Egon Balas

Disjunctive Programming

Springer

Disjunctive Programming

Egon Balas

Disjunctive Programming

Springer

Egon Balas
Tepper School of Business
Carnegie Mellon University
Pittsburgh, PA, USA

ISBN 978-3-030-00147-6   ISBN 978-3-030-00148-3 (eBook)
https://doi.org/10.1007/978-3-030-00148-3

Library of Congress Control Number: 2018957148

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book is meant for students and practitioners of optimization, mainly integer and nonconvex optimization, as an introduction to, and review of, the recently developed discipline of disjunctive programming. It should be of interest to all those who are trying to overcome the limits set by convexity to our problem-solving capability. Disjunctive programming is optimization over disjunctive sets, the first large class of nonconvex sets shown to be convexifiable in polynomial time.

There are no prerequisites for the understanding of this book, other than some knowledge of the basics of linear and integer optimization. Clarity of exposition was a main objective in writing it. Where background material is required, it is indicated through references.

Habent sua fata libelli, goes the Latin saying: *books have their own fate*. The basic document on Disjunctive Programming is the July 1974 technical report "Disjunctive Programming: Properties of the convex hull of feasible points. MSRR #348", which however has not appeared in print until 24 years later, when it was published as an invited paper with a remarkable foreword by Gerard Cornuejols and Bill Pulleyblank [7] (for a history of this episode see the introduction to [11]). The new theory, which was laid out without implementation of its cutting planes, let alone computational experience, stirred little if any enthusiasm at the time of its inception. But 15 years later, when Sebastian Ceria, Gerard Cornuejols and myself recast essentially the same results in a new framework which we called lift-and-project [19], the reaction was quite different. This time our work was focused on algorithmic aspects, with the cutting planes generated in rounds and embedded into an enumerative (branch and cut) framework, and was accompanied by an efficient computer code, MIPO, that was able to solve many problem instances that had been impervious to solution by branch-and-bound alone. This has led to a general effort on the part of software builders to incorporate cutting planes into a new generation of integer programming solvers, with the outcome known as the revolution in the state of the art in mixed integer programming that took place roughly in the period 1990–2005.

From a broader perspective, disjunctive programming is one of the early bridges between convex and nonconvex programming.

Several friends and colleagues have contributed to the improvement of this book through their comments and observations. Their list includes David Bernal, Dan Bienstock, Gérard Cornuéjols, Ignacio Grossmann, Michael Juenger, Alex Kazachkov, Tamas Kis, Andrea Qualizza, Thiago Serra, and an anonymous editor of the Springer Series Algorithms and Combinatorics.

The research underlying the results reported on in this book was supported generously throughout the last four decades by the National Science Foundation and the US Office of Naval Research. The writing of the book itself was supported by NSF Grant 1560828 and ONR Contract N000141512082.

Pittsburgh, PA, USA Egon Balas

Contents

Chapter 1
Disjunctive Programming and Its Relation to Integer Programming

1.1 Introduction

Disjunctive programming is optimization over disjunctive sets. A (linear) disjunctive set is the solution set of a system of (linear) inequalities joined by the logical connectives of conjunction ($\wedge$, "and", juxtaposition), disjunction ($\vee$, "or"), negation ($\neg$, "complement of"). The name reflects the fact that it is the presence of disjunctions that makes these sets nonconvex. Indeed, any (linear) disjunctive set is equivalent to a union of polyhedra, and while polyhedra are convex, their unions are not.

Linear programming arose during and immediately after World War II, and the appearance of computers in the early postwar years made the simplex method a practical tool for solving real world problems. It soon became clear that nonlinear convex optimization problems, i.e. problems whose locally optimal solutions are also globally optimal, can also be dealt with in promising ways. However, going beyond convexity turned out to be radically different. Integer programs, the most common representation of nonconvex optimization problems, for decades could only be solved at the level of toy models. Attempts at generating the integer hull, i.e. the convex hull of feasible integer points, which would reduce the problem to optimization over a convex set, turned out to be extremely difficult. From the point of view of solvability, a huge gap arose between convex and nonconvex optimization problems.

The theoretical significance of disjunctive programming consists in the fact that it is the first general class of nonconvex optimization problems amenable to compact convexification in the following sense: every linear disjunctive set in $\mathbb{R}^n$ is equivalent to a union of polyhedra in $\mathbb{R}^n$, and the closed convex hull of such a union is a polyhedron in $\mathbb{R}^{q\times n}$, where q is the number of polyhedra in the union. Thus the closed convex hull of a union of polyhedra has a compact representation in a higher dimensional space.

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_1

Logical conditions are pervasive in optimization problems inspired by real-world situations. Discontinuities, interruptions, sudden changes, logical alternatives, implications, either/or decisions, threshold conditions, etc. are common occurrences in such situations. Representing such constraints in optimization problems through 0-1 variables was the primary original motivation for integer programming. The so-called "big-M" method was the main tool for such representations, and the ability to formulate almost any logical condition through a couple of constraints and binary variables made integer programming into a universal modeling tool that possessed the flexibility that could not be accommodated by linear or convex programming formulations. However, the introduction of integer variables into an otherwise easily solvable linear or convex program made the resulting formulations radically harder to solve. Integer programming problems became the prototype of problems shown in the early 70s to be $\mathcal{NP}$-hard, a class for which no polynomial-time solution methods are known.

On the other hand, disjunctive programming, which also started in the early 70s, came about as an attempt to use logical conditions to help solving integer programs. Just as logical conditions can be represented as integrality constraints via the use of binary variables, the opposite is also true: integrality constraints can be represented via disjunctions. Furthermore, a closer look at logical or disjunctive constraints reveals that they can always be brought to the form of unions of polyhedra. Hence the discovery of a compact representation of the closed convex hull of a union of polyhedra points to a crucial breakthrough in the struggle to describe the integer hull, the convex hull of feasible integer points.

This discovery had repercussions in two distinct directions. On the one hand, it has led to the development of methods to use the disjunctive representation of integrality conditions towards approximating the integer hull through cutting planes, i.e. inequalities that cut off part of the linear programming polyhedron, but no feasible integer points: disjunctive cuts, lift-and-project cuts. On the other hand, it has led to the search for, and discovery of, higher dimensional representations of a given integer program, whose linear programming relaxation is a closer approximation of the integer hull than that of the original formulation: extended formulations. Both developments had a major impact on the development of integer programming during the last four decades.

1.2 Intersection Cuts

The origins of disjunctive programming are to be found in the concept of intersection cuts for integer programming, introduced around 1970.

We will consider (linear) mixed integer programs in the form

$$\min\{cx : x \in P_I\}, \tag{MIP}$$

where P_I represents the set of feasible integer points. The linear programming relaxation of (MIP), of central importance in all major solution methods, is

$$\min\{cx : x \in P\}, \tag{LP}$$

where

$$\begin{aligned} P &:= \{x \in \mathbb{R}^N : Ax \geq b, x \geq 0\} \\ &= \{x \in \mathbb{R}^N : \tilde{A}x \geq \tilde{b}\}, \quad N = \{1, \ldots, n\}, \end{aligned}$$

and where A is $m \times n$. Thus P_I is the set

$$P_I := P \cap \{x \in \mathbb{R}^N : x_j \in \mathbb{Z}, j \in N' \subseteq N\}.$$

We will assume throughout that the data of P are rational.

Introducing surplus variables into the system of inequalities defining P, $x_{n+i} = \sum_{j=1}^{n} a_{ij}x_j - b_i$, $i = 1, \ldots, m$, we obtain a system in $\mathbb{R}^{n+m}$. If $\bar{x} \in \mathbb{R}^{n+m}$ is an optimal solution to (LP) with I and J the index sets of basic and nonbasic variables, respectively, then the optimal simplex tableau is

$$\begin{aligned} x_i &= \bar{x}_i - \sum_{j \in J} \bar{a}_{ij}x_j, \; i \in I \\ x_j &\geq 0, \qquad\qquad\qquad j \in I \cup J \end{aligned}$$

When convenient, we will use the notations x_N and x_J for the subvectors of x with components indexed by $j \in N$ and $j \in J$, respectively.

Associated with every basic solution $\bar{x}$ to (LP) is the LP-cone $C(\bar{x})$ whose apex is $\bar{x}$ and whose facets are defined by the n hyperplanes that form the basis of $\bar{x}$. In order to avoid possible confusions in the case of a degenerate $\bar{x}$ which is the intersection of more than n hyperplanes, we will sometimes denote this cone by $C(J)$, meaning that the n tight inequalities defining the cone are those that are nonbasic at $\bar{x}$. Thus $C(J)$ has n extreme rays with direction vectors r^j, $j \in J$, with $r_i^j = -\bar{a}_{ij}$ for $i \in I$ and $r_j^j = 1$, $r_i^j = 0$ for $i \in J \setminus \{j\}$.

The convex hull of feasible integer points, $\text{conv}(P_I)$, is also known as the integer hull. The convex hull of integer points in $C(J)$ is known as the corner polyhedron [82], denoted $\text{corner}(J)$. The relationship between the various polyhedral relaxations of P_I introduced above can be summarized as $C(J) \supset P \supset \text{conv}(P_I)$ and $C(J) \supset \text{corner}(J) \supset \text{conv}(P_I)$.

The central object of research in integer programming is of course $\text{conv}(P_I)$, the integer hull, whose knowledge would transform a mixed integer program into a linear one. This elusive object has defied the strenuous efforts spent on identifying it, due to two circumstances: (a) its facets are hard to come by, and (b) their numbers are typically exponential in the problem dimensions. Therefore attempts to identify the convex hull were early on replaced by efforts to approximate it by generating inequalities that cut off parts of P, but no point of P_I, called cutting planes. Clearly, facet defining inequalities are the strongest possible cutting planes.

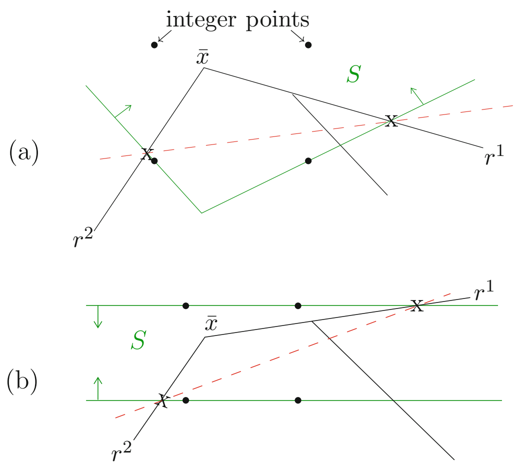
The first family of cutting planes, whose validity and convergence properties were discovered by Gomory, were the pure [80] and mixed [81] integer Gomory cuts. Both of these were derived from the condition that if x_k is a basic variable whose value $\bar{a}_{k0}$ in the current solution is fractional, then in any integer solution the value of x_k has to lie outside the interval $[\lfloor\bar{a}_{k0}\rfloor, \lceil\bar{a}_{k0}\rceil]$, i.e. the solution itself has to lie outside the strip $\{x : \lfloor\bar{a}_{k0}\rfloor \leq x_k \leq \lceil\bar{a}_{k0}\rceil\}$. The basic idea of an intersection cut generalizes this condition to the following broader requirement: if the current solution $\bar{x}$ is fractional, then given any convex set S whose interior, int S, contains $\bar{x}$ but no point of P_I, the region $P \cap \text{int}\, S$ can be cut off without cutting off any feasible integer point. Convex sets with the above property are called P_I-free. A straightforward way to generate a hyperplane, i.e. a cut, that accomplishes this, is to intersect the n extreme rays of the LP cone $C(J)$ with the boundary bd S of the P_I-free convex set S and use the hyperplane through the n intersection points as the cutting plane.

Figure 1.1a illustrates an intersection cut in two dimensions when S is the wedge defined by two halfplanes. In Fig. 1.1b, the two halfplanes define a strip, and the resulting intersection cut is a mixed integer Gomory cut.

Algebraically, intersecting the boundary of S with the n extreme rays of $C(J)$ amounts to finding, for each extreme ray $\bar{x} - \bar{a}_j\lambda_j$, $\lambda_j \geq 0$, $j \in J$, the value of $\lambda_j^* = \max\{\lambda_j : \bar{x} - \bar{a}_j\lambda_j \in S\}$, where $-\bar{a}_j$ has components $-\bar{a}_{ij}$ for $i \in I$, $-\bar{a}_{jj} = -1$ and $-\bar{a}_{ij} = 0$ for $i \in J \setminus \{j\}$.

Theorem 1.1 ([4]) *For $j \in J$, let λ_j^* be the value of λ_j for which the extreme ray $\bar{x} - \bar{a}_j\lambda_j$ of $C(J)$ intersects bd(S).*

Fig. 1.1 Intersection cuts



Then the intersection cut

$$\sum_{j \in J} \frac{1}{\lambda_j^*} x_j \geq 1$$

cuts off $\bar{x}$ *but no point of* P_I.

Proof The part of P cut off by the intersection cut, namely $\{x \in P : \sum_{j \in J} \frac{1}{\lambda_j^*} x_j < 1\}$, is contained in int S, which by definition contains no integer points. □

See [5] for related development.

The intersection cut expressed in terms of the nonbasic variables indexed by J, as above, will be denoted $\pi_J x_J \geq 1$. The same cut, when expressed in terms of the structural variables (indexed by N), will be denoted $\pi_N x_N \geq \pi_0$.

Now an intersection cut from a P_I-free convex set S, where, say,

$$S := \{x \in \mathbb{R}^n : d^i x \leq d_0^i, i \in Q\}$$

can also be viewed as a cut derived from the condition that every feasible point satisfy the disjunction

$$\bigvee_{i \in Q} (d^i x \geq d_0^i), \tag{1.1}$$

i.e. as a disjunctive cut.

Indeed, if int S contains no point of P_I, then every $x \in P_I$ must satisfy at least one of the inequalities obtained by reversing those defining S, where reversing means taking the weak complement. So a disjunctive cut seems to be just a different way of looking at an intersection cut. But this seemingly innocuous change of perspective has unexpected implications. If (1.1) is a valid disjunction that must be satisfied by every solution, then so is the disjunction obtained from (1.1) by adding to its terms all inequalities defining P, namely

$$\bigvee_{i \in Q} \begin{pmatrix} \tilde{A}x \geq \tilde{b} \\ d^i x \geq d_0^i \end{pmatrix} \tag{1.2}$$

Clearly, (1.2) is stronger, i.e. more restrictive, than (1.1). But beyond this consideration, the formulation (1.2) suggests a way to describe the family of valid cuts, i.e. of inequalities satisfying condition (1.2), as those defining the convex hull of a union of polyhedra. And this is the crux of the matter. But before we address the question of finding the convex hull of a union of polyhedra, we have to address the question of how to deal with disjunctive sets of inequalities that do not seem to conform to this model? For instance, what about a set like the one defined by the condition $((ax \leq b) \vee (cx \geq d)) \Rightarrow \left(\begin{pmatrix} g^1 x \geq g_0^1 \\ g^2 x \geq g_0^2 \end{pmatrix} \vee \begin{pmatrix} h^1 x \leq h_0^1 \\ h^2 x \leq h_0^2 \end{pmatrix} \right)$.

1.3 Inequality Systems with Logical Connectives

Mathematical logic analyzes methods of reasoning. More specifically, reasoning with logical connectives is the object of Propositional Calculus [101]. The basic building blocks of Propositional Calculus are sentences (also called propositions), denoted by the letters of the alphabet, that can take one of the two values *true* or *false*. Inequalities or equations are sentences that are true if satisfied, false if violated. To these sentences, logical connectives are applied to obtain new sentences, whose truth or falsity depends on the truth or falsity of the component sentences. These logical connectives are the *conjunction* ($\wedge$, "and", juxtaposition), *disjunction* ($\vee$, (nonexclusive) "or"), *negation* ($\neg$, "complement"), *conditional* ($\Rightarrow$, "if ... then"), *biconditional* ($\equiv$, "if and only if"). In our context, the tools of Propositional Calculus are applied to sentences or statements consisting of linear inequalities or sets thereof. Since conjunction and disjunction are analogous to intersection and union, elementary set theory [83] is also relevant in this context. A statement is true if the corresponding inequality system is feasible, false otherwise. Examining the logical connectives listed above, we find that the conditional $A \Rightarrow B$ is equivalent to the disjunction $(\neg A) \vee B$, and therefore the solution set of a system of linear inequalities joined to each other by these connectives is convex (polyhedral) as long as the connective "or" is not used; in other words, the disjunction is the sole connective whose presence results in a system that defines a nonconvex set. Hence the name of *disjunctive sets* for sets of inequalities and/or equations joined by logical connectives that include disjunctions, and *disjunctive programming* for optimization over such sets.

The validity of statements in Propositional Logic is proved by using truth tables; namely:

A	B	$A \vee B$	$A \wedge B$
T	T	T	T
T	F	T	F
F	T	T	F
F	F	F	F

This table says that if both statement letters A and B are true, then both $A \vee B$ and $A \wedge B$ are true; if both A and B are false, then both $A \vee B$ and $A \wedge B$ are false; but if exactly one of A and B is true, then $A \vee B$ is true but $A \wedge B$ is false.

Working with logical connectives requires knowledge of the basic laws of the Algebra of Propositions, which are as follows:

Associative law: $(A \vee B) \vee C = A \vee (B \vee C)$, $(A \wedge B) \wedge C = A \wedge (B \wedge C)$
Commutative law: $A \vee B = B \vee A$, $A \wedge B = B \wedge A$
Distributive law: $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$, $A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$

DeMorgan's law: $\neg(A \vee B) = \neg A \wedge \neg B$, $\neg(A \wedge B) = \neg A \vee \neg B$
Law of absorption: $(A \wedge B) \vee A = A$, $(A \vee B) \wedge A = A$

A logical statement, hence a disjunctive set, has many equivalent forms, and the use of the above laws allows us to move from one to another. The two extreme forms of a logical statement are the conjunctive and disjunctive normal forms.

A disjunction (a conjunction) is called elementary, if its terms do not contain further disjunctions (conjunctions). For instance, the expression $ax + by \leq c \vee dx + ey \leq f$ is an elementary disjunction; while the expression $ax + by \leq c, dx + ey \leq f$ is an elementary conjunction. The set defined by an elementary disjunction is called an *elementary disjunctive set*.

A disjunctive set is in *conjunctive normal form* (CNF) if it is the conjunction of terms (conjuncts) consisting of elementary disjunctions. It is in *disjunctive normal form* (DNF) if it is the disjunction of terms (disjuncts) consisting of elementary conjunctions. Note that statement letters (i.e. single inequalities) are considered (degenerate) elementary conjunctions or disjunctions.

For instance the following statement is in conjunctive normal form:

$$\begin{pmatrix} a_1 x \geq b_1 \\ \vdots \\ a_m x \geq b_m \end{pmatrix} \wedge (x_1 = 0 \vee x_1 = 1) \wedge \cdots \wedge (x_n = 0 \vee x_n = 1)$$

or, more concisely,

$$\begin{aligned} &Ax \geq b \\ &x_j \in \{0, 1\}, \ j = 1, \dots, n \end{aligned}$$

The same expression in disjunctive normal form reads

$$\begin{bmatrix} a_1 x \geq b_1 \\ \vdots \\ a_m x \geq b_m \\ \cdots\cdot \\ x_1 = 0 \\ x_2 = 0 \\ \vdots \\ x_n = 0 \end{bmatrix} \vee \begin{bmatrix} a_1 x \geq b_1 \\ \vdots \\ a_m x \geq b_m \\ \cdots\cdot \\ x_1 = 1 \\ x_2 = 0 \\ \vdots \\ x_n = 0 \end{bmatrix} \vee \cdots \vee \begin{bmatrix} a_1 x \geq b_1 \\ \vdots \\ a_m x \geq b_m \\ \cdots\cdot \\ x_1 = 1 \\ x_2 = 1 \\ \vdots \\ x_n = 1 \end{bmatrix}$$

or, more concisely,

$$\begin{aligned} Ax &\geq b \\ x &\in \{0, 1\}^n \end{aligned}$$

Note that the relationship between disjunction and conjunction (union and intersection) bears a certain similarity with the relationship between ordinary addition and multiplication; namely, they both abide by the associative and commutative laws. However, whereas multiplication is distributive with respect to addition, the converse is false: addition is not distributive with respect to multiplication. By contrast, disjunction and conjunction (union and intersection) are distributive with respect to each other, which makes a significant difference.

It is precisely this distributive property of disjunction and conjunction with respect to each other which makes it possible to put any logical statement in either conjunctive or disjunctive normal form.

Example 1 Consider the expression

$$(AB \vee CD) \wedge (E \vee F) \wedge G,$$

which is neither in disjunctive nor in conjunctive normal form. To bring it to DNF, we use the distributivity of conjunction to obtain

$$ABEG \vee ABFG \vee CDEG \vee CDFG;$$

and to bring it to CNF we use the distributivity of disjunction:

$$(A \vee C) \wedge (A \vee D) \wedge (B \vee C) \wedge (B \vee D) \wedge (E \vee F) \wedge G.$$

The ability to put any logical statement, hence any disjunctive set, in any of these two normal forms is crucial to the study of their properties: as it will soon become obvious, each of the two crucial properties of disjunctive sets is derived from, and applies to, one and only one of the two normal forms. This is remarkable in light of the fact that the two normal forms are just two ways to describe the same entity, namely the set of solutions to the system described in these two forms.

Of the two normal forms, the one that is of foremost interest to us, is the DNF, which has a geometric interpretation as a union of polyhedra.

Once the inequalities of a disjunctive set are given, the disjunctive and conjunctive normal forms are unique. However, it is important to notice that the inequalities expressing the conditions of a given problem, can usually be chosen in more than one way. For instance, the constraint set

$$\begin{aligned} 3x_1 + x_2 - 2x_3 + x_4 &\leq 1, \\ x_1 + x_2 + x_3 + x_4 &\leq 1 \\ x_j = 0 \text{ or } 1,\ j = 1, \ldots, 4, \end{aligned}$$

when put in disjunctive normal form, becomes a disjunction with $2^4 = 16$ terms. But the same constraint set can also be expressed as

$$3x_1 + x_2 - 2x_3 + x_4 \leq 1$$
$$\vee_{i=1}^{4}(x_i = 1, x_j = 0, j \neq i) \vee (x_j = 0, \forall j),$$

whose disjunctive normal form has only five terms.

1.4 Valid Inequalities for Disjunctive Sets

We start with a characterization of all valid inequalities for a disjunctive set. For that purpose we need the disjunctive set in DNF.

Theorem 1.2 ([6, 8] Farkas' Lemma for Disjunctive Sets) *Let*

$$F = \bigcup_{h \in Q} P_h, \qquad P_h := \{x \in \mathbb{R}^n : A^h x \geq b^h\}, h \in Q, \tag{1.3}$$

where each A^h is a $m_h \times n$ matrix, each b^h is an m_h-vector, and Q is a finite index set. Let $Q^* = \{h \in Q : P_h \neq \emptyset\}$. Then the inequality $\alpha x \geq \alpha_0$ is satisfied by all $x \in F$ if and only if there exist vectors $u^h \in \mathbb{R}^{m_h}$, $u^h \geq 0$, such that $\alpha = u^h A^h$, $\alpha_0 \leq u^h b^h$, $h \in Q^*$.

Proof The basic fact about valid inequalities for a disjunctive set F is that $\alpha x \geq \alpha_0$ is valid for F if and only if it is valid for every polyhedron in the union that makes up F. For suppose $\alpha x \geq \alpha_0$ cuts off some point p in one of the polyhedra P_h. Then $\alpha x \geq \alpha_0$ is invalid for F, since $p \in F$.

Now one of the forms of the famous Farkas' Lemma asserts that an inequality $\alpha x \geq \alpha_0$ is implied by (is a consequence of) the system $A^h x \geq b^h$ (where the latter is feasible) if and only if there exists a vector $u^h \geq 0$ such that $u^h A^h = \alpha$ and $u^h b^h \geq \alpha_0$. □

Remark 1.3 If in Theorem 1.2 we define P^h as $\{x \in R^n : A^h x \geq b^h, x \geq 0\}$, $h \in Q$, then the condition $\alpha = u^h A^h$, $h \in Q^*$, is to be replaced with $\alpha \leq u^h A^h$, $h \in Q^*$. Let's call the resulting version Theorem 1.2′.

Remark 1.4 If the ith inequality defining P_h is replaced by an equation, the ith component of u^h is to be made unrestricted in sign for the Theorem to remain valid.

An alternative way to describe the family of inequalities defined by Theorem 1.2′ is the set of those $\alpha x \geq \alpha_0$ satisfying

$$\begin{aligned} \alpha_j &= \max\nolimits_{h \in Q^*} u^h A^h_j \;\; j \in N \\ \alpha_0 &\leq \min\nolimits_{h \in Q^*} u^h b^h \end{aligned} \tag{1.4}$$

for some $u^h \geq 0$, $h \in Q^*$.

Since $Q^* \subseteq Q$, the if part of the Theorem remains of course valid if Q^* is replaced by Q.

Since (1.4) defines *all* the valid inequalities for (1.3), every valid cutting plane for a disjunctive program can be obtained from (1.4) by choosing suitable multipliers u_i^h. The simplest such cut is from the condition $x_i \leq 0 \vee x_i \geq 1$ imposed on the basic variable x_i of a simplex tableau, $x_i = a_{i0} - \sum_{j \in J} a_{ij} x_j$ where $0 < a_{i0} < 1$.

Rewriting $x_i \leq 0 \vee x_i \geq 1$ as

$$\sum_{j \in J} a_{ij} x_j \geq a_{i0} \vee \sum_{j \in J} (-a_{ij}) x_j \geq 1 - a_{i0}$$

and applying (1.4) yields

$$\sum_{j \in J} \max \left\{ \frac{a_{ij}}{a_{i0}}, \frac{(-a_{ij})}{(1 - a_{i0})} \right\} x_j \geq 1 \tag{1.4$'$}$$

Note that the cut derived from $x_i \leq 0, \vee x_i \geq 1$ is by far not unique; it depends on the set on nonbasic variables in terms of which x_i as a basic variable is expressed.

An important feature of the general expression (1.4) is that putting a given cut in the form (1.4) may suggest ways of improving it by an appropriate choice of the multipliers.

We illustrate this by the following

Example 2 Consider the mixed integer program whose constraint set is

$$\begin{aligned}
&x_1 = 0.2 + 0.4(-x_3) + 1.3(-x_4) - 0.01(-x_5) + 0.07(-x_6) \\
&x_2 = 0.9 - 0.3(-x_3) + 0.4(-x_4) - 0.04(-x_5) + 0.1(-x_6) \\
&x_j \geq 0, \quad j = 1, \dots, 6, \quad x_j \text{ integer}, \quad j = 1, \dots, 4.
\end{aligned}$$

This problem is taken from the paper [87], which also lists six cutting planes derived from the extreme valid inequalities for the associated group problem:

$$\begin{aligned}
&0.75x_3 + 0.875x_4 + 0.0125x_5 + 0.35x_6 \geq 1 \\
&0.778x_3 + 0.444x_4 + 0.40x_5 + 0.111x_6 \geq 1, \\
&0.333x_3 + 0.667x_4 + 0.033x_5 + 0.35x_6 \geq 1, \\
&0.50x_3 + x_4 + 0.40x_5 + 0.25x_6 \geq 1, \\
&0.444x_3 + 0.333x_4 + 0.055x_5 + 0.478x_6 \geq 1, \\
&0.394x_3 + 0.636x_4 + 0.346x_5 + 0.155x_6 \geq 1.
\end{aligned}$$

The first two of these inequalities are the mixed-integer Gomory cuts derived from the row of x_1 and x_2, respectively. To show how they can be improved, we first derive them as they are. To do this, for a row of the form

$$x_i = a_{i0} + \sum_{i \in J} a_{ij}(-x_i),$$

with x_j integer-constrained for $j \in J_1$, continuous for $j \in J_2$, one defines $f_{ij} = a_{ij} - \lfloor a_{ij} \rfloor$, $j \in J \cup \{0\}$, and $\varphi_{i0} = f_{i0}$,

$$\varphi_{ij} = \begin{cases} f_{ij}, & j \in J_1^+ = \{j \in J_1 | f_{i0} \geq f_{ij}\}, \\ f_{ij} - 1, & j \in J_1^- = \{j \in J_1 | f_{i0} < f_{ij}\}, \\ a_{ij}, & j \in J_2 \end{cases}$$

Then every x which satisfies the above equation and the integrality constraints on x_j, $j \in J_1 \cup \{i\}$, also satisfies the condition

$$y_i = \varphi_{i0} + \sum_{j \in J} \varphi_{ij}(-x_j), \qquad y_i \text{ integer.}$$

For the two equations of the example, the resulting conditions are

$$y_1 = 0.2 - 0.6(-x_3) - 0.7(-x_4) - 0.01(-x_5) + 0.07(-x_6), \; y_1 \text{ integer},$$
$$y_2 = 0.9 + 0.7(-x_3) + 0.4(-x_4) - 0.04(-x_5) + 0.1(-x_6), \quad y_2 \text{ integer}.$$

Since each y_i is integer-constrained, they have to satisfy the disjunction $y_i \leq 0 \vee y_i \geq 1$. Applying formula (1.4′), the cuts for the cases $i = 1, 2$ are

$$\frac{0.6}{0.8}x_3 + \frac{0.7}{0.8}x_4 + \frac{0.01}{0.8}x_5 + \frac{0.07}{0.2}x_6 \geq 1,$$

and

$$\frac{0.7}{0.9}x_3 + \frac{0.4}{0.9}x_4 + \frac{0.04}{0.1}x_5 + \frac{0.1}{0.9}x_6 \geq 1.$$

These are precisely the first two inequalities of the above list.

Since all cuts discussed here are stated in the form ≥ 1, the smaller the jth coefficient, the stronger is the cut in the direction j. We would thus like to reduce the size of the coefficients as much as possible.

Now suppose that instead of $y_1 \leq 0 \vee y_1 \geq 1$, we use the disjunction

$$\{y_1 \leq 0\} \vee \left\{ \begin{array}{l} y_1 \geq 1 \\ x_1 \geq 0 \end{array} \right\},$$

which of course is also satisfied by every feasible x.

Then, applying Theorem 1.2 in the form (1.4) with multipliers 5, 5 and 15 for $y_1 \leq 0$, $y_1 \geq 1$ and $x_1 \geq 0$ respectively, we obtain the cut whose coefficients are

$$\max\left\{\frac{5 \times (-0.6)}{5 \times 0.2}, \frac{5 \times 0.6 + 15 \times (-0.4)}{5 \times 0.8 + 15 \times (-0.2)}\right\} = -3,$$
$$\max\left\{\frac{5 \times (-0.7)}{5 \times 0.2}, \frac{5 \times 0.7 + 15 \times (-1.3)}{5 \times 0.8 + 15 \times (-0.2)}\right\} = -3.5,$$
$$\max\left\{\frac{5 \times (-0.01)}{5 \times 0.2}, \frac{5 \times 0.01 + 15 \times 0.01}{5 \times 0.8 + 15 \times (-0.2)}\right\} = 0.2,$$
$$\max\left\{\frac{5 \times 0.07}{5 \times 0.2}, \frac{5 \times (-0.07) + 15 \times (-0.07)}{5 \times 0.8 + 15 \times (-0.2)}\right\} = 0.35,$$

that is

$$-3x_3 - 3.5x_4 + 0.2x_5 + 0.35x_6 \geq 1.$$

The sum of coefficients on the left hand side has been reduced from 1.9875 to -5.95.

Similarly, for the second cut, if instead of $y_2 \leq 0 \vee y_2 \geq 1$ we use the disjunction

$$\left\{ \begin{array}{l} y_2 \leq 0 \\ x_1 \geq 0 \end{array} \right\} \vee \{y_2 \geq 1\},$$

with multipliers $10, 40$ and 10 for $y_2 \leq 0$, $x_1 \geq 0$ and $y_2 \geq 1$ respectively, we obtain the cut

$$-7x_3 - 4x_4 + 0.4x_5 - x_6 \geq 1.$$

Here the sum of left hand side coefficients has been reduced from 1.733 to -11.6.

1.5 Duality for Disjunctive Programs

In this section we state a duality theorem for disjunctive programs, which generalizes to this class of problems the duality theorem of linear programming.

Consider the disjunctive program

$$z_0 = \min cx, \qquad \bigvee_{h \in Q} \left\{ \begin{array}{c} A^h x \geq b^h \\ x \geq 0 \end{array} \right\}, \tag{DP}$$

where A^h is a matrix and b^h a vector, $\forall h \in Q$.

We define the *dual* of (DP) to be the problem

$$w_0 = \max w, \qquad \bigwedge_{h \in Q} \left\{ \begin{array}{r} w - u^h b^h \leq 0 \\ u^h A^h \leq c \\ u^h \geq 0 \end{array} \right\}. \tag{DD}$$

The constraint set of (DD) requires each u^h, $h \in Q$, to satisfy the corresponding bracketed system, and w to satisfy each of them.

Let

$$X_h = \{x | A^h x \geq b^h, x \geq 0\}, \quad \bar{X}_h = \{x | A^h x \geq 0, x \geq 0\};$$
$$U_h = \{u^h | u^h A^h \leq c, u^h \geq 0\}, \; \bar{U}_h = \{u^h | u^h A^h \leq 0, u^h \geq 0\}.$$

Further, let

$$Q^* = \{h \in Q | X_h \neq \emptyset\}, \qquad Q^{**} = \{h \in Q | U_h \neq \emptyset\}.$$

We will assume the following

Regularity Condition

$$(Q^* \neq \emptyset, Q \setminus Q^{**} \neq \emptyset) \Rightarrow Q^* \setminus Q^{**} \neq \emptyset;$$

i.e., if (DP) is feasible and (DD) is infeasible, then there exists $h \in Q$ such that $X_h \neq \emptyset$, $U_h = \emptyset$.

Theorem 1.5 ([9, 10]) *Assume that* (DP) *and* (DD) *satisfy the regularity condition. Then exactly one of the following two situations holds.*

(1) Both problems are feasible; each has an optimal solution and $z_0 = w_0$.
(2) One of the problems is infeasible; the other one either is infeasible or has no finite optimum.

Proof

(1) Assume that both (DP) and (DD) are feasible. If (DP) has no finite minimum, then there exists $h \in Q$ such that $\bar{X}_h \neq \emptyset$ and $\bar{x} \in \bar{X}_h$ such that $c\bar{x} < 0$. But then $U_h = \emptyset$, i.e., (DD) is infeasible; a contradiction.

Thus (DP) has an optimal solution, say $\bar{x}$. Then the inequality $cx \geq z_0$ is a consequence of the constraint set of (DP); i.e., $x \in X_h$ implies $cx \geq z_0$, $\forall h \in Q$. But then for all $h \in Q^*$, there exists $u^h \in U_h$ such that $u^h b^h \geq z_0$. Further, since (DD) is feasible, for each $h \in Q \setminus Q^*$ there exists $\hat{u}^h \in U_h$; and since $X_h = \emptyset$ (for $h \in Q \setminus Q^*$), there also exists $\bar{u}^h \in \bar{U}_h$ such that $\bar{u}^h b^h > 0$, $\forall h \in Q \setminus Q^*$. But then, defining

$$u^h(\lambda) = \hat{u}^h + \lambda \bar{u}^h, \quad h \in Q \setminus Q^*,$$

for λ sufficiently large, $u^h(\lambda) \in U_h$, $u^h(\lambda)b^h \geq z_0$, $\forall h \in Q \setminus Q^*$.

Hence for all $h \in Q$, there exist vectors u^h satisfying the constraints of (DD) for $w = z_0$. To show that this is the maximal value of w, we note that since $\bar{x}$ is optimal for (DP), there exists $h \in Q$ such that

$$c\bar{x} = \min\{cx | x \in X^h\}.$$

But then by linear programming duality,

$$\begin{aligned} c\bar{x} &= \max\{u^h b^h | u^h \in U_h\} \\ &= \max\{w | w - u^h b^h \leq 0, u^h \in U_h\} \\ &\geq \max\left\{ w \,\middle|\, \bigwedge_{h \in Q} (w - u^h b^h \leq 0, u^h \in U_h) \right\} \end{aligned}$$

i.e., $w \leq z_0$, and hence the maximum value of w is $w_0 = z_0$.

(2) Assume that at least one of (DP) and (DD) is infeasible. If (DP) is infeasible, $X_h = \emptyset, \forall h \in Q$; hence for all $h \in Q$, there exists $\bar{u}^h \in \bar{U}_h$ such that $\bar{u}^h b^h > 0$.

If (DD) is also infeasible, we are done. Otherwise, for each $h \in Q$ there exists $\hat{u} \in U_h$. But then defining

$$u^h(\lambda) = \hat{u}^h + \lambda \bar{u}^h, \qquad h \in Q,$$

$u^h(\lambda) \in U_h, h \in Q$, for all $\lambda > 0$, and since $\bar{u}^h b^h > 0, \forall h \in Q$, w can be made arbitrarily large by increasing λ; i.e., (DD) has no finite optimum.

Conversely, if (DD) is infeasible, then either (DP) is infeasible and we are done, or else, from the regularity condition, $Q^* \setminus Q^{**} \neq \emptyset$; and for $h \in Q^* \setminus Q^{**}$ there exists $\hat{x} \in X_h$ and $\bar{x} \in \bar{X}_h$ such that $c\bar{x} < 0$. But then

$$x(\mu) = \hat{x} + \mu\bar{x}$$

is a feasible solution to (DP) for any $\mu > 0$, and since $c\bar{x} < 0$, z can be made arbitrarily small by increasing μ; i.e., (DP) has no finite optimum. □

The above theorem asserts that either situation 1 or situation 2 holds for (DP) and (DD) if the regularity condition is satisfied. The following Corollary shows that the condition is not only sufficient but also necessary.

Corollary 1.6 *If the regularity condition does not hold, then if* (DP) *is feasible and* (DD) *is infeasible,* (DP) *has a finite minimum (i.e., there is a "duality gap").*

Proof Let (DP) be feasible, (DD) infeasible, and $Q^* \setminus Q^{**} = \emptyset$, i.e., for every $h \in Q^*$, let $U_h \neq \emptyset$. Then for each $h \in Q^*$, $\min\{cx | x \in X_h\}$ is finite, hence (DP) has a finite minimum. □

Remark 1.7 The theorem remains true if some of the variables of (DP) [of (DD)] are unconstrained, and the corresponding constraints of (DD) [of (DP)] are equalities.

The regularity condition can be expected to hold in all but some rather peculiar situations. In linear programming duality, the case when both the primal and the dual problem is infeasible only occurs for problems whose coefficient matrix A has the very special property that there exists $x \neq 0$, $u \neq 0$, satisfying the homogeneous system

$$Ax \geq 0,\ x \geq 0;$$
$$uA \leq 0,\ u \geq 0.$$

In this context, our regularity condition requires that, if the primal problem is feasible and the dual is infeasible, then at least one of the matrices A^h whose associated set U_h are infeasible, should not have the above mentioned special property.

Though most problems satisfy this requirement, nevertheless there *are* situations when the regularity condition breaks down, as illustrated by the following example.

Consider the disjunctive program

$$\min -x_1 - 2x_2,$$
$$\left\{\begin{array}{r} -x_1 + x_2 \geq 0 \\ -x_1 - x_2 \geq -2 \\ x_1, x_2 \geq 0 \end{array}\right\} \bigvee \left\{\begin{array}{r} -x_1 + x_2 \geq 0 \\ x_1 - x_2 \geq 1 \\ x_1, x_2 \geq 0 \end{array}\right\} \tag{DP}$$

and its dual

$$
\begin{array}{rrrrrl}
\max w, & & & & & \\
w & & +\, 2u_2^1 & & & \le \ \ 0, \\
 & -\, u_1^1 & -\ \ u_2^1 & & & \le -1, \\
 & u_1^1 & -\ \ u_2^1 & & & \le -2, \\
w & & & & -\, u_2^2 & \le \ \ 0, \\
 & & & -\, u_1^2 & +\, u_2^2 & \le -1, \\
 & & & u_1^2 & -\, u_2^2 & \le -2, \\
 & & & & u_i^k & \ge \ \ 0, \qquad i = 1, 2; \quad k = 1, 2.
\end{array}
\tag{DD}
$$

The primal problem (DP) has an optimal solution $\bar{x} = (0, 2)$, with $c\bar{x} = -4$; whereas the dual problem (DD) is infeasible. This is due to the fact that $Q^* = \{1\}$, $Q \setminus Q^{**} = \{2\}$ and $X_2 = \emptyset$, $U_2 = \emptyset$, i.e., $Q^* \setminus Q^{**} = \emptyset$, hence the regularity condition is violated. Here

$$
X_2 = \left\{ x \in \mathbb{R}_+^2 \;\middle|\; \begin{array}{c} -x_1 + x_2 \ge 0 \\ x_1 - x_2 \ge 1 \end{array} \right\}, \qquad U_2 = \left\{ u \in \mathbb{R}_+^2 \;\middle|\; \begin{array}{c} -u_1^2 + u_2^2 \le -1 \\ u_1^2 - u_2^2 \le -2 \end{array} \right\}.
$$

Chapter 2
The Convex Hull of a Disjunctive Set

2.1 The Convex Hull Via Lifting and Projection

There are two ways to describe the convex hull of a disjunctive set, and each one provides its insights. Here we discuss the first one.

Lifting is a topic introduced into the discrete optimization literature as the lifting of inequalities, in the following sense: given a (MIP) in $\mathbb{R}^n$ as defined in Sect. 1.1, suppose we know $q < n$ coefficients of a valid inequality; how do we find values for the remaining $n - q$ coefficients that leave the inequality valid? There is a rich literature on this and related topics. Here we refer to lifting in a broader sense, as an operation which in a way is the reverse of projection. Typically, discrete or combinatorial optimization problems can be formulated in more than one way. Given a formulation in, say, $\mathbb{R}^n$, by lifting we mean finding a formulation in a higher dimensional space that offers some advantage. In the current context, our objective is to find a description of the convex hull of a disjunctive set, i.e. of a union of polyhedra, and we attain our objective by formulating the convex hull in a higher dimensional space and projecting back the resulting formulation onto the original space.

One possible way to derive this higher-dimensional formulation is the following. Consider a disjunctive program

$$\begin{aligned} &\min cx \\ &\textstyle\bigvee_{h\in Q}(A^h x \geq b^h) \end{aligned} \tag{DP}$$

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_2

Its dual, defined (for the case where $x \geq 0$) in Sect. 1.5, can be stated as the linear program

$$\begin{aligned} \max\ & w \\ w - u^h b^h &\leq 0 \\ u^h A^h &= c \\ u^h &\geq 0 \quad h \in Q \end{aligned} \tag{DD}$$

(Here equation replaces inequality in (DD) because x is not sign restricted in (DP).)

Taking the linear programming dual of (DD) we get (denoting $|Q| = q$)

$$\begin{array}{lllll} \min & cy^1 + & \cdots & + cy^q & \\ & A^1 y^1 - b^1 y_0^1 & & & \geq 0 \\ & & \ddots & & \vdots \\ & & & A^q y^q - b^q y_0^q & \geq 0 \\ & y_0^1 + & \cdots & + y_0^q & = 1 \\ & & & y_0^1, \ldots, y_0^q & \geq 0 \end{array}$$

If we now represent the sum of the y^h by the new variable $x \in \mathbb{R}^n$, the result is

$$\begin{aligned} \min\ & cx \\ x - \sum_{h \in Q} y^h &= 0 \\ A^h y^h - b^h y_0^h &\geq 0 \quad h \in Q \\ \sum_{h \in Q} y_0^h &= 1 \\ y_0^h &\geq 0,\ h \in Q. \end{aligned} \tag{DP'}$$

The fact that (DP′) is the linear programming dual of (DD) suggests that this problem is equivalent to the disjunctive program (DP). This is indeed the case, and furthermore the constraint set of (DP′) describes, in its higher-dimensional space, the convex hull of the feasible set of (DP). Indeed, consider the polyhedra $P_h := \{x \in \mathbb{R}^n : A^h x \geq b^h\}$, $h \in Q$, and let $F := \bigcup_{h \in Q} P_h$, $Q^* := \{h \in Q : P_h \neq \emptyset\}$.

Theorem 2.1 ([6]) *The closed convex hull of F, $\operatorname{cl}\operatorname{conv} F$, is the set of those x for which there exist vectors $(y^h, y_0^h) \in \mathbb{R}^{n+1}$, $h \in Q^*$, satisfying*

$$\begin{aligned} x - \sum_{h\in Q^*} y^h \qquad\qquad &= 0 \\ A^h y^h - b^h y_0^h &\geq 0 \\ y_0^h &\geq 0 \quad h \in Q^* \\ \sum_{h\in Q^*} y_0^h &= 1. \end{aligned} \tag{2.1}$$

We give two proofs of this theorem.

First Proof Let $P_Q := \operatorname{cl}\operatorname{conv} F$, let $\mathcal{P}$ be the set of vectors $(x, \{y^h, y_0^h\}_{h\in Q^*})$ satisfying (2.1), and let $X(\mathcal{P})$ be the set of $x \in \mathbb{R}^n$ for which there exist vectors (y^h, y_0^h), $h \in Q^*$ such that $(x, \{y^h, y_0^h\}_{h\in Q^*}) \in \mathcal{P}$.

If Q is finite and $F \neq \emptyset$, then Q^* is nonempty.

(i) We first show that $P_Q \subseteq X(\mathcal{P})$. If $x \in P_Q$, then x is the convex combination of at most $|Q^*|$ points u^h, each from a different P_h, i.e.

$$x = \sum_{h\in Q^*} u^h \lambda_h, \lambda_h \geq 0, \qquad h \in Q^*, \text{ with } \sum_{h\in Q^*} \lambda_h = 1,$$

where for each $h \in Q^*$, u^h satisfies $A^h u^h \geq b^h$. But if $(x, \{u^h\lambda_h, \lambda_h\}_{h\in Q^*})$ satisfies the above constraints, then $(x, \{y^h, y_0^h\}_{h\in Q^*})$, where $y_0^h = \lambda_h$ and $y^h = u^h\lambda_h$, clearly satisfies the constraints of $\mathcal{P}$; hence $x \in P_Q \Rightarrow x \in X(\mathcal{P})$.

(ii) Next we show that $X(\mathcal{P}) \subseteq P_Q$. Let $\bar{x} \in X(\mathcal{P})$, with $(\bar{x}, \{\bar{y}^h, \bar{y}_0^h\}_{h\in Q^*}) \in \mathcal{P}$. Let $Q_1^* := \{h \in Q^* : \bar{y}_0^h > 0\}$, $Q_2^* := \{h \in Q^* : \bar{y}_0^h = 0\}$.

For $h \in Q_1^*$, $\frac{\bar{y}^h}{\bar{y}_0^h} \in P_h$, hence $\frac{\bar{y}^h}{\bar{y}_0^h} = \sum_{i\in U_h} \mu^{hi} u^{hi} + \sum_{k\in V_h} \nu^{hk} v^{hk}$ for some $u^{hi} \in \operatorname{vert} P_h$, $i \in U_h$, and $v^{hk} \in \operatorname{dir} \mathrm{P_h}$, $k \in V_h$, where U_h and V_h are finite index sets, $\mu^{hi} \geq 0$, $i \in U_h$, $\nu^{hk} \geq 0$, $k \in V_h$, and $\sum_{i\in V_h} \mu^{hi} = 1$; or, setting $\mu^{hi}\bar{y}_0^h = \theta^{hi}$, $\nu^{hk}\bar{y}_0^h = \sigma^{hk}$,

$$\bar{y}^h = \sum_{i\in U_h} \theta^{hi} u^{hi} + \sum_{k\in V_h} \sigma^{hk} v^{hk}$$

with $\theta^{hi} \geq 0$, $i \in U_h$, $\sigma^{hk} \geq 0$, $k \in V_h$, and $\sum_{i\in U_h} \theta^{hi} = \bar{y}_0^h$.

For $h \in Q_2^*$, $\bar{y}^h$ is either 0 or a nontrivial solution to the homogeneous system $A^h y^h \geq 0$, hence $\bar{y}^h = \sum_{k\in V_h} \sigma^{hk} v^{hk}$, $\sigma^{hk} \geq 0$, $k \in V_h$ for some $v^{hk} \in \operatorname{dir} \mathrm{P_h}$,

$k \in V_h$. Let $\bar{Q}_2^* := \{h \in Q_2^* : \bar{y}^h \neq 0\}$. Then

$$\begin{aligned}\bar{x} &= \sum_{h \in Q^*} \bar{y}^h \\ &= \sum_{h \in Q_1^*} \left(\sum_{i \in U_h} \theta^{hi} u^{hi} + \sum_{k \in V_h} \sigma^{hk} v^{hk} \right) + \sum_{h \in Q_2^*} \left(\sum_{k \in V_h} \sigma^{hk} v^{hk} \right)\end{aligned}$$

with $\sum_{h \in Q_1^*} \sum_{i \in U_h} \theta^{hi} = \sum_{h \in Q_1^*} \bar{y}_0^h = 1$, i.e. $\bar{x}$ is the convex combination of finitely many points and directions of F. Hence $\bar{x} \in \operatorname{cl} \operatorname{conv} F$. □

Corollary 2.2 *The following relationships characterize the connection between P_Q and $\mathcal{P}$:*

(i) if x^* is an extreme point of P_Q, then $(\bar{x}, \{\bar{y}^h, \bar{y}_0^h\}_{h \in Q^*})$ is an extreme point of $\mathcal{P}$, with $\bar{x} = x^*$, $(\bar{y}^k, \bar{y}_0^k) = (x^*, 1)$ for some $k \in Q^*$, and $(\bar{y}^h, \bar{y}_0^h) = (0, 0)$ for $h \in Q \setminus \{k\}$.
(ii) if $(\bar{x}, \{\bar{y}^h, \bar{y}_0^h\}_{h \in Q^*})$ is an extreme point of $\mathcal{P}$, then $\bar{y}^k = \bar{x}$ and $\bar{y}_0^k = 1$ for some $k \in Q^*$, $(\bar{y}_h, \bar{y}_0^h) = (0, 0)$ for $h \in Q \setminus \{k\}$, and $\bar{x}$ is an extreme point of P_Q.

Proof

(i) If x^* is an extreme point of P_Q, then x^* is an extreme point of P_k for some $k \in Q^*$, hence $\bar{x} = x^*$ is a basic solution to $A^k x \geq b^k$ for some $k \in Q^*$. Therefore $A^k \bar{y}^k \geq b^k \bar{y}_0^k$ for $(\bar{y}_k, \bar{y}_0^k) = (\bar{x}, 1)$ and $(\bar{y}^h, \bar{y}_0^h) = (0, 0)$ for $h \in Q \setminus \{k\}$, and $(\bar{x}, \{\bar{y}^h, \bar{y}_0^h\}_{h \in Q})$ is an extreme point of $\mathcal{P}$.
(ii) If $(\bar{x}, \{\bar{y}^h, \bar{y}_0^h\}_{h \in Q^*}$ is an extreme point of $\mathcal{P}$, then $\bar{y}_0^k = 1$ for some $k \in Q^*$ and $\bar{y}_0^h = 0$ for all $h \in Q \setminus \{k\}$; for otherwise $\bar{x}$ is the convex combination of as many points $\frac{\bar{y}^h}{\bar{y}_0^h}$ as there are positive $\bar{y}_0^h$. Hence $(\bar{x}, 1) = (\bar{y}^k, \bar{y}_0^k)$, $(\bar{y}^h, \bar{y}_0^h) = (0, 0)$, $h \in Q \setminus \{k\}$, and $\bar{x}$ is an extreme point of P_k and also of P_Q. □

Next we give an alternative proof of Theorem 2.1 via projection. This will show the way to obtain the expression of the convex hull in the space of the original variables.

Given a polyhedron of the form

$$S := \{(x, u) \in \mathbb{R}^n \times \mathbb{R}^p : Ax + Bu \leq b\},$$

where A, B and b have m rows, the *projection of S onto the x-space* is defined as

$$\operatorname{Proj}_x(S) := \{x \in \mathbb{R}^n : \exists u \in \mathbb{R}^p : (x, u) \in S\}.$$

Projecting S into the x-space is done by use of the *projection cone*

$$W := \{v \in \mathbb{R}^m : vB = 0, v \geq 0\}$$

associated with $\text{Proj}_x(S)$. Namely,

$$\text{Proj}_x(S) = \{x \in \mathbb{R}^n : (vA)x \le vb, v \in \text{extr } W\} \tag{2.2}$$

where extr W denotes the set of extreme rays of W. (See the next section for a proof of this statement and discussion of its variants).

Second Proof of Theorem 2.1 As before, let P_Q denote the closed convex hull of F, and let $\mathcal{P}$ be the set of vectors satisfying (2.1). We show that P_Q is the projection of $\mathcal{P}$ onto the subspace of x.

The projection cone of the system (2.1) is

$$W := \{w : -\alpha + u^h A^h = 0, \alpha_0 - u^h b^h \le 0, u^h \ge 0, h \in Q^*\}$$

where $w = (\alpha, \alpha_0, \{u^h\}_{h \in Q^*})$ has components α associated with the set of equations involving x, α_0 associated with the last equation, and u^h, $h \in Q^*$, associated with the corresponding inequality sets.

The projection of $\mathcal{P}$ is then the set of those $x \in \mathbb{R}^n$ that satisfy the inequalities of (2.2) obtained by premultiplying the system (2.1) with each extreme ray of W. These inequalities are of the form

$$\left(\alpha \cdot I_n + \sum_{h \in Q^*} \cdot O^n_{m_h}\right) x \ge \alpha_0 \cdot 1 + \sum_{h \in Q^*} u^h \cdot O^h_{m_h} \tag{2.3}$$

where I_n is the identity matrix of order n, and $O^n_{m_h}$ is the $m_h \times n$ zero matrix. Rewriting (2.3) in the simpler form

$$\alpha x \ge \alpha_0, \tag{7a}$$

we notice that x satisfies (2.3) for every extreme ray (direction vector) of W if and only if it satisfies (2.3) for every $w \in W$. But the set of such inequalities $\alpha x \ge \alpha_0$ is the same as the one defined by Theorem 1.2, which is the set of all valid inequalities for the disjunctive set F. Clearly, this set defines the closed convex hull of F. □

In order to use this characterization of the convex hull, one needs to know which P_h are nonempty. This inconvenience is considerably mitigated by the fact, to be shown below, that the information in question becomes irrelevant if the systems $A^h y^h \ge b^h$ satisfy a condition that is often easy to check. Let $(2.1)_Q$ be the constraint set obtained from (2.1) by substituting Q for Q^*, and let $\mathcal{P}_Q$ be the set obtained from $\mathcal{P}$ by the same substitution.

For each $P_h = \{y \in \mathbb{R}^n | A^h y \ge b^h\}$, define the cone $C_h = \{y \in \mathbb{R}^n | A^h y \ge 0\}$. If $P_h \ne \emptyset$, then C_h is the recession cone of P_h, i.e.,

$$C_h = \{y | x + \lambda y \in P_h, \forall x \in P_h, \lambda \ge 0\}.$$

For arbitrary sets $S_h \subset \mathbb{R}^n$, $h \in M$, we denote

$$\sum_{h \in M} S_h = \left\{ x \in \mathbb{R}^n \,\middle|\, x = \sum_{h \in M} y^h \text{ for some } y^h \in S_h, h \in M \right\}.$$

Theorem 2.3 *$\mathcal{P}_Q = \mathcal{P}$ if and only if*

$$C_k \subseteq \sum_{h \in Q^*} C_h, \qquad \forall k \in Q \setminus Q^*. \tag{2.4}$$

Proof For $k \in Q \setminus Q^*$, $A^k y - b^h y_0 \geq 0$, $y_0 \geq 0$ implies $y_0 = 0$. Therefore

$$\mathcal{P}_Q = \mathcal{P} + \sum_{k \in Q \setminus Q^*} C_k$$

and from Theorem 1.2,

$$\mathcal{P}_Q = \operatorname{cl} \operatorname{conv} F + \sum_{k \in Q \setminus Q^*} C_k.$$

But condition (2.4) holds if and only if

$$\left(\sum_{k \in Q \setminus Q^*} C_k \right) \subseteq \operatorname{cl} \operatorname{conv} F$$

hence $\mathcal{P}_Q = \mathcal{P}$ if and only if (2.4) holds. □

Clearly, if F is a union of polytopes, then all recession cones are empty and Theorem 2.3 holds with Q^* replaced by Q. The reason that the presence of unbounded polyhedra creates a complication, is that for some $h \in Q$ we may have a situation where $P_h = \{x : A^h x \geq b^h\} = \emptyset$ but $\{x : A^h x \geq 0\} \neq \emptyset$.

2.1.1 Tightness of the Lifted Representation

Theorem 2.1 represents the convex hull of a union of polyhedra in a space whose dimension grows only linearly with the number of polyhedra in the union. A natural question to ask is whether this representation is best possible. This question is addressed in a recent paper by Conforti et al. [59]. Since the number of constraints in our representation is the same as the number of constraints of the q polyhedra involved in the union, which may all be needed, the question is whether the number of additional variables used in the representation can be reduced. Since

$\text{conv}(P^1 \cup \ldots \cup P^q) = \text{conv}(\text{conv}(P^1 \cup \ldots \cup P^{q-1}) \cup P^q)$, it is sufficient to consider the case $q = 2$. Thus the authors of [59] consider the case of the union of two polyhedra in $\mathbb{R}^n$, P^1 and P^2, defined respectively by f_1 and f_2 constraints, and they ask for the minimum number of additional variables needed to represent $\text{conv}(P^1 \cup P^2)$. Their answer is that this number is $n+1$, which is the number used in our construction when applied to the case $q = 2$. Indeed, our Theorem 2.1 uses, besides the vectors y^1 and y^2 corresponding to the variables of the two polytopes P^1 and P^2, the additional n-vector x, and the two scalar variables y_0^1, y_0^2, which can be replaced by the single variable y_0 in view of the constraint $y_0^1 + y_0^2 = 1$, a total of $n+1$ $(= n+2-1)$ additional variables. In the general case of a union of q polytopes, the minimum number of additional variables needed to represent $\text{conv}(\cup_{h\in Q} P^h)$ is $n + |Q| - 1$. Thus [59] concludes that the formulation of Theorem 2.1 is "best possible".

2.1.2 *From the Convex Hull to the Union Itself*

From Corollary 2.2, all basic solutions of the system (2.1) or $(2.1)_Q$ satisfy the condition $y_0^h \in \{0, 1\}$, $h \in Q$. On the other hand, a solution satisfying this condition need not be basic. It is then natural to ask the question, what do such solutions represent? The next theorem addresses this issue.

We denote by $\mathcal{P}_Q^I$ the set of those $x \in \mathbb{R}^n$ for which there exist vectors $(y^h, y_0^h) \in \mathbb{R}^{n+1}$, $h \in Q$, satisfying the constraints of $(2.1)_Q$ and the condition $y_0^h = 0$ or 1, $h \in Q$; i.e.,

$$\mathcal{P}_Q^I := \{x \in \mathcal{P}_Q | y_0^h \in \{0, 1\}, h \in Q\}.$$

Theorem 2.4 *Let* $F = \bigcup_{h\in Q} P_h$, $Q^* = \{h \in Q | P_h \neq \emptyset\}$, *and* $Q^{**} = \{h \in Q^* | P_h \not\subseteq P_j \forall j \in Q^* \setminus \{h\}\}$. *If F satisfies*

$$C_h = C_j \quad \forall h, j \in Q^{**} \tag{2.5}$$

and

$$C_k \subseteq C_h \quad \forall k \in Q \setminus Q^*, h \in Q^{**} \tag{2.6}$$

then

$$X(\mathcal{P}_Q^I) = F.$$

Proof With or without (2.5) and (2.6), $\mathcal{P}_Q^I \supseteq F$. Indeed, if $x \in P_h$ for some $h \in Q$, then x together with the vectors $(y^h, y_0^h) = (x, 1)$, $(y^k, y_0^k) = (0, 0)$, $k \in Q \setminus \{h\}$, satisfies the constraints defining $\mathcal{P}_Q^I$. It remains to be shown that if (2.5) and (2.6) hold, $\mathcal{P}_Q^I \subseteq F$.

Suppose (2.5) and (2.6) are satisfied and let $x \in \mathcal{P}_Q^I$. Then there exists $k \in Q^{**}$, $Q' \subseteq Q^{**}$ and $Q'' \subseteq Q \setminus Q^*$, such that

$$x = y^k + \sum_{h \in Q' \cup Q''} y^h,$$

and x together with the vectors $(y^k, 1)$, $(y^h, 0)$, $h \in Q' \cup Q''$, and $(y^j, y_0^j) = (0, 0)$, $j \in Q \setminus Q' \cup Q'' \cup \{k\}$, satisfies $(2.1)_Q$. But then $y^k \in P_k$ and $y^h \in C_k$ for $h \in Q'$ (from (2.5)) and for $h \in Q''$ (from (2.6)). Thus $x \in P_k$. □

In other words, if F satisfies (2.5), (2.6), and the variables $y_0^h \in [0, 1]$ are restricted to $\{0, 1\}$, $h \in Q$, then $(2.1)_Q$ represents not $\operatorname{cl\,conv} F$, but F itself. This raises the following question. What exactly is the relationship between integrality conditions and disjunctions? When can integrality conditions be represented as disjunctions, and conversely, when can disjunctions be represented via integrality conditions?

The answer to the first question is pretty straightforward: The condition "x is integer" can be expressed as a disjunction with a finite number of terms if and only if the domain of x is bounded. Indeed, "$L \leq x \leq U$, x integer" can be expressed as

$$x = L \vee x = L + 1 \vee \ldots \vee x = U - 1 \vee x = U,$$

whereas in the absence of the bounds the disjunction would have infinitely many terms.

On the other hand, the answer to the second question is contained in the last theorem. Namely, a disjunctive set can be represented as the constraint set of an integer program if and only if, when expressed in disjunctive normal form, it is a union of polyhedra whose recession cones are all equal to each other (Jeroslow and Lowe [86]).

To illustrate this last condition, consider the disjunction

$$\begin{pmatrix} x_1 = -1 \\ x_2 \geq 0 \end{pmatrix} \vee \begin{pmatrix} x_1 = 1 \\ x_2 \geq 0 \end{pmatrix}.$$

Since the recession cone of both polyhedra is $x_2 \geq 0$, i.e. the same, the integer programming representation of the disjunction is

$$\begin{array}{rrrrrrl} x_1 & -x_{11} & & -x_{12} & & & = 0 \\ x_2 & & -x_{21} & & -x_{22} & & = 0 \\ & x_{11} & & +1\,y_0^1 & & & = 0 \\ & & x_{21} & -0\,y_0^1 & & & \geq 0 \\ & & & x_{12} & & -1\,y_0^2 & = 0 \\ & & & & x_{22} & -0\,y_0^2 & \geq 0 \\ & & & y_0^1 & & +\;y_0^2 & = 1 \\ & & & & & y_0^1, y_0^2 & \in \{0, 1\} \end{array} \tag{2.7}$$

Adding the first, third, fifth and seventh equations yields

$$x^1 + 2y_0^1 = 1;$$

adding the second, fourth and sixth constraints gives $x_2 \geq 0$, so (2.7) can be reduced to

$$\begin{aligned} &x_1 + 2y_0^1 = 1 \\ &x_2 \geq 0 \\ &y_0^1 \in \{0, 1\}, \end{aligned}$$

which is equivalent to the above disjunction: either $y_0^1 = 0$ and $x^1 = 1$, $x_2 \geq 0$, or $y_0^1 = 1$ and $x_1 = -1$, $x_2 \geq 0$.

But now consider a disjunction

$$\begin{pmatrix} x_1 \leq -1 \\ x_2 \geq 0 \end{pmatrix} \vee \begin{pmatrix} x_1 \geq 1 \\ x_2 \geq 0 \end{pmatrix}.$$

This time the recession cones of the two polyhedra are different, and consequently our attempted integer programming representation akin to the above one, which would be obtained from (2.7) by replacing the third and fifth equations with the inequalities $\leq$ and $\geq$, respectively, is invalid. To see why this is so, if $y_0^1 = 1$ and $y_0^2 = 0$, then

$$x_{11} \leq -1, \quad x_{21} \geq 0, \quad x_{12} \geq 0, \quad x_{22} \geq 0$$

and if $y_0^1 = 0$ and $y_0^2 = 1$, then

$$x_{11} \leq 0, \quad x_{21} \geq 0, \quad x_{12} \geq 1, \quad x_{22} \geq 1;$$

in either case, $x_1 = x_{11} + x_{12}$ is undetermined.

For a more detailed treatment of the integer programming representability of unions of polyhedra see [86].

It is worth stopping for a moment to compare this representation of the disjunctive set F with its standard mixed integer programming representation, sometimes called the big M-representation, as the set of those $x \in \mathbb{R}^n$ satisfying

$$\begin{aligned} A^h x \geq b^h - M_h(1 - \delta_h), \quad & h \in Q \\ \sum_{h \in Q} \delta_h = 1 & \\ \delta_h \in \{0, 1\}. & \end{aligned} \tag{2.8}$$

Here M_h is a sufficiently large positive number such that whenever $\delta_h = 0$ for some $h \in Q$, the corresponding inequality is deactivated; and since exactly one of the $|Q|$ variables δ_h is 1, exactly one of the inequalities indexed by Q is imposed.

While the mixed integer programming representation of F as the set of $x \in X(\mathcal{P}_Q^I)$ is tightest possible in the well-defined sense that removing the integrality requirement of the y_0^h, $h \in Q$, yields the closed convex hull of F, which is its strongest possible continuous relaxation, removing the integrality requirement from (2.8) yields in general a much weaker relaxation than the convex hull (an extreme example of this is given in Theorem 4.11).

2.2 Some Facts About Projecting Polyhedra

In this section we explore some properties of projection relevant to its applications to disjunctive programming. Consider a polyhedron of the form (note that the roles of A and B are reversed relative to the previous section)

$$Q := \{(u, x) \in \mathbb{R}^p \times \mathbb{R}^q : Au + Bx \leq b\},$$

where A, B and b have m rows, the *projection of* Q *into* $\mathbb{R}^q$, or *into the* x*-space*, is defined as

$$\text{Proj}_x(Q) := \{x \in \mathbb{R}^q : \exists u \in \mathbb{R}^p : (u, x) \in Q\}.$$

Thus projecting the set Q into $\mathbb{R}^q$ can be viewed as finding a matrix $C \in \mathbb{R}^{m \times q}$ and a vector $d \in \mathbb{R}^m$ such that

$$\mathrm{Proj}_x(Q) = \{x \in \mathbb{R}^q : Cx \leq d\}.$$

This task can be accomplished as follows. Call the polyhedral cone

$$W := \{v : vA = 0, v \geq 0\}$$

the *projection cone* associated with $\mathrm{Proj}_x(Q)$. Then we have

Theorem 2.5

$$\mathrm{Proj}_x(Q) = \{x \in \mathbb{R}^q : (vB)x \leq vb, v \in \mathit{extr}\, W\}$$

where extr W denotes the set of extreme rays of W.

Proof W is a pointed cone (since $W \subseteq \mathbb{R}^m_+$), hence it is the conical hull of its extreme rays. Therefore the system of inequalities defining $\mathrm{Proj}_x(Q)$ holds for all $x \in \mathrm{extr}\, W$ if and only if it holds for all $x \in W$.

Now let $\bar{x} \in \mathrm{Proj}_x(Q)$. Then there exists $\bar{u} \in \mathbb{R}^p$ such that $A\bar{u} + B\bar{x} \leq b$. Premultiplying this system with any $v \in W$, $\bar{x}$ satisfies $(vB)x \leq vb$.

Conversely, suppose $\hat{x} \in \mathbb{R}^q$ satisfies $(vB)x \leq vb$ for all $v \in W$. Then there exists no $v \geq 0$ satisfying $vA = 0$ and $v(B\hat{x} - b) > 0$. But then, by Farkas' Lemma, there exists $\hat{u}$ satisfying $A\hat{u} \leq b - B\hat{x}$, i.e. $\hat{x} \in \mathrm{Proj}_x(Q)$. □

Several comments are in order.

First, if some of the inequalities defining Q are replaced with equations, the corresponding components of v become unrestricted in sign. In such a case, if the projection cone W is not pointed, then it is not the convex hull of its extreme rays. Nevertheless, like any polyhedral cone, W is finitely generated, and any finite set $\widehat{W}$ of its generators can play the role previously assigned to extr W.

Second, the set Q may be defined, besides the system of inequalities involving the variables u to be projected out, by any additional constraints—not necessarily linear—not involving u: such constraints will become part of the projection, without any change. In other words, for an arbitrary set $S \subseteq \mathbb{R}^q$, the projection of

$$Q := \{(u, x) \in \mathbb{R}^p \times \mathbb{R}^q : Au + Bx \leq b, x \in S\}$$

into $\mathbb{R}^q$ is

$$\mathrm{Proj}_x(Q) := \{x \in \mathbb{R}^q : (vB)x \leq vb \text{ for all } v \in W, x \in S\}$$

where W is the projection cone introduced earlier.

Third, $\mathrm{Proj}_x(Q)$ as given by Theorem 2.5 may have redundant inequalities, even when the latter come only from extreme rays of W; i.e., an extreme ray of W does not necessarily give rise to a facet of $\mathrm{Proj}_x(Q)$. It would be nice to be able to identify the extreme rays of W which give rise to facets of $\text{Proj}_x(Q)$, but unfortunately this in general is not possible: whether an inequality $(vB)x \le vb$ is or is not facet defining for $\text{Proj}_x(Q)$ depends on B and b as well as on v. We will return to this question later.

Fourth, we have the simple fact following from the definitions, that if $x \in \mathbb{R}^q$ is an extreme point of $\text{Proj}_x(Q)$, then there exists $u \in \mathbb{R}^p$ such that (u, x) is an extreme point or an extreme ray of Q. This has the following important consequence:

Proposition 2.6 *If Q is an integral polyhedron, then* $\text{Proj}_x(Q)$ *is an integral polyhedron.*

Proof If $\text{Proj}_x(Q)$ has a fractional vertex x, there exists u such that (u, x) is a vertex or extreme ray of Q. □

2.2.1 Well Known Special Cases

There are two well known algorithms whose iterative steps are special cases of projection:

If the matrix A in the definition of Q has a single column a, i.e., if

$$Q := \{(u_0, x) : au_0 + Bx \le b\},$$

then

$$\text{extr}\, W = \{v^k : a_k = 0\} \cup \{v^{ij} : a_i \cdot a_j < 0\},$$

where

$$v^k := e_k, \quad v^{ij} := a_i e_j - a_j e_i,$$

with e_k the k-th unit vector, and we have one step of *Fourier Elimination* (see e.g. [114]).

If Q is of the form

$$Q := \left\{ (u, x, x_0) \,\middle|\, \begin{array}{ll} -cu - dx + x_0 & \le 0 \\ A'u + B'x & \le b' \end{array} \right\},$$

then

$$W = \{(v_0, v) : -v_0 c + vA' = 0;\, v_0, v \ge 0\},$$

$$\text{Proj}_{(x_0,x)}(Q) := \left\{ (x_0, x) \,\middle|\, \begin{array}{r} x_0 + (vB' - d)x \le vb',\ \forall v : (1, v) \in \text{extr}\, W \\ (vB')x \le vb',\ \forall v : (0, v) \in \text{extr}\, W \end{array} \right\}$$

and we have the constraint set of the *Benders Master Problem* (see [42]).

The results of the next two subsections are from [30]

2.2.2 Dimensional Aspects of Projection

When we have a polyhedral representation of some combinatorial object, we would like it to be nonredundant. This makes us interested in identifying among the inequalities defining the polyhedron, say Q, those which are facet inducing, since the latter provide a minimal representation. Further, when we project Q onto a subspace, we would like to know whether the facets of Q project into facets of the projection. To be able to answer this and similar questions, we need to look at the relationship between the dimension of a polyhedron and the dimension of its projections.

Consider the polyhedron $Q := \{(u, x) \in \mathbb{R}^p \times \mathbb{R}^q : Au + Bx \leq b\} \neq \emptyset$, where A, B and b have m rows, and let us partition (A, B, b) into $(A^=, B^=, b^=)$ and $(A^\leq, B^\leq, b^\leq)$, where $A^=u + B^=x = b^=$ is the *equality subsystem* of Q, the set of equations corresponding to the inequalities satisfied at equality by every $(u, x) \in Q$. Let $r := \text{rank}(A^=, B^=) = \text{rank}(A^=, B^=, b^=)$, where the last equality follows from $Q \neq \emptyset$.

Let $\dim(Q)$ denote the dimension of Q. It is well known that $\dim(Q) = p+q-r$, and that Q is full-dimensional, i.e. $\dim(Q) = p + q$, if and only if the equality subsystem is vacuous. When this is the case, then $\dim(\text{Proj}_x(Q)) = q$, for otherwise $\text{Proj}_x(Q)$ has a nonempty equality subsystem that must also be valid for Q, contrary to the assumption that the latter is full dimensional.

Consider now the general situation, when Q is not necessarily full dimensional. Recall that $r = \text{rank}(A^=, B^=)$, and define $r^* := \text{rank}(A^=)$. Clearly, $0 \leq r^* \leq \min\{r, p\}$. It can then be shown [30] that

Theorem 2.7 $\dim(\text{Proj}_x(Q)) = \dim(Q) - p + r^*$.

It follows that $\dim(\text{Proj}_x(Q)) = \dim(Q)$ if and only if $r^* = p$, i.e., the projection operation is dimension-preserving if and only if the matrix $A^=$ is of full column rank.

2.2.3 When Is the Projection of a Facet a Facet of the Projection?

We now turn to the question of what happens to the facets of Q under projection. Since a facet of a polyhedron is itself a polyhedron, the dimension of its projection can be deduced from Theorem 2.7.

Let $\alpha u + \beta x \leq \beta_0$ be a valid inequality for Q, and suppose

$$F_Q := \{(u, x) \in Q : \alpha u + \beta x = \beta_0\}$$

is a facet of Q. Let $\binom{\alpha}{A}^{=} u + \binom{\beta}{B}^{=} x = \binom{\beta_0}{b}^{=}$ be the equality subsystem of F_Q, and let $r_F := \operatorname{rank}\left(\binom{\alpha}{A}^{=}, \binom{\beta}{B}^{=}\right)$. Notice that $r_F - r = 1$, since by assumption $\dim(F_Q) = \dim(Q) - 1$. Further, denote $r_F^* := \operatorname{rank}\left(\binom{\alpha}{A}^{=}\right)$. We may interpret $r_F^* - r^*$ as the difference between the number of dimensions "lost" in projecting Q (which is $p - r^*$) and in projecting F_Q (which is $p - r_F^*$). From Theorem 2.7 we then have

Corollary 2.8 $\dim(\text{Proj}_x(F_Q)) = \dim(\text{Proj}_x(Q)) - 1 + (r_F^* - r^*)$

Indeed, $\dim(\text{Proj}_x(F_Q)) = \dim(F_Q) - p + r_F^* = \dim(Q) - 1 - p + r_F^*$, and substituting for $\dim(Q)$ its value given by Theorem 2.7 yields the Corollary.

We are now in a position to state what happens to the facets of Q under projection.

Corollary 2.9 *Let F_Q be a facet of Q. Then* $\text{Proj}_x(F_Q)$ *is a facet of* $\text{Proj}_x(Q)$ *if and only if $r_F^* = r^*$.*

Proof Outline [30] From Corollary 2.8 it is clear that $\text{Proj}_x(F_Q)$ has the dimension of a facet of $\text{Proj}_x(Q)$ if and only if $r_F^* = r^*$. However, this by itself does not amount to a proof, unless we can guarantee that $\text{Proj}_x(F_Q)$ is a face of $\text{Proj}_x(Q)$, which is far from obvious: in general, the projection of a face need not be a face of the projection. Indeed, think of a 3-dimensional pyramid, and its projection on its base: neither the top vertex of the pyramid, nor the 3 edges incident with it, become faces of the projection as a result of the operation.

However, if $\text{Proj}_x(F_Q)$ is a set of the form $\{x \in \text{Proj}_x(Q) : \beta x = \beta_0\}$ for some valid inequality $\beta x \leq \beta_0$ for $\text{Proj}_x(Q)$, then clearly $\text{Proj}_x(F_Q)$ is a face of $\text{Proj}_x(Q)$. This is the case in our situation: since F_Q is a facet of Q, it has a defining inequality $\alpha u + \beta x \leq \beta_0$. Now if $r_F^* = r^*$, then there exists a vector λ such that $\alpha = \lambda A^{=}$; hence subtracting $\lambda A^{=} u + \lambda B^{=} x = \lambda b^{=}$ from this defining inequality yields $(\beta - \lambda B^{=})x \leq \beta_0 - \lambda b^{=}$, which can be written as $\beta' x \leq \beta_0'$. Thus $\text{Proj}_x(F_Q) = \{x \in \text{Proj}_x(Q) : \beta' x = \beta_0'\}$, i.e. $\text{Proj}_x(F_Q)$ is a face of $\text{Proj}_x(Q)$; hence a facet. □

Another consequence of Theorem 2.7 which is not hard to prove is this:

Corollary 2.10 *Let $r^* = p$, i.e. let $A^{=}$ be of full column rank, and let* $0 \leq d \leq \dim(Q) - 1$. *Then every d-dimensional face of Q projects into a d-dimensional face of* $\text{Proj}_x(Q)$.

