manuscript No.
(will be inserted by the editor)

Pierre Bonami[*] · Andrea Lodi[**] · Andrea Tramontani[***] · Sven Wiese[†]

On Mathematical Programming with Indicator Constraints

February 5, 2015

Abstract. In this paper we review the relevant literature on mathematical optimization with logical implications, i.e., where constraints can be either active or disabled depending on logical conditions to hold. In the case of convex functions, the theory of disjunctive programming allows one to formulate these logical implications as convex nonlinear programming problems in a space of variables lifted with respect to its original dimension. We concentrate on the attempt of avoiding the issue of dealing with large NLPs. In particular, we review some existing results that allow to work in the original space of variables for two relevant special cases where the disjunctions corresponding to the logical implications have two terms. Then, we significantly extend these special cases in two different directions, one involving more general convex sets and the other with disjunctions involving three terms.

Computational experiments comparing disjunctive programming formulations in the original space of variables with straightforward bigM ones show that the former are computationally viable and promising.

1. Introduction

In Mathematical Optimization one is often facing the problem of modeling logical implications. As a simple example consider the optimization problem

$$\begin{aligned} \min \ & f(x,z) \\ \text{subject to } & h(x,z) \le 0 \\ & \left.\begin{aligned} [z_k = 0] \implies [x \in S_0^k] \\ [z_k = 1] \implies [x \in S_1^k] \end{aligned}\right\} \ \forall\, k = 1,\dots,K \\ & x \in \mathbb{R}^m \\ & z \in \{0,1\}^K \end{aligned} \qquad (P_0)$$

where one wants to minimize a function $f : \mathbb{R}^{m+K} \to \mathbb{R}$ subject to a set of constraints. Global constraints are expressed by the function $h : \mathbb{R}^{m+K} \to \mathbb{R}$. Moreover, K pairs of logical constraints are given, each one involving a binary *indicator* variable z_k. This variable indicates that either x is constrained to belong to the set S_0^k or to the set S_1^k. In other words, each z_k indicates whether the constraints defining either of the two sets are imposed or not, i.e., if a set of constraints is "switched on or off".

Systems like (P_0) can be found in the literature in a number of applications. This is either because they model *explicit* logical arguments like "if facility j is *inactive*, then no client i can be assigned to it", as in the *facility location* [19]; or because the mathematical programming formulation is constructed by imposing a specific order through (otherwise *implicit*) logical implications like "either job i is executed on machine k before job j or vice versa", as in *job shop* scheduling [13]. Devising efficient and computationally effective methods to deal with logical implications is one of the most fundamental needs to enhance the Mathematical Programming solvers' capability of facing real-world optimization problems [41,37].

Address(es) of author(s) should be given

[*] CPLEX Optimization, IBM Spain, Sta. Hortensia 26-28, 28002 - Madrid (M), Spain, e-mail: `pierre.bonami@es.ibm.com`

[**] Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione "Guglielmo Marconi" - Università di Bologna, e-mail: `andrea.lodi@unibo.it`

[***] CPLEX Optimization, IBM Italy, Via Martin Luther King 38/2, Bologna (BO), Italy 40132, e-mail: `andrea.tramontani@it.ibm.com`

[†] Dipartimento di Ingegneria dell'Energia Elettrica e dell'Informazione "Guglielmo Marconi" - Università di Bologna, e-mail: `sven.wiese@unibo.it`

In order to formulate system (P_0) as a Mathematical Programming problem one needs to remove the logical implications and write "explicit" constraints. The most straightforward way of doing that is by using the so-called *bigM* formulation, where constraints are activated or deactivated by multiplying the binary indicator variable by a very large (precomputed) constant. Alternatively, one can use *disjunctive programming* techniques [5], which are the main topic of the present paper. Before going into the details of those two possibilities in Section 2, we anticipate that we will consider the general mathematical programming problem

$$\begin{aligned} \min\ & f(x,z) \\ \text{subject to}\ & h(x,z) \le 0 \\ & (x,z_k) \in \bigcup_{j \in \mathcal{J}} \Gamma_j^k \ \forall\, k = 1,\dots,K \\ & x \in \mathbb{R}^m \\ & z \in \mathcal{J}^K \end{aligned} \tag{P_1}$$

where $\Gamma_j^k := S_j^k \times \{j\}$ and $S_j^k \subset \mathbb{R}^m\ \forall\, j,k$. Unlike (P_0), where z_k are binary variables, in the disjunctive programming problem (P_1) we consider an arbitrary set of disjunctive terms indexed in $\mathcal{J}$.

It is worth noting that a somewhat parallel representation of optimization problems including logical implications in terms of boolean variables is given by the notion of *Generalized Disjunctive Programming* (GDP). For a recent review on the GDP modeling paradigms, we refer the reader to [27], and to [39,26] for the algorithmic aspects. In GDP terms, system (P_1) can be equivalently expressed as

$$\begin{aligned} \min\ & f(x) + \sum_{k=1}^{K} c_k \\ \text{subject to}\ & h(x) \le 0 \\ & \bigvee_{l=1}^{L_k} \begin{bmatrix} Y_l^k \\ g_l^k(x) \le 0 \\ c_k = \gamma_l^k \end{bmatrix} \quad \forall\, k = 1,\dots,K \\ & \Omega(Y) = true \\ & 0 \le x \le U \\ & c \in \mathbb{R}^K \\ & Y_l^k \in \{true, false\} \end{aligned} \tag{P_2}$$

where $\Omega(Y)$ denotes logical propositions, often but not necessarily assumed to be expressed in Conjunctive Normal Form [1]. Although technically equivalent, in the remainder of the paper we will work with system (P_1) and abandon the GDP representation (P_2).

Paper Contribution. The theory of disjunctive programming [5,6,18] allows to manage the union of bodies Γ_j^k as in (P_1), provided they are convex sets. However, the drawbacks are the difficulty of practically solving the resulting *Nonlinear Programming* problems (NLP)s and the fact that those NLPs are defined in lifted spaces, i.e., with an increased number of variables.

In this paper we review the relevant literature on mathematical optimization with logical implications, and we concentrate on the attempt of avoiding the issue of dealing with large NLPs. In particular, we review some existing results that allow to work in the original space of variables for two relevant special cases of single disjunctions with $|\mathcal{J}| = 2$. Then, we significantly extend these special cases by considering pairs of related disjunctions that lead to either a more general set S_k^j or to disjunctions with $|\mathcal{J}| = 3$.

Computational experiments comparing disjunctive programming formulations in the original space of variables with bigM ones show that the former are computationally viable and promising.

Paper Organization. The paper is organized as follows. In Section 2, we give a more detailed description of the techniques used to model logical implications and highlight the relevant literature. In Sections 3.1–3.2 we formally discuss the two special cases of (P_1) in which a way to avoid additional variables has been proposed in the literature. In Section 4 we discuss two new special cases and we present the main theoretical contribution of the paper. In Section 5 we report on an extensive computational experience on three of the four discussed special cases. Section 6 draws some conclusions and envisions some future algorithmic directions.

2. Disjunctive Convex Optimization

Most of the optimization literature dealing with systems of type (P_1) or alike is concerned with sets

$$S = \{x \in \mathbb{R}^m \mid g_i(x) \leq 0,\ i = 1, \dots, \ell\}, \tag{1}$$

which are described as the intersection of the level sets of ℓ given functions. (Here, we drop, for easiness of notation, the indices k and j.) With this in mind, the essence of a system like (P_1) is the fact that imposing one or more constraints of the form $g_i(x) \leq 0$ is linked to discrete decisions. There are two classical ways of modeling such a phenomenon in Mathematical Programming, leading to *Mixed-Integer Nonlinear Programming* models (MINLP)s or *Mixed-Integer Linear Programming* models (MILP)s, provided the involved functions are affine.

2.1. The bigM approach

The first modeling technique of expressing a logical implication, widely known as bigM method, is very straightforward and works as follows. Let us assume that the discrete decision linked to the constraint $g_i(x) \leq 0$ is modeled by the binary variable z_i. Then, in a mathematical programming formulation, one can impose the constraint

$$g_i(x) \leq M_i(1 - z_i), \tag{2}$$

where M_i is a very large positive constant. If $z_i = 1$, then constraint $g_i(x) \leq 0$ is imposed. Conversely, if $z_i = 0$, then constraint (2) is satisfied by any value of $x \in \mathcal{F}$, where $\mathcal{F}$ describes the feasible set for x (e.g., the feasible region of the underlying MI(N)LP), and provided that $M_i \geq \sup_{x \in \mathcal{F}} g_i(x)$. Such a requirement for the definition of M_i already leads to a major difficulty encountered in mathematical programming with indicator constraints: the quantity $\sup_{x \in \mathcal{F}} g_i(x)$ might not be easily (or not even at all) computable. This might be the case, for example, when the set $\mathcal{F}$ itself is unbounded. In such cases, setting a "reasonably" high value of M_i will usually do the job *in practice*, but there is no theoretical guarantee that the system with a bigM constraint (2) is actually equivalent to the original one. In fact, when the set $\mathcal{F}$ is unbounded, the indicator constraints might not be represented as a MILP (see [33] for a definition of sets that are MILP representable) and there might be no representation too by an MINLP with a convex feasible region (see [32] for an example). If, instead, one assumes that this difficulty does not occur, i.e., it is possible, in practice, to compute $\sup_{x \in \mathcal{F}} g_i(x)$ or at least an upper bound on its value, then the bigM method leads to a valid reformulation of (P_1). Nevertheless, even under this assumption, the bigM method has two main drawbacks. The first one is trivially rooted in numerical risks associated with choosing a bigM value such that $1/M_i$ comes close to the machine precision, or, in any case, to the tolerances that are used by any mathematical programming solver working in floating point arithmetics (see, e.g., [36]). The other drawback affects on a more algorithmic level the current generation of MI(N)LP solvers, i.e., the solution method that they implement. More precisely, MI(N)LP solvers heavily rely on the iterative solution of the continuous relaxation of the given MI(N)LP. At the same time, it is well-known that the bigM formulations involving constraints (2) are characterized by continuous relaxations whose tightness depends on the value of the M_i, but which are typically very weak, i.e., very far away from the optimal solution value (see, e.g., [41]). This is because, in the continuous relaxation, a value of the binary variable z_i very close to 1 is enough to satisfy a constraint, thus deactivating the original implication $g_i(x) \leq 0$. Note that, recently, a way of improving on both of the above difficulties is strengthening the M_i values within the branch-and-bound tree by domain propagation and separation of *local* cutting planes, i.e., cuts valid only within the subtree they have been generated [40]. Essentially, based on the branching decisions taken so far, one can compute a better estimate of the bigM values and replace constraints (2) with stronger versions, although only locally valid. This is a classical approach followed by Global Optimization solvers, and its use within MILP and convex MINLP solvers is currently under investigation [40].

2.2. The Disjunctive Approach

In this paper we are concerned with reformulating (P_1) as a MI(N)LP in a disjunctive programming fashion. The disjunctive programming paradigm has been originally proposed by Balas [5,6] for the special case of (1) defined as the intersection of the level sets of affine functions, i.e., polyhedral sets, and concerns the minimization of a linear function over the union of polyhedra. Dropping this restriction and extending the point of view to general convex functions, disjunctive programming becomes the minimization of a convex function over the union of convex sets. Note that this is slightly more general than (P_1): The convex sets Γ_k^j, whose union is taken in (P_1), already have a specific structure, while this is not true for the sets S_k^j. In the affine (linear) case, the theory of disjunctive programming also provided the foundation of lift-and-project cutting planes [7,8,22], where the problem of optimizing a linear function over the union of polyhedra is solved as a separation routine for devising linear inequalities used to strengthen the continuous relaxation of an MILP. Lift-and-project cuts are an ingredient of many existing commercial and open-source MILP solvers, like Cbc, CPLEX, XPRESS, to mention a few, and they are part of the default setting of CPLEX [43].

