, Mathematical Programming 124, 2010.

[50] Saxena, A., Bonami, P., Lee, J., Convex Relaxations of Non-Convex Mixed Integer Quadratically Constrained Programs: Projected Formulations, to appear in Mathematical Programming, 2009.

[51] Sturm, J. F., SeDuMi: An Optimization Package over Symmetric Cones. Available at `http://sedumi.mcmaster.ca`

[52] Sherali, H. D., Adams, W. P., A reformulation-linearization technique for solving discrete and continuous nonconvex problems. Kluwer, Dordrecht 1998.

[53] Sherali, H. D., Fraticelli, B. M. P. Enhancing RLT relaxations via a new class of semidefinite cuts. Journal of Global Optimization 22, 2002.

[54] Shor, N.Z., Quadratic optimization problems. Tekhnicheskaya Kibernetika 1, 1987.

[55] Sivaramakrishnan, K., Mitchell, J., Semi-infinite linear programming approaches to semidefinite programming (SDP) problems. Novel Approaches to Hard Discrete Optimization, edited by P.M. Pardalos and H. Wolkowicz, Fields Institute Communications Series, American Mathematical Society, 2003.

[56] Sivaramakrishnan, K., Mitchell, J., Properties of a cutting plane method for semidefinite programming, Technical Report, Department of Mathematics, North Carolina State University, September 2007.

[57] Toh, K. C., Todd, M. J., Tütüncü, R. H., SDPT3: A MATLAB software for semidefinite-quadratic-linear programming. Available at `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`

[58] Vandenberghe, L., Boyd, S., Semidefinite Programming. SIAM Review 38 (1), 1996.

[59] Vandenbussche, D., Nemhauser, G. L., A branch-and-cut algorithm for nonconvex quadratic programs with box constraints. Mathematical Programming 102(3), 2005.

[60] Wolkowicz, H., Saigal, R., Vandenberghe, L., Handbook of Semidefinite Programming: Theory, Algorithms, and Applications. Springer, 2000.