2.3 Projection with a Minimal System of Inequalities

As discussed at the beginning of this section, the inequalities of the system defining $\mathrm{Proj}_x(Q)$ are not necessarily facet inducing, even though they are in 1-1 correspondence with the extreme rays of the projection cone W. In other words, the system $(vB)x \leq vb$, $\forall v \in \mathrm{extr}\, W$, is not necessarily minimal. In fact, this system often contains a large number of redundant inequalities. It has been shown however [14], that if a certain linear transformation is applied to Q, which replaces the coefficient matrix B of x with the identity matrix I, the resulting polyhedron $\tilde{Q}$ has a projection cone $\tilde{W}$ that is pointed, and the projection of $\tilde{Q}$ is of the form

$$\mathrm{Proj}_x(\tilde{Q}) = \{x \in \mathbb{R}^q : vx \leq v_0 \text{ for all } (v, v_0) \in \mathbb{R}^{m+1} \text{ such that } (v, w, v_0) \in \mathrm{extr}\, \tilde{W}\},$$

with $\mathrm{Proj}_x(\tilde{Q}) = \mathrm{Proj}_x(Q)$. Here w is a set of auxiliary variables generated by the above transformation. Furthermore, we have the following

Proposition 2.11 ([14]) *Let* $\mathrm{Proj}_x(Q)$ *be full dimensional. Then the inequality* $vx \leq v_0$ *defines a facet of* $\mathrm{Proj}_x(Q)$ *if and only if* (v, v_0) *is an extreme ray of the cone* $\mathrm{Proj}_{(v,v_0)}(\tilde{W})$.

The linear transformation that takes Q into $\tilde{Q}$ is of low complexity ($O(\max\{m, q\}^3)$, where B is $m \times q$). The need to project the cone $\tilde{W}$ onto the subspace (v, v_0) arises only when B is not of full row rank. In the important case when B is of full row rank, we have the following stronger result:

Corollary 2.12 ([14]) *Let* $\mathrm{Proj}_x(Q)$ *be full dimensional and rank*$(B) = m$. *Then the inequality* $vx \leq v_0$ *defines a facet of* $\mathrm{Proj}_x(Q)$ *if and only if* (v, v_0) *is an extreme ray of* $\tilde{W}$.

In many important cases the matrix B is of the form $B = \binom{I}{0}$, which voids the need for the transformation discussed above and leads to a projection cone whose extreme rays yield facet inducing inequalities for $\mathrm{Proj}_x(Q)$. This is the case encountered, for instance, in the characterization of the perfectly matchable subgraph polytope of a graph [35]; as well as in the projection used in the convex hull characterization of a disjunctive set discussed in the previous section.

2.4 The Convex Hull Via Polarity

Next we give a different derivation of the convex hull of a disjunctive set.

The polar of an arbitrary set $S \subseteq \mathbb{R}^n$ is $S^0 := \{x \in \mathbb{R}^n : xy \leq 1,\ \forall y \in S\}$.

In [6, 10], we defined the *reverse polar* as

$$S^\# := \{x \in \mathbb{R}^n : xy \geq 1,\ y \in S\}.$$

Reverse polars share many properties with ordinary polars. Thus

(a) $S^\#$ is convex
(b) $(\lambda S)^\# = (\frac{1}{\lambda})S^\#$ for any $\lambda > 0$
(c) $S \subseteq T \Rightarrow S^\# \supseteq T^\#$
(d) $(S \cup T)^\# = S^\# \cap T^\#$
(e) If S is polyhedral, so is $S^\#$
(f) $\dim S^\# = n - \operatorname{lin} S^\#$ (where dim and lin stand for dimension and lineality)

The proofs of these properties parallel those of the analogous properties of ordinary polars. Here are some properties of $S^\#$ that differ from those of S^0. The first one is the condition for boundedness. For ordinary polars, S^0 is bounded if and only if $0 \in \operatorname{int} \operatorname{cl} \operatorname{conv} S$. On the other hand, for reverse polars we have

Proposition 2.13 $0 \in \operatorname{cl} \operatorname{conv} S \Leftrightarrow S^\# = \emptyset \Leftrightarrow S^\#$ *is bounded*

Proof If $S^\# \neq \emptyset$, there exists $x \in \mathbb{R}^n$ such that $xy \geq 1$ for all $y \in S$, hence the hyperplane $xy = 1$ separates 0 from $\operatorname{cl} \operatorname{conv} S$, i.e. $0 \notin \operatorname{cl} \operatorname{conv} S$. It follows that if $0 \in \operatorname{cl} \operatorname{conv} S$, then $S^\# = \emptyset$. Also, if x satisfies $xy \geq 1$ for all $y \in S$, then so does λx for all $\lambda > 1$, i.e. $S^\#$ is unbounded. It follows that if $S^\#$ is bounded, then $\operatorname{cl} \operatorname{conv} S$ contains some y such that $xy \not\geq 1$ for any x, i.e. $y = 0$. □

The main difference, however, between ordinary and reverse polars is in terms of involution. For ordinary polars

$$S^{00} = \operatorname{cl} \operatorname{conv} (S \cup \{0\}) \tag{2.9}$$

In other words, if $0 \in \operatorname{cl} \operatorname{conv} S$, then $S^{00} = S$. For reverse polars, on the other hand, we have

Theorem 2.14 *Let* $0 \notin \operatorname{cl} \operatorname{conv} S$. *Then*

$$S^{\#\#} = \operatorname{cl} \operatorname{conv} S + \operatorname{cl} \operatorname{cone} \mathrm{S} \tag{2.10}$$

Proof Let v^i, $i = 1, \ldots, p$ and w^j, $j = 1, \ldots, q$ be the vertices and extreme direction vectors of $\operatorname{cl} \operatorname{conv} S$. The reverse polar $S^\# := \{x \in \mathbb{R}^n : xy \geq 1, \ \ \forall y \in S\}$ of S is nonempty (since $0 \notin \operatorname{cl} \operatorname{conv} S$). Then

$$\begin{aligned} S^{\#\#} &:= \{x \in \mathbb{R}^n : xy \geq 1, \ \ \forall y \in S^\#\} \\ &= \{x \in \mathbb{R}^n : xy \geq 1 \text{ for all } y \text{ such that} \\ &\qquad\qquad\qquad yv^i \geq 1, \quad i = 1, \ldots, p \text{ and} \\ &\qquad\qquad\qquad yw^j \geq 0, \quad j = 1, \ldots, q \qquad \}. \end{aligned}$$

Now from Farkas' Lemma $xy \geq 1$ is a consequence of the system $yu^i \geq 1$, $i = 1, \ldots, p$, $yv^j \geq 0$, $j = 1, \ldots, q$ (consistent, since $S^\# \neq \emptyset$) if and only if there exists a set of $\theta_i \geq 0$, $i = 1, \ldots, p$, and $\sigma_j \geq 0$, $j = 1, \ldots, q$, such that $x = \sum_{i=q}^{p} \theta_i v^i + \sum_{j=1}^{q} \sigma_j w^j$, with $\sum_{i=1}^{p} \theta_i v^i \geq \alpha_0$. Thus $S^{\#\#}$ is the set of those points that are positive combinations of vertices and extreme directions of $\mathrm{cl\,conv}\, S$, as claimed. □

So as an important consequence, from (2.9) and (2.10) we have

Corollary 2.15 *For any $S \subseteq \mathbb{R}^n$ with $0 \notin \mathrm{cl\,conv}\, S$,* $\mathrm{cl\,conv}\, S = S^{00} \cap S^{\#\#}$.

In the case when S is a disjunctive set F, either $0 \notin \mathrm{cl\,conv}\, F$, or else this condition can be easily imposed without affecting the solution to the optimization problem.

Example 1 Consider the disjunctive set

$$F = \left\{ x \in \mathbb{R}^2 \,\middle|\, \begin{array}{rrl} -x_1 & -x_2 & \geq -2 \\ x_1 & & \geq 0 \\ & x_2 & \geq 0 \\ x_1 \geq 1 & \vee\ x_2 & \geq 1 \end{array} \right\}$$

illustrated in Fig. 2.1a. Its reverse polars for $\alpha_0 = 1$ and $\alpha_0 = -1$ are the sets

$$F^{\#}_{(1)} = \left\{ y \in \mathbb{R}^2 \,\middle|\, \begin{array}{rrl} 2y_1 & & \geq 1 \\ & 2y_2 & \geq 1 \\ y_1 & & \geq 1 \\ & y_2 & \geq 1 \end{array} \right\} = \left\{ y \in \mathbb{R}^2 \,\middle|\, \begin{array}{rrl} y_1 & & \geq 1 \\ & y_2 & \geq 1 \end{array} \right\}$$

and

$$F^{\#}_{(-1)} = \left\{ y \in \mathbb{R}^2 \,\middle|\, \begin{array}{rrl} 2y_1 & & \geq -1 \\ & 2y_2 & \geq -1 \\ y_1 & & \geq -1 \\ & y_2 & \geq -1 \end{array} \right\} = \left\{ y \in \mathbb{R}^2 \,\middle|\, \begin{array}{rrl} 2y_1 & & \geq -1 \\ & 2y_2 & \geq -1 \end{array} \right\}$$

shown in Fig. 2.1b, c.

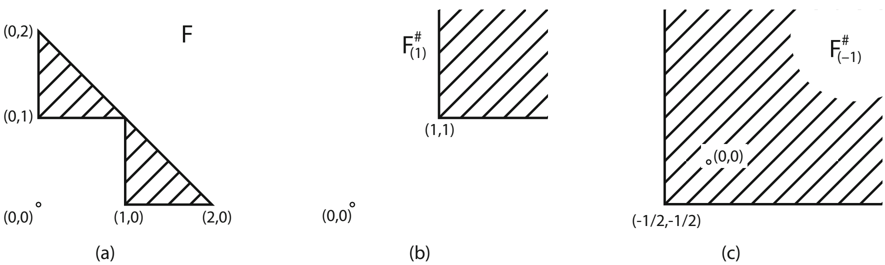


Fig. 2.1 The disjunctive set F and its reverse polars $F^{\#}_{(1)}$ and $F^{\#}_{(-1)}$

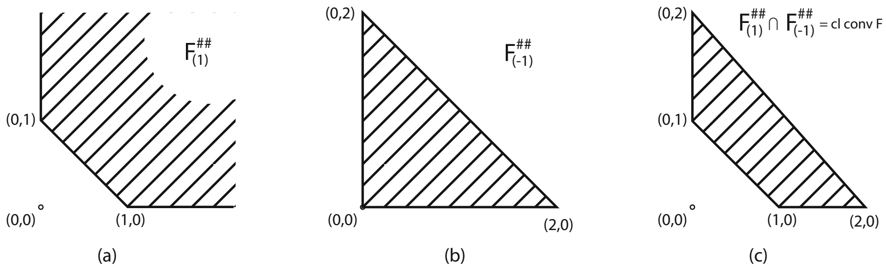


Fig. 2.2 The sets $F^{\#\#}$ corresponding to $\alpha_0 = 1$, $\alpha_0 = -1$, and their intersection

Finally, the sets $F^{\#\#}$ corresponding to $\alpha_0 = 1$ and $\alpha_0 = -1$ (shown in Fig. 2.2a, b) are

$$F^{\#\#}_{(1)} = \left\{ x \in \mathbb{R}^2 \;\middle|\; \begin{array}{l} x_1 + x_2 \geq 1 \\ x_1 \qquad \geq 0 \\ \qquad x_2 \geq 0 \end{array} \right\}, \qquad F^{\#\#}_{(-1)} = \left\{ x \in \mathbb{R}^2 \;\middle|\; \begin{array}{l} -x_1 - x_2 \geq -2 \\ x_1 \qquad \geq 0 \\ \qquad x_2 \geq 0 \end{array} \right\}$$

and their intersection (shown in Fig 2.2c) is

$$F^{\#\#}_{(1)} \cap F^{\#\#}_{(-1)} = \text{cl conv}\, F = \left\{ x \in \mathbb{R}^2 \;\middle|\; \begin{array}{l} -x_1 - x_2 \geq -2 \\ x_1 + x_2 \geq 1 \\ x_1 \qquad \geq 0 \\ \qquad x_2 \geq 0 \end{array} \right\}.$$

The next theorem is needed to prove some other essential properties of scaled polars.

Theorem 2.16

$$F^{\#\#\#}_{(\alpha_0)} = F^{\#}_{(\alpha_0)}$$

Proof If $\alpha_0 \leq 0$, this follows from the corresponding property of ordinary polars. If $\alpha_0 > 0$ and $0 \in \text{cl conv}\, F$, then $F^{\#}_{(\alpha_0)} = \emptyset$, $F^{\#\#}_{(\alpha_0)} = \mathbb{R}^n$, and $F^{\#\#\#}_{(\alpha_0)} = \emptyset = F^{\#}_{(\alpha_0)}$. Finally, if $\alpha_0 > 0$ and $0 \notin \text{cl conv}\, F$, then

$$\begin{aligned} F^{\#\#\#}_{(\alpha_0)} &= (\text{cl conv}\, F + \text{cl cone F})^{\#} \\ &= \{y \in \mathbb{R}^n : xy \geq \alpha_0, \quad \forall x \in (\text{cl conv}\, F + \text{cl cone F})\} \\ &= \{y \in \mathbb{R}^n : xy \geq \alpha_0, \quad \forall x \in \text{cl conv}\, F\} = F^{\#}_{(\alpha_0)}. \end{aligned}$$

□

Next we relate the characterization of $\mathrm{cl\,conv}\, F$ through polarity to its earlier characterization through projection.

Since some of our statements involve both the standard and the reverse polar, we define the *scaled* polar of F with scaling factor α_0 as

$$F^{\#}_{(\alpha_0)} := \{y \in \mathbb{R}^n : xy \geq \alpha_0, \quad \forall x \in F\}.$$

Clearly, it is the sign rather than the value of α_0 that matters in this definition, and the three relevant cases are $\alpha_0 \in \{1, -1, 0\}$. For $\alpha_0 = 1$, $F^{\#}_{(\alpha_0)}$ is the reverse polar as defined above. For $\alpha_0 = -1$, $F^{\#}_{(\alpha_0)} = -F^0$, i.e. $F^{\#}_{(\alpha_0)}$ is the negative of the ordinary polar. Finally, for $\alpha_0 = 0$, $F^{\#}_{(\alpha_0)}$ is the negative of the polar cone C^0, defined for an arbitrary cone C as $C^0 := \{y \in \mathbb{R}^n : xy \leq 0, \quad \forall x \in C\}$.

With this general definition, the scaled polar of $\mathrm{cl\,conv}\, F$ can be written as

$$F^{\#}_{(\alpha_0)} := \left\{ y \in \mathbb{R}^n \;\middle|\; \begin{array}{l} v^i y \geq \alpha_0 \;\; i = 1, \ldots, p \\ w^j y \geq 0 \;\; j = 1, \ldots, q \end{array} \right\}$$

where the v^i and w^j are the extreme points and extreme directions, respectively, of $\mathrm{cl\,conv}\, F$. Now going back to our Theorem 1.2 which we called the Farkas' Lemma for disjunctive sets, the family of inequalities satisfied by all $x \in F$—which is the same as $F^{\#}_{(\alpha_0)}$—consists of those inequalities $\alpha x \geq \alpha_0$ for which there exist vectors $u^h \geq 0$ of appropriate dimensions such that $\alpha = u^h A^h$, $\alpha_0 \leq u^h b^h$, $h \in Q^*$. So from Theorem 1.2 we have an alternative definition of the scaled polar of F, namely

$$F^{\#}_{(\alpha_0)} = \left\{ y \in \mathbb{R}^n \;\middle|\; \begin{array}{l} y - u^h A^h = 0 \\ \qquad u^h b^h \geq \alpha_0 \\ \qquad u^h \geq 0, \quad h \in Q \end{array} \right\}$$

To make the connection even more transparent, consider Theorem 2.1, which defines $\mathrm{cl\,conv}\, F$ as the projection of the higher dimensional system (2.1) into the subspace of the x-variables. The projection cone used for this is

$$W = \{(\alpha, \alpha_0, \{u^h\}_{h \in Q^*}) : \alpha - u^h A^h = 0, \alpha_0 - u^h b^h \leq 0, u^h \geq 0, h \in Q^*\}.$$

Comparing W to $F^{\#}_{(\alpha_0)}$ shows that $F^{\#}_{(\alpha_0)}$ is just the projection W_0 of W onto the subspace of the (α, α_0), $W_0 = \mathrm{Proj}_{(\alpha, \alpha_0)} W$, i.e.

$$W_0 := \{(\alpha, \alpha_0) \in \mathbb{R}^{n+1} : \alpha - u^h A^h = 0,\; \alpha_0 - u^h b^h \leq 0,\; u^h \geq 0,\; h \in Q^*\}.$$

Corollary 2.17 $F^{\#}_{(\alpha_0)} = \{\alpha \in \mathbb{R}^n : (\alpha, \alpha_0) \in W_0\}$.

The above results can be used to characterize the facets of $\mathrm{cl\,conv}\, F$ when the latter is full-dimensional.

Theorem 2.18 *Let* $\dim F = n$*. The inequality* $\alpha x \geq \alpha_0$*, with* $\alpha_0 \neq 0$*, defines a facet of* $\mathrm{cl\,conv}\, F$ *if and only if* (α, α_0) *is an extreme ray of* W_0*.*

Proof From the last Corollary, (α, α_0) is an extreme ray of W_0 if and only if α is a vertex of $F^{\#}_{(\alpha_0)}$. Further,

$$F^{\#}_{(\alpha_0)} = \left\{ y \in \mathbb{R}^n \;\middle|\; \begin{array}{l} vy \geq \alpha_0, \;\; \forall v \in \mathrm{vert}\, F^{\#\#}_{(\alpha_0)} \\ \\ wy \geq 0 \quad \forall w \in \mathrm{dir}\, \mathrm{F}^{\#\#}_{(\alpha_0)} \end{array} \right\},$$

and α is a vertex of $F^{\#}_{(\alpha_0)}$ if and only if it satisfies at equality a nonhomogeneous subset of rank n of the system defining $F^{\#}_{(\alpha_0)}$. We claim that this condition is equivalent to the condition for $\alpha x \geq \alpha_0$ to define a facet of $\mathrm{cl\,conv}\, F$. Indeed, from Theorem 2.14 it is not hard to see that $\alpha x \geq \alpha_0$ defines a facet of $\mathrm{cl\,conv}\, F$ if and only if it defines a facet of $F^{\#\#}_{(\alpha_0)}$. On the other hand, $\alpha x \geq \alpha_0$ defines a facet of $F^{\#\#}_{(\alpha_0)}$ if and only if α is a vertex of $F^{\#}_{(\alpha_0)}$. □

Since the facets of $\mathrm{cl\,conv}\, F$ are vertices (in the nonhomogeneous case) or extreme direction vectors (in the homogeneous case) of the convex polyhedron $F^{\#}_{(\alpha_0)}$, they can be found by maximizing or minimizing some properly chosen linear function on $F^{\#}$, i.e. by solving a linear program of the form

$$\begin{array}{l} \min gy \\ \left.\begin{array}{r} y - \theta^h A^h \geq 0, \\ \theta^h b^h \geq \alpha_0, \\ \theta^h \geq 0 \end{array}\right\} h \in Q^*, \end{array} \qquad P_1^*(g, \alpha_0)$$

or its dual

$$\begin{array}{l} \max \sum_{h \in Q^*} \alpha_0 \xi_0^h, \\ \left.\begin{array}{r} \displaystyle\sum_{h \in Q^*} \xi^h = g, \\ b^h \xi_0^h - A^h \xi^h \leq 0, \\ \xi_0^h \geq 0, \xi^h \geq 0 \end{array}\right\} h \in Q^*. \end{array} \qquad P_2^*(g, \alpha_0)$$

From Proposition 2.13, if $\alpha_0 \leq 0$ then $F^{\#}_{(\alpha_0)} \neq \emptyset$, i.e., $P_1^*(g, \alpha_0)$ is always feasible; and from Theorem 2.14 if $\alpha_0 > 0$, then $P_1^*(g, \alpha_0)$ is feasible if and only if $0 \notin \mathrm{cl\,conv}\, F$. This latter condition expresses the obvious fact that an inequality which cuts off the origin can only be derived from a disjunction which itself cuts off the origin.

Two problems arise in connection with the use of the above linear programs to generate facets of $\text{cl conv } F$. The first one is that sometimes only Q is known, but Q^* is not. This can be taken care of by working with Q rather than Q^*. Let $P_k(g, \alpha_0)$ denote the problem obtained by replacing Q^* with Q in $P_k^*(g, \alpha_0)$, $k = 1, 2$. It was shown in [6], that if $P_2(g, \alpha_0)$ has an optimal solution $\bar{\xi}$ such that

$$(\bar{\xi}_0^h = 0, \bar{\xi}^h \neq 0) \Rightarrow h \in Q^*,$$

then every optimal solution of $P_1(g, \alpha_0)$ is an optimal solution of $P_1^*(g, \alpha_0)$. Thus, one can start by solving $P_2(g, \alpha_0)$. If the above condition is violated for some $h \in Q \setminus Q^*$, then h can be removed from Q and $P_2(g, \alpha_0)$ solved for Q redefined in this way. When necessary, this procedure can be repeated.

The second problem is that, since the facets of $\text{cl conv } F$ of primary interest are the nonhomogeneous ones (in particular those with $\alpha_0 > 0$, since they cut off the origin), one would like to identify the class of vectors g for which $P_1^*(g, \alpha_0)$ has a finite minimum. It was shown in [6], that $P_1^*(g, \alpha_0)$ has a finite minimum if and only if $\lambda g \in \text{cl conv } F$ for some $\lambda > 0$; and that, should g satisfy this condition, $\alpha x \geq \alpha_0$ is a facet of $\text{cl conv } F$ (where F is again assumed full-dimensional) if and only if $\alpha = \bar{y}$ for every optimal solution $(\bar{y}, \bar{\theta})$ to $P_1^*(g, \alpha_0)$.

We now give a numerical example for a facet calculation.

Example 2 Find all those facets of $\text{cl conv } F$ which cut off the origin (i.e., all facets of the form $\alpha x \geq 1$), where $F \subset \mathbb{R}^2$ is the disjunctive set

$$F = F_1 \ \vee \ F_2 \ \vee \ F_3 \ \vee \ F_4,$$

with

$$F_1 = \{x| - x_1 + 2x_2 \geq 6,\ 0 \leq x_1 \leq 1,\ x_2 \geq 0\},$$

$$F_2 = \{x|4x_1 + 2x_2 \geq 11,\ 1 \leq x_1 \leq 2.5,\ x_2 \geq 0\},$$

$$F_3 = \{x| - x_1 + x_2 \geq -2,\ 2.5 \leq x_1 \leq 4,\ x_2 \geq 0\},$$

$$F_4 = \{x|x_1 + x_2 \geq 6,\ 4 \leq x_1 \leq 6,\ x_2 \geq 0\},$$

(see Fig. 2.3).

After removing some redundant constraints, F can be restated as the set of those $x \in \mathbb{R}^2_+$ satisfying

$$\{-x_1 + 2x_2 \geq 6\} \ \vee \ \left\{ \begin{array}{l} 4x_1 + 2x_2 \geq 11 \\ -x_1 + \ x_2 \ \geq -2 \end{array} \right\} \ \vee \ \{x_1 + x_2 \geq 6\}$$

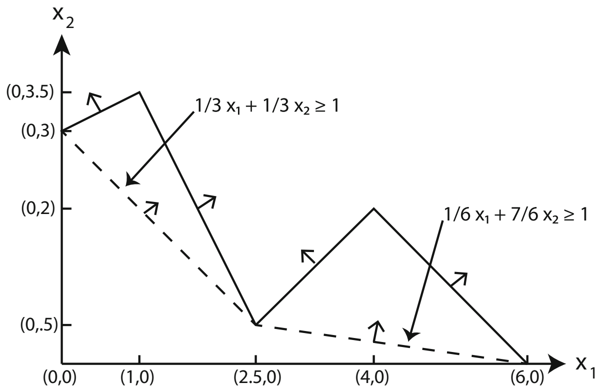


Fig. 2.3 Two facets (in dotted lines) of conv F

and the corresponding problem $P_1(g, 1)$ is

$$
\begin{array}{llllll}
\min g_1 y_1 + g_2 y_2 & & & & & \\
y_1 & & \theta_1^1 & & & \geq 0 \\
& y_2 & -\, 2\theta_1^1 & & & \geq 0 \\
y_1 & & & -\, 4\theta_1^2 + \theta_2^2 & & \geq 0 \\
& y_2 & & -\, 2\theta_1^2 - \theta_2^2 & & \geq 0 \\
y_1 & & & & -\, \theta_1^3 & \geq 0 \\
& y_2 & & & -\, \theta_1^3 & \geq 0 \\
& & 6\theta_1^1 & & & \geq 1 \\
& & & 11\theta_1^2 - 2\theta_2^2 & & \geq 1 \\
& & & & 6\theta_1^3 & \geq 1 \\
& & \multicolumn{4}{l}{\theta_1^1, \theta_1^2, \theta_2^2, \theta_1^3 \geq 0.}
\end{array}
$$

Solving this linear program for $g = (1, 1)$, yields the optimal points

$$(y; \theta) = (\tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{6}, \tfrac{1}{9}, \tfrac{1}{9}, \tfrac{1}{6}), \quad \text{and} \quad (y; \theta) = (\tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{6}, \tfrac{1}{9}, \tfrac{1}{9}, \tfrac{1}{3}),$$

which have the same y-component: $(\frac{1}{3}, \frac{1}{3})$. These points are optimal (and the associated y is unique) for all $g > 0$ such that $g_1 < 5g_2$. For $g_1 = 5g_2$, in addition to the above points, which are still optimal, the points

$$(y; \theta) = (\tfrac{1}{3}, \tfrac{7}{6}, \tfrac{1}{6}, \tfrac{2}{9}, \tfrac{13}{18}, \tfrac{1}{6}) \quad \text{and} \quad (y; \theta) = (\tfrac{1}{6}, \tfrac{7}{6}, \tfrac{7}{12}, \tfrac{2}{9}, \tfrac{13}{18}, \tfrac{1}{6}),$$

which again have the same y-component $y = (\frac{1}{6}, \frac{7}{6})$, also become optimal; and they are the only optimal solutions for all $g > 0$ such that $g_1 > 5g_2$.

We have thus found that the convex hull of F has two facets which cut off the origin, corresponding to the two vertices $y^1 = (\frac{1}{3}, \frac{1}{3})$ and $y^2 = (\frac{1}{6}, \frac{7}{6})$ of $F^{\#}_{(1)}$,

$$\tfrac{1}{3}x_1 + \tfrac{1}{3}x_2 \geq 1,$$

$$\tfrac{1}{6}x_1 + \tfrac{7}{6}x_2 \geq 1.$$

*

The higher dimensional characterization of the convex hull of unions of polyhedra has been generalized by Ceria and Soares [52] to unions of general (i.e. not necessarily polyhedral) convex sets. A slightly narrower generalization, due to Stubbs and Mehrotra [116], characterized the convex hull of mixed 0-1 convex sets. These two results have opened the door to nonlinear disjunctive programming, currently an active research area not covered in this book.

Chapter 3
Sequential Convexification of Disjunctive Sets

In this section we discuss the following problem [6]. Given a disjunctive set in conjunctive normal form

$$\begin{aligned} Ax \geq b, \quad x \geq 0, \\ \bigvee_{i \in Q_j} (d^i x \geq d_0^i), \quad j \in S \end{aligned} \tag{3.1}$$

is it possible to generate the convex hull of feasible points by imposing the disjunctions one by one, at each step calculating a “partial” convex hull, i.e., the convex hull of the set defined by the inequalities generated earlier, plus one of the disjunctions?

For instance, in the case of an integer program, is it possible to generate the convex hull of feasible points by first producing all the facets of the convex hull of points satisfying the linear inequalities, plus the integrality condition on, say, x_1; then adding all these facet-inequalities to the constraint set and generating the facets of the convex hull of points satisfying this amended set of inequalities, plus the integrality condition on x_2; etc.? The question has obvious practical importance, since calculating facets of the convex hull of points satisfying one disjunction is a considerably easier task, as shown in the previous section, than calculating facets of the convex hull of the full disjunctive set.

The answer to the above question is negative in general, but positive for a very important class of disjunctive sets, which we term *facial*. The class includes (pure or mixed) 0-1 programs.

The fact that in the general case the above procedure does not produce the convex hull of the feasible points can be illustrated on the following 2-variable problem.

Example 1 Given the set

$$F_0 = \{x \in \mathbb{R}^2 | -2x_1 + 2x_2 \leq 1, \ \ 2x_1 - 2x_2 \leq 1, \ \ 0 \leq x_1 \leq 2, \ \ 0 \leq x_2 \leq 2\}$$

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_3

find $F = \mathrm{cl\,conv}\,(F_0 \cap \{x|x_1, x_2 \text{ integer}\})$. Denoting

$$F_1 = \mathrm{cl\,conv}\,(F_0 \cap \{x|x_1 \text{ integer}\}), \qquad F_2 = \mathrm{cl\,conv}\,(F_1 \cap \{x|x_2 \text{ integer}\}),$$

the question is whether $F_2 = F$. As shown in Fig. 3.1, the answer is no, since

$$F_2 = \left\{ x \;\middle|\; \begin{array}{r} 2x_1 - x_2 \geq 0 \\ -2x_1 + 3x_2 \geq 0 \\ -2x_1 + x_2 \geq -2 \\ 2x_1 - 3x_2 \geq -2 \end{array} \right\}, \quad \text{while} \quad F = \left\{ x \;\middle|\; \begin{array}{c} -x_1 + x_2 = 0 \\ 0 \leq x_1 \leq 2 \\ 0 \leq x_2 \leq 2 \end{array} \right\},$$

If the order in which the integrality constraints are imposed is reversed, the outcome remains the same.

3.1 Faciality as a Sufficient Condition

Consider a disjunctive program DP with its constraint set stated in the conjunctive normal form (3.1), and denote

$$F_0 = \{x \in \mathbb{R}^n | Ax \geq b, \; x \geq 0\}$$

The disjunctive program (and its constraint set) is called *facial* if every inequality $d^i x \geq d_{i0}$ that appears in a disjunction of (3.1), defines a face of F_0, i.e., if for all $i \in Q_j$, $j \in S$, the set

$$F_0 \cap \{x|d^i x \geq d_{i0}\}$$

is a face of F_0.

Clearly, DP is facial if and only if for every $i \in Q_j$, $j \in S$, $F_0 \subseteq \{x|d^i x \leq d_{i0}\}$.

The class of disjunctive programs that have the facial property includes the most important cases of disjunctive programming, like the 0-1 programming (pure or mixed), nonconvex quadratic programming, separable programming, the linear complementarity problem, etc.; but not the general (pure or mixed) integer programming problem. In all the above mentioned cases the inequalities $d^i x \geq d_{i0}$ of each disjunction actually define not just arbitrary faces, but facets, i.e., $(n-1)$-dimensional faces of F_0, where n is the dimension of F_0.

The fact that integer programs are not sequentially convexifiable is illustrated in Fig. 3.1. The feasible set of a 2-variable integer program is shown in Fig. 3.1a. Imposing integrality on x_1 and then on x_2 results in the feasible sets of Fig. 3.1b, c. But the shaded area of Fig. 3.1c is not the convex hull of the feasible (integer) set, which is shown in Fig. 3.1d.

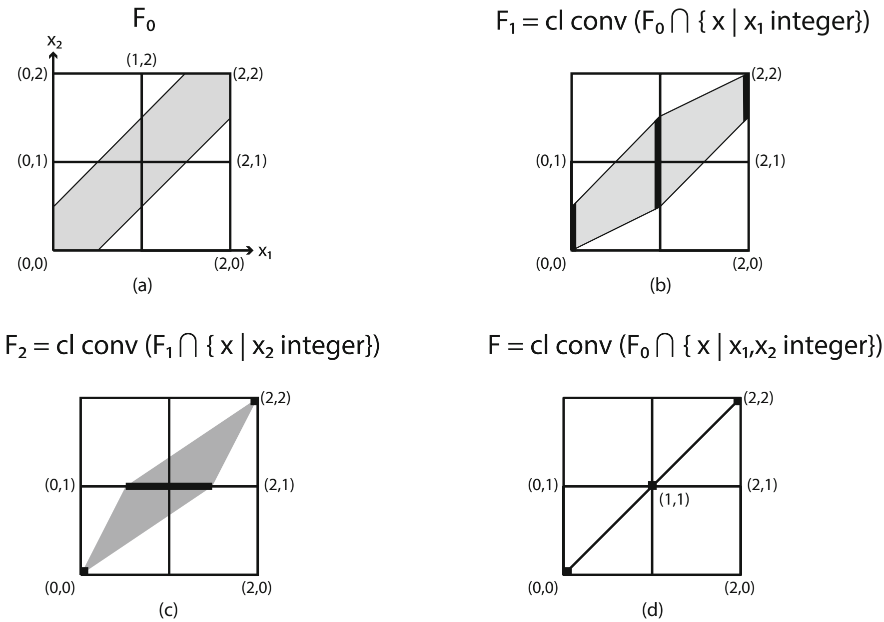


Fig. 3.1 For general integer programs sequential convexification fails

Theorem 3.1 *Let the constraint set of* DP *be represented as*

$$F = \left\{ x \in F_0 \,\middle|\, \bigvee_{i \in Q_j} (d^i x \geq d_{i0},\ j \in S) \right\},$$

where

$$F_0 = \{x \in \mathbb{R}^n | Ax \geq b,\ x \geq 0\},$$

and S, as well as Q_j, $j \in S$, are finite sets.

For an arbitrary ordering of S, define recursively

$$F_j = \text{conv}\left[\bigvee_{i \in Q_j} (F_{j-1} \cap \{x | d^i x \geq d_{i0}\}) \right], \quad j = 1, \ldots, |S|.$$

If F is facial, then

$$F_{|S|} = \text{conv} F.$$

The proof of this theorem [6] uses the following auxiliary result.

Lemma 3.2 *Let $P_1, \ldots, P_r$ be a finite set of polyhedra, and $P = \bigcup_{h=1}^{r} P_h$.*
Let $H^+ = \{x \in \mathbb{R}^n | dx \leq d_0\}$ be an arbitrary halfspace, and $H^- = \{x \in \mathbb{R}^n | dx \geq d_0\}$. If $P \subseteq H^+$, then

$$H^- \cap \operatorname{conv} P = \operatorname{conv}(H^- \cap P).$$

Proof Let $H^- \cap \operatorname{conv} P \neq \emptyset$ (otherwise the Lemma holds trivially). Clearly, $(H^- \cap P) \subseteq (H^- \cap \operatorname{conv} P)$, and therefore

$$\operatorname{conv}(H^- \cap P) \subseteq \operatorname{conv}(H^- \cap \operatorname{conv} P) = H^- \cap \operatorname{conv} P.$$

To prove $\supseteq$, notice that any point in $\operatorname{conv} P$ can be expressed as a convex combination of r points $u_1, \ldots, u_r$, one from each P_h. Thus let $x = \sum_{k=1}^{r} \lambda_k u_k$, with $\sum_{k=1}^{r} \lambda_k = 1, \quad \lambda_k \geq 0, \quad k = 1, \ldots, r$. Further, $P \subseteq H^+$ implies $du_k \leq d_0$, $k = 1, \ldots, r$. We claim that in the above expression for x, if $\lambda_k > 0$ then $du_k \geq d_0$. Indeed, if there exists $\lambda_k > 0$ such that $du_k < d_0$, then

$$\begin{aligned} dx = d\left(\sum_{k=1}^{r} \lambda_k u_k\right) &< d_0 \left(\textstyle\sum_{k=1}^{r} \lambda_k\right) \\ &= d_0, \end{aligned}$$

a contradiction. Hence x is the convex combination of points $u^k \in H^- \cap P$, or $x \in \operatorname{conv}(H^- \cap P)$. □

Another relation to be used in the proof of the theorem, is the fact that for arbitrary $S_1, S_2 \subseteq \mathbb{R}^n$,

$$\operatorname{conv}(\operatorname{conv} S_1 \cup \operatorname{conv} S_2) = \operatorname{conv}(S_1 \cup S_2).$$

Proof of Theorem 3.1 For $j = 1$, the statement is true from the definitions and the obvious relation

$$\bigvee_{i \in Q_j} \left(F_0 \cap \{x | d^i x \geq d_{i0}\}\right) = \left\{ x \in F_0 \,\middle|\, \bigvee_{i \in Q_j} (d^i x \geq d_{i0}) \right\}. \tag{3.2}$$

To prove the theorem by induction on j, suppose the statement is true for $j = 1, \ldots, k$. Then

$$\begin{aligned} F_{k+1} &= \operatorname{conv}\left[\bigvee_{i \in Q_{k+1}} \left(F_k \cap \{x | d^i x \geq d_{i0}\}\right)\right] \text{ (by definition)} \\ &= \operatorname{conv}\left[\bigvee_{i \in Q_{k+1}} \left(\left\{x | d^i x \geq d_{i0}\right\} \cap \operatorname{conv}\left\{ x \in F_0 \,\middle|\, \bigvee_{i \in Q_j} (d^i x \geq d_{i0}, \quad j \in 1, \ldots, k \right\}\right)\right] \end{aligned}$$

(from the assumption)

$$= \text{conv}\left[\bigvee_{i\in Q_{k+1}} \text{conv}\left(\{x|d^i x \geq d_{i0}\} \cap \left\{x \in F_0 \,\middle|\, \bigvee_{i\in Q_j} (d^i x \geq d_{i0}, \quad j = 1, \ldots, k\right\}\right)\right]$$

(from Lemma 3.2, since DP is facial, hence $F_0 \subseteq \{x|d^i x \leq d_{i0}\}$.)

$$= \text{conv}\left[\left(\bigvee_{i\in Q_{k+1}} \{x|d^i x \geq d_{i0}\right) \cap \left\{x \in F_0 \,\middle|\, \bigvee_{i\in Q_j} (d^i x \geq d_{i0}), \quad j = 1, \ldots, k\right\}\right]$$

(from (3.2) and the distributivity of $\cap$ with respect to $\vee$)

$$= \text{conv}\left\{x \in F_0 \,\middle|\, \bigvee_{i\in Q_j} (d^i x \geq d_{i0}), \quad j = 1, \ldots, k+1\right\},$$

i.e., the statement is also true for $j = k + 1$. □

Theorem 3.1 implies that for a facial disjunctive program with feasible set F, the convex hull of F can be generated in $|S|$ stages, (where $|S|$ is the number of elementary disjunctions in the CNF of F), by generating at each stage a "partial" convex hull, namely the convex hull of a disjunctive program with only one elementary disjunction.

In terms of a 0-1 program, for instance, the above result means that the problem

$$\min\{cx|Ax \geq b, \;\; x \geq 0, \;\; x_j = 0 \text{ or } 1, \;\; j = 1, \ldots, n\},$$

is equivalent to (has the same convex hull of its feasible points, as)

$$\min\{cx|Ax \geq b, \;\; x \geq 0, \;\; \alpha^i x \geq \alpha_{i0}, \;\; i \in H_1, \;\; x_j = 0 \text{ or } 1, \;\; j = 2, \ldots, n\} \tag{3.3}$$

where $\alpha^i x \geq \alpha_{i0}$, $i \in H_1$, are the facets of

$$F_1 = \text{conv}\{x|Ax \geq b, \;\; x \geq 0, \;\; x_1 = 0 \text{ or } 1\}.$$

In other words, x_1 is guaranteed to be integer-valued in a solution of (3.3) although the condition $x_1 = 0$ or 1 is not present among the constraints of (3.3). A 0-1 program in n variables can thus be replaced by one in $n - 1$ binary variables at the cost of introducing new linear inequalities. The inequalities themselves are not expensive to generate, since the disjunction that gives rise to them ($x_1 = 0 \vee x_1 = 1$) has only two terms. The difficulty lies rather in the number of facets that one would have to generate, were one to use this approach for solving 0-1 programs.

The sequential convexifiability of 0-1 programs (pure or mixed) is the main property distinguishing this class from the more general class of mixed integer programs. This property makes it possible to generate the convex hull of the feasible set in n stages (where n is the number of 0-1 variables), at each stage generating the convex hull of a two-term disjunctive set, and to rank the cuts obtained in the process according to the stage where they are derived (see more on this in Chap. 6).

*

The property of sequential convexifiability was shown by Stubbs and Mehrotra [116] to also hold for convex (not necessarily linear) mixed 0-1 programs.

3.2 A Necessary Condition for Sequential Convexifiability

Theorem 3.1 states that facial disjunctive sets are sequentially convexifiable. However, faciality is not a necessary condition for sequential convexifiability. In [39] a necessary and sufficient condition is given for sequential convexifiability of a generalized disjunctive set, where the generalization consists in allowing the disjunctions to have infinitely many terms. Such a generalized disjunctive set is termed a reverse convex set, and it can be expressed in the form

$$g_i(x) \geq 0, i = 1, \ldots, q \tag{3.4}$$

where for each i, $g_i(x)$ is a convex function from $\mathbb{R}^n$ to $\mathbb{R}$. The system (3.4) represents the conjunctive normal form of a q-term disjunctive set whose disjunctions may have infinitely many terms. Disjunctive sets whose disjunctions are restricted to finitely many terms are sets of the form (3.4) where each $g_i(x)$ is piecewise linear (in principle it could be just piecewise convex).

The central result of [38] is a necessary and sufficient condition for the sequential convexifiability of a generalized disjunctive set. We will state this condition for the case of disjunctive sets discussed here, dealing with disjunctions whose terms are of the form $D_j := \vee_{i \in Q_j}(d^i x \geq d_{i0})$. The validity of sequential convexification is based on the fact that at the j-th step of the procedure, for all j, we have

$$\text{conv}[(\text{conv} F_{j-1}) \cap D_j] = \text{conv}(F_{j-1} \cap D_j) \tag{3.5}$$

Denoting $\bar{D}_j := \vee_{i \in Q_j}(d^i x \leq d_{i0})$, the result in question is

Theorem 3.3 *F_{j-1} and D_j satisfy (3.5) if and only if they satisfy the constraint boundary condition*

$$(x \in (F_{j-1} \cap \bar{D}_j) \text{ and } y \in (F_{j-1} \cap D_j) \text{ implies } ([x, y] \cap \text{bd}\, \bar{D}_j) \in \text{conv}(F_{j-1} \cap D_j) \tag{3.6}$$

Here $[x, y]$ denotes the line segment between x and y, and bd $\bar{D}_j$ denotes the boundary of $\bar{D}_j$ in the affine space spanned by $\bar{D}_j$. The main content of this Theorem, which is its "if" part, says that in order to ascertain that all points of $(\text{conv}F_{j-1}) \cap D_j$ belong to $\text{conv}(F_{j-1} \cap D_j)$ it is sufficient to check that all those points on the boundary of $\bar{D}_j$ that can be expressed as the convex combination of just two points of F_{j-1} belong to $\text{conv}(F_{j-1} \cap D_j)$. The "only if" part of the theorem states the easily verifiable converse.

Obviously, this necessary and sufficient condition for the sequential convexifiability of a disjunctive set is cumbersome to check, and so its significance is theoretical rather than practical. This is in stark contrast with the sufficient condition given earlier, namely faciality of the disjunctive set, which is straightforward to check and thus of great practical importance.

Chapter 4
Moving Between Conjunctive and Disjunctive Normal Forms

As discussed in Chap. 1 (on inequality systems with logical connectives), disjunctive sets have many equivalent forms, of which the two extremes are the conjunctive normal form (CNF) and the disjunctive normal form (DNF). Although these two normal forms are at the opposite ends of the variety of equivalent forms, they share a property *not* common to all forms: each of them is an *intersection of unions of polyhedra*.

4.1 The Regular Form and Basic Steps

We will say that a disjunctive set that is the intersection of unions of polyhedra is in *regular form* (RF). Thus the RF is

$$F = \bigcap_{j \in T} S_j, \quad S_j = \bigcup_{i \in Q_j} P_i, \tag{4.1}$$

where each P_i is a polyhedron in $\mathbb{R}^n$.

The CNF is the RF in which every S_j is elementary, i.e., every polyhedron P_i is a halfspace. The DNF, on the other hand, is the RF in which $|T| = 1$. Notice that if F is in the RF given by (4.1), each S_j is in DNF. A disjunctive set S_j in DNF will be called *improper* if $S_j = P_i$ for some $i \in Q_j$, *proper* otherwise. Any disjunctive set S_j such that $|Q_j| = 1$ is improper. If S_j is improper then it is convex (and polyhedral).

Next we define an operation which, when applied to a disjunctive set in RF, results in another RF with one less conjunct, i.e., an operation which brings the disjunctive set closer to the DNF. There are several advantages to having a disjunctive set in DNF, i.e., expressed as a union of polyhedra; beyond this, the motivation for the basic step introduced here will become clearer below when we discuss relaxations of disjunctive sets.

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_4

Theorem 4.1 ([12]) *Let F be the disjunctive set in* RF *given by (4.1). Then F can be brought to* DNF *by $|T|-1$ applications of the following basic step, which preserves regularity:*

For some $k, l \in T$, $k \neq l$, bring $S_k \cap S_l$ to DNF*, by replacing it with*

$$S_{kl} = \bigcup_{\substack{i \in Q_k \\ j \in Q_l}} (P_i \cap P_j). \tag{4.2}$$

Proof S_{kl} is the DNF of $S_k \cap S_l$. Indeed, by the distributivity of $\cup$ and $\cap$, we have

$$S_k \cap S_l = \left(\bigcup_{i \in Q_k} P_i \right) \cap \left(\bigcup_{j \in Q_l} P_j \right) = \bigcup_{\substack{j \in Q_k \\ j \in Q_l}} (P_i \cap P_j) = S_{kl}.$$

The set F given by (4.1) is the intersection of $|T|$ unions of polyhedra. Every application of the basic step replaces the intersection of p unions of polyhedra (for some positive integer p) by the intersection of $p-1$ unions of polyhedra. Regularity is thus preserved, and after $|T|-1$ basic steps, F becomes a single union of polyhedra, i.e., is in DNF. □

Remark 4.2 Deleting repetitions, (4.2) can be written as

$$S_{kl} = \left(\bigcup_{i \in Q_k \cap Q_l} P_i \right) \cup \left(\bigcup_{\substack{i \in Q_k \setminus Q_l \\ j \in Q_l \setminus Q_k}} (P_i \cap P_j) \right). \tag{4.2$'$}$$

Remark 4.3 If $S_k = P_{i_0}$ for some $i_0 \in Q_k$, i.e., S_k is improper, then

$$S_{kl} = \begin{cases} P_{i_0} & \text{if } i_0 \in Q_l, \\ \bigcup_{j \in Q_l} (P_{i_0} \cap P_j) & \text{otherwise.} \end{cases} \tag{4.3}$$

Every basic step reduces by one the number of conjuncts S_j in the RF to which it is applied. On the other hand, it is also of interest to know the effect of a basic step on the number of polyhedra whose unions are the conjuncts of the RF. When the basic step is applied to a pair of conjuncts S_k, S_l that are both proper disjunctive sets, namely unions of polyhedra indexed by Q_k and Q_l, respectively, then the set S_{kl} resulting from the basic step is the union of p polyhedra, where

$$p = |Q_k \setminus Q_l| \times |Q_l \setminus Q_k| + |Q_k \cap Q_l|.$$

This is to be compared with the number of polyhedra in the unions defining S_k and S_l, which is $|Q_k| + |Q_l|$. Obviously, more often than not a basic step applied to a pair of proper disjunctive sets results in *an increase in the number of polyhedra* whose union is taken. On the other hand, when one of the two disjunctive sets, say S_k, is improper, then S_{kl} is the union of *at most as many polyhedra* as S_l.

4.2 The Hull Relaxation and the Associated Hierarchy

Given a disjunctive set in CNF with t conjuncts, where the ith conjunct is the union of q_i halfspaces, and given the same disjunctive set in DNF, as the union of q polyhedra, we have the bounding inequality

$$q \leq q_1 \times \cdots \times q_t.$$

Because performing a basic step on a pair S_k, S_l such that S_k is improper, results in a set S_{kl} that is the union of no more polyhedra than is S_l, it is often useful to carry out a parallel basic step, defined as follows:

For F given by (4.1), and $S_k = P_{i_0}$ for some $i_0 \in Q_k$ (i.e., S_k improper), replace $\cap_{j\in T} S_j$ by $\cap_{j\in T\setminus\{k\}} S_{kj}$, where each S_{kj} is defined by (4.3).

Note that if some of the basic steps of Theorem 4.1 are replaced by parallel basic steps, the total number of steps required to bring F to DNF remains the same.

We need one more result before introducing the hierarchy of relaxations of a disjunctive set. Namely, we want to characterize the convex hull of an elementary disjunctive set.

Theorem 4.4 *Let* $F = \cup_{i\in Q} H_i^+ = \{x \in \mathbb{R}^n : \vee_{i\in Q}(a^i x \geq a_{i0})\}$. *Then*

$$\operatorname{cl}\operatorname{conv} F = \begin{cases} \mathbb{R}^n & \text{if } F \text{ is proper,} \\ H_k^+ & \text{if } F \text{ is improper, with } F = H_k^+. \end{cases}$$

Proof If $F = H_k^+$ for some $k \in Q$, $\operatorname{cl}\operatorname{conv} F = H_k^+$ since H_k^+ is closed and convex. Suppose now that F is proper, and let $\bar{x}$ be an arbitrary but fixed point in $\mathbb{R}^n$. From Theorem 2.1, $\bar{x} \in \operatorname{cl}\operatorname{conv} F$ if and only if the system

$$\begin{aligned} &\sum_{i\in Q} y^i = \bar{x}, \\ &a^i y^i - a_{i0} y_0^i \geq 0, \qquad i \in Q, \\ &\sum_{i\in Q} y_0^i = 1, \\ &y_0^i \geq 0, \qquad\qquad\quad i \in Q, \end{aligned}$$

has a solution. From the theorem of the alternative, this is the case if and only if the system

$$\begin{aligned} &-u_0^i a^i + v = 0, \qquad i \in Q, \\ &u_0^i a_{i0} - v_0 \geq 0, \\ &v\bar{x} - v_0 < 0, \\ &u_0^i \geq 0, \qquad\qquad i \in Q, \end{aligned} \tag{4.4}$$

where $u_0^i \in \mathbb{R}$, $i \in Q$, $v_0 \in \mathbb{R}$, and $v \in \mathbb{R}^n$, has no solution.

Since F is proper, there exists no $k \in Q$ such that $H_i^+ \subseteq H_k^+$, $\forall i \in Q$; hence there exist no scalars $u_0^i \geq 0$, $i \in Q$, such that $u_0^i a_0^i = u_0^k a_0^k$, $\forall i \in Q$. Thus (4.4) has no solution for any $\bar{x}$, and hence $\bar{x} \in \operatorname{cl}\operatorname{conv} F$ for all $\bar{x} \in \mathbb{R}^n$, i.e., $\operatorname{cl}\operatorname{conv} F = \mathbb{R}^n$. □

The convex hull of a proper elementary disjunctive set is thus $\mathbb{R}^n$, i.e., replacing such a set with its convex hull is tantamount to throwing away all the constraints that define it. This of course is not true for more general disjunctive sets, as will become clear soon.

The system (2.1) which defines the convex hull of a disjunctive set in DNF is easy to write down, but is unwieldy when the set Q is large; and for a mixed integer program whose feasible set F is expressed as a disjunctive set in DNF, Q tends to be large.

On the other hand, the feasible set of most discrete optimization problems, when given as a disjunctive set in CNF, has conjuncts that are the unions of small numbers of halfspaces, often only two. Performing some basic steps one obtains a set in RF whose conjuncts are still the unions of small numbers of polyhedra. Note that if a disjunctive set is in the RF given by (4.1), each conjunct S_j is in DNF; hence we know how to take its convex hull. Naturally, taking the convex hull of each conjunct is in general not going to deliver the convex hull of the disjunctive set, but can serve as a relaxation of the latter. This takes us to an important class of relaxations.

Given a disjunctive set in regular form

$$F = \bigcap_{j \in T} S_j$$

where each S_j is a union of polyhedra, we define the *hull-relaxation* of F, denoted h-rel F, as

$$\text{h-rel}\, F := \bigcap_{j \in T} \operatorname{cl}\operatorname{conv} S_j .$$

The hull-relaxation of F is not to be confused with the convex hull of F: its usefulness comes precisely from the fact that it involves taking the convex hull of each union of polyhedra *before* intersecting them.

Next we relate the hull-relaxation of a disjunctive set to the usual linear programming relaxation of the feasible set of a mixed integer program. Obviously, the hull-relaxation of any disjunctive set is polyhedral, since the intersection of polyhedra is a polyhedron. Suppose now that we have a disjunctive set in CNF,

$$F_0 = \bigcap_{j \in T} D_j ,$$

where each D_j is an elementary disjunction, i.e., a union of halfspaces. Let $T^* = \{j \in T | D_j \text{ is improper}\}$, $T^{**} = T \setminus T^*$, and denote

$$P_0 = \bigcap_{j \in T^*} D_j,$$

with $P_0 = \mathbb{R}^n$ if $T^* = \emptyset$. By definition, P_0 is a polyhedron; and it can be viewed as the "polyhedral part" of F_0 , i.e., the intersection of those elementary disjunctive sets that are halfspaces. Thus a disjunctive set in CNF can be represented as

$$F_0 = P_0 \cap \left(\bigcap_{j \in T^{**}} D_j \right)$$

where P_0 is a polyhedron and each D_j, $j \in T^{**}$, is a proper elementary disjunctive set.

Lemma 4.5 h-rel$F_0 = P_0$.

Proof

$$\text{h-rel } F_0 = \text{h-rel } \left(P_0 \cap \left(\bigcap_{j \in T^{**}} D_j \right) \right) = \text{cl conv } P_0 \cap \left(\bigcap_{j \in T^{**}} \text{cl conv } D_j \right)$$

by the definition of the hull-relaxation. But cl conv $P_0 = P_0$ and from Theorem 4.4, cl conv $D_j = \mathbb{R}^n$ for all $j \in T^{**}$. This yields the equality stated in the lemma. □

When the feasible set of a (pure or mixed integer) 0-1 program is stated in CNF (which is the usual way of stating it), T^* is the index set of all the conjunctive, i.e., ordinary linear constraints, and T^{**} is the index set of the disjunctions $x_j \leq 0 \vee x_j \geq 1$. Thus P_0 is the linear programming feasible set, and the hull-relaxation of a (pure or mixed-integer) 0-1 program stated in CNF is *identical to the usual linear programming relaxation*.

The next question we address is, what is the hull-relaxation of a disjunctive set that is not in CNF. As we will see, the hull-relaxation of a disjunctive set in RF is weakest when the RF is the CNF. To see this, it suffices to look at the effect of a basic step in the sense of relating the hull-relaxation of the RF before the basic step to that of the RF after the basic step. In particular, of interest is the comparison of taking the convex hull before versus after the basic step.

Lemma 4.6 *For* $j = 1, 2$, *let*

$$S_j = \bigcup_{i \in Q_j} P_i,$$

where each P_i, $i \in Q_j$, $j = 1, 2$, *is a polyhedron. Then*

$$\text{cl conv } (S_1 \cap S_2) \subseteq (\text{cl conv } S_1) \cap (\text{cl conv } S_2). \tag{4.5}$$

Proof Certainly $S_1 \cap S_2 \subseteq (\text{cl conv } S_1) \cap (\text{cl conv } S_2)$, and since $\text{cl conv } (S_1 \cap S_2)$ is the smallest closed convex set to contain $S_1 \cap S_2$, (4.5) follows. □

The next theorem introduces a hierarchy of relaxations that takes a disjunctive set in CNF all the way to its DNF through a sequence of basic steps, each of which results in a tightening of the relaxation.

Theorem 4.7 *For $i = 0, 1, \ldots, t$, let*

$$F_i = \bigcap_{j \in T_i} S_j$$

be a sequence of regular forms of a disjunctive set, such that

(i) F_0 *is in* CNF, *with* $P_0 = \cap_{j \in T_0^*} S_j$*;*
(ii) F_t *is in* DNF*;*
(iii) *for* $i = 1, \ldots, t$, F_i *is obtained from* F_{i-1} *by a basic step.*

Then

$$P_0 = \text{h} - \text{rel} F_0 \supseteq \text{h} - \text{rel} F_1 \supseteq \cdots \supseteq \text{h} - \text{rel} F_t = \text{cl conv } F_t,$$

Proof The first equality holds by Lemma 4.5 since F_0 is in CNF. The last equality holds by the definition of a hull-relaxation, since F_t is in DNF, i.e., $|T_t| = 1$. Each inclusion holds by Lemma 4.6, since for $k = 1, \ldots, t$, F_k is obtained from F_{k-1} by a basic step. □

4.3 When to Convexify a Subset

In order to make best use of the hierarchy of relaxations defined in Theorem 4.7, one would like to know which basic steps result in a strict inclusion of the convex hull taken *after* as opposed to *before* the given basic step. Since taking the convex hull results in an increase in the number of variables, one should take it only when this results in a tighter formulation. The next theorem addresses this question.

Theorem 4.8 *For $j = 1, 2$, let*

$$S_j = \bigcup_{i \in Q_j} P_i,$$

where each P_i, $i \in Q_j$, $j = 1, 2$, is a polyhedron. Then

$$\text{cl conv } (S_1 \cap S_2) = (\text{cl conv } S_1) \cap (\text{cl conv } S_2) \tag{4.6}$$

if and only if every extreme point (extreme direction) of $(\text{cl conv } S_1) \cap (\text{cl conv } S_2)$ *is an extreme point (extreme direction) of $P_i \cap P_k$ for some $(i, k) \in Q_1 \times Q_2$.*

Proof Let T_L and T_R denote the left-hand side and right-hand side, respectively, of (4.6). Then

$$T_L = \operatorname{cl}\operatorname{conv}\left(\bigcup_{\substack{i\in Q_1\\ k\in Q_2}} (P_i \cap P_k)\right).$$

Thus $x \in T_L$ if and only if there exist scalars $\lambda_j \geq 0$, $j \in V$ and $\mu_l \geq 0$, $l \in W$, such that $\sum_{j\in V}\lambda_j = 1$ and

$$x = \sum_{j\in V} v_j\lambda_j + \sum_{l\in W} w_l\mu_l,$$

where V and W are the sets of extreme points and extreme direction vectors, respectively, of the union of all $P_i \cap P_k$, $(i,k) \in Q_1 \times Q_2$.

On the other hand, $x \in T_R$ if and only if there exist scalars $\lambda'_j \geq 0$, $j \in V'$ and $\mu'_l \geq 0$, $l \in W'$, such that $\sum_{j\in V'}\lambda'_j = 1$ and

$$x = \sum_{j\in V'} v'_j\lambda'_j + \sum_{l\in W'} w'_l\mu'_l,$$

where V' and W' are the sets of extreme points and extreme direction vectors, respectively, of T_R. If the condition of the theorem holds, i.e., if $V' \subseteq V$ and $W' \subseteq W$, then $T_R \subseteq T_L$, and since from Lemma 4.6 $T_L \subseteq T_R$, we have $T_L = T_R$ as claimed. If, on the other hand, $V' \setminus V \neq \emptyset$ or $W' \setminus W \neq \emptyset$, then there exists $x \in T_R \setminus T_L$, hence $T_L \subsetneq T_R$. □

One immediate consequence of this theorem is

Corollary 4.9 *Let*

$$K = \{x \in \mathbb{R}^n | 0 \leq x_j \leq 1, \quad j = 1, \ldots, n\},$$

and

$$S_j = \{x \in K | x_j \leq 0 \vee x_j \geq 1\}, \quad j = 1, \ldots, n.$$

Then

$$\operatorname{conv}\bigcap_{j=1}^{n} S_j = \bigcap_{j=1}^{n} \operatorname{conv} S_j.$$

Thus basic steps that replace a set of disjunctive constraints of the form

$$x_j \leq 0 \vee x_j \geq 1, \quad j \in T$$

by a disjunctive constraint of the form

$$\bigvee_{S\subseteq T} (x_j \leq 0, \quad j \in S, \quad x_j \geq 1, \quad j \in T \setminus S)$$

before taking the hull-relaxation, do *not* produce a stronger relaxation: taking the convex hull before or after the execution of such basic steps produces the same result. In order to obtain a stronger hull-relaxation, the basic steps to be performed must involve some other constraints.

Next we illustrate on some examples various situations where taking the convex hull before or after a basic step does make a difference.

Example 1 (Fig. 4.1) Let $P_1 = \{x \in \mathbb{R}^2 | x_1 = 0, 0 \leq x_2 \leq 1\}$, $P_2 = \{x \in \mathbb{R}^2 | x_1 = 1, 0 \leq x_2 \leq 1\}$, $P_3 = \{x \in \mathbb{R}^2 | -x_1 + x_2 \geq 0.5, x_1 \geq 0, x_2 \leq 1\}$, $P_4 = \{x \in \mathbb{R}^2 | x_1 - x_2 \geq 0.5, x_1 \leq 1, x_2 \geq 0\}$, and let $F = S_1 \cap S_2$, with $S_1 = P_1 \cup P_2$, $S_2 = P_3 \cup P_4$. Then

$$\text{cl conv } S_1 = \{x \in \mathbb{R}^2 | 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1\},$$

$$\text{cl conv } S_2 = \{x \in \mathbb{R}^2 | 0.5 \leq x_1 + x_2 \leq 1.5, 0 \leq x_2 \leq 1, 0 \leq x_2 \leq 1\},$$

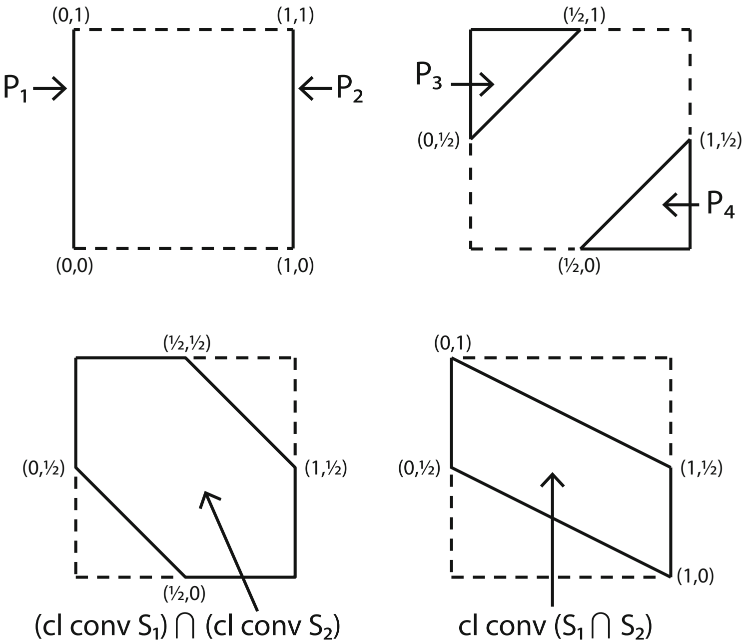


Fig. 4.1 Example 1 where $\text{cl conv}\,(S_1 \cap S_2) \subsetneq (\text{cl conv}\, S_1) \cap (\text{cl conv}\, S_2)$

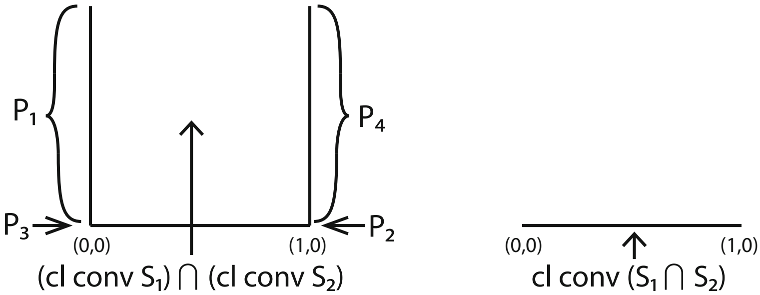


Fig. 4.2 Example 2 where $\mathrm{cl\,conv}\,(S_1 \cap S_2) \subsetneqq (\mathrm{cl\,conv}\,S_1) \cap (\mathrm{cl\,conv}\,S_2)$

and

$$(\mathrm{cl\,conv}\,S_1) \cap (\mathrm{cl\,conv}\,S_2) = \mathrm{cl\,conv}\,S_2.$$

On the other hand, $S_1 \cap S_2 = (P_1 \cup P_3) \cap (P_2 \cup P_4)$ (since $P_1 \cap P_4 = P_2 \cap P_3 = \emptyset$), and

$$\mathrm{cl\,conv}\,(S_1 \cap S_2) = \{x \in \mathbb{R}^2 | 1 \leq x_1 + 2x_2 \leq 2, 0 \leq x_1 \leq 1\}.$$

Here (4.5) holds as strict inclusion, because the vertices (0.5, 0) and (0.5, 1) of $(\mathrm{cl\,conv}\,S_1) \cap (\mathrm{cl\,conv}\,S_2)$ are not vertices of either $P_1 \cap P_3$ or $P_2 \cap P_4$, although the first one is a vertex of P_4, and the second one a vertex of P_3.

Example 2 (Fig. 4.2) Let $P_1 = \{x \in \mathbb{R}^2 | x_1 = 0, x_2 \geq 0\}$, $P_2 = \{x \in \mathbb{R}^2 | x_1 = 1, x_2 = 0\}$, $P_3 = \{x \in \mathbb{R}^2 | x_1 = 0, x_2 = 0\}$, $P_4 = \{x \in \mathbb{R}^2 | x_1 = 1, x_2 \geq 0\}$, and let $F = S_1 \cap S_2$, with $S_1 = P_1 \cup P_2$, $S_2 = P_3 \cup P_4$. Then

$$\begin{aligned} \mathrm{cl\,conv}\,S_1 = \mathrm{cl\,conv}\,S_2 &= \{x \in \mathbb{R}^2 | 0 \leq x_1 \leq 1, x_2 \geq 0\} \\ &= (\mathrm{cl\,conv}\,S_1) \cap (\mathrm{cl\,conv}\,S_2) \end{aligned}$$

whereas

$$\begin{aligned} \mathrm{cl\,conv}\,(S_1 \cap S_2) &= \mathrm{cl\,conv}\,((P_1 \cup P_3) \cap (P_2 \cup P_4)) \\ &= \{x \in \mathbb{R}^2 | 0 \leq x_1 \leq 1, x_2 = 0\}. \end{aligned}$$

Here (4.5) holds as strict inclusion because (0, 1) is an extreme direction vector of $(\mathrm{cl\,conv}\,S_1) \cap (\mathrm{cl\,conv}\,S_2)$, but not of $P_1 \cap P_3$ or $P_2 \cap P_4$.

It is an important practical problem to identify typical situations when it is useful to perform some basic step, i.e., to intersect two conjuncts of a RF before taking their convex hull. The usefulness of such a step can be measured in terms of the gain in strength of the hull-relaxation versus the price one has to pay in terms of the increase in the size of the expression. Since the convex hull of an elementary disjunctive set is $\mathbb{R}^n$, i.e., taking the convex hull of such sets does not constrain the problem at all, one should intersect each elementary disjunctive set S_j in the given RF with some other conjunct S_k before taking the hull-relaxation. This can be done at no cost (in terms of new variables) if S_k is improper. Often intersecting a single improper conjunct S_k with each proper disjunctive set S_j appearing in the same RF before taking the hull-relaxation can substantially strengthen the latter without much increase in problem size. As to which improper conjunct S_k to select, a general principle that one can formulate is that the more restrictive is S_k with respect to each S_j, the better suited it is for the purpose. The next example illustrates this.

Example 3 Consider the 0-1 program

$$\min\{z = -x_1 + 4x_2 | - x_1 + x_2 \geq 0; x_1 + 4x_2 \geq 2; x_1, x_2 \in \{0, 1\}\} \tag{P}$$

illustrated in Fig. 4.2.

The usual linear programming relaxation gives the optimal solution $\bar{x}_1 = \bar{x}_2 = \frac{2}{5}$, with a value of $\bar{z} = \frac{6}{5}$. This of course corresponds to taking the hull-relaxation of the CNF of the feasible set of (P), which contains as conjuncts the improper disjunctive sets corresponding to each of the inequalities of (P) (including $0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1$) and the two proper disjunctive sets $S_1 = \{x \in \mathbb{R}^2 | x_1 \leq 0 \vee x_1 \geq 1\}$, $S_2 = \{x \in \mathbb{R}^2 | x_2 \leq 0 \vee x_2 \geq 1\}$. If P_0 is the intersection of all the improper disjunctive sets, the hull-relaxation of the CNF of (P) is $F_0 = P_0 \cap \text{conv} S_1 \cap \text{conv} S_2$.

Let us write $K = \{x \in \mathbb{R}^2 | 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1\}$, and $P_0 = P_{01} \cap P_{02}$, with $P_{01} = \{x \in K | - x_1 + x_2 \geq 0\}$, $P_{02} = \{x \in K | x_1 + 4x_2 \geq 2\}$. Now suppose we intersect each of S_1 and S_2 with P_{01} before taking the convex hull, i.e., use the hull relaxation $F_1 = P_{02} \cap \text{conv}(P_{01} \cap S_1) \cap \text{conv}(P_{01} \cap S_2)$. We find that $\text{conv}(P_{01} \cap S_1) = \text{conv}(P_{01} \cap S_2) = \{x \in K | - x_1 + x_2 \geq 0\}$, and hence $F_1 = F_0$, i.e., these particular basic steps bring no gain in the strength of the relaxation (see Fig. 4.3).

Suppose instead that we intersect S_1 and S_2 with P_{02} before taking the convex hull, i.e., use the hull relaxation $F_2 = P_{01} \cap \text{conv}(P_{02} \cap S_1) \cap \text{conv}(P_{02} \cap S_2)$. Then

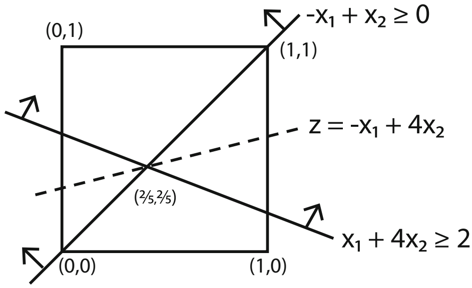


Fig. 4.3 Illustration of Example 3

$\text{conv}(P_{02} \cap S_1) = \{x \in K | x_1 + 4x_2 \geq 2\}$, $\text{conv}(P_{02} \cap S_2) = \{x \in K | x_2 = 1\}$, and $F_2 = \{x \in K | x_2 = 1\}$, which is a stronger relaxation than F_0. Using the relaxation F_2 instead of F_0, i.e., solving $\min\{z = -x_1 + 4x_2 | x \in F_2\}$, yields $\hat{x}_1 = \hat{x}_2 = 1$, with $\hat{z} = 3$, which happens to be the optimal solution of (P).

Note that P_{01} cuts off only one vertex of $\text{conv}(S_1 \cap K) = \text{conv}(S_2 \cap K) = K$, whereas P_{02} cuts off two vertices of K.

4.4 Parsimonious MIP Representation of Disjunctive Sets

In our discussion of the hierarchy of relaxations of a disjunctive set, our guiding criterion in deciding when to perform a basic step and/or when to take a convex hull, was that of getting as tight a linear programming representation as possible. While this criterion certainly makes sense, nevertheless we have to pay attention to the fact that this tighter relaxation usually comes at the price of an increase in the number of polyhedra in the resulting expression,

$$F_i = \bigcap_{j \in T_i} S_j, \quad S_j = \bigcup_{h \in Q_k} P_h \tag{4.7}$$

and hence in the number of 0-1 variables in the integer programming representation of this expression. However, this difficulty can be avoided: the next theorem gives a mixed integer representation of F_i which uses the same number of 0-1 variables as F_0.

Let F_0 be the disjunctive set in CNF consisting of those $x \in \mathbb{R}^n$ satisfying

$$\bigvee_{s \in Q_r} (a^s x \geq b_s), \quad r \in T_0 \tag{4.8}$$

and let F be the same set in RF obtained from F_0 by some sequence of basic steps, given as the set of $x \in \mathbb{R}^n$ satisfying

$$\bigvee_{i \in Q_j} (A^i x \geq b^i), \quad j \in T. \tag{4.9}$$

Then every $j \in T$ corresponds to some subset T_{0j} of T_0 , with $T_0 = \cup_{j \in T} T_{0j}$ such that the disjunction in (4.9) indexed by j is the disjunctive normal form of the set of disjunctions in (4.8) indexed by T_{0j}. In other words, every system $A^i x \geq b^i$, $i \in Q_j$, contains $|T_{0j}|$ inequalities $a^s x \geq a_{s0}$, one from each disjunction $r \in T_{0j}$ of (4.8), and there are as many elements of Q_j (systems $A^i x \geq b^i$, $i \in Q_j$) as there are ways of choosing these inequalities.

Let M_i be the index set of the inequalities $a^s x \geq a_{s0}$ making up the system $A^i x \geq b^i$. From the above, $|M_i| = |T_{0j}|$ for all $i \in Q_j$.

Consider now the mixed integer program with the following constraint set:

$$\begin{aligned}
x - \sum_{i \in Q_j} y^i &= 0, && j \in T, \\
A^i y^i - b^i y_0^i &\geq 0, && i \in Q_j, \quad j \in T, \\
y_0^i &\geq 0, && \\
\sum_{i \in Q_j} y_0^i &= 1, && j \in T,
\end{aligned} \tag{4.10}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$\begin{aligned}
\sum_{i \in Q_j | s \in M_i} y_0^i - \delta_s^r &= 0, && s \in Q_r, \quad r \in T_0, \\
\sum_{s \in Q_r} \delta_s^r &= 1, && r \in T_0, \\
\delta_s^r &\in \{0, 1\}, && s \in Q_r, \quad r \in T_0.
\end{aligned} \tag{4.11}$$

Theorem 4.10 *Assume that F satisfies the conditions of Theorem 2.4. Then the constraint set (4.9) is equivalent to (4.10), (4.11), in that for every solution x to (4.9) there exist vectors (y^i, y_0^i), $i \in Q_j$, $j \in T$ and scalars δ_s^r, $s \in Q_r$, $r \in T_0$, that together with x satisfy (4.10), (4.11); and conversely, the x-component of any solution to (4.10), (4.11) is a solution to (4.9).*

Proof If we write

$$F = \bigcap_{j \in T} S_j, \quad S_j = \bigcup_{i \in Q_j} P_i, \quad P_i = \{x \in \mathbb{R}^n | A^i x \geq b^i\}, \quad i \in Q_j, \quad j \in T,$$

then for every $j \in T$ the j-th subsystem (4.10) represents the system (2.1) of Theorem 2.1, with Q_j of (4.10) corresponding to Q^* of (2.1); and from Theorems 2.1 and 2.3, the set $X(\mathcal{P}^I_{Q_j})$ of Theorem 2.1 defines $\text{cl conv}\, S_j$, $j \in T$. Further, from Theorem 2.4, $S_j = X(\mathcal{P}^I_{Q_j})$, the set defined by (4.10) and the conditions $y_0^i \in \{0, 1\}$, $i \in Q_j$, $j \in T$. Since the set of those $x \in \mathbb{R}^n$ satisfying (4.9) is $F = \cap_{j \in T} X(\mathcal{P}^I_{Q_j})$, it only remains to be shown that the constraints (4.10) enforce the condition $y_0^i \in \{0, 1\}$, $i \in Q_j$, $j \in T$, and do not exclude any solution to (4.9) that satisfies this condition.

Let $\bar{x} \in F$, and let $(\bar{y}^i, \bar{y}_0^i)$, $i \in Q_j$, $j \in T$ (together with $\bar{x}$) satisfy (4.9) with $\bar{y}_0^i \in \{0, 1\}$, $i \in Q_j$, $j \in T$. Then $\bar{y}_0^i = 1$ for exactly one $i \in Q_j$, say $i(j)$, $\bar{y}_0^i = 0$ for $i \in Q_j \setminus \{i(j)\}$, for every $j \in T$. Now for $r \in T_{0j}$, $j \in T$, let

$$\bar{\delta}_s^r = \begin{cases} 1 \text{ if } s \in M_{i(j)}, \\ 0 \text{ if } s \in Q_r \setminus M_{i(j)}. \end{cases}$$

Then clearly

$$\sum_{i \in Q_j | s \in M_i} \bar{y}_0^i = \bar{\delta}_s^r, \quad s \in Q_r, \quad r \in T_0.$$

Further, by construction, each system $A^i x \geq b^i$ contains exactly one inequality $a^s x \geq a_{s0}$ of every disjunction $r \in T_0$ of (4.8), hence

$$\sum_{s \in Q_r} \bar{\delta}_s^r = 1, \quad r \in T_0.$$

Thus for any solution $\{\bar{x}, (\bar{y}^i, \bar{y}_0^i), i \in Q_j, j \in T\}$, to the system (4.10) amended by $y_0^i \in \{0, 1\}, \forall i$, there exists $\bar{\delta}$ which together with $\bar{y}$ satisfies (4.11).

Conversely, let $\{\hat{x}, (\hat{y}^i, \hat{y}_0^i), i \in Q_j, j \in T\}$ be a solution to the system (4.10) such that $0 < \hat{y}_0^k < 1$ for some $k \in Q_j$, $j \in T$. From

$$\sum_{i \in Q_j} \hat{y}_0^i = 1,$$

it then follows that $\hat{y}_0^i < 1$ for all $i \in Q_j$; and since there can be no pair $i_1, i_2 \in Q_j$ such that $s \in M_{i_1} \cap M_{i_2}$ for all $s \in Q_r$, $r \in T_0$, it follows that

$$0 < \sum_{i \in Q_j | s \in M_i} \hat{y}_0^i < 1$$

for some $s \in Q_r$, $r \in T_0$; hence some $\hat{\delta}_s^r$ must be fractional in order for (4.11) to be satisfied. □

Theorem 4.10 provides a way of representing any disjunctive set in regular form as the feasible set of a mixed-integer program with the same number of 0-1 variables as would be required to represent the same disjunctive set in CNF.