As anticipated, Balas' results have been extended in a highly influential paper by Ceria and Soares [18] to the case of optimizing over the union of sets (1), i.e., described as the intersection of the level sets of convex functions that are allowed to be nonlinear. The key ingredient to be able to optimize over the union of sets is the ability of describing the convex hull of this union. If this can be done by the level sets of convex functions, then the union $\bigcup_{j\in\mathcal{J}} \Gamma_k^j$, as defined by (P_1), can be replaced by *conv* $\left(\bigcup_{j\in\mathcal{J}} \Gamma_k^j\right)$, thus leading to an MINLP. This constitutes the second way of dealing with logical implications, and it is somewhat opposite to the bigM method. Indeed, the bigM drawback of weak continuous relaxations is obviously overcome because there is no tighter convex relaxation of a single disjunction than its convex hull.

The main result in [18] is precisely a theorem that shows how to build the convex hull of the union of a finite number of sets, each one of which is described by convex functions. Before stating that theorem we need to introduce the so-called *perspective function*, which plays an important role in it. Namely,

Definition 1 *For a given closed convex function $g : \mathbb{R}^m \to \mathbb{R}\cup\{\infty\}$, the perspective function $\tilde{g} : \mathbb{R}^{m+1} \to \mathbb{R}\cup\{\infty\}$ is defined as*

$$\tilde{g}(x,\lambda) = \begin{cases} \lambda \cdot g\left(\frac{x}{\lambda}\right), & \lambda > 0, \\ \infty, & \lambda \le 0. \end{cases} \tag{3}$$

The perspective of a closed convex function is known to be convex, but not necessarily closed. We have now all the required elements to state the main result of [18]. Namely,

Theorem 1 ([18]) *Let $C_j = \{x \in \mathbb{R}^m \mid g_{j,i}(x) \le 0, i = 1,\dots,\ell_j\} \ne \emptyset$, for $j \in \mathcal{J}$, and assume each $g_{j,i} : \mathbb{R}^m \to \mathbb{R}$ is a closed convex function. Then, conv $(\cup_{j\in\mathcal{J}} C_j) = proj_x cl(C)$, where*

$$C = \left\{ \begin{array}{ll} (x, x^1, \dots, x^{|\mathcal{J}|}, \lambda_1, \dots, \lambda_{|\mathcal{J}|}) \in \mathbb{R}^{(|\mathcal{J}|+1)m+|\mathcal{J}|}, & \\ x = \sum_{j\in\mathcal{J}} x^j, & \\ \tilde{g}_{j,i}(x^j, \lambda_j) \le 0, & i = 1,\dots,\ell_j, \ \forall\, j \in \mathcal{J}, \\ \sum_{j\in\mathcal{J}} \lambda_j = 1, & \\ \lambda_j \ge 0, & \forall\, j \in \mathcal{J} \end{array} \right\}.$$

The construction of Theorem 1 begins in a straightforward manner by creating copies x^j of the initial variable x, each one constrained to lie in one of the disjunctive sets C_j. Then, the original x is a convex combination of these copies with weights λ_j. In this way, the resulting bilinear terms $\lambda_j x^j$ lead to a set description of the convex hull that is, however, nonconvex. A simple transformation is then used to obtain the convex set C containing the perspective function. A big effort in [18] is the insight that the closure of the perspective function also captures the elements in the convex hull that have zero weights λ_j for some $j \in \mathcal{J}$.

Although Theorem 1 provides a complete formal description of the convex hull of the union of the convex sets we are concerned with, it bears two main practical difficulties. First, the closure of the perspective function does not have an algebraic representation in general. In other words, the perspective function becomes non-differentiable for $\lambda_j \to 0$. This is a significant difference with respect to the linear case, where differentiability with respect to λ_j can be recovered by trivial algebra. In the nonlinear case, the resulting numerical difficulties were already noticed by Stubbs and Mehrotra [42] in the process of designing a branch-and-cut algorithm for general 0–1 mixed convex programming problems, based on lift-and-project cuts separated through an alternative version of Theorem 1. It is worth noting that for more than ten years there was no significant progress in the separation of lift-and-project cuts for general mixed-integer convex programming problems. More recently, two major steps towards the effective use of disjunctive cuts for mixed-integer convex programming have been made by Bonami [14] and Kılınç, Linderoth and Luedtke [35,34], which circumvent the non-differentiability issue algorithmically through the solution of "some" (potentially many) easier optimization problems. (The reader is referred to [16] and [11] for recent surveys on disjunctive cuts for MINLPs and applications and extensions of disjunctive inequalities, respectively.) However, in the more general setting of Theorem 1, i.e., not restricted to the separation of cutting planes, there has been and still is active research going on regarding the non-differentiability issue, see, e.g., [26,25].

The second difficulty associated with the practical use of Theorem 1 is the fact that the initial system is lifted into a space with a multiple dimension because a copy of the initial space is created for each disjunctive set. Note that the set C in Theorem 1 is a subset of $\mathbb{R}^{(|\mathcal{J}|+1)m+|\mathcal{J}|}$, i.e., it is defined in a space whose dimension makes the optimization over it computationally challenging. The focus of the present paper is on reviewing and providing new results that deal with this second difficulty. This is done by projecting out additional variables, which is, of course, a hard task in its most general form, but possible in several special cases that are presented in the next two sections.

3. Working in the original space of variables: single disjunctions

In this section we review two special cases in which the convex hull description of Theorem 1 can be projected onto the space of the original variables. More precisely, we assume disjunctions are "unrelated" and we consider a single disjunction at a time (thus, we then drop the index k throughout the section). The considered special cases have $|\mathcal{J}| = 2$ and are such that one term of the disjunction is either a single point (Section 3.1) or a box (Section 3.2).

Further, in the following (including Section 4) we assume that the subset of variables needed to express the disjunction is of dimension n. In other words, for $j \in \mathcal{J}$, we assume the sets S_j to be subsets of $\mathbb{R}^n$. Note that this might be a subspace of the original space $\mathbb{R}^m$ in (P_1). The results in this section and those in Section 4 can be used in a program like (P_1) by lifting everything into the original space $\mathbb{R}^m$. Bold letters like $\mathbf{u}$ stand for constant vectors in $\mathbb{R}^n$, and $(\mathbf{u})_i$ for its i-th component.

3.1. Constraint vs. nucleus

We consider here the case in which either a subset of variables (those involved in the disjunction) is forced to zero, or a set of constraints is imposed. This occurs in many applications, for example uncapacitated facility location with quadratic costs [28], stochastic service systems design [21], scheduling with controllable processing times [3] or the unit commitment problem [24]. In the notation of (P_1), this special case can be written as

$S_0 = \{x \in \mathbb{R}^n \mid x = \mathbf{0}\}$, and
$S_1 = \{x \in \mathbb{R}^n \mid \mathbf{l}_1 \le x \le \mathbf{u}_1, g_{1,i}(x) \le 0,\ i = 1, \dots, \ell\}$.

In this situation, Theorem 1 translates into

Theorem 2 ([29]) *Let $\mathcal{J} = \{0,1\}$ and for $j \in \mathcal{J}$ define $\Gamma_j := S_j \times \{j\}$, with S_j specified as above and non-empty. We then have that if Γ_1 is a convex set, $conv(\Gamma_0 \cup \Gamma_1) = cl(\Gamma)$, where*

$$\Gamma = \left\{ \begin{array}{ll} (x,z) \in \mathbb{R}^{n+1}, & \\ \tilde{g}_{1,i}(x,z) \le 0, & i = 1, \dots, \ell, \\ z\mathbf{l}_1 \le x \le z\mathbf{u}_1, & \\ 0 < z \le 1 & \end{array} \right\}.$$

In this first step, a fundamental property that persists throughout all the results in the following sections can be seen. The indicator variable z of the initial implication (or disjunction) takes over the role of the weight or multiplier inside the perspective function $\tilde{g}_i$. We make two remarks.

- Theorem 2 can also be formulated for the situation in which on the zero side, the variables are not forced to zero but to an arbitrary single point. Then, everything can be shifted to $\mathbf{0}$.
- Theorem 2 is also extendable to the situation in which on the zero side the set S_0 is not a single point, but a ray [29]. The two corresponding sets would be

$$\hat{S}_0 = \{(x, y) \in \mathbb{R}^{n+1} \mid x = \mathbf{0}, y \geq 0\}, \text{ and}$$
$$\hat{S}_1 = \{(x, y) \in \mathbb{R}^{n+1} \mid \mathbf{l}_1 \leq x \leq \mathbf{u}_1, g_{1,i}(x) \leq 0,\ i = 1, \ldots, \ell, g_{1,\ell+1}(x) \leq y\}.$$

If the form of the functions $g_{1,i}$ is known in more detail, in some cases one can avoid the need to take the closure and thus the differentiability issue, for example for polynomial functions [29].

The remarkable computational advantages of Theorem 2 are discussed, for example, in [29,24].

3.2. On/Off constraint

We consider now a slightly more general case where the set S_0 is not a point but a box. Namely,

$$S_0 = \{x \in \mathbb{R}^n \mid \mathbf{l}_0 \leq x \leq \mathbf{u}_0\}, \text{ and}$$
$$S_1 = \{x \in \mathbb{R}^n \mid \mathbf{l}_1 \leq x \leq \mathbf{u}_1, g_1(x) \leq 0\}.$$

We note that the special case of Section 3.1 with $\ell = 1$ is obtained by setting $\mathbf{l}_0 = \mathbf{u}_0 = \mathbf{0}$. Again, the above case occurs in several applications, for example, in telecommunication for the so-called delay-constrained routing problem [12].

In order to formulate an extension of Theorem 2 (under an additional condition), we need the following definition.

Definition 2 ([31]) *A function $g : \mathbb{R}^n \to \mathbb{R}$ is called independently non-decreasing (resp. non-increasing) in the i-th coordinate, if $\forall\, x \in dom(g)$ and $\forall\, \lambda > 0$, we have $g(x+\lambda e_i) \geq g(x)$ (resp. $g(x+\lambda e_i) \leq g(x)$). We say that g is independently monotone in the i-th coordinate, if it is either independently non-decreasing or non-increasing. Finally, g is called isotone, if it is independently monotone in every coordinate.*

For any subset $I \subset N := \{1, \ldots, n\}$ we denote by $\bar{I}$ the complement of I in N, i.e., $\bar{I} := N \setminus I$. For an isotone function g we denote by $J_1(g)$ (resp. $J_2(g)$), the set of indices in which g is independently non-decreasing (resp. independently non-increasing). If a function g is constant in coordinate i, it is both independently non-decreasing and non-increasing. In such a case, we assume that the index i is arbitrarily assigned to exactly one of the sets $J_1(g)$ and $J_2(g)$. In this way, we always get a partition of the index set, i.e., for any isotone function g, $J_1(g) \,\dot{\cup}\, J_2(g) = N$.

From now on, we assume the existence of an underlying set of closed convex functions $\{g_j : \mathbb{R}^n \to \mathbb{R} \mid j \in \mathcal{J}\}$ with associated bounds $\mathbf{l}_j, \mathbf{u}_j$ and define for all $I \subset N$, $x \in \mathbb{R}^n$, $z > 0$ and $j, j' \in \mathcal{J}$ the function $h_I^{j,j'} : \mathbb{R}^{n+1} \mapsto \mathbb{R}^n$, with

$$\left(h_I^{j,j'}\right)_i (x, z) := \begin{cases} (\mathbf{l}_j)_i & i \in I \cap J_1(g_j), \\ (\mathbf{u}_j)_i & i \in I \cap J_2(g_j), \\ x_i - \frac{(1-z)\left(\mathbf{u}_{j'}\right)_i}{z} & i \in \bar{I} \cap J_1(g_j), \\ x_i - \frac{(1-z)\left(\mathbf{l}_{j'}\right)_i}{z} & i \in \bar{I} \cap J_2(g_j), \end{cases}$$

for $i = 1, \ldots, n$, and $q_I^{j,j'} = g^j \circ h_I^{j,j'}$.

