

Disjunctive Inequalities: Applications and Extensions

Pietro Belotti* Leo Liberti† Andrea Lodi‡
Giacomo Nannicini§ Andrea Tramontani‡

1 Introduction

A general optimization problem can be expressed in the form

$$\min\{cx : x \in S\}, \quad (1)$$

where $x \in \mathbb{R}^n$ is the vector of decision variables, $c \in \mathbb{R}^n$ is a linear objective function and $S \subset \mathbb{R}^n$ is the set of feasible solutions of (1). Because S is generally hard to deal with, a possible approach for tackling (1) is to optimize the objective function over a suitable relaxation (i.e., easy to solve) $P \supseteq S$. Let x^* be the optimal solution over P . If $x^* \in S$ the problem is solved. Otherwise, one can derive a valid inequality for S in order to separate x^* from S , i.e., an inequality $\alpha x \geq \beta$ satisfied by all the feasible solutions in S and such that $\alpha x^* < \beta$. The addition of the cutting plane $\alpha x \geq \beta$ to the constraints defining P leads to a tighter relaxation $P' = P \cap \{x \in \mathbb{R}^n : \alpha x \geq \beta\}$ and the process can eventually be iterated.

The disjunctive approach to the separation problem, as introduced by Balas [5], considers defining an intermediate set $Q \supseteq S$ not containing x^* and separating x^* from Q . The set Q is obtained by applying to P a valid disjunction D for the set S , such as

$$D := \{x \in \mathbb{R}^n : \bigvee_{h=1}^q D^h x \geq d_0^h\}, \quad (2)$$

where $D^h \in \mathbb{R}^{m_h \times n}$, $d_0^h \in \mathbb{R}^{m_h}$ ($h = 1, \dots, q$), and $S \subseteq D$ (i.e., any feasible solution of (1) satisfies at least one of the conditions of D). Thus, the set Q , denoted as the *disjunctive hull* of D , is defined as $Q := \text{conv}(P \cap D)$, and any valid inequality for Q that cuts off x^* is a disjunctive cut for the problem (1).

*Dept. of Industrial and Systems Engineering, Lehigh University

†LIX, École Polytechnique

‡DEIS, Università di Bologna

§Tepper School of Business, Carnegie-Mellon University

Since the early Nineties, disjunctive inequalities have been successfully exploited both in the context of Mixed Integer Linear Programs (MILPs) as well as in that of Mixed Integer Nonlinear Programs (MINLPs). Entry #1.4.4.1 gives a general overview of disjunctive programming. In the present entry we survey some applications and extensions of disjunctive inequalities with special emphasis to recent developments. The paper is organized as follows. In Section 2 we recall the basic ingredients of disjunctive inequalities for MILPs and we report on recent results on this topic. In Section 3 the application of disjunctive constraints both as modeling tool and cutting planes is discussed for MINLPs. Finally, in Section 4 we consider the fairly new context of application of disjunctive inequalities as sophisticated branching conditions in enumerative algorithms.

2 Disjunctive inequalities in MILP

In the special case of a Mixed Integer Linear Program, S is defined as $S = \{x \in \mathbb{R}^n : Ax \geq b, x \geq 0, x_i \in \mathbb{Z} \forall i \in N^I\}$, where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $N^I \subseteq \{1, \dots, n\}$ is the set of variables constrained to be integer. In this context, the intrinsic difficulty of the problem is due to the integrality restrictions on the variables in N^I . Thus, the relaxation P that is typically considered is the polyhedron associated with S , i.e., $P = \{x \in \mathbb{R}^n : Ax \geq b, x \geq 0\}$, hence the disjunctive hull Q (for a fixed disjunction D) is defined as a union of polyhedra. More precisely,

$$Q := \text{conv} \left(\bigcup_{h=1}^q P_h \right),$$

with $P_h := P \cap \{x \in \mathbb{R}^n : D^h x \geq d_0^h\}$. In such a case, Balas [5, 7] has shown that Q is a polyhedron as well. Even if a full description of Q in the space of the x variables may require an exponential number of constraints, the key result of Balas [5, 7] is that Q has a compact representation in a higher-dimensional space. Namely, there exists a polyhedron $\tilde{Q} := \{(x, y) \in \mathbb{R}^{n+p} : Bx + Cy \geq d\}$ whose projection onto the x -space is Q , and \tilde{Q} has around qn variables and $qm + \sum_{h=1}^q m_h$ constraints. This implies that separating x^* from Q can be solved by linear programming.