4.5 An Illustration: Machine Sequencing Via Disjunctive Graphs

Next we illustrate the concepts and methods discussed above on the example of the following well-known job shop scheduling (machine sequencing) problem: n operations are to be performed on different items using a set of machines, where the duration of operation i is d_i. The objective is to minimize total completion time, subject to (1) precedence constraints between the operations, and (2) the condition that a machine can process *only one item at a time*, and operations *cannot be* *interrupted*. The problem is usually stated [3] as

$$\begin{aligned} &\min t_n \\ &\quad t_j - t_i \geq d_i, && (i, j) \in Z, \\ &\quad t_i \geq 0, && i \in V, \\ &\quad t_j - t_i \geq d_i \ \vee \ t_i - t_j \geq d_j, && (i, j) \in W^+, \end{aligned} \tag{P}$$

where t_i is the starting time of operation i (with n the dummy job "finish"), V is the set of operations, Z the set of pairs constrained by precedence relations, and W^+ the set of pairs that use the same machine and therefore cannot overlap in time. It is often useful to represent the problem by a *disjunctive graph* $G = (V, Z, W)$, with vertex set V and two kinds of directed arc sets: conjunctive (or usual) arcs, indexed by Z, and disjunctive arcs, indexed by W. The set W consists of pairs of disjunctive arcs and is of the form $W = W^+ \cup W^-$, with $(i, j) \in W^+$ if and only if $(j, i) \in W^-$. The subset of nodes corresponding to each machine, together with the disjunctive arcs joining them to each other, forms a *disjunctive clique*. A selection $S \subset W$ consists of exactly one member of each pair of W: i.e., there are 2^q possible selections, where $q = \frac{1}{2}|W|$; G is illustrated in Fig. 4.4, where the disjunctive arcs are shown by dotted lines. If $\mathcal{S}$ denotes the set of selections, for every $S \in \mathcal{S}$, $G_S = (V, Z \cup S)$ is an ordinary directed graph; and the problem (P(S)) obtained from (P) by replacing the set of disjunctive constraints indexed by W^+ with the set of conjunctive constraints indexed by S is the dual of a longest path (critical path) problem in G_S. Thus solving (P) amounts to finding a selection $S \in \mathcal{S}$ that minimizes the length of a critical path in G_S.

The usual mixed integer programming formulation of (P) represents each disjunction

$$t_j - t_i \geq d_i \vee t_i - t_j \geq d_j \tag{4.12}$$

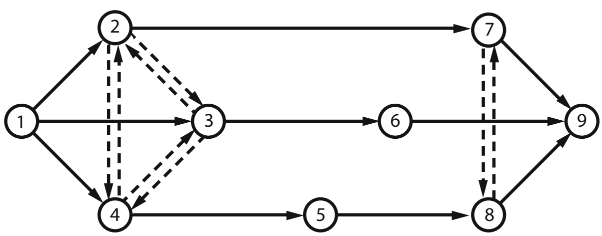


Fig. 4.4 Disjunctive graph with two disjunctive cliques

by the constraint set

$$
\begin{aligned}
&t_j - t_i - (d_i - L_{ij})y_{ij} \geq L_{ij},\\
&-t_j + t_i + (d_j - L_{ji})y_{ij} \geq d_j, \qquad (4.13)\\
&y_{ij} \in \{0, 1\},
\end{aligned}
$$

where L_{ij} is a lower bound on $t_j - t_i$. Unless one wants to use a very crude lower bound L_{ij}, one has to derive lower and upper bounds, L_k and U_k, respectively, on each t_i, $i \in V$, and set $L_{ij} = L_j - U_i$. L_j can be taken to be the length of a longest path from node 1 (the source) to node j in the (conjunctive) graph $G_\emptyset = (V, Z)$, and U_j the difference between the length of a critical path in G_S for some arbitrary selection $S \in \mathcal{S}$, and the length of a longest path from node j to node n (the sink) in $G_\emptyset$.

The constraint set (4.13) accurately represents (4.12) (amended with the bounds $L_k \leq t_k \leq U_k$, $k = 1, 2$), but its linear programming relaxation $(4.13)_L$, obtained by replacing $y_{ij} \in \{0, 1\}$ by $0 \leq y_{ij} \leq 1$, has no constraining power, as shown by the next theorem.

Theorem 4.11 *If the disjunction (4.12) is proper, then every t_i, t_j that satisfies*

$$L_i \leq t_i \leq U_i, \qquad L_j \leq t_j \leq U_j \qquad (4.14)$$

also satisfies $(4.13)_L$.

Proof It suffices to show the four extreme points (L_i, L_j), (L_i, U_j), (U_i, L_j), (U_i, U_j) of the two-dimensional box defined by (4.14) satisfy $(4.13)_L$ for some y_{ij}. We first write $(4.13)_L$ in the form

$$
\begin{aligned}
&(L_j - U_i)(1 - y_{ij}) + d_i y_{ij} \leq t_j - t_i \leq -d_j(1 - y_{ij}) + (U_j - L_i)y_{ij}\\
&0 \leq y_{ij} \leq 1
\end{aligned}
\qquad (4.13)_L
$$

and note that (L_i, U_j) and (L_j, U_i) satisfy (4.13) for $y_{ij} = 1$ and $y_{ij} = 0$, respectively. To show that (L_i, L_j) satisfies $(4.13)_L$ for some y_{ij}, we substitute (L_i, L_j) into $(4.13)_L$ and obtain

$$\frac{d_j - L_i + L_j}{d_j - L_i + U_j} \leq y_{ij} \leq \frac{U_i - L_i}{U_i - L_j + d_i} \qquad (4.15)$$

To see that (4.15) is feasible, note that the right-hand side increases with U_i; so (4.15) is feasible if it is for the smallest admissible value of U_i, which is $L_j + d_j$ (for smaller U_i (4.12) becomes improper). Substituting $L_j + d_j$ for U_i we obtain that (4.15) is feasible whenever $L_i + d_i \leq U_j$, which is a condition for (4.12) to be proper.

An analogous argument shows that (U_i, U_j) satisfies $(4.13)_L$ for some y_{ij}. $\square$

4.5.1 A Disjunctive Programming Formulation

Consider now the mixed integer representation of (4.12) associated with the hull-relaxation of the feasible set of (P). If the latter is given in CNF, as is usually the case, applying the hull-relaxation to this form yields nothing, since the convex hull of the disjunctive set defined by (4.12) is $\mathbb{R}^2$, the space of (t_i, t_j). If we perform a sequence of basic steps of the type defined in Chap. 4 and introduce into each disjunct of (4.12) the lower and upper bounds on t_i and t_j, this replaces every elementary disjunctive set D_{ij} defined by a pair of constraints (4.12), by a disjunctive set

$$S_{ij} = \left\{ (t_i, t_j) \middle| \begin{pmatrix} t_j - t_i \geq d_i \\ L_i \leq t_i \leq U_i \\ L_j \leq t_j \leq U_j \end{pmatrix} \vee \begin{pmatrix} t_i - t_j \geq d_j \\ L_i \leq t_i \leq U_i \\ L_j \leq t_j \leq U_j \end{pmatrix} \right\}.$$

The feasible set of (P) is then of the form

$$F = P_0 \cap \left(\bigcap_{(i,j)\in W^+} S_{ij} \right) \tag{4.16}$$

where P_0 is the polyhedron defined by the inequalities (4.14) and $t_j - t_i \geq d_i$, $(i, j) \in Z$. Further, we have (since all S_{ij} are bounded, $\text{cl conv}\, S_{ij} = \text{conv} S_{ij}$)

$$\text{h-rel } F = P_0 \cap \left(\bigcap_{(i,j)\in W^+} \text{conv} S_{ij} \right),$$

and from Theorem 2.1, the convex hull of S_{ij} is the set of those (t_i, t_j) satisfying the constraints

$$\begin{aligned}
& t_k - t_k^1 - t_k^2 = 0, && k = i, j, \\
& t_j^1 - t_i^1 \geq d_i y_{ij}, && \\
& -t_j^2 + t_i^2 \geq d_j (1 - y_{ij}), && \\
& L_k y_{ij} \leq t_k^1 \leq U_k y_{ij}, && k = i, j, \\
& L_k (1 - y_{ij}) \leq t_k^2 \leq U_k (1 - y_{ij}), && \\
& 0 \leq y_{ij} \leq 1. &&
\end{aligned} \tag{4.17}$$

Also, from Corollary 2.2, the set of those (t_i, t_j) satisfying (4.17) and $y_{ij} \in \{0, 1\}$ is S_{ij}, since both disjuncts of S_{ij} are bounded polyhedra; and thus using (4.17) with

$y_{ij} \in \{0, 1\}$ for all $(i, j) \in W^+$ is a valid mixed integer formulation of (P). This representation uses the same number of 0-1 variables as the usual one, but introduces two new continuous variables, t_k^1, t_k^2, for every original variable t_k, with associated bounding inequalities $L_k y_{ij} \le t_k^1 \le U_k y_{ij}$, $L_k(1 - y_{ij}) \le t_k^2 \le U_k(1 - y_{ij})$. At the price of this increase in the number of variables and constraints, one obtains as the hull-relaxation a linear program whose feasible set is considerably tighter than in the usual formulation, since each constraint set (4.17) defines the convex hull of S_{ij}. It is not hard to see that each of the two points (L_i, L_j) and (U_i, U_j) violates (4.17) unless it is contained in one of the two halfspaces defined by $t_j - t_i \ge d_i$ and $t_i - t_j \ge d_j$.

4.5.2 A Tighter Disjunctive Programming Formulation

Let us now perform some further basic steps on the regular form (4.16) before taking the hull-relaxation. In particular, let us intersect all S_{ij} such that i and j belong to the same disjunctive clique K. If we denote $T(K) := \cap(S_{ij} : i, j \in K, i \ne j)$, and if $|K| = p$, then

$$T(K) = \{t \in \mathbb{R}^p | t_i - t_j \ge d_j \vee t_j - t_i \ge d_i, i, j \in K, i \ne j, L_i \le t_i \le U_i, i \in K\}.$$

Taking the basic steps in question consists of putting $T(K)$ in disjunctive normal form. Let $\langle K \rangle$ denote the subgraph of G induced by K, i.e., the disjunctive clique with node set K. A selection in $\langle K \rangle$, as defined at the beginning of this section, is a set of arcs containing one member of each disjunctive pair. Thus if $\langle K \rangle$ is viewed simply as the complete digraph on K, then a selection is the same thing as a tournament in $\langle K \rangle$. If S_k denotes the kth selection in $\langle K \rangle$ and Q indexes the selections of $\langle K \rangle$, then the DNF of $T(K)$ is $T(K) = \cup_{k \in Q} T_k(K)$, where

$$T_k(K) = \{t \in \mathbb{R}^p | t_j - t_i \ge d_i, (i, j) \in S_k, L_i \le t_i \le U_i, i \in K\}.$$

It is easy to see that if S_k contains a cycle, then $T_k(K) = \emptyset$. Let $Q^* = \{k \in Q | S_k \text{ is acyclic}\}$. Every selection is known to contain a directed Hamilton path, and for acyclic selections this path is unique. Furthermore, every acyclic selection is the transitive closure of its Hamilton path.

Let P_k denote the directed Hamilton path of the acyclic selection S_k; then S_k is the transitive closure of P_k, and the inequalities $t_j - t_i \ge d_i$, $(i, j) \in P_k$, obviously imply the remaining inequalities of $T_k(K)$, corresponding to arcs $(i, j) \in S_k \setminus P_k$. Thus a more economical expression for the DNF of T is $T(K) = \cup_{k \in Q^*} T_k(K)$, with

$$T_k(K) = \{t \in \mathbb{R}^p | t_j - t_i \ge d_i, (i, j) \in P_k, L_i \le t_i \le U_i, i \in K\}.$$

Now let M be the index set of the disjunctive cliques in G, and K_m the node set of the mth such clique. Then the RF obtained from (4.16) by performing the basic steps described above is

$$F = P_0 \cap \left(\bigcap_{m \in M} T(K_m) \right), \tag{4.18}$$

and the hull-relaxation of this form is

$$\text{h-rel}\ F = P_0 \cap \left(\bigcap_{m \in M} \operatorname{conv} T(K_m) \right). \tag{4.19}$$

For $m \in M$, let Q_m^* index the acyclic selections in $\langle K_m \rangle$; and for $k \in Q_m^*$, let S_k^m and P_k^m denote the kth acyclic selection in $\langle K_m \rangle$, and its directed Hamilton path, respectively. Then introducing a continuous variable λ_k^m for every acyclic selection S_k^m and a 0-1 variable y_{ij} for every disjunctive pair of arcs $\{(i, j), (j, i)\}$, and using Theorem 4.10, we obtain the following mixed integer formulation of problem (P) based on the hull-relaxation (4.19):

$$
\begin{aligned}
&\min\ t_n\\
&t_j - t_i \geq d_i, (i, j) \in Z,\\
&t_j - \sum_{k \in Q_m^*} t_j^k = 0,\ j \in K_m, m \in M,\\
&\left.\begin{aligned}
&-t_{j(1,k)}^k + t_{j(2,k)}^k - d_{j(1,k)} \lambda_k^m \geq 0,\\
&\qquad \ddots \qquad \ddots \qquad \vdots \qquad \vdots\\
&- t_{j(p_k-1,k)}^k + t_{j(p_k,k)}^k - d_{j(p_k-1,k)} \lambda_k^m \geq 0,\\
&t_{j(1,k)}^k - t_{j(p_k,k)}^k + (U_{j(p_k,k)} - L_{j(1,k)}) \lambda_k^m \geq 0,
\end{aligned}\right\}\ k \in Q_m^*, m \in M\\
&\sum_{k \in Q_m^*} \lambda_k^m = 1,\ m \in M,\\
&\sum_{k \mid (i,j) \in S_k} \lambda_k^m + y_{ij} = 1,\\
&\sum_{k \mid (i,j) \in S_k} \lambda_k^m - y_{ij} = 0,\ (i, j) \in W^+\\
&t_j,\ t_j^k \geq 0,\ \ \forall j, k,\ \ \lambda_k^m \geq 0,\ \ \forall k, m,\ \ y_{ij} \in \{0, 1\},\ \ (i, j) \in W^+.
\end{aligned}
\tag{$\mathcal{P}$}
$$

Here $j(i, k)$ indexes the i-th task on path P_k^m, and $p_k = |P_k^m| (= |K_m|)$.

Theorem 4.12 *Problem ($\mathcal{P}$) is equivalent to* (P)*: if t is a feasible solution to* (P)*, there exist vectors t^k and scalars λ_k^m, $k \in Q_m^*$, $m \in M$, and a vector y, satisfying the constraints of $\mathcal{P}$; and conversely, if $t, t^k, \lambda_k^m, k \in Q_m^*, m \in M$, and y satisfy the constraints of ($\mathcal{P}$), then t is a feasible solution to* (P).

Table 4.1 Strong LP versus usual LP values

Problem	No. of operations	No. of machines	Value of		
			Usual LP	Strong LP (rounded)	IP
1	7	2	18	26	31
2	14	4	8	11	13
3	14	4	20	26	35
4	17	4	8	10	12
5	17	4	7	8	10
6	17	4	7	10	12
7	17	4	8	9	12

Proof ($\mathcal{P}$) is the representation of (P) given in Theorem 4.10, with the set F as defined in (4.18), and with the upper bounding inequalities $-t_j^k + U_j\lambda_k \geq 0$, $j \in K_m$, replaced by the single inequality $t_{j(1,k)}^k - t_{j(p_k,k)}^k + (U_{j(p_k,k)} - L_{j(1,k)})\lambda_k \geq 0$, for each $m \in M$. The role of the upper bounding inequalities is to force each t_j^k to 0 when $\lambda_k^m = 0$, and the inequality that replaces them in ($\mathcal{P}$) does precisely that: together with the inequalities associated with the arcs of P_k^m, it defines a directed cycle in $\langle K_m \rangle$ and thus $\lambda_k^m = 0$ forces to 0 all t_j^k, $j \in K_m$. □

The linear programming relaxation of ($\mathcal{P}$) is stronger than the linear programming relaxation of the common mixed integer formulation of (P). Computational experience on a few small problems indicates that the value of this stronger linear programming relaxation tends to be considerably higher than that of the usual linear programming relaxation. This is illustrated in Table 4.1 on a small sample of test problems. Problems 1, 2 and 3 are from [3, 103], and [98, p. 138] respectively. Problems $4, \cdots, 7$ were randomly generated with $d_i \in [1, 5]$.

4.6 Disjunctive Programs with Trigger Variables

A union of polyhedra, $\cup_{i \in Q} P^i$, where $P^i = \{x : A^i x \geq b^i\}$, $i \in Q$, expresses the condition that at least one of the systems $A^i x \geq b^i$ must hold. Sometimes one wants to express more complex conditions on a union of polyhedra, like for instance (a) if $A^i x \geq b^i$ for $i = 1, 2$, then $A^i x \geq b^i$ for $i = 3$; or (b) $A^i x \geq b^i$ for $i = 1$ if and only if ($A^i x \geq b^i$ for $i = 2$ and 3, and $A^i x \ngeq b^i$ for $i = 4$. To model such situations, Raman and Grossmann [108] introduced a model that assigns a Boolean variable Y_i to each of the constraint sets $i \in Q$, and uses it to specify the role of the given term in some conditions like (a) or (b) above involving those terms. To be more specific, the i-th term of the disjunction is activated (triggered) if Y_i = true, deactivated if Y_i = false. In a (somewhat simplified version of) this model, the disjunction

$$\bigvee_{i \in Q} (A^i x \geq b^i)$$

is replaced with

$$
\begin{gathered}
\vee \begin{pmatrix} Y_i \\ A^i x \geq b^i \end{pmatrix} \\
\underset{i \in Q}{\veebar} Y_i \\
\Omega_{i \in Q}(Y_i) = \text{ True} \\
Y_i \in \{\text{True, False}\}.
\end{gathered}
$$

Here the second line expresses the condition that exactly one of the Y_i must be true, while the third line expresses a more complex condition on the Y_i, like for instance (a) or (b) above. Grossmann and several of his coauthors used this model fruitfully both in its linear and nonlinear versions (in the latter case the linear constraint sets $A^i x \geq b^i$ are replaced by convex nonlinear ones $G^i(x) \geq h^i$) to handle a variety of problems that arise in process engineering and related aspects of the chemical industry, see for instance [108, 118]. They call this formulation generalized disjunctive programming (GDP), but a few years later Sawaya and Grossmann [113] showed that the GDP model can be reformulated as a disjunctive programming problem (linear if the GDP is linear, nonlinear otherwise). This was to be expected, since the arsenal of disjunctive programming can in principle accommodate any kind of logical conditions. Nevertheless, the GDP model was shown to be useful in formulating many real-world situations. Furthermore, the hull-relaxation procedure described in Chap. 4 was successfully adopted by the above mentioned authors to the GDP model in the pursuit of tighter relaxations.

Chapter 5
Disjunctive Programming and Extended Formulations

The fact that the convex hull of a disjunctive set has a compact representation in a higher dimensional space has spurred interest in finding convenient higher dimensional representations of various combinatorial objects. This general approach is known in the literature as *extended formulation*. More generally, combinatorial optimization problems typically have more than one formulation, and choosing the most convenient one is a nontrivial exercise. The most relevant choice criterion is the strength (tightness) of the linear programming relaxation, since solving the problem typically requires solving repeatedly amended versions of the linear programming relaxation, and the tighter the relaxations the fewer of them have to be solved. Extended formulations that eliminate the need for the integrality conditions, as is the case of the one for the union of polyhedra, are called perfect formulations. Considerable efforts have been spent on finding the smallest possible perfect formulation for various classes of integer and combinatorial problems. Martin [100] shows that a significant class of dynamic programming recursions lead to perfect formulations. Goemans [79] gives an extended formulation of the permutahedron (the convex hull of all permutations of the first n integers) of size $n \log n$ and shows that this is asymptotically the smallest possible perfect formulation. Kaibel and Pashkovich [90] and Kaibel [89] examine interesting symmetry aspects of extended formulations. See also [110, 119].

In a paper that attracted considerable interest (among other things, because it refuted some false but hard to disprove claims concerning the traveling salesman problem), Yannakakis [122] proved that under some symmetry assumptions the traveling salesman polytope cannot have a compact size perfect formulation. This has led to efforts, sometimes successful, towards finding exponential lower bounds on the possible size of extended formulations of other combinatorial polyhedra [74, 111].

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_5

5.1 Comparing the Strength of Different Formulations

The main tool in comparing (and sometimes also in obtaining) different formulations is projection. Typically the comparisons are made between two problem formulations, say P and Q, such that P is expressed in terms of variables $x \in \mathbb{R}^n$, while Q is in terms of variables $(x, y) \in \mathbb{R}^n \times \mathbb{R}^q$. To compare the strength of the two LP relaxations, one has to express both P and Q in terms of the same variables, and so one uses projection to eliminate y and re-express Q in terms of x, i.e., as $\text{Proj}_x(Q)$. One then compares the LP relaxation of this latter set to that of P.

Before we give several examples of how this can be done, we wish to point out that projection can be used to compare two different formulations of the same problem even when they are expressed in completely different (nonoverlapping) sets of variables, provided the two sets are related by an affine transformation. Suppose, for instance, that we have two equivalent formulations of a problem whose feasible sets are P and Q:

$$P := \{x \in \mathbb{R}^n : Ax \le a_0\}, \quad Q := \{y \in \mathbb{R}^q : By \le b_0\}$$

where A is $m \times n$, B is $p \times q$, and

$$x = Ty + c \tag{5.1}$$

for some $T \subset \mathbb{R}^n \times \mathbb{R}^q$ and $c \in \mathbb{R}^n$.

Define

$$Q^+ := \left\{ (x, y) \in \mathbb{R}^{n+q} \,\middle|\, \begin{array}{r} x - Ty = c \\ By \le b_0 \end{array} \right\}$$

and project Q^+ onto $\mathbb{R}^n$. Using the projection cone

$$W := \{(v, w) \in \mathbb{R}^{n+p} : -vT + wB = 0, w \ge 0\},$$

we obtain

$$\text{Proj}_x(Q^+) := \{x \in \mathbb{R}^n : vx \le vc + wb_0, \forall (v, w) \in W\}.$$

Then to compare P with Q, we compare P with $\text{Proj}_x(Q^+)$.

The affine transformation (5.1) can also be used to derive a different analytical comparison between P and Q that does not make use of projection [105].

5.1.1 The Traveling Salesman Problem

Consider the constraint set of the TSP defined on the complete digraph G on $n+1$ nodes in two well known formulations:

Dantzig–Fulkerson–Johnson [68]

$$\begin{aligned}
&\sum(x_{ij} : j = 0, \ldots, n) = 1 \qquad && i = 0, \ldots, n \\
&\sum(x_{ij} : i = 0, \ldots, n) = 1 \qquad && j = 0, \ldots, n \\
&\sum(x_{ij} : i \in S, j \in S) \leq |S| - 1, \quad && S \subseteq \{0, \ldots, n\}, \quad 2 \leq |S| \leq \tfrac{n+1}{2} \\
&x_{ij} \in \{0, 1\}, \quad i, j = 0, \ldots, n
\end{aligned}$$

Miller–Tucker–Zemlin [102]

$$\begin{aligned}
\sum(x_{ij} : j = 0, \ldots, n) = 1 \qquad & i = 0, \ldots, n \\
\sum(x_{ij} : i = 0, \ldots, n) = 1 \qquad & j = 0, \ldots, n \\
u_i - u_j + n x_{ij} \leq n - 1 \quad & i, j = 1, \ldots, n, \quad i \neq j \\
x_{ij} \in \{0, 1\} \quad & i, j = 0, \ldots, n
\end{aligned}$$

The M-T-Z formulation introduces n node variables, but replaces the exponentially large set of subtour elimination constraints by $n(n-1)$ new constraints that achieve the same goal. Thus on the face of it the M-T-Z formulation is considerably more compact. However, projecting the constraint set of the M-T-Z formulation into the subspace of the arc variables by using the cone

$$W = \{v | vA = 0, \quad v \geq 0\}, \tag{5.2}$$

where A is the transpose of the node-arc incidence matrix of G, yields the inequalities

$$\sum(x_{ij} : (i, j) \in C) \leq \frac{n-1}{n} |C|$$

for every directed cycle C of G. These inequalities are strictly weaker than the corresponding subtour elimination inequalities of the D-F-J formulation, so the latter provides a tighter LP relaxation and is therefore the preferable formulation.

5.1.2 The Set Covering Problem

Consider the set covering problem

$$\min\{cx | Ax \geq e, \quad x \in \{0,1\}^n\} \tag{SC}$$

where $e = (1, \ldots, 1)$ and A is a matrix of 0's and 1's. Let M and N index the rows and columns of A, respectively, and

$$M_j = \{i \in M | a_{ij} = 1\}, \; j \in N, \qquad N_i = \{j \in N | a_{ij} = 1\}, \; i \in M.$$

The following uncapacitated plant location problem is known to be equivalent to (SC):

$$\begin{aligned} \min \textstyle\sum(c_j x_j : j \in N) & \\ \textstyle\sum(u_{ij} : j \in N_i) \geq 1, & \quad i \in M \\ -\textstyle\sum(u_{ij} : i \in M_j) + |M_j| x_j \geq 0, & \quad j \in N \\ u_{ij} \geq 0, \; i \in M, \; j \in N; \; x_j \in \{0,1\}, & \; j \in N \end{aligned} \tag{UPL}$$

If we now project this constraint set onto the x-space by using the cone

$$W = \{v | v_i - v_j \leq 0, \; j \in N_i, \; i \in M; \; v_i \geq 0, \; i \in M\},$$

we obtain the inequalities

$$\begin{aligned} \textstyle\sum(|M_j| x_j : j \in N_S) \geq |S|, & \qquad \forall S \subseteq M; \\ x_j \geq 0, & \qquad j \in N \end{aligned}$$

(where $N_S := \cup(N_i : i \in S)$), each of which is dominated (strictly if $S \neq M$) by the sum of the inequalities of (SC) indexed by S. Hence the LP relaxation of the UPL formulation is weaker than that of (SC).

If, on the other hand, we use the so called "strong" (or disaggregated) formulation of the uncapacitated plant location problem, namely

$$\begin{aligned} \min \textstyle\sum(c_j x_j : j \in N) & \\ \textstyle\sum(u_{ij} : j \in N_i) \geq 1 & \;\; i \in M \\ -u_{ij} + x_j \geq 0 & \;\; i \in M, \; j \in N; \\ u_{ij} \geq 0, \; x_j \in \{0,1\}, & \;\; i \in M, \; j \in N; \end{aligned} \tag{UPL$'$}$$

then the projected inequalities include

$$\sum(x_j : j \in N_i) \geq 1, \quad i \in M;$$
$$x_j \geq 0 \quad j \in N$$

thus yielding the same LP relaxation as that of (SC).

5.1.3 Nonlinear 0-1 Programming

Consider the nonlinear inequality

$$\sum_{j \in N} a_j \left(\prod_{i \in Q_j} x_i \right) \leq b,$$
$$x_i \in \{0, 1\}, \quad i \in Q_j, \quad j \in N$$

where $a_j > 0$, $j \in N$, and $\prod$ denotes product.

It is a well known linearization technique due to Fortet [77] to replace this inequality by the system

$$\begin{aligned} \sum_{j \in N} a_j y_j \qquad\qquad &\leq b \\ -y_j + \sum_{i \in Q_j} x_i &\leq |Q_j| - 1 \quad j \in N \\ y_j - \qquad x_i &\leq 0 \qquad\qquad i \in Q_j, \quad j \in N \\ y_j \geq 0, \; x_i \in \{0, 1\}&, \; i \in Q_j, \quad j \in N \end{aligned} \tag{5.3}$$

When the number $n = |N|$ of nonlinear terms is small relative to the number of variables x_i, this is perfectly satisfactory. However, often the number of nonlinear terms is a high-degree polynomial in the number of variables, in which case it is desirable to eliminate the new variables y_j, $j \in N$.

The cone needed for projecting the above constraint set onto the x-space has for every $M \subseteq N$ an extreme direction vector

$$v_M = (1; w_M; 0),$$

where 1 is a scalar, 0 is the zero vector with $\sum(|Q_j| \,:\, j \in N)$ components, and $w_M \in \mathbb{R}^n$ is defined by

$$w_j = \begin{cases} a_j & \text{if } j \in M \\ 0 & \text{if } j \in N - M. \end{cases}$$

The corresponding projected inequalities are

$$\sum_{i \in Q_M} \Big(\sum_{j: i \in Q_j} a_j \Big) x_i \le b - \sum_{j \in M} a_j(|Q_j| - 1), \;\; M \subseteq N, \tag{5.4}$$

where $Q_M := \cup(Q_j \,:\, j \in M)$, which is precisely the linearization of Balas and Mazzola [28], arrived at by other means. Again, whether (5.4) is preferable to (5.3) depends on the ratio between the number of variables versus the number of nonlinear terms.

5.2 Proving the Integrality of Polyhedra

One of the important uses of projection in combinatorial optimization is to prove the integrality of certain polyhedra. It often happens that the LP relaxation of a certain formulation, say P, does not satisfy any of the known sufficient conditions for it to have the integrality property; but there exists a higher dimensional formulation whose LP relaxation, say Q, satisfies such a condition (for the relevant variables). In such a case all we need to do is to show that P is the projection of Q onto the subspace of the relevant variables. It then follows that P is integral.

We will illustrate the procedure on several examples.

5.2.1 Perfectly Matchable Subgraphs of a Bipartite Graph

Let $G = (V, E)$ be a bipartite graph with bipartition $V = V_1 \cup V_2$, let $G(W)$ denote the subgraph induced by $W \subseteq V$, and let X be the set of incidence vectors of vertex sets W such that $G(W)$ has a perfect matching.

The Perfectly Matchable Subgraph (PMS-) polytope of G is then conv(X), the convex hull of X. Its linear characterization [34] can be obtained by projection as follows.

Fact (From the Kőnig-Hall Theorem)

$G(W)$ has a perfect matching if and only if

$$|W \cap V_1| = |W \cap V_2|$$

and for every $S \subseteq W \cap V_1$

$$|S| \leq |N(S)|,$$

where

$$N(S) := \{j \in N \,|(i, j) \in E \text{ for some } i \in S\}.$$

Theorem 5.1 ([34]) *The PMS polytope of the bipartite graph G is defined by the system*

$$\begin{aligned} 0 \leq x_i \leq 1 \quad & i \in V \\ x(V_1) - x(V_2) = 0 & \\ x(S) - x(N(S)) \leq 0 \quad & S \subseteq V_1. \end{aligned} \tag{5.5}$$

If $0 \leq x_i \leq 1$ is replaced by $x_i \in \{0, 1\}$, the theorem is simply a restatement of the Kőnig-Hall condition in terms of incidence vectors. So the proof of the theorem amounts to showing that the polytope defined by (5.5) is integral.

Note that the coefficient matrix of (5.5) is not totally unimodular.

To prove the theorem, we restate the constraint set in terms of vertex and edge variables, x_i and u_{ij}, respectively, i.e., in a higher dimensional space. We then obtain the system

$$\begin{array}{llll} u(i, N(i)) \;\; - x_i & & = 0 & i \in V_1 \\ u(N(j), j) & \;\; - x_j & = 0 & j \in V_2 \\ & x(V_1) - x(V_2) & = 0 & \\ u_{ij} \geq 0,\; (i, j) \in E;\; 0 \leq x_i \leq 1, & & & i \in V \end{array} \tag{5.6}$$

whose coefficient matrix is totally unimodular. Here $u(i, N(i)) := \sum(u_{ij} : j \in N(i))$ and $u(N(j), j) := \sum(u_{ij} : i \in N(j))$. Thus the polyhedron defined by (5.6) is integral, and if it can be shown that its projection is the polyhedron defined by (5.5), this is proof that the latter is also integral. This is indeed the case (see [34] for a proof). Moreover, the projection yields the system (5.5) with the condition $S \subseteq V_1$ replaced by

$$S \subseteq V_1 \text{ such that } G(S \cup N(S)) \text{ and } G((K_1 \setminus S) \cup (K_2 \setminus N(S))) \text{ are connected,}$$

where K is the component of G containing $S \cup N(S)$, and $K_i = K \cap V_i$, $i = 1, 2$.

This in turn allows one to weaken the “if” requirement of the Kőnig-Hall Theorem to “if $|S| \leq |N(S)|$ for all $S \subseteq V_1$ such that $G(S \cup N(S))$ and $G((K_1 \setminus S) \cup (K_2 \setminus N(S)))$ are connected.”

5.2.2 *Assignable Subgraphs of a Digraph*

The well known *Assignment Problem* (AP) asks for assigning n people to n jobs. When represented on a digraph, an assignment, i.e. a solution to AP, consists of a collection of arcs spanning G that forms a node-disjoint union of cycles (a cycle decomposition). A digraph $G = (V, A)$ is *assignable* (admits a cycle decomposition) if the assignment problem on G has a solution. The Assignable Subgraph Polytope of a digraph is the convex hull of incidence vectors of node sets W such that $G(W)$ is assignable.

Let $\deg^+(v)$ and $\deg^-(v)$ denote the outdegree and indegree, respectively, of v and for $S \subset V$, let $\Gamma(S) := \{j \in V | (i, j) \in A \text{ for some } i \in S\}$.

Projection can again be used to prove the following

Theorem 5.2 ([13]) *The Assignable Subgraph Polytope of the digraph G is defined by the system*

$$0 \le x_i \le 1 \quad i \in V$$

$$x(S \setminus \Gamma(S)) - x(\Gamma(S) \setminus S) \le 0, \quad S \subseteq V.$$

5.2.3 *Path Decomposable Subgraphs of an Acyclic Digraph*

An acyclic digraph $G = (V, A)$ with two distinguished nodes, s and t, is said to admit an *s-t path decomposition* if there exists a collection of interior node disjoint s-t paths that cover all the nodes of G. The s-t Path Decomposable Subgraph Polytope of G is then the convex hull of incidence vectors of node sets $W \subseteq V - \{s, t\}$ such that $G(W \cup \{s, t\})$ admits an s-t path decomposition.

For $S \subseteq V$, define

$$\Gamma^*(S) := \begin{cases} (\Gamma(S) \setminus \{t\}) \cup \Gamma(s) & \text{if } t \in \Gamma(S) \\ \Gamma(S) & \text{if } t \notin \Gamma(S) \end{cases}$$

where, as before, $\Gamma(S) = \{j \in V : (i, j) \in A \text{ for some } i \in S\}$. Projection can then be used to prove the following

Theorem 5.3 ([13]) *The s-t Path Decomposable Subgraph Polytope of the acyclic digraph $G = (V, A)$ is defined by the system*

$$0 \le x_i \le 1 \quad i \in V$$

$$x(S \setminus \Gamma^*(S)) - x(\Gamma^*(S) \setminus S) \le 0 \quad S \subseteq V - \{s, t\}$$

5.2.4 *Perfectly Matchable Subgraphs of an Arbitrary Graph*

For an arbitrary undirected graph $G = (V, E)$, the perfectly matchable subgraph (PMS-) polytope, defined as before, can also be characterized by projection, but this is a considerably more arduous task than in the case of a bipartite graph. The difficulty partly stems from the fact that the projection cone in this case is not pointed and thus instead of the extreme rays one has to work with a finite set of generators. On the other hand, an interesting feature of the technique used in this case is that a complete set of generators did not have to be found; it was sufficient to identify a subset of generators that produce all facet defining inequalities of the PMS-polytope.

For all $W \subset V$, let $G(W)$ be the subgraph of G induced by W, let $c(W)$ be the number of components of $G(W)$, and let $N(W)$ be the set of neighbors of W, i.e. $N(W) := \{j \in V \setminus W) : (i, j) \in E$ for some $i \in W\}$.

Theorem 5.4 ([35]) *The PMS polytope of an arbitrary graph $G = (V, E)$ is defined by the system*

$$0 \le x_i \le 1 \quad i \in V$$

$$x(S) - x(N(S)) \le |S| - c(S) \tag{5.7}$$

for all $S \subseteq V$ such that every component of $G(S)$ consists of a single node or else is a nonbipartite graph with an odd number of nodes.

For further details and a proof see [35].

For additional examples of the use of extended formulation to obtain tighter representations of combinatorial optimization problems see the survey [57].

Of course, the possibilities of compactification through extended formulation are not unlimited. An interesting result concerning the limitations one faces when attempting this kind of approach is Yannakakis' theorem [122], which gives a lower bound on the number of variables and constraints needed for any extended formulation of a polytope in terms of its nonnegative rank, a concept specifically defined for this purpose.

Chapter 6
Lift-and-Project Cuts for Mixed 0-1 Programs

Consider now the mixed 0-1 programming problem

$$\min\{cx : Ax \geq b, \quad x \geq 0, \quad x_j \in \{0, 1\}, \quad j = 1, \ldots, p\} \tag{MIP}$$

where $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and all data are rational. The linear programming relaxation of (MIP) is $\min\{cx : x \in P\}$, where

$$\begin{aligned} P &:= \{x \in \mathbb{R}^n : Ax \geq b, \quad x \geq 0, \quad x_j \leq 1, \quad j = 1, \ldots, p\} \\ &= \{x \in \mathbb{R}^n : \tilde{A}x \geq \tilde{b}\} \end{aligned}$$

with $\tilde{A} \in \mathbb{R}^{(m+p+n)\times n}$ and $\tilde{b} \in \mathbb{R}^{m+p+n}$. The feasible solutions to (MIP) are then the points of the disjunctive set

$$D = \{x \in P : (x_j \leq 0 \vee x_j \geq 1), \quad j = 1, \ldots, p\}. \tag{6.1}$$

Since D is facial, its convex hull can be generated sequentially, imposing its disjunctions consecutively, one at a time. In particular, to generate the convex hull of D sequentially, we define for each $j = 1, \ldots, p$,

$$P_{j0} := \{x \in P : x_j = 0\}, \qquad P_{j1} := \{x \in P : x_j = 1\}.$$

where after each iteration P gets updated by adding the inequalities generated during the iteration. To generate the (closed) convex hull $\text{conv}(P_{j0} \cup P_{j1})$, we use its higher dimensional representation given in Theorem 2.1, where it is defined (see (2.1)) as the set of those $x \in \mathbb{R}^n$ for which there exist vectors $y, z \in \mathbb{R}^n$ and

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_6

$y_0, z_0 \in \mathbb{R}$ such that

$$\begin{aligned} x \; - \; y \quad\quad - \; z \quad\quad &= 0 \\ \tilde{A}y - \tilde{b}y_0 \quad\quad &\geq 0 \\ - \; y_j \quad\quad &= 0 \\ \tilde{A}z - \tilde{b}z_0 &\geq 0 \\ z_j \; - \; z_0 &= 0 \\ y_0 \quad\quad + \; z_0 &= 1 \end{aligned} \tag{6.2}$$

Here we have omitted the constraints $y_0 \geq 0$ and $z_0 \geq 0$, present in (2.1), because they are implied by the remaining constraints.

As shown in Sect. 2.1, the projection of (6.2) onto the x-space yields the set of inequalities $\alpha x \geq \beta$ given by Theorem 1.2, which in this case amounts to those defined by $(\alpha, \beta) \in \mathbb{R}^{n+1}$ for which there exist vectors $u, v \in \mathbb{R}^n_+$ and scalars u_0, v_0 satisfying

$$\begin{aligned} \alpha - u\tilde{A} + u_0 e_j \quad\quad\quad\quad &= 0 \\ \alpha \quad\quad\quad\quad - v\tilde{A} - v_0 e_j &= 0 \\ \beta - u\tilde{b} \quad\quad\quad\quad &= 0 \\ \beta \quad\quad\quad\quad - v\tilde{b} \quad - v_0 &= 0 \\ u \geq 0, \quad\quad v \geq 0 & \end{aligned} \tag{6.3}$$

Thus, according to Theorem 1.2, $\text{conv}(P_{j-1,0} \cup P_{j-1,1})$ is defined by the (α, β)-components of solutions to (6.3). Furthermore, according to Theorem 2.18, if D of (6.1) is full-dimensional, then the inequality $\alpha x \geq \beta$ ($\neq 0$) defines a facet of $\text{conv}(P_{j-1,0} \cup P_{j-1,1})$ if and only if (α, β) is an extreme ray of $W_0 = \text{Proj}_{(\alpha,\beta)} W$, where W is the cone defined by (6.3).

6.1 Disjunctive Rank

The sequential convexifiability of D provides a way to classify the inequalities yielding the convex hull of D according to their position in the process of generating it. Namely, denoting $P_0 := P$ and for $j = 1, \ldots, p$, $P_j = \text{conv}(x \in P_{j-1} : x_j \in \{0, 1\})$, one can define the *disjunctive rank* [19] of an inequality $\alpha x \geq \beta$ for a mixed 0-1 program as the smallest integer k for which there exists an ordering of $\{1, \ldots, p\}$ such that $\alpha x \geq \beta$ is valid for P_k. In other words, an inequality is of rank k if it can be obtained by k, but not by fewer than k, applications of the recursive procedure defined above. Clearly, the disjunctive rank of a cutting plane for 0-1 programs is bounded by the number of 0-1 variables. It is known that the number of 0-1 variables is not a valid bound for the Chvàtal rank of an inequality.

The above definition of the disjunctive rank is based on using the disjunctions $x_j \in \{0, 1\}$, $j = 1, \ldots, p$. Tighter bounds can be derived by using stronger disjunctions. For instance, a 0-1 program whose constraints include the generalized upper bounds $\sum(x_j : j \in Q_i) = 1$, $i = 1, \ldots, t$, with $|Q_i| = |Q_j| = q$, $Q_i \cap Q_j = \emptyset$, $i, j \in \{1, \ldots, t\}$, and $|\cup_{i=1}^t Q_i| = p$, can be solved as a disjunctive program with the disjunctions

$$\bigvee_{j \in Q_i} (x_j = 1), \quad i = 1, \ldots, t (= p/q)$$

in which case the disjunctive rank of any cut is bounded by the number $t = p/q$ of generalized upper bounds.

6.2 Fractionality of Intermediate Points

In generating conv$D$ sequentially, suppose

$$P_1 = \text{conv}(P : x_1 \in \{0, 1\})$$

has already been generated, and now we generate

$$P_{1j} = \text{conv}(P_1 : x_j \in \{0, 1\}).$$

By virtue of the sequential convexifiability of 0-1 programs,

$$P_{1j} := \text{conv}\{x \in P : x_1 \in \{0, 1\}, \ \ x_j \in \{0, 1\}\};$$

in other words, by imposing the 0-1 condition on x_j we automatically enforce the condition $x_1 \in \{0, 1\}$ too. But what about the intermediate solutions generated "on the way" from P_1 to P_{1j} for some j? As we add new cutting planes to our linear program, will the resulting solutions satisfy $x_1 \in \{0, 1\}$? In general, this cannot be guaranteed, as illustrated by the two-dimensional example of Fig. 6.1, where a valid cut generated in the process of getting from P_1 to P_{12} produces the fractional solution x^*.

The desired property cannot be maintained even if we restrict ourselves to the use of facet defining cutting planes, as illustrated in Fig. 6.2. Here P_1 is the convex hull of A, D, E, x^1, B, C, x^2, and the cutting plane through the points A, B, C is facet defining for both P_{12} and P_{13}. Yet, this hyperplane intersects the edge $[x^1, x^2]$ in its interior, hence in a point fractional in all three components.

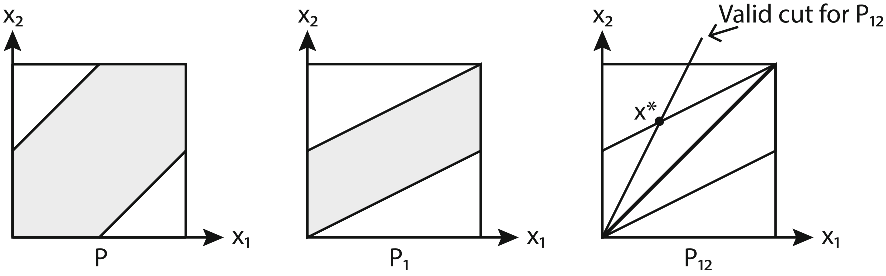


Fig. 6.1 $0 < x_1^* < 1,\ \ 0 < x_2^* < 1$

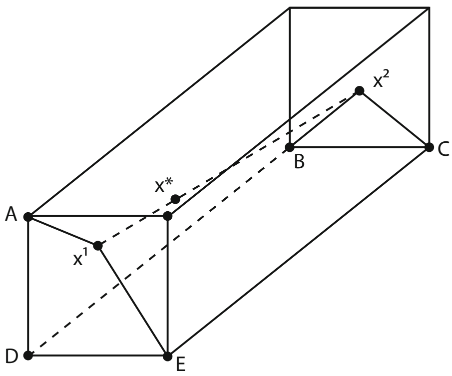


Fig. 6.2 Hyperplane ABC (not shown) intersects edge $[x^1, x^2]$ in a point x^* with $0 < x_j^* < 1,\ \ j = 1, 2, 3$

However, a property almost as strong as the one whose absence we have just illustrated, still holds:

Theorem 6.1 ([32]) *Let P_1 and P_{1j}, $j \in \{2, \dots, n\}$, be as above. Let $\alpha x \geq \beta$ be a valid inequality for P_{1j}, $j \in \{2, \dots, n\}$, and let x^* be an extreme point of $P_1 \cap \{x : \alpha x \geq \beta\}$. Then $0 < x_1^* < 1$ implies $0 < x_j^* < 1$ for any $j \in \{2, \dots, n\}$.*

Proof Suppose $x_j^* \in \{0, 1\}$ for some $j \in \{2, \dots, n\}$; then $x^* \in P_{1j}$. If x^* is an extreme point of P_{1j}, then $x_1^* \in \{0, 1\}$ and we are done. If x^* is not extreme, then $\alpha x^* = \beta$ and there exist points $x^1, x^2 \in P_{1j}$, $x^1 \neq x^2$, such that $x^* = \lambda x^1 + (1 - \lambda)x^2$ for some $0 < \lambda < 1$, and $\alpha x^1 > \beta$. But this contradicts the assumption that $\alpha x \geq \beta$ is valid for P_{1j}. $\square$

6.3 Generating Cuts

In order the generate valid inequalities that cut off by a maximum amount the LP optimal solution, say $\bar{x}$, one wants to minimize $\bar{x}^T\alpha - \beta$ subject to the constraints (6.3). This is the approach taken in [19, 20]. However, since all the constraints of (6.3) are homogeneous, i.e., the polyhedron defined by (6.3) is a cone, this optimization problem has no finite minimum. To turn it into a linear program with a finite optimal solution, the cone defined by (6.3) needs to be truncated. This can be done by amending (6.3) with one or more normalization constraints. Possible normalizations include (a) $-1 \leq \beta \leq 1$; (b) $-1 \leq \alpha_j \leq 1$, $j = 1, \ldots, n$; (c) $\sum_{j=1}^n |\alpha_j| \leq 1$; (d) $\sum_{j=1}^n ||\alpha_j|| \leq 1$; (e) $u_0 + v_0 = 1$; (f) $ue + ve + u_0 + v_0 = 1$ (where e is the summation vector). Using normalization (f), the Cut Generating Linear Program for the disjunction $x_k \leq 0 \vee x_k \geq 1$ is

$$
\begin{array}{llllll}
\min \alpha\bar{x} - \beta & & & & & \\
\alpha & & -u\tilde{A} + u_0 e_k & & = 0 & \\
\alpha & & & -v\tilde{A} - v_0 e_k & = 0 & \\
& -\beta + u\tilde{b} & & & = 0 & (CGLP)_k \\
& -\beta & & +v\tilde{b} + v_0 & = 0 & \\
& & ue + u_0 & + ve + v_0 & = 1 & \\
& & & u, u_0, v, v_o & \geq 0 &
\end{array}
$$

The normalization used affects the outcome of the optimization quite severely, so its choice is important. For instance, normalization (e), sometimes called the trivial normalization, is known [19] to yield a cut equivalent to the intersection cut from the set $S := \{x : 0 \leq x_k \leq 1\}$. A considerable amount of experimentation has led to the conclusion that normalization (f) is the most advantageous, though in no well defined sense can it be said to dominate the others.

6.4 Cut Lifting

An important property of L&P cuts is that they can be derived from a subproblem, which we will call the *reduced* problem, involving only those variables that play an active role in determining the optimum. The cuts valid for the reduced problem can then be lifted into valid cuts for the original problem by using closed form expressions to compute those coefficients missing from the reduced problem. This fact is of crucial importance, as it drastically reduces the cost of generating L&P cuts: problems with hundreds of thousands of variables typically have optimal solutions with a few hundred active—i.e., basic—variables. But apart from this fact, in the branch and bound context this property makes it possible to generate cuts at one node of the search tree, defined by certain variables being fixed at certain values, and therefore valid only at the descendants of the given node, and lift them into cuts valid throughout the search tree. To be specific, let $\bar{x}$ be the optimal solution to $\min\{cx : x \in P\}$, and let the cut generating linear program with normalization $\beta = 1$ or $\beta = -1$, denoted (CGLP), have optimal solution $w = (\alpha, u, u_0, v, v_0)$. To construct the reduced problem,

(a) complement all the variables at their upper bound; let $\bar{\bar{x}}$ be the resulting solution;
(b) delete all the structural variables x_i such that $\bar{\bar{x}}_i = 0$;
(c) delete the lower and upper bounding constraints on the deleted variables.

Let the reduced problem have variables indexed by $R := \{i \in N' : 0 < \bar{\bar{x}} < 1\} \cup \{i \in N \setminus N' : \bar{\bar{x}} > 0\}$, where $N' := \{1, \ldots, p\}$, and constraint set $\tilde{A}^R x^R \geq \tilde{b}^R$, and let the associated cut generating linear program with the same normalization $\beta = 1$ or $\beta = -1$ be denoted (CGLP)^R.

Theorem 6.2 ([19]) *Let $w^R = (\alpha^R, u^R, u_0^R, v^R, v_0^R)$ be an optimal solution to (CGLP)^R. Extend w^R to a solution $\bar{w} = (\bar{\alpha}, \bar{u}, \bar{u}_0, \bar{v}, \bar{v}_0)$ of (CGLP) by defining $\bar{u}_0 = u_0^R$, $\bar{v}_0 = v_0^R$,*

$$\bar{u}_i = u_i^R, \quad \bar{v}_i = v_i^R$$

for $i = 1, \ldots, m$,

$$\bar{u}_{m+i} = \begin{cases} (v^R - u^R)\tilde{A}_i^R & \text{if } v^R\tilde{A}_i^R > u^R\tilde{A}_i^R \\ 0 & \text{otherwise} \end{cases}$$

$$\bar{v}_{m+i} = \begin{cases} (u^R - v^R)\tilde{A}_i^R & \text{if } u^R\tilde{A}_i^R > v^R\tilde{A}_i^R \\ 0 & \text{otherwise} \end{cases}$$

for $i \in N \setminus R$,

$$\bar{u}_{m+n+i} = \bar{v}_{m+n+i} = 0 \text{ for } i \in N' \setminus R, \text{ and}$$

$$\bar{\alpha}_i = \begin{cases} \alpha_i & i \in R \\ \bar{u}\tilde{A}_i^R & i \in N \setminus R. \end{cases}$$

Proof By construction, $\bar{w}$ satisfies all constraints of (CGLP) missing from (CGLP)^R, while the remaining constraints are not affected. Thus $\bar{w}$ is feasible for (CGLP). It is shown in [19] that the reduced costs of (CGLP) associated with $\bar{w}$ are all nonnegative; hence $\bar{w}$ is an optimal solution to (CGLP). □

A more transparent way to describe the transition from w^R to $\bar{w}$ is as follows: An alternative representation of (CGLP), call it $(\overline{CGLP})$, is by (a) deleting from the set $\tilde{A}x \geq \tilde{b}$ the nonnegativity constraints $x \geq 0$, and denoting the resulting set by

$\tilde{\tilde{A}}x \geq \tilde{\tilde{b}}$; and (b) replacing the two sets of equations containing α with inequalities, i.e., replacing (6.3) with

$$\begin{aligned}
&\alpha - u\tilde{\tilde{A}} + u_0 e_j & & \geq 0 \\
&\alpha & - v\tilde{\tilde{A}} - v_0 e_j & \geq 0 \\
&\beta - u\tilde{\tilde{b}} & & = 0 \\
&\beta & - v\tilde{\tilde{b}} - v_0 & = 0 \\
& & u \geq 0, \quad v \geq 0 &
\end{aligned} \tag{6.3$'$}$$

Then any solution that minimizes $\alpha\bar{x} - \beta$ subject to (6.3)$'$, satisfies $\alpha_i = \max\{\alpha_i^1, \alpha_i^2\}$, where

$$\alpha_i^1 = \begin{cases} u\tilde{\tilde{A}}_i & i \neq j \\ u\tilde{\tilde{A}}_i + u_0 & i = j \end{cases} \qquad \alpha_i^2 = \begin{cases} v\tilde{\tilde{A}}_i & i \neq j \\ v\tilde{\tilde{A}}_i - v_0 & i = j \end{cases} \tag{6.4}$$

Using this representation, we have

Corollary 6.3 ([20]) *Let $\alpha^R x^R \geq \beta$ be the lift-and-project cut generated by solving $(\overline{CGLP})^R$. Then $\alpha x \geq \beta$ is a valid lifting of $\alpha^R x^R \geq \beta$ to the solution space of $(\overline{CGLP})$, with*

$$\alpha_i = \begin{cases} \alpha_i^R & i \in R \\ \max\{\alpha_i^1, \alpha_i^2\} & i \in N \setminus R \end{cases}$$

where

$$\alpha_i^1 = u^R \tilde{\tilde{A}}_i^R, \qquad \alpha_i^2 = v^R \tilde{\tilde{A}}_i^R,$$

and where $\tilde{\tilde{A}}_i^R$ is the subvector of $\tilde{\tilde{A}}_i$ with the same row set as $\tilde{\tilde{A}}^R$.

Proof $(\overline{CGLP})$ has more variables and constraints than $(\overline{CGLP})^R$, namely the constraints

$$\begin{aligned}
\alpha_i - u\tilde{\tilde{A}}_i \geq 0 \quad & \text{for } i \in N \setminus R \\
\alpha_i - v\tilde{\tilde{A}}_i \geq 0 \quad & \text{for } i \in N \setminus R
\end{aligned}$$

and the variables u_i and v_i corresponding to the constraints that have been removed from $\tilde{A}x \geq \tilde{b}$. A feasible solution w to $(\overline{CGLP})$ can be obtained from w^R by setting the extra variables u_i, v_i to 0 and setting α_i, $i \in N \setminus R$, as stated in the Corollary. □

6.5 Cut Strengthening

The lift-and-project cut $\alpha x \geq \beta$ can be strengthened to $\gamma x \geq \beta$ by using the integrality condition on the nonbasic variables. A theory of cut strengthening for more general disjunctive programs [24] will be discussed in a subsequent chapter. Here we state the strengthening procedure for the cut $\alpha x \geq \beta$ derived from the disjunction $x_j \leq 0 \vee x_j \geq 1$.

Theorem 6.4 *Let $\alpha x \geq \beta$ be a lift-and-project cut from $(\overline{CGLP})_j$, with $\alpha_i = \max\{\alpha_i^1, \alpha_i^2\}$, with α_i^1 and α_i^2 as in (6.4). Then $\gamma x \geq \beta$ is valid for (MIP), with*

$$\gamma_k = \begin{cases} \min\{\alpha_k^1 + u_0\lceil \bar{m}_k \rceil, \alpha_k^2 - v_0 \lfloor \bar{m}_k \rfloor\}, & k \in N' \\ \alpha_k (= \max\{\alpha_k^1, \alpha_k^2\}), & k \in N \setminus N' \end{cases} \tag{6.5}$$

where N' is the set of integer-constrained variables, α_k^1 and α_k^2 are as in (6.4), and $\bar{m} \in R^{N'}$, with

$$\bar{m}_k = \frac{\alpha_k^2 - \alpha_k^1}{u_0 + v_0}, \qquad k \in N'. \tag{6.6}$$

Proof If the disjunction $x_j \leq 0 \vee x_j \geq 1$ is valid, then so is the disjunction

$$x_j - mx \leq 0 \quad \vee \quad x_j - mx \geq 1 \tag{6.7}$$

for any vector m with integer components and $m_k = 0$ for $k \in N \setminus N'$. Then the coefficients of the cut derived from the disjunction (6.7) are

$$\gamma_k = \begin{cases} \max\{\alpha_k^1 + u_0 m_k,\ \alpha_k^2 - v_0 m_k\} & k \in N' \\ \max\{\alpha_k^1, \alpha_k^2\} & k \in N \setminus N' \end{cases}$$

We can now choose the integers $m_k, k \in N'$, so as to make γ_k as small as possible. These minimizing values are obtained by first finding the continuous value of m_k that makes the two terms of the maximand equal, which is $\bar{m}_k$ as in (6.6), and then taking $m_k = \lfloor \bar{m}_k \rfloor$ or $m_k = \lceil \bar{m}_k \rceil$, whichever makes γ_k smaller. This yields the expression in (6.5). □

It is worth noting that while a linear transformation of the system $\tilde{A}x \geq \tilde{b}$ leaves the solution set of the (CGLP) unchanged, it may change the result of the above strengthening procedure. Suppose, for instance, that instead of applying the disjunction $x_j \leq 0 \vee x_j \leq 1$ to $\tilde{A}x \geq \tilde{b}$, deriving the L&P cut $\alpha x \geq \beta$, and applying to it the above strengthening procedure to obtain $\gamma x \geq \beta$, we first solve the LP relaxation of (MIP), and then apply the disjunction $x_j \leq 0 \vee x_j \geq 1$ to the resulting system. The cut we obtain is then the intersection cut from the convex set $S := \{x : 0 \leq x_j \leq 1\}$, and its strengthened version is the mixed integer Gomory cut. However, if we reexpress the latter cut in terms of the structural variables (i.e., those indexed by N), we do in general get a cut different from $\gamma x \geq \beta$.

6.6 Impact on the State of the Art in Integer Programming

Next we discuss the role of disjunctive programming, in particular its lift-and-project incarnation, in the revolution in the state of the art in integer programming that took place roughly during the 10–15 year period starting around the mid-1990s.

First about the revolution itself. Integer programming started in the late 1950s as a way of representing conditions that eluded the linear or convex formulations of optimization problems for which solution methods were found during the previous decade: discontinuities, sharp turns, either/or conditions, implications, nonconvexities of various kinds. All these conditions could be represented in an integer programming framework via binary or more general discrete variables, thus extending the scope of optimization problems to all kinds of earlier unmanageable situations. However, this marvelous new universal tool for dealing with nonconvexities and discontinuities remained for several decades a toy model, unable to solve instances with more than a few dozen variables, i.e. incapable of coping with most practical problems. From the beginning there were two distinct ways of approaching the solution of integer programs, enumeration (branch and bound) and convexification (cutting planes). Research on cutting planes gave rise to a rich and elegant theory, but practical attempts at convexification via cutting planes foundered due to seemingly insurmountable numerical difficulties. Enumerative or branch and bound methods on the other hand were more or less straightforward, relatively easy to implement, but they required a computational effort exponential in the number of variables and therefore their problem solving capability was limited to instances with 30–40 variables. Thus until the early 1990s the only commercial integer solvers available were of the pure branch-and-bound type, and although their increasing level of sophistication led to gradual improvements in their reliability, flexibility and various other features, their ability to solve practical problems remained restricted—with the exception of some very special structures—to a few dozen variables.

Then around the mid-1990s things started to change: the commercial solvers started improving their ways of solving the linear programming relaxations; using heuristics during the search; preprocessing the instances before running them etc., but most importantly, they started incorporating cutting planes in their branch-and-bound procedure and as a result, their performance improved spectacularly. In just a few years, the technology of solving MIP's underwent a revolution. To see this, consider the fact that one of the leading commercial solvers, CPLEX, has increased its MIP solving speed more than 29,000 times between its version 1.2 of 1991 and its version 11.0 of 2007 [45], as illustrated by Fig. 6.3.

The chart in this figure represents the outcome of a massive computational test comparing the performance (on the same computer) of the successive versions of CPLEX between 1.2 of 1991 and 11.0 of 2007 on 1892 representative models of

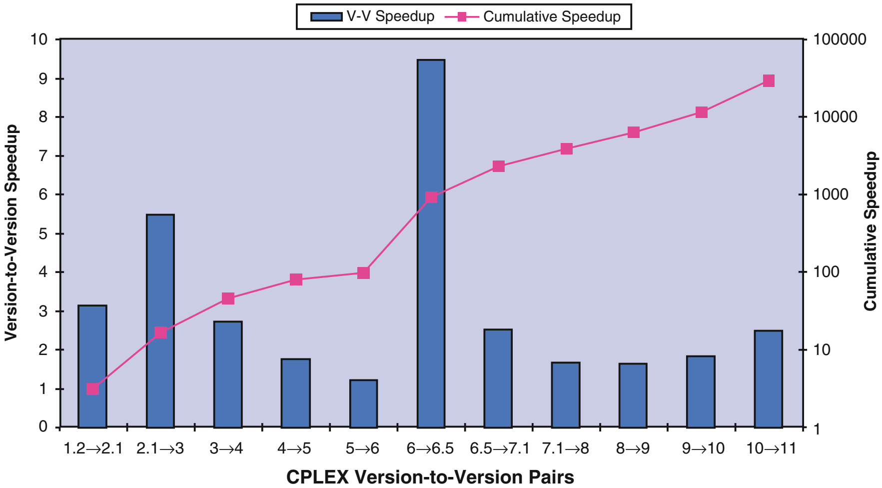


Fig. 6.3 CPLEX performance 1991–2007

an extensive library of real-world problems collected over the years from academic and industry sources. The scale on the left refers to the bars in the chart and the scale on the right to the piece-wise linear line through the middle. Thus the first bar shows that CPLEX 2.1 was about 3.1 times faster in this test than CPLEX 1.2, and each subsequent bar shows the number of times the given version was faster than the previous one. The biggest jump occurred with version 6.5 in 1998. The cumulative speedup shown by the broken line and the scale on the right is over 29,000. While this fantastic improvement has multiple sources, an important one being the speedup in the LP codes solving the relaxations, the use of cutting planes is the central one. After 2007 this speedup continued for several more years, although at a slower rate—for instance, Gurobi reports a 20.5-fold speed up between its version 1.0 of 2009 and its version 5.5 of 2013. Today the vast majority of MIPs encountered in practice can be solved in useful time, although there are instances, and entire classes of instances, that remain beyond reach by a reasonable computing effort—after all, MIP is an NP-hard problem.

What was it that suddenly led to the use of cutting planes, shunned for so many years as nice theoretical devices that unfortunately are of no practical use? Well, in the mid-1990s, as part of the work on lift–and-project cuts, the computer code MIPO (for Mixed Integer Program Optimizer) based on branch-and-bound combined with lift-an-project cuts was developed by Sebastian Ceria (see [20]), which on many of the test problems outperformed the best commercial or academic solvers available at the time. MIPO was using lift-and-project cuts at the root node but also at certain branching nodes, and was using them in a new way: rather than adding individual cuts and reoptimizing the LP relaxation after each addition, it would generate the cuts in rounds between any two reoptimizations. Also, due to the cut lifting feature, cuts generated at a node could be lifted into valid cuts for the entire tree, which was hitherto thought to be infeasible. Furthermore, soon Cornuejols and Ceria discovered that these features could also be applied to the generation of Gomory cuts, contrary to the prevailing view at the time, and an extensive computational experiment was undertaken to prove that Gomory cuts properly used within a branch and bound framework can bring vast improvements over the use of branch-and-bound alone [21]. Presentation of these results at a 1994 workshop provoked surprise and stirred interest in trying to incorporate cutting planes into the existing branch-and-bound codes. Since lift-and-project cuts were at that point still computationally expensive to generate (we are referring to the period prior to the developments to be discussed in the next chapter), while Gomory cuts were computationally cheap, they were the first ones to be incorporated into the commercial codes. Some other cuts based on special structures (knapsack, flow cover, clique) were also implemented, but their effect on the outcomes was much weaker than that of the Gomory cuts. The lift-and-project cuts themselves were incorporated into the commercial codes a few years later, after the developments to be discussed in the next chapter made their use computationally affordable.

Chapter 7
Nonlinear Higher-Dimensional Representations

Apart from the extended formulations discussed in Chap. 5, a number of authors have proposed nonlinear higher dimensional constructions that provide tighter relaxations of $\operatorname{conv} P_I$. In this chapter we briefly review these constructions and point out their relation to the lift-and-project technique of disjunctive programming.

7.1 Another Derivation of Lift-and-Project Cuts

In this section we discuss an alternative derivation of lift-and-project cuts [19], which is closely related to the "cones of matrices" approach of Lovász and Schrijver [99]. This derivation is for mixed 0-1 programs, but it can easily be extended to general MIP's. We change notation in order to facilitate comparisons with related more general procedures.

Define

$$\begin{aligned} K &:= \{x \in \mathbb{R}^n : Ax \geq b, x \geq 0, x_j \leq 1, j = 1, \ldots, p\} \\ &:= \{x \in \mathbb{R}^n : \tilde{A}x \geq \tilde{b}\} \end{aligned}$$

and

$$K^0 := \{x \in K : x_j \in \{0, 1\}, j = 1, \ldots, p\}.$$

K and K^0 correspond to P and P_I in our notation for general MIP's.

K^0 is the feasible set of a mixed 0-1 programming problem with n variables, p of which are 0-1 constrained. K is the standard linear programming relaxation.

In this section we consider procedures for finding $\operatorname{conv}(K^0)$, the convex hull of K^0.

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_7

Consider the following *Sequential convexification procedure*

Step 0. Select an index $j \in \{1, \ldots, p\}$.
Step 1. Multiply $\tilde{A}x \geq \tilde{b}$ with $1 - x_j$ and x_j to obtain the nonlinear system

$$\begin{aligned} (1 - x_j)(\tilde{A}x - \tilde{b}) &\geq 0, \\ x_j(\tilde{A}x - \tilde{b}) &\geq 0. \end{aligned} \tag{7.1}$$

Step 2. Linearize (7.1) by substituting y_i for $x_i x_j$, $i = 1, \ldots, n$, $i \neq j$, and x_j for x_j^2. Call the polyhedron defined by the resulting system $M_j(K)$.
Step 3. Project $M_j(K)$ onto the x-space. Call the resulting polyedron $P_j(K)$.

The linearization used in Step 2 yields, among others, the inequalities $y_i \geq 0$, $y_i \leq x_i$, for $i = 1, \ldots, n$, and $y_i \leq x_j$, $y_i \geq x_i + x_j - 1$, for $i = 1, \ldots, p$. If the system defining K has m constraints and n variables, the system defining $M_j(K)$ has $2m$ constraints and $2n - 1$ variables.

As to the projection used in Step 3, projecting $M_j(K)$ onto the x-space amounts to eliminating the variables y_i by taking certain specific linear combinations of the inequalities defining $M_j(K)$.

Theorem 7.1 $P_j(K) = \text{conv}(K \cap \{x \in \mathbb{R}^n : x_j \in \{0, 1\}\})$.

This theorem asserts that the result of the above 3-step procedure is the convex hull of the union of the two polyhedra $\{x \in K : y_j \leq 0\}$ and $\{x \in K : x_j \geq 1\}$. We will prove this by showing that the result of this procedure is precisely the representation of this convex hull given by Theorem 2.1.

Proof $M_j(K)$ is the polyhedron whose defining system is obtained from (7.1) by setting $y_i := x_i x_j$, $i = 1, \ldots, n$, $y_j = x_j = x_j^2$. The result of these operations is

$$\begin{aligned} \tilde{A}x - \tilde{b} - \tilde{A}y + \tilde{b}x_j &\geq 0 \\ \tilde{A}y - \tilde{b}x_j &\geq 0. \end{aligned} \tag{7.2}$$

If we now define $y_0 := x_j$, $z := x - y$, $z_0 := 1 - y_j$, (7.2) can be rewritten as

$$\begin{array}{rcl} x - z - y &=& 0 \\ \tilde{A}z - \tilde{b}z_0 &\geq& 0 \\ z_j &=& 0 \\ \tilde{A}y - \tilde{b}y_0 &\geq& 0 \\ y_j - y_0 &=& 0 \\ z_0 + y_0 &=& 1 \end{array} \tag{7.3}$$

where the equation $z_j = 0$ follows from $y_j = x_j$, $z_j = x_j - y_j$; and the equation $y_j - y_0 = 0$ follows from $y_j = x_j = y_0$.

If we now replace in (7.3) $z_j = 0$ with $-z_j = 0$ and interchange z with y, we get the system (6.2), which was shown to be the special case of (2.1) where $|Q| = 2$ and the disjunction whose terms are indexed by $|Q|$ is $x_j = 0 \vee x_j = 1$. Thus our Theorem is a special case of Theorem 2.1. $\square$

Since any x that satisfies $\tilde{A}x \geq \tilde{b}$ and $0 \leq x_j \leq 1$ clearly satisfies both $(1 - x_j)(\tilde{A}x - \tilde{b}) \geq 0$ and $x_j(\tilde{A}x - \tilde{b}) \geq 0$, the multiplications performed in Step 1 above are not responsible for tightening the constraints of K. Further, replacing $x_i x_j$ with y_i for all $i \neq j$ in Step 2 above cannot tighten those constraints either. Yet, unless the 0-1 constraint is redundant for variable j, the projection $P_j(K)$ of the set $M_j(K)$ resulting from Step 2 is strictly contained in K. The only operation that is "accountable" for this tightening is the replacement of the term x_j^2 by x_j. Indeed, while this substitution does not eliminate any points for which $x_j \in \{0, 1\}$, it may cut off points x with $0 < x_j < 1$.

For $t \geq 2$, define $P_{i_1,\ldots,i_t}(K) = P_{i_t}(P_{i_{t-1}} \cdots (P_{i_1}(K)) \cdots)$.

Theorem 7.2 *For any $t \in \{1, \ldots, p\}$,*

$$P_{i_1,\ldots,i_t}(K) = \text{conv}(K \cap \{x \in \mathbb{R}^n : x_j \in \{0, 1\}, j = i_1, \ldots, i_t\}).$$

Corollary 7.3 $P_{1,\ldots,p}(K) = \text{conv}(K^0)$.

Theorem 7.2 and its Corollary follow directly from Theorem 3.1 on the sequential convexifiability of facial disjunctive programs, of which the mixed 0-1 program is a special case.

The above derivation of lift-and-project cuts was inspired by the work of Lovász and Schrijver, which we address next.

7.2 The Lovász-Schrijver Construction

In [99], L. Lovász and A. Schrijver proposed the following procedure:

Step 1 Multiply $\tilde{A}x \geq \tilde{b}$ with x_j and $1 - x_j$, $j = 1, \ldots, p$, to obtain the nonlinear system

$$\begin{aligned}
&(1 - x_1)(\tilde{A}x - \tilde{b}) \geq 0,\\
&x_1(\tilde{A}x - \tilde{b}) \geq 0,\\
&(1 - x_2)(\tilde{A}x - \tilde{b}) \geq 0,\\
&x_2(\tilde{A}x - \tilde{b}) \geq 0,\\
&\vdots\\
&(1 - x_p)(\tilde{A}x - \tilde{b}) \geq 0,\\
&x_p(\tilde{A}x - \tilde{b}) \geq 0.
\end{aligned} \tag{7.4}$$

Step 2. Linearize (7.4) by substituting y_{ij} for $x_i x_j$, setting $y_{ij} = y_{ji}$, $i = 1, \ldots, n$, $j = 1, \ldots, p$, $i \neq j$, and substituting x_j for x_j^2, $j = 1, \ldots, p$. Call the polyhedron defined from the resulting system $M(K)$.

Step 3. Project $M(K)$ onto the x-space. Call the resulting polyhedron $N(K)$.

Note that, if the system defining K has m constraints and n variables, of which p are 0-1 constrained in K^0, the system defining $M(K)$ has $2pm$ constraints and $pn + n - \frac{1}{2}p(p+1)$ variables. Lovász and Schrijver have shown that $N(K)$ has the following properties:

Theorem 7.4 ([99].) $N(K) \subseteq \operatorname{conv}(K \cap \{x \in \mathbb{R}^n : x_j \in \{0, 1\}\})$, $j = 1, \ldots, p$.

Let $N^1(K) = N(K)$ and $N^t(K) = N(N^{t-1}(K))$, for $t \geq 2$.

Theorem 7.5 ([99]) $N^p(K) = \operatorname{conv}(K^0)$.

In other words, iterating the above procedure p times yields the integer hull. Theorem 7.1 implies Theorem 7.4, since $N(K) \subseteq P_j(K)$. Corollary 7.3 implies Theorem 7.5. Again, this follows from $N(K) \subseteq P_j(K)$, $j = 1, \ldots, p$. Note, however, that the Lovász and Schrijver relaxation $N(K)$ is not only stronger than $P_j(K)$ for any j, but also stronger than $\bigcap_{j=1}^p P_j(K)$; the inclusion $N(K) \subseteq \bigcap_{j=1}^p P_j(K)$ will in general be strict (an important exception being the stable set polytope; see section 2.5 of [19]). This is so because the construction of $N(K)$, unlike that of $\bigcap_{j=1}^p P_j(K)$, involves setting $y_{ij} = y_{ji}$ for all i, j before projecting away the y-variables. Furthermore, the coefficient matrix of the system resulting from the linearization of Step 2 has been shown in [99] to be positive semidefinite, a condition which can be used to further strengthen any relaxation based on this construction.

7.3 The Sherali-Adams Construction

Somewhat earlier than Lovász and Schrijver, Sherali and Adams had proposed a similar convexification procedure [112]. Let K and K^0 be defined as above, and let $t \in \{1, \ldots, p\}$.

Step 1. Multiply $\tilde{A}x \geq \tilde{b}$ with every product of the form $\left[\prod_{j \in J_1} x_j\right] \left[\prod_{j \in J_2} (1 - x_j)\right]$, where J_1 and J_2 are disjoint subsets of $\{1, \ldots, p\}$ such that $|J_1 \cup J_2| = t$. Call the resulting nonlinear system (NL$_t$).

Step 2. Linearize (NL$_t$) by (i) substituting x_j for x_j^2; and (ii) substituting a variable w_J for every product $\prod_{j \in J} x_j$, where $J \subseteq \{1, \ldots, p\}$, and v_{Jk} for every product $x_k \prod_{j \in J} x_j$ where $J \subseteq \{1, \ldots, p\}$ and $k \in \{p+1, \ldots, n\}$. Call the polyhedron defined by the resulting system X_t.

Step 3. Project X_t onto the x-space. Call the resulting polyhedron K_t.

It is easy to see that $K^0 \subseteq K_p \subseteq \cdots K_1 \subseteq K$. In addition, Sherali and Adams proved the following:

Theorem 7.6 *[112]* $K_p = \text{conv}(K^0)$.

Next we prove a result which shows that Theorem 7.6 also follows from Theorem 7.2.

Theorem 7.7 *For* $t = 1, \ldots, p$, $K_t \subseteq P_{1,\ldots,t}(K)$.

Proof Let $A^j x \geq b^j$ denote a minimal linear system describing $P_{1,\ldots,j}(K)$, for $j = 1, \ldots, t$. Let $\alpha x \geq \beta$ be one of the inequalities defining $P_{1,\ldots,t}(K)$. Then $\alpha x \geq \beta$ can be obtained by taking a nonnegative linear combination of the inequalities $(1 - x_t)(A^{t-1}x - b^{t-1}) \geq 0$ and $x_t(A^{t-1}x - b^{t-1}) \geq 0$, with multipliers that eliminate the nonlinear products $x_i x_t$, $i \neq t$, and substituting x_t^2 by x_t. By the same argument every inequality of the system $A^{t-1}x - b^{t-1} \geq 0$ can be obtained by taking a nonnegative linear combination of the inequalities $(1 - x_{t-1})(A^{t-2}x - b^{t-2}) \geq 0$ and $x_{t-1}(A^{t-2}x - b^{t-2}) \geq 0$, with multipliers that eliminate all products $x_i x_{t-1}$, $i \neq t - 1$ and setting $x_{t-1}^2 = x_{t-1}$. By inductively repeating this argument we can obtain $\alpha x \geq \beta$ in terms of the inequalities of (NL$_t$), by first substituting x_j^2 by x_j, $j = 1, \ldots, t$, and then eliminating the remaining nonlinear terms using as multipliers the product of the multipliers used in each step of the induction. Therefore $\alpha x \geq \beta$ is valid for K_t and the result follows. □

Now Theorem 7.6 follows from Corollary 7.3 and Theorem 7.7. It also follows from Theorem 7.5 and a proposition in [99] that shows that $K_t \subseteq N^t(K)$.

7.4 Lasserre's Construction

In [94, 95], J.B. Lasserre introduced a relaxation that strengthens the Sherali-Adams construction by adding to it positive semidefiniteness constraints on a set of so-called moment matrices whose rows and columns are indexed by all subsets of $\{1, \ldots, n\}$ and on m additional moment matrices obtained from each row of the system $Ax - b \geq 0$. For further information see the above cited articles and the book [96]. The Lasserre relaxation is obtained by projecting the resulting set onto $\mathbb{R}^n$. It is tighter than the relaxations discussed in the preceding section, but computationally more demanding.