We can now state the main result of [31] in our slightly different notation.

Theorem 3 ([31]) *Let $\mathcal{J} = \{0, 1\}$ and for $j \in \mathcal{J}$ define $\Gamma_j := S_j \times \{j\}$, with S_j specified as above and non-empty. We then have that if g_1 is isotone, $conv(\Gamma_0 \cup \Gamma_1) = cl(\Gamma')$, where*

$$\Gamma' = \begin{Bmatrix} (x, z) \in \mathbb{R}^{n+1}, & \\ z q_I^{1,0}(\frac{x}{z}, z) \leq 0, & \forall\, I \subset N, \\ z\mathbf{l}_1 + (1-z)\mathbf{l}_0 \leq x \leq z\mathbf{u}_1 + (1-z)\mathbf{u}_0, & \\ 0 < z \leq 1 & \end{Bmatrix}.$$

Hijazi et al. [31] have shown that formulating the delay-constrained routing problem [12] by means of Theorem 3 leads to a significant computational advantage over a straightforward bigM formulation.

It is worth discussing the trade-off between applying Theorem 3 versus Theorem 1. On the one side, the advantage of Theorem 3 is the fact that we project back onto the original space of variables. In fact, we can express $conv(\Gamma_0 \cup \Gamma_1) = cl(\Gamma')$ as a subset of a $(n+1)$-dimensional space. A direct application of Theorem 1 would lead to expressing the convex hull as the projection of a subset of a $(3n+5)$-dimensional space. On the other side, this gain does not come for free. In Theorem 3, we need exponentially many constraints. In particular, including all the simple bound constraints, we have $2^n + 2n + 2$ constraints opposed to $4n + 9$ that would result from a direct application of Theorem 1. Similar considerations apply to the results presented in the following sections. In essence, an upper bound on the number of constraints needed to describe the convex hull is always exponential, although only in the number of variables involved in the constraint. In the following, we will see that sometimes it is possible to detect redundant constraints among the exponentially many ones. Moreover, these many constraints could be potentially added in a cutting-plane fashion. In Section 5, we will also show some concrete examples that illustrate the growth of the number of constraints when applying the results presented in this paper, depending on the application, and in particular on the number of variables that are involved in the disjunction.

3.2.1. On/Off linear constraint. We now assume that g_1 is an affine function. Such functions are easily seen to be isotone, and thus Theorem 3 is still valid. However, the constraints needed to describe the convex hull have a form that allows to avoid the need to take the closure. For any affine function $g_j(x) = a_{j,0} + \sum_{i=1}^n a_{j,i}x_i$, $j' \in \mathcal{J}$ and $I \subset N$ we define the linear function

$$H_I^{j,j'}(x,z) := \sum_{i\in\bar{I}} a_{j,i}x_i + z\left(a_{j,0} + \sum_{i\in I, a_{j,i}>0} a_{j,i}\left(\mathbf{l}_j\right)_i + \sum_{i\in I, a_{j,i}<0} a_{j,i}\left(\mathbf{u}_j\right)_i\right) - (1-z)\left(\sum_{i\in\bar{I}, a_{j,i}>0} a_{j,i}\left(\mathbf{u}_{j'}\right)_i + \sum_{i\in\bar{I}, a_{j,i}<0} a_{j,i}\left(\mathbf{l}_{j'}\right)_i\right).$$

It is easy to check that, for example, $H_I^{1,0}(x,z) = zq_I^{1,0}(\frac{x}{z}, z)$. Then, Theorem 3 reads as

Corollary 1 ([30]) *Consider the situation of Theorem 3. If g_1 is affine, $conv(\Gamma_0 \cup \Gamma_1) = \Gamma''$, where*

$$\Gamma'' = \begin{Bmatrix} (x,z) \in \mathbb{R}^{n+1}, & \\ H_I^{1,0}(x,z) \le 0, & \forall\ I \subset N, \\ z\mathbf{l}_1 + (1-z)\mathbf{l}_0 \le x \le z\mathbf{u}_1 + (1-z)\mathbf{u}_0, & \\ 0 \le z \le 1 & \end{Bmatrix}.$$

Observation 1 *In this special case of g_1 being an affine function, we note that in Corollary 1 we can restrict to subsets $I \subset \tilde{N} := \{i \in N \mid a_{1,i} \neq 0\}$.*

Observation 2 *Consider the case where S_1 is described by several linear constraints, for example $S_1 = \{x \in \mathbb{R}^n | Dx \le b, \mathbf{l}_1 \le x \le \mathbf{u}_1\}$ with $D \in \mathbb{R}^{m\times n}$. If the coefficients of D are monotonous for every column, i.e., if for all $j \in N$, and for any pair $i, i' \in \{1,\dots,m\}$, $d_{ij}d_{i'j} \ge 0$ holds, then Corollary 1 can be naturally extended.*

Observation 3 *Consider the context of Corollary 1 and assume further that*

$$S_1 \subset S_0. \tag{4}$$

This is equivalent to saying $proj_x\Gamma_1 \subset proj_x\Gamma_0$ and holds, for example, if $\mathbf{l}_0 = \mathbf{l}_1$ and $\mathbf{u}_0 = \mathbf{u}_1$, which is the case in many applications. Furthermore, it is easy to see that the non-redundant facets of Γ'' among the set of inequalities $H_I^{1,0}(x,z) \le 0$ indexed in $I \subset \tilde{N}$ are binding on at least one point $(x',1) \in \Gamma_1$, i.e., $H_I^{1,0}(x',1) = 0$. Now, if $\frac{\partial H_I^{1,0}}{\partial z}(x,z)$ was negative, then we would have $H_I^{1,0}(x',0) > 0$, which *contradicts the fact that $proj_x \Gamma_1 \subset proj_x \Gamma_0$ and that $H_I^{1,0}(x,z)$ is valid for Γ_0. Thus, in order to describe $conv(\Gamma_0 \cup \Gamma_1)$ in Corollary 1, under assumption (4) we can further restrict to subsets $I \in \hat{N}$, where*

$$\hat{N} := \left\{ I \subset \tilde{N} \;\middle|\; \frac{\partial H_I^{1,0}}{\partial z}(x,z) \geq 0 \right\}.$$

The derivatives are also easy to compute. Namely,

$$\frac{\partial H_I^{1,0}}{\partial z}(x,z) = a_{1,0} + \sum_{i \in I, a_{1,i}>0} a_{1,i} (\mathbf{l}_1)_i + \sum_{i \in I, a_{1,i}<0} a_{1,i} (\mathbf{u}_1)_i + \sum_{i \in \bar{I}, a_{1,i}>0} a_{1,i} (\mathbf{u}_0)_i + \sum_{i \in \bar{I}, a_{1,i}<0} a_{1,i} (\mathbf{l}_0)_i .$$

To the best of our knowledge, there is no computational investigation so far on the quality of the bound achievable by the reformulation based on Corollary 1. We will show in Section 5.1 some results on it by solving supervised classification problems [17] with the standard bigM formulation and with the perspective one.

4. Working in the original space of variables: pairs of related disjunctions

In this section we consider the case where two disjunctions of the type introduced in Section 3.2 are related to each other. Of course, we are still interested in characterizing the cases in which the convex hull description of Theorem 1 can be projected onto the space of the original variables. Namely, in Section 4.1 we show how to deal with the case in which two disjunctions are "complementary", i.e., precisely one of their associated indicator variables must take value one. This is a significant generalization of the result in Section 3.2 also because it can be interpreted as if both terms of a binary disjunction are actual constraints given as the level sets of a single function. Moreover, in Section 4.2 we consider the case in which "at most" one of the indicator variables associated with a pair of related disjunctions can take value one and we examine its relationship with a ternary disjunction.

4.1. Complementary disjunctions

We now consider a pair of disjunctions of the type discussed in Section 3.2, which are complementary to each other in the sense that the indicator constraint $g_0(x) \leq 0$ described by the first disjunction is deactivated when the indicator constraint of the second disjunction, $g_1(x) \leq 0$, is active and vice versa. To describe this situation we consider the sets

$S_0 = \{x \in \mathbb{R}^n \mid \mathbf{l}_0 \leq x \leq \mathbf{u}_0\}$,
$S_1 = \{x \in \mathbb{R}^n \mid \mathbf{l}_1 \leq x \leq \mathbf{u}_1, g_1(x) \leq 0\}$,
$\bar{S}_0 = \{x \in \mathbb{R}^n \mid \mathbf{l}_0 \leq x \leq \mathbf{u}_0, g_0(x) \leq 0\}$ and
$\bar{S}_1 = \{x \in \mathbb{R}^n \mid \mathbf{l}_1 \leq x \leq \mathbf{u}_1\}$.

We denote by z the indicator variable associated with the first two sets: when it is equal to one, the constraint $g_1(x) \leq 0$ is active. Conversely, the indicator variable $\bar{z}$ is associated with the second pair of sets: when it is equal to one, the constraint $g_0(x) \leq 0$ is active. The complementarity is assured by the fact that the two indicators have to sum up to one. Therefore, we define

$$H^= := \{(z, \bar{z}) \mid z + \bar{z} = 1\} \text{ and } L^= := \mathbb{R}^n \times H^=.$$

Then, we can prove the following result.

Theorem 4 *Let $\mathcal{J} = \{0,1\}$ and for $j \in \mathcal{J}$ define $\Gamma_j := S_j \times \{j\} \times [0,1]$, with S_j specified as above and non-empty and $\bar{\Gamma}_j := \bar{S}_j \times [0,1] \times \{1-j\}$, with $\bar{S}_j$ specified as above and non-empty. If g_0 and g_1 are isotone functions with $J_1(g_0) = J_2(g_1)$, then $conv\left((\Gamma_0 \cup \Gamma_1) \cap \left(\bar{\Gamma}_0 \cup \bar{\Gamma}_1\right) \cap L^=\right) = conv\left(\Gamma_0 \cup \Gamma_1\right) \cap conv\left(\bar{\Gamma}_0 \cup \bar{\Gamma}_1\right) \cap L^= = cl(\Gamma^*)$, where*

$$\Gamma^* = \left\{ \begin{array}{l} (x, z, \bar{z}) \in \mathbb{R}^{n+2}, \\ \left. \begin{array}{l} z q_I^{1,0}(\frac{x}{z}, z) \leq 0 \\ \bar{z} q_I^{0,1}(\frac{x}{\bar{z}}, \bar{z}) \leq 0 \end{array} \right\} \forall\, I \subset N, \\ z\mathbf{l}_1 + (1-z)\mathbf{l}_0 \leq x \leq z\mathbf{u}_1 + (1-z)\mathbf{u}_0, \\ 0 < z, \bar{z} < 1, \\ z + \bar{z} = 1 \end{array} \right\}.$$

Proof. We will first show the second equality above. First of all, we note that

$$\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1 = ((S_0 \times \{0\}) \cup (S_1 \times \{1\})) \times [0,1]$$

and thus

$$conv\,(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1) = conv\,((S_0 \times \{0\}) \cup (S_1 \times \{1\})) \times [0,1].$$

To the convex hull on the right-hand-side one can apply Theorem 3 and then lift it to its cartesian product with $[0,1]$ and the same applies to $\bar{\mathit{\Gamma}}_0 \cup \bar{\mathit{\Gamma}}_1$. By intersecting the two resulting sets and $L^=$, one gets exactly the set $cl(\mathit{\Gamma}^*)$ after noting that the constraints:

$$z\mathbf{l}_1 + (1-z)\mathbf{l}_0 \le x \le z\mathbf{u}_1 + (1-z)\mathbf{u}_0$$

and

$$\bar{z}\mathbf{l}_0 + (1-\bar{z})\mathbf{l}_1 \le x \le \bar{z}\mathbf{u}_0 + (1-\bar{z})\mathbf{u}_1$$

are identical after substituting $\bar{z}$ by $1-z$.