Indeed, each polyhedron P_h ($h = 1, \dots, q$) can be defined as $P_h = \{x \in \mathbb{R}^{n+1} : A^h x \geq b^h, x \geq 0\}$, where

$$A^h = \begin{bmatrix} A \\ D^h \end{bmatrix} \text{ and } b^h = \begin{bmatrix} b \\ d_0^h \end{bmatrix}.$$

Another key result of Balas [7] is that all the inequalities $\alpha x \geq \beta$ valid for Q are described by the polyhedral cone

$$Q^\# = \{(\alpha, \beta) \in \mathbb{R}^{n+1} : \alpha \geq u^h A^h, \beta \leq u^h b^h \text{ for some } u^h \geq 0, h = 1, \dots, q\}.$$

Furthermore, for a full-dimensional polyhedron Q there is a one to one correspondence among the extreme rays of $Q^\#$ and the facets of Q .

2.1 Separating disjunctive cuts in MILP

Typically, disjunctive cuts are separated by considering a very special subset of 2-term disjunctions (i.e., with $q = 2$), namely, the so-called split disjunctions of the form

$$\pi x \leq \pi_0 \quad \vee \quad \pi x \geq \pi_0 + 1, \quad (3)$$

with $\pi \in \mathbb{Z}^n$, $\pi_0 \in \mathbb{Z}$, $\pi_i = 0 \forall i \notin N^I$. Disjunctive cuts arising from disjunctions of the form (3) are also known as *split cuts*, see, e.g., Cook, Kannan, and Schrijver [19] and entry #1.4.3.7.

Given a solution x^* of $P \setminus S$, a common approach for separating x^* from S is to consider an *elementary* split disjunction of the form

$$x_i \leq \pi_0 \quad \vee \quad x_i \geq \pi_0 + 1, \quad (4)$$

where $x_i \in N^I$, $x_i^* \notin \mathbb{Z}$, $\pi = e_i$ is the i -th unit vector and $\pi_0 = \lfloor x_i^* \rfloor$ ¹. Thus, the disjunctive hull Q is simply defined as the union of the two polyhedra $P_0 = \{x \in P : x_i \leq \pi_0\}$ and $P_1 = \{x \in P : x_i \geq \pi_0 + 1\}$. By Farkas lemma, a most-violated disjunctive cut $\alpha x \geq \beta$ valid for Q can be found by solving the so-called *Cut Generating Linear Program* (CGLP), that determines the Farkas multipliers (u, u_0, v, v_0) associated with the inequalities defining P_0 and P_1 so as to maximize the violation of the resulting cut with respect to x^* :

$$\begin{aligned} \text{(CGLP)} \quad & \min \quad \alpha x^* - \beta \\ & \alpha \geq uA - u_0 e_i, \quad \beta \leq ub - u_0 \pi_0, \\ & \alpha \geq vA + v_0 e_i, \quad \beta \leq vb + v_0(\pi_0 + 1), \\ & u, v, u_0, v_0 \geq 0. \end{aligned} \quad (5)$$

Once a violated cut has been found as a solution of (5), the cut can be easily strengthened *a posteriori* through the Balas and Jeroslow [10] procedure. Such a strengthening can be seen as finding the best split disjunction for a given set of Farkas multipliers.

By construction, any feasible CGLP solution with negative objective function value corresponds to a violated disjunctive cut. However, the feasible CGLP set is a cone and needs to be truncated by means of a suitable normalization condition, so as to produce a bounded LP in case a violated cut exists. Different normalization conditions might lead to completely different results in terms of the *strength* of the separated cuts because they heavily affect the choice of the optimal CGLP solution.

Several normalizations have been proposed in the literature, see, e.g., Ceria and Soares [17], Balas and Perregaard [11] and Fischetti, Lodi and Tramontani [24]. The most natural and simple normalization is

$$u_0 + v_0 = 1, \quad (6)$$

¹For 0–1 mixed integer programs, split cuts arising from elementary disjunctions of the form (4) are denoted as *lift-and-project* cuts (see, e.g., [9] and entry #1.4.3.8).

and takes into account only the dual multipliers of the disjunctive constraints. Such a normalization has been considered, for instance, by Balas and Saxena [13] to optimize over the first *split closure*, i.e., the polyhedron obtained by adding to P all split cuts which could be derived from the original set of linear constraints (see, entries #1.4.3.1 and #1.4.3.7). One of the most widely-used (and effective) truncation condition reads instead

$$eu + ev + u_0 + v_0 = 1, \tag{7}$$

where $e = (1, \dots, 1)$. This latter condition was proposed by Balas [6] and investigated, for instance, by Ceria and Soares [17] and Balas and Perregaard [11, 12].

Recently, Fischetti, Lodi and Tramontani [24] analyzed computationally the two above normalization conditions, thus showing that the truncation condition (7) generally outperforms (6). Indeed, normalization (7) naturally enforces the separation of *sparse* and *low rank* inequalities that turn out to be more effective in terms of the percentage of the integrality gap closed.