7.5 The Bienstock-Zuckerberg Lift Operator

In [43], D. Bienstock and M. Zuckerberg used subset algebra to define a lift operator that is one of the strongest so far known. Their operator is valid for any 0/1 integer program but yields strong formulations in the particular case of the *set* *covering* problem,

$$\begin{aligned} &\min\ c^T x \\ \text{s.t.}\quad &Ax \ge \mathbf{e}, \quad x \in \{0,1\}^n \end{aligned} \tag{7.5}$$

A is an $m \times n$, 0/1 matrix and $\mathbf{e} = (1, \ldots, 1)^T \in \mathbb{R}^m$. The approach in [43] was motivated by the result of Balas and Ng [29] which characterized all facets for (7.5) with coefficients in $\{0, 1, 2\}$. This last result is relevant because symmetric set-covering systems tend to produce highly fractional extreme points that are very difficult to cut-off by standard mixed-integer programming systems but are cut-off by combinatorial inequalities with small integral coefficients. Moreover, one can produce set-covering systems with an exponential number of facets with coefficients in $\{0, 1, 2\}$.

Bienstock and Zuckerberg set out to describe all valid inequalities with coefficients in $\{0, 1, \ldots, k\}$, for any fixed integer $k > 1$. Note that unless a row of $Ax \ge \mathbf{e}$ is of the form $x_j \ge 1$ for some j, which is easily eliminated, every nondominated inequality valid for (7.5) has nonnegative coefficients. Hence with increasing k we obtain a more complete description of all valid inequalities for (7.5).

More generally, Bienstock and Zuckerberg [43] introduces the notion of *pitch*. An inequality $\alpha^T x \ge b$ valid for (7.5) with $\alpha \ge 0$ is said to have pitch $\le k$ if the sum of the k smallest positive α_j is at least b (thus, any inequality with coefficients in $\{0, 1, \ldots, k\}$ has pitch $\le k$). The following is proved in [43].

Theorem 1 *For any fixed integer $k \ge 1$ there exists a compact, extended formulation for (7.5) whose solutions satisfy all valid inequalities with pitch $\le k$.*

In this statement, *compact* means of polynomial size (for fixed k), and *extended* means lifted. Bienstock and Zuckerberg continued their work in [44] where they addressed the relationship of pitch of valid inequalities and Gomory rank. They proved the following result:

Theorem 2 *Let $r > 0$ and $0 < \epsilon < 1$ be given, and let $G^r(A)$ be the rank-r Gomory closure of (7.5). Then there is a compact, extended formulation for (7.5) whose projection $\tilde{P}$ to $\mathbb{R}^n$ satisfies:*

$$\forall c \in \mathbb{R}^n,\ \min\{c^T x : x \in \tilde{P}\} \ \ge\ (1-\epsilon)\min\{c^T x : x \in \tilde{G^r}(A)\}$$

Theorem 2 is obtained from Theorem 1 by appropriately choosing k as a function of r and ϵ.

Chapter 8
The Correspondence Between Lift-and-Project Cuts and Simple Disjunctive Cuts

From the fact that the constraint set (6.3) of $(\mathrm{CGLP})_j$ defines the convex hull of $P \cap \{x : x_j \in \{0, 1\}\}$, and that $\mathrm{conv}\, P_I$, the integer hull, can be derived by imposing the disjunctions $x_j \leq 0 \vee x_j \geq 1$ sequentially, it follows that any valid cut for a mixed 0-1 program can be represented as a lift-and-project cut. In this chapter we discuss the exact correspondence between lift-and-project cuts for a mixed 0-1 program and earlier cuts from the literature based on [33].

We will consider $(\mathrm{CGLP})_k$, the cut generating linear program based on the disjunction $-x_k \geq 0 \vee x_k \geq 1$, with the normalization constraint $ue + u_0 + ve + v_0 = 1$, i.e., with the constraint set

$$
\begin{array}{llllll}
\alpha & - u\tilde{A} + u_0 e_k & & & = 0 \\
\alpha & & - v\tilde{A} - v_0 e_k & & = 0 \\
-\beta + u\tilde{b} & & & & = 0 \\
-\beta & & + v\tilde{b} + v_0 & & = 0 \\
& ue + u_0 & + ve + v_0 & & = 1 \\
& u, \quad u_0, & v, \quad v_0 & & \geq 0
\end{array}
\tag{8.1}
$$

where e is the summation vector, and u and v have $m + p + n$ components (i.e., A is $m \times n$, and p out of the n variables are binary).

Lemma 8.1 *In any basic solution to (8.1) that yields an inequality $\alpha x \geq \beta$ not dominated by the constraints of the LP relaxation of (MIP), both u_0 and v_0 are positive.*

Proof If $u_0 = 0$, then $\alpha = u\tilde{A}$, $\beta = u\tilde{b}$; and if $v_0 = 0$, then $\alpha = v\tilde{A}$, $\beta = v\tilde{b}$. In either case, $\alpha x \geq \beta$ is a nonnegative linear combination of the inequalities of $\tilde{A}x \geq \tilde{b}$. □

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_8

8.1 Feasible Bases of the CGLP Versus (Feasible or Infeasible) Bases of the LP

Since the components of (α, β) are unrestricted in sign, we may assume w.l.o.g. that they are all basic. We then have

Lemma 8.2 ([33]) *Let $(\bar{\alpha}, \bar{\beta}, \bar{u}, \bar{u}_0, \bar{v}, \bar{v}_0)$ be a basic solution to (8.1), with $\bar{u}_0, \bar{v}_0 > 0$ and all components of $(\bar{\alpha}, \bar{\beta})$ basic. Further, let the basic components of $\bar{u}$ and $\bar{v}$ be indexed by M_1 and M_2, respectively. Then $M_1 \cap M_2 = \emptyset$, $|M_1 \cup M_2| = n$, and the $n \times n$ submatrix $\hat{A}$ of $\tilde{A}$ whose rows are indexed by $M_1 \cup M_2$ is nonsingular.*

Proof Removing from (8.1) the nonbasic variables and subscripting the basic components of u and v by M_1 and M_2, respectively, we get

$$\begin{array}{rllll} \alpha - u_{M_1}\tilde{A}_{M_1} & + u_0 e_k & & & = 0 \\ \alpha & & - v_{M_2}\tilde{A}_{M_2} & - v_0 e_k & = 0 \\ -\beta + u_{M_1}\tilde{b}_{M_1} & & & & = 0 \\ -\beta & & + v_{M_2}\tilde{b}_{M_2} & + v_0 & = 0 \\ u_{M_1}e_{|M_1|} & + u_0 & + v_{M_2}e_{|M_2|} & + v_0 & = 1, \\ u_{M_1}, & u_0, & v_{M_2}, & v_0 & \geq 0 \end{array} \tag{8.2}$$

where $\tilde{A}_{M_1}$ ($\tilde{A}_{M_2}$) is the submatrix of $\tilde{A}$ whose rows are indexed by M_1 (by M_2). Eliminating the variables α, β, unrestricted in sign, we obtain the system

$$\begin{array}{lll} (u_{M_1}, -v_{M_2})\binom{\tilde{A}_{M_1}}{\tilde{A}_{M_2}} & - (u_0 + v_0)e_k & = 0 \\ (u_{M_1}, -v_{M_2})\binom{\tilde{b}_{M_1}}{\tilde{b}_{M_2}} & \qquad - v_0 & = 0 \\ u_{M_1}e_{|M_1|} + v_{M_2}e_{|M_2|} + u_0 & + v_0 & = 1 \end{array} \tag{8.3}$$

of $n + 2$ equations, of which $(\bar{u}_{M_1}, \bar{u}_0, \bar{v}_{M_2}, \bar{v}_0)$ is the unique solution. Since the number of variables, like that of the constraints, is $n+2$, it follows that $|M_1|+|M_2| = n$.

Now suppose that

$$\hat{A} := \begin{pmatrix} \tilde{A}_{M_1} \\ \tilde{A}_{M_2} \end{pmatrix}$$

is singular. Then there exists a vector $(u^*_{M_1}, v^*_{M_2})$ such that $(u^*_{M_1}, -v^*_{M_2})\hat{A} = 0$ and $u^*_{M_1} e_{|M_1|} + v^*_{M_2} e_{|M_2|} = 1$. By setting

$$v_0^* = (u^*_{M_1}, -v^*_{M_2}) \begin{pmatrix} \tilde{b}_{M_1} \\ \tilde{b}_{M_2} \end{pmatrix}, \quad u_0^* = -v_0^*$$

we obtain a solution $(u^*_{M_1}, u^*_0, v^*_{M_2}, v^*_0)$ to (8.3). By assumption $\bar{u}_0 > 0$ and $\bar{v}_0 > 0$, hence the solution $(u^*_{M_1}, u^*_0, v^*_{M_2}, v^*_0)$ differs from $(\bar{u}_{M_1}, \bar{u}_0, \bar{v}_{M_2}, \bar{v}_0)$. But this contradicts the fact that $(\bar{u}_{M_1}, \bar{u}_0, \bar{v}_{M_2}, \bar{v}_0)$ is the unique solution to (8.3), which proves the $\hat{A}$ is nonsingular.

If $|M_1 \cap M_2| \neq \emptyset$ then $\hat{A}$ will be singular, hence $|M_1 \cup M_2| = n$. □

Now define $J := M_1 \cup M_2$, and consider the system obtained from $\tilde{A}x \geq \tilde{b}$ by replacing the n inequalities indexed by J with equalities, i.e., by setting the corresponding surplus variables to 0. Since the submatrix $\hat{A}$ of $\tilde{A}$ whose rows are indexed by J is nonsingular, these n equations define a basic solution, with an associated simplex tableau whose *nonbasic* variables are indexed by J. Recall that in the $(\text{CGLP})_k$ solution that served as our starting point, $J := M_1 \cup M_2$ was the index set of the *basic* components of (u, v). Writing $\hat{b}$ for the subvector of $\tilde{b}$ corresponding to $\hat{A}$, and s_J for the surplus variables indexed by J, we have

$$\hat{A}x - s_J = \hat{b}$$

or

$$x = \hat{A}^{-1}\hat{b} + \hat{A}^{-1}s_J. \tag{8.4}$$

Here some components of s_J may be surplus variables in an inequality of the form $x_j \geq 0$. Such a variable is of course equal to, and therefore can be replaced by, x_j itself. If that is done, then the row of (8.4) corresponding to x_k (a basic variable, since $k \notin J$) can be written as

$$x_k = \bar{a}_{k0} - \sum_{j \in J} \bar{a}_{kj} x_j \tag{8.5}$$

where $\bar{a}_{k0} = e_k \hat{A}^{-1}\hat{b}$ and $\bar{a}_{kj} = -(\hat{A}^{-1})_{kj}$. Notice that (8.5) is the expression of x_k in the simplex tableau with nonbasic variables indexed by J.

Note that this basic solution (and the associated simplex tableau) need not be feasible, either in the primal or in the dual sense. On the other hand, it has the following property.

Lemma 8.3 $0 < \bar{a}_{k0} < 1$.

Proof From (8.3),

$$(u_{M_1}, -v_{M_2}) = (u_0 + v_0)e_k \hat{A}^{-1},$$
$$(u_0 + v_0)e_k \hat{A}^{-1}\hat{b} = v_0 \tag{8.6}$$

and since $u_0 > 0$, $v_0 > 0$,

$$0 < \bar{a}_{k0} = e_k \hat{A}^{-1}\hat{b} = \frac{v_0}{u_0 + v_0} < 1. \qquad \square$$

Theorem 8.4A ([33]) *Let $\alpha x \geq \beta$ be the lift-and-project cut associated with a basic feasible solution $(\alpha, \beta, u, u_0, v, v_0)$ to (CGLP)$_k$, with $u_0, v_0 > 0$, all components of α, β basic, and the basic components of u and v indexed by M_1 and M_2, respectively.*

Let $\pi s_J \geq \pi_0$ be the simple disjunctive cut from the disjunction $x_k \leq 0 \vee x_k \geq 1$ applied to the row (8.5) of the simplex tableau defined by $J := M_1 \cup M_2$.

Then $\pi s_J \geq \pi_0$ is equivalent to $\alpha x \geq \beta$.

Proof From Lemma 8.2 we have that the matrix $\hat{A} = \tilde{A}_j$ is nonsingular, so (8.5) is well-defined. The cut from the disjunction $x_k \leq 0 \vee x_k \geq 1$ applied to (8.5) can be written as $\pi s_J \geq \pi_0$, with

$$\pi_0 := \bar{a}_{k0}(1 - \bar{a}_{k0}) \tag{8.7}$$

where $\bar{a}_{k0} = e_k \hat{A}^{-1}\hat{b}$, and

$$\pi_j := \max\{\pi_j^1, \pi_j^2\}, \quad j \in J,$$

with

$$\pi_j^1 := \bar{a}_{kj}(1 - \bar{a}_{k0}) = -(\hat{A}^{-1})_{kj}(1 - \bar{a}_{k0}), \quad \pi_j^2 := -\bar{a}_{kj}\bar{a}_{k0} = (\hat{A}^{-1})_{kj}\bar{a}_{k0}. \tag{8.8}$$

From (8.8), $\pi_j^1 \cdot \pi_j^2 < 0$, hence $\pi_j \geq 0$, which implies $u_J \geq 0$, $v_J \geq 0$.

The equivalence of $\pi s_J \geq \pi_0$ to the cut $\alpha x \geq \beta$ corresponding to the basic solution $w = (\alpha, \beta, u, u_0, v, v_0)$ of (CGLP)$_k$ is obtained by showing that

$$\begin{aligned} \theta\alpha &= \pi \hat{A}, \qquad \theta\beta = \pi_0 + \pi\hat{b}, \\ \theta u_J &= \pi - \pi^1, \; \theta v_J = \pi - \pi^2, \\ \theta u_0 &= 1 - \bar{a}_{k0}, \; \theta v_0 = \bar{a}_{k0} \end{aligned} \tag{8.9}$$

for some $\theta > 0$. This we do by showing that $\alpha, \beta, u, u_0, v, v_0$ as defined by (8.9) satisfies (8.2). Indeed, using (8.7) and (8.8),

$$\begin{aligned}
\theta(\alpha - u_J\hat{A} + u_0 e_k) &= \pi\hat{A} - (\pi - \pi^1)\hat{A} + (1 - \bar{a}_{k0})e_k = \pi^1\hat{A} + (1 - \bar{a}_{k0})e_k \\
&= -e_k\hat{A}^{-1}\hat{A}(1 - \bar{a}_{k0}) + (1 - \bar{a}_{k0})e_k = 0 \\
\theta(-\beta + u_J\hat{b}) &= -(\pi_0 + \pi\hat{b}) + (\pi - \pi^1)\hat{b} = -\pi_0 - \pi^1\hat{b} \\
&= -\bar{a}_{k0}(1 - \bar{a}_{k0}) + e_k\hat{A}^{-1}\hat{b}(1 - \bar{a}_{k0}) \\
&= -\bar{a}_{k0}(1 - \bar{a}_{k0}) + \bar{a}_{k0}(1 - \bar{a}_{k0}) = 0 \\
\theta(\alpha - v_J\hat{A} - v_0 e_k) &= \pi\hat{A} - (\pi - \pi^2)\hat{A} - \bar{a}_{k0}e_k = \pi^2\hat{A} - \bar{a}_{k0}e_k \\
&= e_k\hat{A}^{-1}\hat{A}\bar{a}_{k0} - \bar{a}_{k0}e_k = 0 \\
\theta(-\beta + v_J\hat{b} + v_0) &= -(\pi_0 + \pi\hat{b}) + (\pi - \pi^2)\hat{b} + \bar{a}_{k0} = -\pi_0 - \pi^2\hat{b} + \bar{a}_{k0} \\
&= -\bar{a}_{k0}(1 - \bar{a}_{k0}) - e_k\hat{A}^{-1}\hat{b}\bar{a}_{k0} + \bar{a}_{k0} \\
&= -\bar{a}_{k0}(1 - \bar{a}_{k0}) - \bar{a}_{k0}\bar{a}_{k0} + \bar{a}_{k0} = 0.
\end{aligned}$$

We further have that

$$\theta u_j = \pi_j - \pi_j^1 = \max\{\pi_j^1, \pi_j^2\} - \pi_j^1 = \begin{cases} \pi_j^2 - \pi_j^1 & \text{if } j \in M_1 \\ 0 & \text{if } j \in M_2 \end{cases}$$

and

$$\theta v_j = \pi_j - \pi_j^2 = \max\{\pi_j^1, \pi_j^2\} - \pi_j^2 = \begin{cases} 0 & \text{if } j \in M_1 \\ \pi_j^1 - \pi_j^2 & \text{if } j \in M_2 \end{cases}$$

so u and v as defined by (8.9) is zero for every component not in M_1 and M_2 respectively. Finally, if we choose θ in (8.9) such that the normalization constraint

$$u_{M_1}e_{M_1} + u_0 + v_{M_2}e_{M_2} + v_0 = 1$$

is satisfied, then we have that $\alpha, \beta, u, u_0, v, v_0$ as defined by (8.9) satisfies the system (8.2). Since w is a basic solution and therefore is the unique solution to (8.2), it must be as defined by (8.9).

The cut $\theta\alpha x \geq \theta\beta$ defined by (8.9) is $(\pi\hat{A})x \geq (\pi_0 + \pi\hat{b})$. Substituting for x using (8.4) we obtain the cut $\pi s_J \geq \pi_0$, which shows the equivalence. □

Theorem 8.4A has the following converse.

Theorem 8.4B ([33]) *Let $\hat{A}$ be any $n \times n$ nonsingular submatrix of $\tilde{A}$ and $\hat{b}$ the corresponding subvector of $\tilde{b}$, such that*

$$0 < e_k \hat{A}^{-1}\hat{b} < 1,$$

and let J be the row index set of $(\hat{A}, \hat{b})$. Further, let $\pi s_J \geq \pi_0$ be the simple disjunctive cut obtained from the disjunction $x_k \leq 0 \vee x_k \geq 1$ applied to the expression of x_k in terms of the nonbasic variables indexed by J. Further, let (M_1, M_2) be any partition of J such that $j \in M_1$ if $\pi_j^1 < \pi_j^2$ (i.e., $\bar{a}_{kj} < 0$) and $j \in M_2$ if $\pi_j^1 > \pi_j^2$ (i.e., $\bar{a}_{kj} > 0$), where π_j^1, π_j^2 are defined by (8.8).

Now let $(\alpha, \beta, u, u_0, v, v_0)$ be the basic feasible solution to $(\mathrm{CGLP})_k$ in which all components of α, β are basic, both u_0 and v_0 are positive, and the basic components of u and v are indexed by M_1 and M_2, respectively.

Then the lift-and-project cut $\alpha x \geq \beta$ is equivalent to $\pi s_J \geq \pi_0$.

Proof First we show that the choice of basic variables for $(\mathrm{CGLP})_k$ in the Theorem is well-defined, i.e., that they form a basis. We proceed as in the proof of Lemma 8.2 by first eliminating all variables chosen to be nonbasic from the system (8.1), which results in the system (8.2). If we further eliminate α and β we obtain (8.3). Here $\binom{\tilde{A}_{M_1}}{\tilde{A}_{M_2}} = \hat{A}$, which is nonsingular. From this it follows that (8.3) has a unique solution and hence (8.2) also has a unique solution. Therefore, the choice of basic variables forms a basis.

Next we show the equivalence. Consider the solution $(\alpha, \beta, u, u_0, v, v_0)$ defined by (8.9). Following the proof of Theorem 8.4A we have that this solution satisfies (8.2) for a certain θ and that $\alpha x \geq \beta$ is equivalent to the cut $\pi s_J \geq \pi_0$. We have shown above that (8.2) has a unique solution for the choice of basis given in Theorem 8.4B, hence this basic solution must be as defined by (8.9), i.e. feasible (with $u, v \geq 0$). $\square$

Note that in spite of the close correspondence that Theorems 8.4A and 8.4B establish between bases of $(\mathrm{CGLP})_k$ and those of (LP), this correspondence is in general not one to one. Of course, when $\pi_j^1 \neq \pi_j^2$ for all $j \in J$, then the partition (M_1, M_2) of J is unique. But when $\pi_j^1 = \pi_j^2$ for some $j \in J$, which is only possible when $\pi_j^1 = \pi_j^2 = 0$ (since π_j^1 and π_j^2, when nonzero, are of opposite signs), then the corresponding index j can be assigned to either M_1 or M_2, each assignment yielding a different basis for $(\mathrm{CGLP})_k$. However, although the two bases are different, the associated basic solutions to $(\mathrm{CGLP})_k$ are the same, since the components of u and v corresponding to an index $j \in J$ with $\pi_j^1 = \pi_j^2 = 0$ are 0, and hence the pivot in $(\mathrm{CGLP})_k$ that takes one basis into the other is degenerate; i.e., the change of bases does not produce a change of solutions.

The system (8.1) consists of a cone, given by the homogeneous constraints, truncated by the normalization constraint. Since (8.1) is bounded, the extreme points of (8.1) are in one-to-one correspondence with the extreme rays of the unnormalized cone. Hence the relationship in Theorems 8.4A and 8.4B can also be interpreted as one between basic (feasible or infeasible) solutions to (LP) and extreme rays of the cone defined by the homogeneous subsystem of (8.1).

Next we show that the correspondence between the cuts $\alpha x \geq \beta$ and $\pi s_j \geq \pi_0$ established in Theorems 8.4A, 8.4B carries over to the strengthened version of these cuts.

8.2 The Correspondence Between the Strengthened Cuts

The strengthened lift-and-project cut $\gamma x \geq \beta$ was defined in Theorem 6.4 of Chap. 6. The simple disjunctive cut from $x_k \leq 0 \vee x_k \geq 1$, written as $\pi s_J \geq \pi_0$, where π_0 and π are defined by (8.7) and (8.8), can be strengthened by the same procedure, replacing π with $\bar{\pi}$, defined as

$$\bar{\pi}_j = \begin{cases} \min\{f_{kj}(1-\bar{a}_{k0}), (1-f_{kj})\bar{a}_{k0}\}, & j \in J \cap N' \\ \pi_j & j \in J \setminus N' \end{cases} \tag{8.10}$$

Theorem 8.5 *Theorems 8.4A and 8.4B remain valid if the inequalities $\alpha x \geq \beta$ and $\pi s_J \geq \pi_0$ are replaced by the strengthened lift-and-project cut $\gamma x \geq \beta$ defined by Theorem 6.4, and the mixed integer Gomory cut (or strengthened simple disjunctive cut) $\bar{\pi} s_J \geq \pi_0$ defined by (8.10), respectively.*

Proof Outline The only coefficients of the cut $\pi s_J \geq \pi_0$ that can possibly be strengthened are those π_j such that $j \in J \cap N'$, i.e., x_j is a structural integer-constrained nonbasic variable. It is proved in [32] that these are precisely the indices of the coefficients of the cut $\alpha x \geq \beta$ that can be strengthened, and that the resulting strengthened coefficients γ_j are the same as the $\bar{\pi}_j$. □

8.3 Bounds on the Number of Essential Cuts

The correspondences established in Theorems 8.4A, 8.4B and 8.5 allow us to derive some new bounds on the number of undominated cuts of each type. Indeed, since every valid inequality for $\{x \in P : (x_k \leq 0) \vee (x_k \geq 1)\}$ is dominated by some lift-and-project cut corresponding to a basic feasible solution of $(\text{CGLP})_k$, the number of undominated valid inequalities is bounded by the number of bases of $(\text{CGLP})_k$, which in turn cannot exceed

$$\binom{\#\text{ variables}}{\#\text{ constraints}} = \binom{2(m+p+n+1)+n+1}{2n+3},$$

a rather weak bound. But from Theorems 8.4A/8.4B it follows that a much tighter upper bound is also available, namely the number of ways to choose a subset of n

variables to be nonbasic in a simplex tableau where x_k is basic, that is, the number of subsets J of cardinality n of the set $\{1, \ldots, m+p+n\} \setminus \{k\}$:

Corollary 8.6 *The number of facets of the polyhedron* $\operatorname{conv}(P_{k0} \cup P_{k1})$ *is bounded by*

$$\binom{m+p+n-1}{n}.$$

Thus the elementary closure $\cap_{k=1}^{p} P_k$ of P with respect to the lift-and-project operation has at most $p\binom{m+p+n-1}{n}$ facets.

Similarly, the number of undominated simple disjunctive cuts obtainable by applying the disjunction $x_k \leq 0 \vee x_k \geq 1$ to the expression of x_k in terms of any other variables is bounded by

$$\binom{m+p+n-1}{n},$$

and the number of cuts of this type for all $k \in \{1, \ldots, p\}$ is consequently at most $p\binom{m+p+n-1}{n}$.

If we try to extend these bounds to strengthened lift-and-project cuts, or to strengthened simple disjunctive cuts, we run into the problem that the extension is only valid if we restrict ourselves to strengthened cuts derived from basic solutions. But this is not satisfactory, since although any unstrengthened cut in either class is dominated by some unstrengthened cut corresponding to a basic solution, the same is not true of the strengthened cuts: a strengthened cut from a nonbasic solution need not be dominated by any strengthened cut from a basic solution: counterexamples are easy to produce.

On the other hand, the correspondences established in Theorems 8.4A/8.4B and 8.5 have an important consequence on the rank of the LP relaxation of a 0-1 mixed integer program with respect to the various families of cuts examined here.

8.4 The Rank of P with Respect to Different Cuts

Let P again denote the feasible region of the LP relaxation. It is well known that in the case of a pure 0-1 program, i.e. when $p = n$, the rank of P with respect to the family of (pure integer) fractional Gomory cuts can be strictly greater than n [73]. We now show that by contrast, the rank of P with respect to the family of mixed integer Gomory cuts is at most p.

We first recall the definition of rank. We say that P has rank k with respect to a certain family of cuts (or with respect to a certain cut generating procedure) if k is the smallest integer such that, starting with P and applying the cut generating procedure recursively k times, yields the convex hull of 0-1 points in P.

Theorem 8.7 *The rank of P with respect to each of the following families of cuts is at most p, the number of 0-1 variables:*

(a) unstrengthened lift-and-project cuts;
(b) simple disjunctive cuts;
(c) strengthened lift-and-project cuts;
(d) mixed integer Gomory cuts or, equivalently, strengthened simple disjunctive cuts.

Proof Denote, as before, $P := \text{conv}\{x \in P : x_j \in \{0, 1\}, j = 1, \dots, p\}$.

Given the sequential convexifiability of facial disjunctive programs, if we define $P_0 := P$ and for $j = 1, \dots, p$,

$$P_j := \text{conv}\{P^{j-1} \cap \{x \in \mathbb{R}^n : x_j \in \{0, 1\}\}\},$$

then

$$P_p = P_D.$$

Since P_j can be obtained from P^{j-1} by unstrengthened lift-and-project cuts, this implies (a).

From Theorems 8.4A/8.4B, at any iteration j of the above procedure, each lift-and-project cut used to generate P_j , corresponding to some basic solution of $(\text{CGLP})_j$, can also be obtained as a simple disjunctive cut associated with some nonbasic set J , with $|J| = n$. Hence the whole sequential convexification procedure can be stated in terms of simple disjunctive cuts rather than lift-and-project cuts, which implies (b).

Turning now to strengthened lift-and-project cuts, if at each iteration j of the above procedure we use strengthened rather than unstrengthened lift-and-project cuts corresponding to basic solutions of $(\text{CGLP})_j$,we obtain a set $\tilde{P}_j$ instead of P_j, with $\tilde{P}_j \subseteq P_j$. Clearly, using the same recursion as above, we end up with $\tilde{P}^p = P_D$. This proves (c).

Finally, since every strengthened lift-and-project cut corresponding to a basic solution of $(\text{CGLP})_j$ is equivalent to a mixed integer Gomory cut derived from the row corresponding to x_j of a simplex tableau with a certain nonbasic set J with $|J| = n$ (Theorem 8.5), the procedure discussed under (c) can be restated as an equivalent procedure in terms of mixed integer Gomory cuts, which proves (d). □

In [62] it was shown that the bound established in Theorem 8.7 is tight for the mixed integer Gomory cuts, by providing a class of examples with rank p.

Chapter 9
Solving $(\text{CGLP})_k$ on the LP Simplex Tableau

The major practical consequence of the correspondence established in Theorems 8.4A/8.4B is that the cut generating linear program $(\text{CGLP})_k$ need not be formulated and solved explicitly; instead, the procedure for solving it can be mimicked on the linear programming relaxation (LP) of the original mixed 0-1 problem. Apart from the fact that this replaces a large linear program with a smaller one, it also drastically reduces the number of pivots for the following reason.

A basic solution x to (LP), feasible or not, with nonbasic index set J, corresponds to a basic feasible solution w to $(\text{CGLP})_k$ with $u_0 > 0$, $v_0 > 0$, all components of α, β basic, and u_i, v_j basic for some $i \in M_1$, and $j \in M_2$, respectively, such that $M_1 \cup M_2 = J$. When $\bar{a}_{kj} \neq 0$ for all $j \in J$, the partition of J into (M_1, M_2) is unique, and so is the $(\text{CGLP})_k$ basis associated with the solution w. But when this is not the case, i.e., $\bar{a}_{kj} = 0$ for some $j \in J$ (which is the usual situation), then w has as many associated bases as there are valid partitions of J (by valid we mean that $j \in M_1$, if $\bar{a}_{kj} > 0$, $j \in M_2$ if $\bar{a}_{kj} < 0$). All these bases are degenerate, as they are associated with the same solution w, so they can be obtained from each other through degenerate pivots in the $(\text{CGLP})_k$. On the other hand, a single pivot in (LP), which replaces J with a set J' that differs from J in a single element, produces a partition (M_1', M_2') of J' which may have little in common with the partition of J. Therefore a single pivot in (LP) corresponds to a potentially long sequence of pivots in the $(\text{CGLP})_k$ to account for the transition from (M_1, M_2) to (M_1', M_2'), and these pivots are in general nondegenerate (see more details on this in the next section).

We will now describe the procedure of [33] that mimics on (LP) the optimization of $(\text{CGLP})_k$.

Let $S := \{1, \ldots, m + p\}$ and $N := \{m + p + 1, \ldots, m + p + n\}$ index the surplus variables and the structural variables in (LP), respectively, where we use s to denote the vector of surplus variables in $Ax \geq b$. With this indexing we have a direct correspondence between the variables of (LP) and the surplus variables of $\tilde{A}x \geq \tilde{b}$. Note, however, that the latter system contains the extra surplus variables

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_9

from the rows

$$x_j - s_j = 0, \quad j = m+p+1, \ldots, m+p+n$$

which of course are equal to the corresponding structural variables. The simplex tableau for (LP) is determined uniquely by the set of variables chosen to be nonbasic. If we let I and J index the set of basic and nonbasic variables, respectively, then the simplex tableau for (LP) with such a choice of basis can be written

$$\begin{aligned} x_i + \sum_{j \in N \cap J} \bar{a}_{ij} x_j + \sum_{j \in S \cap J} \bar{a}_{ij} s_j &= \bar{a}_{i0} \text{ for } i \in N \cap I \\ s_i + \sum_{j \in N \cap J} \bar{a}_{ij} x_j + \sum_{j \in S \cap J} \bar{a}_{ij} s_j &= \bar{a}_{i0} \text{ for } i \in S \cap I \end{aligned} \tag{9.1}$$

Here $\bar{a}_{ij}$ denotes the coefficient for nonbasic variable j in the row for basic variable i, and $\bar{a}_{i0}$ is the corresponding right-hand side constant. When dealing with a row of the simplex tableau (9.1) we will identify the nonbasic variables $x_{N \cap J}$ with the corresponding s_J so that we can write a row k of the tableau in the more concise form

$$x_k + \sum_{j \in J} \bar{a}_{kj} s_j = \bar{a}_{k0} \tag{9.2}$$

We start with the simple disjunctive cut $\pi s_J \geq \pi_0$ derived from the optimal simplex tableau by applying the disjunction $x_k \leq 0 \vee x_k \geq 1$ to the expression (9.2). As mentioned before, the coefficients of this cut are

$$\pi_0 := (1 - \bar{a}_{k0})\bar{a}_{k0}$$

and

$$\pi_j := \max\{\pi_j^1, \pi_j^2\}, \quad j \in J,$$

with $\pi_j^1 := (1 - \bar{a}_{k0})\bar{a}_{kj}$, $\pi_j^2 := -\bar{a}_{k0}\bar{a}_{kj}$.

We know that the lift-and-project cut $\alpha x \geq \beta$ equivalent to $\pi s_J \geq \pi_0$ corresponds to the basic solution (CGLP)$_k$ defined by (8.9). We wish to obtain the lift-and-project cut corresponding to an optimal solution to (CGLP)$_k$ by performing the improving pivots in (LP).

We start by examining a pivot on an element $\bar{a}_{ij}$, $i \neq k$ of the simplex tableau of (LP). The effect of such a pivot is to add to the cut row (9.2) the i-th row multiplied by $\gamma_j := -\bar{a}_{kj}/\bar{a}_{ij}$, and hence to replace (9.2) by

$$x_k = -\bar{a}_{k0} + \gamma_j \bar{a}_{i0} - \sum_{h \in J \setminus \{j\}} (\bar{a}_{kh} + \gamma_j \bar{a}_{ih}) s_h - \gamma_j x_i \tag{9.3}$$

Now if $0 < \bar{a}_{k0} + \gamma_j \bar{a}_{i0} < 1$, i.e. if $(-\bar{a}_{k0}/\bar{a}_{i0}) < \gamma_j < ((1 - \bar{a}_{k0})/\bar{a}_{i0})$, then we can apply the disjunction $x_k \leq 0 \vee x_k \geq 1$ to (9.3) instead of (9.2), to obtain a cut $\pi^\gamma s_{J^\gamma} \geq \pi_0^\gamma$, where $J^\gamma := (J \setminus \{j\}) \cup \{i\}$ and s_i denotes x_i . The question is how to choose the pivot element $\bar{a}_{ij}$ in order to make $\pi^\gamma s_{J^\gamma} \geq \pi_0^\gamma$ a stronger cut than $\pi s_J \geq \pi_0$, in fact as strong as possible. This choice involves two elements. First, we choose a row i, some multiple of which is to be added to row k; second, we choose a column in row i, which sets the sign and size of the multiplier. Note that we can pivot on *any* nonzero $\bar{a}_{ij}$ since we do not restrict ourselves to feasible bases.

As to the first choice, pivoting in row i, i.e. pivoting the variable x_i out of the basis, corresponds in the simplex tableau for (CGLP)$_k$ to pivoting into the basis one of the nonbasic variables u_i or v_i. Clearly, such a pivot is an improving one in terms of the objective function of (CGLP)$_k$ only if either u_i or v_i have a negative reduced cost. Below we give the expressions for r_{ui} and r_{vi}, the reduced costs of u_i and v_i respectively, in terms of the coefficients $\bar{a}_{kj}$ and $\bar{a}_{ij}$, $j \in J \cup \{0\}$.

As to the second choice, one can identify the index $j \in J$ such that pivoting on $\bar{a}_{ij}$ maximizes the improvement in the strength of the cut, by first maximizing the improvement over all $j \in J$ with $\gamma_j = -\bar{a}_{kj}/\bar{a}_{ij} > 0$, then over all $j \in J$ with $\gamma_j < 0$, and choosing the larger of the two improvements. This is done by computing two evaluation functions $f^+(\gamma)$, $f^-(\gamma)$ for the two cases, $\gamma_j > 0$ and $\gamma_j < 0$.

9.1 Computing Reduced Costs of (CGLP)$_k$ Columns for the LP Rows

In the following we will consider a basic solution (x, s) to (LP). In the simplex tableau (9.1) corresponding to this solution we let, as earlier, $B := N \cap I$ and $R := N \cap J$ denote the basic and nonbasic structural variables, respectively, and we let $P := S \cap I$ and $Q := S \cap J$ denote the basic and nonbasic surplus variables, respectively.

Lemma 9.1 *In the simplex tableau (9.1) corresponding to the basic solution (x, s), the coefficients $\bar{a}_{ij}$ for $i = 1, \dots, m + p + n$, $j \in J$, and the right-hand sides $\bar{a}_{i0}$ for $i = 1, \dots, m + p + n$ satisfy*

$$\bar{a}_{ij} = -(\tilde{A}_i \tilde{A}_J^{-1})_j \tag{9.4}$$

$$\bar{a}_{i0} = \tilde{A}_i \tilde{A}_J^{-1} \tilde{b}_J - \tilde{b}_i \tag{9.5}$$

Proof From Lemma 8.2 we know that $\tilde{A}_J$ is invertible. Let us first consider the basis matrix for (LP) and its inverse. We can write the nontrivial constraints of (LP) as

$$\begin{aligned} A_Q^B x_B \qquad\quad &+ A_Q^R x_R - s_Q = b_Q \\ A_P^B x_B - s_P &+ A_P^R x_R \qquad\;\; = b_P \end{aligned}$$

where $A = (A^B \ A^R)$ and $A^B_Q(A^R_Q)$ is the submatrix of $A^B(A^R)$ with row sets indexed by Q. The basis matrix, E, for (LP) and its inverse, E^{-1}, are

$$E = \begin{pmatrix} A^B_Q & 0 \\ A^B_P & -I \end{pmatrix} \qquad E^{-1} = \begin{pmatrix} (A^B_Q)^{-1} & 0 \\ A^B_P(A^B_Q)^{-1} & -I \end{pmatrix}$$

The constraints of $\tilde{A}x \geq \tilde{b}$ indexed by J are

$$\begin{aligned} A^B_Q x_B + A^R_Q x_R - s_Q \qquad &= b_Q \\ x_R \qquad - s_R &= 0 \end{aligned}$$

where s_R are those surplus variables corresponding to x_R.

We can thus write $\tilde{A}_J$ and its inverse, $\tilde{A}_J^{-1}$ as

$$\tilde{A}_J = \begin{pmatrix} A^B_Q & A^R_Q \\ 0 & I \end{pmatrix} \qquad \tilde{A}_J^{-1} = \begin{pmatrix} (A^B_Q)^{-1} & -(A^B_Q)^{-1}A^R_Q \\ 0 & I \end{pmatrix}$$

There are four cases to consider for the coefficients of the simplex tableau for (LP); index i can be either that of a structural variable ($i \in B$) or that of a surplus variable ($i \in P$), and likewise for index j ($j \in R$ or $j \in Q$). For each of the four cases, (9.5) is shown in [33] to hold.

For the right-hand side $\bar{a}_{i0}$ there are only two cases, depending on whether i indexes a structural or a surplus variable. Again, for each of these two cases, (9.5) is shown in [33] to hold. □

Recall that $\bar{a}_{ij}$ and $\bar{a}_{i0}$ denotes respectively the coefficient for variable j in row i and the right-hand side of row i, in the simplex tableau of (LP) for the current solution (x, s), whereas $(\bar{x}, \bar{s})$ is the *optimal* solution to (LP).

Theorem 9.2 *Let* $(\alpha, \beta, u, u_0, v, v_0)$ *be a basic feasible solution to (8.1) with* $u_0, v_0 > 0$*, all components of* α, β *basic, and the basic components of* u *and* v *indexed by* M_1 *and* M_2*, respectively. Let* $\bar{s}$ *be the vector of surplus variables of* $\tilde{A}x \geq \tilde{b}$ *corresponding to the solution* $\bar{x}$*.*

The reduced costs of u_i *and* v_i *for* $i \notin J \cup \{k\}$ *in this basic solution are, respectively*

$$\begin{aligned} r_{u_i} &= \sigma\left(-\sum_{j\in M_1}\bar{a}_{ij} + \sum_{j\in M_2}\bar{a}_{ij} - 1\right) - \sum_{j\in M_2}\bar{a}_{ij}\bar{s}_j + \bar{a}_{i0}(1-\bar{x}_k) \\ r_{v_i} &= \sigma\left(+\sum_{j\in M_1}\bar{a}_{ij} - \sum_{j\in M_2}\bar{a}_{ij} - 1\right) - \sum_{j\in M_1}\bar{a}_{ij}\bar{s}_j + \bar{a}_{i0}\bar{x}_k \end{aligned} \tag{9.6}$$

where

$$\sigma = \frac{\sum_{j \in M_2} \bar{a}_{kj}\bar{s}_j - \bar{a}_{k0}(1-\bar{x}_k)}{1 + \sum_{j \in J} |\bar{a}_{kj}|}$$

Proof If we restrict (8.1) to the basic variables plus u_i and v_i, and eliminate α, β, we obtain the system

$$\begin{aligned}
u_{M_1}\tilde{A}_{M_1} + u_i\tilde{A}_i - u_0 e_k &= v_{M_2}\tilde{A}_{M_2} + v_i\tilde{A}_i + v_0 e_k\\
u_{M_1}\tilde{b}_{M_1} + u_i\tilde{b}_i \qquad &= v_{M_2}\tilde{b}_{M_2} + v_i\tilde{b}_i + v_0\\
\sum_{j \in M_1} u_j + u_i + u_0 + \sum_{j \in M_2} v_j + v_i + v_0 &= 1
\end{aligned}$$

The first two equations can be rewritten

$$\begin{aligned}
(u_{M_1}, -v_{M_2})\tilde{A}_J + (u_i - v_i)\tilde{A}_i &= (u_0 + v_0)e_k\\
(u_{M_1}, -v_{M_2})\tilde{b}_J + (u_i - v_i)\tilde{b}_i &= v_0
\end{aligned}$$

From Lemma 8.2 we know that $\tilde{A}_J$ is invertible, so

$$\begin{aligned}
(u_{M_1}, -v_{M_2}) &= (u_0 + v_0)e_k\tilde{A}_J^{-1} \quad - (u_i - v_i)\tilde{A}_i\tilde{A}_J^{-1}\\
v_0 &= (u_0 + v_0)e_k\tilde{A}_J^{-1}\tilde{b}_J - (u_i - v_i)(\tilde{A}_i\tilde{A}_J^{-1}\tilde{b}_J - \tilde{b}_i)
\end{aligned} \tag{9.7}$$

Now, using Lemma 9.1 we can identify the expressions $-(e_k\tilde{A}_J^{-1})_j$ and $-(\tilde{A}_i\tilde{A}_J^{-1})_j$ with the coefficients $\bar{a}_{kj}$ (since $\tilde{A}_k = e_k$) and $\bar{a}_{ij}$ in the simplex tableau of (LP) for the basic solution with variables indexed by J being nonbasic. Likewise we can identify the expressions $e_k\tilde{A}_J^{-1}\tilde{b}_J$ and $\tilde{A}_i\tilde{A}_J^{-1}\tilde{b}_J - \tilde{b}_i$ with the right-hand side constants $\bar{a}_{k0}$ (since $\tilde{b}_k = 0$) and $\bar{a}_{i0}$ of the simplex tableau. With this substitution we have that

$$\begin{aligned}
u_j &= -(u_0 + v_0)\bar{a}_{kj} + (u_i - v_i)\bar{a}_{ij} \text{ for } j \in M_1\\
v_j &= (u_0 + v_0)\bar{a}_{kj} - (u_i - v_i)\bar{a}_{ij} \quad \text{for } j \in M_2\\
v_0 &= (u_0 + v_0)\bar{a}_{k0} - (u_i - v_i)\bar{a}_{i0}
\end{aligned} \tag{9.8}$$

We can now write the normalization constraint as

$$1 = \sum_{j \in M_1} u_j + \sum_{j \in M_2} v_j + u_i + v_i + u_0 + v_0$$

(substituting u_j and v_j from (9.8))

$$
\begin{aligned}
&= \sum_{j\in M_1} \big(-(u_0+v_0)\bar{a}_{kj} + (u_i - v_i)\bar{a}_{ij}\big) \\
&\quad + \sum_{j\in M_2} \big((u_0+v_0)\bar{a}_{kj} - (u_i - v_i)\bar{a}_{ij}\big) \\
&\quad + u_i + v_i + u_0 + v_0 \\
&= (u_0+v_0)\left(-\sum_{j\in M_1}\bar{a}_{kj} + \sum_{j\in M_2}\bar{a}_{kj} + 1\right) \\
&\quad + (u_i - v_i)\left(\sum_{j\in M_1}\bar{a}_{ij} - \sum_{j\in M_2}\bar{a}_{ij}\right) + u_i + v_i.
\end{aligned}
$$

Since (9.8) is satisfied for the current basic solution with $u_i = v_i = 0$ and since $u_{M_1}, v_{M_2} \geq 0$, it follows from (9.8) that $j \in M_1 \Rightarrow \bar{a}_{kj} \leq 0$ and $j \in M_2 \Rightarrow \bar{a}_{kj} \geq 0$, so

$$
-\sum_{j\in M_1}\bar{a}_{kj} + \sum_{j\in M_2}\bar{a}_{kj} = \sum_{j\in J}|\bar{a}_{kj}|.
$$

We thus have

$$
u_0 + v_0 = \frac{1 - (u_i - v_i)\left(\sum_{j\in M_1}\bar{a}_{ij} - \sum_{j\in M_2}\bar{a}_{ij}\right) - u_i - v_i}{1 + \sum_{j\in J}|\bar{a}_{kj}|} \tag{9.9}
$$

We can now write the objective function of (CGLP)$_k$ in terms of u_i and v_i as

$$
\begin{aligned}
\alpha\bar{x} - \beta &= v_{M_2}(\tilde{A}_{M_2}\bar{x} - \tilde{b}_{M_2}) + v_i(\tilde{A}_i\bar{x} - \tilde{b}_i) \\
&\quad + v_0(e_k\bar{x} - 1)
\end{aligned}
$$

(use that $\bar{s}_{M_2} = \tilde{A}_{M_2}\bar{x} - \tilde{b}_{M_2}$ and $\bar{s}_i = \tilde{A}_i - \tilde{b}_i$)

$$
= v_{M_2}\bar{s}_{M_2} + v_i\bar{s}_i + v_0(\bar{x}_k - 1)
$$

(substitute for v_{M_2} and v_0 using (9.8))

$$
\begin{aligned}
&= (u_0+v_0)\sum_{j\in M_2}\bar{a}_{kj}\bar{s}_j - (u_i - v_i)\sum_{j\in M_2}\bar{a}_{ij}\bar{s}_j + v_i\bar{s}_i \\
&\quad + (u_0+v_0)\bar{a}_{k0}(\bar{x}_k - 1) - (u_i - v_i)\bar{a}_{i0}(\bar{x}_k - 1)
\end{aligned}
$$

$$= (u_0 + v_0)\left(\sum_{j\in M_2} \bar{a}_{kj}\bar{s}_j - \bar{a}_{k0}(1-\bar{x}_k)\right)$$
$$+ (u_i - v_i)\left(-\sum_{j\in M_2} \bar{a}_{ij}\bar{s}_j + \bar{a}_{i0}(1-\bar{x}_k)\right) + v_i\bar{s}_i$$

If we substitute for $(u_0 + v_0)$ from (9.9) and use the definition of σ we obtain

$$\alpha\bar{x} - \beta = \sigma + u_i\left(-\sigma\sum_{j\in M_1}\bar{a}_{ij} + \sigma\sum_{j\in M_2}\bar{a}_{ij} - \sigma\right.$$
$$\left. - \sum_{j\in M_2}\bar{a}_{ij}\bar{s}_j + \bar{a}_{i0}(1-\bar{x}_k)\right)$$
$$+ v_i\left(+\sigma\sum_{j\in M_1}\bar{a}_{ij} - \sigma\sum_{j\in M_2}\bar{a}_{ij} - \sigma\right.$$
$$\left. + \sum_{j\in M_2}\bar{a}_{ij}\bar{s}_j - \bar{a}_{i0}(1-\bar{x}_k) + \bar{s}_i\right).$$

Substituting for $\bar{s}_i$ from $\bar{s}_i = \bar{a}_{i0} - \sum_{j\in M_1}\bar{a}_{ij}\bar{s}_j - \sum_{j\in M_2}\bar{a}_{ij}\bar{s}_j$, replaces the expression in the parentheses following v_i by $\sigma\sum_{j\in M_1}\bar{a}_{ij} - \sigma\sum_{j\in M_2}\bar{a}_{ij} - \sigma - \sum_{j\in M_1}\bar{a}_{ij}\bar{a}_j + \bar{a}_{i0}\bar{x}_k$. We can then read the reduced costs r_{u_i} and r_{v_i} as the coefficients of u_i and v_i. □

For a given partition (M_1, M_2) of J, the expressions for the reduced costs depend only linearly on the tableau coefficients; thus the reduced costs for all nonbasic variables of $(\text{CGLP})_k$ can be computed by one multiplication with the basis inverse [106].

9.2 Computing Evaluation Functions for the LP Columns

Next we turn to the search for the optimal pivot column, i.e., the computation of the evaluation functions $f^+(\gamma)$, $f^-(\gamma)$.

Theorem 9.3 *The pivot column in row i of the (LP) simplex tableau that is most improving with respect to the cut from row k, is indexed by that $l^* \in J$ that minimizes $f^+(\gamma_l)$ if $\bar{a}_{kl}\bar{a}_{il} < 0$ or $f^-(\gamma_l)$ if $\bar{a}_{kl}\bar{a}_{il} > 0$, over all $l \in J$ that satisfy $\frac{-\bar{a}_{k0}}{\bar{a}_{i0}} < \gamma_l <$*

$\frac{1-\bar{a}_{k0}}{\bar{a}_{i0}}$, *where* $\gamma_l := -\frac{\bar{a}_{kl}}{\bar{a}_{il}}$, *and for* $\gamma \geq 0$,

$$f^+(\gamma) := \frac{\sum_{j\in J}(-(\bar{a}_{k0}+\gamma\bar{a}_{i0})\bar{a}_{kj} + \max\{\bar{a}_{kj}, -\gamma\bar{a}_{ij}\})\bar{x}_j - (1-\bar{a}_{k0}-\gamma\bar{a}_{i0})\bar{a}_{k0}}{1+|\gamma|+\sum_{j\in J}|\bar{a}_{kj}+\gamma\bar{a}_{ij}|},$$

and for $\gamma \leq 0$,

$$f^-(\gamma) := \frac{\sum_{j\in J}(-(\bar{a}_{k0}+\gamma\bar{a}_{i0})\bar{a}_{kj} + \max\{\bar{a}_{kj}+\gamma\bar{a}_{ij}, 0\})\bar{x}_j - (1-\bar{a}_{k0})(\bar{a}_{k0}+\gamma\bar{a}_{i0})}{1+|\gamma|+\sum_{j\in J}|\bar{a}_{kj}+\gamma\bar{a}_{ij}|}$$

Proof Consider row k and row i of the (LP) simplex tableau

$$x_k + \sum_{j\in J}\bar{a}_{kj}s_j = \bar{a}_{k0} \qquad x_i + \sum_{j\in J}\bar{a}_{ij}s_j = \bar{a}_{i0}.$$

If we add row i to row k with weight $\gamma \in \mathbb{R}$ we obtain the composite row

$$x_k + \gamma x_i + \sum_{j\in J}(\bar{a}_{kj}+\gamma\bar{a}_{ij})s_j = \bar{a}_{k0}+\gamma\bar{a}_{i0}. \tag{9.10}$$

From (9.10) we can derive a simple disjunctive cut $\pi^\gamma s_J \geq \pi_0^\gamma$ using the disjunction $x_k \leq 0 \vee x_k \geq 1$ if the right-hand side of (9.10) satisfies $0 < \bar{a}_{k0}+\gamma\bar{a}_{i0} < 1$, i.e., if γ satisfies $\frac{-\bar{a}_{k0}}{\bar{a}_{i0}} < \gamma < \frac{1-\bar{a}_{k0}}{\bar{a}_{i0}}$.

A pivot on column l in row i has the effect of adding $\gamma_l = -\frac{\bar{a}_{kl}}{\bar{a}_{il}}$ times row i to row k. For this value of γ, (9.10) becomes the k-th row of the simplex tableau resulting from the pivot. We want to identify the column l such that the simple disjunctive cut, $\pi^\gamma s_J \geq \pi_0^\gamma$ we derive from the composite row (9.10) with $\gamma = \gamma_l$ minimizes $\pi^\gamma \bar{s}_J - \pi_0^\gamma$.

For any γ in the interval $\frac{-\bar{a}_{k0}}{\bar{a}_{i0}} < \gamma < \frac{1-\bar{a}_{k0}}{\bar{a}_{i0}}$, the simple disjunctive cut from the disjunction $x_k \geq 0 \vee x_k \geq 0$ applied to (9.10) has coefficients

$$\pi_i^\gamma = \max\{(1-\bar{a}_{k0}-\gamma\bar{a}_{i0})\gamma, -(\bar{a}_{k0}+\gamma\bar{a}_{i0})\gamma\}$$
$$\pi_j^\gamma = \max\left\{(1-\bar{a}_{k0}-\gamma\bar{a}_{i0})(\bar{a}_{kj}+\gamma\bar{a}_{ij}), -(\bar{a}_{k0}+\gamma\bar{a}_{i0})(\bar{a}_{kj}+\gamma\bar{a}_{ij})\right\} \text{ for } j \in J$$
$$\pi_0^\gamma = (1-\bar{a}_{k0}-\gamma\bar{a}_{i0})(\bar{a}_{k0}+\gamma\bar{a}_{i0})$$

Given the simple disjunctive cut $\pi_i^\gamma x_i + \pi^\gamma s_J \geq \pi_0^\gamma$, there exists a corresponding solution $(\alpha, \beta, u, u_0, v, v_0)$ to (CGLP)$_k$ given by (8.9). The corresponding π^1 and π^2 in (8.8) are

$$\begin{array}{ll} \pi_i^{\gamma,1} = (1-\bar{a}_{k0}-\gamma\bar{a}_{i0})\gamma & \pi_j^{\gamma,1} = (1-\bar{a}_{k0}-\gamma\bar{a}_{i0})(\bar{a}_{kj}+\gamma\bar{a}_{ij}) \\ \pi_i^{\gamma,2} = -(\bar{a}_{k0}+\gamma\bar{a}_{i0})\gamma & \pi_j^{\gamma,2} = -(\bar{a}_{k0}+\gamma\bar{a}_{i0})(\bar{a}_{kj}+\gamma\bar{a}_{ij}) \end{array} \quad \text{for } j \in J$$

Then using (8.9) we have that the components (u, u_0, v, v_0) of this solution satisfy

$$\begin{aligned} u_i + v_i &= (\pi_i^{\gamma} - \pi_i^{\gamma,1}) + (\pi_i^{\gamma} - \pi_i^{\gamma,2}) \\ &= |\pi_i^{\gamma,1} - \pi_i^{\gamma,2}| = |\gamma| \\ u_j + v_j &= (\pi_j^{\gamma} - \pi_j^{\gamma,1}) + (\pi_j^{\gamma} - \pi_j^{\gamma,2}) \\ &= |\pi_j^{\gamma,1} - \pi_j^{\gamma,2}| = |\bar{a}_{kj} + \gamma\bar{a}_{ij}| \qquad \text{for } j \in J \\ u_0 + v_0 &= (1 - \bar{a}_{k0} - \gamma\bar{a}_{i0}) + (\bar{a}_{k0} + \gamma\bar{a}_{i0}) = 1 \end{aligned} \tag{9.11}$$

The solution to $(\text{CGLP})_k$ corresponding to the cut $\pi_i^{\gamma} x_i + \pi^{\gamma} s_J \geq \pi_0^{\gamma}$ satisfies all the constraints of $(\text{CGLP})_k$ except the normalization constraint. To also satisfy the normalization constraint we have to scale the cut by the sum of the multipliers u, v, u_0 and v_0, which from (9.11) becomes

$$\sum_{i=1}^{m+p+n} u_i + u_0 + \sum_{i=1}^{m+p+n} v_i + v_0 = 1 + |\gamma| + \sum_{j\in J} |\bar{a}_{kj} + \gamma\bar{a}_{ij}|$$

The objective function of $(\text{CGLP})_k$ corresponding to this cut is thus obtained by scaling the function $\pi_i^{\gamma}\bar{x}_i + \pi^{\gamma}\bar{s}_J - \pi_0^{\gamma}$ by the sum of multipliers, namely

$$\frac{\pi_i^{\gamma}\bar{x}_i + \pi^{\gamma}\bar{s}_J - \pi_0^{\gamma}}{1 + |\gamma| + \sum_{j\in J} |\bar{a}_{kj} + \gamma\bar{a}_{ij}|} \tag{9.12}$$

The expressions for π^{γ} and π_0^{γ} involve terms with γ^2. We can eliminate such terms by subtracting π_i^{γ} times row i from the cut $\pi_i^{\gamma} s_J \geq \pi_0^{\gamma}$. The result depends on the sign of γ, since $\pi_i^{\gamma} = (1 - \bar{a}_{k0} - \gamma\bar{a}_{i0})\gamma$ if $\gamma > 0$, and $\pi_i^{\gamma} = -(\bar{a}_{k0} + \gamma\bar{a}_{i0})\gamma$ if $\gamma < 0$. Thus,

$$\begin{aligned} &\pi_j^{\gamma+} = \max\{(1 - \bar{a}_{k0} - \gamma\bar{a}_{i0})\bar{a}_{kj}, -(\bar{a}_{k0} + \gamma\bar{a}_{i0})\bar{a}_{kj} - \gamma\bar{a}_{ij}\} \\ \gamma > 0: \quad &\phantom{\pi_j^{\gamma+}} = -(\bar{a}_{k0} + \gamma\bar{a}_{i0})\bar{a}_{kj} + \max\{\bar{a}_{kj}, -\gamma\bar{a}_{ij}\} \\ &\pi_0^{\gamma+} = (1 - \bar{a}_{k0} - \gamma\bar{a}_{i0})\bar{a}_{k0} \\ &\pi_j^{\gamma-} = \max\{(1 - \bar{a}_{k0} - \gamma\bar{a}_{i0})\bar{a}_{kj} + \gamma\bar{a}_{ij}, -(\bar{a}_{k0} + \gamma\bar{a}_{i0})\bar{a}_{kj})\} \\ \gamma < 0: \quad &\phantom{\pi_j^{\gamma-}} = -(\bar{a}_{k0} + \gamma\bar{a}_{i0})\bar{a}_{kj} + \max\{\bar{a}_{kj} + \gamma\bar{a}_{ij}, 0\} \\ &\pi_0^{\gamma-} = (1 - \bar{a}_{k0})(\bar{a}_{k0} + \gamma\bar{a}_{i0}) \end{aligned}$$

Using $\pi^{\gamma+}\bar{s}_J - \pi_0^{\gamma+}$ and $\pi^{\gamma-}\bar{s}_J - \pi_0^{\gamma-}$ in place of $\pi_i^{\gamma}\bar{x}_i + \pi^{\gamma}\bar{s}_J - \pi_0^{\gamma}$ in (9.12), the objective function of (CGLP)$_k$ corresponding to these cuts becomes

$$\frac{\pi^{\gamma+}\bar{s}_J - \pi_0^{\gamma+}}{1 + |\gamma| + \sum_{j\in J} |\bar{a}_{kj} + \gamma\bar{a}_{ij}|}$$

and

$$\frac{\pi^{\gamma-}\bar{s}_J - \pi_0^{\gamma-}}{1 + |\gamma| + \sum_{j\in J} |\bar{a}_{kj} + \gamma\bar{a}_{ij}|}$$

If we insert the expressions for $\pi^{\gamma+}$, $\pi_0^{\gamma+}$, $\pi^{\gamma-}$ and $\pi_0^{\gamma-}$ into the above, we obtain $f^+(\gamma)$ and $f^-(\gamma)$ respectively, as stated in the theorem. $\square$

Next we sketch an algorithm for finding an optimal lift-and-project cut by pivoting in the simplex tableau of (LP).

9.3 Generating Lift-and-Project Cuts by Pivoting in the LP Tableau

Step 0. Solve (LP). Let $\bar{x}$ be an optimal solution and let k be such that $0 < \bar{x}_k < 1$.

Step 1. Let J index the nonbasic variables in the current basis. For each row $i \neq k$ of the simplex tableau of (LP), compute the reduced costs r_{u_i} with $M_1 = \{j \in J : \bar{a}_{kj} < 0 \vee (\bar{a}_{kj} = 0 \wedge \bar{a}_{ij} > 0)\}$, $M_2 = J \setminus M_1$, and r_{v_i} with $M_1 = \{j \in J : \bar{a}_{kj} < 0 \vee (\bar{a}_{kj} = 0 \wedge \bar{a}_{ij} < 0)\}$, $M_2 = J \setminus M_1$ of u_i, v_i, according to (9.6).

Step 2. Let i_* be a row with $r_{u_{i_*}} < 0$ or $r_{v_{i_*}} < 0$. If no such row exists, stop.

Step 3. Identify the most improving pivot column J_* in row i_* by minimizing $f^+(\gamma_j)$ over all $j \in J$ with $\gamma_j > 0$ and $f^-(\gamma_j)$ over all $j \in J$ with $\gamma_j < 0$ and choosing the more negative of these two values.

Step 4. Pivot on $\bar{a}_{i_* j_*}$ and go to Step 1.

At termination, the simple disjunctive cut from the transformed row k is a quasi-optimal lift-and-project cut; the mixed-integer Gomory cut from row k is a quasi-optimal strengthened lift-and-project cut. Next we explain the term "quasi-optimal".

When we compute the reduced costs in Step 1, we create a partition (M_1, M_2) of J according to Theorem 8.4B. When $\bar{a}_{kj} = 0$ for some $j \in J$ we are free to choose whether to assign j to M_1 or M_2. By assigning such a j to M_1 if $\bar{a}_{ij} > 0$ and to M_2 otherwise in the case of r_{u_i}, we make sure that the corresponding (CGLP) basis permits a nondegenerate pivot in column u_i. Thus, if $r_{u_i} < 0$, this pivot improves the solution. This is because with such a choice of M_1 and M_2 it is possible to increase u_i by a small amount in (9.8) without driving any of the u_j and v_j negative.

Equivalently for r_{v_i}. Hence, when step 2 does not find a negative reduced cost, there is no improving pivot in the simplex tableau of (LP).

When the algorithm comes to a point where Step 2 finds no row $i \neq k$ with $r_{u_i} < 0$ or $r_{v_i} < 0$, i.e. all the reduced costs of the large tableau are nonnegative, we could conclude that the solution is optimal if the reduced costs had all been calculated with respect to the same basis of the large tableau, i.e. with respect to the same partition (M_1, M_2). However, this is not the case, since the attempt to find a pivot that improves the cut from row k as much as possible makes us use a different partition (M_1, M_2) for every row i, as explained above. While in the absence of 0 entries in row k the partition (M_1, M_2) is unique (the same for all i), the presence of 0's in row k allows us to use different partitions for different rows, thereby gaining in efficiency. When all the reduced costs calculated in this way are nonnegative, then in order to make sure that the cut is optimal, we would have to recalculate the reduced costs from a unique basis of the large tableau, i.e. a unique partition (M_1, M_2). However, experience shows that the cuts obtained this way are on the average of roughly the same strength as those obtained by solving explicitly the (CGLP). That's why we call the cuts quasi-optimal.

9.4 Using Lift-and-Project to Choose the Best Mixed Integer Gomory Cut

The above algorithm for finding an optimal lift-and-project cut through a sequence of pivots in the simplex tableau of (LP) can also be interpreted as an algorithm for finding the strongest mixed integer Gomory cut from a given row among all (feasible or infeasible) simplex tableaus. The first pivot in the sequence results in the replacement of the mixed integer Gomory (MIG) cut from the row associated with x_k in the optimal simplex tableau of (LP) (briefly row k) with the MIG cut from the same row k of another simplex tableau (not necessarily feasible), the one resulting from the pivot. The new cut is guaranteed to be more violated by the optimal LP solution $\bar{x}$ than was the previous cut. Each subsequent pivot results again in the replacement of the MIG cut from row k of the current tableau with a MIG cut from row k of a new tableau, with a guaranteed improvement of the cut. This algorithm is essentially an exact version of the heuristic procedure for improving mixed integer Gomory cuts described in Sect. 1.4.

The nature of this improvement is best understood by viewing the MIG cut as a simple disjunctive cut, and considering the strengthening of the disjunction—a dichotomy between two inequalities—through the addition of multiples of other inequalities to either term, before actually taking the disjunction.

Here is a brief illustration of what this strengthening procedure means, on an example small enough for the purpose, yet hard enough for standard cuts, the Steiner triple problem with 15 variables and 35 constraints (problem `stein15` of [46]). To mitigate the effects of symmetry, we replaced the objective function of $\sum_j x_j$ by $\sum_j jx_j$.

The linear programming optimum is

$$\bar{x} = (1, 1, 1, 1, 1, 0.5, 0.5, 0.5, 0.5, 0.5, 0, 0, 0, 0, 0)$$

with a value of 35. The integer optimum is

$$x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0),$$

with value 45.

Generating one mixed integer Gomory cut from each of the five fractional variables and solving the resulting linear program yields a solution x^1 of value 39. Iterating this procedure 10 times, each time generating one MIG cut from each of the fractional variables of the current solution and solving the resulting linear program, yields the solution

$$x^{10} = (0.97, 1, 1, 0.93, 1, 0.73, 0.64, 0.33, 0.61, 0.86, 0.67, 0, 0.35, 0.55, 0.32)$$

with a value of 42.73.

But if instead of using the five MIG cuts as they are, we first improve them by our pivoting algorithm, then use the five improved cuts in place of the original ones, we get a solution $\tilde{x}^1$ of value 41.41. If we then iterate this procedure 10 times, using every time the improved cuts in place of the original ones, we obtain the solution

$$\tilde{x}^{10} = (1, 1, 1, 1, 1, 1, 1, 0.94, 0.72, 0.29, 0, 0, 0, 0, 0)$$

with a value of 44.85.

The difference between x^{10} and $\tilde{x}^{10}$ is striking. However, even more striking are the details. Here we limit ourselves to discussing the first out of the ten iterations of the above procedure. Here is how the improving pivots affect the amount of violation, defined as $\beta - \alpha\bar{x}$ for the cut $\alpha x \geq \beta$, and the distance, meaning the Euclidean distance of $\bar{x}$ from the cut hyperplane:

	Violation	Distance
Cut from x_6: original MIG	0.0441	0.1443
optimal (after 3 pivots)	0.0833	0.2835
Cut from x_7: original MIG	0.0625	0.1768
optimal (after 1 pivot)	0.0714	0.2085
Cut from x_8: original MIG	0.0577	0.2023
optimal (after 1 pivot)	0.0833	0.2835
Cut from x_9: original MIG	0.0500	0.1744
optimal (after 3 pivots)	0.0833	0.2887
Cut from x_{10}: original MIG	0.0500	0.1744
optimal (after 4 pivots)	0.0833	0.2835

In the process of strengthening the 5 MIG cuts in the first iteration, 12 new cuts are generated. If, instead of replacing the original MIG cuts with the improved ones, we keep all the cuts generated and solve the problem with all 17 cuts (the 5 initial ones plus the 12 improved ones), we get exactly the same solution $\tilde{x}^1$ as with the 5 final improved cuts only: the original MIG cuts as well as the intermediate cuts resulting from the improving pivots (except for the last one) are made redundant by the 5 final improved cuts.

A similar behavior is exhibited on the problems `stein27` (with 27 variables and 117 constraints) and `stein45` (45 variables, 330 constraints).

Since the algorithm described here starts with a MIG cut from the optimal simplex tableau and stops with a MIG cut from another (usually infeasible) simplex tableau, one may ask what is the role of lift-and-project theory in this process? The answer is that it provides the guidance needed to get from the first MIG cut to the final one. It provides the tools, in the form of the reduced costs from the (CGLP) tableau of the auxiliary variables u_i and v_i , for identifying a pivot that is guaranteed to improve the cut, if one exists. Over the last several decades there have been numerous attempts to improve mixed integer Gomory cuts by deriving them from tableau rows combined in different ways, but none of these attempts has succeeded in defining a procedure that is *guaranteed* to find an improved cut when one exists. The lift-and-project approach has done just that.

In the last three chapters we discussed lift-and-project procedures for generating cutting planes for mixed integer programs from disjunctions of the form $x_j \leq 0$ or $x_j \geq 1$, called split disjunctions. However, the family of split disjunctions is more general: any disjunction of the form $\pi x \leq pi_0 \vee \pi x \geq \pi_0 + 1$, where π is an integer vector in the space of integer-constrained variables and π_0 is an integer, is called a split disjunction. It is not hard to see, that the procedures and results discussed in these three chapters apply to this more general class of split disjunctions. Indeed, we have

Proposition 9.4 *A split disjunction*

$$\pi x \leq \pi_0 \vee \pi x \geq \pi_0 + 1$$

applied to an instance (MIP) is equivalent to a disjunction $y \leq 0 \vee y \geq 1$ applied to the instance (MIP) amended with the constraint $y = \pi x - \pi_0$.

Proof Obvious. □

Chapter 10
Implementation and Testing of Variants

The discovery of the possibility of generating L&P cuts through pivoting in the LP tableau, without recourse to the higher-dimensional (CGLP), has opened the door to the introduction of this class of cuts into commercial optimizers. The first among those was XPRESS, and the early results, reported by Michael Perregaard [107] who did the implementation, can be summarized as follows:

- XPRESS generates L&P cuts in rounds of 50 and restricts to 10 the number of pivots used to generate a cut.
- Solving the CGLP implicitly through pivoting in the LP tableau requires 5% of the number of iterations and 1.5% of the time it takes to solve it explicitly.
- XPRESS uses L&P cuts in addition to, not in place of, Gomory cuts. The extra cost of generating them is unjustified for easy problems, but pays off handily for the harder ones.

The next implementation, in the COIN-OR open source framework [55], is due to Pierre Bonami and is publicly available [53] since 2006. It implements the procedures described in [18], which we will discuss in detail below. There are two additional commercial implementations that we are aware of, one in MOPS by Franz Wesselmann [121] and the other one in CPLEX by Andrea Tramontani [117]. Bonami's implementation [53] served as a basis for a large scale computational experiment discussed in [18], comparing several versions of lift-and-project.The main focus of [18] is a deeper understanding of the correspondence between a given pivot in the LP tableau and the corresponding block pivot (sequence of pivots) in the (CGLP) tableau.

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_10

10.1 Pivots in the LP Tableau Versus Block Pivots in the CGLP Tableau

The central fact of this correspondence is that: a *single pivot* in the LP corresponds to a *sequence of (mostly nondegenerate) pivots* in the CGLP. Furthermore, the length of this sequence can be quite significant, bounded only by $|J|$, and it depends on the value of γ for which $f^+(\gamma)$ or $f^-(\gamma)$ attains its minimum. Suppose, for the sake of simplicity, that $\min\{f^+(\gamma), f^-(\gamma)\}$ is attained by $f^+(\gamma)$ for some value $\gamma_{j_t} > 0$ of γ. (The situation is analogous if the minimum is attained by $f^-(\gamma)$.) This means that as γ increases from 0, $f^+(\gamma)$ decreases from $f^+(0)$ until it reaches the value of $f^+(\gamma_{j_t})$, after which it increases.

Let

$$x_k + \gamma x_i + \sum \left((\bar{a}_{kj} + \gamma \bar{a}_{ij})x_j : j \in J\right) = \bar{a}_{k0} + \gamma \bar{a}_{i0} \qquad (10.1)_\gamma$$

be the combined source row resulting from a pivot in row i, where γ is the generic notation for $\gamma_\ell = -\frac{\bar{a}_{k\ell}}{\bar{a}_{i\ell}}$ for some $\ell \in J$. Let $\gamma_{j_1}, \ldots, \gamma_{j_t}$ be the sequence of non-decreasing values of γ for which some coefficient $\bar{a}_{kj} + \gamma \bar{a}_{ij}$ of $(10.1)_\gamma$ becomes 0, and let $f^+(\gamma)$ attain its minimum for $\gamma = \gamma_{j_t}$.

Theorem 10.1 ([18]) *If $(10.1)_{\gamma_{j_t}}$ is the combined source row resulting from a pivot on $\bar{a}_{ij_t}$ and $\pi x \geq \pi_0$ is the cut from the disjunction $x_k + \gamma_{j_t} x_i \leq 0 \vee x_k + \gamma_{j_t} x_i \geq 1$, then the equivalent L&P cut $\alpha x \geq \beta$ corresponds to a basic feasible solution to $(CGLP)_k$ resulting from a sequence of t pivots defined as follows.*

(a) The first pivot introduces into the basis the nonbasic variable u_i or v_i (the one with negative reduced cost) corresponding to the x_i row of the LP tableau, and removes from the basis the variable u_{j_1} or v_{j_1} corresponding to column j_1 of the LP tableau

(b) Each pivot except for the first and the last one, exchanges

(b1) a nonbasic variable u_{j_h} for a basic variable $u_{j_{h+1}}$ if $\bar{a}_{kj_h}$ switches its sign from positive to negative when γ is increased beyond γ_{j_h} for $h = 2, \ldots, t-1$; or

(b2) a nonbasic variable v_{j_h} for a basic variable $v_{j_{h+1}}$ if $\bar{a}_{kj_h}$ switches its sign from negative to positive when γ is increased beyond γ_{j_h} $(h = 2, \ldots, t-1)$.

(c) The last pivot exchanges the nonbasic variable $u_{j_{t-1}}$ (or $v_{j_{t-1}}$) with the basic variable u_{j_t} (or v_{j_t}) corresponding to column j_t of the LP tableau.

Throughout the sequence, the pivot corresponding to the transition from γ_{j_h} to $\gamma_{j_{h+1}}$ is nondegenerate if and only if $\gamma_{j_h} \neq \gamma_{j_{h+1}}$.

Proof Let (M_1, M_2) be a valid partition of J, the nonbasic set of the starting LP tableau, and let (M_1', M_2') be a valid partition of the nonbasic set $J' := (J \cup \{i\}) \setminus \{j_t\}$ of the LP tableau resulting from the pivot on $\bar{a}_{ij_t}$. Whenever (M_1', M_2') differs from

(M_1, M_2) in more than one element, the two associated solutions to $(\text{CGLP})_k$ are not adjacent. Thus executing the single pivot in $(\text{CGLP})_k$ that corresponds to the pivot on $\bar{a}_{ij_t}$, i.e. exchanging the nonbasic variable u_i (or v_i) for the basic variable u_{j_t} (or v_{j_t}) would yield an infeasible CGLP solution that is not associated with any valid cut. To obtain a basic *feasible* solution to $(\text{CGLP})_k$ one needs to perform a sequence of primal feasible pivots. This sequence is specified by the sequence of sign changes of coefficients of the combined row $(10.1)_\gamma$ as a result of the increase of γ from 0 to γ_{j_t}. Each sign change corresponds to moving the corresponding index from M_1 to M_2 or vice versa, which in turn corresponds to replacing a basic u_j with the corresponding v_j or vice versa. However, since pivots only allow for exchanging two u_j variables among themselves or two v_j variables among themselves, replacing a certain basic u_{j_*} with v_{j_*} takes two pivots, not one: first u_{j_*} is pivoted out (in exchange for some nonbasic u_h), then v_{j_*} is pivoted in (in exchange for some basic v_ℓ). The result is the sequence of pivots described in the Theorem. To see why the pivots in the sequence are nondegenerate if and only if $\gamma_{j_h} \neq \gamma_{j_{h+1}}$, note that each pivot in the sequence corresponds to a different value of $f^+(\gamma)$, i.e. causes a different amount of change in the CGLP objective, if and only if this condition holds. □

Next we illustrate the correspondence between pivots in the two tableaux on a small example, a set covering instance with 9 variables and 13 inequalities, obtained from the well known instance `stein9` by adding the inequality $\sum x_j \geq 4$.

The structural variables are $x_1, \dots, x_9$, the surplus variables associated with the inequalities are $s_1, \dots, s_{13}$ while those associated with the upper and lower bounds on the structurals are $s_{14}, \dots, s_{22}$ and $s_{23}, \dots, s_{31}$, respectively. The variables of $(\text{CGLP})_3$, after eliminating α and β, are u_0, v_0 and u_i, v_i, for $i = 1, \dots, 31$, i.e. each pair u_i, v_i gets the same subscript as the surplus variable s_i of the corresponding inequality of $\tilde{A}x \geq \tilde{b}$. The optimal solution to the LP relaxation for the objective function we chose $(\min \sum jx_j)$ is

$$\bar{x} = \left(1, \tfrac{2}{3}, \tfrac{2}{3}, \tfrac{2}{3}, \tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3}, 0, 0\right).$$

We choose as source row for a cut the one corresponding to basic variable x_3:

$$x_3 + \tfrac{1}{3}s_4 + \tfrac{1}{3}s_5 + \tfrac{1}{3}s_6 + \tfrac{2}{3}s_8 - \tfrac{1}{3}s_9 - \tfrac{2}{3}s_{13} - \tfrac{2}{3}s_{14} + \tfrac{2}{3}s_{30} - \tfrac{1}{3}s_{31} = \tfrac{2}{3}$$

The intersection cut from $0 \leq x_3 \leq 1$, if generated, would be

$$\tfrac{1}{9}s_4 + \tfrac{1}{9}s_5 + \tfrac{1}{9}s_6 + \tfrac{2}{9}s_8 + \tfrac{2}{9}s_9 + \tfrac{4}{9}s_{13} + \tfrac{4}{9}s_{14} + \tfrac{2}{9}s_{30} + \tfrac{2}{9}s_{31} \geq \tfrac{2}{9}$$

This cut would have to be scaled in order to obtain the corresponding lift-and-project cut (to satisfy the normalization constraint, in this case $\sum u_i + u_0 + \sum v_i + v_0 = 1$) by a factor of $\frac{3}{16}$, which would yield a violation (i.e. an objective function value of $(\text{CGLP})_3$) of $-\left(\frac{2}{9} * \frac{3}{16}\right) = -0.04167$.