Now, we come to the first equality and prove it by showing two set inclusions. First, the left-hand-side set is trivially included in the right-hand-side one. The converse set inclusion requires a bit more work. Note that we can write

$$(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1) \cap \left(\bar{\mathit{\Gamma}}_0 \cup \bar{\mathit{\Gamma}}_1\right) \cap L^= = \left(\bar{S}_0 \times \{0\} \times \{1\}\right) \cup \left(S_1 \times \{1\} \times \{0\}\right).$$

Using this identity and according to the classical result in [18], we can formulate $conv\,\left((\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1) \cap \left(\bar{\mathit{\Gamma}}_0 \cup \bar{\mathit{\Gamma}}_1\right) \cap L^=\right)$ as the projection onto $(x, z, \bar{z})$ of the closure of

$$\begin{aligned}\big\{(x, x^0, x^1, z, z^0, z^1, \bar{z}, \bar{z}^0, \bar{z}^1, \lambda^0, \lambda^1) \in \mathbb{R}^{3n+8} \mid\ & x = x^0 + x^1, z = z^0 + z^1, \bar{z} = \bar{z}^0 + \bar{z}^1, \lambda^0 + \lambda^1 = 1,\\ & \tilde{g}_0(x^0, \lambda^0) \le 0, \tilde{g}_1(x^1, \lambda^1) \le 0, \lambda^0\mathbf{l}_0 \le x^0 \le \lambda^0\mathbf{u}_0,\\ & \lambda^1\mathbf{l}_1 \le x^1 \le \lambda^1\mathbf{u}_1, z^0 = 0, z^1 = \lambda^1, \bar{z}^0 = \lambda^0, \bar{z}^1 = 0, \lambda^0, \lambda^1 > 0\big\}.\end{aligned}$$

By eliminating $z^0 = 0$ and $\bar{z}^1 = 0$, identifying $z = z^1 = \lambda^1$ and $\bar{z} = \bar{z}^0 = \lambda^0$, and substituting $x^0 = x - x^1$, we obtain

$$\begin{aligned}\big\{(x, x^1, z, \bar{z}) \in \mathbb{R}^{2n+2} \mid\ & \tilde{g}_0(x - x^1, \bar{z}) \le 0, \tilde{g}_1(x^1, z) \le 0, x - \bar{z}\mathbf{u}_0 \le x^1 \le x - \bar{z}\mathbf{l}_0,\\ & z\mathbf{l}_1 \le x^1 \le z\mathbf{u}_1, 0 < z, \bar{z} < 1, z + \bar{z} = 1\big\}.\end{aligned}$$

Define the above set as $\hat{\mathit{\Gamma}}$ and for ease of notation substitute x^1 by y, that is,

$$\hat{\mathit{\Gamma}} = \left\{\begin{array}{l}(x, x^1, z, \bar{z}) \in \mathbb{R}^{2n+2},\\ zg_1(y/z) \le 0,\\ \bar{z}g_0\left((x-y)/\bar{z}\right) \le 0,\\ x - \bar{z}\mathbf{u}_0 \le y \le x - \bar{z}\mathbf{l}_0,\\ z\mathbf{l}_1 \le y \le z\mathbf{u}_1,\\ 0 < z, \bar{z} < 1\end{array}\right\}.$$

It remains to show that $cl(\mathit{\Gamma}^*) \subset proj_{(x,z,\bar{z})}cl(\hat{\mathit{\Gamma}})$. Therefore, let $(x, z, \bar{z}) \in \mathit{\Gamma}^*$ and define $y \in \mathbb{R}^n$ as

$y_i = \max\{z\,(\mathbf{l}_1)_i\,, x_i - \bar{z}\,(\mathbf{u}_0)_i\}\ \forall\ i \in J_1(g_1)$, and
$y_i = \min\{z\,(\mathbf{u}_1)_i\,, x_i - \bar{z}\,(\mathbf{l}_0)_i\}\ \forall\ i \in J_2(g_1)$.

Then, one can check that there is a set $I \subset N$ such that $h_I^{1,0}(x/z, z) = y/z$. Furthermore, taking into account that $J_2(g_1) = J_1(g_0)$, we can also check that $h_{\bar{I}}^{0,1}(x/\bar{z}, \bar{z}) = \frac{x-y}{\bar{z}}$. Because $zq_I^{1,0}(x/z, z) = zg_1(y/z) \le 0$ and $\bar{z}q_{\bar{I}}^{0,1}(x/\bar{z}, \bar{z}) = \bar{z}g_0\left((x-y)/\bar{z}\right) \le 0$, we deduce that $g_1(y/z) \le 0$ and $g_0\left((x-y)/\bar{z}\right) \le 0$. The lower and upper bounds on y are easily checked to hold and we have that $(x, y, z, \bar{z}) \in \hat{\mathit{\Gamma}}$.
We now come to the closure. Consider any point $(x, z, \bar{z}) \in cl(\mathit{\Gamma}^*)$ and let $(x_k, z_k, \bar{z}_k)$ be a sequence of points in $\mathit{\Gamma}^*$ such that $\lim_{k\to\infty}(x_k, z_k, \bar{z}_k) = (x, z, \bar{z})$. For every $k \in \mathbb{N}$ one can define y_k as above to get a point $(x_k, y_k, z_k, \bar{z}_k) \in \hat{\mathit{\Gamma}}$, and y_k converges to some $y \in \mathbb{R}^n$. Thus, $\lim_{k\to\infty}(x_k, y_k, z_k, \bar{z}) = (x, y, z, \bar{z}) \in cl(\hat{\mathit{\Gamma}})$. With this, the proof is complete. □

The interpretation of Theorem 4 is somewhat surprising. Under the technical condition that the functions g_0 and g_1 be isotone, and that in addition $J_2(g_1) = J_1(g_0)$, the theorem shows that computing the convex hull of the intersection of the two disjunctions does not allow any improvement with respect to intersecting the individual convex hulls of the two single disjunctions considered in isolation. Without this technical condition, this may not be true (see Example 2). Of course, Theorem 4 also shows that everything can be done in the original space of variables.

Observation 4 *Because the two disjunctions above are complementary, i.e., $\bar{z} = 1 - z$, the whole situation could be described as a single disjunction with a single indicator variable, that is, as a subset of $\mathbb{R}^{n+1}$. For example,*

$$(x, z) \in \left(\bar{S}_0 \times \{0\}\right) \cup (S_1 \times \{1\}) .$$

Its convex hull is then just the projection of the closure of Γ^* from Theorem 4 onto the first $n + 1$ variables, which is trivial to compute. The fact that the whole situation can be managed by projecting out $\bar{z}$ was already somewhat anticipated in the proof of Theorem 4.

We formalize Observation 4 in the following corollary.

Corollary 2 *Let $\mathring{\Gamma}_0 := \bar{S}_0 \times \{0\}$ and $\mathring{\Gamma}_1 := S_1 \times \{1\}$, with $\bar{S}_0$ and S_1 specified as above and non-empty. Thus, we have that if $J_2(g_1) = J_1(g_0)$, then $conv(\mathring{\Gamma}_0 \cup \mathring{\Gamma}_1) = cl(\mathring{\Gamma})$, where*

$$\mathring{\Gamma} = \left\{ \begin{array}{l} (x, z) \in \mathbb{R}^{n+1}, \\ \left.\begin{array}{l} z q_I^{1,0}(\frac{x}{z}, z) \le 0, \\ (1 - z) q_I^{0,1}(\frac{x}{1-z}, 1 - z) \le 0, \end{array}\right\} \forall\ I \subset N, \\ z\mathbf{l}_1 + (1 - z)\mathbf{l}_0 \le x \le z\mathbf{u}_1 + (1 - z)\mathbf{u}_0, \\ 0 < z < 1 \end{array} \right\} .$$

The condition about the index sets of the isotone functions required for Theorem 4 and Corollary 2 is interpreted in the following observation.

Observation 5 *If $J_2(g_1) = J_1(g_0)$, then there is a vertex v of the rectangle $[\mathbf{l}_0, \mathbf{u}_0]$ such that $g_0(v) \le 0$. In fact, because $\bar{S}_0 \ne \emptyset$, there is a $x^0 \in [\mathbf{l}_0, \mathbf{u}_0]$ with $g_0(x^0) \le 0$. If we set*

$$v_i := \begin{cases} (\mathbf{l}_0)_i & i \in J_1(g_0) \\ (\mathbf{u}_0)_i & i \in J_2(g_0) \end{cases},$$

then $g_0(v) \le g_0(x^0) \le 0$. Furthermore, because also $S_1 \ne \emptyset$, there is a $x^1 \in [\mathbf{l}_1, \mathbf{u}_1]$ with $g_1(x^1) \le 0$. If we then define $w \in [\mathbf{l}_1, \mathbf{u}_1]$ as the opposing vertex, i.e.,

$$w_i := \begin{cases} (\mathbf{u}_1)_i & v_i = (\mathbf{l}_0)_i \\ (\mathbf{l}_1)_i & v_i = (\mathbf{u}_0)_i \end{cases},$$

due to $J_2(g_1) = J_1(g_0)$, we get that $g_1(w) \le g_1(x^1) \le 0$. Thus, the resulting convex hull contains at least two such opposing vertices of the underlying hyperractangles.

We now give an example of Corollary 2.

Example 1 *Let $\mathbf{l}_0 = \mathbf{l}_1 = \mathbf{0} \in \mathbb{R}^2$ and $\mathbf{u}_0 = \mathbf{u}_1 = \mathbf{1} \in \mathbb{R}^2$, and consider the functions $g_0(x_1, x_2) = \exp\left(2x_1 - \frac{9}{5}\right) + x_2 - 1$ and $g_1(x_1, x_2) = \max\left\{\frac{1}{2} - x_1, \frac{1}{2} - x_2\right\}$. One can show that they are both isotone and that $J_1(g_0) = J_2(g_1) = \{1, 2\}$ and $J_2(g_0) = J_1(g_1) = \emptyset$. The two disjunctive sets*

$$\{x \in \mathbb{R}^2 \mid \mathbf{l}_0 \le x \le \mathbf{u}_0, g_0(x) \le 0\} \times \{0\} \text{ and } \{x \in \mathbb{R}^2 \mid \mathbf{l}_1 \le x \le \mathbf{u}_1, g_1(x) \le 0\} \times \{1\}$$

are shown in Figure 1(a). Figures 1(b) and 1(c), respectively show the non-redundant boundaries of the convex hull of