Unfortunately, any normalization can in some cases produce weak cuts. The projection of the disjunctive cone (5) onto the polar space (α, β) yields precisely the cone $Q^\#$. As discussed, for a full-dimensional polyhedron Q there is a one to one correspondence among the extreme rays of $Q^\#$ and the facets of Q . However, this correspondence is lost in the cone (5), that is defined in a lifted space that explicitly involves the Farkas multipliers (u, v, u_0, v_0) . Indeed, there are many extreme rays of the CGLP cone that correspond to dominated cuts, and this property is independent of the normalization used to truncate the cone. Fischetti, Lodi and Tramontani [24] gave a theoretical characterization of *weak rays* of the disjunctive cone (5) that lead to dominated cuts, and showed how the presence of redundant constraints in the formulation of P_0 and P_1 increases the number of weak rays in (5), thus affecting the quality of the generated disjunctive cuts. It remains an open question how to find a computationally efficient way of dealing with redundant constraints in the separation.

A key step in the separation of disjunctive cuts for MILPs is the work by Balas and Perregaard [12]. Even if disjunctive cuts can be separated by linear programming, solving the CGLP in the lifted space (u, v, u_0, v_0) may be in general too time consuming. Addressing the normalization (7), Balas and Perregaard [12] provided a precise correspondence between the bases of the CGLP truncated with (7) and the bases of the original LP $Ax - Is = b, x \geq 0, s \geq 0$. Then, they developed an elegant and efficient way of solving the CGLP by performing pivot operations in the original tableau involving (x, s) variables only.

In particular, Balas and Perregaard [12] also showed that the *Mixed Integer Gomory* (MIG) cut [27] associated with the row of the basic variable $x_i, i \in N^I$, in the optimal simplex tableau of the system $Ax - Is = b, x \geq 0, s \geq 0$, corresponds to a basic solution of the CGLP truncated with (7). Thus, solving the CGLP with normalization (7) can be interpreted as a way of strengthening the MIG cut from the tableau. Lately, Fischetti, Lodi and Tramontani [24]

have shown that the basic CGLP solution yielding the MIG cut is, in fact, an *optimal* solution if the CGLP is truncated with normalization (6). Hence, the strengthening of the MIG cut lies in the better behavior of the normalization (7) with respect to (6).

The work by Balas and Perregaard [12] has been recently extended by Balas and Bonami [8]. In [8], the authors considered different normalization conditions of the form

$$\lambda u + \lambda v + u_0 + v_0 = \lambda_0,$$

with $\lambda \in \mathbb{R}_+^m$, $\lambda_0 \in \mathbb{R}_+$, and developed a procedure that iteratively combines the “pivoting mechanism” of [12] for finding the best set of Farkas multipliers for a fixed disjunction, with the Balas and Jeroslow [10] strengthening of the disjunction for a given set of multipliers. The results presented in [8] show that disjunctive inequalities can be profitably used to improve on the performance of general purpose Branch-and-Cut algorithms.

3 Disjunctive inequalities in MINLP

Disjunctions are essential for modeling several types of nonconvex constraints, including those arising in MINLP. For these problems, S is defined as

$$S = \{x \in \mathbb{R}^n : g_j(x) \leq 0 \forall j \in M, x_i \in \mathbb{Z} \forall i \in N^I\},$$

where, for all $j \in M$, $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$ is a multivariate (possibly nonconvex) function. When all of the g_j are affine, we have an MILP, which is therefore a subclass of MINLP.

While the integrality of a subset of variables represents an important class of nonconvex constraints, there exist other nonconvexities in MINLP that can give rise to disjunctions, and that can be used by Branch-and-Bound algorithms. Below we describe a few examples of how disjunctions are used to generate disjunctive inequalities for the general MINLP case and for some special cases.

3.1 General nonconvex MINLP

Despite the more general nonlinear setting, most of the literature in MINLP and global optimization consider linear disjunctions of the form (2). Horst and Tuy [32] described a general procedure whereby disjunctions of the form (2) are repeatedly generated and, correspondingly, a linear cut is separated that is valid for all sets $S \cap \{x \in \mathbb{R}^n : D^h x \geq d_0^h\}$ for all elements of a disjunction, i.e., for all $h = 1, 2, \dots, q$. They proceeded to prove that such a procedure is guaranteed to converge under mild conditions.