Since there are no 0 coefficients in the source row, J has the unique valid partition (M_1, M_2), with

$$M_1 = \{9, 13, 14, 31\} \quad \text{and} \quad M_2 = \{4, 5, 6, 8, 30\},$$

and the corresponding basis of $(\text{CGLP})_3$ consists, besides u_0 and v_0, of the variables u_9, u_{13}, u_{14}, u_{31}, and v_4, v_5, v_6, v_8 and v_{30}. Thus the feasible solution to $(\text{CGLP})_3$ corresponding to the partition of J dictated by the coefficients of the x_3 row is *nondegenerate*, i.e. the associated basis is *unique*.

Computing the reduced costs r_{u_i}, r_{v_i} of the variables u_i, v_i of $(\text{CGLP})_3$ associated with the rows of the LP simplex tableau, we find that the variable u_{12} has a negative reduced cost of -0.25. Thus we perform a pivot in the corresponding row of LP,

$$s_{12} - 0s_4 - 0s_5 + 1s_6 + 1s_8 - 0s_9 - 1s_{13} - 1s_{14} + 1s_{30} - 2s_{31} = 0.$$

The variable s_{12} leaves the basis. To choose the entering variable, i.e. the pivot column, we compute the function $f_{12}(\gamma)$ whose value represents the amount by which $\bar{x}$ violates the intersection cut generated from the source row (the row of x_3) after the pivot, i.e. after adding to the source row γ_j times the row of s_{12}, where $\gamma_j = -\bar{a}_{3j}/\bar{a}_{12,j}$. Depending on the column where we choose to pivot, γ_j may be positive or negative, and in general the evaluation function f is calculated for both cases. In this instance, however, the ratio γ_j happens to be negative for all j with $\bar{a}_{12,j} \neq 0$. Figure 10.1 plots the function $f_{12}^{-}(\gamma_j)$, which has three breakpoints, namely at $\gamma_{31} = -\frac{1}{6}$, at $\gamma_6 = -\frac{1}{3}$, and at $\gamma_8 = \gamma_{13} = \gamma_{14} = \gamma_{30} = -\frac{2}{3}$, with

$$f_{12}(-\tfrac{1}{6}) = -0.05128$$

$$f_{12}(-\tfrac{1}{3}) = -0.05556$$

$$f_{12}(-\tfrac{2}{3}) = -0.05556$$

We choose s_{14} as the entering variable and perform the pivot; s_{12} leaves and s_{14} enters the basis. As a result, $-\frac{2}{3}$ times the s_{12} row is added to the x_3 row, which becomes

$$x_3 + \tfrac{1}{3}s_4 + \tfrac{1}{3}s_5 - \tfrac{1}{3}s_6 + 0s_8 - \tfrac{1}{3}s_9 - \tfrac{2}{3}s_{12} + 0s_{13} + 0s_{30} + 1s_{31} = \tfrac{2}{3}$$

As a result of the pivot, besides the coefficient of s_{14} (which entered the basis), those of several nonbasic variables (s_8, s_{13}, s_{30}) have become 0. If we were to generate the intersection cut from $0 \leq x_3 \leq 1$ applied to the new x_3 row, it would be

$$\tfrac{1}{9}s_4 + \tfrac{1}{9}s_5 + \tfrac{2}{9}xs_6 + 0x_8 + \tfrac{2}{9}s_9 + \tfrac{4}{9}s_{12} + 0s_{13} + 0s_{30} + \tfrac{1}{3}s_{31} \geq \tfrac{2}{9},$$

which cuts off $\bar{x}$ by a larger amount than the earlier cut.

Fig. 10.1 Plot of the evaluation function $f_{12}^{-}(\gamma)$

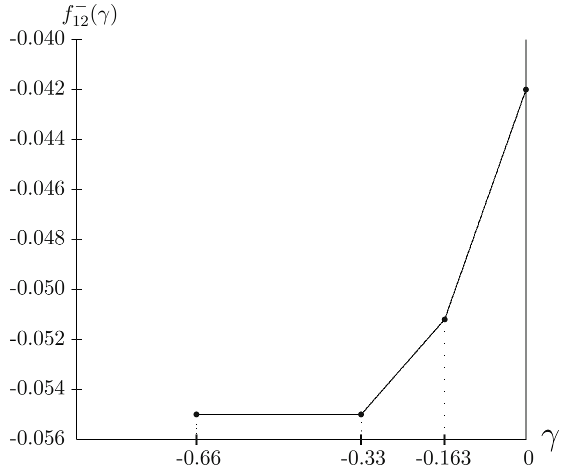


Table 10.1 $(\text{CGLP})_3$ pivots corresponding to the pivot in LP

Pivot #	Entering	Leaving
1	u_{12}	u_{31}
2	v_{31}	v_6
3	u_6	u_{14}

Turning now to what corresponds in $(\text{CGLP})_3$ to the new LP basis resulting from the pivot, we see that the new set $J' = (J \setminus \{14\}) \cup \{12\}$ can be partitioned into (M_1', M_2') in several ways because of the presence of several 0 coefficients. If we adopt the viewpoint that we leave (M_1', M_2') as close as possible to (M_1, M_2), i.e. move only those variables whose coefficients have changed sign, we obtain

$$M_1' = \{6, 9, 12, 13\} \quad \text{and} \quad M_2' = \{4, 5, 8, 30, 31\},$$

a partition that differs from (M_1, M_2) in several, not just one index. Thus, besides the variable u_{12} becoming basic in place of u_{14}, we also see that u_{31} and v_6 have left the basis, while u_6 and v_{31} have entered it. Thus the (feasible) basis of $(\text{CGLP})_3$ corresponding to the new (infeasible) basis differs from the earlier one in more than one component, i.e. is nonadjacent to the latter, and can be obtained from it only through a "block pivot," or sequence of pivots, that performs the above exchanges, as shown in Table 10.1.

All three pivots are nondegenerate, i.e. involve actual changes in the associated solutions to $(\text{CGLP})_3$. In addition, several degenerate pivots could be performed, corresponding to switching between components of the source row in LP (u_{13}, v_8, v_{30}).

Details of the computation, like the successive LP tableaus, can be found under [55].

Let x^* be the optimal solution to (LP).
Let $k \in \{1, \dots, p\}$with x_k^* fractional.
Let I and J be the index sets of basic and non-basic variables in an optimal basis of (LP).
Let $\overline{A}$ be the optimal tableau.
Let $num\_pivots := 0$.
while $num\_pivots < pivot\_limit$ **do**
  Compute the reduced costs r_{u_i}, r_{v_i} for each $i \notin J \cup \{k\}$
  if There exists i such that $r_{u_i} < 0 \vee r_{v_i} < 0$
   then
    Let $\hat{i} := \arg \min_{i \notin J \cup \{k\}} \min\{r_{u_i}, r_{v_i}\}$,
    Let $J' = \{j \in J : |\overline{a}_{\hat{i}j}| \geq \epsilon\}$ be the set of admissible pivots.
    Let $J^+ = J' \cap \{j \in J : \gamma_j = -\bar{a}_{kj}/\bar{a}_{\hat{i}j} > 0\}$.
    Let $\hat{j} := \arg\min\{\arg \min_{j \in J^+} f^+(\gamma_j), \arg \min_{j \in J' \setminus J^+} f^-(\gamma_j)\}$.
    Perform a pivot in (LP) by pivoting out $\hat{i}$ and pivoting in $\hat{j}$.
    Let $I := I \cup \{\hat{j}\} \setminus \{\hat{i}\}$.
    Let $\overline{A}$ be the updated tableau in the new basis.
    Let $num\_pivots += 1$.
   else /* *cut is optimal.* */
    Generate the MIG cut from row k of the current tableau.
    exit
  fi
od

Fig. 10.2 Lift-and-project procedure

In this example the pivot sequence in CGLP corresponding to a single pivot in LP was rather short. In larger problems the sequence may consist of hundreds and even thousands of pivots. For example, instance `air04` has 8904 variables (all 0-1) and 823 constraints. Solving the linear programming relaxation and choosing as source row the one corresponding to basic variable x_{895}, 20 pivots are performed in the LP tableau (in the subspace of fractional variables) to push the cut violation from -0.00009 to -0.00132. The first pivot in LP gives rise to 194 pivots in $(\text{CGLP})_{895}$. The 20 LP pivots correspond to a total of 1397 CGLP pivots.

Before discussing the computational experience of [18] with lift-and-project cuts generated from the LP simplex tableau, we briefly describe the Variants of the procedure that were implemented and tested. Variant 1 is the original Balas-Perregaard [33] procedure, as sketched in the pseudo-code of Fig. 10.2. Variants 2 and 3 are described in the next two sections.

10.2 Most Violated Cut Selection Rule

Here we present a variant of the lift-and-project procedure which uses a new rule for choosing the leaving and entering variables in the pivot sequence. The lift-and-project procedure in the (LP) tableau usually requires a remarkably small number of pivots to obtain the optimal L&P cut, nevertheless it may be computationally interesting to reduce this number further by studying alternate rules for this choice. The rule discussed here performs, at each iteration, the pivot to the adjacent basis in (LP) for which the objective of (CGLP)$_k$ is decreased by the largest amount or, equivalently, the one for which the intersection cut obtained from the source row k of the resulting (LP) tableau is the most violated by x^*.

Let us denote by $f_i^+(\gamma)$ (resp. $f_i^-(\gamma)$) the function $f^+(\gamma)$ (resp. $f^-(\gamma)$) defined for source row k and a row i of the tableau. Recall that these functions give the violation of the intersection cut derived from the row obtained by adding γ times row i to row k, depending on the sign of γ. Thus, the violation of the cut in the adjacent basis of (LP) where variable i leaves the basis and variable j enters the basis is given by $f_i^+(\gamma_j)$ if $\gamma_j = -\overline{a}_{kj}/\overline{a}_{ij} > 0$ and $f_i^-(\gamma_j)$ if $\gamma_j = -\overline{a}_{kj}/\overline{a}_{ij} < 0$, and the most violated intersection cut which can be derived from an adjacent basis has violation

$$\hat{\sigma} = \min_{i \in I \setminus \{k\}} \min\{\min_{j \in J^+} f_i^+(\gamma_j), \min_{j \in J^-} f_i^-(\gamma_j)\}$$

where I is the basic index set and J^+, J^- are the index sets for $\gamma_j > 0$ and $\gamma_j < 0$, respectively.

Here the variables $\hat{i}$ and $\hat{j}$ for which this minimum is attained are selected as the leaving and entering variables respectively. By computing the reduced costs r_{u_i} and r_{v_i}, we first identify all the candidate rows for an improving pivot. Then for each such row i we minimize the functions f_i^+ and f_i^-.

This clearly amounts to more computation at each iteration than the selection rule used in Variant 1, where only one minimization of the evaluation function is performed at each pivot. But on the other hand, the cut violation is increased at each iteration by an amount at least as large as under the selection rule of Variant 1, and therefore one may expect to obtain in less iterations a cut with a given violation. In particular, in the presence of zero elements in the source row, it presents the advantage that fewer degenerate pivots in (CGLP)$_k$ are performed.

10.3 Iterative Disjunctive Modularization

L&P cuts are obtained from disjunctions of the type

$$(u\tilde{A}x - u_0 x_k \geq u\tilde{b}) \vee (v\tilde{A}x + v_0 x_k \geq v\tilde{b} + v_0)$$

where solving the (CGLP)$_k$ optimizes the multipliers u, u_0, v and v_0. Once the optimal values for these multipliers are obtained, the cut can be further strengthened by using modular arithmetic on the coefficients of the integer-constrained components of x. As shown in Theorem 6.4 (of Chap. 6) and its proof, this latter operation can be interpreted as subtracting from x_k on each side of the disjunction a product of the form mx, where m is an integer vector with 0 components for the continuous variables, and then, after deriving the cut coefficients as functions of m, optimizing the components of m over all integer values. In other words, the strengthened deepest intersection cut is the result of a sequence of two optimization procedures, first in the multipliers u, v, u_0 and v_0, then in the components of m. But this raises the quest for a procedure that would simultaneously optimize both the continuous multipliers and the integer vector m. While this is an intricate task, equivalent to finding an optimal split cut, which has been treated elsewhere [38, 69], the iterative disjunctive modularization procedure described below is meant to approximate this goal.

Consider again the equation of the source row

$$x_k = \bar{a}_{k0} - \sum_{j \in J} \bar{a}_{kj} s_j \tag{10.2}$$

for an intersection cut or a MIG cut. By applying disjunctive modularization to this equation we mean deriving from it the modularized equation

$$y_k = \varphi_{k0} - \sum_{j \in J} \varphi_{kj} s_j \tag{10.3}$$

where y_k is a new, integer-constrained variable of unrestricted sign, $\varphi_{k0} = \bar{a}_{k0}$,

$$\varphi_{kj} := \begin{cases} \bar{a}_{kj} - \lfloor \bar{a}_{kj} \rfloor, & j \in J_1^+ := \{j \in J_1 : \bar{a}_{kj} - \lfloor \bar{a}_{kj} \rfloor \le \bar{a}_{k0}\} \\ \bar{a}_{kj} - \lceil \bar{a}_{kj} \rceil, & j \in J_1^- := \{j \in J_1 : \bar{a}_{kj} - \lfloor \bar{a}_{kj} \rfloor > \bar{a}_{k0}\} \\ \bar{a}_{kj} & j \in J_2 := J \setminus J_1 \end{cases}$$

and $J_1 := J \cap \{1, \ldots, p\}$.

Clearly, every set of s_j, $j \in J$, that satisfies (10.2) with x_k integer, also satisfies (10.3) with y_k integer; hence the equation (10.3) is valid. Also, it is easy to see that the intersection cut derived from (10.3) is the strengthened intersection cut, or MIG cut derived from (10.2). However, at this point we do not intend to generate a cut. Instead, we append (10.3) to the optimal (LP) tableau and declare it the source row in place of (10.2) for the entire pivoting sequence. Further, after each pivot in row $\hat{\imath}$ and column $\hat{\jmath}$ the transformed row of y_k, say

$$y_k = \varphi'_{k0} - \sum_{j \in J'} \varphi'_{kj} s_j$$

where $J' := (J \setminus \{\hat{\jmath}\}) \cup \{\hat{\imath}\}$, is treated again with disjunctive modularization. Namely, this time the row of y_k is replaced with

$$y_k = \bar{\varphi}_{k0} - \sum_{j \in J'} \overline{\varphi}_{kj} s_j$$

where $\overline{\varphi}_{k0} = \varphi'_{k0}$, and

$$\overline{\varphi}_{kj} := \begin{cases} \varphi'_{kj} - \lfloor \varphi'_{kj} \rfloor, & j \in (J'_1)^+ \\ \varphi'_{kj} - \lceil \varphi'_{kj} \rceil, & j \in (J'_1)^- \\ \varphi'_{kj} & j \in J'_2 \end{cases} \tag{10.4}$$

with $(J'_1)^+$, $(J'_1)^-$ and J'_2 defined analogously to J_1^+, J_1^- and J_2.

The expressions used for calculating the reduced costs r_{u_i}, r_{v_i} and the evaluation functions $f^+(\gamma)$, $f^-(\gamma)$ used for selecting the pivot element at each iteration remain valid, except for the fact that the entries $\bar{a}_{kj}$ of the current row (10.2) of x_k are replaced (since this is no longer the source row) with the entries $\overline{\varphi}_{kj}$ of the current row of y_k.

It is clear that the modularized source row, if used for cut generation, would yield a cut that dominates the one from the unmodularized source row. It can also be shown that every iteration of the cut generating algorithm that uses disjunctive modularization improves the cut obtainable from the source row.

10.4 Computational Results

The algorithm for generating L&P cuts from the (LP) tableau was implemented, in all three of its Variants discussed above, as a cut generator called `CglLandP` [53] in the COIN-OR framework. This generator is open-source and is available since September 2006 as part of the Cut Generation Library [55]. All the computations have been carried out on a computer equipped with a 2.93 GHz Intel Xeon CPU. The version of the COIN-OR code used is the release 2.2.0 of Cbc with an enhanced version of the `CglLandP` cut generator.

Table 10.2 presents a comparison of methods for generating 10 rounds of cuts at the root node, where a round means a cut for each of the 50 most fractional integer-constrained variables. For each lift-and-project cut generated a maximum of 10 nondegenerate pivots is performed in the LP tableau. In this experiment, the problems are preprocessed with COIN `CglPreproces` procedure and then 10 rounds of cuts are generated. The test set consists of all 65 problems from the MIPLIB.3 library. The four methods compared are mixed integer Gomory (MIG) cuts, and the three variants of lift-and-project (L&P) cuts presented in this paper: Variant 1 (the algorithm of [33]), Variant 2 (the algorithm using the most violated cut selection rule) and Variant 3 (the algorithm using iterative disjunctive modularization). For each of the methods, the running time and the percentage of the initial integrality gap closed are reported.

As can be seen from the table, generating lift-and-project cuts with the three different variants proposed here is more expensive than generating MIG cuts. For the given test set, it took on the average 0.74 s per instance to perform 10 rounds

Table 10.2 Comparing 10 rounds of different cuts at the root node

			Lift-and-project cuts					
	MIG cuts		Variant 1		Variant 2		Variant 3	
Name	CPU(s)	%gap	CPU(s)	%gap	CPU(s)	%gap	CPU(s)	%gap
10teams	0.18	100.00	2.39	100.00	5.60	100.00	2.58	100.00
air03	0.04	100.00	0.58	100.00	0.52	100.00	0.59	100.00
air04	2.55	13.01	19.62	19.08	54.70	20.81	21.70	15.00
air05	1.55	8.34	16.72	18.13	42.93	21.31	13.67	14.46
arki001	0.13	46.17	5.04	49.47	2.35	47.82	4.86	47.00
bell3a	0.00	70.73	0.02	70.74	0.01	70.74	0.02	70.74
bell5	0.00	90.10	0.08	92.20	0.04	92.43	0.09	91.45
blend2	0.01	27.13	0.15	28.83	0.24	33.45	0.15	27.91
cap6000	0.15	60.70	0.56	54.47	0.90	54.47	0.87	54.47
dano3mip	11.01	0.06	86.56	0.10	301.55	0.11	133.77	0.19
danoint	0.06	1.74	2.29	1.41	5.54	1.85	2.35	1.44
dcmulti	0.03	74.33	1.46	89.71	2.13	87.61	1.39	84.40
dsbmip	0.07	no_gap	5.40	no_gap	3.22	no_gap	4.94	no_gap
egout	0.00	99.45	0.01	100.00	0.01	100.00	0.01	100.00
enigma	0.00	no_gap	0.05	no_gap	0.03	no_gap	0.04	no_gap
fast0507	23.97	3.04	237.68	3.91	761.70	7.14	230.59	3.58
fiber	0.04	89.22	1.23	94.72	0.65	93.78	0.85	92.83
fixnet6	0.04	85.85	0.74	87.73	1.06	88.51	0.75	87.73
flugpl	0.00	14.23	0.01	17.99	0.01	17.50	0.01	17.38
gen	0.01	87.43	0.20	96.72	0.15	82.51	0.21	97.27
gesa2	0.03	88.08	5.15	81.92	0.66	79.68	4.82	94.15
gesa2_o	0.04	83.27	3.53	76.48	1.19	94.44	4.29	74.17
gesa3	0.04	59.63	5.29	88.54	2.00	88.79	6.43	85.11
gesa3_o	0.03	62.94	3.75	90.06	1.44	87.84	3.71	84.41
gt2	0.00	100.00	0.04	95.56	0.04	95.58	0.05	96.19
harp2	0.09	38.26	1.58	55.90	1.50	54.40	1.57	58.59
khb05250	0.03	96.11	0.38	98.88	0.22	98.95	0.38	98.88
l152lav	0.18	26.46	2.85	44.36	7.55	45.51	2.83	47.91
lseu	0.00	51.86	0.05	75.73	0.05	84.51	0.08	73.53
markshare1	0.00	0.00	0.02	0.00	0.02	0.00	0.02	0.00
markshare2	0.00	0.00	0.02	0.00	0.03	0.00	0.03	0.00
mas74	0.01	8.30	0.17	9.03	0.22	8.67	0.16	8.60
mas76	0.01	7.28	0.13	8.87	0.19	8.52	0.11	7.78
misc03	0.01	18.97	0.34	27.21	0.81	29.74	0.22	20.19
misc06	0.02	62.00	0.37	100.00	0.16	97.83	0.29	100.00
misc07	0.02	0.79	0.57	3.84	0.55	4.85	0.44	2.77
mitre	0.03	100.00	1.71	100.00	0.48	100.00	2.90	100.00
mkc	0.15	36.44	7.22	45.66	2.20	38.20	6.93	30.41
mod008	0.01	32.22	0.04	29.20	0.06	53.71	0.05	41.97

(continued)

Table 10.2 (continued)

			Lift-and-project cuts					
	MIG cuts		Variant 1		Variant 2		Variant 3	
Name	CPU(s)	%gap	CPU(s)	%gap	CPU(s)	%gap	CPU(s)	%gap
mod010	0.01	100.00	2.31	95.79	0.41	100.00	0.40	100.00
mod011	0.34	41.99	23.44	18.75	12.65	35.59	29.00	18.21
modglob	0.01	45.50	1.16	66.24	0.80	62.52	1.16	66.24
noswot	0.00	no_gap	0.12	no_gap	0.05	no_gap	0.23	no_gap
nw04	6.17	99.20	84.46	80.70	93.41	100.00	93.57	96.57
p0033	0.00	100.00	0.00	100.00	0.00	100.00	0.00	100.00
p0201	0.02	63.66	0.68	79.93	0.83	89.56	0.68	79.72
p0282	0.01	21.83	0.18	88.83	0.24	87.86	0.16	93.73
p0548	0.02	92.96	0.46	99.83	0.26	95.72	0.49	99.48
p2756	0.05	97.42	1.14	97.87	0.43	97.11	1.11	97.03
pk1	0.00	0.00	0.11	0.00	0.16	0.00	0.11	0.00
pp08a	0.03	82.61	0.86	92.89	0.43	89.82	0.80	94.87
pp08aCUTS	0.04	59.97	0.97	83.62	0.46	76.44	0.99	86.47
qiu	0.14	10.81	6.55	24.26	14.77	30.83	6.41	24.26
qnet1	0.14	34.56	2.73	40.28	4.97	44.28	3.84	41.59
qnet1_	0.12	55.77	1.99	69.49	3.20	68.08	3.50	71.01
rentacar	0.09	18.94	2.06	36.02	2.12	38.19	2.06	36.02
rgn	0.01	31.92	0.08	62.69	0.07	64.72	0.08	57.49
rout	0.06	13.14	1.72	32.58	3.94	41.54	1.72	34.40
set1ch	0.04	71.37	2.14	71.38	0.60	73.62	2.07	69.54
seymour	0.19	17.62	57.53	22.18	15.16	21.71	61.19	21.71
stein27	0.01	0.00	0.13	0.00	0.18	0.00	0.13	0.00
stein45	0.01	0.00	0.53	0.00	0.80	0.00	0.52	0.00
swath	0.46	26.78	16.70	28.45	4.80	29.08	12.77	28.46
vpm1	0.01	61.66	0.11	73.67	0.06	85.45	0.06	89.09
vpm2	0.01	51.52	0.41	69.87	0.34	63.45	0.39	66.82
Average	0.75	49.09	9.578	56.32	20.98	57.85	10.43	56.5
Geo. Mean	–	34.08	–	42.59	–	45.32	–	42.27

of MIG cuts, while it took between 10 and 23 s per instance for the three variants of lift-and-project cuts. This should be seen in the light of Perregaard's observation (see [107]) that lift-and-project cut generation takes on average less than 5% of the total time needed to solve a mixed integer program. The experience of the Balas-Bonami project [18] in this respect is that on the set of problems solved, the average and the geometric mean of the time Variant 1 spent on cut generation was 17% and 7%, respectively, of the total time needed to solve the problem; but if we restrict ourselves to the harder instances that took at least 30 s to solve, these numbers shrink to 9% and 2% respectively. It should be noted that most of the computing time spent in the lift-and-project procedure is spent calling the routines of the underlying LP solver to compute the rows of the tableau, multiply the basis inverse by a vector and pivot. Overall 61% of the CPU time is spent in these procedures. For the 8 problems for which the cut generation took more than 10 s, 74% of the computing time was spent on these procedures. In the current implementation, these operations are performed by calling the procedures of COIN-OR LP solver (CLP) externally through the Open Solver Interface. A tighter integration with the LP solver should reduce the overall time of the procedure.

The extra computational cost of generating cuts made it possible to close a significantly larger fraction of the integrality gap, namely 56% versus 49% for the MIG cuts, if we look at the average numbers, or between 42 and 44% for the L&P cuts, versus 34% for the MIG cuts, if we look at the geometric means. To put it differently, the geometric mean of the gap closed by the 3 variants of the L&P cuts is roughly between 1.2 and 1.3 times larger than that of the gap closed by MIG cuts.

To make the comparisons easier to interpret, the instances were grouped according to their degree of difficulty, measured by the percentage of the gap closed by 10 rounds of MIG cuts at the root node:

– Group 1 (20 instances): gap closed $> 75\%$
– Group 2 (12 instances): $50\% \leq$ gap closed $\leq 75\%$
– Group 3 (33 instances): gap closed $< 50\%$.

The results are shown in Table 10.3.

As can be seen from the table, the improvement of lift-and-project cuts over MIG cuts is larger for the harder instances where MIG cuts close less gap. For instances of Group 3, the various lift-and-project procedures close about 50% more gap than the MIG cuts, for instances of Group 2 lift-and-project cuts close about 30% more gap, while for instances in Group 1 all methods are roughly equivalent.

To more thoroughly assess the effectiveness of lift-and-project cuts, it is of course necessary to solve the instances to completion by running a branch-and-cut code and using these cuts to strengthen the LP relaxation. Table 10.4 describes the results of generating 10 rounds of cuts at the root node with 50 cuts per round, and then solving the problem by branch-and-bound without further cut generation. Again, the four cut generation methods tested are MIG cuts and the three variants of L&P cuts. For all three variants, the limit on the number of pivots for generating a cut is set to 10. These runs were performed by using `CbC 2.2` (COIN-OR Branch and Cut) with some specific settings: a 42,000 s time limit for solving each problem was imposed; all the default cut generation procedures of `Cbc` were deactivated; the variable selection strategy used was strong branching with the default parameters of `Cbc` (i.e., performing strong branching on the five most fractional variables); the

Table 10.3 Average gap closed by 10 rounds of cuts at the root node

Method	MIG	Variant 1	Variant 2	Variant 3
Group 1	91.44	93.74	94.53	95
Group 2	60.98	78.98	80.03	78.69
Group 3	20.98	30.13	32.69	29.51

Table 10.4 Comparing complete resolutions with cut-and-branch with 10 rounds of cuts

	MIG cuts		Lift-and-project cuts					
			Variant 1		Variant 2		Variant 3	
	Time (s)	# Nodes	Time (s)	# Nodes	Time (s)	# Nodes	Time (s)	# Nodes
Group A								
air03	0.66	1	1.06	1	1.09	1	1.02	1
dcmulti	1.98	79	5.22	57	8.11	109	4.05	61
dsbmip	2.06	44	6.60	43	7.15	53	6.28	42
egout	0.03	11	0.02	3	0.02	1	0.02	3
enigma	0.99	819	3.70	9180	2.19	3246	0.89	962
fiber	7.81	565	6.73	261	9.33	381	5.49	253
fixnet6	4.39	187	8.48	119	8.52	159	7.90	139
flugpl	0.15	643	0.17	737	0.15	689	0.17	737
gen	0.33	25	0.67	27	0.27	15	0.80	29
gesa3	3.94	199	10.22	87	8.92	115	15.79	113
gesa3_o	4.36	237	9.45	75	9.02	133	8.86	141
gt2	0.08	66	0.24	190	0.06	17	0.13	47
khb05250	0.69	35	0.68	13	0.79	19	0.73	17
l152lav	4.70	63	11.51	109	8.55	51	9.69	123
misc03	1.09	181	1.85	75	1.85	79	2.12	135
misc06	0.57	22	0.40	5	0.41	11	0.53	7
mitre	0.32	2	1.71	1	0.59	1	2.45	1
mod008	0.42	417	0.95	919	0.69	585	0.28	131
mod010	0.37	1	4.17	9	3.98	1	3.08	1
nw04	13.01	1	56.48	1	54.00	1	65.13	1
p0033	0.01	1	0.02	1	0.01	1	0.01	1
p0201	1.49	120	2.67	47	2.25	35	2.41	75
p0282	0.44	335	0.63	323	0.60	365	0.62	337
p0548	1.04	187	0.82	74	13.01	6476	0.98	90
p2756	4.35	321	5.00	230	5.65	395	7.95	467
qnet1	11.29	63	14.11	55	15.92	59	13.14	69
qnet1_o	11.34	165	12.11	73	1246	67	12.53	65
rentacar	2.64	13	5.66	17	5.04	19	5.50	17
rgn	0.62	723	1.30	1121	0.69	703	1.14	1,103
vpm1	2.26	2970	0.27	68	0.11	14	0.09	4
Group B								
10teams	36.79	304	57.38	405	113.16	911	105.69	704
air04	131.25	223	166.62	243	193.91	265	155.76	189
air05	139.46	471	152.70	325	178.04	433	147.95	351
bell3a	8.70	44,851	8.41	33,761	9.31	34,153	9.21	33,747
bell5	59.35	264,611	3.03	13,315	45.61	216,747	16.18	83,197
blend2	4.49	1721	9.77	2398	6.29	2102	5.82	1351
cap6000	28.57	10,339	27.95	10,625	29.52	10,307	31.71	10,215
gesa2	30.49	12,333	15.88	3335	21.78	6213	24.34	5325
gesa2_o	39.62	9195	31.58	8105	29.72	3247	37.77	6405

(continued)

Table 10.4 (continued)

	MIG cuts		Lift-and-project cuts					
			Variant 1		Variant 2		Variant 3	
	Time (s)	# Nodes	Time (s)	# Nodes	Time (s)	# Nodes	Time (s)	# Nodes
harp2	14,674.71	3.92463e+06	9694.80	2.6572e+06	22,686.87	4.68211e+06	2996.50	1.26206e+06
lseu	4.10	22,641	1.29	3093	0.78	1100	0.83	1087
mas76	650.84	674,109	223.86	820,231	173.68	750,947	315.52	681,103
misc07	74.27	10,851	105.84	21,713	95.47	18,583	82.94	16,639
mod011	476.88	3717	418.78	3127	413.58	3445	394.47	3139
modglob	124.63	155,829	10.60	5059	196.32	256,581	10.52	5059
nw04	13.01	1	56.48	1	54.00	1	65.13	1
pk1	77.70	261,919	88.73	296,929	90.87	302,399	119.94	406,847
pp08a	229.49	149,457	5.89	1025	22.67	12,079	5.87	1025
pp08aCUTS	109.62	68,023	11.32	2991	31.30	10,679	11.15	3079
qiu	397.35	12,673	598.43	10,049	638.63	13,849	596.93	10,049
rout	1089.71	333,114	1037.04	214,307	2101.87	345,029	993.90	147,329
stein27	0.97	3679	1.08	3767	1.04	3529	1.04	3617
stein45	28.85	58,021	27.81	44,163	24.12	50,997	26.64	56,059
vpm2	34.56	47,163	30.57	36,298	13.00	15,798	14.68	17,850

node selection strategy was best bound. Table 10.4 shows, for each instance and each method, the computing time and the number of search tree nodes needed to solve the problem.

Among the 65 instances of MIPLIB.3, 12 were not solved in the 11.7 h time limit with any of the methods tested (namely `arki01`, `dano3mip`, `danoint`, `fast0507`, `mas74`, `markshare1`, `markshare2`, `mkc`, `rout`, `set1ch`, `seymour` and `swath`.) Statistics for these 12 problems are not included. The remaining 53 instances are divided into two groups:

Group A: instances solved with MIG cuts or Variant 1 of the L&P procedure in less than 30 s and 1000 nodes.

Group B: instances whose solution required more than 30 s or 1000 nodes with both MIG cuts and Variant 1 of L&P.

Table 10.5 gives the averages and geometric means for the two groups of instances of Table 10.4. As it shows, the branch-and-bound trees generated by each of Variants 1, 2 and 3 are considerably smaller than the one obtained with the MIG cuts.

However, Table 10.5 also shows that relative performance of the L&P cuts versus the MIG cuts tends to improve as we go from the easy instances to the harder ones. This is in accordance with the well known fact that cutting planes are of little or no use in solving easy problems, but they gain increasing significance as the problems get harder. On the hard instances of group B of Tables 10.4 and 10.5, the geometric mean of the number of search tree nodes generated by the three Variants of L&P cuts is 44.5, 70.7 and 42.7%, respectively, of the corresponding number for the MIG cuts. In terms of computing times, the geometric mean of the CPU time needed for the

Table 10.5 Complete resolutions compared between easy and hard instances

	MIG cuts		Lift-and-project cuts					
			Variant 1		Variant 2		Variant 3	
	Time (s)	# Nodes	Time (s)	# Nodes	Time (s)	# Nodes	Time (s)	# Nodes
Group A								
Average	2.781	283.2	5.763	464	6.048	460	5.993	172.4
Geo. mean	1.0418	62.752	1.7955	48.177	1.6047	44.875	1.5569	38.564
Group B								
Average	802.3	2.639e+05	553.5	1.823e+05	1179	2.931e+05	265.5	1.198e+05
Geo. mean	73.188	23,449	42.081	10,444	61.397	16,590	42.694	10,021
All instances								
Average	349.7	1.147e+05	243.4	7.937e+04	515.1	1.275e+05	118.6	5.211e+04
Geo. mean	6.5942	820.34	7.0579	497.26	7.8028	583.92	6.5516	430.6

three Variants of L&P cuts is 57.4, 83.8 and 58.3%, respectively, of the CPU time for MIG cuts.

10.5 Testing Alternative Normalizations

An additional feature of the project [18] was to test alternative normalizations. While the standard normalization

$$ue + u_0 + ve + v_0 = 1, \tag{10.5}$$

where $e = (1, \ldots, 1)$, is known to be best performing on average among those tested, some of the untested alternatives were implemented and tested. In particular, a generalized version of (10.5),

$$\sum_{i=1}^{m+p+1} (u_i + v_i)\lambda_i + u_0 + v_0 = \lambda_0, \tag{10.6}$$

where $\lambda_0 > 0$ and $\lambda_i \geq 0$, $i = 1, \ldots, n$, was considered, and the corresponding modified expressions for the reduced costs r_{u_i}, r_{v_i} and the evaluation functions $f^+(\gamma)$, $f^-(\gamma)$ were obtained (see Theorems 3 and 4 of [18]). Furthermore, the following three versions of (10.6) were computationally tested, on the set of instances shown in Table 10.2:

(a) The unweighted version, with $\lambda_i = 1, i = 1, \ldots, m + p + n$
(b) The weighted version, with $\lambda_i = ||A^i||_1$, where A^i is the i-th row of A
(c) The Euclidean normalization, with $\lambda_i = ||A^i||_2$, proposed by Fischetti et al. [76].

Table 10.6 Gaps closed with the three versions of normalization (averages over all instances)

	Variant 1	Variant 2	Varient 3
	% Gap	% Gap	% Gap
Version 1 (unweighted)	56.32	57.85	56.5
Version 2 (weighted)	56.19	57.5	56.4
Version 3 (Euclidean)	54.79	57.49	56.29

In all cases, λ_0 was set to $n + 1$.

The experiments showed that

(a) The size of λ_0 did not meaningfully affect the outcomes
(b) None of the three versions performed uniformly better than the others. In particular, Table 10.6 shows the geometric means of gap closed by each of the three versions in the case of each of the three variants.

10.6 The Interplay of Normalization and the Objective Function

As already mentioned, the normalization

$$u_0 + v_0 = 1 \tag{10.7}$$

yields the standard intersection cuts from $\{x : 0 \leq x_k \leq 1\}$, also known as the simple disjunctive cut. Other normalization constraints yield stronger cuts, and a considerable amount of experimentation has shown the equation

$$ue + ve + u_0 + v_0 = 1 \tag{10.8}$$

to be overall the best performing normalization constraint. The reasons for this good performance of (10.8), as well as its limitations, were recently analyzed by Fischetti et al. [76] and here we summarize their conclusions. Their starting point is an experiment that compares the results of applying 10 rounds of cuts under the normalizations (10.7) and (10.8) (called trivial and standard, respectively, in [76]), to a battery of 15 well known MIP instances. It should be mentioned that instead of the split disjunction on x_k, Fischetti et al. [76] considers the more general spilt disjunction $\{x : \pi x \leq \pi_0 \vee \pi x \geq \pi_0 + 1\}$. The outcome of the experiment is as follows (here $S(u, v)$ is the support of the vector $u + v$, i.e. $S(u, v) = \{i : u_i + v_i > 0\}$):

Average	Number of cuts	Gap closed (%)	$\lvert S(u, v)\rvert$
Normalization (10.7)	236.33	37.58	35.59
Normalization (10.8)	253.67	56.87	17.52

The fact that the standard normalization yields stronger cuts, i.e. closes a larger gap, than the trivial one, comes of course as no surprise, and was to be expected. What does, however, come as a surprise, is that the cuts of the standard normalization (10.8) have a much sparser dual support (i.e. are obtained by combining much fewer constraints, actually about half as many) than those under the trivial normalization (10.7). Since the original constraint set is in general sparser than the subsequent cuts that result from iterations that combine multiple rows, the fact that the cuts obtained with normalization (10.8) are the combination of half as many rows as the cuts obtained with normalization (10.7) suggests that they must be considerably sparser. This is confirmed in [76] for specific instances where the difference is dramatic, but unfortunately no global numbers are given on this aspect. However, a very insightful explanation of the phenomenon is provided, which goes as follows.

The fact that the sum of multipliers (the variables u_i, v_i are multipliers with which the rows of A are combined into a cut) is restricted to 1 at every iteration of the successive cut generating procedure, means that the individual weight with which a given constraint enters the combination for a cut at the k-th iteration becomes smaller and smaller with the increase of k. This apparently favors the inclusion of sparser inequalities over denser ones, which slows the increase in density otherwise common to recursive cut generation. To put it differently, the normalization which restricts the sum of multipliers u_i, v_i to a constant tends to implicitly penalize the rank of the cuts to be generated, because higher rank cuts will be more "expensive" in terms of their multiplier sum. Also, since lower rank cuts are preferred (to be taken into the combination for new cuts) and since the rank 0 cuts (the original constraints) tend to be sparser than the higher rank cuts, the cuts generated with normalization (10.8) tend to remain relatively sparse.

While this explains at least partially the success of the standard normalization, at the same time it throws some light on some of its weaknesses. The same feature (the sum of multipliers being fixed to a constant) that has the advantages discussed above, also has the drawback that as a result, the relative size of the multipliers u_i, v_i depends on the scaling of the i-th constraint. For it is easy to see that multiplying constraint i of $Ax \geq b$ by $k > 1$, implies that the corresponding multipliers u_i and v_i will be divided by k, which in turn is equivalent to reducing the coefficients of u_i and v_i in the normalization constraint from 1 to $1/k$. Thus the scaled i-th constraint becomes "cheaper" if one interprets the righthand side of (10.8) as a resource. In [76], a 2-dimensional example is given in which multiplying one of the constraints by k, which of course does not affect the LP optimum, produces a CGLP whose optimal solution—and so the resulting cut—heavily depends on k. There are two optimal solutions depending on whether $k \leq 8$ or $k \geq 8$, but the basic solution that yields the deepest cut , and the only facet defining one, is never optimal under the usual objective function, $\min \alpha\bar{x} - \beta$.

Another inconvenient aspect of the normalization (10.8) is that although the hyperplane that it defines intersects the projection cone W in a single hyperplane, in the (α, β)-subspace this may correspond to a truncation of the cone W_0 with multiple hyperplanes, and thus an extreme point of the resulting polyhedron may not correspond to an extreme ray of W_0. To avoid this difficulty, another normalization was proposed in [32], whose generic form is

$$\alpha y = 1 \tag{10.9}$$

Let $(\mathrm{CGLP})^y$ denote the cut generating linear program with this normalization. Let P_D denote the convex hull of the disjunctive set for which we are generating cutting planes, and let W and W_0 be the projection cones in the full space of $(\mathrm{CGLP})^y$ and in the (α, β) subspace, respectively. Recall that W_0 is the reverse polar of P_D. The advantage of the normalization (10.9) is that it intersects W_0 with a single hyperplane in the (α, β)-space and thus has the effect that every extreme point of the resulting polyhedron corresponds to an extreme ray of W_0. This does not imply that every extreme point of the higher-dimensional $(\mathrm{CGLP})^y$ corresponds to an extreme ray of W_0; but it does imply that if the objective $\min(\alpha\bar{x} - \beta)$ is bounded then there exists an optimal extreme point of $(\mathrm{CGLP})^y$ that corresponds to an extreme ray of W_0.

Theorem 10.2 ([32]) *Let $(CGLP)^y$ be feasible. Then it has a finite minimum if and only if $\bar{x} + y\lambda \in P_D$ for some $\lambda \in \mathbb{R}$.*

Proof If $(\mathrm{CGLP})^y$ is unbounded in the direction of minimization, then there exists $(\tilde{\alpha}, \tilde{\beta}) \in P_Q^*$ with $\bar{x}^T\tilde{\alpha} < \tilde{\beta}$ and $y^T\tilde{\alpha} = 0$. But then $(\bar{x} + y\lambda)^T\tilde{\alpha} < \tilde{\beta}$ for all $\lambda \in \mathbb{R}$, hence $\bar{x} + y\lambda \notin P_D$.

Conversely, if $\bar{x} + y\lambda \notin P_D$ for all $\lambda \in \mathbb{R}$, there exists $(\hat{\alpha}, \hat{\beta}) \in \mathbb{R}^{n+1}$ such that $\hat{\alpha}x \geq \hat{\beta}$ for all $x \in P_D$ and $\hat{\alpha}(\bar{x} + y\lambda) < \hat{\beta}$ for all $\lambda \in \mathbb{R}$, i.e. $\hat{\alpha}y = 0$ and $\hat{\alpha}\bar{x} < \hat{\beta}$. But then $(\hat{\alpha}, \hat{\beta})$ is a direction of unboundedness for $(\mathrm{CGLP})^y$. □

Theorem 10.3 ([32]) *If $(CGLP)^y$ has an optimal solution $(\tilde{\alpha}, \tilde{\beta})$, then*

$$\bar{x}^T\tilde{\alpha} - \tilde{\beta} = \lambda^* := \min\{\lambda : \bar{x} + y\lambda \in P_D\}$$

and

$$(\bar{x} + y\lambda^*)^T\tilde{\alpha} = \tilde{\beta}.$$

Proof Let $(\tilde{\alpha}, \tilde{\beta})$ be an optimal solution to $(\mathrm{CGLP})^y$, and define $\lambda^* := \min\{\lambda : \bar{x} + y\lambda \in P_D\}$. Since $\tilde{\alpha}y \neq 0$, there exists $\lambda^0 \in \mathbb{R}$ such that $(\bar{x} + y\lambda^0)^T\tilde{\alpha} = \tilde{\beta}$. Further, $\bar{x}^t\tilde{\alpha} - \tilde{\beta} = y^T\tilde{\alpha}\lambda^0$. We claim that $\lambda^0 = \lambda^*$. For suppose $\lambda^0 > \lambda^*$. Then

$$\begin{aligned}(\bar{x}^T + y\lambda^*)^T\tilde{\alpha} - \tilde{\beta} &= \bar{x}^T\tilde{\alpha} - \tilde{\beta} + y^T\tilde{\alpha}\lambda^* \\ &= -\lambda^0 + \lambda^* \qquad (\text{since } y^T\tilde{\alpha} = 1) \\ &< 0,\end{aligned}$$

i.e. the point $\bar{x} + y\lambda^*$ violates the inequality $\tilde{\alpha}x \geq \tilde{\beta}$, contradicting $\bar{x} + y\lambda^* \in P_D$.

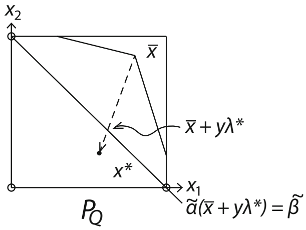


Fig. 10.3 $y = x^* - \bar{x}$

Now suppose $\lambda^0 < \lambda^*$. Then there exists a hyperplane $\bar{\alpha}x = \bar{\beta}$ such that $\bar{\alpha}x \geq \bar{\beta}$ for all $x \in P_Q$, $\bar{\alpha}(\bar{x} + y\lambda^*) = \bar{\beta}$, and $\bar{\alpha}(\bar{x} + y\lambda^0) < \bar{\beta}$. Further, since $\bar{\alpha}y\lambda^0 < \bar{\alpha}y\lambda^*$ and $\lambda^0 < \lambda^*$, it follows that $\bar{\alpha}y > 0$ and so w.l.o.g. we may assume that $(\bar{\alpha}, \bar{\beta})$ is scaled so as to make $\bar{\alpha}y = 1$. But then

$$\bar{\alpha}\bar{x} - \bar{\beta} < \bar{\alpha}y\lambda^0 = \tilde{\alpha}\bar{x} - \tilde{\beta},$$

contradicting the optimality of $(\tilde{\alpha}, \tilde{\beta})$ for $(\text{CGLP})^y$. □

Corollary 10.4 *[32] Let $y := \bar{x}$ for some $x^* \in P_Q$. Then $(\text{CGLP})^y$ has an optimal solution $(\tilde{\alpha}, \tilde{\beta})$ such that (i) $\tilde{\alpha}\bar{x} < \tilde{\beta}$; and (ii) $\tilde{\alpha}x = \tilde{\beta}$ is the supporting hyperplane of P_Q that intersects the line segment $(\bar{x}, x^*]$ at the point closest to x^*.*

Theorem 10.3 and Corollary 10.4 are illustrated in Fig. 10.3.

10.7 Bonami's Membership LP

A variant of the cut generating linear program due to Pierre Bonami [47] that has some computational advantages is as follows. Consider a mixed integer linear program of the form

$$\max\{cx : Ax = b, x \in \mathbb{Z}_+^p \times \mathbb{R}_+^{n-p}\}, \tag{MILP}$$

whose linear relaxation P has optimal solution $\hat{x}$. Here A is $m \times n$.

Lift-and-project cuts for (MILP) from a disjunction of the form

$$\begin{pmatrix} Ax = b \\ x \geq 0 \\ -x_k \geq -\lfloor \hat{x}_k \rfloor \end{pmatrix} \vee \begin{pmatrix} Ax = b \\ x \geq 0 \\ x_k \geq \lceil \hat{x}_k \rceil \end{pmatrix} \tag{10.10}$$

for some $k \in \{1, \dots, p\}$ such that $\lfloor \hat{x}_k \rfloor < \hat{x}_k < \lceil \hat{x}_k \rceil$ can be generated by solving the cut generating linear program

$$
\begin{array}{llr}
\min\ \alpha\bar{x} - \beta & & \\
\alpha \quad - A^T u - s + e_k u_0 & = 0 & \\
\alpha \quad\quad - A^T v - t - e_k v_0 & = 0 & \\
\quad - \beta + b^T u - \lfloor \hat{x}_k \rfloor u_0 & = 0 & \quad (\mathrm{CGLP})_k \\
\quad - \beta \quad + b^T v + \lceil \bar{x}_k \rceil v_0 & = 0 & \\
\alpha \in \mathbb{R}^n, \beta \in \mathbb{R}; u, v \in \mathbb{R}^m; s, t \in \mathbb{R}^n_+; u_0, v_0 \in \mathbb{R}_+ & &
\end{array}
$$

subject to some normalization constraint. It is known that if the normalization constraint used is

$$ue + ve + se + te + u_0 + v_0 = 1, \tag{10.11}$$

then the cut $\bar{\alpha}x \geq \bar{\beta}$ defined by an optimal solution to $(\mathrm{CGLP})_k$ is maximally violated by $\hat{x}$, i.e. the strongest possible in a well defined sense. It is also known that if instead of (10.11), the normalization constraint

$$u_0 + v_0 = 1 \tag{10.12}$$

is used, the cut defined by an optimal solution to $(\mathrm{CGLP})_k$ is the simple disjunctive cut from

$$
\begin{gathered}
x_k = \hat{x}_k - \textstyle\sum_{j \in J} \bar{a}_{kj} x_j, \\
x_k \leq \lfloor \hat{x}_k \rfloor \vee x_k \geq \lceil \bar{x}_k \rceil
\end{gathered} \tag{10.13}
$$

where the equation is the expression for x_k in the optimal simplex tableau. Let's denote by $(\mathrm{CGLP})_k^{(10.11)}$ and $(\mathrm{CGLP})_k^{(10.12)}$ the $(\mathrm{CGLP})_k$ with normalization (10.11) and (10.12), respectively.

In [47], Pierre Bonami derived a cut generating linear program in $\mathbb{R}^n$, equivalent to $(\mathrm{CGLP})_k^{(10.12)}$ in the sense of yielding the same cut as the latter. In particular, he has shown that $(\mathrm{CGLP})_k^{(10.12)}$ is equivalent to the linear program

$$
\begin{array}{ll}
\min\ \hat{f}_k b^T \lambda + \hat{x}^T s & (-\hat{f}_k \lceil \hat{x}_k \rceil) \\
\quad A^T \lambda + \quad s - t = e_k & \\
\quad \lambda \in \mathbb{R}_m; s, t \geq \mathbb{R}^n_+ &
\end{array} \tag{10.14}
$$

where $\hat{f}_k = \hat{x}_k - \lfloor \hat{x}_k \rfloor$, and whose dual

$$
\begin{array}{llr}
\max & y_k \quad - \hat{f}_k \lceil \hat{x}_k \rceil & \\
& Ay = b\hat{f}_k & \quad (\mathrm{MLP})_k \\
& 0 \leq y_k \leq \hat{x} & \\
& y \in \mathbb{R}^n &
\end{array}
$$

he calls the *membership LP*.

It is shown in [47] that

1. For $\hat{x} \in P$ such that $\lfloor \hat{x}_k \rfloor < \lceil \hat{x}_k \rceil$, $\hat{x} \in \text{conv}\{x \in P : x_k \leq \lfloor \hat{x}_k \rfloor \vee x_k \geq \lceil \hat{x}_k \rceil\}$ if and only if (MLP)$_k$ has a solution y such the $y_k \geq \hat{f}_k \lceil \hat{x}_k \rceil$ (Proposition 1).
2. For $\hat{x} \in \text{vert}\, P$ such that $\lfloor \hat{x}_k \rfloor < \hat{x}_k < \lceil \hat{x}_k \rceil$, the unique solution to (MLP)$_k$ is $\bar{y} = \hat{f}_k \hat{x}$; hence $\bar{y}_k < \hat{f}_k \lceil \hat{x}_k \rceil$ (Proposition 2).

These results imply that an optimal solution $(\bar{\lambda}, \bar{s}, \bar{t})$ to the dual (10.14) of (MLP)$_k$ provides a valid inequality that cuts off $\hat{x}$, namely

$$\bar{s}^T x + (x_k - \lfloor \hat{x}_k \rfloor) b^T \bar{\lambda} - (x_k - \lfloor \hat{x}_k \rfloor) \lceil \hat{x}_k \rceil \geq 0$$

or

$$\bar{s}^T x + (b^T \bar{\lambda} - \lceil \hat{x}_k \rceil) x_k \geq (b^T \bar{\lambda} - \lceil \bar{x}_k \rceil) \lfloor \bar{x}_k \rfloor. \tag{10.15}$$

Furthermore, it is shown in [47] that if $\hat{x} \in \text{vert}\, P$, (10.15) is the simple disjunctive cut from (10.13). Since this latter cut can be read off the simplex tableau associated with the basic solution $\hat{x}$ to P, it would seem that using the membership LP in this context does not bring any advantage. However, when $\hat{x} \in P \setminus \text{vert}\, P$, i.e. when $\hat{x} \in P$ is not a basic solution, then the cut (10.15) is something different; namely, it is the simple disjunctive cut from the simplex tableau associated with the optimal basis of (MLP)$_k$, which is also a basis for the linear system defining P different from the basis of $\hat{x}$. This circumstance was used in [47] to develop an efficient algorithm for approximating the strengthened lift-and-project closure by systematically generating rank 1 inequalities that cut off the optimal solution $\tilde{x}$ to the linear program amended with the cuts. Furthermore, this procedure for generating rank 1 lift-and-project cuts through the use of the membership LP was also incorporated in CPLEX 12.5.1 [117] with favorable computational results.

10.8 The Split Closure

An important concept in integer programming due to Chvátal [54], is that of closure. It is mainly used to assess the strength of various classes of cutting planes applied to the LP relaxation P of a MIP. Given a class $\mathcal{C}$ of cutting planes, the $\mathcal{C}$-closure of P is the set obtained by applying to P all the cuts in $\mathcal{C}$. We distinguish between the *elementary* closure, obtained by applying the cuts to P without reoptimizing after adding cuts, and the *rank* k closure, obtained by recursively adding cuts and reoptimizing k times. When $\mathcal{C}$ is the set of split cuts, i.e. cuts from a split disjunction

$$\pi x \leq \pi_0 \quad \vee \quad \pi x \geq \pi_0 + 1 \tag{10.16}$$

where (π, π_0) is an integer vector, the $\mathcal{C}$-closure is called the *split closure*. It was shown in [61] that if P is a rational polyhedron, its split closure is also a rational polyhedron.

This raises the question as to whether optimizing a linear function over the split closure can be done in polynomial time. The answer was shown in [51, 63] to be negative: this problem is $\mathcal{NP}$-hard. It then follows that it is also $\mathcal{NP}$-hard to find a split that cuts off a given fractional point $\bar{x} \in P$ or show that none exists. Another question, addressed in [67], is: what are the characteristics of the split closure of a strictly convex body. The answer is that while such a closure is defined by a finite number of split disjunctions, it is nevertheless not necessarily polyhedral. In a very recent paper, Basu and Molinaro [41] analyze split cuts from the perspective of cut generating functions via geometric lifting. The literature on the subject keeps growing.

Recently, there have been several computational studies meant to assess empirically the strength of different closures: the Chvátal-Gomory closure [75], the lift-and-project closure [48], the split closure [38] and the MIR closure [69]. From a disjunctive programming point of view, the most surprising finding was that optimizing over the elementary split closure yields a value unexpectedly close to the optimum of the mixed integer program itself [38]. Here are some details:

Consider a mixed integer program

$$\min\{cx : Ax \geq b, x_j \text{ integer}, j \in N_1 \subseteq N\}, \tag{MIP}$$

where $A_{m \times n}$ and b are rational, $N = \{1, \ldots, n\}$ and $Ax \geq b$ subsumes both $x \geq 0$ and the upper bounds on x_j, $j \in N_1$. Let P be the constraint set of the LP relaxation of (MIP), i.e. $P = \{x \in \mathbb{R}^n : Ax \geq b\}$ and let $P_I = \text{conv}(P \cap \{x : x_j \in \mathbb{Z}, j \in N_1\})$. Given any integer vector $(\pi, \pi_0) \in \mathbb{Z}^{n+1}$ such that $\pi_j = 0$, $j \in N \setminus N_1$, we call a split cut any valid inequality for P_I derived from a disjunction of the form

$$\begin{pmatrix} Ax \geq b \\ -\pi x \geq -\pi_0 \end{pmatrix} \vee \begin{pmatrix} Ax \geq b \\ \pi x \geq \pi_0 + 1 \end{pmatrix} \tag{$D(\pi, \pi_0)$}$$

Split cuts derived from $D(\pi, \pi_0)$ directly, i.e. without iterating the cut generating procedure, are elementary (or rank 1). We will denote by $\mathcal{C}$ the elementary split closure of P.

Optimizing over the elementary split closure is accomplished by repeatedly solving the following separation problem: given a fractional point, say x, find a rank 1 split cut violated by x or show that none exists. This separation problem is formulated in [38] as a parametric mixed integer linear program with a single parameter in the objective function and the right hand side. A specialized algorithmic framework is then used to generate and solve the resulting sequence of MIP's. This procedure was implemented in COIN-OR [55], using CPLEX to solve the MIP's.

The procedure was applied to all of the instances of MIPLIB 3.0 and several classes of structured IP's and MIP's. It was run on each instance until either the objective function (representing the difference between the left and right sides of a potential cut) was shown to have a nonnegative minimum, thus proving that split cuts cannot cut more of P, or else no new violated cut could be found within 1 h of the last one. The outcome was as follows:

- The percentage of the duality gap closed was 72.78% for the 41 MIP's, and 72.47% for the 24 pure IP's. For 24 of the 65 (41+24) instances, which is more than $\frac{1}{3}$, the gap closed was between 98 and 100%.
- For the specially structured IP's and MIP's the gap closed, as it was to be expected, differed substantially between classes:
  - For (capacitated) facility location problems, the gap closed was 100% for 37 instances, between 99 and 100% for 20 instances, and between 86 and 100% for the remaining 83 instances.
  - For network design problems, on 20 fixed charge network flow instances the gap closed was between 84 and 99%, but for 35 multicommodity capacitated network design instances it was between 44 and 94%.
  - Finally, for 100 capacitated lot sizing instances the gap closed was between 69 and 96%.

Thus the main finding of [38] is that the elementary split closure closes a remarkably high proportion (72–73%!) of the integrality gap.

Chapter 11
Cuts from General Disjunctions

11.1 Intersection Cuts from Multiple Rows

In the early years of the twenty-first century the topic of cutting planes from split disjunctions seemed to have been exhausted, and attention turned to cuts from more general (non-split) disjunctions. This started with the pioneering 2007 paper by Andersen et al. [2] on intersection cuts from two rows of the simplex tableau, which was followed within 3 years by a number of contributions that rounded out the theory behind such cuts [16, 40, 49, 64, 70]. While the P_I-free or lattice-free convex set used to derive the first intersection cuts tested in practice was the strip defined by a row of the simplex tableau associated with an integer-constrained fractional basic variable, the above sequence of papers looked at intersection cuts derived from lattice-free convex sets defined by two such rows of the simplex tableau. But whereas a single row defines a unique lattice-free convex set, a pair of rows can serve as a basis for deriving several classes of lattice-free convex sets. The latter can be classified as strips, triangles of several types, and quadrilaterals. Precise classifications have been given in the above papers; here we use the geometric interpretation of [16, 37] to tie together these various configurations as the possible shapes of a parametric cross-polytope (octahedron). One advantage of this interpretation is that it applies to the more general case of intersection cuts from multiple rows of the simplex tableau.

Consider the q-dimensional unit cube centered at $(0, \ldots, 0)$, $K_q := \{x \in \mathbb{R}^q : -\frac{1}{2} \leq x_j \leq \frac{1}{2},\ j \in Q\}$. Its polar, $K_q^o := \{x \in \mathbb{R}^q : xy \leq 1,\ \forall y \in K_q\}$, 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 [5] of K_q:

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

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_11

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

$$\begin{aligned} -x_1 - \cdots - x_q &\leq \tfrac{1}{2}q \\ x_1 - \cdots - x_q &\leq \tfrac{1}{2}q \\ &\vdots \\ x_1 + \cdots + x_q &\leq \tfrac{1}{2}q \end{aligned} \tag{11.1}$$

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

The first inequality of (11.1) has all coefficients negative. This is followed by q inequalities with exactly one positive coefficient in all possible positions. The next subset contains all $q(q-1)$ inequalities with exactly two positive coefficients, etc., and the last inequality has all q coefficients positive.

Moving the center of the coordinate system to $(\frac{1}{2}, \cdots, \frac{1}{2})$ changes the righthand side coefficient of the i-th inequality in (11.1) from $\frac{1}{2}q$ to a value equal to the sum of positive coefficients on the lefthand side of the inequality. Indeed, if q^+ and q^- denotes the number of positive and negative coefficients, then $\frac{1}{2}q + \frac{1}{2}q^+ - \frac{1}{2}q^- = q^+$.

Next we replace K_q^* with its parametric version, by introducing the parameters $v_{ik}, i = 1, \ldots, t = 2^q, k = 1, \ldots, q$, to obtain the system

$$\begin{aligned} -v_{11}x_1 - \cdots - v_{1q}x_q &\leq 0 \\ v_{21}x_1 - \cdots - v_{2q}x_q &\leq v_{21} \\ -v_{31}x_1 + \cdots - v_{3q}x_q &\leq v_{32} \\ \vdots \qquad\qquad &\quad \vdots \\ v_{t1}x_1 + \cdots + v_{tq}x_q &\leq v_{t1} + \ldots + v_{tq} \\ v_{ik} \geq 0,\ i = 1, \ldots, t = 2^q,\ k &= 1, \ldots, q. \end{aligned} \tag{11.2}$$

Note that the constraints of (11.2) 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 (11.2) is homogeneous in the parameters v_{ik}, so every one of its inequalities can be normalized. For reasons explained in [37], it is convenient to use the normalization

$$
\begin{aligned}
f_1 v_{11} + \cdots + \quad & f_q v_{1q} = 1 \\
(1-f_1)v_{21} + \cdots + \quad & f_q v_{2q} = 1 \\
\cdots \quad & \quad \cdots \\
(1-f_1)v_{t1} + \cdots + (1 & - f_q)v_{tq} = 1
\end{aligned}
\tag{11.3}
$$

where the f_i come from the representation of x_i as

$$x_i = f_i - \sum_{j \in J} \bar{a}_{ij} s_j, \quad s_j \geq 0, j \in J$$

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 (11.2) and (11.3). It is not hard to see that for any fixed set of v_{ik}, (11.2) 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 $f - \sum_{j \in J} \bar{a}_j s_j$ for x into (11.2), we have

Theorem 11.1 *For any values of the parameters v_{ik} satisfying (11.2) and (11.3), the intersection cut $\tilde{\alpha}s \geq 1$ from $\widetilde{K}^{*(n)}(v)$ has coefficients $\tilde{\alpha}_j = \frac{1}{s_j^*}$, where*

$$s_j^* = \max\{s_j : f - \bar{a}_j s_j \in K^{*(n)}(v)\}. \tag{11.4}$$

Proof This is a special case of the basic Theorem 1.1 on intersection cuts. □

The paper [37] gives a sufficient condition for $\tilde{\alpha}s \geq 1$ to be facet defining for the integer hull.

In the case of $q = 2$, i.e. of intersection cuts from two rows of the simplex tableau, the system (11.3) defines the parametric octahedron

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

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. Depending on the values taken by the parameters, $P_{\text{octa}}(v, w)$ may be a quadrilateral (i.e. a full-fledged octahedron in $\mathbb{R}^2$), a triangle, or an infinite strip. In the rest of our discussion we refer to these configurations using the short reference indicated in parenthesis. It can easily be verified that the value-configurations of the parameters v_i, w_i which give rise to maximal convex sets are the following:

- (S) If exactly four components of (v, w) are positive, P_{octa} is the vertical strip $\{x \in \mathbb{R}^2 : 0 \le x_1 \le 1\}$ if $v_i > 0, i = 1, \ldots, 4$; or the horizontal strip $\{x \in \mathbb{R}^2 : 0 \le x_2 \le 1\}$ if $w_i > 0, i = 1, \ldots, 4$ (see Fig. 11.1a, b).
- (T_A) If exactly five components of (v, w) are positive, P_{octa} is a triangle with 1 tilted face (type A) (by "tilted" we mean a face that is not parallel to any of the two axes). Figure 11.1c illustrates the case with $v_1, w_2, v_3, w_3, v_4 > 0$; $w_1, v_2, w_4 = 0$. When in addition $v_i = w_i$ for some $i \in \{1, \ldots, 4\}$ P_{octa} becomes a triangle with vertices $(0, 0)$; $(2, 0)$; $(0, 2)$ or one of the other three configurations symmetric to this one. This corresponds to what is called a triangle of type 1 in [70]. In the general case T_A corresponds to a triangle of type 2 in [70].
- (T_B) If exactly six components of (v, w) are positive, P_{octa} is a triangle with 2 tilted faces (type B). Figure 11.1d illustrates the case with

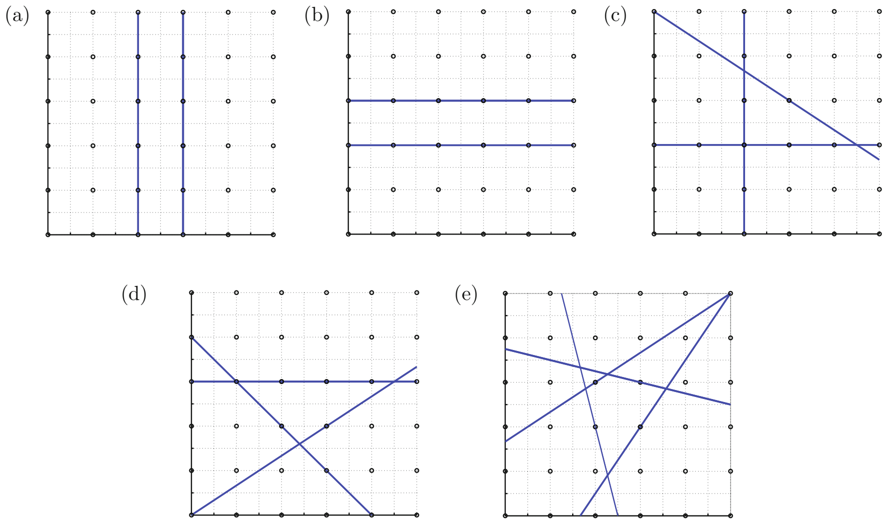


Fig. 11.1 Configurations of the parametric octahedron. (**a**) 4 non-zeros—vertical strip. (**b**) 4 non-zeros—horizontal strip. (**c**) 5 non-zeros—triangle of type A. (**d**) 6 non-zeros—triangle of type B. (**e**) 8 non-zeros—quadrilateral

$v_1, w_1, v_2, w_2, w_3, w_4 > 0$; $v_3, v_4 = 0$. This configuration corresponds to a triangle of type 2 in [70].

- (Q) If all eight components of (v, w) are positive, P_{octa} is a quadrilateral. See Fig. 11.1e.

The case with seven components of (v, w) positive does not correspond to a maximal parametric octahedron.

The sufficient condition for an intersection cut from the q-dimensional parametric cross-polytope to be facet defining for the integer hull specializes in the 2-dimensional case to easily verifiable conditions for each of the possible cut configurations. Necessary and sufficient conditions for inequalities in the class discussed here to be facet defining for the integer hull are given in [49].

11.2 Standard Versus Restricted Intersection Cuts

Intersection cuts were originally defined [4] as inequalities obtained by intersecting the extreme rays of the LP cone $C(J)$, where J is the cobasis of the LP optimal solution, with the boundary of a P_I-free convex set S, i.e. a convex set whose interior contains no *feasible integer point*.

However, when interest in intersection cuts was more recently revived as a result of its refocussing on multiple rows of a simplex tableau (see [2] and the ensuing literature), the above definition was replaced with a narrower one, namely as inequalities obtained by intersecting the extreme rays of $C(J)$ with the boundary of a lattice-free convex set S', i.e. a convex set whose interior contains no integer point (feasible or infeasible).

This definition is more restrictive than the original one, since it excludes intersection cuts obtained from P_I-free sets that are not lattice-free, whereas the original definition includes all intersection cuts from convex lattice-free sets, as these are all P_I-free. In the discussion below we will refer to intersection cuts obtained from P_I-free convex sets as standard (SIC), and to those obtained from lattice-free convex sets as restricted (RIC).

In most of the specific cases considered so far in the literature this difference does not matter, since the lattice-free sets used to generate cuts are P_I-free. This is the case with split cuts and cuts obtained by combining splits, like cuts from triangles or quadrilaterals. But if the lattice-free set S' has a facet whose relative interior contains only infeasible integer points, then switching to a P_I-free set S larger than S' may yield a stronger cut. Furthermore, intersection cuts from a lattice-free set S', when expressed in terms of the nonbasic variables, have all their coefficients nonnegative, as is easily seen from the definition of the cut. On the other hand, intersection cuts from a P_I-free set may have negative coefficients in terms of the nonbasic variables. This is easiest to see if we express the intersection cut from the P_I-free polyhedron S with facets defined by $\sum_{j\in J} d_{ij}x_j \leq d_{i0}$, $i \in Q$, as disjunctive cuts, $\delta x \geq \delta_0$

from $\vee_{i \in Q}(d_{ij}x_j \geq d_{i0})$, having coefficients

$$\delta_j = \max_{i \in Q} \frac{d_{ij}}{d_{i0}}.$$

Clearly, if $d_{ij} < 0$ for all $i \in Q$, then $\delta_j < 0$. This cannot occur for a lattice-free convex set S', since in the case of the latter, the only rays that do not intersect the boundary of S' are those parallel to some facet of S', in which case they have $d_{ij} = 0$ in the inequality defining that facet.

Example 1 Consider the instance

$$\begin{aligned} \min \quad & x_1 + 2x_2 \\ & 4x_1 + 4x_2 \geq 3 \\ & -x_1 + 3x_2 \geq \tfrac{5}{4} \\ & 2x_1 + 4x_2 \leq 5 \\ & x_1, \quad x_2 \geq 0 \quad \text{integer} \end{aligned}$$

whose linear programming relaxation is the shaded area in Fig. 11.2. The optimal LP solution is $\bar{x} = (\frac{1}{4}, \frac{2}{4})$, and the associated simplex tableau is

		x_1	x_2	s_1	s_2	s_3
x_1	$\frac{1}{4}$	1	0	$\frac{3}{16}$	$-\frac{1}{4}$	0
x_2	$\frac{1}{2}$	0	1	$\frac{1}{16}$	$\frac{1}{4}$	0
s_3	$\frac{5}{2}$	0	0	$-\frac{5}{8}$	$-\frac{1}{2}$	1

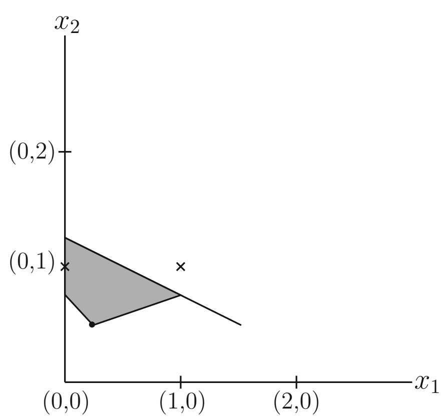


Fig. 11.2 Example 1: LP relaxation

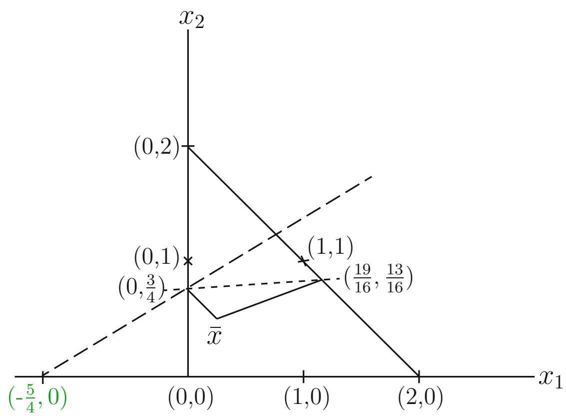


Fig. 11.3 Intersection cut from S' and from S

The intersection cut from the lattice-free triangle $S' := \{x \in \mathbb{R}_2 : x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \leq 2\}$, shown in Fig. 11.3, is $(-\frac{1}{19})x_1 + x_2 \geq \frac{3}{4}$, defined by the two intersection points $(0, \frac{3}{4})$ and $(\frac{19}{16}, \frac{13}{16})$. But since the integer point (1,1) is infeasible, the lattice-free set S' can be replaced with the P_I-free set $S := \{x \in \mathbb{R}^2 : x_1 \geq 0, x_2 \geq 0\}$ which yields the cut $-\frac{4}{5}x_1 + \frac{4}{3}x_2 \geq 1$, defined by the two intersection points $(0, \frac{3}{4})$ and $(-\frac{5}{4}, 0)$. Note that the latter point is obtained by intersecting the hyperplane $x_2 = 0$ with the *negative* extension of the edge $\binom{\bar{x}_1}{\bar{x}_2} + \binom{\frac{3}{16}}{\frac{1}{16}}\lambda$.

More generally, when considering the *class* of cuts from lattice-free convex sets versus the *class* of cuts from P_I-free convex sets, the two are significantly different and the latter, as it is to be expected, is considerably larger than the former and its strongest members dominate those of the former. However, the restricted definition defines a class that is more "regular". This has led to the discovery of some basic properties of RIC's described as properties of intersection cuts, although they do not apply to intersection cuts as originally defined. In particular, an interesting feature of RIC's described in [56] is that if corner(J) denotes the corner polyhedron associated with the cobasis J, i.e. the convex hull of integer points in $C(J)$, then every nontrivial inequality defining a facet of the corner(J) is an intersection cut. The proof uses the definition of intersection cuts from lattice-free sets, and it breaks down if we replace the lattice-free set by a P_I-free set. In other words, the result, while correct for RIC's, is not valid for SIC's. On the other hand, SIC's have a much stronger property: they define the facets of the integer hull itself:

Theorem 11.2 ([25]) *Every facet of* $\mathrm{conv}P_I$ *that cuts off some vertex of* P *is defined by a standard intersection cut.*

Proof Let F be a facet of $\mathrm{conv}P_I$ defined by the inequality $\varphi x \geq \varphi_0$ satisfied by all $x \in P_I$, but violated by some $x \in P$. Then F contains $\dim \mathrm{conv}P_I$ affinely independent integer points of convP_I, and

$$\{x \in \mathbb{R}^n : \varphi x < \varphi_0\} \cap P_I = \emptyset.$$

Hence the interior of the set $T := \{x \in \mathbb{R} : \varphi x \leq \varphi_0\}$ contains no point of P_I, i.e. T is a P_I-free convex set. On the other hand, int T contains some vertex v of P cut off by F. Hence the standard intersection cut from $C(B(v))$, the cone associated with the basis B defining the vertex v, is precisely $\varphi x \geq \varphi_0$. □

Corollary 11.3 *Every vertex v of a corner polyhedron such that $v \notin \text{conv} P_I$ is cut off by some SIC.*

As discussed in Chap. 1, intersection cuts from a given polyhedron S are equivalent to cuts from the disjunction between the inequalities that are the weak complements of those defining S. Looked at from this perspective, clearly both definitions of intersection cuts give rise to disjunctive cuts equivalent to their counterparts. The difference is that the disjunction corresponding to a P_I-free set may be tighter (more constraining) than the disjunction corresponding to a lattice-free set.

11.3 Generalized Intersection Cuts

A related development that we must briefly describe before discussing lift-and-project cuts from arbitrary (non-split) disjunctions, is the introduction of generalized intersection cuts (GIC's) [27]. These are cuts obtained by intersecting the boundary of a P_I-free convex set S (defined in the same way as for intersection cuts) with the edges or rays of some polyhedron C containing the LP relaxation P. If C is chosen to be P itself, then the resulting cuts define conv$(P \setminus \text{int}\, S)$.

The motivation for this class of cuts is as follows. It is well known that the recursive cut generation procedure (generate a cut, add it to the constraint set and reoptimize, generate the next cut, add it and reoptimize, etc.) leads to dual degeneracy, multiple optima for the LP relaxation, the need of ever larger determinants to distinguish between optimal bases close to each other, etc. which tends to create numerical difficulties. In order to avoid such difficulties, a new cut generating paradigm is proposed along the following lines: generate a collection of cuts of increasing strength until you reach a predetermined limit on their number; then use cuts from the resulting collection, chosen by different criteria, to tighten the LP relaxation, thus being able to reach deeper cuts without recourse to recursion.

The basic result on GIC's is as follows. Let S be a P_I-free convex set such that $\bar{x} \in \text{int}\, S$. Let $C_K := \{x \in \mathbb{R}^n : \tilde{A}_K x \geq \tilde{b}_K\}$, where $\tilde{A}_K$ is a submatrix consisting of the rows of $\tilde{A}$ indexed by K, $|K| \geq n$ and with rank$(\tilde{A}_K) = $ rank$(\tilde{A})$. Further, for any extreme ray (extended edge) r^j originating at a vertex $v \in \text{int}\, S$ of C_K, let $p^j = r^j \cap \text{bd}\, S$, and let $\mathcal{P}$ be the index set of such intersection points. Finally, let $\mathcal{R}$ be the index set of extreme rays of C_K such that $r^j \cap \text{bd}\, S = \emptyset$. Then we have

Theorem 11.4 ([27]) *Any solution to either of the systems*

$$\begin{aligned} \alpha p^j &\geq \beta, \quad p^j \in \mathcal{P} \\ \alpha r^j &\geq 0, \quad r^j \in \mathcal{R} \end{aligned} \tag{11.5}$$

for $\beta \in \{1, -1\}$ *such that* $\alpha z < \beta$ *for some vertex* z *of* C_k *yields a valid cut for* $\operatorname{conv} P_I$.

It is not hard to see that the collection of all GIC's from P and S is $\operatorname{conv}(P \setminus \operatorname{int} S)$.

11.4 Generalized Intersection Cuts and Lift-and-Project Cuts

In this section we examine the correspondence between GIC's from a P_I-free set S and $L\&P$ cuts from a disjunction related to S in a particular way.