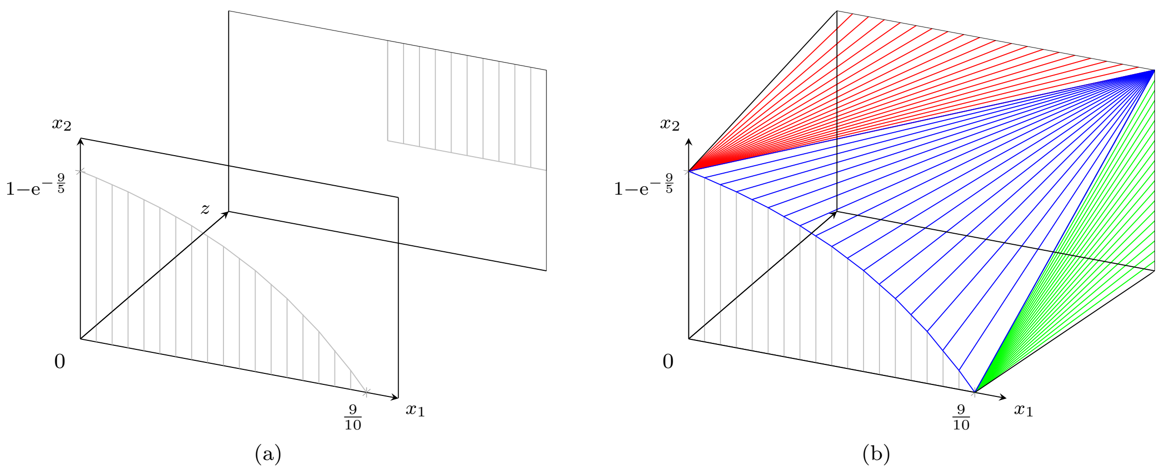
$$\{x \in \mathbb{R}^2 \mid \mathbf{l}_0 \le x \le \mathbf{u}_0, g_0(x) \le 0\} \times \{0\} \text{ and } \{x \in \mathbb{R}^2 \mid \mathbf{l}_1 \le x \le \mathbf{u}_1\} \times \{1\}$$

on the one hand, and those of the convex hull of

$$\{x \in \mathbb{R}^2 \mid \mathbf{l}_1 \le x \le \mathbf{u}_1, g_1(x) \le 0\} \times \{1\} \text{ and } \{x \in \mathbb{R}^2 \mid \mathbf{l}_0 \le x \le \mathbf{u}_0\} \times \{0\}$$

on the other. The convex hulls in Figures 1(b) and 1(c) are obtained by Theorem 3 and Corollary 1, respectively. Figure 1(d) then shows the intersection of the latter two, giving the convex hull of the sets

$$\{x \in \mathbb{R}^2 \mid \mathbf{l}_0 \le x \le \mathbf{u}_0, g_0(x) \le 0\} \times \{0\} \text{ and } \{x \in \mathbb{R}^2 \mid \mathbf{l}_1 \le x \le \mathbf{u}_1, g_1(x) \le 0\} \times \{1\}.$$



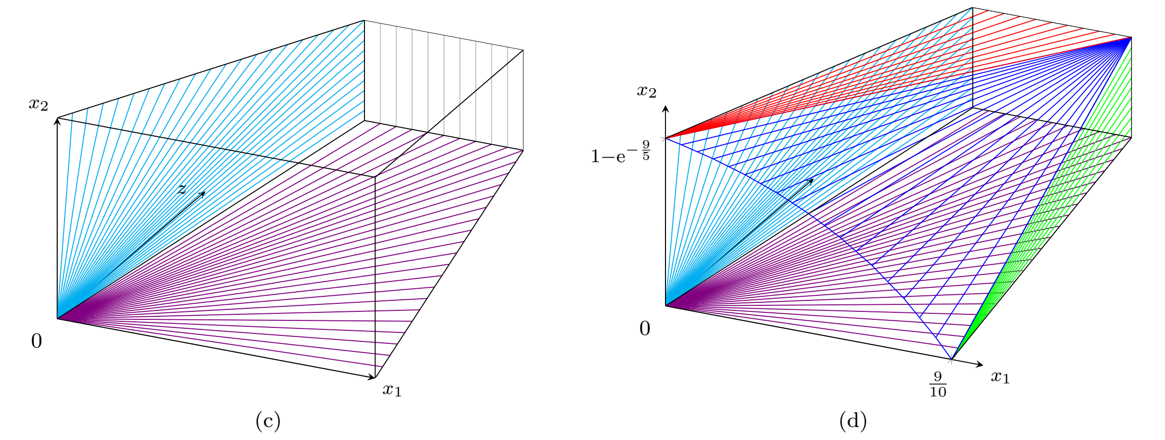


Fig. 1: Illustration of the construction of the convex hull of two disjunctive sets.

Now, we switch from $\mathbb{R}^2$ to $\mathbb{R}$ in order to see what happens when the condition $J_2(g_1) = J_1(g_0)$ is not fulfilled.

Example 2 *Let $l^0 = l^1 = 0$ and $u^0 = u^1 = 1$, and consider the two functions $H^0(x) = x - \frac{1}{2}$ and $H^1(x) = x - \frac{3}{4}$, both of which are non-decreasing in their first and only coordinate. Figures 2(a) and 2(b) show the convex hull when considering one on/off constraint at a time. In Figure 2(c) we see that their intersection does not give the convex hull of the two complementary indicator constraints.*

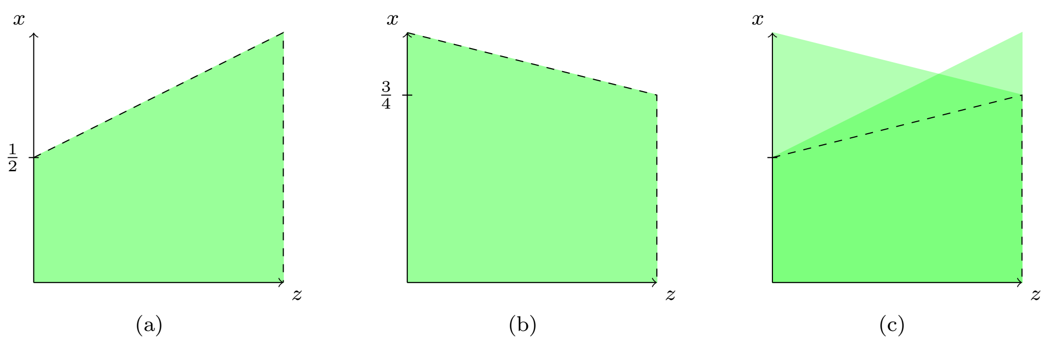


Fig. 2: Counterexample for the case in which $J_2(g_1) \neq J_1(g_0)$.

Informally, the condition $J_2(g_1) = J_1(g_0)$ can be described as the fact that one function is non-increasing in those coordinates where the other one is non-decreasing and vice versa. This holds true in a number of applications, starting from the well-known (and already mentioned) job shop scheduling problem [13], where for each pair of jobs i, j and each machine k, either i is executed on k before j or vice versa.

In Section 5.2 we will compare the quality of the bound provided by the disjunctive reformulation obtained by applying Corollary 2 above and the straightforward bigM formulation for the job shop scheduling problem.

4.2. "Almost" complementary disjunctions

In this section we will start again from the setting of Section 4.1 but weaken the requirement on the relation between the two disjunctions. In particular, we relax the constraint $z + \bar{z} = 1$ by imposing $z + \bar{z} \leq 1$ instead. This means that in addition to the two possibilities of activating one constraint at a time, we also allow that none of them is active. By slightly modifying the notation from Section 4.1, we will consider the three sets

$S_0 = \{x \in \mathbb{R}^n \mid \mathbf{l}_0 \leq x \leq \mathbf{u}_0\}$,
$S_1 = \{x \in \mathbb{R}^n \mid \mathbf{l}_1 \leq x \leq \mathbf{u}_1, g_1(x) \leq 0\}$, and
$S_{-1} = \{x \in \mathbb{R}^n \mid \mathbf{l}_{-1} \leq x \leq \mathbf{u}_{-1}, g_{-1}(x) \leq 0\}$.

Again, we denote by z the indicator variable that activates the constraint $g_1(x) \leq 0$, and by $\bar{z}$ the one that activates the constraint $g_{-1}(x) \leq 0$. Moreover, we extend (4) as

$$S_1 \subset S_0 \text{ and } S_{-1} \subset S_0. \tag{5}$$

From an application perspective, such a ternary case holds in several interesting contexts, for example, for the well-known traveling salesman problem with time windows (TSPTW) [20]. There, for each pair of cities i, j, either city j is visited *immediately after* city i (say, S_1), or *immediately before* (say, S_{-1}), or the visit happens "far from" i (say, S_0). If we define

$$H^{\leq} := \{(z, \bar{z}) \mid z + \bar{z} \leq 1\} \text{ and } L^{\leq} := \mathbb{R}^n \times H^{\leq},$$

and for $j \in \mathcal{J} = \{0, 1\}$

$$\mathit{\Gamma}_j := S_j \times \{j\} \times [0, 1] \text{ and } \bar{\mathit{\Gamma}}_j := S_{-j} \times [0, 1] \times \{j\},$$

then the situation can be modeled as the intersection of two disjunctions and $L^{\leq}$, namely

$$(x, z, \bar{z}) \in (\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1) \cap \left(\bar{\mathit{\Gamma}}_0 \cup \bar{\mathit{\Gamma}}_1\right) \cap L^{\leq}. \tag{6}$$

Observation 6 *It is easy to check that disjunction* (6) *is equivalent to the ternary disjunction*

$$(x, z, \bar{z}) \in (S_0 \times \{0\} \times \{0\}) \cup (S_1 \times \{1\} \times \{0\}) \cup (S_{-1} \times \{0\} \times \{1\}).$$

In this case, we don't know a description of the convex hull in the original space of variables but using Theorem 3, we can compute a superset of the convex hull of the disjunction (6):

Corollary 3 *Let S_0, S_1 and S_{-1} be specified as above and non-empty. Thus, we have that if g_{-1} and g_1 are isotone functions, then $conv\left((\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1) \cap \left(\bar{\mathit{\Gamma}}_0 \cup \bar{\mathit{\Gamma}}_1\right) \cap L^{\leq}\right) \subseteq conv\,(\mathit{\Gamma}_0 \cup \mathit{\Gamma}_1) \cap conv\left(\bar{\mathit{\Gamma}}_0 \cup \bar{\mathit{\Gamma}}_1\right) \cap L^{\leq} = cl(\breve{\mathit{\Gamma}})$, where*

$$\breve{\mathit{\Gamma}} = \left\{ \begin{array}{l} (x, z, \bar{z}) \in \mathbb{R}^{n+2}, \\ \left. \begin{array}{l} z q_I^{1,0}(\frac{x}{z}, z) \leq 0 \\ \bar{z} q_I^{-1,0}(\frac{x}{\bar{z}}, \bar{z}) \leq 0 \end{array} \right\} \forall\, I \subset N, \\ z\mathbf{l}_1 + (1 - z)\mathbf{l}_0 \leq x \leq z\mathbf{u}_1 + (1 - z)\mathbf{u}_0, \\ \bar{z}\mathbf{l}_{-1} + (1 - \bar{z})\mathbf{l}_0 \leq x \leq \bar{z}\mathbf{u}_{-1} + (1 - \bar{z})\mathbf{u}_0, \\ 0 < z, \bar{z} < 1, \\ z + \bar{z} \leq 1 \end{array} \right\}.$$

Proof. The set inclusion is a simple observation, while the set equality follows by Theorem 3. □

In Section 5.3 we will discuss some computational results comparing the straightforward bigM formulation of the traveling salesman problem with time windows with the disjunctive one based on Corollary 3 above.

4.2.1. Ternary disjunctions. Of course, a ternary case like the one described in the previous section can be managed by means of a unique indicator variable, say $\tilde{z}$, which will lead to (P_1) with $\mathcal{J} = \{-1, 0, 1\}$. It is natural to ask, on the one side, if it is possible to write the counterpart of Theorem 3 for this special case, and, on the other hand, which is the relationship between these two ways of representing essentially the same disjunction. The answer to the first question is positive under the technical condition of the functions g_{-1} and g_1 being affine (discussed later), while that to the second one is once again a bit surprising: we will show in the following that working with two distinct indicators (as in Section 4.2) is stronger than with one indicator variable only.