One of the lower bounding procedures for nonconvex MINLP consists of *reformulating* the set of constraints into a set of nonconvex equality constraints of the form $x_k = f_k(x_1, x_2, \dots, x_{k-1})$, with $f_k : \mathbb{R}^{k-1} \rightarrow \mathbb{R}$ nonlinear [52, 54]. A lower bound can then be obtained by solving an LP relaxation of the reformulation, with $\bar{n} \geq n$ variables: $P_{\text{LP}} = \{x \in \mathbb{R}^{\bar{n}} : Ax \geq b\}$. P_{LP} is constructed

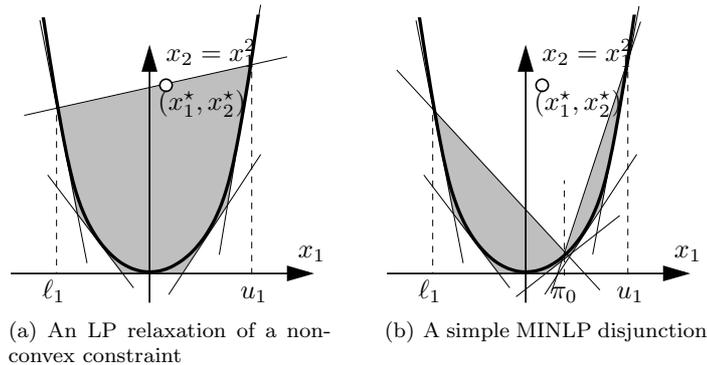


Figure 1: An MINLP disjunction. In (a), the shaded area is the LP relaxation of the constraint $x_2 = f_2(x_1) = (x_1)^2$ with $x_1 \in [\ell_1, u_1]$, whereas (x_1^*, x_2^*) is the value of x_1 and x_2 in the optimum of the LP relaxation. In (b), a disjunction $x_1 \leq \pi_0 \vee x_1 \geq \pi_0$, albeit not violated by (x_1^*, x_2^*) , can be used to generate two LP relaxations (the two smaller shaded areas) which in turn allow to construct a disjunction $D^1x \geq d^1 \vee D^2x \geq d^2$ violated by x^* .

by adding, for each constraint $x_k = f_k(x_1, x_2, \dots, x_{k-1})$, a system of linear constraints $A^kx \geq b^k$. An example of this step is shown in Figure 1(a) for the constraint $x_2 = f_2(x_1) = x_1^2$, where the inequalities of $A^kx \geq b^k$ delimit the shaded area.

Let x^* be the optimal solution to P_{LP} . If x^* satisfies integrality constraints and all nonlinear constraints $x_k = f_k(x_1, x_2, \dots, x_{k-1})$, the problem is solved. If it satisfies all integrality constraints while violating at least one of the nonlinear ones, a disjunction is sought that is violated by x^* . The simple disjunction $x_i \leq \pi_0 \vee x_i \geq \pi_0$, although valid for MINLP, is clearly not violated. However, the linear relaxations P'_{LP} and P''_{LP} of $S' = S \cap \{x \in R^n : x_i \leq \pi_0\}$ and $S'' = S \cap \{x \in R^n : x_i \geq \pi_0\}$, respectively, depicted in Figure 1(b), contain new linear constraints that can be used to construct a disjunction of type (2) that is violated by x^* .

Since a disjunctive inequality for $P'_{\text{LP}} \cup P''_{\text{LP}}$ is also valid for $S' \cup S''$, a procedure similar to that used in MILP, where the CGLP is solved using the LP relaxation, can be applied—see for instance Belotti [14]. Zhu et al. [55] proposed a simple extension to MINLP where the CGLP is solved for disjunctions on binary variables.

3.2 Special classes of MINLP

Specific classes of disjunctions arise from MINLPs with a certain structure. For MINLP with binary variables and whose continuous relaxation is convex, Stubbs and Mehrotra [53] generalized the procedure proposed by Balas, Ceria and Cornu ejols [9] and described a separation procedure based on a convex

optimization problem.

A similar (specialized) procedure has been successfully applied to Mixed Integer Conic Programming (MICP) problems, which can be thought of as MILP amended by a set of conic constraints. The second order cone and the cone of symmetric semidefinite matrices are among the most important classes of conic constraints in this class. Çezik and Iyengar [18] and lately Drewes [23] proposed, for MICP where all disjunctions are generated from binary variables, an application of the lifting procedure to the conic case, whereby disjunctive inequalities are obtained by solving a continuous conic optimization problem. Analogously to Frangioni and Gentile [26], restricting to a special type of convex constraint (second order or semidefinite cone) allows to obtain more specialized and thus efficient procedures for obtaining a disjunctive inequality.

Mathematical Programs with Equilibrium Constraints (MPEC). These MINLPs contain nonconvex constraints $x^\top y = 0$, with $x \in \mathbb{R}_+^k$, $y \in \mathbb{R}_+^k$. These can be more easily stated as $x_i y_i = 0$ for all $i = 1, 2, \dots, k$, and give rise to simple disjunctions $x_i = 0 \vee y_j = 0$. Júdice et al. [33] studied an MPEC where the only nonlinear constraints are the complementarity constraints. Relaxing the latter yields an LP. Disjunctive cuts are generated from solutions of the LP that violate a complementarity constraint (i.e., $x_i > 0$ and $y_j > 0$) through the observation that both variables are basic and by applying standard disjunctive arguments to the corresponding tableau rows.