As before, let $P := \{x \in \mathbb{R}^n : \tilde{A}x \geq \tilde{b}\}$, and let

$$S := \{x \in \mathbb{R}^n : d^t x \leq d_{t0}, \quad t \in T\}$$

be a maximal P_I-free polyhedron. We associate with P and S the disjunction

$$D(P, S) = \left\{ x \in \mathbb{R}^n : \bigvee_{t \in T} \begin{pmatrix} \tilde{A}x \geq \tilde{b} \\ d^t x \geq d_{t0} \end{pmatrix}, \right\}$$

and the CGLP constraint set corresponding to $D(P, S)$

$$\begin{aligned} \alpha - u^t \tilde{A} - u_0^t d^t &= 0 \\ -\beta + u^t \tilde{b} + u_0^t d_{t0} &= 0, \ t \in T \\ \textstyle\sum_{t \in T} (u^t e + u_0^t) &= 1 \\ u^t, u_0^t &\geq 0, \ t \in T \end{aligned} \tag{11.6}$$

Theorem 11.5 *The family of GIC's from S is equivalent to the family of L&P cuts from* $D(P, S)$ *corresponding to those basic feasible solutions to (11.6) such that* $u_0^t > 0, \forall t \in T$.

Proof The family of L&P cuts from $D(P, S)$ is known to define the convex hull of the union of polyhedra $P^t := \{x \in \mathbb{R}^n : \tilde{A}x \geq \tilde{b}, d^t x \geq d_{t0}\}, t \in T$; and the members of this family associated with basic feasible solutions to (11.6) such that $u_0^t > 0, \forall t \in T$, are known to define all the facets of this convex hull that do not belong to the constraints defining P. Since the collection of these facets defines $\mathrm{conv}(P \setminus \mathrm{int}\, S)$, it implies all GIC's.

Conversely, by the definition of GIC's, they are all the valid inequalities for $\mathrm{conv}(P \setminus \mathrm{int}\, S)$ which cut off some part of P. □

Consider now a disjunction of a more general type, namely one in which the inequalities $d^t x \geq d_{t0}$ are replaced with sets of (multiple) inequalities $D^t x \geq d_0^t$:

$$\left\{ x \in \mathbb{R}^n : \bigvee_{t \in T} \begin{pmatrix} \tilde{A}x \geq \tilde{b} \\ D^t x \geq d_0^t \end{pmatrix} \right\}. \tag{11.7}$$

In this case the system (11.6) becomes

$$\begin{aligned} \alpha - u^t \tilde{A} - v^t D^t &= 0 \\ -\beta + u^t \tilde{b} + v^t d_0^t &= 0 \quad t \in T \\ \textstyle\sum_{t \in T} (u^t e + v^t e) &= 1 \\ u^t, v^t &\geq 0, \; t \in T \end{aligned} \tag{11.8}$$

and Theorem 11.5 generalizes to the following

Theorem 11.6 *Let $\bar{\alpha} x \geq \bar{\beta}$ be the L&P cut associated with the basic feasible solution $(\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t \in T})$ to (11.8), where $\bar{v}^t e > 0$ for all $t \in T$. Then $\bar{\alpha} x \geq \bar{\beta}$ is equivalent to a GIC from the P_I-free polyhedron*

$$S(\bar{v}) := \{x \in \mathbb{R}^n : (\bar{v}^t D^t)x \leq \bar{v}^t d_0^t, t \in T\}.$$

Proof Obviously, $\mathrm{int}\, S(\bar{v})$ contains no point satisfying the disjunction (11.7). If we denote $\delta^t := \bar{v}^t D^t$, $\delta_{t0} := \bar{v}^t d_0^t$, $t \in T$, then $S(\bar{v})$ becomes

$$S(\bar{v}) = \{x \in \mathbb{R}^n : \delta^t x \leq \delta_0^t, t \in T\},$$

and (11.8) takes the form of (11.6) with δ^t and δ_{t0} taking the place of d^t and d_{t0}, respectively. On the other hand, from Theorem 11.5, the family of GIC's from $S(\bar{v})$ is equivalent to the family of those L&P cuts associated with basic feasible solutions to (11.6) such that $u_0^t > 0$ for all $t \in T$. □

Thus while L&P cuts from a disjunction of the form $D(P, S)$ correspond to GIC's from the P_I-free polyhedron S, L&P cuts from a disjunction of the form (11.7) correspond to GIC's from the family of P_I-free polyhedra

$$S(v) = \{x \in \mathbb{R}^n : (v^t D^t)x \leq v^t d_0^t, t \in T\}.$$

11.5 Standard Intersection Cuts and Lift-and-Project Cuts

Recall that in the case of split cuts, a precise correspondence could be established [33] between intersection cuts and L&P cuts. Namely, if the intersection cut $\pi x \geq 1$ is derived from row k of a simplex tableau defined by the nonbasic index set J, then it was shown to be equivalent to the L&P cut $\alpha x \geq \beta$ associated with any basic feasible solution $(\alpha, \beta, u, u_0, v, v_0)$ to the CGLP such that the basic components of u and v are indexed by a certain partition $[M_1, M_2]$ of J that depends on row k. Unfortunately, this correspondence does not extend to cuts from non-split disjunctions. Here we examine the general case, following [26]. First we show that an intersection cut from a P_I-free polyhedron S can be represented as a L&P cut from a solution to the CGLP constraint set (11.6), then we give a sufficient condition for a L&P cut from (11.6) to correspond to an intersection cut from S. This raises the intriguing question as to what happens when the sufficient condition is not satisfied; which we address in the sequel.

Recall from Chap. 1 that the intersection cut from an optimal LP solution $x = \bar{x} - \sum_{j \in J} \bar{a}^j x_j$ and a P_I-free convex set S is

$$\sum_{j \in J} \frac{1}{\lambda_j^*} x_j \geq 1,$$

where

$$\lambda_j^* = \max\{\lambda_j : \bar{x} - \bar{a}^j \lambda_j \in S\}.$$

More specifically, using the expression of S in terms of its boundary planes,

$$S := \{x \in \mathbb{R}^n : d^t x \leq d_{t0}, t \in T\}$$

and denoting $\bar{x} = \bar{a}_0$, we have

$$\begin{aligned} \lambda_{jt}^* &= \max\{\lambda_{jt} : d^t(\bar{a}_0 - \bar{a}^j \lambda_{jt}) \leq d_{t0}\}, \quad t \in T \\ &= \frac{d_{t0} - d^t \bar{a}_0}{-d^t \bar{a}^j} \end{aligned}$$

and

$$\lambda_j^* = \min_{t \in T} \frac{d_{t0} - d^t \bar{a}_0}{-d^t \bar{a}^j}.$$

We will also use for the intersection cut the alternative notation $\pi x_J \geq 1$, with

$$\pi_j = \max_{t \in T} \pi_j^t (= \frac{1}{\lambda_j^*}), \text{ where } \pi_j^t = \frac{1}{\lambda_{jt}^*}.$$

Denoting the above fraction by $\dfrac{\bar{d}_{t0}}{\bar{d}_j^t}$ and defining $\hat{A} = \tilde{A}_J, \hat{b} = \tilde{b}_J$, we have

Theorem 11.7 *The intersection cut $\pi x_J \geq 1$ from S and the simplex tableau defined by J is equivalent to the L&P cut $\alpha x_N \geq \beta$ from a basic feasible solution to (11.6) in which, for each $t \in T$, all but one of the variables u_j^t with $j \in J$ are basic and all the variables u_j^t with $j \notin J$ are nonbasic, except for the u_0^t, which are all basic and positive. The solution of (11.6) is given by*

$$\theta\alpha = \pi \hat{A}$$

$$\theta\beta = \pi \hat{b} + 1$$

$$\theta u_j^t = \pi - \pi^t, t \in T$$

$$\theta u_0^t = 1/(-\bar{d}_0^t), t \in T$$

where $\theta > 0$ is a scaling factor.

Proof First we verify that $\alpha = u_J^t \hat{A} + u_0^t d^t$. That is,

$$\theta(\alpha - u_J^t \hat{A} - u_0^t d^t) = \pi \hat{A} - (\pi - \pi^t)\hat{A} + \frac{1}{\bar{d}_{t0}} d^t = \pi^t \hat{A} + \frac{1}{\bar{d}_{t0}} d^t = \sum_{j \in J} \frac{d^t \bar{a}_j}{\bar{d}_{t0}} \hat{A}_j + \frac{1}{\bar{d}_{t0}} d^t$$

$$= \frac{1}{\bar{d}_{t0}} \left(\sum_{j \in J, i \in N} d_i^t (\bar{a}_{ij}) \hat{A}_j + d^t \right) = \frac{1}{\bar{d}_{t0}} \left(\sum_{i \in N} d_i^t \sum_{j \in J} (-e_i^T \hat{A}^{-1})_j \hat{A}_j + d^t \right) = 0,$$

where we used the fact that $-(e_i^T \hat{A}^{-1})_j = \bar{a}_{ij}$ [33], and $\sum_{j \in J} (e_i^T \hat{A}^{-1})_j \hat{A}_j = e_i^T$. Here θ is well defined, as both sides of the equation are of the same sign. Since $\bar{a}_0 \in \text{int}\, S$, each u_0^t is positive. One may similarly verify that $\beta = u_J^t \hat{b} + u_0^t d_{t0}$. The scaling factor θ is used to ensure that $\sum_{t \in T} (u_0^t + \sum_{j \in J} u_j^t) = 1$.

Now we show that the two cuts are equivalent. That is, using the equations already proved, and the fact that $\hat{A} x_N - \hat{b} = x_J$ [33], we have

$$\theta(\alpha x_N - \beta) = \pi \hat{A} x_N - (\pi \hat{b} + 1) = \pi x_J - 1.$$

Finally, we prove that the solution of CGLP constructed above is basic. First, we remove from (11.6) the variables u_j^t corresponding to $j \notin J$, and eliminate the α

and β variables to obtain

$$\begin{aligned} u_J^1 \hat{A} - u_J^t \hat{A} + \quad u_0^1 d^1 - \ u_0^t d^t &= 0 \\ u_J^1 \hat{b} - u_J^t \hat{b} + \quad u_0^1 d_{10} - u_0^t d_{t0} &= 0 \end{aligned} \quad t \in T \setminus \{1\}$$
$$\sum_{t \in T} (u^t e + \quad u_0^t) = 1 \qquad (11.9)$$
$$u^t, u_0^t \geq 0,\ t \in T.$$

Now we construct a basis of (11.9). Let $M_t' = \{j \in J \ : \ (\pi - \pi^t)_j > 0\}$. Notice that no $j \in J$ may belong to all the sets M_t', $t \in T$, since $\pi_j = \max_{t \in T} \pi_j^t$, and thus for each $j \in J$ there exists $t \in T$ with $(\pi - \pi^t)_j = 0$. If some $j \in J$ belongs to less than $|T| - 1$ of the sets M_t' (which may occur if $\pi_j = \pi_j^t$ for more than one $t \in T$), then we assign each such j arbitrarily to some of the sets M_t' so that finally we obtain the sets M_t, $t \in T$, and each $j \in J$ occurs exactly in $|T| - 1$ of these sets.

After these preparations, we claim that the variables $G = \{u_0^t \ : \ t \in T\} \cup \left(\bigcup_{t \in T} \{u_j^t : j \in M_t\}\right)$ constitute a basis of (11.9). To prove our claim, we derive a new system of equations from (11.9) as follows. Since $\hat{A}$ is nonsingular, we can multiply the first equation by $\hat{A}^{-1}$ from the right to get

$$u_J^1 - u_J^t - u_0^1 \bar{d}^1 + u_0^t \bar{d}^t = 0,\ t \in T \setminus \{1\}, \qquad (11.10)$$

where $\bar{d}^1 = -d^1 \hat{A}^{-1}$ and $\bar{d}^t = -d^t \hat{A}^{-1}$ as shown in [22, 92]. By substituting this into the second equation of (11.9), we obtain

$$0 = (u_0^1 \bar{d}^1 - u_0^t \bar{d}^t)\hat{b} + u_0^1 d_{10} - u_0^t d_{t0} = -u_0^1 \bar{d}_{10} + u_0^t \bar{d}_{t0}, \quad t \in T \setminus \{1\} \qquad (11.11)$$

where we used $\bar{d}_{t0} = d^t \hat{A}^{-1} \hat{b} - d_{t0}$ from [22, 92]. Now, the set of variables G does not contain any of the u_j^t with $j \in J \setminus M_t$, $t \in T$. Consequently, (11.10) can be rewritten as

$$u_{M_1}^1 - u_{M_t}^t - u_0^1 \bar{d}^1 + u_0^t \bar{d}^t = 0,\ t \in T \setminus \{1\}.$$

This implies

$$u_{M_k}^k - u_{M_t}^t - u_0^k \bar{d}^k + u_0^t \bar{d}^t = 0,\ k \neq t \in T. \qquad (11.12)$$

By using (11.12) we can express $u_{M_t}^t$ as a combination of the vectors $\bar{d}^t$ and $\bar{d}^k$ as follows. Since each $j \in J$ occurs in exactly $|T| - 1$ sets M_t, for each $t \in T$ and for each $j \in M_t$ there exists a unique $k \in T$ with $j \notin M_k$. Hence, we have

$$u_{M_t}^t = \sum_{k \in T \setminus \{t\}} (u_0^t \bar{d}_{M_t \setminus M_k}^t - u_0^k \bar{d}_{M_t \setminus M_k}^k), \quad t \in T.$$

Therefore, using the last equation of (11.9), we obtain

$$1 = \sum_{t\in T}(u_0^t + u_{M_t}^t e) = \sum_{t\in T} u_0^t(1 + \sum_{k\in T\setminus\{t\}} (\bar{d}_{M_t\setminus M_k}^t - \bar{d}_{M_k\setminus M_t}^t)e). \tag{11.13}$$

Observe that the system consisting of Eq. (11.13) and the equations

$$-u_0^1\bar{d}_{10} + u_0^t\bar{d}_{t0} = 0, \quad t \in T\setminus\{1\} \tag{11.14}$$

involves only the variables u_0^t, $t \in T$. It suffices to show that it has a unique solution, because then the value of the variables $u_{M_t}^t$ is uniquely defined. To prove this, suppose that the coefficient matrix of (11.13)–(11.14) is singular. Using (11.14) this implies

$$(1 + \sum_{k\in T\setminus\{1\}} (\bar{d}_{M_1\setminus M_k}^1 - \bar{d}_{M_k\setminus M_1}^1)e) + \sum_{t\in T\setminus\{1\}} \frac{\bar{d}_{10}}{\bar{d}_{t0}}(1 + \sum_{k\in T\setminus\{t\}} (\bar{d}_{M_t\setminus M_k}^t - \bar{d}_{M_k\setminus M_t}^t)e) = 0.$$

Since $\bar{d}_{10} < 0$ by assumption, we can divide through the last equation by it, and after rearranging terms we get

$$\sum_{t\in T}\left(\frac{1}{\bar{d}_{t0}} + \sum_{k\in T\setminus\{t\}}\left(\frac{1}{\bar{d}_{t0}}\bar{d}_{M_t\setminus M_k}^t - \frac{1}{\bar{d}_{k0}}\bar{d}_{M_t\setminus M_k}^k\right)e\right) = 0$$

However, this last expression is nothing else but -1 times

$$\sum_{t\in T}(1/(-\bar{d}_{t0}) + (\pi - \pi^t)e).$$

But this number is positive, so we have encountered a contradiction. □

Note that Theorem 11.7 is valid for either of the two definitions of an intersection cut, standard or restricted, as long as the convex set S used to derive the intersection cut is the same (whether P_I-free or lattice-free) as the one used in the definition of the CGLP.

Next we turn to the converse direction, and provide a sufficient condition for a L&P cut to represent an intersection cut. Notice that in the proof of Theorem 11.7, all the CGLP variables with positive values have subscripts indexed by the set J, where J corresponds to the nonbasic variables in a simplex tableau of the LP relaxation of (MIP). Moreover, for each $j \in J$, there exists $t \in T$ with $u_j^t = 0$. We will prove that the second one of these conditions holds for all basic solutions to (11.6), whereas the first one is sufficient for a L&P cut to correspond to an intersection cut.

Let M denote the row index set of $\tilde{A}$.

Proposition 11.8 *In any basic solution to (11.6), for every $i \in M$ there exists $t \in T$ such that $u_i^t = 0$.*

Proof By contradiction. Let $w = (\alpha, \beta, \{u^t, u_0^t\}_{t\in T})$ be a basic feasible solution to (11.6), and suppose there exists some $i \in M$ such that $u_i^t > 0$ for all $t \in T$. Let

$$u_i^{t^*} = \min_{t\in T} u_i^t$$

and define a new solution $\bar{w}$ by setting

$$\bar{u}_h^t = \begin{cases} u_h^t - u_i^{t^*} & h = i \\ u_h^t & h \in M \setminus \{i\} \end{cases} \qquad t \in T$$

$$\bar{u}_0^t = u_0^t, \quad t \in T$$

$$\bar{\alpha} = \alpha - u_i^{t^*}\tilde{a}_i \quad \text{(where } \tilde{a}_i \text{ denotes row } i \text{ of } \tilde{A})$$

$$\bar{\beta} = \beta - u_i^{t^*}\tilde{b}_i$$

Clearly $\bar{w}$ satisfies all constraints of (11.6) except for the normalization constraint. We remedy this by rescaling $\bar{w}$ so that in the resulting solution $\tilde{w}$ the sum of the variables $\tilde{u}_h^t$ and $\tilde{u}_0^t$ for all h and t is 1. Now $\tilde{w}$ has one less nonzero component than w, which contradicts the fact that w is a basic solution. □

Theorem 11.9 *Let $(\alpha, \beta, \{u^t, u_0^t\}_{t\in T})$ be a basic feasible solution to (11.6) such that $u_0^t > 0$, $t \in T$. If there exists a nonsingular $n \times n$ submatrix $\tilde{A}_J$ of $\tilde{A}$ such that $u_j^t = 0$ for all $j \notin J$ and $t \in T$, then the L&P cut $\alpha x \geq \beta$ is equivalent to the intersection cut $\pi x_J \geq 1$ from S and the LP simplex tableau with nonbasic set J.*

Proof Suppose the condition of the Theorem is satisfied. Recall the first set of constraints of (11.6): $\alpha - u^t\tilde{A} - u_0^t d^t = 0$ for $t \in T$. Since there exists a nonsingular $\tilde{A}_J$ such that $u_j^t = 0$ for all $j \notin J$, we can restrict the set of variables to u_J^t, $t \in T$. Therefore, we can infer that the solution of (11.6) satisfies the following set of equations:

$$\alpha - u_J^t\hat{A} - u_0^t d^t = 0, \quad t \in T,$$

where $\hat{A} = \tilde{A}_J$. Since $\hat{A}$ is invertible, we can multiply this equation from the right by $\hat{A}^{-1}$ to obtain

$$\alpha\hat{A}^{-1} = u_J^t + u_0^t d^t\hat{A}^{-1}, \quad t \in T.$$

Since for each $j \in J$ there exists $t_j \in T$ with $u_j^{t_j} = 0$ by Proposition 11.8, we have

$$(\alpha\hat{A}^{-1})_j = u_0^{t_j}(d^{t_j}\hat{A}^{-1})_j = u_0^{t_j}\sum_{i\in J} d_i^{t_j}(e_i^T\hat{A}^{-1})_j = u_0^{t_j}\sum_{i\in J} d_i^{t_j}(-\bar{a}_{ij}) = u_0^{t_j}d^{t_j}(-\bar{a}_j).$$

We introduce a scaling factor $\theta > 0$ to be chosen later. Let $\pi_j := \frac{1}{\theta}(\alpha\hat{A}^{-1})_j$, and $\pi_j^t := \frac{1}{\theta}u_0^t d^t(-\bar{a}_j)$ for $t \in T$. Then the above derivation, and $u_j^t \geq 0$ imply that $\pi_j = \max_{t\in T}\pi_j^t$.

Now we use the second equation of (11.6): $\beta - u^t\tilde{b} - u_0^t d_{t0} = 0$, $t \in T$. Again, since $u_j^t = 0$ for $j \notin J$ we can rewrite this equation as

$$\beta = u_J^t \hat{b} + u_0^t d_{t0}, \quad t \in T.$$

Next we substitute for u_J^t the expression $\alpha\hat{A}^{-1} - u_0^t d^t \hat{A}^{-1}$ to obtain

$$\beta = (\alpha\hat{A}^{-1} - u_0^t d^t \hat{A}^{-1})\hat{b} + u_0^t d_{t0} = \alpha\hat{A}^{-1}\hat{b} + u_0^t(d_{t0} - d^t\hat{A}^{-1}\hat{b}), \quad t \in T. \tag{11.15}$$

Since $\hat{A}^{-1}\hat{b} = \bar{a}_0$, we deduce

$$\beta - \alpha\bar{a}_0 = u_0^t d_{t0} - u_0^t d^t \bar{a}_0, \quad t \in T.$$

Here, the left hand side is positive since $\alpha x \geq \beta$ cuts off $\bar{a}_0$, and the right hand side is positive since $\bar{a}_0 \in \operatorname{int} S$. Now we choose θ as $\beta - \alpha\bar{a}_0$, hence

$$u_0^t = \theta/(d_{t0} - d^t\bar{a}_0), \quad t \in T.$$

However, this implies that

$$\pi_j^t = \frac{1}{\theta} u_0^t d^t(-\bar{a}_j) = \frac{d^t(-\bar{a}_j)}{d_{t0} - d^t\bar{a}_0}, \quad t \in T, j \in J.$$

This agrees with our former definition of $\pi_j^t := 1/\lambda_{tj}^*$. Consequently, $\pi x_J \geq 1$ is the intersection cut from S and the basic solution corresponding to the nonbasic set J.

Finally, using the definition of π, we see that $\frac{1}{\theta}\alpha = \pi\hat{A}$, and $\frac{1}{\theta}\beta = \pi\hat{b} + 1$ from (11.15). Hence, we have

$$\frac{1}{\theta}(\alpha x_N - \beta) = \pi x_J - 1,$$

that is, the two cuts are equivalent. □

Now suppose that instead of (11.6), we have a CGLP of the form (11.8), corresponding to a disjunction (11.7) with multiple inequalities per term. Then Theorem 11.9 generalizes to the following.

Theorem 11.10 *Let $(\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t\in T})$ be a basic feasible solution to (11.8) such that $\bar{v}^t e > 0$ for all $t \in T$. If there exists a nonsingular $n \times n$ submatrix $\tilde{A}_J$ of $\tilde{A}$ such that $\bar{u}_j^t = 0$ for all $j \notin J$ and $t \in T$, then the lift-and-project cut $\bar{\alpha}x \geq \bar{\beta}$ is equivalent to the intersection cut $\pi x_J \geq 1$ from the set*

$$S(\bar{v}) := \{x \in \mathbb{R}^N : (\bar{v}^t D^t)x \leq \bar{v}^t d_0^t, t \in T\}$$

and the LP simplex tableau with nonbasic set J.

Proof Analogous to that of Theorem 11.9. □

Theorems 11.9 and 11.10 give sufficient conditions for a lift-and-project cut from a certain disjunction to correspond to an equivalent intersection cut from a convex polyhedron S "complementary" to that disjunction in a well-defined sense. We recall that in this context it does not matter whether the intersection cut in question is from a P_I-free S or just a lattice-free S, as long as the same S is used in the definition of the CGLP as in that of the intersection cut. In the case of Theorem 11.9, i.e. of a simple disjunction $D(P, S)$, S is the polyhedron obtained by complementing (reversing) each of the inequalities of $D(P, S)$ other than those of P; hence S is the same for any solution of the CGLP (11.6). On the contrary, in the case of Theorem 11.10, i.e. of a disjunction of the form (11.7) with multiple inequalities per term, $S(\bar{v})$ is different for different solutions of the CGLP (11.8). To be more specific, $S(\bar{v})$ is the polyhedron obtained by complementing (reversing) each of the combined inequalities $(\bar{v}^t D^t) \geq \bar{v}^t d_0^t$, $t \in T$, where the weights of the combination are part of the CGLP solution. In other words, the intersection cut that corresponds to a given solution of (11.8), comes from a set $S(\bar{v})$ that is itself a function of that solution. Another way of looking at this is to say that the family of intersection cuts corresponding to the family of CGLP solutions comes from a parametric polyhedron S whose parameters are set by the given CGLP solution. While the cuts are valid for any nonnegative parameter values, they are facet defining only for parameter values corresponding to basic feasible solutions of the CGLP.

Given a L&P cut $\alpha x_N \geq \beta$ obtained from a basic feasible solution to the CGLP system (11.6) (or (11.8)), Theorem 11.9 (respectively 11.10) gives a sufficient condition for the existence of an intersection cut $\pi x_J \geq 1$ equivalent to $\alpha x_N \geq \beta$. Next we examine the conditions under which this sufficient condition is also necessary.

An inequality $\gamma^1 x \geq \gamma_{10}^1$ is said to dominate the inequality $\gamma^2 x \geq \gamma_{20}$ on P if every $x \in P$ that satisfies $\gamma^1 x \geq \gamma_{10}$ also satisfies $\gamma^2 x \geq \gamma_{20}$.

Theorem 11.11 *Let $\bar{w} := (\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{u}_0^t\}_{t \in T})$ be a basic feasible solution to (11.6) such that $\bar{u}_0^t > 0$, $t \in T$.*

If $\bar{w}$ does not satisfy the (sufficient) condition of Theorem 11.9, and there is no basic feasible solution $\tilde{w}$ to (11.6) with $(\tilde{\alpha}, \tilde{\beta}) = \mu(\bar{\alpha}, \bar{\beta})$ for some $\mu > 0$ that satisfies the condition of Theorem 11.9, then there exists no intersection cut from S equivalent to $\bar{\alpha} x_N \geq \bar{\beta}$. Furthermore, if $(\bar{\alpha}, \bar{\beta})$ uniquely minimizes $\alpha \bar{x}_N - \beta$ over (11.6), then $\bar{\alpha} \bar{x}_N - \bar{\beta} < \tilde{\alpha} \bar{x}_N - \tilde{\beta}$ for any L&P cut $\tilde{\alpha} x \geq \tilde{\beta}$ equivalent to an intersection cut from S and there exists no intersection cut from S whose L&P equivalent dominates $\bar{\alpha} x \geq \bar{\beta}$ on P.

Proof Suppose LP admits a basis with nonbasic variables J such that the intersection cut $\pi x_J \geq 1$ derived from S is equivalent to $\bar{\alpha} x_N \geq \bar{\beta}$, i.e., there is a scaling factor μ such that $\mu(\bar{\alpha} x_N - \bar{\beta}) = \pi x_J - 1$. Then by Theorem 11.7 there exists a basic feasible solution $(\tilde{\alpha}, \tilde{\beta}, \{\tilde{u}^t, \tilde{u}_0^t\}_{t \in T})$ of CGLP which gives rise to a L&P cut $\tilde{\alpha} x_N \geq \tilde{\beta}$ equivalent to $\pi x_J \geq 1$, i.e., $\theta(\tilde{\alpha} x_N - \tilde{\beta}) = \pi x_J - 1$. But then $(\tilde{\alpha} x_N - \tilde{\beta}) = (\mu/\theta)(\bar{\alpha} x_N - \bar{\beta})$, i.e., the two cuts are equivalent. However, $(\tilde{\alpha}, \tilde{\beta}, \{\tilde{u}^t, \tilde{u}_0^t\}_{t \in T})$ satisfies the conditions of Theorem 11.9, which contradicts the assumption of the theorem.

As for the last statement, if $(\bar{\alpha}, \bar{\beta})$ uniquely minimizes $\alpha\bar{x} - \beta$ over (11.6), then it is a vertex of W^0, the projection onto the (α, β)-subspace of the feasible set of (11.6) without the normalization constraint, see page 28 of [6], in particular condition (ii). Therefore, from Theorem 4.5 of [6], $\bar{\alpha}x \geq \bar{\beta}$ defines a facet of $\text{conv}P_D(\bar{x})$, the convex hull of the disjunctive set $D(P, S)$. But a facet of a polyhedron cannot be dominated by any other valid inequality for the polyhedron, hence $\bar{\alpha}x \geq \bar{\beta}$ cannot be dominated by any L&P cut equivalent to an intersection cut from S, as any such cut is valid for $\text{conv}P_D(\bar{x})$. □

An analogous theorem holds for the sufficient condition of Theorem 11.10.

Theorem 11.12 *Let $\bar{w} = (\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t\in T})$ be a basic feasible solution to (11.8) such that $\bar{v}^t e > 0$, $t \in T$.*

If $\bar{w}$ does not satisfy the (sufficient) condition of Theorem 11.10, and there is no basic feasible solution $\tilde{w}$ to (11.8) with $(\tilde{\alpha}, \tilde{\beta}) = \mu(\bar{\alpha}, \bar{\beta})$ for some $\mu > 0$ that satisfies the condition of Theorem 11.10, then there exists no intersection cut from any member of the family of polyhedra

$$S(v) := \{x \in \mathbb{R}^N : (v^t D^t)x \leq v^t d_{t0}, t \in T\},$$

where $v \geq 0$, $v \neq 0$, equivalent to $\bar{\alpha}\bar{x}_N \geq \bar{\beta}$. Furthermore, if $(\bar{\alpha}, \bar{\beta})$ uniquely minimizes $\alpha\bar{x}_N - \beta$ over (11.8), then $\bar{\alpha}\bar{x}_N - \bar{\beta} < \tilde{\alpha}\bar{x}_N - \tilde{\beta}$ for any L&P cut $\tilde{\alpha}x \geq \tilde{\beta}$ equivalent to an intersection cut from $S(v)$ and there exists no intersection cut from $S(v)$ whose L&P equivalent dominates $\bar{\alpha}x \geq \bar{\beta}$ on P.

Proof (along the lines of the proof of Theorem 11.11).

(a) We prove the first statement by contradiction. Suppose LP admits a basis with nonbasic variables indexed by J such that the intersection cut $\pi x_J \geq 1$, derived from $S(\tilde{v})$ for some $\tilde{v}$ satisfying the conditions of the theorem, is equivalent to $\bar{\alpha}x_N \geq \bar{\beta}$; i.e., there exists a scaling factor $\mu > 0$ such that $\mu(\bar{\alpha}x_N - \bar{\beta}) = \pi' x_N - 1$ (where $\pi' x_N$ is πx_J expressed in the cobasis indexed by N). Then by Theorem 11.9 there exists a basic feasible solution $(\tilde{\alpha}, \tilde{\beta}, \{\tilde{u}^t, \tilde{v}^t\}_{t\in T})$ of CGLP which gives rise to a L&P cut $\tilde{\alpha}x_N \geq \tilde{\beta}$ equivalent to $\pi' x_N \geq 1$, i.e. such that $\theta(\tilde{\alpha}x_N - \tilde{\beta}) = \pi' x_N - 1$. But then $(\tilde{\alpha}x_N - \tilde{\beta}) + \frac{\mu}{\theta}(\bar{\alpha}x_N - \bar{\beta})$, i.e. the two cuts are equivalent. However, $(\tilde{\alpha}, \tilde{\beta}, \{\tilde{u}^t, \tilde{v}^t\}_{t\in T})$ satisfies the conditions of Theorem 11.9, which contradicts the assumption of the current theorem that there is no basic feasible solution $\tilde{w}$ to (11.8) with $(\tilde{\alpha}, \tilde{\beta}) = \mu(\bar{\alpha}, \bar{\beta})$ for some $\mu > 0$ that satisfies the condition of Theorem 11.9.

(b) As to the second statement, if $(\bar{\alpha}, \bar{\beta})$ uniquely minimizes $\alpha\bar{x} - \beta$ over (11.8), then it is an extreme ray of W_0, the projection onto the (α, β)-subspace of the projection cone W underlying (11.8). Therefore $\bar{\alpha}x \geq \bar{\beta}$ defines a facet of $\text{conv}P_D$, the convex hull of the disjunctive set (11.7). But a facet of a polyhedron cannot be dominated by any other valid inequality for the polyhedron, hence $\bar{\alpha}x \geq \bar{\beta}$ cannot be dominated by any L&P cut equivalent to an intersection cut from $S(v)$, as any such cut is valid for $\text{conv}P_D$.

□

A feasible basis for the CGLP system (11.6) (or (11.8)) and the associated solution will be called *regular* if the cut that it defines is equivalent to an intersection cut, i.e. if it satisfies the condition of Theorem 11.9 (respectively 11.10), *irregular* otherwise. In the sequel we discuss the properties of irregular CGLP bases and solutions. A cut defined by an irregular solution w is *irregular*, unless there exists a regular solution w' with the same (α, β)-component as that of w, in which case the cut is *regular*.

11.6 The Significance of Irregular Cuts

The occurrence of irregular cuts may seem as no more than an occasional, accidental occurrence in the cut generating process, but it is nothing of the kind. Instead, as shown in [26], irregular cuts are highly significant, both for their properties and for their frequency. Below we address both of these features.

The fact that cuts from general disjunctions differ in some fundamental ways from cuts from split disjunctions was intimated by some findings in the literature preceding [26]. Thus, Andersen et al. [1], in extending the results of [33] from split disjunctions of the form $x_k \leq 0$ or $x_k \geq 1$ to more general split disjunctions, have used the relationship (in our notation)

$$\operatorname{conv}(P \setminus \operatorname{int} S) = \bigcap_{J \in \mathcal{N}} \operatorname{conv}(C(J) \setminus \operatorname{int} S)$$

to prove that the split closure is polyhedral. Here $C(J)$ is the cone associated with the LP solution $\bar{x}$ defined by the cobasis J, while $\mathcal{N}$ is the collection of all J corresponding to feasible bases. They then explored the question of whether this relationship generalizes to cuts from split disjunctions to cuts from non-split disjunctions (with the corresponding set S), and reached the negative conclusion that cases where the equality in the above equation is replaced by strict inclusion cannot be excluded. They illustrate their finding with a 2-dimensional counterexample. An analogous conclusion was reached and exemplified for non-split two-term disjunctions by Kis [92].

The results of [26] summarized in Theorems 11.7–11.12 above amplify this conclusion and make it more specific, by using the lift-and-project representation to pinpoint the gap between the two sides of the above equation. More specifically, these results characterize those facets of $\operatorname{conv}(P \setminus \operatorname{int} S)$ that are not facets of $\operatorname{conv}(C(J) \setminus \operatorname{int} S)$ for any $J \in \mathcal{N}$.

As mentioned in Sect. 11.2, restricted intersection cuts, i.e. intersection cuts from lattice-free (as opposed to P_I-free) convex sets, define the facets of the corner polyhedron. In other words, the intersection cut derived from the cone $C(J)$ and a lattice-free convex set S is valid not only for the integer program in question, but also for corner(J), the convex polyhedron associated with J. So no intersection cut from $C(J)$ and a lattice-free convex set can cut off any part of corner(J), and obviously this property carries over to those regular lift-and-project cuts equivalent to them. On the other hand, irregular L&P cuts may cut off parts of the corner polyhedron associated with the cobasis J defining the objective function of the (CGLP). Next we describe some situations in which specific parts of corner(J) are cut off by an irregular L&P cut.

Suppose that the disjunction $\vee_{t\in T} D^t x \geq d_0^t$ specializes to the disjunctive normal form of the expression $\{x \in \mathbb{R}^n : x_j \leq \lfloor \bar{x}_j \rfloor \text{ or } x_j \geq \lceil \bar{x}_j \rceil \text{ for } j \in Q\}$, which is

$$\left\{ x \in \mathbb{R}^n : \bigvee_{t\in T} \begin{pmatrix} x_j \leq \lfloor \bar{x}_j \rfloor & j \in Q^- \\ x_j \geq \lceil \bar{x}_j \rceil & j \in Q^+ \end{pmatrix} \right\} \tag{11.16}$$

where $Q^+ \cup Q^- = Q$ and T indexes the set of $2^{|Q|}$ bipartitions of Q. Let $(\overline{11.7})$ and $(\overline{11.8})$ be the expressions (11.7), (11.8) in which $\vee_{t\in T}(D^t x \geq d_0^t)$ takes the form (11.15).

Theorem 11.13 *Let $\bar{x}$ be a basic feasible solution of the LP relaxation with J the index set of nonbasic variables, and $(\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t\in T})$ a basic feasible solution of $(\overline{11.8})$ such that $\bar{\alpha}x \geq \bar{\beta}$ cuts off $\bar{x}$. Suppose there exist a point $\bar{y} \in$* corner$(J) \setminus P$, *and a $t^* \in T$ such that $\bar{v}^{t^*} D^{t^*} \bar{y} = \bar{v}^{t^*} d^{t^*}$ and the set of indices $\{i \mid \bar{u}_i^{t^*} > 0\}$ can be partitioned into two subsets (F^+, F^-) with F^- nonempty and F^+ possibly empty, such that*

- *$\bar{y}$ satisfies all the inequalities $\tilde{A}_i x \geq \tilde{b}_i$ with $i \in F^+$ at equality and*
- *$\bar{y}$ violates all the inequalities $\tilde{A}_i x \geq \tilde{b}_i$ with $i \in F^-$.*

Then $\bar{\alpha}x \geq \bar{\beta}$ is an irregular L&P cut and $\bar{y}$ is a point in corner(J) *which is cut off by $\bar{\alpha}x \geq \bar{\beta}$.*

Proof Since $(\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t\in T})$ is a basic feasible solution of $(\overline{11.8})$, we have $\bar{\alpha} = \sum_i \bar{u}_i^{t^*} \tilde{A}_i + \bar{v}^{t^*} D^{t^*}$, and $\bar{\beta} = \sum_i \bar{u}_i^{t^*} \tilde{b}_i + \bar{v}^{t^*} d^{t^*}$, where $\bar{u}, \bar{v} \geq 0$. Therefore, we deduce that $\bar{\alpha}\bar{y} < \bar{\beta}$, since

$$\begin{aligned} \bar{\alpha}\bar{y} = (\sum_i \bar{u}_i^{t^*} \tilde{A}_i + \bar{v}^{t^*} D^{t^*})\bar{y} &= \sum_{i\in F^+} \bar{u}_i^{t^*} \tilde{A}_i \bar{y} + \sum_{i\in F^-} \bar{u}_i^{t^*} \tilde{A}_i \bar{y} + \bar{v}^{t^*} D^{t^*} \bar{y} \\ &< \sum_{i\in F^+} \bar{u}_i^{t^*} \tilde{b}_i + \sum_{i\in F^-} \bar{u}_i^{t^*} \tilde{b}_i + \bar{v}^{t^*} d^{t^*} = \bar{\beta}, \end{aligned}$$

where the inequality follows from the conditions of the theorem.

Finally, since $\bar{\alpha}x \geq \bar{\beta}$ is not valid for corner(J), and $\bar{v}^{t^*} \neq 0$ since $\bar{\alpha}x \geq \bar{\beta}$ is not valid for P, it is an irregular L&P cut. □

The numerical example below gives a simple illustration of this theorem.

Recall that $\bar{\alpha}x \geq \bar{\beta}$ is the irregular L&P cut associated with the solution to $(\overline{11.8})$ that minimizes $\alpha\bar{x} - \beta$, and that $C(J)$ is the LP cone defined above.

Theorem 11.14 *If there is an extreme ray of the LP cone $C(J)$ with direction vector r such that $\bar{\alpha}r < 0$, then $\bar{\alpha}x \geq \bar{\beta}$ cuts off some point of* corner(J).

Proof Suppose $\bar{x} + r\lambda$, $\lambda \geq 0$ is an extreme ray of $C(J)$ such that $\bar{\alpha}r < 0$. Since $\bar{\alpha}\bar{x} < \bar{\beta}$, it follows that $\bar{\alpha}(\bar{x} + r\lambda) < \bar{\beta}$ for all $\lambda \geq 0$, i.e. the hyperplane $\bar{\alpha}x = \bar{\beta}$ does not intersect the ray $\bar{x} + r\lambda$. This means that $C' := C(J) \cap \{x : \bar{\alpha}x < \bar{\beta}\}$ is unbounded. Clearly, C' contains integer points, and they all belong to corner(J) and are cut off by $\bar{\alpha}x \geq \bar{\beta}$. □

Now suppose no such extreme ray of $C(J)$ exists, i.e. $\bar{\alpha}r^j \geq 0$ for all n extreme rays of $C(J)$. Let $H^+ := \{x \in \mathbb{R}^N : \bar{\alpha}x \geq \bar{\beta}\}$, H^- the weak complement of H^+. Then $\bar{\alpha}x \geq \bar{\beta}$ cuts off a part of corner(J) if and only if $(C(J) \cap H^-) \setminus P$ contains some integer point. In fact, $C(J) \cap H^- \setminus P$ is precisely the part of $C(J) \setminus P$ cut off by $\bar{\alpha}x \geq \bar{\beta}$.

In light of the properties of irregular cuts just discussed, it is important from a practical point of view to assess their frequency. This can be done in several ways. First of all, given any regular basis B of a CGLP, it was shown in [26] that one can always derive numerous irregular bases obtainable from B by only a few pivots. Second, the frequency of bases that satisfy the regularity condition among all possible bases is very low. But of course, these are only some indications that point to the high frequency of irregular cuts; the actual frequency can only be established by numerical experimentation. This raises the question of finding efficient ways of establishing whether a L&P cut is regular or not. Next we address this question.

Given a basic feasible solution $\bar{w} = (\bar{\alpha}, \bar{\beta}, \{\bar{u}^t, \bar{v}^t\}_{t \in T})$ to a CGLP with constraint set (11.8), here is an easy to check necessary and sufficient condition for $\bar{w}$ to be regular.

Theorem 11.15 *Let $\tilde{A}_R$ be the $|R| \times n$ submatrix of $\tilde{A}$ whose rows are indexed by $R := \{j \in Q : \bar{u}^t_j > 0$ for some $t \in T\}$. Then $\bar{w}$ is a regular solution to CGLP if and only if $\tilde{A}_R$ is of full row rank.*

Proof Sufficiency. Assume rank$(\tilde{A}_R) = |R|$. Then $|R| \leq n$. We show that in this situation $\bar{w}$ is regular.

Case 1. $|R| = n$. Then $\tilde{A}_R$ is a $n \times n$ nonsingular submatrix of $\tilde{A}$ such that $u^t_j = 0$ for all $j \notin R$ and all $t \in T$, i.e. w satisfies the condition of Theorem 11.10.

Case 2. $|R| < n$. Then $\tilde{A}$ has $n - |R|$ rows $\tilde{A}_j$ with $u^t_j = 0$ which can be added to $\tilde{A}_R$ in order to form a $n \times n$ nonsingular matrix $\tilde{A}_{R'}$, since $\tilde{A}$ contains I_n. Substituting $A_{R'}$ for A_R then reduces this case to Case 1.

Necessity. Assume rank$(\tilde{A}_R) < |R|$. We show that in this situation w is irregular. Under the assumption, any $n \times n$ nonsingular submatrix $\tilde{A}_J$ of $\tilde{A}$ can have among its rows at most rank$(\tilde{A}_R)$ rows of $\tilde{A}_R$, thus leaving $|R| -$rank$(\tilde{A}_R)$ rows j such that $\bar{u}^t_j > 0$ outside of $\tilde{A}_J$. Therefore no such $\tilde{A}_J$ meets the condition of Theorem 11.10, hence w is irregular. □

Thus deciding whether $\bar{w}$ is regular or not is easy. However, if $\bar{w}$ is irregular, then according to Theorem 11.10 the cut $\bar{\alpha}x \geq \bar{\beta}$ is irregular only if there exists no regular basic feasible solution $\tilde{w}$ of (11.8) with $(\tilde{\alpha}, \tilde{\beta}) = \theta(\bar{\alpha}, \bar{\beta})$ for some $\theta > 0$, a condition whose checking involves the solution of a mixed integer program.

Numerical experimentation shows that the frequency of irregular solutions (bases) is much higher than that of regular ones; but the frequency of irregular *cuts* is considerably lower. In other words, cuts obtained from irregular solutions can often be also obtained from alternative, regular solutions. As a result, the frequency of irregular cuts is highly erratic, with huge differences from one problem class to the other.

11.7 A Numerical Example

Next we give a numerical example that illustrates the occurrence of irregular optimal solutions to the CGLP, resulting in a cut that is violated by the optimal LP solution by more than any SIC from any basis. Furthermore, this irregular cut cuts off part of every corner polyhedron associated with P_I.

Consider the MIP

$$\begin{aligned} &\min y \\ &\text{such that} \\ &y - 1.1x_1 + x_2 \geq -0.15 \\ &y + x_1 - 1.1x_2 \geq -0.2 \\ &y + x_1 + x_2 \geq 0.6 \\ &x_1, x_2 \in \{0, 1\},\ y \geq 0 \end{aligned}$$

The convex body of the feasible solutions is depicted in Fig. 11.4. The optimal solution of the LP relaxation is $x_1^* = 23/105, x_2^* = 8/21, y^* = 0$. The LP in standard form (with surplus variables) is:

$$\begin{aligned} &\min y \\ &\text{such that} \\ &y - 1.1x_1 + x_2 - s_1 = -0.15 \\ &y + x_1 - 1.1x_2 - s_2 = -0.2 \\ &y + x_1 + x_2 - s_3 = 0.6 \\ &x_1, x_2 \in [0, 1],\ s_1, s_2, s_3, y \geq 0 \end{aligned}$$

The simplex tableau for one of the optimal solutions is

s_1	x_1	x_2	y	s_2	s_3	RHS
1			21/10	1	1/10	29/100
	1		1	−10/21	−11/21	23/105
		1	0	10/21	−10/21	8/21

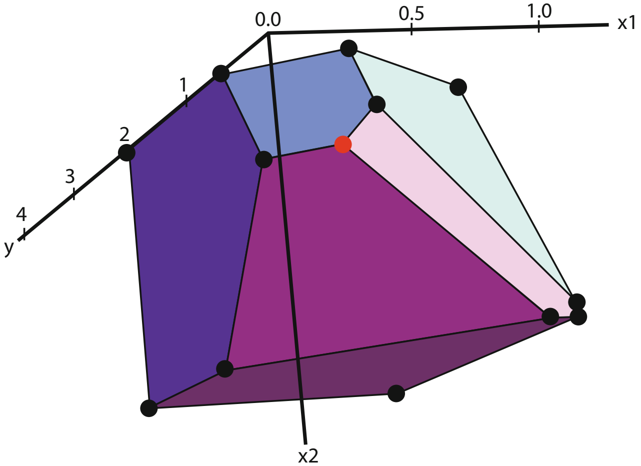


Fig. 11.4 The convex body of feasible points of the LP relaxation. The optimum LP solution is marked by a red dot

We will formulate a CGLP with respect to the 3-term disjunction

$$-x_1 \geq 0 \ \vee \ -x_2 \geq 0 \ \vee \ x_1 + x_2 \geq 2$$

corresponding to the lattice-free polyhedron S in $\mathbb{R}^3$ defined by triangle $\{x \in \mathbb{R}^3 : x_1 \geq 0, x_2 \geq 0, x_1 + x_2 \leq 2\}$. Using this, the CGLP is

$$\min \frac{29}{100}u_1^1 + \frac{23}{105}u_5^1 + \frac{82}{105}u_6^1 + \frac{8}{21}u_7^1 + \frac{13}{21}u_8^1 - \frac{23}{105}v_1$$

$$\begin{pmatrix} \tilde{A}^T \\ \tilde{b}^T \\ \tilde{A}^T \\ \tilde{b}^T \\ e^T \end{pmatrix} u^1 + \begin{pmatrix} -\tilde{A}^T \\ -\tilde{b}^T \\ 0 \\ 0 \\ e^T \end{pmatrix} u^2 + \begin{pmatrix} 0 \\ 0 \\ -\tilde{A}^T \\ -\tilde{b}^T \\ e^T \end{pmatrix} u^3 + \begin{pmatrix} -e_{x_1} \\ 0 \\ -e_{x_1} \\ 0 \\ 1 \end{pmatrix} u_0^1 +$$

$$\begin{pmatrix} e_{x_2} \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_0^2 + \begin{pmatrix} 0 \\ 0 \\ -e_{x_1,x_2} \\ -2 \\ 1 \end{pmatrix} u_0^3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

where $\tilde{A}$ and $\tilde{b}$ are defined by

$$\tilde{A} = \begin{pmatrix} 1 & -1.1 & 1 \\ 1 & 1 & -1.1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix} \qquad \text{and} \qquad \tilde{b} = \begin{pmatrix} -0.15 \\ -0.20 \\ 0.60 \\ 0 \\ 0 \\ -1 \\ 0 \\ -1 \end{pmatrix},$$

and e_{x_i} is a unit vector with 1 in the row corresponding to variable x_i (c.f. $\tilde{A}^T$), $i = 1, 2$, and e_{x_1,x_2} has two 1's, in the respective rows. Notice that u_j^t corresponds to row j of $\tilde{A}$, $j = 1, \ldots, 8$, the first 3 rows represent the three equations of the LP, the fourth row corresponds to $y \geq 0$, the fifth and sixth to $x_1 \geq 0$ and $-x_1 \geq -1$, and the last two to $x_2 \geq 0$ and $-x_2 \geq -1$, respectively.

The optimal basis of CGLP consists of the variables $\{u_0^1, u_0^2, u_0^3, u_2^1, u_3^1, u_1^2, u_3^2, u_2^3, u_4^3\}$, and the basic solution is

v^1	u_2^1	u_3^1	v^2	u_1^2	u_3^2	v^3	u_1^3	u_4^3
0.182156	0.0813197	0.124497	0.170771	0.086741	0.119075	0.0296236	0.00542131	0.200395

As can be seen, the basic variables among the u_j^ℓ correspond to 4 distinct constraints of the LP relaxation, whereas the latter has only three nonbasic variables in any basis. The L&P cut corresponding to the optimal CGLP solution is

$$0.205816y + 0.0236602x_1 + 0.0350449x_2 \geq 0.0584339 \tag{11.17}$$

(see Fig. 11.5).

The violation of this cut is -0.0399009, and one may verify that no intersection cut from any basis of the LP relaxation of MIP has the same violation when represented as a solution of CGLP. Notice that the optimal CGLP basis is irregular, and this inequality cuts into all the corner polyhedra coming from *any* basis of the LP relaxation, which has been verified case by case. For instance, consider the corner polyhedron defined with respect to the optimal simplex tableau:

$$\begin{array}{lllll} x_1 & +y & -10/21s_2 & -11/21s_3 & = 23/105 \\ \quad x_2 & & +10/21s_2 & -10/21s_3 & = 8/21 \\ x_1, x_2 \in \mathbb{Z} & y \geq 0 & s_2 \geq 0 & s_3 \geq 0 & \end{array} \tag{11.18}$$

Proposition 11.16 *The inequality (11.17) is not valid for the corner polyhedron (11.18)*

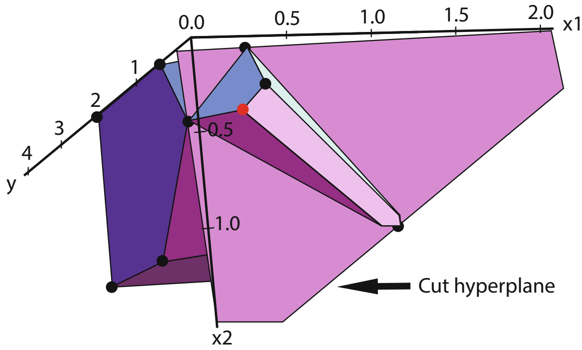


Fig. 11.5 Cut (11.17)

Proof Consider the point $\bar{w} = (\bar{x}_1 = 1, \bar{x}_2 = 0, \bar{y} = 0, \bar{s}_1 = -\frac{95}{100}, \bar{s}_2 = \frac{6}{5}, \bar{s}_3 = \frac{2}{5})$. Since this point has integral x_1 and x_2 coordinates, and satisfies both of the equality constraints defining the corner polyhedron (11.18), and all the nonbasic variables, (y, s_2, s_3), are non-negative, it is in the corner polyhedron.

Now, we verify that the point $\bar{w}$ just defined is cut off by (11.17):

$$0.205816\bar{y} + 0.0236602\bar{x}_1 + 0.0350449\bar{x}_2 = 0.0236602 \not\geq 0.0584339.$$

Finally, notice that $\bar{s}_1 < 0$, that is, $\bar{w} \in \text{corner}(\bar{x}) \setminus P$. $\square$

In fact, inequality (11.17) defines a facet of the convex hull of feasible solutions of the MIP.

11.8 Strengthening Cuts from General Disjunctions

Disjunctive Programming derives cutting planes for mixed integer programs by representing some of the integrality constraints as disjunctions. If the terms of these disjunctions contain variables that are themselves integer constrained, then this circumstance can be used to strengthen the cuts obtained from the disjunction. In case of a split disjunction, this strengthening involves the standard modularization discussed in Sect. 6.5. In the case of cuts derived from more general disjunctions, the situation is more complex. This section discusses strengthening procedures for cuts obtained from general disjunctions.

First we discuss the case of disjunctions consisting of the simultaneous consideration of multiple splits, like those corresponding to intersection cuts from a parametric octahedron, discussed in Sect. 11.1, because for this case there is a strengthening procedure of the nonbasic integer-constrained variables that does not apply to other kinds of disjunctions. Then we will discuss a procedure valid for any kind of disjunctive constraints.

Using the expression

$$x_i = f_i - \sum_{j\in J} r_i^j s_j, \quad s_j \geq 0,\ j \in J,\ s_j \text{ integer},\ j \in J_1 \subseteq J,$$

where $r_i^j = \bar{a}_{ij}$ for $i \in I$, $r_j^j = 1$, and $r_i^j = 0$ for $i \in J \setminus \{j\}$, the disjunction

$$x_i \leq \lfloor x_i \rfloor \vee x_i \geq \lceil x_i \rceil, i = 1, \dots, k \tag{11.19}$$

can be restated in disjunctive normal form as

$$\begin{pmatrix} \sum_{j\in J} r_1^j s_j \geq f_1 \\ \sum_{j\in J} r_2 s_j \geq f_2 \\ \vdots \\ \sum_{j\in J} r_k^j \quad \geq f_k \end{pmatrix} \vee \begin{pmatrix} -\sum_{j\in J} r_1^j s_j \geq 1 - f_1 \\ \sum_{j\in J} r_2^j s_j \quad \geq f_2 \\ \vdots \\ \sum_{j\in J} r_k^j s_j \quad \geq f_k \end{pmatrix} \vee \dots \vee \begin{pmatrix} -\sum_{j\in J} r_1^j s_j \geq 1 - f_1 \\ -\sum_{j\in J} r_2^j s_j \geq 1 - f_2 \\ \vdots \\ -\sum_{j\in J} r_k^j s_j \geq 1 - f_k \end{pmatrix}, \tag{11.20}$$

having $q = 2^k$ terms of k constraints per term.

Lemma 11.17 *Let (11.20+) be the system obtained from (11.20) by replacing some or all r_i^j, $i \in \{1, \dots, k\}$, $j \in J_1$, with $r_i^j - m_i^j$, $m_i^j \in \mathbb{Z}$. Then (11.20+) is satisfied if and only if (11.20) is.*

Proof Replacing r_i^j in (11.20) with $r_j^i - m_i^j$ for some j such that $j \in J_1$ amounts to replacing x_i in (11.19) with $x_i + m_i^j s_j$. This leaves the disjunction $x_i \leq \lfloor x_i \rfloor \vee x_i \geq \lceil x_i \rceil$ valid, since x_i is integer if and only if $x_i - m_i^j s_j$ is. □

Consider now the (CGLP) associated with the general disjunction (11.7) restated here as

$$\left\{ x \in \mathbb{R}^n : \bigvee_{h\in Q} \begin{pmatrix} \tilde{A}x \geq \tilde{b} \\ D^h x \geq d_0^h \end{pmatrix} \right\} \tag{11.21}$$

in which $\vee_{h\in Q}(D^h x \geq d_0^h)$ takes the form (11.19), restated in disjunctive normal form as (11.20). If the (CGLP) solution is $\bar{w} = (\bar{\alpha}, \bar{\beta}, \bar{u}, \bar{v})$, then the resulting cut coefficients are $\bar{\alpha}_j = \max \alpha_j^h$, with $\bar{\alpha}_j^h = \bar{u}^h \tilde{A}^j + \bar{v}^h D^{hj}$ where D^{hj} is the j-th column of D^h. Using the notation (11.20), the j-th column of D^h is the coefficient vector of s_j in the corresponding term of (11.21), whose components are r_i^j or $-r_i^j$, depending only on h. Denoting this vector by r^{jh}, we can rewrite the expression for α_j^h as

$$\bar{\alpha}_j^h = \bar{u}^h \tilde{A}^j + \bar{v}^h r^{jh}. \tag{11.22}$$

If we now replace each component $\pm r_i^j$ of r^{jh} with $\pm(r_i^j - m_i^j)$ for some integer m_i^j, the expression for $\bar{\alpha}_j^h$ becomes

$$\bar{\alpha}_j^h = \bar{u}^h \tilde{A}^j + \bar{v}^h r^{jh} - \bar{v}^h m^{jh} \tag{11.22+}$$

where m^{jh} is the k-vector with components $\pm m_i^{jh}$, with the sign the same as that of r_i^{jh}.

The next question is, how to choose the integers m_i^j in order to maximally strengthen the resulting cut.

Theorem 11.18 *The cut coefficient $\bar{\alpha}_j$ is minimized subject to preserving the validity of the cut by choosing the vector m^j as the optimal solution to the integer program*

$$\begin{aligned} &\min \alpha_j \\ &\quad \alpha_j + \bar{v}^h m^{jh} \geq \bar{v}^h r^{jh} + \kappa_j^h, \ h \in Q \\ &\qquad\qquad m^{jh} \in \mathbb{Z}^k, \qquad\qquad h \in Q \end{aligned} \tag{11.23}$$

where $\kappa_j^h = \bar{u}^h \tilde{A}^j$.

Proof From Lemma 11.17, the disjunction (11.19), and hence the cut $\bar{\alpha}x \geq \bar{\beta}$ that it implies, remains valid for any coefficient obtained from $\bar{\alpha}$ by replacing some or all r_i^j with $r_i^j - m_i^j$ for some or all $m_i^j \in \mathbb{Z}$. The integer program (11.23) chooses for each $\bar{\alpha}_j$ the replacement that minimizes it subject to maintaining its validity. Note that although the integer program (11.23) involves $q = |Q|$ vectors m^{jh}, these k-component vectors differ among themselves only through the signs of their components; in other words, each of the $q = 2^k$ vectors has the same k components in absolute value; they only different among themselves in the sign-distribution of their components (thus the i-th component of each of the q vectors m^{jh} is either m_i^j or $-m_i^j$). Hence the integer program (11.23) has q inequalities but only k variables, all of which appear in every inequality. □

For instance, in case of a disjunction (11.19) involving 3 variables (3 splits), the integer program to be solved for computing the optimal strengthening of each cut coefficient has 3 variables and $2^3 = 8$ inequalities:

$$\begin{aligned}
\min\ & \alpha_j \\
& \alpha_j + \bar{v}_1^1 m_1^j + \bar{v}_2^1 m_2^j + \bar{v}_3^1 m_3^j \geq \bar{v}_1^1 r_1^j + \bar{v}_2^1 r_2^j + \bar{v}_3^1 r_3^j + \kappa_j^1 \\
& \alpha_j + \bar{v}_1^2 m_1^j + \bar{v}_2^2 m_2^j - \bar{v}_3^2 m_3^j \geq \bar{v}_1^2 r_1^j + \bar{v}_2^2 r_2^j - \bar{v}_3^2 r_3^j + \kappa_j^2 \\
& \alpha_j + \bar{v}_1^3 m_1^j - \bar{v}_2^3 m_2^j + \bar{v}_3^3 m_3^j \geq \bar{v}_1^3 r_1^j - \bar{v}_2^3 r_2^j + \bar{v}_3^3 r_3^j + \kappa_j^3 \\
& \qquad \vdots \\
& \alpha_j - \bar{v}_1^8 m_1^j - \bar{v}_2^8 m_2^j - \bar{v}_3^8 m_3^j \geq -\bar{v}_1^8 r_1^j - \bar{v}_2^8 r_2^j - \bar{v}_3^8 r_3^j + \kappa_j^8 \\
& m_i^j \in \mathbb{Z}, \forall i, j.
\end{aligned}$$

The strengthening procedure discussed above is valid for cuts from split disjunctions or multiple split disjunctions, because such disjunctions are equivalent to forcing some integer-constrained variables with fractional values to their nearest integer value. The procedure is not valid for other disjunctions that do not have this property.

Example 2 Consider the instance

$$\begin{aligned}
x_1 &= \tfrac{1}{3} + \tfrac{4}{3}s_1 + \tfrac{13}{2}s_2 - \tfrac{9}{4}s_3 - \tfrac{4}{3}s_4 \\
x_2 &= \tfrac{1}{3} + \tfrac{7}{2}s_1 - \tfrac{7}{3}s_2 - \tfrac{7}{6}s_3 + \tfrac{5}{4}s_4 \\
s_j &\geq 0, j \in J := \{1, \ldots, 4\}; x_1, x_2 \in \mathbb{Z}, s_j \in \mathbb{Z}, j \in J.
\end{aligned}$$

We first derive three unstrengthened intersection cuts from the above two-row linear program (see [36] for details):

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

Next we apply the strengthening procedure:

- For the ray $(r_1^1, r_1^2) = (\frac{4}{3}, \frac{7}{2})$,
  the minimum is attained for $(m_1^1, m_1^2) = (1, 3)$, which yields $\alpha_1 = 0.5519$
- For the ray $(r_2^1, r_2^2) = (\frac{13}{2}, -\frac{7}{3})$,
  the minimum is attained for $(m_2^1, m_2^2) = (6, -3)$, which yields $\alpha_2 = 0.8019$
- For the ray $(r_3^1, r_3^2) = (-\frac{9}{4}, -\frac{7}{6})$,
  the minimum is attained for $(m_3^1, m_3^2) = (-2, -1)$, which yields $\alpha_3 = 0.5$
- For the ray $(r_4^1, r2_4) = (-\frac{4}{3}, \frac{5}{4})$,
  the minimum is attained for $(m_4^1, m_4^2) = (-1, 1)$, which yields $\alpha_4 = 0.549$

So we obtained the strengthened cut 1:

$$0.5519s_1 + 0.819s_2 + 0.5s_3 + 0.549s_4 \geq 1$$

Note that each coefficient of this strengthened cut is strictly smaller than the corresponding coefficient of the original cut.

Applying the same strengthening procedure to the two remaining cuts, we obtain

$$\text{Strengthened cut 2: } 0.75s_1 + \quad 1s_2 + 0.5s_3 + \quad 0.375s_4 \geq 1$$
$$\text{Strengthened cut 3: } \quad 0.5s_1 + 0.75s_2 + 0.5s_3 + 0.5946s_4 \geq 1$$

Next we discuss a monoidal strengthening procedure for general disjunctions in disjunctive normal form, say

$$\left\{ x \in \mathbb{R}^n : \bigvee_{h \in Q} (A^h x \geq a_0^h) \right\},$$

provided that a lower bound b_0^h is known for the lefthand side $A^h x$ of each term. Suppose now that this condition is satisfied; then the constraint set of the problem at hand can be stated as

$$A^h x \geq b_0^h$$
$$x \geq 0 \tag{11.24}$$
$$\bigvee_{h \in Q} (A^h x \geq a_0^h) \tag{11.25}$$

and

$$x_j \text{ integer}, \ j \in J_1 \subseteq J, \tag{11.26}$$

with $a_0^h \geq b_0^h, \quad i \in Q$.

Let $|Q| = q$ and let a_j^h represent the j-th column of A^h, $j \in J, h \in Q$. Then the disjunctive cut from (11.11), (11.12) is $\alpha x \geq \alpha_0$, with

$$\alpha_j = \max_{i \in Q} \theta^h a_j^h, \quad \alpha_0 = \min \theta^h a_0^h \tag{11.27}$$

for some vectors $\theta^h \geq 0, h \in Q$.

Theorem 11.19 ([24]) *Define the monoid*

$$M = \left\{ \mu \in \mathbb{R}^q \,\middle|\, \sum_{h \in Q} \mu_h \geq 0, \mu_h \text{ integer}, h \in Q \right\}. \tag{11.28}$$

Then every $x \in \mathbb{R}^n$ that satisfies (11.11), also satisfies the inequality

$$\sum_{j \in J} \alpha_j x_j \geq \alpha_0, \tag{11.29}$$

where

$$\alpha_j = \begin{cases} \inf\limits_{\mu^j \in M} \max\limits_{h \in Q} \theta^h [a_j^h + \mu_h^j (a_0^h - b_0^h)], & j \in J_1 \\ \max\limits_{h \in Q} \theta^h a_j^h, & j \in J \setminus J_1 = J_2 \end{cases} \tag{11.30}$$

and

$$\alpha_0 = \min_{h \in Q} \theta^h a_0^h. \tag{11.31}$$

To prove this theorem we will use the following auxiliary result

Lemma 11.20 ([10]) *Let $\mu^j \in M$, $\mu^j = (\mu_h^j)$, $j \in J_1$. Then for every $x \in \mathbb{R}^n$ satisfying (11.13) and $x \geq 0$, either*

$$\sum_{j \in J_1} \mu_h^j x_j = 0, \quad \forall h \in Q \tag{11.32}$$

or

$$\bigvee_{h \in Q} \left(\sum_{j \in J_1} \mu_h^j x_j \geq 1 \right) \tag{11.33}$$

holds.

Proof If the statement is false, there exists $\bar{x} \geq 0$ satisfying (11.13) and such that

$$\sum_{h \in Q} \sum_{j \in J_1} \mu_h^j \bar{x}_j < 0,$$

On the other hand, from $\bar{x} \geq 0$ and the definition of M,

$$\sum_{h \in Q} \sum_{j \in J_1} \mu_h^j \bar{x}^j \geq 0,$$

a contradiction. □

Proof of Theorem 11.19. We first show that every x which satisfies (11.11), (11.12) and (11.13), also satisfies

$$\bigvee_{h \in Q} \left[\sum_{j \in J_1} \left[a_j^h + \mu_h^j (a_0^h - b_0^h) \right] x_j + \sum_{j \in J_2} a_j^h x_j \geq a_0^h \right] \tag{11.12$'$}$$

for any set of $\mu^j \in M$, $j \in J_1$. To see this, write (11.12$'$) as

$$\bigvee_{h \in Q} \left[\sum_{j \in J} a_j^h x_j + (a_0^h - b_0^h) \sum_{j \in J_1} \mu_h^j x_j \geq a_0^h \right]. \tag{11.12$''$}$$

From Lemma 11.20, either (11.19) or (11.33) holds for every $x \geq 0$ satisfying (11.13). If (11.19) holds, then (11.12$''$) is the same as (11.12) which holds by assumption. If (11.33) holds, there exists $k \in Q$ such that $\sum_{j \in J_1} \mu_k^j x_j = 1 + \lambda$ for some $\lambda \geq 0$. But then the k-th term of (11.12$''$) becomes

$$\sum_{j \in J} a_j^k x_j \geq b_0^k - \lambda(a_0^k - b_0^k)$$

which is satisfied since $\lambda(a_0^k - b_0^k) \geq 0$ and x satisfies (11.11). This proves that every feasible x satisfies (11.12$'$).

Applying to (11.12$'$) the expression (11.14) then produces the cut (11.16) with coefficients defined by (11.17), (11.18). Taking the infimum over M is justified by the fact that (11.16) is valid with α_0 as in (11.18), α_j as in (11.17) for $j \in J_2$, and

$$\alpha_j = \max_{h \in Q} \theta^h \left[a_j^h + \mu_h^j (a_0^h - b_0^h) \right]$$

for $j \in J_1$, for arbitrary $\mu^j \in M$. □ □

Sometimes it is convenient to use a different form of the cut of Theorem 11.19.

Corollary 11.21 *Let the vectors σ^h, $h \in Q$, satisfy*

$$\sigma^h (a_0^h - b_0^h) = 1, \quad \sigma^h a_0^h > 0. \tag{11.34}$$

Then every $x \in \mathbb{R}^n$ that satisfies (11.11), (11.12) and (11.13), also satisfies

$$\sum_{j \in J} \beta_j x_j \geq 1, \tag{11.16$'$}$$

where

$$\beta_j = \begin{cases} \min\limits_{\mu^j \in M} \max\limits_{h \in Q} \dfrac{\sigma^h a_j^h + \mu_h^j}{\sigma^h a_0^h}, & j \in J_1, \\ \max\limits_{h \in Q} \dfrac{\sigma^h a_j^h}{\sigma^h a_0^h}, & j \in J_2. \end{cases} \tag{11.17$''$}$$

Proof Given any σ^h, $h \in Q$, satisfying (11.34), if we apply Theorem 11.19 by setting $\theta^h = (\sigma^h / \sigma^h a_0^h)$, $h \in Q$, in (11.17) and (11.18), we obtain the cut (11.16′), with β_j defined by (11.17′), $j \in J$. □

Note that the cut-strengthening procedure of Theorem 11.19 requires, in order to be applicable, the existence of lower bounds on each component of $A^h x$, $\forall h \in Q$. This is a genuine restriction, but one that is satisfied in many practical instances. Thus, if x is the vector of nonbasic variables associated with a given basis, assuming that $A^h x$ is bounded below for each $h \in Q$ amounts to assuming that the basic variables are bounded below and/or above. In the case of a 0-1 program, for instance, such bounds not only exist but are quite tight.

Derivation of an optimal set of μ_h requires the solution of a special type of optimization problem. Two efficient algorithms are available [24] for doing this when the multipliers σ^h are fixed. Overall optimization would of course require the simultaneous choice of the σ^h and the μ_h, but a good method for doing that is not yet available.

The following algorithm, which is one of the two procedures given in [24], works with fixed σ^h $h \in Q$. It first finds optimal noninteger values for the μ_h, $h \in Q$, and rounds them down to produce an initial set of integer values. The optimal integer values, and the corresponding value of β_j, are then found by applying an iterative step k times, where $k \leq |Q| - 1$, $|Q|$ being the number of terms in the disjunction from which the cut is derived.

Algorithm for Calculating $\boldsymbol{\beta_j}$, $\boldsymbol{j \in J_1}$, of (11.17)′

Denote

$$\alpha_j = \sigma^h a_j^h, \quad \lambda_h = (\sigma^h a_0^h)^{-1}, \tag{11.35}$$

and

$$\gamma = \sum_{h \in Q} \alpha_h \Bigg/ \sum_{h \in Q} \frac{1}{\lambda_h} \tag{11.36}$$

Calculate

$$\mu_h^* = \frac{\gamma}{\lambda_h} - \alpha_h, \quad h \in Q, \tag{11.37}$$

Set $\mu_h = \lfloor \mu_h^* \rfloor$, $h \in Q$, define $k = -\sum_{h\in Q} \lfloor \mu_h \rfloor$, and apply k times the following

Iterative Step Find

$$\lambda_s(\alpha_s + \mu_s + 1) = \min_{h\in Q}(\alpha_h + \mu_h + 1)$$

and set

$$\mu_s \leftarrow \mu_s + 1, \quad \mu_h \leftarrow \mu_h, \quad h \in Q \setminus \{s\}.$$

This algorithm was shown in [24] to find an optimal set of μ_h (and the associated value of β_h) in k steps, where $k = -\sum_{h\in Q} \lfloor \mu_h^* \rfloor \le |Q| - 1$.

Example 3 Consider the integer program with the constraint set

$$\begin{aligned}
&x_1 = \tfrac{1}{6} + \tfrac{7}{6}(-x_5) - \tfrac{2}{6}(-x_6) + \tfrac{5}{6}(-x_7),\\
&x_2 = \tfrac{2}{6} + \tfrac{1}{6}(-x_5) + \tfrac{1}{6}(-x_6) - \tfrac{1}{6}(-x_7),\\
&x_3 = \tfrac{3}{6} - \tfrac{2}{6}(-x_5) + \tfrac{4}{6}(-x_6) - \tfrac{1}{6}(-x_7),\\
&x_4 = \tfrac{1}{6} + \tfrac{4}{6}(-x_5) + \tfrac{5}{6}(-x_6) - \tfrac{1}{6}(-x_7),\\
&x_1 + x_2 + x_3 + x_4 \ge 1,\\
&x_j = 0 \text{ or } 1,\ j = 1,\ldots,4; \quad x_h \ge 0 \text{ integer},\ j = 5, 6, 7.
\end{aligned}$$

We wish to generate a strengthened cut from the disjunction

$$x_1 \ge 1 \vee x_2 \ge 1 \vee x_3 \ge 1 \vee x_4 \ge 1.$$

The unstrengthened cut from this disjunction is

$$\tfrac{2}{3}x_5 + \tfrac{2}{5}x_6 + \tfrac{1}{3}x_7 \ge 1.$$

To apply the strengthening procedure, we note that each x_j, $j = 1, 2, 3, 4$, is bounded below by 0. Using $\sigma^h = 1$ we obtain

$$\begin{aligned}
\beta_5 &= \min_{\mu\in M} \max\left\{\tfrac{6}{5}(-\tfrac{7}{6} + \mu_1), \tfrac{6}{4}(-\tfrac{1}{6} + \mu_2), \tfrac{6}{3}(\tfrac{2}{6} + \mu_3), \tfrac{6}{5}(-\tfrac{4}{6} + \mu_4)\right\},\\
\beta_6 &= \min_{\mu\in M} \max\left\{\tfrac{6}{5}(\tfrac{2}{6} + \mu_1), \tfrac{6}{4}(-\tfrac{1}{6} + \mu_2), \tfrac{6}{3}(-\tfrac{4}{6} + \mu_3), \tfrac{6}{5}(-\tfrac{5}{6} + \mu_4)\right\},\\
\beta_7 &= \min_{\mu\in M} \max\left\{\tfrac{6}{5}(-\tfrac{5}{6} + \mu_1), \tfrac{6}{4}(\tfrac{1}{6} + \mu_2), \tfrac{6}{3}(\tfrac{1}{6} + \mu_3), \tfrac{6}{5}(\tfrac{1}{6} + \mu_4)\right\}.
\end{aligned}$$

Next we apply the above Algorithm for calculating β_j:

$$\text{For } j = 5 : \gamma = -\tfrac{10}{17}, \mu_1^* = \tfrac{23}{34}, \mu_2^* = -\tfrac{23}{102}, \mu_3^* = -\tfrac{32}{51}, \mu_4^* = \tfrac{11}{51}.$$

Thus our starting values are $\lfloor \mu_1^* \rfloor = 0$, $\lfloor \mu_2^* \rfloor = -1$, $\lfloor \mu_3^* \rfloor = -1$, $\lfloor \mu_4^* \rfloor = 0$.
Since $k = -(-1) - (-1) = 2$, the Iterative step is applied twice:

1. $\min\{-\frac{1}{5}, -\frac{1}{4}, \frac{2}{3}, \frac{2}{5}\} = -\frac{1}{4}, s = 2; \mu_1 = 0, \mu_2 = -1+1 = 0, \mu_3 = -1, \mu_4 = 0.$
2. $\min\{-\frac{1}{5}, \frac{5}{4}, \frac{2}{3}, \frac{2}{5}\} = -\frac{1}{5}, s = 1; \mu_1 = 1, \mu_2 = 0, \mu_3 = -1, \mu_4 = 0.$

These are the optimal μ_h, and

$$\beta_5 = \max\{-\tfrac{1}{5}, -\tfrac{1}{4}, -\tfrac{4}{3}, -\tfrac{4}{5}\} = -\tfrac{1}{5}.$$

For $j = 6$; $\gamma = -\frac{8}{17}$, $\lfloor \mu_1^* \rfloor = -1$, $\lfloor \mu_2^* \rfloor = -1$, $\lfloor \mu_3^* \rfloor = 0$, $\lfloor \mu_4^* \rfloor = 0$; $k = 2$.

1. $\min\{\frac{2}{5}, -\frac{1}{4}, \frac{2}{3}, \frac{1}{5}\} = -\frac{1}{4}, s = 2; \mu_1 = -1, \mu^2 = 0, \mu_3 = 0, \mu_4 = 0.$
2. $\min\{\frac{2}{5}, \frac{5}{4}, \frac{2}{3}, \frac{1}{5}\} = \frac{1}{5}, s = 4; \mu = -1, \mu = 0, \mu = 0, \mu = 0.$

$$\beta_6 = \max\{-\tfrac{4}{5}, -\tfrac{1}{4}, -\tfrac{4}{3}, \tfrac{1}{5}\} = \tfrac{1}{5}.$$

For $j = 7$; $\gamma = -\frac{2}{17}$, $\lfloor \mu_1^* \rfloor = 0$, $\lfloor \mu_2^* \rfloor = -1$, $\lfloor \mu_3^* \rfloor = -1$, $\lfloor \mu_4^* \rfloor = -1$; $k = 3$.

1. $\min\{\frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{5}\} = \frac{1}{5}, s = 1; \mu_1 = 1, \mu_2 = -1, \mu_3 = -1, \mu_4 = -1;$
2. $\min\{\frac{7}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{5}\} = \frac{1}{5}, s = 4; \mu_1 = 1, \mu_2 = -1, \mu_3 = -1, \mu_4 = 0;$
3. $\min\{\frac{7}{5}, \frac{1}{4}, \frac{1}{3}, \frac{7}{5}\} = \frac{1}{4}, s = 2; \mu_1 = 1, \mu_2 = 0, \mu_3 = -1, \mu_4 = 0;$

$$\beta_7 = \max\{\tfrac{1}{5}, \tfrac{1}{4}, -\tfrac{5}{3}, \tfrac{1}{5}\} = \tfrac{1}{4}.$$

Thus the strengthened cut is

$$-\tfrac{1}{5}x_5 + \tfrac{1}{5}x_6 + \tfrac{1}{4}x_7 \geq 1.$$

As can be seen, the strengthening made each coefficient smaller, and made the coefficient of x_5 negative.