To deal with the ternary case with a unique indicator variable we need to extend the definition of the set $\hat{N}$ of Observation 3 to pairs of indices $j, j' \in \mathcal{J} = \{-1, 0, 1\}$:

$$\hat{N}^{j,j'} := \left\{ I \subset \tilde{N} \,\middle|\, a_{j,0} + \sum_{i \in I, a_{j,i} > 0} a_{j,i} \left(\mathbf{l}_j\right)_i + \sum_{i \in I, a_{j,i} < 0} a_{j,i} \left(\mathbf{u}_j\right)_i + \sum_{i \in \bar{I}, a_{j,i} > 0} a_{j,i} \left(\mathbf{u}_{j'}\right)_i + \sum_{i \in \bar{I}, a_{j,i} < 0} a_{j,i} \left(\mathbf{l}_{j'}\right)_i \geq 0. \right\}$$

Then, we can prove the following result.

Theorem 5 *Let $\mathcal{J} = \{-1, 0, 1\}$ and for $j \in \mathcal{J}$ define $\Gamma_j := S_j \times \{j\}$, with S_j specified as above and non-empty. Moreover, let us assume that (5) holds and that g_{-1} and g_1 are affine functions. Then, $conv(\Gamma_0 \cup \Gamma_1 \cup \Gamma_{-1}) = \Gamma'''$, where*

$$\Gamma''' = \begin{Bmatrix} (x, \tilde{z}) \in \mathbb{R}^{n+1}, & \\ H_I^{1,0}(x, \tilde{z}) \leq 0, & \forall\, I \in \hat{N}^{1,0}, \\ H_I^{-1,0}(x, -\tilde{z}) \leq 0, & \forall\, I \in \hat{N}^{-1,0}, \\ \tilde{z}\mathbf{l}_1 + (1 - \tilde{z})\mathbf{l}_0 \leq x \leq \tilde{z}\mathbf{u}_1 + (1 - \tilde{z})\mathbf{u}_0, & \\ -\tilde{z}\mathbf{l}_{-1} + (1 + \tilde{z})\mathbf{l}_0 \leq x \leq -\tilde{z}\mathbf{u}_{-1} + (1 + \tilde{z})\mathbf{u}_0, & \\ -1 \leq \tilde{z} \leq 1 & \end{Bmatrix}.$$

Proof. Define the two relaxed sets

$$\tilde{\Gamma}_1 = \begin{Bmatrix} (x, \tilde{z}) \in \mathbb{R}^{n+1}, & \\ H_I^{1,0}(x, \tilde{z}) \leq 0, & \forall\, I \in \hat{N}^{1,0}, \\ \tilde{z}\mathbf{l}_1 + (1 - \tilde{z})\mathbf{l}_0 \leq x \leq \tilde{z}\mathbf{u}_1 + (1 - \tilde{z})\mathbf{u}_0, & \\ -1 \leq \tilde{z} \leq 1 & \end{Bmatrix}$$

and

$$\tilde{\Gamma}_{-1} = \begin{Bmatrix} (x, \tilde{z}) \in \mathbb{R}^{n+1}, & \\ H_I^{-1,0}(x, -\tilde{z}) \leq 0, & \forall\, I \in \hat{N}^{-1,0}, \\ -\tilde{z}\mathbf{l}_{-1} + (1 + \tilde{z})\mathbf{l}_0 \leq x \leq -\tilde{z}\mathbf{u}_{-1} + (1 + \tilde{z})\mathbf{u}_0, & \\ -1 \leq \tilde{z} \leq 1 & \end{Bmatrix},$$

and note that $\Gamma''' = \tilde{\Gamma}_1 \cap \tilde{\Gamma}_{-1}$. By Corollary 1 and Observation 3, because (4) holds for $(1, 0)$ and $(-1, 0)$, then $\tilde{\Gamma}_1 \cap (\mathbb{R}^n \times [0, 1]) = conv(\Gamma_0 \cup \Gamma_1)$ and $\tilde{\Gamma}_{-1} \cap (\mathbb{R}^n \times [-1, 0]) = conv(\Gamma_0 \cup \Gamma_{-1})$. Thus, $(\Gamma_0 \cup \Gamma_1) \subset \tilde{\Gamma}_1$ and $(\Gamma_0 \cup \Gamma_{-1}) \subset \tilde{\Gamma}_{-1}$.

By the first of these two set inclusions, for any $I \in \hat{N}^{1,0}$, $H_I^{1,0}(x, \tilde{z}) \leq 0$ is a valid inequality for Γ_0. That is, $H_I^{1,0}(v, 0) \leq 0$ for any $v \in proj_x \Gamma_0$. Since $proj_x \Gamma_{-1} \subset proj_x \Gamma_0$ and $\frac{\partial H_I^{1,0}}{\partial \tilde{z}}(x, \tilde{z}) \geq 0$, we deduce that $H_I^{1,0}(w, -1) \leq 0$ for all $w \in proj_x \Gamma_{-1}$, i.e., that $H_I^{1,0}(x, \tilde{z}) \leq 0$ is valid for Γ_{-1}. In a similar way, because (4) holds for $(1, 0)$, one can show that the bound inequalities in $\tilde{\Gamma}_1$ are valid for Γ_{-1}. Hence, we deduce that $(\Gamma_0 \cup \Gamma_1 \cup \Gamma_{-1}) \subset \tilde{\Gamma}_1$.

In an analogue fashion, noting that $\frac{\partial H_I^{-1,0}}{\partial (-\tilde{z})}(x, -\tilde{z}) \geq 0$ for all $I \in \hat{N}^{-1,0}$ and that (4) holds for $(-1, 0)$, one gets $(\Gamma_0 \cup \Gamma_1 \cup \Gamma_{-1}) \subset \tilde{\Gamma}_{-1}$.

All in all follows that $(\Gamma_0 \cup \Gamma_1 \cup \Gamma_{-1}) \subset \left(\tilde{\Gamma}^1 \cap \tilde{\Gamma}_{-1}\right) = \Gamma'''$, and because Γ''' is clearly a convex set, $conv\left(\Gamma_0 \cup \Gamma^1 \cup \Gamma_{-1}\right) \subset \Gamma'''$.

In order to see the converse set inclusion, let $(x, \tilde{z}) \in \tilde{\Gamma}_1 \cap \tilde{\Gamma}_{-1}$. If $\tilde{z} \in [0,1]$, by Corollary 1 we get $(x, \tilde{z}) \in conv\,(\Gamma_0 \cup \Gamma_1)$. Otherwise, $(x, \tilde{z}) \in conv\,(\Gamma_0 \cup \Gamma_{-1})$. In either case, $(x, \tilde{z}) \in conv\,(\Gamma_0 \cup \Gamma_1 \cup \Gamma_{-1})$ and we conclude that $conv\,(\Gamma_0 \cup \Gamma_1 \cup \Gamma_{-1}) = \Gamma'''$. □

Theorem 5 is formulated for affine functions because for general isotone functions this would require a definition of the perspective functions for negative values of the multiplier λ (namely, the indicator $\tilde{z}$) different from the definition given in (3). To the best of our knowledge, such a definition has not been proposed yet. This is due to the fact that the perspective function with a negative multiplier looses the convexity-preserving property.

In order to compare the two ways of modeling the ternary disjunction proposed in the present and in the previous section, we maintain the more restrictive assumption of affine functions required in Theorem 5, under which Corollary 3 is, of course, still valid. Therefore, we define

$H := \{(z, \bar{z}, \tilde{z}) \mid \tilde{z} = z - \bar{z}\}$ and $L := \mathbb{R}^n \times H$,
$\Gamma_0 := S_0 \times \{0\} \times \{0\} \times [-1,1]$,
$\Gamma_1 := S_1 \times \{1\} \times \{0\} \times [-1,1]$,
$\Gamma_{-1} := S_{-1} \times \{0\} \times \{1\} \times [-1,1]$,
$\tilde{\Gamma}_0 := S_0 \times [0,1] \times [0,1] \times \{0\}$,
$\tilde{\Gamma}_1 := S_1 \times [0,1] \times [0,1] \times \{1\}$, and
$\tilde{\Gamma}_{-1} := S_{-1} \times [0,1] \times [0,1] \times \{-1\}$.

and consider the two disjunctions

$$(x, z, \bar{z}, \tilde{z}) \in (\Gamma_0 \cup \Gamma_1 \cup \Gamma_{-1}) \cap L \text{ and } (x, z, \bar{z}, \tilde{z}) \in \left(\tilde{\Gamma}_0 \cup \tilde{\Gamma}_1 \cup \tilde{\Gamma}_{-1}\right) \cap L.$$

These two are equivalent in terms of indicator constraints, in the sense that either $z, \bar{z}$ (together) or $\tilde{z}$ alone indicate the activity of the involved constraints, and with the trivial transformation $\tilde{z} = z - \bar{z}$ one can switch from one case to the other without losing information. However, in terms of sets, the two disjunctions are not the same. By Corollary 3 and Theorem 5 we can (after projecting out either $\tilde{z}$ or $z, \bar{z}$, respectively) compute a superset of the convex hull of the first disjunction above and the convex hull itself of the second disjunction above, and then lift them back into $\mathbb{R}^{n+3}$ and intersect with L. The relation of the two resulting sets is characterized by the following corollary.

Corollary 4 *$conv\,(\Gamma_0 \cup \Gamma_1 \cup \Gamma_{-1}) \cap L \subseteq \dot{\Gamma}$ and $conv\left(\tilde{\Gamma}_0 \cup \tilde{\Gamma}_1 \cup \tilde{\Gamma}_{-1}\right) \cap L = \check{\Gamma}$, where*

$$\dot{\Gamma} = \left\{\begin{array}{ll} (x, z, \bar{z}, \tilde{z}) \in \mathbb{R}^{n+3} & \\ H_I^{1,0}(x, z) \le 0, & \forall\ I \in \hat{N}^{1,0}, \\ H_I^{-1,0}(x, \bar{z}) \le 0, & \forall\ I \in \hat{N}^{-1,0}, \\ z\mathbf{l}_1 + (1-z)\mathbf{l}_0 \le x \le z\mathbf{u}_1 + (1-z)\mathbf{u}_0, & \\ \bar{z}\mathbf{l}_{-1} + (1-\bar{z})\mathbf{l}_0 \le x \le \bar{z}\mathbf{u}_{-1} + (1-\bar{z})\mathbf{u}_0, & \\ 0 < z, \bar{z} < 1, & \\ -1 < \tilde{z} < 1, & \\ z + \bar{z} \le 1, & \\ \tilde{z} = z - \bar{z} & \end{array}\right\}$$

and

$$\check{\Gamma} = \left\{\begin{array}{ll} (x, z, \bar{z}, \tilde{z}) \in \mathbb{R}^{n+3}, & \\ H_I^{1,0}(x, \tilde{z}) \le 0, & \forall\ I \in \hat{N}^{1,0}, \\ H_I^{-1,0}(x, -\tilde{z}) \le 0, & \forall\ I \in \hat{N}^{-1,0}, \\ \tilde{z}\mathbf{l}_1 + (1-\tilde{z})\mathbf{l}_0 \le x \le \tilde{z}\mathbf{u}_1 + (1-\tilde{z})\mathbf{u}_0, & \\ -\tilde{z}\mathbf{l}_{-1} + (1+\tilde{z})\mathbf{l}_0 \le x \le -\tilde{z}\mathbf{u}_{-1} + (1+\tilde{z})\mathbf{u}_0, & \\ 0 < z, \bar{z} < 1, & \\ -1 < \tilde{z} < 1, & \\ \tilde{z} = z - \bar{z} & \end{array}\right\}.$$

In addition, $\dot{\Gamma} \subseteq \check{\Gamma}$. Thus, $conv\,(\Gamma_0 \cup \Gamma_1 \cup \Gamma_{-1}) \cap L \subseteq conv\left(\tilde{\Gamma}_0 \cup \tilde{\Gamma}_1 \cup \tilde{\Gamma}_{-1}\right) \cap L$.