Disjunctive cuts have been proposed by Audet, Haddad and Savard [4] in the context of Linear Bilevel Programming, where the optimality of a *lower level* subproblem is required for a solution to be feasible for the *upper level* problem, and is enforced through complementarity constraints.

Quadratically Constrained Quadratic Programming (QCQP). Saxena, Bonami and Lee [50, 51] proposed two classes of disjunctive cuts for QCQP problems, which contain constraints of the type $x^\top Qx + b^\top x + c \leq 0$, i.e., nonconvex because in general $Q \not\preceq 0$. These problems are reformulated as linear programs with an extra nonconvex constraint of the form $Y = xx^\top$, where Y is an $n \times n$ matrix of auxiliary variables. Relaxing these constraints to $Y - xx^\top \succeq 0$, thereby allowing solutions such that $xx^\top - Y \succeq 0$, leads to a (convex) semidefinite program, which yields good lower bounds [31, 49].

Moreover, Saxena, Bonami and Lee [50, 51] used the nonconvex constraint $xx^\top - Y \succeq 0$ to obtain disjunctive cuts as follows. Disjunctions are derived from the nonconvex constraint $(v^\top x)^2 \geq v^\top Y v$, where vector v is obtained from the negative eigenvalues of the matrix $\bar{x}\bar{x}^\top - \bar{Y}$, and (\bar{x}, \bar{Y}) is a solution of the relaxation. In the second paper [50], this procedure is refined to generate cuts for the non-reformulated problem, thus avoiding the need of the auxiliary matrix variable Y .

3.3 Generalized Disjunctive Programming.

Generalized Disjunctive Programming (GDP) is an extension of Disjunctive Programming [5] originally proposed in [48] as a modelling framework (with a Branch-and-Bound based general-purpose solution algorithm) targeting problems in chemical engineering and process synthesis. GDP explicitly formulates conditional constraints via boolean variables and logic formulæ. This emphasizes the role that logic-based transformations (such as De Morgan’s laws or passing from conjunctive to disjunctive normal forms) have on the formulation. Furthermore, it allows the specification of different lower bounding problems.

The GDP is formulated as follows:

$$\min f(x) + \sum_{k \in K} c_k \quad (8)$$

$$g(x) \leq 0 \quad (9)$$

$$\forall k \in K \quad \bigvee_{i \in D_k} (Y_{ik} \wedge c_k = \gamma_{ik} \wedge r_{ik}(x) \leq 0) \quad (10)$$

$$\Omega(Y) \wedge x \in X \subseteq \mathbb{R}^n \wedge c \in \mathbb{R}^{|K|}, \quad (11)$$

where K and D_k ($k \in K$) are sets of indices, γ_{ik} is a known parameter for all $k \in K, i \in D_k$, Y are boolean and x, c are continuous decision variables, Ω is a Conjunctive Normal Form (CNF) logic formula having free variables Y and containing the clauses $\bigvee_{i \in D_k} Y_{ik}$ for all $k \in K$, with \bigvee denoting “exclusive or”. The formula $\Omega(Y)$ acts as a constraint in the sense that it should be true in order for Y to be a feasible solution of (8)-(11).

It is clear that (8)-(11) can be reformulated to an MINLP by replacing $c_k = \gamma_{ik}$ by $Y_{ik}(c_k - \gamma_{ik}) = 0$ and $r_{ik}(x) \leq 0$ by $Y_{ik}r_{ik}(x) \leq 0$, and writing $\Omega(Y)$ as the corresponding boolean expression in the $\{0, 1\}$ -variables Y . When X is a vector of non-empty intervals (say $[0, U]$), the products involving the Y variables can be further reformulated to “big-M” type constraints, which are convex whenever $r_{ik}(x)$ are convex. Thus, a “big-M” convex relaxation can be easily obtained from (8)-(11) using well-known techniques on the possibly nonconvex functions f and g (see, e.g., [15]). The interest of formulation (8)-(11), however, is that it can be used to yield a different convex relaxation [29] which is in general stronger than the “big-M” one, and rests on a convex-hull

type reformulation of the disjunctive constraints (10):

$$\begin{aligned}
\forall k \in K \quad x &= \sum_{i \in D_k} v^{jk} \\
\forall k \in K \quad c_k &= \sum_{i \in D_k} \lambda_{ik} \gamma_{ik} \\
\sum_{i \in D_k} \lambda_{ik} &= 1 \\
\forall k \in K, i \in D_k \quad \lambda_{ik} r_{ik} (v^{jk} / \lambda_{ik}) &\leq 0 \\
\forall k \in K, i \in D_k \quad 0 \leq v^{ik} &\leq \lambda_{ik} U \\
\forall k \in K, i \in D_k \quad \lambda_{ik} &\in [0, 1].
\end{aligned}$$