The strengthening procedure discussed in this section produces the seemingly paradoxical situation that weakening a disjunction by adding a new term to it, may result in a strengthening of the cut derived from the disjunction; or, conversely, dropping a term from a disjunction may lead to a weakening of the inequality derived from the disjunction. For instance, if the disjunction used in Example 3 is replaced by the stronger one

$$x_1 \geq 1 \vee x_2 \geq 1 \vee x_3 \geq 1,$$

then the cut obtained by the strengthening procedure is

$$-\tfrac{1}{5}x_5 + \tfrac{2}{5}x_6 + \tfrac{1}{4}x_7 \geq 1,$$

which is weaker than the cut of the example, since the coefficient of x_6 is $\frac{2}{5}$ instead of $\frac{1}{5}$. The explanation of this strange phenomenon is to be sought in the fact that the strengthening procedure uses the lower bounds on each term of the disjunction. In Example 3, besides the disjunction $x_1 \geq 1 \vee x_2 \geq 1 \vee x_3 \geq 1 \vee x_4 \geq 1$, the procedure also uses the information that $x_h \geq 0$, $h = 1, 2, 3, 4$. When the above disjunction is strengthened by omitting the term $x_4 \geq 1$, the procedure does not any longer use the information that $x_4 \geq 0$.

11.9 Stronger Cuts from Weaker Disjunctions

Consider the q-term disjunction

$$\bigvee_{i \in Q} \left(\sum_{j \in J} a_{ij} x_j \geq a_{i0} \right), \tag{11.38}$$

where $x_j \geq 0$, $j \in J$, $a_{i0} > 0$, $i \in Q$, $|Q| = q$, and the disjunctive cut $\beta x \geq 1$ from (11.38), where

$$\beta_j = \max_{i \in Q} \left\{ \frac{a_{ij}}{a_{i0}} \right\}. \tag{11.39}$$

As stated earlier, applying monoidal cut strengthening to $\beta x \geq 1$ yields $\bar{\beta} x \geq 1$, with

$$\bar{\beta}_j := \left\{ \begin{array}{ll} \min\limits_{m \in M} \max\limits_{i \in Q} \left\{ \frac{a_{ij} + (a_{i0} - b_i) m^i_j}{a_{i0}} \right\} & j \in J_1 \\ \beta_j & j \in J \setminus J_1, \end{array} \right\} \tag{11.40}$$

where b_i is a lower bound on $\sum_{j \in J} a_{ij} x_j$, M is the monoid $M := \{m \in \mathbb{Z}^q : \sum_{i \in Q} m^i \geq 0\}$, and m^i_j is the i-th component of $m_j \in M$. The validity of $\bar{\beta} x \geq 1$ follows from the following

Proposition 11.22 *Any x satisfying $x_j \geq 0$, $j \in J$, $x_j \in \mathbb{Z}$, $j \in J_1$, and (11.38) such that $\sum_{j \in J} a_{ij} x_j \geq b_i$, $i \in Q$, also satisfies the disjunction*

$$\bigvee_{i \in Q} \left(\sum_{j \in J_1} (a_{ij} + (a_{i0} - b_i) m^i_j) x_j + \sum_{j \in J \setminus J_1} a_{ij} x_j \geq a_{i0} \right). \tag{11.41}$$

(See [24] for a proof.)

A glance at expression (11.40) suggests that the role of the integer m^i_j, $i \in Q$, 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 $\sum_{i \in Q} m^i_j \geq 0$. 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 section we characterize the family of cuts obtainable through this technique. We then apply the results to the special case of simple split disjunctions, and to a class of intersection cuts from two rows of the simplex tableau. 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. Finally, we briefly discuss the case of cuts obtained by strictly weakening a disjunction.

The next Theorem introduces a new class of valid cuts derived from disjunctions equivalent to (11.38) but with additional redundant terms.

Theorem 11.23 *Let Q and a_{ij} be as in (11.38), and let $M := \{m \in \mathbb{Z}^q : \sum_{i \in Q} m^i \geq 0\}$. For each $k \in Q$, the cut $\bar{\beta}^k x \geq 1$, with*

$$\bar{\beta}^k_j := \begin{cases} \min\left\{ \dfrac{a_{kj} + a_{k0} - b_k}{a_{k0}}, \min\limits_{\substack{m_j \in M \\ m^k_j \geq 0}} \max\limits_{i \in Q} \left\{ \dfrac{a_{ij} + (a_{i0} - b_i) m^i_j}{a_{i0}} \right\} \right\} & j \in J_1 \\ \beta_j & j \in J \setminus J_1 \end{cases} \tag{11.42}$$

Proof Consider the disjunction

$$\bigvee_{i \in Q} \left(\sum_{j \in J} a_{ij} x_j \geq a_{i0} \right) \vee \underbrace{\left(\sum_{j \in J} a_{kj} x_j \geq a_{k0} \right) \vee \cdots \vee \left(\sum_{j \in J} a_{kj} x_j \geq a_{k0} \right)}_{r \text{ terms}}. \tag{11.43}$$

It contains $(q + r)$ terms of which $(r + 1)$ are copies of the k-th term of (11.38). 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 (11.43) and Proposition 11.22 a cut $\gamma^k x \geq 1$ with coefficients

$$\gamma_j^k(m_j) := \begin{cases} \max \left\{ \begin{array}{l} \max\limits_{i \in Q} \left\{ \dfrac{a_{ij} + (a_{i0} - b_i)m_j^i}{a_{i0}} \right\}, \\ \qquad \dfrac{a_{kj} + (a_{k0} - b_k)m_j^{q+1}}{a_{k0}}, \\ \quad \ldots, \dfrac{a_{kj} + (a_{k0} - b_k)m_j^{q+r}}{a_{k0}} \end{array} \right\} & j \in J_1 \\ \beta_j & j \in J \setminus J_1 \end{cases} \tag{11.44}$$

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 $\gamma_j^k = \beta_j^k = \beta_j$ 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 $\gamma_j^k(m_j) = \tilde{\beta}_j^k$. We consider two elements of the monoid M', each of which defines valid cut coefficients. Let

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

If we replace the values m_j with m_{1j} in (11.44), all the ratios corresponding to the terms different from k are reduced to a value less than or equal to $\frac{a_{kj} + a_{k0} - b_k}{a_{k0}}$; therefore,

$$\gamma_j^k(m_{1j}) = \frac{a_{kj} + a_{k0} - b_k}{a_{k0}}.$$

The monoidal condition $\sum_{i=1}^{q+r} m_{1j}^i = \sum_{i=1, i \neq k}^{q} t_j^i + r + 1 \geq 0$ is satisfied if we choose r to be $r = \max_{j \in J_1} \sum_{i \in Q \setminus \{k\}} (-t_j^i) - 1$.

Now, let m_{2j} be the vector in M' that minimizes the expression $\gamma_j^k(m_j)$ for $j \in J_1$ in (11.44) according to the general formula (11.40); i.e.,

$$\gamma_j^k(m_{2j}) = \min_{m_j \in M'} \max \left\{ \frac{a_{ij} + (a_{i0} - b_i)m_j^i}{a_{i0}} : i \in \{1, \ldots, q+r\} \right\}.$$

Since both m_{1j} and m_{2j} yield valid cut coefficients for (11.44), we can choose $\gamma_j^k(m_j) = \min\{\gamma_j^k(m_{1j}), \gamma_j^k(m_{2j})\}$ to obtain a valid cut $\gamma x \geq 1$ for (11.44). Such a cut, however, may not be valid for the original disjunction (11.38), since any choice of $m_j \in M'$ in which $m_j^k < 0$, and which is optimal for (11.38), may not be valid for (11.44), where $m_j \in M'$ requires $\sum_{i \in Q \setminus \{k\}} m_j^i \geq r(-m_j^k)$. On the other hand, restricting the choices to $m_j^k \geq 0$ makes optimal choices for (11.38) also valid for (11.44). Hence the definition of β_j^k in the theorem. □

Theorem 11.23 can also be proved using Remark 3.1 of [36] that states that Proposition 11.22 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 (11.42) can be obtained by applying the general monoidal cut strengthening on (11.38) using the monoid $M(\bar{\mu})$ where $\bar{\mu}$ is

$$\bar{\mu} := \begin{cases} 1 & i \in Q \setminus \{k\} \\ r+1 & i = k. \end{cases} \tag{11.45}$$

and r is defined as before.

We call $\tilde{\beta}^k x \geq 1$ the k-th *Lopsided cut* associated with the disjunction (11.38). The upshot of Theorem 11.23 is that given a q-term disjunction (11.38) 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 (11.42) we need to determine

$$\min_{\substack{m_j \in M \\ m_j^k \geq 0}} \max_{i \in Q} \left\{ \frac{a_{ij} + (a_{i0} - b_i) m_j^i}{a_{i0}}. \right\} \tag{11.46}$$

We can solve (11.46) using Algorithm 1 of [36] with the expression $\lambda_r(\alpha_r + m_r) = \max_{i\in Q} \lambda_i(\alpha_i + m_i)$ replaced by

$$\lambda_r(\alpha_r + m^r) = \begin{cases} \max\limits_{i\in Q} \lambda_i(\alpha_i + m^i) \text{ if } m^k \geq 1 \\ \max\limits_{\substack{i\in Q \\ i \neq k}} \lambda_i(\alpha_i + m^i) \text{ otherwise} \end{cases} \tag{11.47}$$

The expression (11.47) 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 is linear in $\sum_{i\in Q} |m^i|$, i.e., it is quite inexpensive.

Corollary 11.24 *If $\tilde{\beta}_j^k < \bar{\beta}_j$ then $\tilde{\beta}_j^k = \frac{a_{kj} + a_{k0} - b_k}{a_{k0}}$ for $j \in J_1$.*

Proof Notice that

$$\tilde{\beta}_j^k < \bar{\beta}_j = \min_{m_j \in M} \max_{i\in Q} \left\{ \frac{a_{ij} + (a_{i0} - b_i) m_j^i}{a_{i0}} \right\} \leq \min_{\substack{m_j \in M \\ m_j^k \geq 0}} \max_{i\in Q} \left\{ \frac{a_{ij} + (a_{i0} - b_i) m_j^i}{a_{i0}} \right\}. \tag{11.48}$$

From (11.42) and (11.48) we have $\tilde{\beta}_j^k = \frac{a_{ij}+a_{k0}-b_k}{a_{k0}}$. □

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

Corollary 11.25 *For each $k \in Q$, the cut $\delta^k x \geq 1$, with*

$$\delta_j^k := \begin{cases} \min\left\{\frac{a_{kj}+a_{k0}-b_k}{a_{k0}}, \max_{i\in Q}\left\{\frac{a_{ij}}{a_{i0}}\right\}\right\} & j \in J_1 \\ \beta_j & j \in J \setminus J_1 \end{cases} \tag{11.49}$$

is valid.

11.9.1 Simple Split Disjunction

Now consider a Mixed Integer Program, and let

$$\begin{aligned} y = a_0 - \sum_{j\in J} a_j x_j \\ x_j \geq 0,\, j \in J \\ x_j \in \mathbb{Z},\, j \in J_1 \subseteq J \end{aligned} \tag{11.50}$$

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 (11.50) can be derived as a disjunctive cut from $(y \leq 0) \vee (y \geq 1)$, or

$$\left(\sum_{j\in J} a_j x_j \geq a_0\right) \vee \left(\sum_{j\in J} (-a_j) x_j \geq 1 - a_0\right) \tag{11.51}$$

as $\alpha x \geq 1$, with

$$\alpha_j := \max\left\{\frac{a_j}{a_0}, \frac{-a_j}{1-a_0}\right\},\, j \in J, \tag{11.52}$$

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

$$\bar{\alpha}_j := \begin{cases} \min\left\{\frac{a_j - \lfloor a_j \rfloor}{a_0}, \frac{-a_j + \lceil a_j \rceil}{1-a_0}\right\} & j \in J_1 \\ \alpha_j & j \in J \setminus J_1. \end{cases} \tag{11.53}$$

As (11.51) and (11.52) are a special case of (11.38) and (11.39), the coefficients $\bar{\alpha}_j$ of (11.53) 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 (11.53) becomes (11.54)

$$\bar{\beta}_j := \begin{cases} \min\limits_{m_j^1, m_j^2 \in M} \max\left\{ \dfrac{a_j + m_j^1}{a_0}, \dfrac{-a_j + m_j^2}{1 - a_0} \right\} & j \in J_1 \\ \alpha_j & j \in J \setminus J_1 \end{cases} \tag{11.54}$$

which is a special case of (11.40). 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 11.26 *$\alpha^+ x \geq 1$ and $\alpha^- x \geq 1$ are valid cuts for (11.50), with*

$$\alpha_j^+ := \begin{cases} \dfrac{-a_j + 1}{1 - a_0} & j \in J_1^+ := \{j \in J_1 : a_j > 1\} \\ \min\left\{ \dfrac{a_j - \lfloor a_j \rfloor}{a_0}, \dfrac{-a_j + \lceil a_j \rceil}{1 - a_0} \right\} & j \in J_1^> := \{j \in J_1 : a_0 - 1 \leq a_j \leq 1\} \\ \max\left\{ \dfrac{a_j}{a_0}, \dfrac{-a_j}{1 - a_0} \right\} & j \in (J \setminus J_1) \cup \{j \in J_1 : a_j < a_0 - 1\} \end{cases} \tag{11.55}$$

and

$$\alpha_j^- := \begin{cases} \dfrac{a_j + 1}{a_0} & j \in J_1^- := \{j \in J_1 : a_j < -1\} \\ \min\left\{ \dfrac{a_j - \lfloor a_j \rfloor}{a_0}, \dfrac{-a_j + \lceil a_j \rceil}{1 - a_0} \right\} & j \in J_1^< := \{j \in J_1 : -1 \leq a_j \leq a_0\} \\ \max\left\{ \dfrac{a_j}{a_0}, \dfrac{-a_j}{1 - a_0} \right\} & j \in (J \setminus J_1) \cup \{j \in J_1 : a_j > a_0\} \end{cases} \tag{11.56}$$

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 11.23 to (11.51) 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

$$A = \frac{-a_j + 1}{1 - a_0}$$

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

- If $a_j > 1$ then $A < 0$, $B \geq 0$ and $\tilde{\beta}_j^2 = A = \alpha_j^+$.

- If $a_0 \le a_j \le 1$ then $\tilde{\beta}_j^2 = A = B = \frac{-a_j + \lceil a_j \rceil}{1-a_0} = \alpha_j^+$.
- If $0 < a_j < a_0$ then $A > 1$, $B \le 1$ 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 11.27 *If $a_j \ge a_0 - 1$, $j \in J_1$ and $J_1^+ \ne \emptyset$, the cut $\alpha^+ x \ge 1$ strictly dominates the GMI cut. if $a_j \le a_0$, $j \in J_1$ and $J_1^- \ne \emptyset$, the cut $\alpha^- x \ge 1$ strictly dominates the GMI cut.*

Example 4 Consider the following row of a simplex tableau

$$y = 0.2 - 1.5x_1 + 0.3x_2 + 0.4x_3 + 0.6x_4 - 4.3x_5 - 0.1x_6 \tag{11.57}$$

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

$$0.625x_1 + 0.375x_2 + 0.5x_3 + 0.75x_4 + 0.875x_5 + 0.5x_6 \ge 1. \tag{11.58}$$

Applying Theorem 11.26 we obtain a cut $\alpha^+ x \ge 1$ that strictly dominates (11.58). For (11.57) 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 (11.58). The cut $\alpha^+ x \ge 1$ is then

$$-0.625x_1 + 0.375x_2 + 0.5x_3 + 0.75x_4 - 4.125x_5 + 0.5x_6 \ge 1. \tag{11.59}$$

Note that (11.59) is tight for the solution $(\bar{y}, \bar{x}_1, \dots, \bar{x}_6)$ while the GMI cut (11.58) has a slack of 1.25.

In Fig. 11.6 we illustrate graphically the value of the cut coefficients for the GMI cut and the two lopsided cuts given in (11.53), (11.55) and (11.56) 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 [24, 57], the same holds also for the two cuts in Theorem 11.26. Indeed, it can be shown that the cut $\alpha^+ x \ge 1$ can be obtained by dividing the source row in (11.50) by a large number and then deriving a GMI cut, and a similar procedure yields the cut $\alpha^- x \ge 1$. However, in the case of a more general disjunction we do not know of any alternative method for deriving the cut of Theorem 11.23 or Corollary 11.25.

In Table 11.1 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

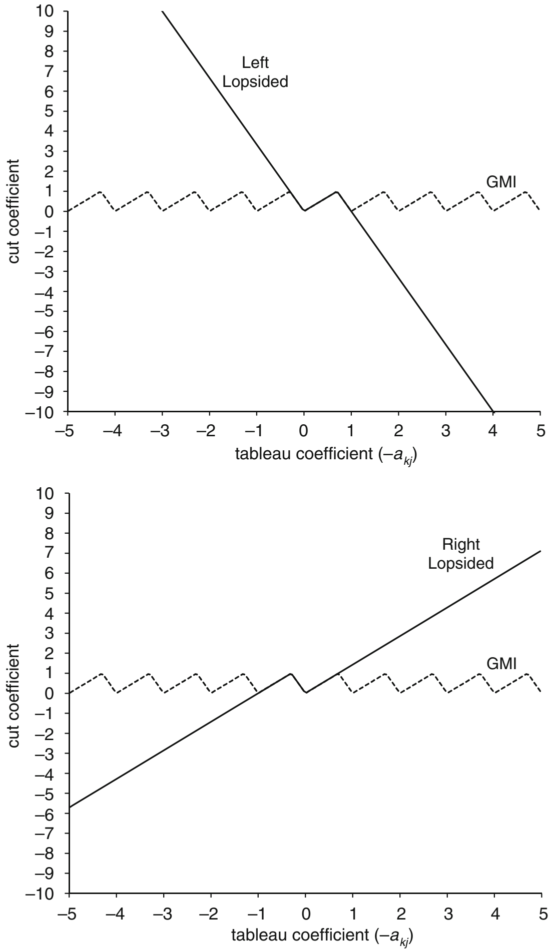


Fig. 11.6 Coefficient values for the GMI and the lopsided cuts for the case $a_0 = 0.3$

relaxations we consider the duality gap closed which is computed as

$$\text{Gap} = 100\frac{C_{\text{opt}} - LP_{\text{opt}}}{IP_{\text{opt}} - LP_{\text{opt}}} \tag{11.60}$$

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 (11.60). The column imp shows the difference between $G_{\%}$ and $G + L_{\%}$, and the column imp$_{\%}$ indicates the percentage improvement produced by adding the lopsided cuts on top of the GMI relaxation. Table 11.1 shows only those instances for which the percentage improvement given by the lopsided cuts exceeds 1%.

Table 11.1 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	15.93	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
max76	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

Although in many problems the lopsided cuts produce no improvement, in some cases the impact is substantial.

11.9.2 Multiple Term Disjunctions

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

$$\begin{aligned}
y_1 &= a_{10} - \sum_{j \in J} a_{1j} x_j \\
y_2 &= a_{20} - \sum_{j \in J} a_{2j} x_j \\
x_j &\geq 0,\ j \in J \\
\\
x_j &\in \mathbb{Z},\ j \in J_1 \subseteq J
\end{aligned} \tag{11.61}$$

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 [10, 70] $\alpha^{ic} x \geq 1$ with coefficients

$$\alpha_j^{ic} := \min_{\mu > 0} \left\{ \frac{1}{\mu} : (a_{10}, a_{20}) + \mu(-a_{1j}, -a_{2j}) \in S \right\} \tag{11.62}$$

fpr $j \in J$. When $J_1 \neq \emptyset$ the standard strengthening used, for instance, in [70], yields a stronger cut $\alpha^{str}x \geq 1$ where

$$\alpha_j^{str} := \min_{\substack{\mu>0 \\ p_{1j}, p_{2j} \in \mathbb{Z}}} \left\{ \frac{1}{\mu} : (a_{10}, a_{20}) + \mu(-a_{1j} + p_{1j}, -a_{2j} + p_{2j}) \in S \right\}. \tag{11.63}$$

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

$$S = \{(y_1, y_2) \in \mathbb{R}^2 : d_{1i}y_1 + d_{2i}y_2 \leq e_i, i \in Q\}, \tag{11.64}$$

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

$$\bigvee_{i \in Q} \left(\sum_{j \in J} (-d_{1i}a_{1j} - d_{2i}a_{2j})x_j \geq e_i - d_{1i}a_{20} \right). \tag{11.65}$$

Furthermore, if lower bounds are known for the left hand sides of the terms of (11.65), then monoidal strengthening [36] can be used to obtain another strengthened cut $\bar{\beta}x \geq 1$, generally different from (11.63).

We now consider the lopsided cuts derived from Theorem 11.23 applied to the disjunction (11.65). 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.

$$S = \{(y_1, y_2) \in \mathbb{R}^2 : y_1 \geq 0; y_2 \geq 0; y_1 + y_2 \leq 2\}. \tag{11.66}$$

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

$$\left(\sum_{j \in J} a_{1j}x_j \geq a_{10} \right) \vee \left(\sum_{j \in J} a_{2j}x_j \geq a_{20} \right) \vee \left(\sum_{j \in J} (-a_{1j} - a_{2j})x_j \geq 2 - a_{10} - a_{20} \right). \tag{11.67}$$

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

$$\alpha_j^{ic} = \max \left\{ \frac{a_{1j}}{a_{10}}, \frac{a_{2j}}{a_{20}}, \frac{-a_{1j} - a_{2j}}{2 - a_{10} - a_{20}} \right\}, j \in J \tag{11.68}$$

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

$$\alpha_j^{str} := \left\{ \begin{array}{ll} \min\limits_{p_{1j}, p_{2j} \in \mathbb{Z}} \max \left\{ \dfrac{a_{1j} + p_{1j}}{a_{10}}, \dfrac{a_{2j} + p_{2j}}{a_{20}}, \dfrac{-a_{1j} - a_{2j} - p_{1j} - p_{2j}}{2 - a_{10} - a_{20}} \right\} & jinJ_1 \\ \alpha_j^{ic} & j \in J \setminus J_1 \end{array} \right\} \tag{11.69}$$

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 (11.67). Therefore the monoidal strengthened cut $\bar{\beta}_j x \geq 1$ has coefficients

$$\bar{\beta}_j := \begin{cases} \min\limits_{m_j \in M} \max \left\{ \dfrac{a_{1j} + m_{1j}}{a_{10}}, \dfrac{a_{2j} + m_{2j}}{a_{20}}, \dfrac{-a_{1j} - a_{2j} - 2m_{3j}}{2 - a_{10} - a_{20}} \right\} & j \in J_1 \\ \alpha_j^{ic} & j \in J \setminus J_1 \end{cases} \tag{11.70}$$

where $M = \{m \in \mathbb{Z}^3 : \sum_{i=1}^{3} m^i \geq 0\}$. Theorem 11.23 yields the cuts $tbeta^k x \geq 1$, $k \in \{1, 2, 3\}$ where

$$\tilde{\beta}_j^k := \begin{cases} \min \left\{ \frac{a_{1j}+1}{a_{10}}, \min\limits_{\substack{m_j \in M \\ m_1 \geq 0}} \max \left\{ \frac{a_{1j}+m_j^1}{a_{10}}, \frac{a_{2j}+m_j^2}{a_{20}}, \frac{-a_{1j}-a_{2j}+2m_j^3}{2-a_{10}-a_{20}} \right\} \right\} & k = 1, j \in J_1 \\ \min \left\{ \frac{a_{2j}+1}{a_{20}}, \min\limits_{\substack{m_j \in M \\ m_2 \geq 0}} \max \left\{ \frac{a_{1j}+m_j^1}{a_{10}}, \frac{a_{2j}+m_j^2}{a_{20}}, \frac{-a_{1j}-a_{2j}+2m_j^3}{2-a_{10}-a_{20}} \right\} \right\} & k = 2, j \in J_1 \\ \min \left\{ \frac{-a_{1j}-a_{2j}+1}{2-a_{10}-a_{20}}, \min\limits_{\substack{m_j \in M \\ m_3 \geq 0}} \max \left\{ \frac{a_{1j}+m_j^1}{a_{10}}, \frac{a_{2j}+m_j^2}{a_{20}}, \frac{-a_{1j}-a_{2j}+2m_j^3}{2-a_{10}-a_{20}} \right\} \right\} & k = 3, j \in J_1 \\ \alpha_j^{ic} & j \in J \setminus J_1 \end{cases} \tag{11.71}$$

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

$$\left\{ \begin{array}{l} a_{1j} + m_1 \leq 0 \\ a_{2j} + m_2 \leq 0 \\ -a_{1j} - a_{2j} + 2m_3 \leq 0 \\ m_1 + m_2 + m_3 \geq 0 \\ a_{1j} + a_{2j} \geq 2 \\ m_i \in \mathbb{Z}, i \in \{1, 2, 3\} \end{array} \right\} \tag{11.72}$$

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

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ -1 & -1 & 0 & 0 & 2 \\ 0 & 0 & -1 & -1 & -1 \\ -1 & -1 & 0 & 0 & 0 \end{pmatrix} b = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -2 \end{pmatrix} \tag{11.73}$$

is a relaxation of (11.72). Let $c = (2, 2, 1, 2, 1)$. Since $cA = 0$, $c \geq 0$, $cb = -2 < 0$ by Farkas Lemma (11.73) is infeasible. Thus (11.72) is also infeasible and we reached a contradiction, therefore $\bar{\beta}_j > 0$. □

Proposition 11.29 *If* $(2 + a_{20})a_{1j} - a_{10}a_{2j} < -2 - a_{10} - a_{20}$ *then* $\tilde{\beta}_j^1 < \bar{\beta}_j$ *for* $j \in J_1$.

Proposition 11.30 *If* $-a_{20}a_{1j} + (2 + a_{10})a_{2j} < -2 - a_{10} - a_{20}$ *then* $\tilde{\beta}_j^2 < \bar{\beta}_j$ *for* $j \in J_1$.

The proofs of Propositions 11.29 and 11.30 are analogous to the proof of Proposition 11.28 but a bit more involved.

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

Proposition 11.31 *If* $(k = 1 \wedge a_{1j} < -1)$ *or* $(k = 2 \wedge a_{2j} < -1)$ *or* $k = 3 \wedge -a_{1j} < -2)$ *then* $\tilde{\beta}_j^k < 0 \leq \alpha_j^{str}$.

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 5 Consider the 2-row relaxation

$$\begin{aligned} P = \{(y, x) \in \mathbb{R}^8 : \; & y_1 = 0.25 - 0.15x_1 + 0.6x_2 - 0.4x_3 - 1.2x_4 - 2.9x_5 + 0.8x_6 \\ & y_2 = 0.5 + 1.15x_1 - 0.1x_2 - 0.2x_3 - 1.6x_4 + 0.5x_5 - 2.5x_6 \\ & x_j \geq 0,\ j \in J = \{1, \ldots, 6\} \\ & x_j \in \mathbb{Z},\ j \in J_1 = \{4, 5, 6\} \\ & y_k \in \{0, 1\}, k \in \{1, 2\} \;\}. \end{aligned} \tag{11.74}$$

The current basic solution to the relaxation (11.74) 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 (11.66) since $(\bar{y}_1, \bar{y}_2)$ is in the interior of S. The coefficients α_j^{ic} are

$$\alpha_j^{ic} = \min_{\mu>0}\left\{\frac{1}{\mu} : a_{10} + \mu(-a_{1j}) \geq 0;\, a_{20} + \mu(-a_{2j}) \geq 0;\, a_{10} + a_{20} + \mu(a_{1j} + a_{2j}) \leq 2\right\} \tag{11.75}$$

Applying (11.75) to the instance (11.74) we get the intersection cut

$$0.8x_1 + 0.4x_2 + 1.5x_3 + 4.8x_4 + 11.6x_5 + 5x_6 \geq 1. \tag{11.76}$$

By standard strengthening we can get the cut

$$0.8x_1 + 0.4x_2 + 1.6x_3 + 0.8x_4 + 0.48x_5 + 0.8x_6 \geq 1. \tag{11.77}$$

Monoidal strengthening applied to (11.67) yields

$$0.8x_1 + 0.4x_2 + 1.6x_3 + 1.2x_4 + 1.28x_5 + x_6 \geq 1, \tag{11.78}$$

which in this case is weaker than (11.77).

If we apply Theorem 11.23 to the disjunction (11.67) with $k = 3$, we obtain the lopsided cut $\tilde{\beta}^3 x \geq 1$ where

$$\tilde{\beta}_j^3 := \begin{cases} \min\left\{\frac{-a_{1j}-a_{2j}+2}{2-a_{10}-a_{20}},\ \min\limits_{\substack{m_j \in M \\ m_j^3 \geq 0}} \max\left\{\frac{a_{1j}+m_j^2}{a_{10}}, \frac{a_{2j}+m_j^2}{a_{20}}, \frac{-a_{1j}-a_{2j}+2m_j^3}{2-a_{10}-a_{20}}\right\}\right\} & j \in J_1 \\ \alpha_j^{ic} & j \in J \setminus J_1 \end{cases} \tag{11.79}$$

Computing these coefficients for the instance (11.74) we get

$$\tilde{\beta}_4^3 = \min\left\{\frac{-1.2-1.6+2\times 1}{2-0.25-0.5}, \max\left\{\frac{1.2-1}{0.25}, \frac{1.6-1}{0.5}, \frac{-1.2-1.6+2\times 2}{2-0.25-0.5}\right\}\right\} = -0.64$$

$$\tilde{\beta}_5^3 = \min\left\{\frac{-2.9+0.5+2\times 1}{2-0.25-0.5}, \max\left\{\frac{2.9-3}{0.25}, \frac{-0.5+1}{0.5}, \frac{-2.9+0.5+2\times 2}{2-0.25-0.5}\right\}\right\} = -0.32$$

$$\tilde{\beta}_6^3 = \min\left\{\frac{0.8-2.5+2\times 1}{2-0.25-0.5}, \max\left\{\frac{-0.8+1}{0.25}, \frac{2.5-2}{0.5}, \frac{0.8-2.5+2\times 1}{2-0.25-0.5}\right\}\right\} = 0.24.$$

Therefore the cut $\tilde{\beta}^3 x \geq 1$ is

$$0.8x_1 + 0.4x_2 + 1.6x_3 - 0.64x_4 - 0.32x_5 + 0.24x_6 \geq 1. \tag{11.80}$$

The lopsided cut (11.80) strictly dominates both the standard strengthened cut (11.77) and the monoidal strengthened cut (11.78).

11.9.3 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 6 Consider the 2-row relaxation

$$
\begin{aligned}
P = \{(y, x) \in \mathbb{R}^9 :\ & y_1 = 0.25 + 0.58x_1 + 0.1x_2 - 0.48x_3 + 0.5x_4 - 1.5x_5 + 2.5x_6 - 7.2x_7 \\
& y_2 = 0.5 + 0.29x_1 - 0.6x_2 + 0.32x_3 + 5.8x_4 - 0.1x_5 + 3.6x_6 + 2.6x_7 \\
& x_j \geq 0,\ j \in J = \{1, \dots, 7\} \\
& x_j \in \mathbb{Z},\ j \in J_1 = \{4, \dots, 7\} \\
& y_k \in \{0, 1\}, k \in \{1, 2\} \ \ \}.
\end{aligned}
\tag{11.81}
$$

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

$$S = \{(y_1, y_2) \in \mathbb{R}^2 : 2y_1 + y_2 \geq 0; 4y_1 - y_2 \leq 4; y_2 \leq 1\}.$$

The set S is a triangle with vertices $q_1 = \left(-\frac{1}{2}, 1\right)$; $q_2 = \left(\frac{5}{4}, 1\right)$; $q_3 = \left(\frac{2}{3}, -\frac{4}{3}\right)$. The intersection cut $\alpha^{ic}x \geq 1$ is

$$0.58x_1 + 0.4x_2 + 0.64x_3 + 11.6x_4 + 3.1x_5 + 7.2x_6 + 11.8x_7 \geq 1. \tag{11.82}$$

The standard strengthened cut $\alpha^{str}x \geq 1$ is

$$0.58x_1 + 0.4x_2 + 0.64x_3 + 0.6286x_4 + 0.6x_5 + 0.6857x_6 + 0.8x_7 \geq 1. \tag{11.83}$$

The cut (11.82) can be obtained from the disjunction

$$(2y_1 + y_2 \leq 0) \vee (4y_1 - y_2 \geq 4) \vee (y_2 \geq 1). \tag{11.84}$$

Applying monoidal strengthening to (11.84) we get the cut $\bar{\beta}x \geq 1$:

$$0.58x_1 + 0.4x_2 + 0.64x_3 + 2.2x_4 + 0.1x_5 + 1.8286x_6 + 1.0286x_7 \geq 1. \tag{11.85}$$

Consider now the disjunction

$$(-2y_1 - y_2 \geq 0) \vee (4y_1 - y_2 \geq 4) \vee (y_2 \geq 1) \vee (-y_2 \geq 1) \tag{11.86}$$

obtained from (11.84) by adding the term $(y_2 \leq -1)$. The disjunction (11.86) is strictly weaker than (11.84) since $(y_1, y_2) = (0.7, -1)$ does not satisfy (11.84) but satisfies (11.86).

Applying monoidal cut strengthening to (11.86) we derive the cut

$$0.58x_1 + 0.4x_2 + 0.64x_3 + 0.3429x_4 + 0.1x_5 + 0.4x_6 - 0.2x_7 \geq 1. \tag{11.87}$$

Consider now the disjunction

$$(-2y_1 - y_2 \geq 0) \vee (4y_1 - y_2 \geq 4) \vee (y_2 \geq 1) \vee (-y_2 \geq 1) \tag{11.88}$$

obtained from (11.86) by adding the term $(y_2 \leq -1)$. Disjunction (11.88) is strictly weaker than (11.86), since $(y_1, y_2) = (0.7, -1)$ does not satisfy (11.86) but satisfies (11.88).

Applying monoidal cut strengthening to (11.88), we derive the cut

$$0.58x_1 + 0.4x_2 + 0.64x_3 + 0.3429x_4 + 0.1x_5 + 0.4x_6 - 0.2x_7 \geq 1, \tag{11.89}$$

which strictly dominates both cuts (11.85) and (11.87). Moreover, it can be shown that (11.89) is not dominated by any combination of cuts (11.85), (11.87), the two GMI cuts derived from (11.83) and the three lopsided cuts derived from disjunction (11.86).

Chapter 12
Disjunctive Cuts from the V-Polyhedral Representation

Given a disjunctive set in disjunctive normal form, i.e. as a union of polyhedra $F := \cup_{h \in Q} P^h$, in Chap. 2 we gave a compact representation of conv$F$ essentially based on the fact that

- an inequality is valid for F if and only if it is valid for each P^h, $h \in Q$.

For this purpose we used the H-polyhedral (or hyperplane-) representation of the polyhedra

$$P^h := \left\{ x \in \begin{pmatrix} \tilde{A}x \geq \tilde{b} \\ D^h x \geq d_0^h \end{pmatrix} \right\}, \quad h \in Q. \tag{12.1}$$

If the system defining each P^h is of dimensions $m \times n$, the higher dimensional representation of conv$F$ has $O(q \times m)$ constraints and $O(q \times n)$ variables. This representation is compact, more precisely linear in q. Next we look at another representation [31], which, while not compact, has important advantages over the above one.

If we represent each P^h in V-polyhedral (or vertex-ray-) form, then

$$F = \bigcup_{h \in Q} P^h, \text{ where } P^h = \text{conv} V^h + \text{cone} R^h, \tag{12.2}$$

with V^h and R^h standing for the set of vertices and extreme rays, respectively, of P^h, $h \in Q$. Using this representation, we have

Proposition 12.1 *The inequality $\alpha x \geq \beta$ is valid for F if and only if*

$$\begin{aligned} &\alpha p \geq \beta \text{ for all } p \in V^h \\ &\alpha r \geq 0 \text{ for all } r \in R^h \quad h \in Q \end{aligned} \tag{12.3}$$

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_12

Proof $\alpha x \geq \beta$ is valid for F if and only if it is valid for all P^h, $h \in Q$; i.e., using (12.2), if and only if it satisfies (12.3). □

Another way of looking at this theorem is as follows.

The family of valid inequalities for F is the reverse polar of F, $F^{\#}_{\beta} := \{\alpha \in R^n : \alpha x \geq \beta, \forall x \in F\}$. But since F is the union of polyhedra, and polarity replaces union by intersection, we have

$$\begin{aligned} F^{\#}_{\beta} &= \left(\bigcup_{h \in Q} P^h\right)^{\#}_{\beta} \\ &= \bigcap_{h \in Q} (P^h)^{\#}_{\beta} \\ &= \{\alpha \in \mathbb{R}^n : \alpha x \geq \beta, \forall x \in P^h, h \in Q\} \\ &= \{\alpha \in \mathbb{R}^n : \alpha x \geq \beta, x \in V^h, h \in Q; \alpha x \geq 0, x \in R^h, h \in Q\} \end{aligned}$$

which is the same as (12.3).

The reader may be struck by the similarity of Proposition 12.1 to Theorem 11.4 of Sect. 11.3 on generalized intersection cuts. Indeed, V-polyhedral cuts can be viewed as an extension of GIC's, in which the intersection points of some edges of P with the boundary of a P_I-free set S are replaced with points outside int (S) that may not lie on any edges of P.

Unlike the representation used earlier, the system (12.3) is not compact, as its number of constraints, given by the number of vertices and extreme rays of all the polyhedra in the union forming F, is in general exponential in n. On the other hand, this is a system in only n variables. Furthermore, the number n of variables in (12.3) is not affected by the number of terms of the disjunction. This is in sharp contract with (CGLP) of the lift-and-project approach, whose number of variables increases with q, the number of terms in the disjunction. So the comparison between the computational efficiency of the two approaches depends on q and shifts in the favor of the V-polyhedral representation with the increase of q. This opens up the possibility of generating cuts from disjunctions with many more terms than the two or four explored so far. As cuts from stronger disjunctions tend to be stronger, this approach also holds the promise of accessing deeper cuts of rank 1, i.e. without recourse to recursion which tends to lead to numerical difficulties. So in order to make efficient use of this representation, we must find a way to generate strong cuts by using a small subset of the constraints. Next we describe ways of doing this.

12.1 V-Polyhedral Cut Generator Constructed Iteratively

The first approach we discuss is that of [31]. Here the intention is to generate cuts by solving (12.3) for the objective function $\min \alpha\bar{x} - \beta$, where $\bar{x}$ is the LP optimum. However, since the constraints of (12.3) are given by the exponentially many vertices and extreme rays of the P^h, $h \in Q$, most of which are unknown, this approach solves (12.3) iteratively, by *row generation*. Furthermore, it amends (12.3) with the normalization constraint $\alpha y = 1$, where y is the constant vector described in Sect. 10.6. Let us denote $V = \bigcup_{h\in Q} V^h$, $R = \bigcup_{h\in Q} R^h$.

Suppose we have $V_1 \subset V$ and $R_1 \subset R$. If we solve

$$\begin{aligned} \min\ & \alpha\bar{x} - \beta \\ \text{s.t.}\ & \alpha p^i - \beta \geq 0 \quad && p^i \in V_1 \\ & \alpha r^i \geq 0 && r^i \in R_1 \\ & \alpha y = 1 \end{aligned} \tag{12.4}$$

we obtain an inequality $\alpha^1 x \geq \beta^1$ valid for all extreme points in V_1 and all extreme rays in R_1, but not necessarily for those in V and R. To check if $\alpha^1 x \geq \beta^1$ is valid for all of $\text{conv}F$, we can solve a linear program of the form

$$\begin{aligned} \min\ & \alpha^1 x \\ \text{s.t.}\ & x \in \text{conv}F \end{aligned} \tag{12.5}$$

If (12.5) is bounded and x^1 is an optimal extreme solution with $\alpha^1 x^1 < \beta^1$, then x^1 is a point from $V \setminus V_1$ that violates the current inequality $\alpha^1 x \geq \beta^1$. Hence we can replace V_1 by the larger set $V_2 = V_1 \cup \{x^1\}$. If (12.5) is unbounded then we can find an extreme ray of (12.5) with a direction vector x^1 such that $\alpha^1 x^1 < 0$, which we can add to R_1 to obtain a larger set $R_2 = R_1 \cup \{x^1\}$. We then repeat this with the new sets V_2 and R_2 to obtain a new inequality $\alpha^2 x \geq \beta^2$, and keep repeating until in some iteration k we obtain an optimal solution x^k to (12.5) which satisfies $\alpha^k x \geq \beta^k$ for the last solution (α^k, β^k) to (12.4). This solution demonstrates that a valid inequality has been found.

Since the x^i obtained from solving (12.5) are *extreme* points or rays of F, the finiteness of V and R guarantees that the process will terminate. The procedure is outlined in Fig. 12.1. In the following we will refer to the problem in Step 2 as the *master problem* and to the problem in Step 3 as the *separation problem*. The procedure described here is isomorphic to applying Benders' decomposition to $(\text{CGLP})_Q$.

So far we have not considered the possibility that the master problem in Step 2 of Fig. 12.1 could be unbounded. This is where we need a certain property of the normalization $\alpha y = 1$. If we choose $y = \bar{x} - x^*$, where $x^* \in \text{conv}(V_1)$, then the master problem will always be bounded (see [32]).

Step 1 Let $k = 1$, $R_1 \subset R$ and $V_1 \subset V$ with $V_1 \neq \emptyset$

Step 2 Let (α^k, β^k) be an optimal solution to the master problem:

$$\begin{array}{lll} \min & \alpha\bar{x} - \beta & \\ \text{s.t.} & \alpha p^i - \beta \geq 0 & p^i \in V_1 \\ & \alpha r^i \geq 0 & r^i \in R_1 \\ & \alpha y = 1 & \end{array}$$

Step 3 Solve the separation problem:

$$\begin{array}{ll} \min & \alpha^k x \\ \text{s.t.} & Ax \geq b \\ & \bigvee_{h \in Q} D^h x \geq d^h \end{array}$$

If the problem is *bounded*, let x^k be an optimal solution. If $\alpha^k x^k \geq \beta^k$ then go to Step 4. Otherwise, set $V_{k+1} = V_k \cup \{x^k\}$ and $R_{k+1} = R_k$.

If the problem is *unbounded*, let x^k be the direction vector of an extreme ray satisfying $\alpha^k x^k < 0$. Set $R_{k+1} = R_k \cup \{x^k\}$ and $V_{k+1} = V_k$.

Set $k \leftarrow k + 1$ and repeat from Step 2.

Step 4 The inequality $\alpha^k x \geq \beta^k$ is a valid inequality for F. Stop.

Fig. 12.1 Iterative procedure for generating a valid inequality: version 1

The iterative procedure of Fig. 12.1 can be modified in its Step 3 as follows. Instead of adding to the master problem the inequality corresponding to the extreme point x^k that minimizes $\alpha^k x$, i.e. violates the inequality $\alpha^k x \geq \beta^k$ by a maximum amount, one can add *all* the violating extreme points or rays encountered in solving the separation problem. We call this *version 2*. Since the separation problem is usually solved by solving an LP over each term of the disjunction, version 2 of the iterative procedure does not require more time to solve the separation problem. On the other hand, it builds up faster the master problem, but it also creates one with a larger number of constraints. On balance, version 2 seems better (see the section on computational results).

12.1.1 Generating Adjacent Extreme Points

In this section we consider one approach towards reducing the number of extreme points that need to be considered in the separation problem.

Let x^F denote an optimal point of F, which can be computed by optimizing over each P^h, $h \in Q$, and choosing the best solution. A reasonable constraint to impose on the cut is to require it to be tight at x^F. Suppose we impose this restriction, i.e. add to (12.4) the equation $\alpha x^F - \beta = 0$, and (α^k, β^k) is the solution to the master problem at iteration k. Then either $\alpha^k x \geq \beta^k$ is a valid inequality for F or there exists a vertex *adjacent* to x^F (or possibly an extreme ray incident with x^F) in conv$F$ which violates $\alpha^k x \geq \beta^k$. This is an immediate consequence of the convexity of conv$F$.

It follows from this observation that when searching for a violating extreme point (or ray) in the separation algorithm of Fig. 12.1 we only need to consider extreme points adjacent to x^F (or extreme rays incident with x^F) in conv$F$.

We now turn to the problem of identifying the extreme points adjacent to x^F. Consider the disjunctive cone C_{x^F} defined by

$$
\begin{aligned}
C_{x^F} = \{(x', x'_0) \in \mathbb{R}^n \times \mathbb{R}_+ | \; & Ax' + (Ax^F - b)x'_0 \geq 0 \\
& \textstyle\bigvee_{h \in Q} D^h x' + (D^h x^F - d^h)x'_0 \geq 0 \,\}
\end{aligned}
$$

This cone is obtained from F by first translating F by $-x^F$ such that x^F is translated into the origin, and then homogenizing the translated polyhedron. The following Theorem gives the desired property. Let cone(C_{x^F}) be the conical hull (positive hull) of C_{x^F}.

Theorem 12.2 *Let C be the projection of cone(C_{x^F}) onto the x-space. Then the extreme rays of the convex cone C are in one-to-one correspondence with the edges of* conv$F$ *incident with x^F.*

Proof We consider the mapping $x \rightarrow x' = x - x^F$ from points $x \in$ conv$F$ to rays $x' \in C$. It should be clear that $x' \in C$, since we have $(x - x^F, 1) \in$ cone(C_{x^F}).

Now, take two distinct points $x^1, x^2 \in$ conv$F$ on any edge of conv$F$. These points are mapped into the vectors $(x^1 - x^F)$ and $(x^2 - x^F)$. If and only if the two points lie on an edge of conv$F$ incident with x^F will the vectors $(x^1 - x^F)$ and $(x^2 - x^F)$ describe the same ray in C. Thus only edges of conv$F$ correspond to edges of C.

The converse is also true. Let x' be an extreme ray of C. Then there exists x'_0 such that (x', x'_0) is a ray of cone(C_{x^F}). In particular we can scale the point such that $(x', x'_0) \in C_{x^F}$. Consider the line segment $[x^F, x^F + x']$. We first show that there exists $\gamma > 0$ such that $\tilde{x} = x^F + \gamma x' \in$ conv$F$. If we plug $\tilde{x}$ into F we obtain $A(x^F + \gamma x') \geq b$ and $D^h(x^F + \gamma x') \geq d^h$ for each $h \in Q$. If $x'_0 = 0$ we have $Ax' \geq 0$ and $D^h x' \geq 0$ for some $h \in Q$. In other words, x' is a ray of F, and in particular $\tilde{x} \in$ conv$F$ for any $\gamma > 0$. If $x'_0 > 0$ and we choose $\gamma \leq \frac{1}{x'_0}$ then $\tilde{x}$ satisfies F since $(x', x'_0) \in C_{x^F}$.

Next we show that $\tilde{x}$ lies on an edge of conv$F$ incident with x^F. For a contradiction, suppose that $\tilde{x}$ is *not* on such an edge. Then there exists $x^1, x^2 \in$ conv$F$ not on the line through x^F and $\tilde{x}$ (i.e., affinely independent from x^F and $\tilde{x}$), such that $\tilde{x}$ is a strict convex combination of x^1 and x^2. Hence there exists $\lambda \in \mathbb{R}$ with $0 < \lambda < 1$ such that $x = \lambda x^1 + (1-\lambda)x^2$. Now, consider the rays $(\tilde{x}', \tilde{x}_0') = (\tilde{x} - x^F, 1)$, $(x^{1'}, x^{1'}_0) = (x^1 - x^F, 1)$ and $(x^{2'}, x^{2'}_0) = (x^2 - x^F, 1)$. Since $\tilde{x}, x^1, x^2 \in$ conv$F$ it follows that $(\tilde{x}', \tilde{x}_0')$, $(x^{1'}, x^{1'}_0)$, $(x^{2'}, x^{2'}_0) \in \text{cone}(C_{x^F})$. Hence $(\tilde{x} - x^F)$, $(x^1 - x^F)$ and $(x^2 - x^F)$ are rays of C. We have that $(\tilde{x} - x^F) = \gamma x'$, so $x' = \frac{\lambda}{\gamma}(x^1 - x^F) + \frac{1-\lambda}{\gamma}(x^2 - x^F)$. We are thus able to write x' as a positive combination of linearly independent vectors, contrary to the fact that x' is an *extreme* vector of C. This shows that the extreme ray x' is the map of the edge of conv$F$ incident with x^F containing $\tilde{x}$. □

Since any vertex adjacent to x^F in conv$F$ by definition shares an edge incident with x^F, the immediate result of this theorem is that one only needs to consider the extreme rays of C. The relationship between rays (x', x_0') of C_{x^F} and points or rays x of F is

$$x = \begin{cases} \frac{x'}{x_0'} + x^F & \text{if } x_0' > 0 \\ x' & \text{if } x_0' = 0 \end{cases}$$

Let $\alpha^k x \geq \beta^k$ be the current iterate from our procedure. To check if there is a violating point adjacent to x^F, we first need to translate and homogenize the inequality, in accordance with what was done to obtain C_{x^F}. The translation results in the inequality $\alpha^k x' \geq \beta^k - \alpha^k x^F$, but since we imposed on (12.4) the constraint $\alpha x^F = \beta$, the righthand side becomes zero, and the coefficient for x_0' after homogenizing will also be zero; hence we obtain the inequality $\alpha^k x' \geq 0$. We can thus state

Proposition 12.3 *Let $x \in F$. $\alpha^k x < \beta^k$ if and only if $\alpha^k x' < 0$ for $(x', x_0') = (x - x^F, 1) \in C_{x^F}$.*

A violating vertex of conv$F$ adjacent to x^F can be found by solving the following disjunctive program:

$$\begin{aligned} &\min \tilde{\alpha} x' \\ &\text{s.t.} \quad Ax' + (Ax^F - b)x_0' \geq 0 \\ &\qquad \bigvee_{h \in Q} D^h x' + (D^h x^F - d^h)x_0' \geq 0 \\ &\qquad \alpha^k x' = -1 \\ &\qquad x_0' \geq 0 \end{aligned} \tag{12.6}$$

The equation $\alpha^k x' = -1$ serves the dual purposes of truncating the cone C_{x^F}, and restricting the feasible set to those solutions that satisfy the condition of Proposition 12.3. Any solution to problem (12.6) corresponds to a violating point in F, but to obtain a violating *extreme* point, one has to minimize an objective over this set. If one chooses $\tilde{\alpha}$ such that $\tilde{\alpha}x' \geq 0$ is valid for C_{x^F} then the problem (12.6) will be bounded.

To guarantee that the solution obtained corresponds to a vertex of F adjacent to x^F, one must first project out x'_0, according to Theorem 12.2. This can be done by e.g. applying the Fourier-Motzkin projection method. The size of the resulting set of constraints will depend on the number of constraints present in the system $D^h x' \geq d^h$, but since in most cases of interest $D^h x' \geq d^2$ can be replaced with a single constraint, the cost of projecting out x'_0 is typically not high.

12.1.2 How to Generate a Facet of conv F in n Iterations

For the iterative procedure in Fig. 12.1, no bound is available on the number of iterations required to obtain a valid inequality, except the trivial bound which is the total number extreme points and extreme rays of conv$F$. In this section we present a method which will find a facet-defining inequality for conv$F$ in a number of iterations that only depends on the dimension of the problem. However, in doing this we can no longer guarantee that the resulting inequality will minimize $\alpha\bar{x} - \beta$ over (12.3).

The basic idea is to start with an inequality that is known to be valid and supporting for F. Finding such an inequality should not pose a problem. Using x^F we can easily give such an inequality: $cx \geq cx^F$. Then through a sequence of rotations this inequality is turned into a facet-defining inequality for F. Each rotation will be chosen such that the new cut will be tight at one more vertex of F than the previous cut.

An illustration is provided in Fig. 12.2. This figure presents two polyhedra, P_1 and P_2, whose union is F. Our initial plane is H_1 which supports F only at the point x_1. The first rotation is performed around the axis a_1 through x_1 which rotates H_1 into H_2. Now H_2 is a plane touching F at the two points x_1 and x_2. Finally, H_2 is rotated around the axis a_2 through the points x_1 and x_2. This brings us to the final plane H_3, which is tight at x_1, x_2 and x_3, a maximum independent set on a facet of conv$F$.

The idea of hyperplane rotation is implemented by performing a linear transformation, in which the current inequality $\alpha^k x \geq \beta^k$ is combined with some target inequality $\tilde{\alpha}x \geq \tilde{\beta}$. The objective is to find a maximal γ such that $(\alpha^k + \gamma\tilde{\alpha})x \geq (\beta^k + \gamma\tilde{\beta})$ is a valid inequality for F. Suppose V_k is the set of extreme points of conv$F$ for which $\alpha^k x \geq \beta^k$ is tight. If $(\tilde{\alpha}, \tilde{\beta})$ is chosen such that the inequality $\tilde{\alpha}x \geq \tilde{\beta}$ is also tight at V_k then the resulting inequality must be tight at V_k. If we

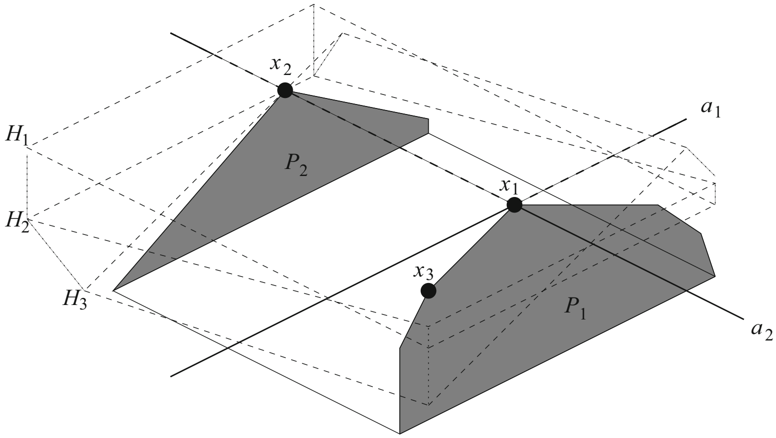


Fig. 12.2 Example showing how an initial supporting hyperplane H_1 is rotated through H_2 into a facet-defining hyperplane H_3

further ensure that $\tilde{\alpha}x \geq \tilde{\beta}$ is *invalid* for F then there is a finite maximal γ for which $(\alpha^k + \gamma\tilde{\alpha})x \geq (\beta^k + \gamma\tilde{\beta})$ is valid for F.

It can be shown that the maximum value γ^* of γ is given by the optimal objective value of the disjunctive program

$$\begin{aligned} \gamma^* = \min\ & \alpha^k x - \beta^k x_0 \\ \text{s.t.}\ & Ax - bx_0 \geq 0 \\ & \textstyle\bigvee_{h \in Q} D^h x - d^h x_0 \geq 0 \\ & x_0 \geq 0 \\ & \tilde{\alpha}x - \tilde{\beta}x_0 = -1 \end{aligned} \tag{12.7}$$

The optimal solution (x', x'_0) obtained from solving (12.7) defines a point $x = \frac{x'}{x'_0} \in F$ (or a ray of F if $x'_0 = 0$) affinely independent of V_k, for which the new inequality is tight.

Next we present an outline of a procedure that finds a facet-defining inequality of $\text{conv}F$ in n iterations. This procedure is given in Fig. 12.3. For simplicity we assume that F is bounded and thus we omit the possibility of extreme rays.

We leave the choice of $(\tilde{\alpha}, \tilde{\beta})$ unspecified, since there are many technical details involved in this. Suffice it to say that the inequality $\tilde{\alpha}x \geq \tilde{\beta}$ should be chosen as one that is "deeper" than $\alpha^k x \geq \beta^k$ with respect to $\bar{x}$, the point one wants to cut off. If this is done and if the initial inequality $\alpha^1 x \geq \beta^1$ already cuts off $\bar{x}$ then the above procedure produces a facet-defining inequality that also cuts off $\bar{x}$.

Step 1	Let $\alpha^1 x \geq \beta^1$ be a valid inequality for F tight for $x^1 \in F$. Set $V_1 = \{x^1\}$ and $k = 1$.
Step 2	Choose a target inequality $\tilde{\alpha}x \geq \tilde{\beta}$ tight for V_k and not valid for F.
Step 3	Solve (12.7) to obtain γ^* and a point x^k.
Step 4	Set $(\alpha^{k+1}, \beta^{k+1}) = (\alpha^k, \beta^k) + \gamma^*(\tilde{\alpha}, \tilde{\beta})$ and $V_{k+1} = V_k \cup \{x^k\}$. Increment $k \leftarrow k+1$.
Step 5	If $k = n$ stop, otherwise repeat from Step 2.

Fig. 12.3 Procedure to obtain a facet-defining inequality for F in n iterations

There are other technical details pertaining to this procedure which we do not cover here, e.g. how to choose the initial inequality $\alpha^1 x \geq \beta^1$, how to decide when a cut $\tilde{\alpha}x \geq \tilde{\beta}$ is "deeper" with respect to $\bar{x}$.

12.1.3 Cut Lifting

An important ingredient of the lift-and-project method is that of working in a subspace [19, 20]. If a variable is at its lower or upper bound in the optimal solution $\bar{x}$ to the LP-relaxation, it can be ignored for the purpose of cut generation. Thus cuts are generated in a subspace and are then *lifted* to a cut that is valid for the full space by computing the coefficients of the missing variables. These coefficients are computed using the multipliers $\{u^h\}_{h \in Q}$ that satisfy the constraints of $(\text{CGLP})_Q$ for the subspace cut coefficients (see [19, 20] for details). The procedures featured in Figs. 12.1 and 12.3 do not cover the cut lifting aspect and thus do not specify how to compute these multipliers; but once we have determined the cut $\alpha x \geq \beta$, we can fix the value of α and β in (CGLP). This will decouple the constraints and leave $|Q|$ independent linear equality problems from which the multipliers associated with (α, β) are easy to calculate. One potential problem with working in a subspace is the choice of the latter: if the space is restricted too much, the feasible region may become empty. To avoid this, one can require that each term of the disjunction be non-empty in the subspace. In the testing to be discussed in the next section, the subspace used was the smallest one that contains the nonzero components of the optimal solution from each of the separate linear programs of the disjunction. This was easy to implement, since the method used to solve the disjunctive programs of Step 3 in the procedures of Figs. 12.1 and 12.3 was to solve a linear program over each term of the disjunction and retain the best solution found.

12.1.4 Computational Testing

To test these ideas, Balas and Perregaard [31] uses disjunctions generated by using a partial branch and bound procedure with no pruning except for infeasibility, to generate a search tree with a fixed number, k, of leaves. The union of subproblems corresponding to these k leaves is then guaranteed to contain the feasible solutions to the mixed integer program under consideration. Therefore the disjunction whose terms correspond to the leaves of this partial search tree is a valid one, although the number of variables whose values are fixed at 0 or 1 in the different terms of the disjunction need not be the same.

The branch-and-bound procedure used here is a simple one whose only purpose is to provide a disjunction of a certain size. The branching rule used is to branch on an integer constrained variable whose fractional part is closest to $\frac{1}{2}$. For node selection the best-first rule is used. This search strategy will quickly grow a sufficiently large set of leaf nodes for a disjunction, and use of the best-first search also ensures a relatively strong disjunction with respect to the objective function.

For each problem instance a round of up to 50 cuts was generated, each from a disjunction coming from a search tree initiated by first branching on a different 0-1 variable fractional in the LP solution. The cuts themselves were generated by five different methods, each using the same disjunctions:

1. By using the simplex method to solve $(\text{CGLP})_Q$ in the higher dimensional space;
2. By using the iterative procedure of Fig. 12.1, version 1;
3. By using the iterative procedure of Fig. 12.1, version 2;
4. By using the iterative procedure that generates only extreme points adjacent to x^F;
5. By using the n-step procedure of Fig. 12.3 to find a facet defining inequality.

The main purpose of this computational testing was to compare the proposed procedures with each other and with the standard procedure of solving $(\text{CGLP})_Q$ from the point of view of their sensitivity to the number of terms in the disjunctions from which the cuts are generated. The comparison, shown in Table 12.1, features the total time required to generate up to 50 cuts for each of the 14 test problems, (a) by solving the higher dimensional $(\text{CGLP})_Q$ as a standard linear program and (b) by using the iterative procedure of Fig. 12.1, version 2 (which adds to the master problem all the violators found in Step 3). These numbers are compared for disjunctions with 2, 4, 8 and 16 terms, with the outcome that the times in column (b) are worse than those in column (a) for disjunctions with 2 terms ($|Q| = 2$), roughly equal or slightly worse for $|Q| = 4$, considerably better for $|Q| = 8$, and vastly better for $|Q| = 16$. For the method featured in column (b), the total computing time grows roughly linearly with $|Q|$: in about half of the 14 problems, the growth is slightly less than linear, and in the other half it is slightly more than linear. The numbers in column (a) grow much faster, which is understandable in light of the fact that the number of variables *and* constraints of the higher-dimensional $(\text{CGLP})_Q$ increases with $|Q|$.

Table 12.1 Total time for up to 50 cuts

	|Q|=2		|Q|=4		|Q|=8		|Q|=16	
	(a)	(b)	(a)	(b)	(a)	(b)	(a)	(b)
BM21	0.07	0.21	0.32	0.40	1.91	0.85	7.85	1.52
EGOUT	0.08	0.16	0.23	0.25	0.82	0.46	5.76	0.82
FXCH.3	0.52	0.80	1.53	1.50	9.86	2.54	65.19	4.01
LSEU	0.07	0.14	0.23	0.27	0.97	0.56	6.33	1.47
MISC05	1.13	1.82	5.15	5.07	52.25	13.72	658.88	33.00
MOD008	0.09	0.13	0.19	0.33	0.60	0.67	2.69	1.54
P0033	0.07	0.13	0.19	0.24	0.78	0.57	2.89	1.14
P0201	1.40	2.09	8.12	4.80	88.26	10.67	609.17	26.45
P0282	0.85	1.90	2.02	4.51	8.94	17.75	86.24	44.36
P0548	2.77	11.85	7.94	9.00	37.75	20.51	276.18	44.25
STEIN45	36.42	148.38	99.71	157.86	280.71	159.04	1082.78	222.32
UTRANS.2	0.50	0.81	1.55	1.50	10.26	3.53	111.06	13.65
UTRANS.3	0.78	1.26	2.71	2.29	26.47	4.93	173.20	9.31
VPM1	0.45	0.79	1.29	1.62	11.38	2.97	81.29	6.82

(a) Solving (CGLP)
(b) Using the Iterative Method of Fig. 12.1

The above described computational experience, limited as it is, shows for all of the 14 instances tested, without exception, that the ratio between the computational effort needed by the approach of this chapter versus the lift-and-project procedure decreases steadily with $|Q|$, the number of terms in the disjunction, and becomes less than 1 somewhere between $|Q| = 4$ and 8.

12.2 Relaxation-Based V-Polyhedral Cut Generators

In our next approach, we seek a relaxation of the feasible set of (12.3) which is of manageable size and still yields strong cuts. The material of this section is based on joint work of the author with his PhD student Alex Kazachkov (see the latter's dissertation [90]). Clearly, although (12.3) has exponentially many constraints, any basic solution, hence any cut, involves only n of them. Each basic solution yields a cut whose direction is defined by the objective for which that basic solution is optimal. The question is, which of the constraints to keep? Since one of our goals is to cut off the LP optimum $\bar{x}$ by as much as possible, it is reasonable to keep those vertices of each P^h which optimize $c\bar{x}$ over P^h. However, if we keep only the constraints associated with these, the resulting subsystem of (12.3) is not guaranteed to yield valid cuts. It is easy to construct examples of a hyperplane that contains the optimal points of some or all of the P^h, $h \in Q$, yet the inequality based on it cuts off some feasible integer points. However, the following important property holds. For $h \in Q$, let $\tilde{P}^h$ be any polyhedron such that

$$P^h \subseteq \tilde{P}^h \subseteq C(x^h),$$

where $C(x^h)$ is the LP cone associated with x^h, and let $\tilde{V}^h$ and $\tilde{R}^h$ denote the set of vertices and extreme rays, respectively, of $\tilde{P}^h$. Then we have

Theorem 12.4 *The inequality $\alpha x \geq \beta$ is valid for P_I if (α, β) satisfies*

$$\begin{aligned} &\alpha p \geq \beta \textit{ for all } p \in \tilde{V}^h \\ &\alpha r \geq 0 \textit{ for all } r \in \tilde{R}^h \quad h \in Q \end{aligned} \tag{12.8}$$

Proof Let $\mathcal{C} := \operatorname{conv} \bigcup_{h \in Q} \tilde{V}^h + \operatorname{cone} \bigcup_{h \in Q} \tilde{R}^h$ and let (α, β) be any solution to (12.8). Then $\alpha x \geq \beta$ for all $x \in \mathcal{C}$ and since $P_I \subseteq \mathcal{C}$, $\alpha x \geq \beta$ does not cut off any point in P_I. □

Next we examine some of the properties of the system (12.8). In order to generate cutting planes, we will solve a linear program over this constraint set for various objective functions. Since the system (12.8) is homogeneous, it has to be normalized in order for the linear program to have a finite solution. The simplest normalization that suggests itself is $\beta \in \{1, -1\}$; and if we choose to work in the nonbasic space associated with $\bar{x}$, the LP optimal solution, then all cuts have positive righthand sides and thus $\beta = 1$ is a valid normalization. If we denote by $\mathcal{P} = \cup_{h \in Q} \tilde{V}^h$ the points and by $\mathcal{R} = \cup_{h \in Q} \tilde{R}^h$ the rays that provide the constraints of (12.8), the linear program becomes

$$\begin{aligned} \min\ &\alpha z \\ &\alpha p \geq 1 \;\; p \in \mathcal{P} \\ &\alpha p \geq 0 \;\; r \in \mathcal{R} \end{aligned} \tag{PRLP}$$

where we call $(\mathcal{P}, \mathcal{R})$ the point-ray collection underlying (PRLP), the point-ray linear program.

Now if we use $\tilde{P}^h = C(x^h)$ for $h \in Q$, we get a V-polyhedral cut generator with n variables and $q(n-q)$ constraints. This is of practically manageable size—the question is, how strong are the cuts provided by its solutions. But before addressing this question, we ask another one: what is the connection between cuts obtained from this V-polyhedral relaxation and the lift-and-project cuts obtained from the (CGLP) based on the same disjunction and using the same objective function. We answer the question for the general case of polyhedra $\tilde{P}^h$ of the form

$$\tilde{P}^h = \{x \in \mathbb{R}^n : \tilde{D}^h x \geq \tilde{d}_0^h\}, h \in Q,$$

where $P^h \subseteq \tilde{P}^h \subseteq C(x^h)$, $h \in Q$. Consider the L&P cut generating linear program

$$\begin{aligned} \alpha - u^h \tilde{D}^h &= 0 \\ -\beta + u^h \tilde{d}_0^h &\geq 0 \qquad h \in Q \\ \sum_{h \in Q} u^h e &= 1 \\ u^h &\geq 0, h \in Q \end{aligned} \tag{12.9}$$

Theorem 12.5 *The V-polyhedral cuts $\alpha x \geq \beta$ corresponding to basic feasible solutions to (12.8) are equivalent to the lift-and-project cuts $\bar{\alpha} x \geq \bar{\beta}$ corresponding to basic feasible solutions $\bar{w} = (\bar{\alpha}, \bar{\beta}, \bar{u}^h, h \in Q)$ to (12.9). They are also equivalent to the generalized intersection cuts from the P_I-free set $S := \{x \in \mathbb{R}^h : \bar{u}^h \tilde{D}^h \leq \bar{u}^h \tilde{d}_0^h, h \in Q\}$.*

Proof The first equivalence follows from the fact that both (12.8) and (12.9) describe conv $\bigcup_{h \in Q} \tilde{P}^h$. The second one follows from the equivalence of GIC's to the corresponding L&P cuts. □

As already mentioned, the system (12.8) seems similar to the system (11.5) defining GIC's. However, the two systems are different. The points $p^j \in \mathcal{P}$ of the system (11.5) giving rise to the GIC's equivalent to the V-polyhedral cuts from (12.8) are intersection points of some (extended) edges of P with the boundary of the P_I-free set $S = \{x \in \mathbb{R}^n : \bar{u}^h \tilde{D}^h \leq \bar{u}^h \tilde{d}_0^h, h \in Q\}$, whereas the points $p \in \bigcup_{h \in Q} \tilde{V}^h$ of the system (12.8) typically lie outside of S. Nevertheless, the two systems yield equivalent cuts.

In view of the equivalence of V-polyhedral cuts from (12.8) and lift-and-project cuts from (12.9), the question arises as to the comparative cost of generating cuts from each of these two representations. A rough estimate follows for the case when $\tilde{P}^h = C(x^h)$.

The effort involved in an iteration of the simplex method on a linear program with an $m \times n$ coefficient matrix is $O(\min\{m, n\}^3)$, and the number of expected iterations is $O(m + n)$ (see Vanderbei [120]). Assuming that A is $m \times n$, with $m = \theta n$ for some $\frac{1}{2} \leq \theta \leq 2$, the computational effort of creating the system (12.8) and using it to generate k cuts can be assessed as follows:

– Computing each of the q points x^h and the extreme rays of $C(x^h)$ by a sequence of dual simplex pivots from the optimal LP solution can be estimated to take no more than $O(q \cdot m^3(m+n)) \approx O(q \cdot n^4)$ operations, the cost of solving q linear programs.
– Solving the system (12.8) whose coefficient matrix is $(q \cdot n) \times n$ can be estimated to take $O(n^3(qn + n)) = O((q + 1)n^4)$ operations.

Thus the total computational effort involved in building the system (12.8) and generating k cuts is roughly

$$O(q \cdot n^4) + k(q + 1)n^4) = O((k(q + 1) + q)n^4) \tag{a}$$

On the other hand, the computational effort involved in generating k lift-and-project cuts from the CGLP with constraint set (12.9) can be assessed as follows:

The system (12.9) has $q(n+1)+1$ constraints (other than the sign restrictions), and $q(n+1)$ variables, so solving it k times can be estimated to take

$$O(k \cdot q^3(n+1)^3 \cdot q(n+1)) = O(kq^4(n+1)^4) \tag{b}$$

Thus we have that

$$\begin{aligned}\frac{(a)}{(b)} &= \frac{(k(q+1)+q)n^4}{kq^4(n+1)^4} \\ &= \frac{1+\frac{1}{q}+\frac{1}{k}}{q^3\left(\frac{n+1}{n}\right)^4} < \frac{1}{q^3}\end{aligned}$$

In other words, the computational effort required to generate the cuts of Theorem 12.4, using $\tilde{P}^h = C(x^h)$, $h \in Q$, is at least q^3 times smaller than that of generating the same cuts through the L&P procedure.

It should be mentioned that the V-polyhedral cuts, like the equivalent lift-and-project cuts, are derived from a disjunction based on the integrality constraints on some basic variables of the LP optimal solution, and they can be strengthened by modularizing the nonbasic integer variables, as discussed in Sect. 11.8. However, the only way known to date of modularizing these cuts is by modularizing their lift-and-project equivalent. Although this equivalent comes from a cut generating linear program (12.9) much smaller than the one for the standard lift-and-project formulation (it has roughly $q \times n$ rather than $q \times (m+n)$ variables), modularizing it would still make the VPC's much more expensive.

12.3 Harnessing Branch-and-Bound Information for Cut Generation

A question that arises naturally when generating cuts from disjunctions is: which disjunctions to use? For instance, given an optimal solution $\bar{x}$ to the LP relaxation of a MIP, valid disjunctions can be derived from any subset of fractional components of $\bar{x}$. Which components to use and in what combination? One way to answer this question is to run a few iterations of a branch-and-bound code that has a number of devices for choosing judiciously the variables on which to branch. To the extent that these techniques work as intended, the set of early (active) nodes of the branch-and-bound tree represent a "useful" disjunction of the feasible set for cut generation purposes. To illustrate this idea, suppose we have a partial branch-and-bound tree of the following form (to simplify the discussion, we assume a mixed 0-1 context):

node 0 (origin)
node 1: $x_1 = 0$
node 2: $x_1 = 1$
node 3: $x_1 = 0, x_2 = 0$ – pruned
node 4: $x_1 = 0, x_2 = 1$
node 5: $x_1 = 1, x_2 = 0$ – pruned
node 6: $x_1 = 1, x_2 = 1$
node 7: $x_1 = 0, x_2 = 1, x_3 = 0$
node 8: $x_1 = 0, x_2 = 1, x_3 = 1$
node 9: $x_1 = 1, x_2 = 1, x_3 = 0$
node 10: $x_1 = 1, x_2 = 1, x_3 = 1$

A valid disjunction is one among all active nodes. Nodes 0, 1, 2, 4 and 6 are inactive since they have been branched on, hence have been replaced by their descendants. Nodes 3 and 5 are inactive because they have been pruned, as their potential descendants cannot contain any improving feasible solution. So the active nodes are 7, 8, 9 and 10, and the above partial branch-and-bound tree implies the following valid disjunction:

$$(x_1, x_2, x_3) \in \{(0, 1, 0), (0, 1, 1), (1, 1, 0), (1, 1, 1)\} \tag{12.10}$$

or, equivalently, $x_2 = 1$ and $(x_1, x_3) = (0, 0) \vee (0, 1) \vee (1, 0) \vee (1, 1)$ which can be used to generate valid cuts for the instance in question.

The insight provided by the partial branch-and-bound procedure of the above example for choosing an appropriate disjunction for cut generation is two-fold: (a) among all possible subsets of variables to which a disjunction could be applied, it specifies a collection of tuples that presumably represent relatively important choices, as they have been selected for the top of the tree; (b) it provides the information that some terms of the (complete) disjunction can be dropped, as they are known to be empty; which of course strengthens the disjunction; and (c) for each term $\tilde{V}^h (= x^h)$ and $\tilde{R}^h (= \text{extr}\, C(x^h))$ are readily available.

In order to generate cuts from partial branch-and-bound trees, we specialize the system (12.8) as follows: we will use $\tilde{P}^h = C(x^h)$, $h \in Q$, where x^h is the optimal solution of the LP-relaxation at active node h and $C(x^h)$ is the associated polyhedral cone. Thus for each $h \in Q$, $\bar{V}^h = \{x^h\}$, and $\bar{R}^h$ is the set of extreme rays of $C(x^h)$. Furthermore, we will work in the nonbasic space of the optimal LP-solution $\bar{x}$, and since we want to generate cuts violated by $\bar{x}$, we set $\beta = 1$. Thus our cut generating LP will be of the form

$$\begin{aligned} \min \quad & \alpha z \\ & \alpha x^h \geq 1 \\ & \alpha r \geq 0, r \in \bar{R}^h \quad h \in Q, \end{aligned} \tag{12.11}$$

where z is an objective function vector chosen by criteria to be discussed below. The system (12.11) has n variables and $O(q \cdot n)$ constraints whose coefficient vectors are readily available from the partial branch and bound tree.

In [91], a computational study of V-polyhedral cut generation from partial branch-and-bound trees was carried out, with the objective of assessing (a) the strength of these cuts as measured by the integrality gap closed by one round of such cuts, and (b) the effectiveness of these cuts when added at the root node of a branch-and-bound procedure. The parameters of the study are defined by (1) the partial branch-and-bound trees used to generate the cuts, (2) the objective functions used in (12.11), and (3) the rank and the number of the cuts generated.

(1) Partial branch-and-bound trees with 2^k active nodes (leaves) were created for $k = 2, 3, \ldots, 6$, with each one used to generate cuts.

Surprisingly, generating cuts from branch-and-bound trees with 2^6 active nodes, i.e. from 64-term disjunctions, proved perfectly manageable.

(2) The following objective functions were used:

- The all 1's vector: $e = (1, \ldots, 1)$
- The optimal solution $\tilde{x}$ to P amended with a round of Gomory cuts
- The point $\hat{x}$ that minimizes cx^h over $h \in Q$, where $cx^h = \min\{cx : x \in P^h\}$
- Some of the remaining x^h and $r \in \bar{R}$.

(3) Only unstrengthened rank 1 cuts were generated, and their number was restricted (for comparability) to the number of Gomory cuts for the corresponding linear programs. The cuts generated were added to Gurobi's cut pool as "user cuts".

The tests were carried out on a set of 184 instances from MIPLIB [93], CORAL [60] and NEOS having at most 5000 rows and columns. The procedure was implemented in the COIN-OR framework and the solver used to test the effectiveness of the cuts when added to the root node of a branch-and-bound tree was Gurobi version 7.5.1 (see [91] for details). Below we summarize the most salient results:

(a) Integrality gap closed by VPC's.

Table 12.2 provides a summary of the average percent gap closed by GMICs, VPCs, and VPCs used together with GMICs, as well as the percent gap closed by

Table 12.2 Summary statistics for percent gap closed by VPC's

Set	# inst		G	DB	V	G+V	GurF	GurF+V	GurL	GurL+V
All	184	Avg (%)	17.28	24.03	15.60	26.95	25.99	33.03	46.48	52.07
		Wins			91	156		143		116
$\geq 10\%$	87	Avg (%)	14.41	37.73	29.55	33.47	20.03	32.59	38.81	49.95
		Wins			71	84		73		68
1K	81	Ave (%)	16.38	25.25	20.73	30.61	23.90	34.14	43.40	52.41
		Wins			51	70		66		57

one round of Gurobi's own cuts added at the root node of a `Gurobi` branch and bound tree and after the last round of cuts added by `Gurobi` at the root.