Proof. The set $\dot{\Gamma}$ follows immediately from $\check{\Gamma}$ in the statement of Corollary 3 and Observation 3. The inclusion $conv\left(\Gamma_0 \cup \Gamma_1 \cup \Gamma_{-1}\right) \cap L \subseteq \dot{\Gamma}$ follows from the Corollary.

Similarly, the set equality $conv\left(\tilde{\Gamma}_0 \cup \tilde{\Gamma}_1 \cup \tilde{\Gamma}_{-1}\right) \cap L = \check{\Gamma}$ follows from Theorem 5.

The set inclusion $\dot{\Gamma} \subseteq \check{\Gamma}$ can be seen by the fact that the two sets are described by the same constraints apart from $z + \bar{z} \leq 1$ and the constraints indexed in $I \in \hat{N}^{1,0}$ and $I \in \hat{N}^{-1,0}$, respectively. Even if one added the former constraint, $z + \bar{z} \leq 1$, to the definition of $\check{\Gamma}$, note that the latter set of constraints is more restricting in $\dot{\Gamma}$, because, for example, $\frac{\partial H_I^{1,0}}{\partial z} \geq 0$, but $z \geq \tilde{z}$ for all $(x, z, \bar{z}, \tilde{z}) \in \dot{\Gamma}$. Similarly, $\frac{\partial H_I^{-1,0}}{\partial -z} \geq 0$, but $\bar{z} \geq -\tilde{z}$ for any $(x, z, \bar{z}, \tilde{z}) \in \dot{\Gamma}$. □

The computation in Section 5.3 shows that the set inclusion $\dot{\Gamma} \subseteq \check{\Gamma}$ can be strict, i.e., $\dot{\Gamma} \subset \check{\Gamma}$.

5. Computational Experiments

In this section we computationally compare on several problems in the literature, exposing logical implications in their definition, the quality of the continuous relaxation of the straightforward bigM formulation vs. the perspective one implemented in the original space of variables by making use of the results in the previous sections. We express the quality of the lower bound, i.e., the solution of the continuous relaxation, in terms of percentage gap with respect to the optimal solution value (or the best known value in case the instance has never been solved to optimality). In addition, we compute and report the percentage gap of the lift-and-project closure [15] of the two distinct formulations. Indeed, in all problems we consider in the following the binary variables we apply the lift-and-project closure to are indicator variables and it is interesting to analyze the impact on the strength of the closure of using the stronger formulation obtained by disjunctive programming with respect to the straightforward bigM one.

In the next sections we do not compare the time spent to solve the continuous relaxation of the disjunctive formulation with that of the bigM. The theoretical results in the previous sections allow us to write the disjunctive formulations in the original space of variables with some limited increase in the number of constraints. Thus, the two formulations are computationally comparable in terms of computing times and we are concerned with assessing their strength in terms of bound.

Note that in the case of linear functions, as mentioned earlier, there are no differentiability issues, i.e., taking the closures in any of the theorems in the previous sections just amounts to take $0 \leq z \leq 1$, which can be implemented in a straightforward fashion.

We also note that in the case of linear indicator constraints, exactly one of the inequalities given by the disjunctive formulation (the one for $I = \emptyset$, see [30]) coincides with the bigM constraint with the value of M chosen as tight as possible. In this way, the disjunctive formulation can be seen as the bigM formulation augmented by a number of valid inequalities.

5.1. Linear On/Off Constraint: supervised classification

Consider a set Ω of m objects, where each object $i \in \Omega$ is characterized by the vector $x_i \in \mathbb{R}^d$ and associated with one of two classes, labeled by $y_i \in \{-1, 1\}$. The task is to find a hyperplane $\omega^T x \leq b$ in $\mathbb{R}^d$ that separates the two classes by maximizing a "confidence" margin. Because it is not always possible to find such a hyperplane, for each object i that is misclassified, one models the violation of the associated constraint $y_i(\omega^T x_i - b) \leq 0$ with a continuous slack variable ξ_i whose value is penalized in the objective function. Recently, Brooks [17] suggested that the misclassification penalty must be upper bounded and beyond that bound a fixed penalty must be paid. Of course, this model introduces a logical implication that is modeled, for each point i, with a binary variable and a bigM constraint in [17]. This is the linear On/Off constraint case discussed in Section 3.2.1 because, for each point i either a linear constraint is active (say, S_1) or the variables are restricted into a box (say, S_0). We give the problem formulation as in [17], but we explicitly write indicator constraints, namely

$$
\begin{aligned}
\min \quad & \frac{\omega^T\omega}{2} + \frac{C}{m}\left(\sum_{i=1}^{m}\xi_i + 2\sum_{i=1}^{m}(1-z_i)\right) \\
& z_i = 1 \implies y_i(\omega^T x_i + b) - 1 + \xi_i \geq 0 \qquad \forall\, i = 1,\dots,m \\
& 0 \leq \xi_i \leq 2 \qquad \forall\, i = 1,\dots,m \\
& \omega \in \mathbb{R}^d \\
& b \in \mathbb{R} \\
& z \in \{0,1\}^m,
\end{aligned}
$$

where C is some non-negative constant. The continuous decision variables $\omega \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are unbounded (and free), and, in order to restrict them into a box (for the S_0 case) we apply some MILP preprocessing in two versions, say "weak" and "strong", see [10] for details.

We note that if each of the indicator constraints above is rewritten as a bigM constraint, this results in a set of m constraints. This is opposed to a set of $m \cdot 2^{d+2}$ constraints resulting from Theorem 3. However, we observed that the number of constraints can in practice be reduced significantly by means of Observation 3.

In Table 1 and Table 2 we report the percentage gap of the continuous relaxation of the bigM formulation in [17] ("bigM lp"), and of the continuous relaxation of the perspective reformulation ("CH lp") for 32 classification instances proposed in [17] strengthened by the "weak" and "strong" preprocessing, respectively. Moreover, Table 1 and Table 2 report the percentage gap of the lift-and-project closure

instance	bigM lp	CH lp	P_e bigM	P_e CH	P_e^* bigM	P_e^* CH
1nl_1	99.67	99.67	97.93	97.90	14.88	18.99
1nl_2	99.59	99.59	97.93	97.93	7.76	13.29
1nl_3	99.90	99.90	99.70	99.70	20.22	14.46
1nl_4	98.23	98.23	89.14	88.98	6.54	15.11
1nl_5	98.24	98.24	91.40	91.38	6.39	8.99
2nl_10	99.57	99.57	96.09	95.98	14.55	14.23
2nl_6	99.88	99.88	98.19	98.18	5.56	8.33
2nl_8	99.21	99.21	85.08	84.76	0.00	3.95
3nl_11	99.77	99.77	99.14	99.13	10.55	10.15
3nl_12	99.82	99.82	99.49	99.49	11.49	8.62
3nl_13	99.83	99.83	99.38	99.38	19.24	19.30
3nl_14	99.87	99.87	99.58	99.58	49.84	24.46
3nl_15	99.83	99.83	99.50	99.50	27.22	24.23
4nl_16	99.76	99.76	98.10	98.05	9.50	7.57
4nl_17	1.32	0.00	0.00	0.00	0.00	0.00
5nl_10	99.76	99.76	98.77	98.76	11.49	12.00
5nl_11	99.78	99.78	98.56	98.55	18.24	12.63
5nl_12	99.83	99.83	99.09	99.08	28.57	14.84
5nl_13	99.73	99.73	98.65	98.65	13.59	3.79
5nl_14	99.75	99.75	98.24	98.22	19.23	18.14
5nl_15	99.80	99.80	98.95	98.94	12.53	19.57
5nl_16	99.68	99.68	98.35	98.35	12.09	9.70
5nl_17	99.72	99.72	98.13	98.13	12.88	21.57
5nl_1	99.78	99.78	98.92	98.92	8.47	9.31
5nl_2	99.78	99.78	98.52	98.51	17.23	16.90
5nl_3	99.84	99.84	99.13	99.12	16.04	18.17
5nl_4	99.77	99.77	98.86	98.86	18.72	19.21
5nl_5	99.79	99.79	98.49	98.48	15.57	21.53
5nl_6	99.85	99.85	99.17	99.17	13.96	10.98
5nl_7	99.77	99.77	98.85	98.85	5.88	7.85
5nl_8	99.78	99.78	98.47	98.46	17.55	30.94
5nl_9	99.84	99.84	99.12	99.12	14.84	12.49
mean	96.58	96.54	94.65	94.62	14.39	14.10

Table 1: Percentage gaps of bigM and disjunctive formulations on supervised classification instances [17] with "weak" preprocessing [10].

computed over the bigM ("P_e bigM") and disjunctive ("P_e CH") formulations. Finally, we also report the percentage gap of the *strengthened* lift-and-project closures ("P_e^* bigM" and "P_e^* CH", respectively), i.e., the closure computed by strengthening each separated lift-and-project cut through the classical procedure of Balas and Jeroslow [9].

instance	bigM lp	CH lp	P_e bigM	P_e CH	P_e^* bigM	P_e^* CH
`2nl_10`	9.12	1.33	0.00	0.00	0.00	0.00
`2nl_6`	0.01	0.00	0.00	0.00	0.00	0.00
`3nl_13`	93.28	82.92	63.03	30.66	1.59	0.09
`3nl_14`	0.01	0.00	0.00	0.00	0.00	0.00
`3nl_15`	93.95	93.93	67.18	64.10	0.85	1.93
`4nl_16`	0.24	0.00	0.00	0.00	0.00	0.00
`4nl_17`	1.06	0.00	0.00	0.00	0.00	0.00
`5nl_14`	0.01	0.00	0.00	0.00	0.00	0.00
`mean`	24.71	22.27	16.28	11.85	0.31	0.25

Table 2: Percentage gaps of bigM and disjunctive formulations on supervised classification instances [17] with "strong" preprocessing [10].

The results in Table 1 computationally confirm the theoretical dominance of the disjunctive formulation with respect to the bigM one. However, the improvement is very small, almost negligible, and although it gets slightly higher if the lift-and-project closure is considered, still it does not look very promising to work with the disjunctive formulation. It is worth noting that also in terms of lift-and-project (unstrengthened) closures there is a theoretical domination of the disjunctive formulation with respect to the bigM one, while the same does not hold for the strengthened versions of the closure that can be seen as heuristic procedures for split cuts (see, e.g., [15]).

In Table 2 we show only the instances among those in Table 1 where the gap is not closed in at least one of the six columns. By considering a much more strengthened version of the initial model (with tighter bounds on ω and b) the quality of the bound provided by the disjunctive formulation looks better than that of the bigM one. More precisely, on the instances `4nl_16` and `4nl_17`, in which the bigM formulation shows a small but still significant gap, the disjunctive formulation improves significantly, showing no gap. On two out of three of the remaining "difficult" instances, namely `2nl_10` and `3nl_13`, the bound provided by the disjunctive reformulation is significantly better. On these last two instances the two formulations did not show any difference in Table 1, thus suggesting that the disjunctive reformulation takes advantage of the strengthening of the bounds of the ω and b variables.

If IBM-CPLEX 12.6.0 is used as a black-box solver for the two formulations we do not observe much of a difference in terms of computing times and branch-and-bound nodes. Precisely, in the case of the disjunctive formulation all constraints that are not present in the bigM one (see the remark immediately before Section 5.1) are added as "user cuts" to IBM-CPLEX, instead of imposing them as regular constraints. This allows the solver to decide a proper way of using the constraints, which is a conservative strategy in case no clear assessment on their strength is possible. As anticipated, no significant difference is observed.