4 Disjunctive inequalities and branching

Within the context of a Branch-and-Bound [37] algorithm, a key decision that has to be repeatedly performed is how to divide a problem into subproblems. Indeed, Branch-and-Bound makes an implicit use of the concept of disjunction: whenever we are not able to solve a problem \mathcal{P} of the form (1), we choose a disjunction D of the feasible region of \mathcal{P} (i.e., a disjunction D of the form (2) valid for S), and we divide \mathcal{P} into two or more subproblems $\mathcal{P}_1, \dots, \mathcal{P}_q$ corresponding to the q terms of the disjunction. Hence, we are guaranteed that the optimal solution of one of the q subproblems coincides with the optimum of \mathcal{P} . In this case, we say that we *branch on the disjunction D* .

In MILP, branching is typically done on two-term disjunctions, and in particular the standard method is to branch on a single variable, i.e., an *elementary* disjunction. Let x^* be the solution to the LP relaxation of the current problem \mathcal{P} , and let $i \in N^I$ such that $x_i^* \notin \mathbb{Z}$. Then, we branch on the variable x_i by dividing \mathcal{P} into \mathcal{P}_0 and \mathcal{P}_1 , where \mathcal{P}_0 is equal to \mathcal{P} with the addition of the constraint $x_i \leq \lfloor x_i^* \rfloor$, and \mathcal{P}_1 is equal to \mathcal{P} with the addition of the constraint $x_i \geq \lceil x_i^* \rceil$. Choosing which variable should be branched on at each step is of fundamental importance for the performance of Branch-and-Bound. We refer to Achterberg, Koch and Martin [2] for a recent survey on this topic.

Even though branching on single variables is the method currently implemented by solvers for integer programs (both linear and nonlinear), it need not be so. For MILPs, branching can be done on any split disjunction of the form (3). By integrality, every feasible solution to \mathcal{P} satisfies any split disjunction. In the branching literature, every disjunction which is not elementary is labeled as a *general* disjunction. Several approaches to branch on general disjunctions have been proposed in the MILP literature. These approaches can be ascribed to one of two categories. The first category contains methods that try to identify “thin” directions of the polyhedron P associated with the LP relaxation of the MILP; the second category focuses on improving as much as possible the LP bound at the children nodes. The concept of thin direction requires the concept of *width* of a full-dimensional polyhedron P along a direction u , which is defined

as $\max_{x,y \in P}(ux - uy)$. Thus, for a *pure* integer program associated with P , the *integer width* is defined as

$$\min_{\pi \in \mathbb{Z}^n \setminus \{0\}} \max_{x,y \in P} (\pi x - \pi y).$$

This definition naturally extends to the mixed integer linear case by considering integer directions $\pi \in \mathbb{R}^n \setminus \{0\}$ with $\pi_j = 0$ for $j \notin N^I$. Mahajan and Ralphs [43] discussed the formulation of optimization models to select both thin directions and split disjunctions that yield maximum bound improvement. Note that both the problems of finding a disjunction with smallest integer width or largest bound improvement are strongly **NP**-hard [44].

4.1 Branching on thin directions

Branching on thin directions of the polyhedron P associated with the LP relaxation of the MILP is a method that derives from the work of Lenstra [39] on solving integer programs in fixed dimension in polynomial time (see also [30, 42]). The idea is as follows. First, some thin directions of P are computed, using the lattice basis reduction algorithm by Lenstra, Lenstra and Lovász [38]. Then, the space is transformed so that these directions correspond to unit vectors, and the problem is solved by Branch-and-Bound in the new space. Branching on simple disjunctions in this space translates back to branching on general disjunctions in the original space. This method has proven successful for some particular instances where standard Branch-and-Bound fails because of the huge size of the enumeration tree: Aardal et al. [1] discussed the solution of the difficult Market Split instances [20], while Krishnamoorthy and Pataki studied decomposable knapsack problems [36]. See also Mehrotra and Li [45].