Column 1 indicates the set of instances: the first two data rows concern the 184 instances for which the disjunctive lower bound is strictly greater than the LP optimal valud, the second two data rows pertain to the subset of the 184 instances for which VPCs close at least 10% of the integrality gap, while the last two data rows refer to the subset of 81 instances with at most 1000 rows and 1000 columns. The first row for each set gives the average for the percent gap closed across the instances. The second row for each set shows the number of "wins", which is defined as an instance for which using VPCs closes strictly more (by at least 10^{-7}) of the integrality gap. For columns "V" and "G+V", wins are relative to column "G"; for "GurF+V" wins are counted with respect to column "GurF"; and for "GurL+V" wins are with respect to column "GurL".

Column 2 gives the number of instances in each set. Column 4 is the percent gap closed by GMICs when they are added to the LP relaxation. Column 5 is the percent gap "closed" by the disjunctive lower bound from the partial tree with 64 leaf nodes. Column 6 is the percent gap closed by VPCs; the subsequent column is the percent gap closed when GMICs and VPCs are used together. Columns 8 and 9 show the percent gap closed by `Gurobi` cuts from one round at the root, first without and then with VPCs added as user cuts. Columns 10 and 11 show the same, but after the last round of cuts at the root. `Gurobi` typically runs in parallel seven copies of an instance with seven random seeds, and columns 8 and 10 use the best result (maximum gap closed) by `Gurobi` across the seven random seeds tested.

The results indicate the strength of VPCs. Namely, using VPCs and GMICs together leads the average percent gap closed at the root to increase from 17% to 27%. The fact that VPCs on their own close less gap than GMICs (15.6% versus 17.3%) is due to the fact that, unlike the GMICs that modularize the coefficients of nonbasic integer variables, the VPCs generated in this study are unstrengthened, i.e. they are derived from a disjunction that uses the integrality of some basic variables, and they do not take advantage of the integrality of the nonbasics. (The mixed integer Gomory cuts without modularization, or without integer-constrained nonbasic variables, are simple disjunctive cuts dominated by the VPC's.) Even so, VPCs on their own close strictly more gap than GMICs for 91 instances; in comparison, we see that for 114 instances the disjunctive lower bound is greater than the optimal value after adding GMICs, so there are only 23 additional instances for which VPCs on their own could have gotten stronger results. For 11 of those 23 instances, we achieve the cut limit; implying that a higher percent gap might be achieved if we permitted more cuts to be generated. When VPCs are used with GMICs together, more gap is closed for 156 of the 184 instances. Perhaps even more indicative of the utility of VPCs are the results when VPCs are used as user cuts within `Gurobi`. For the first round of cuts at the root, the percent gap closed goes from 25% (without VPCs) to 33% (with them), with strictly better outcomes for 143 of the 183 instances. For the last round of cuts at the root, the percent gap closed increased from 46.5% to 52% by using VPCs.

The results are even more pronounced for the other two sets of instances. For the instances in which VPCs perform well (close at least 10% of the integrality gap on their own), VPCs and GMICs together close over double the gap closed by GMICs on their own, with improvements for 84 of the 87 instances in that set, and VPCs provide nearly a 30% improvement in the gap closed after the last round of cuts at the root node of `Gurobi` (50% compared to 39%). For the 81 instances with up to 1000 rows and columns, GMICs close 17% of the integrality gap, VPCs alone close 21%, while together they close 31%.

For the columns including VPCs, the result reported is the maximum percent gap closed across all partial tree sizes tested. One may initially assume that the strongest cuts would always come from the partial tree with 64 leaf nodes. This is indeed true of the disjunctive lower bound, but it does not always hold for VPCs. One reason is that there are likely to be more facet-defining inequalities for the disjunctive hull from the larger disjunctions. As a result, achieving the disjunctive lower bound may become more difficult, in particular considering we set a relatively conservative cut limit. Another reason, on an intuitive level, is that more of the facet-defining inequalities for the deeper disjunctions may not cut away $\bar{x}$, which are cuts we do not generate in these experiments. Finally, the rate of numerical issues goes up as the disjunctions get larger.

(b) Impact of VPC's on branch and bound.

We now turn to the second metric: the effect of our cuts on branch-and-bound in terms of time and number of nodes when VPCs are added as user cuts to `Gurobi`. There are 159 instances for which `Gurobi` is able to solve the instance to optimality within an hour either with or without using VPCs; we call this set "All" for this subsection. Of these 159 instances, there are 97 for which we were successfully able to generate VPCs for all six partial branch-and-bound trees; we refer to this as the "6 trees" set of instances. Table 12.3 contains a summary of the statistics for the set "All" in the top half and for the set "6 trees" in the bottom half. We further divide each set of instances into four bins, where bin $[t, 3600)$ contains the subset of instances which `Gurobi` solved within an hour but took at least t seconds to solve for all experiments. The first column of the table indicates which set and bin is being considered. The second column is the number of instances in that subset. The next column indicates the two summary statistics presented for each subset. The first row, "Gmean", for each subset is a shifted geometric mean (with a shift of 60 for time and 1000 for nodes). The second row, "Wins1", is the number of "wins" for each column with respect to `Gurobi` run with one random seed, a baseline that we will denote by "Gur". A win in terms of time is counted when the "Gur" baseline is at least 10% slower, to account for some variability in runtimes.

We now describe the eight remaining columns of Table 12.3. Column 4 gives the values (times) for "Gur". Column 5 provides the statistics for the fastest solution time by `Gurobi` when VPCs are added as user cuts (without accounting for cut generation time) across the six different partial trees tested per instance, but also including the time from "Gur" as one of the possible minima, indicating the option of not using VPCs for that instance. Column 6 incorporates the total cut generation

Table 12.3 Summary statistics for time to solve instances with branch-and-bound

			Time (s)			Nodes	
Set	# inst		Gur	V	Total	Gur	V
All	159	Gmean	81.48	63.79	68.50	6069	4549
[0, 3600)		Wins1		89	45		109
All	76	Gmean	276.89	199.21	217.23	28642	19152
[10, 3600)		Wins1		43	33		47
All	37	Gmean	869.71	652.80	715.91	88735	61536
[100, 3600)		Wins1		20	17		19
All	11	Gmean	2126.58	2016.00	2017.62	208021	157995
[1000, 3600)		Wins1		2	2		4
6 trees	97	Gmean	65.56	54.04	56.67	6747	5239
[0, 3600)		Wins1		57	30		73
6 trees	41	Gmean	260.64	199.38	210.78	46013	31386
[10, 3600)		Wins1		24	21		29
6 trees	19	Gmean	973.42	765.35	812.85	200983	135861
[100, 3600)		Wins1		9	9		10
6 trees	8	Gmean	2088.90	1983.85	1984.99	376028	261487
[1000, 3600)		Wins1		1	1		3

time, including the time taken to produce the partial branch-and-bound tree and set up the point-ray collection. Columns 7 and 8 give the number of nodes taken to solve each instance to optimality without and with VPCs.

The results show that using VPCs improves upon the baseline "Gur", yielding a considerable reduction in the average number of seconds and nodes to solve each instance.

Chapter 13
Unions of Polytopes in Different Spaces

Important classes of disjunctive sets can be formulated as unions of polyhedra in different spaces. As a rather general example, take a real world situation in which some action involving certain entities has an impact on some other entities. If a situation like this can be formulated as an expression involving one or more logical implications, then we have a model involving unions of polyhedra in different spaces. One reason for studying such situations is that they lend themselves to characterizations that lead to simpler and more efficient solution methods than the basic case of unions of polyhedra in the same space. Our treatment follows [17].

We start with the study of the dominant of a polytope, which turns out to be what is needed for the treatment of disjoint unions. Earlier studies of dominants and monotonization of polyhedra [23, 50] have focused on different aspects.

13.1 Dominants of Polytopes and Upper Separation

Given a polytope $P \subseteq [0, 1]^n$ in the unit cube described explicitly by inequalities, we study the polyhedron of points dominating a point in P, the set of nonnegative n-vectors π for which the associated inequality $\pi x \geq 1$ is valid for P, and the associated separation problem of finding a most violated inequality of this type cutting off a point $x^* \notin P$. After giving some standard definitions, we examine first the case in which P can be written in the form $\{x \in [0, 1]^n : Ax \geq 1\}$ with $A \geq 0$, and then we examine the general case.

We suppose throughout that P is a nonempty polyhedron in the nonnegative orthant, or a polytope rescaled to fit in the unit cube, and $N = \{1, \ldots, n\}$.

Definition 1 For a polyhedron $P \subseteq R^n_+$, $P^+ = P + R^n_+ = \{y : y \geq x \text{ for some } x \in P\}$ is the *dominant* of P.

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3_13

We consider the problem of finding the most violated valid inequality for P of the form $\sum_{j=1}^{n} \pi_j x_j \geq 1$ with $\pi \geq 0$. Notice that if $0 \in P$, there is no valid inequality of this type, and $P^+ = R^n_+$.

Definition 2 For a polyhedron $P \subseteq R^n_+$ and $x^* \in R^n_+$, the upper separation problem for (P, x^*) is that of finding

$$\alpha_P = \min\{\pi x^* : \pi x \geq 1 \text{ for all } x \in P, \pi \in R^n_+\}.$$

Specifically we see that an inequality of the required type cutting off x^* is obtained if and only if $\alpha_P < 1$.

Well-known results on polarity and blocking polyhedra show that the problem of describing the dominant P^+ and the upper separation problem for (P, x^*) are closely related.

Definition 3 For $P \subseteq R^n_+$, $P^* = \{\pi \in R^n_+ : \pi x \geq 1 \text{ for all } x \in P\}$ is the blocker of P.

Note that the blocker differs from the reverse polar in that it is restricted to R^n_+. Now the following observations are standard, see [78, 109].

1. $P^+ = P^{**} = \{x \in R^n_+ : \pi x \geq 1 \text{ for all } x \in P^*\}$.
2. $(P^+)^* = P^*$.
3. If $\{\pi^r\}_{r \in I}$ are the extreme points of P^*, $P^+ = \{x \in R^n_+ : \pi^r x \geq 1 \text{ for all } r \in I\}$ is a minimal description of P^+.
4. $\alpha_P = \min\{\pi x^* : \pi \in P^*\} = \min_{r \in I} \pi^r x^*$. Thus the upper separation problem for (P, x^*) is identical to the separation problem for (P^+, x^*), and the minimum is attained at an extreme point π^r of P^*.

Our goal here is to find a minimal inequality description of P^+ and use it to solve the upper separation problem and calculate α_P .

Definition 4 For a polytope $P \subseteq [0, 1]^n$, P is upper monotone (wrt $[0, 1]^n$) if $P = P^+ \cap [0, 1]^n$.

Suppose that $P \subseteq [0, 1]^n$ is upper monotone with the n-vectors $0 \notin P$ and $1 \in P$. We now assume without loss of generality that we are given such a polytope in the form:

$$P = \{x \in [0, 1]^n : Ax \geq 1\} \text{ with } A \in R^{m \times n}_+. \tag{13.1}$$

We also denote $P_i = \{x \in [0, 1]^n : a^i x \geq 1\}$ where a^i denotes the vector $(a_{i1}, \ldots, a_{in}) \in R^n_+$. Note that as $1 \in P$, $\sum_{j=1}^{n} a_{ij} \geq 1$ for all i. Clearly $P = \bigcap_{i=1}^{m} P_i$.

Proposition 13.1 *If P is an upper monotone polytope in the form (13.1), $P^+ = \bigcap_{i=1}^{n} P_i^+$.*

Proof If $z \in P^+$, $z \geq x$ for some $x \in P$. Then $x \in P_i$ and $x \leq z$, so $z \in P_i^+$ for all i.

Conversely, if $z \in \bigcap_{i=1}^m P_i^+$, $z \geq x^i$ with $x^i \in P_i$ for $i = 1, \ldots, m$. Let $x^* = \max_{i=1,\ldots,m} x^i$, where max is to be taken coordinatewise. Clearly $x^* \in [0, 1]^n$. Also $a^i x^* \geq a^i x^i \geq 1$ for all i. So finally $x^* \in P$ and $z \geq x^*$, and so $z \in P^+$. □

Note that if $P = \bigcap_{i=1}^m P_i$ is not upper monotone, then $P^+ \subseteq \bigcap_{i=1}^m P_i^+$, but the converse is not always true.

Corollary 13.2 *If P is an upper monotone polytope in the form (13.1), $\alpha_P = \min_{i=1,\ldots,m} \alpha_{P_i}$.*

Proof From the definition, $\alpha_{P_i} = \min\{\pi x^* : \pi x \geq 1 \forall x \in P_i, \pi \geq 0\}$. Let $\min_i \alpha_{P_i} = \alpha_{P_k} = \pi^k x^*$ with $\pi^k \in P_k^*$. As $P \subseteq P_k$, $P_k^* \subseteq P^*$ and thus $\alpha_P \leq \pi^k x^* = \min_i \alpha_{P_i}$.

Conversely using observation 4 above, $\alpha_{P_i} = \min_{r \in I_i} \pi^r x^*$ where $\{x \in R_+^n : \pi^r x \geq 1, r \in I_i\}$ is a minimal description of P_i^+. As $P^+ = \bigcap_{i=1}^m P_i^+$, $I \subseteq \bigcup_{i=1}^m I_i$, and thus $\min_i \alpha_{P_i} = \min_{r \in \bigcup_i I_i} \pi^r x^* \leq \min_{i \in I} \pi^r x^* = \alpha_P$. □

Having established the relationship between P and α_P on the one hand, and P_i and α_{P_i}, $i = 1, \ldots, m$ on the other, from now on we assume that P is defined by a single inequality, i.e.

$$P = \{x \in [0, 1]^n : ax \geq 1\} \text{ where } a \in R_+^n. \tag{13.2}$$

In the following we use the notation $a(S) = \sum_{j \in S} a_j$ for any $S \subseteq N$.

Theorem 13.3 *If P is upper monotone of the form (13.2), $P^+ = \{x \in R_+^n : \sum_{j \in S} \frac{a_j x_j}{1 - a(N \setminus S)} \geq 1$ for all $S \subseteq N$ such that $1 - a(N \setminus S) > 0\}$.*

Proof $P^+ = \{z \in R_+^n :$ there exists x such that $x \leq z, ax \geq 1, 0 \leq x \leq 1\}$. Combining constraints $z_j \geq x_j$ with weights a_j for $j \in S$, $ax \geq 1$ with weight 1 and $-x_j \geq -1$ with weights a_j for $j \in N \setminus S$ shows that $\sum_{j \in S} a_j z_j \geq 1 - a(N \setminus S)$ is satisfied for $z \in P^+$ for all $S \subseteq N$. Conversely if $z \in R_+^n$ satisfies all the inequalities, let $S = \{j \in N : z_j < 1\}$, and define x^* by $x_j^* = z_j$ for $j \in S$ and $x_j^* = 1$ for $j \in N \setminus S$. Clearly $z \geq x^*$, $0 \leq x^* \leq 1$ so if we can show that $x^* \in P$, it follows that $z \in P^+$. If $a(N \setminus S) \geq 1$, then $ax^* = \sum_{j \in S} a_j x_j^* + a(N \setminus S) \geq 0 + 1 = 1$, and $x^* \in P$. Otherwise $\sum_{j \in S} \frac{a_j z_j}{1 - a(N \setminus S)} = \sum_{j \in S} \frac{a_j x_j^*}{1 - a(N \setminus S)} \geq 1$, so again $ax^* = \sum_{j \in S} a_j x_j^* + a(N \setminus S) \geq 1$, and $x^* \in P$. □

We now address the separation problem for (P, x^*): Given $x^* \in [0, 1]^n$, $x^* \notin P$, find α_P (see Definition 2); where from Theorem 13.3,

$$\alpha_P = \min_{\mathcal{F}} g(\alpha),$$

with

$$g(\alpha) = \frac{\sum_{j \in S} a_j x_j^*}{1 - a(N \setminus S)}$$

and $\mathcal{F} = \{S \subseteq N : a(N \setminus S) < 1\}$. For $\alpha \geq 0$, let $S(\alpha) = \{j \in N : x_j^* \leq al\}$.

Note first that $\alpha_P = 0$ if and only if $a(N \setminus S(0)) < 1$. Furthermore, since $\alpha_P = \min_r\{\pi^r x^*\}$, (see observations 3, 4 above), it follows that $\alpha_P = \max\{\alpha : \pi^r x^* \geq \alpha$ for all $r\}$.

From the above definitions and remarks one can deduce the following:

Lemma 13.4 *For P upper monotone of the form (13.2), and $\alpha > 0$, the following are equivalent:*

(i) $\alpha_P \geq \alpha$
(ii) $\min_r \pi^r(\frac{x^*}{\alpha}) \geq 1$
(iii) $\frac{x^*}{\alpha} \in P^+$
(iv) $(\min[\frac{x_j^*}{\alpha}, 1], \ldots, \min[\frac{x_n^*}{\alpha}, 1]) \in P$
(v) $\frac{1}{\alpha} \sum_{j \in S(\alpha)} a_j x_j^* + a(N \setminus S(\alpha)) \geq 1$
(vi) $f(\alpha) \equiv \sum_{j \in S(\alpha)} a_j x_j^* + \alpha \cdot a(N \setminus S(\alpha)) \geq \alpha$
(vii) *either* $a(N \setminus S(\alpha)) \geq 1$, *or* $a(N \setminus S(\alpha)) < 1$ *and* $g(\alpha) \geq \alpha$.

When $\alpha = \alpha_P$, each of the inequalities (i), (ii), (v) and (vii) in the Lemma must hold as an equality, and the points in (iii) and (iv) lie on the boundaries of their respective sets. It follows that $S(\alpha_P) = S(x_j^*)$ for some j such that $a(N \setminus S(x_j^*)) < 1$. Thus it suffices to find the largest x_j^* for which $\frac{x^*}{x_j^*} \in P^+$, or equivalently $g(x_j^*) \geq x_j^*$. Call this value x_q^*, and set $\alpha_P = g(x_q^*)$.

Theorem 13.5 *Suppose that P is upper monotone of the form (13.2). Let $x_q^* = \max\{x_j^* : a(N \setminus S(x_j^*)) < 1$ and $x_j^* \leq g(x_j^*)$. Then $S(\alpha_P) = S(x_q^*)$ and*

$$\alpha_P = \frac{\sum_{j \in S(x_q^*)} a_j x_j^*}{1 - a(N \setminus S(x_q^*))}.$$

Example 1 Let $P = \{x \in [0, 1]^4 : 3x_1 + 2x_2 + 0.5x_3 + 0.25x_4 \geq 1\}$ and $x^* = (\frac{1}{30}, \frac{1}{20}, \frac{1}{5}, \frac{4}{5})$. Using Theorem 13.5,

$$g(x_4^*) = \sum_{j=1}^{4} a_j x_j^* = 0.5 < x_4^* = 0.8$$

so $\alpha_P < x_4^*$.

$$g(x_3^*) = \frac{\sum_{j=1}^{3} a_j x_j^*}{1 - a_4} = \frac{0.3}{1 - 0.25} \geq x_3^* = 0.2.$$

As x_3^* is the largest coordinate value less than x_4^*, $x_q^* = x_3^*$, $S(x_3^*) = \{1, 2, 3\}$, and a most violated inequality is

$$4x_1 + \frac{8}{3}x_2 + \frac{2}{3}x_3 \geq 1$$

with $\alpha_P = 0.4$.

Corollary 13.6 *(Calculation of α_P). α_P can be calculated in $O(n)$ time.*

Proof Finding the median value $\alpha = x_r^*$ of $\{x_1^*, \ldots, x_n^*\}$ and then calculating $S(\alpha)$ and $f(\alpha)$ can be carried out in $O(n)$. If $f(\alpha) \geq \alpha$ (or $a(N \setminus S(\alpha)) \geq 1$ or $g(\alpha) \geq \alpha$), $\alpha_P \geq \alpha$ and $q \in (N \setminus S(\alpha)) \cup \{r\}$. Otherwise $q \in S(\alpha) \setminus \{r\}$.

So if $T(n)$ is the complexity of finding x_q^* and $S(x_q^*)$ in a list of length n, we have that $T(n) = cn + T(n/2)$ for some constant $c > 0$, and thus $T(n)$ is $O(n)$. □

Note that whenever $\max_{j \in N} x_j^* \leq \sum_{j \in N} a_j x_j^*$, we have $\alpha_P = \sum_{j \in N} a_j x_j^*$. In particular, this is the case if $x_i^* = x_j^*$ for all $i, j \in N$.

13.2 The Dominant and Upper Separation for Polytopes in $[0, 1]^n$

Here we suppose that $P \subset [0, 1]^n$, and that the polytope P is not necessarily upper monotone. Again we look for a minimal inequality description of P^+. As $P^+ = R_+^n$ if $0 \in P$, we suppose that $0 \notin P$. We will need projections. As usual, given a subset $S \subseteq N$, $P^S = \text{Proj}_{x^S}(P) = \{x^S : \text{there exists } x^{N \setminus S} \text{ with } (x^S, x^{N \setminus S}) \in P\}$.

Note that if $0 \in P^S$, there is no valid inequality for P^S of the form $\pi x \geq 1$ with $\pi > 0$, so $(P^S)^* = \emptyset$.

Theorem 13.7 *Let I^S be the set of valid inequalities of P^S of the form $\sum_{j \in S} \pi_j x_j \geq 1$ with $\pi_j > 0$ for all $j \in S$ that are satisfied at equality by $|S|$ linearly independent points of P^S. Then*

$$P^+ = \{x \in R_+^n : \pi^r x \geq 1 \textit{ for all } r \in \bigcup_S I^S\},$$

and each of these inequalities is facet-defining for P^+.

Proof Consider an inequality $\sum_{j \in S} \pi_j^r x_j \geq 1$ for $r \in I^S$. It is easily shown to be valid for P^+. By definition of I^S, there exist $|S|$ linearly independent points $x_S^1, \ldots, x_S^{|S|}$ of P^S with $\pi^r x_S^i = 1$ for $i = 1, \ldots, |S|$. As P^S is a projection, there exist points $x_{N \setminus S}^1, \ldots, x_{N \setminus S}^{|S|}$ such that $(x_S^i, x_{N \setminus S}^i) \in P$ for $i = 1, \ldots, |S|$. Now consider the points $(x_S^i, x_{N \setminus S}^i \in P$ for $i = 1, \ldots, |S|$ and $(x_S^1, x_{N \setminus S}^1) + e_j$ for $j \in N \setminus S$. These n points lie in P^+, are linearly independent and satisfy $\pi^r x = 1$. So the inequality is facet-defining for P^+.

Conversely let $\pi x \geq 1$ define a facet of P^+. Necessarily $\pi \geq 0$. Let S be the support of π. Suppose that $\pi x \geq 1$ is satisfied at equality by a maximum of $r < |S|$ linearly independent extreme points $x^1, \ldots, x^r$ of P^S. Then $\pi x > 1$ for all other extreme points of P^S, and there exists a vector $\alpha \in R^{|S|}$ with $\alpha \neq 0$ such that $\alpha x^k = 0$ for $k = 1, \ldots, r$. Now there exists an $\epsilon > 0$ such that, taking $\pi^1 = \pi + \epsilon\alpha$ and $\pi^2 = \pi - \epsilon\alpha > 0$, $\pi^i x \geq 1$ is valid for P^S and thus for P^+ for $i = 1, 2$. But as $(\pi, 1) = \frac{1}{2}(\pi^1, 1) + \frac{1}{2}(\pi^2, 1)$, $\pi x \geq 1$ is not a facet of P^+, a contradiction. □

Note that membership in I^S is equivalent to saying that $\pi x \geq 1$ defines a facet of P^S of dimension $|S| - 1$, or more explicitly either $\dim(P^S) = |S| - 1$ and $P^S \subseteq \{x \in R^{|S|} : \pi x = 1\}$, or $\dim(P^S) = |S|$ and $\pi x \geq 1$ defines a facet of P^S.

Corollary 13.8 *Every inequality defining a facet of P^+ has at most* $\dim(P) + 1$ *nonzero coefficients.*

Now we consider the upper separation problem for (P, x^*). As this is equivalent to separation with respect to P^+, the result is immediate.

Proposition 13.9 *For $x^* \in R^n_+$, $\alpha_P = \min_{S \subseteq N} \min_{r \in I^S} \pi^r x^*$.*

In the absence of monotonicity no efficient procedure is known for finding α_P.

13.2.1 Unions of Polytopes in Disjoint Spaces

Here we model the disjunction of two polytopes in the unit cubes of different spaces. Given $P \subseteq [0, 1]^m$ and $Q \subseteq [0, 1]^n$, we study sets of the form

$$
\begin{aligned}
Z &= \{(x, y) \in [0, 1]^m \times [0, 1]^n : x \in P \text{ or } y \in Q\} \\
&= (P \times [0, 1]^n) \cup ([0, 1]^m \times Q).
\end{aligned}
$$

As before, P^* denotes the blocker of P.

Theorem 13.10 $\operatorname{conv}(Z)^* = P^* \times Q^*$.

Proof Suppose $(\pi, \mu) \in \operatorname{conv}(Z)^*$. As $(x, 0) \in Z$ for all $x \in P$, $\pi x + \mu 0 \geq 1$ for all $x \in P$, so $\pi \in P^*$. Similarly $\mu \in Q^*$.

Conversely suppose that $\pi \in P^*$ and $\mu \in Q^*$. If $(x, y) \in \operatorname{conv}(Z)$,

$$(x, y) = \lambda(x^1, y^1) + (1 - \lambda)(x^2, y^2)$$

with $0 \leq \lambda \leq 1$, $x^1 \in P$, $x^2 \in [0, 1]^m$, $y^1 \in [0, 1]^n$, $y^2 \in Q$. So $\pi x + \mu y = \lambda\pi x^1 + (1 - \lambda)\pi x^2 + \lambda\mu y^1 + (1 - \lambda)\mu y^2 \geq \lambda + 0 + 0 + (1 - \lambda) = 1$, and the inequality is valid for $\operatorname{conv}(Z)$. □

Combined with Theorem 13.7, this yields a characterization of all facets with nonnegative coefficients. Let $M = \{1, \ldots, m\}$ and $N = \{1, \ldots, n\}$.

Corollary 13.11 *Given vectors $\pi \in R^m_+$ with support $S \subseteq M$ and $\mu \in R^n_+$ with support $T \subseteq N$, $\pi x + \mu y \geq 1$ is facet-defining for* conv(Z) *if and only if $\pi x \geq 1$ defines a face of P^S of dimension $|S| - 1$ and $\mu y \geq 1$ defines a face of Q^T of dimension $|T| - 1$.*

Now consider the upper separation of (x^*, y^*) wrt the dominant of conv(Z). Let $\alpha_P = \min\{\pi x^* : \pi \in P^*\}$ as above, and let $\alpha_Q = \min\{\mu y^* : \mu \in Q^*\}$.

Proposition 13.12 *If $(x^*, y^*) \in R^m_+ \times R^n_+$, $(x^*, y^*) \in$ conv$(Z)^+$ if and only if $\alpha_P + \alpha_Q \geq 1$.*

Proof Let $\pi^* = \arg\min\{\pi x^* : \pi P^*\}$, and $\mu^* = \arg\min\{\mu y^* : \mu \in Q^*\}$. By Theorem 13.10, $\pi^* x + \mu^* y \geq 1$ is valid for conv$(Z)^+$.

Suppose $\alpha_P + \alpha_Q \geq 1$, and consider any valid inequality $\pi x + \mu y \geq 1$ for conv$(Z)^+$. Now $\pi x^* + \mu y^* \geq \pi^* x^* + \mu^* y^* = \alpha_P + \alpha_Q \geq 1$ and thus $(x^*, y^*) \in$ conv$(Z)^+$. Conversely if $\alpha_P + \alpha_Q < 1$, then $\pi^* x^* + \mu^* y^* < 1$, and $(x^*, y^*) \notin$ conv$(Z)^+$. □

13.3 The Upper Monotone Case

When P and Q are upper monotone polytopes, conv(Z) is upper monotone, and the results of the previous subsection can be made more precise. Combining Corollary 13.11 with Theorem 13.3 gives

Theorem 13.13 *Let $P \subseteq [0, 1]^m$ and $Q \subseteq [0, 1]^n$ be upper monotone polytopes with*

$$P = \{x \in [0, 1]^m : Ax \geq 1\}, \quad Q = \{y \in [0, 1]^n : By \geq 1\},$$

where $A \geq 0$ and $B \geq 0$ are $p \times m$ and $q \times n$ respectively. Then

$$\begin{aligned} \text{conv}(Z) = \{(x, y) \in [0, 1]^m \times [0, 1]^n :& \\ & \sum_{j \in S} \frac{a_{ij}}{1 - \sum_{h \in M \setminus S} a_{ih}} x_j + \sum_{\ell \in T} \frac{b_{k\ell}}{1 - \sum_{h \in N \setminus T} b_{kh}} y_\ell \geq 1 \textit{ for all } S \subseteq M, T \subseteq N \\ \textit{with} \quad & \sum_{h \in M \setminus S} a_{ih} < 1 \textit{ and } \sum_{h \in N \setminus T} b_{kh} < 1, \quad i \in \{1, \dots, p\} \textit{ and } k \in \{1, \dots, q\}. \end{aligned}$$

The separation result is also more specific.

Proposition 13.14 *If P and Q are upper monotone and $(x^*, y^*) \in [0, 1]^m \times [0, 1]^n$, then $(x^*, y^*) \in$ conv(Z) if and only if $\alpha_P + \alpha_Q \geq 1$.*

A polynomial separation algorithm for conv(Z) follows immediately from Corollaries 13.2 and 13.6. Its running time is $O(pm) + O(qn)$.

13.3.1 conv(Z): The General Case

So far we have only considered valid inequalities with nonnegative coefficients. However any valid inequality $\sum_{j \in N} \pi_j x_j \geq \pi_0$ for $P \subseteq [0,1]^n$ with $\pi_j \geq 0$ for $j \in N^+$ and $\pi_j < 0$ for $j \in N^-$ can be written as

$$\sum_{j \in N^+} \pi_j x_j - \sum_{j \in N^-} \pi_j (1 - x_j) \geq \pi_0 - \sum_{j \in N^-} \pi_j.$$

If it is *nontrivial* (i.e. cutting off some part of $[0,1]^n$), then $\pi_0 - \sum_{j \in N^-} \pi_j > 0$ and the inequality can be rewritten in the form

$$\sum_{j \in N^+} \tilde{\pi}_j x_j + \sum_{j \in N^-} \tilde{\pi}_j (1 - x_j) \geq 1,$$

where $\tilde{\pi}_j = \pi_j / (\pi_0 - \sum_{j \in N^-} \pi_j) > 0$ for $j \in N^+$ and $\tilde{\pi}_j = -\pi_j / (\pi_0 - \sum_{j \in N^-} \pi_j) > 0$ for $j \in N^-$.

Now for any partition (N_1, N_2) of N, all valid inequalities with $\pi_j \geq 0$ for $j \in N_1$ and $\pi_j \leq 0$ for $j \in N_2$ can be characterized exactly as above by changing to new variables $\tilde{x}_j = x_j$ for $j \in N_1$ and $\tilde{x}_j = 1 - x_j$ for $j \in N_2$. So Corollary 13.11 and the observation that the polytope P is the intersection of its dominants with respect to the 2^n orthants leads to a complete characterization of $\mathrm{conv}(Z)$.

Theorem 13.15 *Let* $\left\{\sum_{j \in S_r^+} \pi_j^r x_j + \sum_{j \in S_r^-} \pi_j^r (1 - x_j) \geq 1 : r \in V^S\right\}$ *for all* $S \subseteq M$ *and* $r \in V^S$ *denote the valid inequalities of* P^S *defining faces of dimension* $|S| - 1$ *of full support (i.e.* $\pi_j^r > 0$ *for* $j \in S$ *and* (S_r^+, S_r^-) *is a partition of* S*), and define* μ^S, T_S^+, T_S^- *similarly for* Q^T *for all* $T \subseteq N$ *and* $S \in V^T$*. Then*

$$\mathrm{conv}(Z) = \Bigg\{ (x, y) \in [0,1]^m \times [0,1]^n : \sum_{j \in S_r^+} \pi_j^r x_j + \sum_{j \in S_r^-} \pi_j^r (1 - x_j)$$
$$+ \sum_{j \in T_s^1} \mu_j^s y_j + \sum_{j \in T_s^-} \mu_j^s (1 - y_j) \geq 1 \; for \; r \in \bigcup_{S \subseteq M} V^S, s \in \bigcup_{T \subseteq N} V^T \Bigg\}.$$

Example 2 Consider two polytopes in the unit square,

$$P = \{x \in [0,1]^2 : -x_1 + 2x_2 \leq \tfrac{3}{2}, -x_1 + 2x_2 \geq 1\} \text{ and}$$
$$Q = \{y \in [0,1]^2 : y_1 + y_2 \leq 1, y_1 \geq \tfrac{1}{2}, y_2 \geq \tfrac{1}{4}\}.$$

$P = P^{\{1,2\}}$ can be rewritten in the canonical form

$$P = \{x \in [0,1]^2 : 2x_1 + 4(1 - x_2) \geq 1, \tfrac{1}{2}(1 - x_1) + x_2 \geq 1\}.$$

Looking at the projections (typically hard to calculate), $P^{\{1\}} = [0,1]$ and $P^{\{2\}} = \{x_2 \in [0,1] : 2x_2 \geq 1\}$. For Q we obtain

$$\begin{aligned} Q^{\{1,2\}} =&\{y \in [0,1]^2 : (1 - y_1) + (1 - y_2) \geq 1, 2y_1 \geq 1, 4y_2 \geq 1\}, \\ Q^{\{1\}} =&\{y_1 \in [0,1] : 2y_1 \geq 1, 4(1 - y_1) \geq 1\} \text{ and} \\ Q^{\{2\}} =&\{y_2 \in [0,1] : 4y_2 \geq 1, 2(1 - y_2) \geq 1\}. \end{aligned}$$

Combining the second facet $\frac{1}{2}(1 - x_1) + x_2 \geq 1$ of $P^{\{1,2\}}$ with the second facet $4(1 - y_1) \geq 1$ of $Q^{\{1\}}$ gives a facet

$$\tfrac{1}{2}(1 - x_1) + x_2 + 4(1 - y_1) \geq 1$$

of $\text{conv}(Z)$.

The three facets arising from P and its projections combined with the five facets arising from Q and its projections give in total fifteen facets for conv$Z$, other than the simple bound constraints arising from the unit cubes containing P and Q.

13.4 Application 1: Monotone Set Functions and Matroids

Consider the polytope in $[0,1]^n$

$$P(r) = \{x \in [0,1]^n : x(A) \leq r(A) \text{ for all } A \subseteq N\},$$

where the set function $r : 2^N \to R$ satisfies the following conditions:

1. $r(\emptyset) = 0$;
2. $r(A) \leq |A|$ for all $A \subset N$;
3. $r(A) \leq r(B)$ for $A, B \subseteq N$, $A \subseteq B$ (nondecreasing).

Here we will derive the convex hull of the union of two such polytopes in disjoint spaces

$$Z(r^1, r^2) = \{(x, y) \in [0,1]^m \times [0,1]^n : x \in P(r^1) \text{ or } y \in P(r^2)\}.$$

First we consider the *complement* $C(r)$ of a single monotone polytope $P(r)$, namely

$$C(r) = \{w \in [0,1]^n : e - w \in P(r)\},$$

where $e = (1, \dots, 1)^T$. Clearly $C(r)$ is an (upper)-monotone polytope in $[0,1]^n$. By substitution

$$\begin{aligned} C(r) &= \{w \in [0,1]^n : w(A) \geq |A| - r(A) \text{ for all } A \subseteq N\} \\ &= \{w \in [0,1]^n : \frac{w(A)}{|A| - r(A)} \geq 1 \text{ for all } A \subseteq N \text{ with } r(A) < |A|\}. \end{aligned}$$

Note that

$$C(r)^+ = \{w \in R^n_+ : \frac{w(A)}{|A| - r(A)} \geq 1 \text{ for all } A \subseteq N \text{ with } r(A) < |A|\}$$

because the inequality $\frac{w(A \setminus \{j\})+1}{|A|-r(A)} \geq 1$ is dominated by the inequality $w(A \setminus \{j\}) \geq |A \setminus \{j\}| - r(A \setminus \{j\})$ as $r(A \setminus \{j\}) \leq r(A)$ (see Theorem 13.3). Now from Theorem 13.13, conv$\{(C(r^1) \times [0,1]^n) \cup ([0,1]^m \times C(r^2))\}$

$$\begin{aligned} = \{(\bar{x}, \bar{y}) \in & [0,1]^m \times [0,1]^n : \frac{\bar{x}(A)}{|A| - r_1(A)} + \frac{\bar{y}(B)}{|B| - r_2(B)} \geq 1 \\ & \text{for all } A \subseteq M, B \subseteq N, r_1(A) < |A|, r_2(B) < |B|\}, \end{aligned}$$

giving after complementation

Proposition 13.16 *conv*$(Z(r^1, r^2)) = \{(x, y) \in [0,1]^m \times [0,1]^n : \frac{|A|-x(A)}{|A|-r_1(A)} + \frac{|B|-y(B)}{|B|-r_2(B)} \geq 1$ *for all* $A \subseteq M,\ B \subseteq N,\ r_1(A) < |A|,\ r_2(B) < |B|\}$.

For matroid rank functions r_1, r_2 (satisfying 1–3, integer-valued and submodular), this characterization has been obtained earlier in [58] using a different proof technique. The present result is more general. For example, it holds if each r_i is a rank function of an intersection of two matroids.

Note also that when r_i for $i = 1, 2$ are matroid rank functions, it is known that the separation problem for $C(r_1)$ is the problem of finding a maximal fractional packing of bases of the dual matroid into the vector x^*. It follows that $\alpha_{C(r_1)}$ can be calculated in polynomial time even though there are an exponential number of inequalities. Edmonds and Fulkerson treated this problem in [72]. More generally the separation problem for a matroid polytope has been treated by Cunningham [65, 66] and it is a special case of submodular function minimization [85, 115].

13.5 Application 2: Logical Inference

In [84], Hong and Hooker gave a tight integer programming formulation to logical conditions of the form "if less than k propositions of S are true, then at least ℓ proposition of T are true," or in terms of 0-1 variables,

$$\sum_{j \in S} x_j < k \Rightarrow \sum_{j \in T} y_j \geq \ell.$$

Here we derive their result using the toolkit developed in this chapter, and give an efficient separation procedure for the family of inequalities defining the convex hull. Further, we extend these results to more general logical conditions.

For $(x, y) \in \{0, 1\}^m \times \{0, 1\}^n$, and positive integers k, ℓ, the condition "$\sum_{i=1}^m x_i \leq k - 1$ implies $\sum_{j=1}^n y_j \geq 1$" is equivalent to $(x, y) \in Z$ where

$$Z = \{(x, y) \in \{0, 1\}^m \times \{0, 1\}^n : \sum_{i=1}^{m} x_i \geq k \;\; or \;\; \sum_{j=1}^{n} y_j \geq 1\}.$$

So we have the condition restated as the union of two polytopes in different spaces. Furthermore, the two polytopes are of a very special kind. To characterize $\mathrm{conv}(Z)$, we need to characterize sets of the form $X = \{x \in \{0, 1\}^n : \sum_{j=1}^n x_j \geq k\}$, or more specifically the polytope $P = \mathrm{conv}(X)$. Specifically we examine in turn the dominant and separation problem first for a single such polytope and then for the union of two such polytopes.

The Dominant of P It is easily seen directly, or else by using Theorem 13.3 that

$$P^+ = \{x \in R^n_+ : \sum_{j \in S} x_j \geq k - (n - s) \text{ for all } S \subseteq N \text{ with } s = |S| > n - k\},$$

where $N = \{1, \ldots, n\}$.

Separation for P Given $x^* \in [0, 1]^n$, suppose that $j_1, \ldots, j_n$ is a reordering of $\{1, \ldots, n\}$ such that $x^*_{j_1} \leq \cdots \leq x^*_{j_n}$. Then using Theorem 13.5

$$\alpha_P = \min_{t=1,\ldots,k} \frac{x^*_{j_1} + \cdots + x^*_{j_{n-1}+1}}{k - t + 1},$$

where the minimum is attained for the smallest $t \geq 1$ such that $(k - t)x^*_{j_{n-t}+1} \leq x^*_{j_1} + \cdots + x^*_{j_{n-t}}$.

The Convex Hull of Z Now by Theorem 13.13, it follows that conv(Z) is described by

$$\sum_{j\in S}\frac{x_j}{k-(m-s)}+\sum_{j\in T}\frac{y_j}{\ell-(n-t)}\geq 1$$
$$\forall S\subseteq M, T\subseteq N \text{ with } s=|S|>m-k, t=|T|>n-1$$
$$0\leq x\leq 1, 0\leq y\leq 1,$$

which is the result of Hong and Hooker [84].

Separation for conv(Z) Using Corollary 13.6, there is a linear time separation algorithm. See [15] for its derivation and discussion.

As a special case, consider the set

$$Z=\{w\in\{0,1\}^{1+\ell}: kw_0+\sum_{j=1}^{\ell}w_j\leq\ell\}$$

with $k\leq\ell$ analyzed by Padberg [104] in his study of $(1,k)$-configurations. This can be rewritten as

$$\{w\in\{0,1\}^{1+\ell}: w_0=0 \text{ or } \sum_{j=1}^{\ell}w_j\leq\ell-k\},$$

or

$$\{\bar{w}\in\{0,1\}^{1+\ell}: \bar{w}_0=1 \text{ or } \sum_{j=1}^{\ell}\bar{w}_j\geq k\}.$$

Now as a special case of logical inference, we obtain the characterization of Padberg

$$\begin{aligned}\text{conv}(Z)&=\left\{\bar{w}\in[0,1]^{1+\ell}:\bar{w}_0+\sum_{j\in T}\frac{\bar{w}_j}{k-(\ell-t)}\geq 1\forall T\subseteq\{1,\ldots,\ell\},|T|>\ell-k\right\}\\&=\left\{w\in[0,1]^{1+\ell}:[k-(\ell-t)]w_0+\sum_{j\in T}w_j\right.\\&\leq\left.\quad t\ \forall\ T\subseteq\{1,\ldots,\ell\},|T|>\ell-k\right\}.\end{aligned}$$

13.6 More Complex Logical Constraints

The convex hull representation and associated separation procedure for cardinality rules can be extended to more complex logical constraints. To be more precise, any logical constraint of the form $A \to B$, where either A or B or both can be of the form

$$\left(x_1^1, \ldots, x_{m_1}^1\right)_{k_1} \wedge \cdots \wedge \left(x_1^q, \ldots, x_{m_q}^q\right)_{k_q}$$

or of the form

$$\left(x_1^1, \ldots, x_{m_1}^1\right)_{k_1} \vee \cdots \vee \left(x_1^q, \ldots, x_{m_q}^q\right)_{k_q},$$

is amenable to a similar representation. It is assumed that all the variables are distinct.

In each of the four possible cases, $A \to B$ is equivalent to an expression of the form

$$\bigcup_{i=1}^{p} \bigcap_{j=1}^{q_i} P^{ij},$$

where P^{ij} are polytopes in the unit cube defined by single inequalities of the form $\sum_{\ell=1}^{m_{ij}} x_\ell^{ij}(N^{ij}) \geq k_{ij}$, denoted $x^{ij}(N_{ij}) \geq k_{ij}$.

Since the intersection of two polytopes is a polytope, for each $i = 1, \ldots, p$, $\bigcap_{j=1}^{q_i} P^{ij}$ can be written as

$$P^i = \{(x^{i1}, \ldots, x^{iq_i}) \in [0,1]^{|N_{i1}|+\cdots+|N_{iq_i}|} : x^{i1}(N_{i1}) \geq k_{i1}, \ldots, x^{iq_i}(N_{iq_i}) \geq k_{iq_i}\}.$$

To avoid cumbersome notation, instead of describing the convex hull of $\bigcup_{i=1}^{p} P^i$, we will restrict membership in the union to $p = 2$, i.e. will describe the convex hull of $P^1 \cup P^2$. The extension to $p > 2$ is straightforward.

Theorem 13.17 *Consider the two polytopes*

$$P^1 = \{(x^1, \ldots, x^r; y^1, \ldots, y^s) \in [0,1]^{|M_1|+\cdots+|M_r|+|N_1|+\cdots+|N_s|} : x^i(M_i) \geq p_i, i = 1, \ldots, r\}$$

and

$$P^2 = \{(x^1, \ldots, x^r; y^1, \ldots, y^s) \in [0,1]^{|M_1|+\cdots+|M_r|+|N_1|+\cdots+|N_s|} : y^j(N_j) \geq \ell_j, j = 1, \ldots, s\}$$

Then $\operatorname{conv}(P^1 \cup P^2)$ *is defined by the set of inequalities* $0 \le x^i \le 1$, $i = 1, \ldots, r$, $0 \le y^j \le 1$, $j = 1, \ldots, s$, *and*

$$\frac{1}{|S_i| + p_i - |M_i|} x^i(S_i) + \frac{1}{|T_j| + \ell_j - |N_j|} y^j(T_j) \ge 1 \tag{13.3}$$

for all $S_i \subseteq M_i$, $|M_i| - p_i + 1 \le |S_i| \le |M_i|$, $i = 1, \ldots, r$, *and* $T_j \subset N_j$, $|N_j| - \ell_j + 1 \le |T_j| \le |N_j|$, $j = 1, \ldots, s$.

Furthermore, every inequality (13.3) such that $1 \le p_i \le |M_i| - 1$ *and* $1 \le \ell_j \le |N_i| - 1$ *defines a facet of* $\operatorname{conv}(P^1 \cup P^2)$.

The proof and an efficient separation procedure follow the same lines as in the case of 2 (see [15] for details). Note the surprising fact that although we are dealing with a system in $|M_1| + \cdots + |M_r| + |N_1| + \cdots + |N_s|$ variables, namely the r vectors x^i, $i = 1, \ldots, r$, and the s vectors y^j, $j = 1, \ldots, s$, every inequality (13.3) has nonzero coefficients only for one vector x^i and one vector y^j. This is so because all the rs vector pairs belong to disjoint subspaces.

13.7 Unions of Upper Monotone Polytopes in the Same Space

Some of the above results are also relevant to the case where the polytopes are in the same space. If $P = \{x \in [0,1]^n : Ax \ge 1\}$ and $Q = \{x \in [0,1]^n : Bx \ge 1\}$ are arbitrary polytopes in the same space, then using the characterization of [6], $\operatorname{conv}(P \cup Q)$ is given by the system

$$\begin{aligned}
\operatorname{conv}(P \cup Q) = \{x \in [0,1]^n : x = {} & y \qquad\quad + w \\
& Ay - 1y_0 \\
& -y + 1y_0 \\
& \qquad\qquad Bw - 1w_0 \ge 0 \\
& \qquad\qquad -w + w_0 = 1 \\
& \qquad\quad y_0 \qquad + w_0 = 1 \\
& \qquad\qquad y, y_0, w, w_0 \ge 0\}
\end{aligned}$$

Rewriting this expression in a more compact form, we get

$$\begin{aligned}
\operatorname{conv}(P \cup Q) &= \{x \in [0,1]^n : x = y \qquad + \; w \\
&\qquad\qquad y \in Py_0, \quad w \in Qw_0 \\
&\qquad\qquad\qquad y_0 + \qquad w_0 = 1 \\
&\qquad\qquad\qquad y_0\;, \qquad w_0 \ge 0\} \\
&= \{x \in [0,1]^n : x = y + w, (y, w) \in \operatorname{conv}(Z_0)\},
\end{aligned}$$

where $Z_0 = (P \times 0^n) \cup (0^n \times Q)$, and $P_{y_0} = \{xy_0 : x \in P\}$.

Defining $Z = (P \times [0,1]^n) \cup ([0,1]^n \times Q)$ as for Theorem 13.10, we have the following new higher dimensional representation of $\text{conv}(P \cup Q)$.

Theorem 13.18 *If P and Q are upper monotone polytopes in $[0,1]^n$, then*

$$\text{conv}(P \cup Q) = \{x \in [0,1]^n : x = y + w, (y, w) \in \text{conv}(Z)\}$$

Proof As $\text{conv}(Z_0) \subseteq \text{conv}(Z)$, $\text{conv}(P \cup Q) \subseteq \{x \in [0,1]^n : x = y + w, (y, w) \in \text{conv}(Z)\}$.

Conversely suppose $x = y + w$, with $(y, w) \in \text{conv}(Z)$. Then

$$\begin{aligned} (y, w) &= \lambda(y^1, w^1) + (1-\lambda)(y^2, w^2) \\ &\geq \lambda(y^1, 0) + (1-\lambda)(0, w^2) \\ &= (\lambda y^1, (1-\lambda) w^2) \end{aligned}$$

with $y^1 \in P$ and $w^2 \in Q$. So $x \geq \lambda y^1 + (1-\lambda) w^2 \in \text{conv}(P \cup Q)^+$. As all the sets are upper monotone, it follows that if $x \in [0,1]^n$, then $x \in \text{conv}(P \cup Q)$. $\square$

Using the characterization of $\text{conv}(Z)$ given in Theorem 13.13, we then have

Corollary 13.19 *Let P and Q be upward monotone polytopes in $[0,1]^n$ with*

$$P = \{x \in [0,1]^n : Ax \geq 1\}, Q = \{x \in [0,1]^n : Bx \geq 1\},$$

where A and B have p and q rows respectively. Then

$$\text{conv}(P \cup Q) = \Bigg\{ x \in [0,1]^n : x = y + w,$$
$$\sum_{j \in S} \frac{a_{ij}}{1 - \sum_{h \in N \setminus S} a_{ij}} y_j + \sum_{\ell \in T} \frac{b_{k\ell}}{\sum_{h \in N \setminus T} b_{kh}} w_\ell \textit{ for all } S, T \subseteq N$$
$$\textit{with } \sum_{h \in M \setminus S} a_{ih} < 1 \textit{ and } \sum_{h \in N \setminus T} b_{kh} < 1, i \in \{1, \ldots, p\} \textit{ and } k \in \{1, \ldots, q\} \Bigg\}.$$

Theorem 13.18 and Corollary 13.19 characterize the convex hull of the union of two upper monotone polytopes in the same space, whereas Theorem 13.13 does the same thing for the case when the two polytopes lie in disjoint spaces. But what about the case where the two polytopes overlap partly? The following theorem gives the more general result which covers all intermediate situations as well as the two extreme ones: it specializes to Theorem 13.13 when the two spaces are disjoint, and to Theorem 13.18/Corollary 13.19 when they are the same.

Let $P = \{y \in [0,1]^M : Ay \geq 1\}$ and $Q = \{w \in [0,1]^N : Bw \geq 1\}$, where A and B have p and q rows, respectively. As before, let $Z = (P \times [0,1]^N) \cup$

$([0,1]^M \times Q)$. Define $\tilde{P} = \{(y,0) \in [0,1]^M \times [0,1]^{N\setminus M} : y \in P\}$ and $\tilde{Q} = \{(0,w) \in [0,1]^{M\setminus N} \times [0,1]^N : w \in Q$. Note that $\tilde{P}$ and $\tilde{Q}$ are both in $[0,1]^{M\cup N}$.

Theorem 13.20 $\operatorname{conv}(\tilde{P} \cup \tilde{Q})$

$$= \{x{\in}[0,1]^{M\cup N} : x{=}(y^{M\cap N}, y^{M\setminus N}, 0^{N\setminus M}){+}(w^{M\cap N}, 0^{M\setminus N}, w^{N\setminus M}), (y,w){\in}conv(Z)\}$$

$$= \{x{\in}[0,1]^{M\cup N} : x_i{=}y_i + w_i, i \in M\cap N; x_i = y_i, i \in M \setminus N; x_i = w_i, i \in N\setminus M,\ and$$

$$\sum_{j\in S} \frac{a_{ij}}{1-\sum\limits_{h\in M\setminus S} a_{ih}} + \sum_{\ell\in T} \frac{b_{k\ell}}{1-\sum\limits_{h\in N\setminus T} b_{kh}} w_\ell \geq 1$$

$$for\ all\ s \subseteq M\ with\ 1-\sum_{h\in M\setminus S} a_{ih} > 0, T \subseteq N\ with\ 1-\sum_{h\in N\setminus T} b_{kh} > 0,$$

$$and\ all\ i \in \{1,\ldots,p\}, k \in \{1,\ldots,q\}\}$$

Now let r_1 and r_2 be set functions on $N = \{1,\ldots,n\}$ satisfying 1-3 in our Application 1, consider the polytopes $P(r_i) = \{x \in R^n_+ : \sum_{j\in A} x_j \leq r_i(A)$ for $A \subseteq N\}$ for $i = 1,2$. Without loss of generality we may assume by rescaling that the polytopes $P(r_1)$ and $P(r_2)$ are in $[0,1]^n$. As a corollary from Proposition 13.16 and Theorem 13.18, we have

Corollary 13.21 $\operatorname{conv}(P(r_1) \cup P(r_2)) = \{w \in [0,1]^n : w = x + y,\ \frac{|A|-x(A)}{|A|-r_1(A)} + \frac{|B|-y(B)}{|B|-r_2(B)} \geq 1$ *for all* $A \subseteq N,\ B \subseteq N,\ r_1(A) < |A|,\ r_2(B) < |B|\}$.

13.8 Unions of Polymatroids

Now we suppose that $r_i : 2^N \to R$ for $i = 1,2$ are polymatroid rank functions (satisfying 1 and 3 in Application 1 and submodular). We now obtain a description of $\operatorname{conv}(P(r_1) \cup P(r_2))$ in the space of the original variables. Let $\Pi = \{\pi \in R^n_+ : \pi x \leq 1$ for $x \in P(r_1) \cup P(r_2)\}$.

Proposition 13.22 $\Pi = \operatorname{Proj}_\pi\{\pi_j \leq \sum_{A:j\in A} u_A$ *for* $j \in N,\ \sum_A u_A r_i(A) \leq 1$ *for* $i = 1,2,\ \pi \geq 0,\ u_A \geq 0$ *for all* $A \subseteq N\}$.

Proof For fixed $\pi \in R^n_+$, let σ be a permutation such that $\pi_{\sigma(1)} \geq \cdots \geq \pi_{\sigma(n)} \geq 0$ and $A^\pi_j = \{\sigma(1),\ldots,\sigma(j)\}$ for $j = 1,\ldots,n$. Then from Edmonds [71],

$$\max\{\pi x : x \in P(r_i)\} = \min\left\{\sum_{A\subseteq N} u_A r_i(A) : \sum_{A:j\in A} u_A \geq \pi_j, j \in N, u_A \geq 0, A \subseteq N\right\}$$

$$= \sum_{j=1}^{n-1} (\pi_{\sigma(j)} - \pi_{\sigma(j+1)}) r_i(A^\pi_j) + \pi_{\sigma(n)} r_i(A^\pi_n)$$

with optimal dual solution $u_{A_n^\pi} = \pi_{\sigma(n)}, u_{A_j^\pi} = \pi_{\sigma(j)} - \pi_{\sigma(j+1)}$ for $j = 1, \ldots, n-1$ and $u_A = 0$ *independent* of i. It follows that $\pi \in \Pi$ if and only if

$$\{\pi \in R_+^n, u \in R_+^{\mathcal{P}(N)}, \pi_j \le \sum_{A:j \in A} u_A \text{ for } j \in N, \sum_A u_A r_i(A) \le 1 \text{ for } i = 1, 2\} \neq \emptyset.$$

□

The next result is easily checked

Proposition 13.23 *If π is an extreme point of Π, then $\pi_j = \sum_{A:j \in A} u_A$ for $j \in N$ with u extreme in $U = \{u \ge 0 : \sum_A u_a r_i(A) \le 1$ for $i = 1, 2\}$.*

Proof As the polytope is monotone, for given u, an inequality $\sum_{j \in N} \pi_j x_j \le 1$ with $\pi_j < \sum_{A:j \in A} u_A$ for some j is a convex combination of $\sum_j (\sum_{A:j \in A} u_A) x_j \le 1$ and $-x_j \le 0$ for $j \in N$. Alternatively if $u = \frac{1}{2}u^1 + \frac{1}{2}u^2$ with $u^1, u^2 \in U$, set $\pi_j^t = \sum_{A:j \in A} u_A^t$ for $j \in N$ and $t = 1, 2$. Now $\pi^t \in \Pi$ for $t = 1, 2$ and $\pi = \frac{1}{2}\pi^1 + \frac{1}{2}\pi^2 \in \Pi$, so π is not extreme. □

Theorem 13.24 $\text{conv}(P(r_1) \cup P(r_2)) = \{x \in R_+^n :$

$$x(A) \le \max\{r_1(A), r_2(A)\} \forall A \subseteq N$$

$$\frac{r_2(B) - r_1(B)}{r_1(A)r_2(B) - r_1(B)r_2(A)} x(A) + \frac{r_1(A) - r_2(A)}{r_1(A)r_2(B) - r_1(B)r_2(A)} x(B) \le 1$$

$$\forall A, B \subseteq N \textit{ with } (r_1(A) - r_2(A))(r_1(B) - r_2(B)) < 0\}.$$

Proof From Propositions 13.22 and 13.23, it suffices to consider the extreme points of U. In a basic feasible solution of U at most two terms u_A, u_B are nonzero. If there is only one, and $u_A > 0$, then $u_A = \frac{1}{\max\{r_1(A), r_2(A)\}}$, $u_T = 0 \; \forall T \neq A$ giving the inequality

$$x(A) \le \max\{r_1(A), r_2(A)\} \text{ for all } A \subseteq N$$

If both are nonzero, we have $u_A r_1(A) + u_B r_1(B) = 1$, $u_A r_2(B) + u_B r_2(B) = 1$. This has a solution

$$u_A = \frac{r_2(B) - r_1(B)}{r_1(A)r_2(B) - r_2(A)r_1(B)} > 0, u_B = \frac{r_1(A) - r_2(A)}{r_1(A)r_2(B) - r_2(A)r_1(B)} > 0$$

if and only if $(r_1(A) - r_2(A))(r_1(B) - r_2(B)) < 0$. □

This characterization generalizes a result from [58] on the disjunction of matroid polyhedra to the case of polymatroids.

References

1. K. Andersen, G. Cornuéjols, and Y. Li., Split closure and intersection cuts. *Math. Prog. A, 102*, 2005, 451–493.
2. K. Andersen, Q. Louveaux, R. Weismantel, R., L.A. Wolsey, Cutting planes from two rows of a simplex tableau. IPCO 12, *Lecture Notes in Computer Science*, 4513, Springer, 2007, 1–15.
3. E. Balas, Machine sequencing via disjunctive graphs: an implicit enumeration algorithm. *Operations Research, 17*, 1969, 941–957.
4. E. Balas, Intersection cuts—a new type of cutting planes for integer programming. *Operations Research, 19*, 1971, 19–39.
5. E. Balas, Integer programming and convex analysis: intersection cuts from outer polars. *Mathematical Programming, 2,* 1972, 330–382.
6. E. Balas, Disjunctive Programming: Properties of the Convex Hull of Feasible Points. MSRR No. 348, Carnegie Mellon University, July 1974.
7. E. Balas, Disjunctive Programming: Properties of the Convex Hull of Feasible Points. Invited paper with a Foreword by G. Cornuejols and W. Pulleyblank. *Discrete Applied Mathematics, 89*, 1998, 3–44.
8. E. Balas, Disjunctive Programming: Cutting Planes from Logical Conditions. O.L. Mangasarian, R.R. Meyer and S.M. Robinson (editors). *Nonlinear Programming 2*, Academic Press, 1975, 279–312.
9. E. Balas, A Note on Duality in Disjunctive Programming. *Journal of Optimization Theory and Applications, 21*, 1977, 523–528.
10. E. Balas, Disjunctive Programming. *Annals of Discrete Mathematics, 5*, 1979, 3–51.
11. E. Balas, Disjunctive Programming. Reprinting of [10] as Chapter 10 of [88], with an introduction by the author.
12. E. Balas, Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems. *SIAM Journal on Algebraic and Discrete Methods, 6*, 1985, 466–486.
13. E. Balas, The Assignable Subgraph Polytope of a Directed Graph. *Congressus Numerantium, 60,* 1987, 35–42.
14. E. Balas, Projection with a Minimal System of Inequalities. *Computational Optimization and Applications, 10,* 1998, 189–193.
15. E. Balas, Logical Constraints as Cardinality Rules: Tight Representations. *Combinatorial Optimization, 8*, 2004, 115–228. https://doi.org/10.1023/B:JOCO.0000031413.33955.62
16. E. Balas, Multiple-term disjunctive cuts and intersection cuts from multiple rows of the simplex tableau. 20th International Symposium on Mathematical Programming, Chicago, August 23–28, 2009.

E. Balas, *Disjunctive Programming*, https://doi.org/10.1007/978-3-030-00148-3

17. E. Balas, A. Bockmayr, N. Pisaruk and L. Wolsey, On Unions and Dominants of Polytopes. *Mathematical Programming, A, 99,* 2004, 223–239. https://doi.org/10.1007/s10107-003-0432-4
18. E. Balas and P. Bonami, Generating lift-and-project cuts from the LP simplex tableau: open source implementation and testing of new variants. *Mathematical Programming Computation, 1,* 2009, 165–199.
19. E. Balas, S. Ceria, G. Cornuéjols, A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming, 58*, 1993, 295–324.
20. E. Balas, S. Ceria and G. Cornuéjols, Mixed 0-1 Programming by Lift-and-Project in a Branch-and-Cut Framework. *Management Science, 42,* 1996, 1229–1246.
21. E. Balas, S. Ceria, G. Cornuéjols and Natraj, Gomory Cuts Revisited. *OR Letters, 19:1,* 1996, 1–10.
22. E. Balas, G. Cornuéjols, T. Kis, G. Nannicini, Combining Lift and Project and Reduce and Split. *INFORMS Journal on Computing, 25*, 2013, 475–487. https://doi.org/10.1287/ijoc.1120.0515
23. E. Balas and M. Fischetti, On the Monotonization of Polyhedra. In G. Rinaldi and L. Wolsey (editors) *Integer Programming and Combinatorial Optimization*, Proceedings of IPCO 3, CIACO, Louvain-la-Neuve, 1993, 23–38.
24. E. Balas and R.G. Jeroslow, Strengthening Cuts for Mixed Integer Programs. *European Journal of Operations Research, 4*, 1980, 224–234.
25. E. Balas and T. Kis, Intersection cuts–standard versus restricted. *Discrete Optimization, 18* (2015), 189–192. https://doi.org/10.1016/j.disopt.2015.10.001
26. E. Balas and T. Kis, On the relationship between standard intersection cuts, lift-and-project cuts, and generalized intersection cuts. *Mathematical Programming A, 160*, 2016, 85–114. https://doi.org/10.1007/s10107-015-0975-1
27. E. Balas and F. Margot, Generalized intersection cuts and a new cut generating paradigm. *Mathematical Programming A, 137*, 2013, 19–35. https://doi.org/10.1007/s10107-011-0483-x
28. E. Balas and J.B. Mazzola, Nonlinear 0-1 Programming: I. Linearization Techniques, II. Dominance Relations and Algorithms. *Mathematical Programming, 30,* 1984, 1–21 and 22–45.
29. E. Balas and S.M. Ng, On the set covering polytope: I. All facets with coefficients in {0, 1, 2}. *Mathematical Programming, 43,* 1989, 57–69.
30. E. Balas and M. Oosten, On the dimension of projected polyhedra. *Discrete Applied Mathematics 87*, 1998, 1–9.
31. E. Balas and M. Perregaard, Generating Cuts from Multiple-Term Disjunctions. In K. Aardal and B. Gerards (editors). *Integer Programming and Combinatorial Optimization.* Proceedings of IPCO 2001, Utrecht. LNCS 2081, Springer, 2001, 348–360.
32. E. Balas and M. Perregaard, Lift-and-project for mixed 0-1 programming: Recent progress. *Discrete Applied Mathematics, 123*, 2002, 129–154.
33. E. Balas and M. Perregaard, A Precise Correspondence Between Lift-and-Project Cuts, Simple Disjunctive Cuts, and Mixed Integer Gomory Cuts for 0-1 Programming. *Mathematical Programming B, 94,* 2003, 221–245.
34. E. Balas and W.R. Pulleyblank, The Perfectly Matchable Subgraph Polytope of a Bipartite Graph. *Networks, 13,* 1983, 495–518.
35. E. Balas and W.R. Pulleyblank, The Perfectly Matchable Subgraph Polytope of an Arbitrary Graph. *Combinatorica, 9,* 1989, 321–337.
36. E. Balas and A. Qualizza, Monoidal cut strengthening revisited. *Discrete Optimization, 9,* 2012, 40–49.
37. E. Balas and A. Qualizza, Intersection cuts from multiple rows: a disjunctive programming approach. *European Journal of Computational Optimization, 1*, 2013, 3–49. https://doi.org/10.1007/s13675-013-0008-x
38. E. Balas and A. Saxena, Optimizing over the split closure. *Mathematical Programming A, 113,* 2008, 219–240.

39. E. Balas, J. Tama and J. Tind, Sequential Convexification in Reverse Convex and Disjunctive Programming. *Mathematical Programming, 44*, 1989, 337–350.
40. A. Basu, P. Bonami, G. Cornuéjols, F. Margot, On the Relative Strength of Split, Triangle and Quadrilateral Cuts. *Mathematical Programming 126*, 2009, 1220–1229.
41. A. Basu and M. Molinaro, Characterization of the Split Closure via Geometric Lifting. *European Journal of Operational Research 243*, 2015, 745–751.
42. J.F. Benders, Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik, 4,* 1962, 238–252.
43. D. Bienstock, M. Zuckerberg, Subset algebra lift operators for 0-1 integer programming. *SIAM Journal of Optimization, 15,* 2004, 63–95.
44. D. Bienstock, M Zuckerberg, Approximate fixed-rank closures of covering problems, *Mathematical Programming 105*, 2006, 9–27.
45. R.E. Bixby, A brief history of linear and mixed integer programming computation. *Documenta Mathematica*, Extra Volume ISMP, 2012, 107–121.
46. R.E. Bixby, S. Ceria, C.M. McZeal, M.W.P. Savelsbergh, An updated Mixed Integer Programming Library: MIPLIB 3.0. http://www.caam.rice.edu/~bixby/miplib/miplib.html.
47. P. Bonami, On optimizing over lift-and-project closures. *Math Programming Computation, 4*, 2012, 151–179.
48. P. Bonami and M. Minoux, Using rank-1 lift-and-project closures to generate cuts for 0-1 MIP's, a computational investigation. *Discrete Optimization, 2*, 2005, 288–307.
49. Borozan, V., Cornuéjols, G., Minimal valid inequalities for integer constraints. *Mathematics of Operations Research, 34*, 2009, 538–546.
50. S.C. Boyd and W.R. Pulleyblank, Facet generating techniques, Department of Combinatorics and Optimization. University of Waterloo, 1991
51. A. Caprara and A.N. Letchford, On the separation of split cuts and related inequalities. *Mathematical Programming B, 94*, 2003, 279–294.
52. S. Ceria and J. Soares, Convex programming for disjunctive convex optimization. *Mathematical Programming 86*, 1999, 595–614.
53. CglLandP: https://projects.coin-or.org/Cgl/wiki/CglLandP.
54. V. Chvátal, Edmonds polytopes and a hierarchy of combinatorial optimization. *Discrete Mathematics, 4,* 1973, 305–337.
55. COIN-OR website: http://www.coin-or.org/.
56. M. Conforti, G. Cornuéjols and G. Zambelli, Equivalence between intersection cuts and the corner polyhedron. *Oper. Res. Letters, 38,* 2010, 153–155.
57. M. Conforti, G. Cornuejols and G. Zambelli, Extended Formulations in Combinatorial Optimization. *Annals of Operations Research, 204*, 2013, 97–143. https://doi.org/10.1007/s10479-012-1269-0
58. M. Conforti and M. Laurent, On the facial structure of independence system polyhedra. *Math. Oper. Res. 13* (1988) 543–555.
59. M. Conforti, M. di Summa, Y. Faenza, Balas' Formulation for the union of polytopes is optimal. Dipartamento di Matematica "Tullio Levi-Civita", Universita degli Studi di Padova.
60. Computational Optimization Research at Lehigh. MIP Instances, http://coral.ise.lehigh.edu/data-sets/mixed-integer-instances/, September 2017.
61. W. Cook, R. Kannan and A. Schrijver, Chvátal closures for mixed integer programming problems. *Mathematical Programming, 47,* 1990, 155–174.
62. G. Cornuéjols, Y. Li, On the Rank of Mixed 0,1 Polyhedra, in K. Aardal et al. (editors), Integer Programming and Combinatorial Optimization Proceedings of IPCO 8. *Lecture Notes in Computer Science, 2081*, 2001, 71–77.
63. G. Cornuejols and Y. Li, A connection between cutting plane theory and the geometry of numbers. *Mathematical Programming A, 93*, 2002, 123–127.
64. G. Cornuéjols, F. Margot, On the Facets of Mixed Integer Programs with Two Integer Variables and Two Constraints. *Mathematical Programming A, 120*, 2009, 429–456.
65. W.H. Cunningham, Testing membership in matroid polyhedra. *J. Combin. Theory B 36* (1984) 161–188.

66. W.H. Cunningham, On submodular function minimization. *Combinatorica 5* (1985) 185–192.
67. D. Dadush, S.S. Dey and J.P. Vielma, The split closure of a strictly convex body. *Operations Research Letters 39*, 2011,121–126.
68. G.B. Dantzig, R.D. Fulkerson and S.M. Johnson, Solution of a Large-Scale Traveling Salesman Problem. *Operations Research, 2*, 1954, 393–410.
69. S. Dash, O. Günlük, A. Lodi, MIR closures of polyhedral sets. *Mathematical Programming, 121*, 2010, 33–60.
70. Dey, S., Wolsey, L.A., Two Row Mixed-Integer Cuts Via Lifting. *Mathematical Programming B, 124*, 2010, 143–174.
71. J. Edmonds, Submodular functions, matroids, and certain polyhedra. In: Guy, R., Hanani, H., Sauer, N., Schönheim, J., eds. *Combinatorial Structures and their applications*, Gordon and Breach, New York, 1970, pp. 69–87.
72. J. Edmonds and D.R. Fulkerson, D.R., Transversals and matroid partition. *J. Res. Nat. Bur. Standards 69B* (1965) 147–153.
73. F. Eisenbrand and A. Schulz, Bounds on the Chvátal rank of polytopes in the 0-1 cube, in G. Cornuéjols et al. (editors), Integer Programming and Combinatorial Optimization, Proceedings of IPCO 7. *Lecture Notes in Computer Science, 1610*, 1999, 137–150.
74. S. Fiorini, S. Massar, S. Pokutta, H.R. Tiwary, R. de Wolf, Linear versus semidefinite extended formulations: exponential separation and strong lower bounds. *STOC 2012*.
75. M. Fischetti and A. Lodi, Optimizing over the first Chvátal closure. *Mathematical Programming, 110*, 2007, 3–20.
76. M. Fischetti, A. Lodi, Tramontani, On the separation of disjunctive cuts. *Mathematical Programming, 128*, 2011, 205–230, https://doi.org/10.1007/s10107-009-0300-y.
77. R. Fortet, Applications de l'algèbre de Boolean recherche opérationnelle, *Revue Francaise de Recherche Opérationnelle, 4*, 1960, 17–25.
78. D.R. Fulkerson, Blocking and Antiblocking Pairs of Polyhedra. *Math. Program. 1,* (1971) 168–194.
79. M.X. Goemans, Smallest compact formulation for the permutahedron. *Mathematical Programming A*, 2014, https://doi.org/10.100/s101007-014-0757-1.
80. R. Gomory, Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society 64*, 1958, 275–278.
81. R. Gomory, An algorithm for the mixed integer problem. The RAND Corporation, 1960.
82. R. Gomory, Some polyhedra related to combinatorial problems. *Linear Algebra and Its Applications, 2,* 1969, 451–558.
83. P. Halmos. *Naive Set Theory*. Van Nostrand, 1960.
84. Y. Hong and J.N. Hooker, Tight representation of logical constraints as cardinality rules. *Math. Program. 85* (1999) 363–377.
85. S. Iwata, L. Fleisher and S. Fujishige, A combinatorial, strongly polynomial-time algorithm for minimizing submodular functions. *J. ACM 48* (2001) 761–777.
86. R.G. Jeroslow and J. Lowe, Modeling with integer variables. *Math Programming Study 22*, 1984, 167–184.
87. E. Johnson, The group problem for mixed integer programming. *Math Programming Study 2*, 1974, 137–179.
88. M. Juenger et al (editors), *50 Years of Integer Programming 1958–2008: From the Early Years to the State of the Art.* Springer Verlag, 2010, 289–340.
89. V. Kaibel, Extended formulations in combinatorial optimization. *Optima 85*, 2011, 2–7.
90. V. Kaibel, K. Pashkovich, Constructing extended formulations from reflection relations. *Lecture Notes in Computer Science, 6655*, Springer 2011.
91. A. Kazachkov, Non-recursive cut generation. PhD Thesis, the Tepper School of Business, Carnegie Mellon University, 2018.
92. T. Kis, Lift-and-project for general two-term disjunctions. *Discrete Optimization, 12*, 2014, 98–114. https://doi.org/10.1016/j.disopt.2014.02.001
93. T. Koch, T. Achterberg, E. Andersen, O. Bastert, T. Berthold, R.E. Bixby, E. Danna, G. Gamrath, A.M. Gleixner, S. Heinz, A. Lodi, H. Mittelmann, T. Ralphs, Ted D. Salvagnin,

D.E. Steffy, K. Wolter, Kati, MIPLIB 2010: Mixed Integer Programming Library Version 5, *Mathematical Programming Computation, 3*, 2011, 103–163. https://doi.org/10.1007/s12532-011-0025-9.
94. J.B. Lasserre, Global optimization with polynomials and the problem of moments. *SIAM Journal of Optimization, 11,* 2001, 796–817.
95. J.B. Lasserre, An Explicit Exact SPD Relation for Nonlinear Programs. *Lecture Notes in Computer science 2081*, Springer, 2001, 293–303.
96. J.B. Lasserre, Moments, Positive Polynomials and Their Applications. Imperial College Press, London, 2009.
97. S. Lee and I. Grossmann, New algorithms for nonlinear generalized disjunctive programming. *Computers and Chemical Engineering, 24*, 2000, 2125–2142.
98. J. Lenstra, *Sequencing by Enumerative Methods*. Mathematisch Centrul, Amsterdam, 1977.
99. L. Lovász, A. Schrijver, Cones of matrices and set-functions and 0-1 optimization. *SIAM Journal of Optimization, 1,* 1991, 166–190.
100. R.K. Martin, Generating alternative mixed integer programming models using variable definition. *Operations Research, 35*,1987, 820–831.
101. E. Mendelson, *Introduction to Mathematical Logic*. Van Nostrand, 1979.
102. C.E. Miller, A.W. Tucker and R.A. Zemlin, Integer Programming Formulations and Traveling Salesman Problems. *Journal of the ACM, 7*, 1960, 326–329.
103. L. Németi, Das Reihenfolgeproblem in der Fertigungsprogrammierung und Linearplanung mit logischen Bedingungen. *Mathematica*, (Cluj), 6 (1964), 87–99.
104. M.W. Padberg, $(1, k)$-Configurations and facets for packing problems. *Math. Program. 18* (1980) 94–99.
105. M. Padberg and Ting-Yi Sung, An Analytical Comparison of Different Formulations of the Traveling Salesman Problem. *Math. Programming, 52,* 1991, 315–357.
106. M. Perregaard, Generating disjunctive cuts for mixed integer programs. Ph.D. thesis, CMU, 2003.
107. M. Perregaard, A practical implementation of lift-and-project cuts. International Symposium on Mathematical Programming, Copenhagen (2003).
108. R. Raman and I. Grossmann, Modeling and computational techniques for logic-based integer programming. *Computers and Chemical Engineering, 18*, 1994, 563–578.
109. T. Rockafellar, Convex analysis. Princeton University Press, 1970
110. T. Rothvoss, Some 0-1 polytopes need exponential size extended formulations. *Mathematical Programming A, 142*, 2012, 255–268.
111. T. Rothvoss, The matching polytope has exponential complexity. *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, 263–272.
112. H. Sherali, W. Adams, A hierarchy of relaxations between the continuous and the convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics, 3,* 1990, 311–430.
113. N. Sawaya and I. Grossmann, A hierarchy of relaxations for linear generalized disjunctive programming. *European Journal of Operational Research 216*, 2012, 70–82.
114. A. Schrijver, *Theory of Linear and Integer Programming*. Wiley, 1986.
115. A. Schrijver, A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *J. Combinatorial Theory, Ser. B 80* (2000) 346–355.
116. R. Stubbs and S. Mehrotra, A branch-and-cut method for mixed 0-1 convex programming. *Mathematical Programming 86*, 1999, 515–532.
117. A. Tramontani, Lift-and-project cuts in CPLEX 12.5.1 INFORMS meeting in 2013.
118. M. Turkay and I. Grossmann, Logic-based MINLP Algorithms for the optimal synthesis of process networks. *Computers and Chemical Engineering, 20*,1996, 959–978.
119. M. Van Vyve and L. Wolsey, Approximate extended formulations. *Mathematical Programming 105*, 2006, 501–522.
120. R.J. Vanderbei, *Linear Programming: Foundations and Extensions*. Kluwer, 1995.

121. F. Wesselmann, Strengthening Gomory mixed integer cuts: a computational study. University of Paderborn, Germany, 2009.
122. M. Yannakakis, Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences 43*, 1991, 441–466.