4.2 Branching for maximum bound improvement

Another line of research which has been pursued is that of selecting a good general disjunction for branching at each node of the Branch-and-Bound tree, in order to improve as much as possible the bound at the children nodes. Owen and Mehrotra [46] proposed branching on split disjunctions with coefficients in $\{-1, 0, 1\}$ on the integer variables with fractional values at the current node. They generate all possible such disjunctions, and evaluate them using strong branching [2], in order to select the one that gives the largest improvement of the dual bound. Karamanov and Cornuéjols [35] provided a computational study of branching on the split disjunctions that define the Mixed Integer Gomory cuts [27] associated with the optimal simplex tableau of the current Branch-and-Bound node. Several such disjunctions are generated, using the quality of the underlying cut as a proxy for the strength of the disjunction, and strong branching is used to select one. A similar approach was proposed by Cornuéjols, Liberti and Nannicini [21], but instead of considering the split disjunctions associated with MIG cuts that can be read directly from the optimal simplex tableau, an improvement step is applied first: the rows of the simplex tableau

are linearly combined with integer coefficients in order to reduce the coefficients of the resulting combination. This step corresponds to tilting the underlying disjunctions so as to produce stronger MIG cuts (see also the Reduce-and-Split algorithm by Andersen, Cornuéjols and Li [3]).

References

- [1] K. Aardal, R. E. Bixby, C. A. J. Hurkens, A. K. Lenstra, and J. W. Smeltink. Market split and basis reduction: Towards a solution of the Cornuéjols-Dawande instances. *INFORMS Journal on Computing*, 12(3):192–202, 2000.
- [2] T. Achterberg, T. Koch, and A. Martin. Branching rules revisited. *Operations Research Letters*, 33(1):42–54, 2005.
- [3] K. Andersen, G. Cornuéjols, and Y. Li. Reduce-and-split cuts: Improving the performance of mixed integer Gomory cuts. *Management Science*, 51(11):1720–1732, 2005.
- [4] C. Audet, J. Haddad, and G. Savard. Disjunctive cuts for continuous linear bilevel programming. *Optimization Letters*, 1(3):259–267, 2007.
- [5] E. Balas. Disjunctive programming. In P. L. Hammer, E. L. Johnson, and B. H. Korte, editors, *Annals of Discrete Mathematics 5: Discrete Optimization*, pages 3–51. North Holland, 1979.
- [6] E. Balas. A modified lift-and-project procedure. *Mathematical Programming*, 79:19–31, 1997.
- [7] E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89:3–44, 1998.
- [8] 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*, 1:165–199, 2009.
- [9] E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58:295–324, 1993.
- [10] E. Balas and R. Jeroslow. Strengthening cuts for mixed integer programs. *European Journal of Operations Research*, 4:224–234, 1980.
- [11] E. Balas and M. Perregaard. Lift-and-project for mixed 0–1 programming: recent progress. *Discrete Applied Mathematics*, 123:129–154, 2002.
- [12] 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*, 94(2-3):221–245, 2003.

- [13] E. Balas and A. Saxena. Optimizing over the split closure. *Mathematical Programming, Series A*, 113(2):219–240, 2008.
- [14] P. Belotti. Disjunctive cuts for non-convex MINLP. Technical report, Lehigh University, 2009.
- [15] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4):597–634, 2009.
- [16] L. T. Biegler, I. E. Grossmann, and A. W. Westerberg. *Systematic Methods of Chemical Process Design*. Prentice Hall, Upper Saddle River (NJ), 1997.
- [17] S. Ceria and J. Soares. Disjunctive cuts for mixed 0–1 programming: duality and lifting. Technical report, GSB, Columbia University, 1997.
- [18] M. T. Çezik and G. Iyengar. Cuts for Mixed 0-1 Conic Programming. *Math. Programming, Ser. A*, 104:179–200, 2005.
- [19] W. Cook, R. Kannan, and A. Schrijver. Chvátal closures for mixed integer programming problems. *Mathematical Programming*, 47:155–174, 1990.
- [20] G. Cornuéjols and M. Dawande. A class of hard small 0-1 programs. In R. E. Bixby and E. A. Boyd, editors, *Proceedings of the 6th IPCO Conference*, volume 1412 of *Lecture Notes in Computer Science*, pages 284–293. Springer-Verlag, Berlin, 1998.
- [21] G. Cornuéjols, L. Liberti, and G. Nannicini. Improved strategies for branching on general disjunctions. *Mathematical Programming A*, 2009. Accepted for publication.
- [22] G. Cornuéjols and R. Tütüncü. *Optimization Methods in Finance*. Cambridge University Press, Cambridge, 2006.
- [23] S. Drewes. *Mixed Integer Second Order Cone Programming*. PhD thesis, Technische Universität Darmstadt, 2009.
- [24] M. Fischetti, A. Lodi, and A. Tramontani. On the separation of disjunctive cuts. *Mathematical Programming, Series A*, DOI 10.1007/s10107-009-0300-y, 2009.
- [25] C. A. Floudas. Global optimization in design and control of chemical process systems. *Journal of Process Control*, 10:125–134, 2001.
- [26] A. Frangioni and C. Gentile. Perspective cuts for a class of convex 0-1 mixed integer programs. *Mathematical Programming*, 106(2):225–236, 2006.
- [27] R. E. Gomory. An algorithm for the mixed integer problem. Technical Report RM-2597, The RAND Corporation, 1960.

- [28] R. E. Gomory. An algorithm for integer solutions to linear programs. In P. Wolfe, editor, *Recent Advances in Mathematical Programming*, pages 269–302. McGraw-Hill, New York, 1963.
- [29] I. Grossmann and S. Lee. Generalized convex disjunctive programming: Nonlinear convex hull relaxation. *Computational Optimization and Applications*, 26:83–100, 2003.
- [30] M. Grötschel, L. Lovász, and A. Schrijver. Progress in combinatorial optimization. In *Geometric methods in Combinatorial Optimization*, pages 167–183. Academic Press, Toronto, 1984.
- [31] C. Helmberg and F. Rendl. Solving quadratic (0,1)-problems by semidefinite programs and cutting planes. *Math. Prog.*, 82(3):291–315, 1998.
- [32] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer Verlag, Berlin, 1996.
- [33] J. J. Júdice, H. D. Sherali, I. M. Ribeiro, and A. M. Faustino. A complementarity-based partitioning and disjunctive cut algorithm for mathematical programming problems with equilibrium constraints. *Journal of Global Optimization*, 36:89–114, 2006.
- [34] J. Kallrath. Solving planning and design problems in the process industry using mixed integer and global optimization. *Annals of Operations Research*, 140(1):339–373, 2005.
- [35] M. Karamanov and G. Cornuéjols. Branching on general disjunctions. Technical report, Carnegie Mellon University, 2005.
- [36] B. Krishnamoorthy and G. Pataki. Column basis reduction and decomposable knapsack problems. *Discrete Optimization*, 6(3):242–270, 2009.
- [37] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.
- [38] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 4(261):515–534, 1982.
- [39] H. W. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.
- [40] L. Liberti, C. Lavor, and N. Maculan. A branch-and-prune algorithm for the molecular distance geometry problem. *International Transactions in Operational Research*, 15:1–17, 2008.
- [41] L. Liberti, C. Lavor, M. A. C. Nascimento, and N. Maculan. Reformulation in mathematical programming: an application to quantum chemistry. *Discrete Applied Mathematics*, 157(6):1309–1318, 2009.

- [42] L. Lovász and H. E. Scarf. The generalized basis reduction algorithm. *Mathematics of Operations Research*, 17(3):751–764, 1992.
- [43] A. Mahajan and T. K. Ralphs. Experiments with branching using general disjunctions. In *Proceedings of the Eleventh INFORMS Computing Society Meeting*, 2009. To Appear.
- [44] A. Mahajan and T. K. Ralphs. On the complexity of branching on general hyperplanes for integer programming. *SIAM Journal on Optimization*, 2009. To appear.
- [45] S. Mehrotra and Z. Li. On generalized branching method for mixed integer programming. Technical report, Northwestern University, Evanston, Illinois, 2004.
- [46] J. Owen and S. Mehrotra. Experimental results on using general disjunctions in branch-and-bound for general-integer linear program. *Computational Optimization and Applications*, 20:159–170, 2001.
- [47] A. T. Phillips and J. B. Rosen. A quadratic assignment formulation of the molecular conformation problem. Technical report, CSD, Univ. of Minnesota, 1998.
- [48] R. Raman and I. Grossmann. Modelling and computational techniques for logic based integer programming. *Computers and Chemical Engineering*, 18(7):563–578, 1994.
- [49] F. Rendl and R. Sotirov. Bounds for the quadratic assignment problem using the bundle method. *Mathematical Programming*, 109(2-3):505–524, 2007.
- [50] A. Saxena, P. Bonami, and J. Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: Projected formulations, November 2008. Optimization Online.
- [51] A. Saxena, P. Bonami, and J. Lee. Disjunctive cuts for non-convex mixed integer quadratically constrained programs. In A. Lodi, A. Panconesi, and G. Rinaldi, editors, *Proceedings of the 13th Integer Programming and Combinatorial Optimization Conference*, volume 5035 of *Lecture Notes in Computer Science*, pages 17–33, 2008.
- [52] E. M. B. Smith and C. C. Pantelides. A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. *Computers & Chem. Eng.*, 23:457–478, 1999.
- [53] R. A. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86:515–532, 1999.

- [54] M. Tawarmalani and N. V. Sahinidis. *Convexification and global optimization in continuous and mixed-integer nonlinear programming: Theory, algorithms, software and applications*, volume 65 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, Dordrecht, 2002.
- [55] Y. Zhu, Y. Hu, H. Wu, and M. Nakaiwa. An improved branch-and-cut algorithm for mixed-integer nonlinear systems optimization problem. *AIChE Journal*, 54(12):3239–3247, 2008.