5.2. Complementary disjunctions: (linear) job shop

In the classical job shop scheduling problem we are given n jobs, $J_1, \ldots, J_n$, that have to be scheduled without preemption on m machines, $M_1, \ldots, M_m$. Each job is characterized by its proper routing on the machines, as defined by matrix $O \in \mathbb{N}^{m \times n}$, i.e., O specifies the order in which a single job has to be executed on the different machines. More precisely, O_{kj} is the sequence number of machine M_k in the order of operations of job J_j. Furthermore, p_{kj} is the corresponding processing time. The objective is to minimize the makespan, i.e., the time in which the latest machine finishes processing on all machines. Because two jobs, say i, j, cannot overlap on a machine, say k, then a classical and quite weak bigM way of modeling job shop in MILP involves adding indicator variables x_{ij}^k and x_{ji}^k. If $x_{ij}^k = 1$, then i precedes j on machine k, thus $s_{kj} \geq s_{ki} + p_{ki}$, where s_{ki} is the starting time of (the operation of) job i on machine m. Otherwise, the temporal constraint on the starting time variables is deactivated. Of course, the same holds for x_{ji}^k, and $x_{ij}^k + x_{ji}^k = 1$, which is precisely the case of the complementary disjunctions discussed in Section 4.1. A formulation with one complementary binary variable for each triple (i, j, k) projected out and with indicator constraints is

$$
\begin{array}{lll}
\min\ C & & \\
& s_{k'i} \geq s_{ki} + p_{ki} & \forall\ (i,k,k')\ :\ O_{ki}+1 = O_{k'i} \\
& \left.\begin{array}{l} x_{ij}^k = 1 \implies s_{kj} \geq s_{ki} + p_{ki} \\ x_{ij}^k = 0 \implies s_{ki} \geq s_{kj} + p_{kj} \end{array}\right\} & \forall\ (i,j,k)\ :\ i<j \\
& C \geq s_{ki} + p_{ki} & \forall\ (i,k) \\
& s_{ki} \geq 0 & \forall\ (i,k) \\
& x_{ij}^k \in \{0,1\} & \forall\ (i,j,k)\ :\ i<j.
\end{array}
$$

Note that each indicator constraint could be rewritten by either a single bigM constraint or four constraints resulting from Theorem 4, one of which can be shown to be always redundant.

In Table 3 we consider 34 job shop instances from the literature [23, 2, 38] and we report the percentage gap of the continuous relaxation of the classical bigM formulation of the job shop ("bigM lp"), of the continuous relaxation of the perspective reformulation ("CH lp"), and of the four lift-and-project closures (precisely like in the tables of the previous section).

instance	bigM lp	CH lp	P_e bigM	P_e CH	P_e^* bigM	P_e^* CH
abz5-10x10	30.38	28.46	25.27	24.48	23.23	23.40
abz6-10x10	21.31	20.65	17.28	16.06	14.08	15.47
abz7-20x15	37.50	35.30	33.34	*32.42	*32.19	*32.44
abz8-20x15	33.78	31.75	30.53	*29.68	*30.35	*29.68
abz9-20x15	31.22	29.28	27.45	*27.03	*27.32	*26.96
ft06-6x6	14.54	14.54	14.25	13.47	11.86	12.59
ft10-10x10	29.56	28.12	25.68	24.12	22.04	23.93
ft20-20x5	66.78	64.52	63.24	61.56	60.86	60.75
la01-10x5	37.98	34.17	29.34	26.75	29.02	26.75
la02-10x5	39.84	39.70	37.40	34.95	35.45	33.01
la03-10x5	41.54	38.49	32.58	31.67	31.92	31.64
la04-10x5	37.45	34.05	27.95	26.17	27.16	26.16
la05-10x5	35.91	35.91	35.59	34.20	34.66	34.16
la06-15x5	55.39	52.65	49.78	49.01	49.31	47.91
la07-15x5	57.75	56.88	54.97	53.42	53.78	52.86
la08-15x5	57.24	53.56	49.06	47.70	47.03	47.47
la09-15x5	59.83	57.18	53.98	52.75	53.66	52.66
la10-15x5	53.75	52.98	51.30	49.94	50.30	49.32
la11-20x5	66.20	64.12	61.11	*60.47	60.92	60.42
la12-20x5	60.73	60.56	59.07	58.29	56.51	57.23
la13-20x5	66.78	63.97	60.69	58.99	60.08	*57.66
la14-20x5	65.71	64.97	62.04	61.17	60.62	60.80
la15-20x5	68.68	64.90	61.39	60.57	60.59	60.57
la16-10x10	24.12	22.49	20.09	19.80	20.03	19.51
la17-10x10	17.60	16.65	15.30	15.27	15.28	15.12
la18-10x10	21.81	20.89	18.39	17.95	17.81	17.65
la19-10x10	26.72	24.57	21.87	20.59	20.28	19.67
la20-10x10	16.18	16.18	15.58	15.04	13.83	14.72
la21-15x10	31.45	31.43	29.92	28.69	29.01	28.63
la22-15x10	33.22	30.72	27.26	25.63	27.26	25.59
la23-15x10	37.98	35.34	32.14	31.87	32.12	31.38
la24-15x10	24.70	24.65	23.49	22.11	21.78	21.87
la25-15x10	25.99	25.09	23.94	22.80	22.91	22.58
la26-20x10	41.13	39.18	36.59	*35.84	*36.27	*36.11
mean	40.31	38.64	36.11	35.01	34.98	34.60

Table 3: Percentage gaps of bigM and disjunctive formulations on 34 job shop instances [23, 2, 38]. The % gaps are computed with respect to the best known solution for instances `la23` and `la26`; a "*" indicates that the computation of the lift-and-project closure [15] hit the time limit of 2 CPU hours.

The results in Table 3 show that there is some advantage in using the disjunctive formulation with respect to the bigM one, although the improvement in the quality of the bound is not sufficient to make any of the 34 instances above easier enough for a general-purpose MILP solver if used as a black-box solver. More precisely, within 1 hour time limit IBM-CPLEX 12.6.0 solves to optimality the same 17 instances for both formulations and no dominance in performance is shown: the bigM formulation is slightly better in terms of arithmetic mean, while the reverse is true in geometric mean. However, we believe combining the disjunctive reformulation, lift-and-project cuts and effective special-purpose propagation (i.e., tailored for the job shop) could lead to an effective algorithm for optimally solving significant job shop instances.

5.3. "Almost" complementary disjunctions: (linear) TSPTW

In the traveling salesman problem with time windows one needs to find a minimum-cost tour visiting a set of cities exactly once, where each city i must be visited within a given time window $[a_i, b_i]$. To each pair of cities, say i, j, is associated a cost c_{ij} and a travel time t_{ij}, and the classical bigM formulation uses the binary variable x_{ij} that takes value one to indicate that city i is visited in a tour *immediately* before city j. The variable is used to express the cost of a tour and also as indicator to activate the temporal constraint $s_j \geq s_i + t_{ij}$, where s_i is the time in which city i is visited in the tour. We consider the asymmetric version of the problem, thus another binary variable x_{ji} is introduced for stating that j is visited in a tour immediately before i. Of course, only one of the two variables x_{ij} and x_{ji} can take value one in a feasible solution but, differently from the complementary case, the third option that both of them take value 0 is feasible, namely when none of the two cities immediately precedes the other. This is the "almost" complementary case discussed in Section 4.2. The variant of the TSPTW on a directed graph $G = (V, A)$ studied in [20] (i.e., a tour is a hamiltonian path from node p to node q) can be formulated with indicator constraints as

$$\begin{aligned}
\min \sum_{(i,j)\in A} c_{ij} x_{ij} & & \\
\sum_{j\in V\,:\,(i,j)\in A} x_{ij} = 1 & & \forall\, i \in V \setminus \{q\} \\
\sum_{k\in V\,:\,(k,i)\in A} x_{ki} = 1 & & \forall\, i \in V \setminus \{p\} \\
x_{ij} = 1 \implies s_j \geq s_i + t_{ij} & & \forall\, (i,j) \in A \\
a_i \leq s_i \leq b_i & & \forall\, i \in V \\
x_{ij} \in \{0,1\} & & \forall\, (i,j) \in A.
\end{aligned}$$

The number of constraints needed for reformulating each indicator constraint is exactly the same as in the case of the job shop problem in the previous section.

In Table 4 we consider 50 TSPTW instances from the literature [4], and we report the percentage gap of the continuous relaxation of the classical bigM formulation of the TSPTW ("bigMx lp"), of the continuous relaxation of the perspective reformulation based on Corollary 3 ("CHx lp"), and of the four lift-and-project closures. In addition to what reported in the tables of the previous sections, we also report the percentage gaps of the continuous relaxation of the bigM formulation where a ternary variable is used to model the disjunction as described in Section 4.2.1 ("bigMz lp") and the continuous relaxation of the associated perspective formulation according to Theorem 5 ("CHz lp"). For the latter two (weaker) cases, however, we do not compute the closures.

On the one side, the numbers in Table 4 confirm the dominance relationships discussed in Section 4.2.1, and, on the other side, show a neat advantage obtained by using the disjunctive formulation ("CHx lp") with respect to the bigM one ("bigMx lp"). Remarkably, this is also true for closures and the improvement becomes even more significant. As in the job shop case, however, there is no instance that is not solved with the bigM formulation, which is instead solved with the disjunctive formulation within the time limit of one hour using IBM-CPLEX 12.6.0. Nevertheless, over the 36 (out of 50) instances solved by IBM-CPLEX in both cases, there is a neat improvement of both the running times and the number of branch-and-bound nodes. Precisely, by considering only the 21 nontrivial instances for which each of the two formulations is solved in at least 0.5 secs, the running time goes from 107.4 seconds to 72.3 in arithmetic mean, and from 33.1 to 27.4 in geometric mean. In turn, the number of nodes changes from 524,557.7 to 382,049.7 in arithmetic mean, and from 140,002.1 to 96,037.7 in geometric mean. This corresponds to a rough 30% reduction in the number of nodes, which looks like a promising result.

instance	bigMz lp	CHz lp	bigMx lp	CHx lp	P_e bigMx	P_e CHx	P_e^* bigMx	P_e^* CHx
rbg010a	0.14	0.14	0.14	0.14	0.14	0.14	0.14	0.14
rbg016a	3.29	3.01	3.06	2.78	1.52	1.40	0.59	0.26
rbg016b	1.69	1.65	1.68	1.61	1.33	1.15	0.97	0.92
rbg017.2	0.81	0.79	0.81	0.79	0.64	0.19	0.00	0.00
rbg017	79.86	79.82	79.86	79.82	79.75	79.73	79.74	79.66
rbg017a	0.25	0.25	0.25	0.25	0.22	0.20	0.16	0.13
rbg019a	0.75	0.73	0.74	0.73	0.00	0.00	0.00	0.00
rbg019b	0.65	0.65	0.64	0.64	0.46	0.38	0.37	0.37
rbg019c	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
rbg019d	1.35	1.31	1.35	1.31	1.15	1.10	0.88	0.88
rbg020a	0.67	0.66	0.67	0.66	0.64	0.61	0.62	0.59
rbg021.2	0.68	0.68	0.68	0.68	0.68	0.68	0.68	0.68
rbg021.3	0.97	0.94	0.97	0.94	0.91	0.76	0.75	0.72
rbg021.4	0.90	0.90	0.90	0.90	0.86	0.75	0.68	0.68
rbg021.5	0.81	0.81	0.81	0.81	0.78	0.67	0.55	0.55
rbg021.6	0.13	0.13	0.13	0.13	0.12	0.11	0.09	0.09
rbg021.7	0.11	0.11	0.11	0.11	0.11	0.11	0.10	0.10
rbg021.8	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
rbg021.9	0.08	0.08	0.08	0.08	0.08	0.08	0.08	0.08
rbg021	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
rbg027a	0.74	0.74	0.74	0.74	0.74	0.74	0.69	0.70
rbg031a	1.54	1.53	1.54	1.53	1.29	1.01	1.08	0.93
rbg033a	1.43	1.36	1.42	1.33	1.11	0.87	0.90	0.49
rbg034a	1.31	1.25	1.29	1.22	1.03	0.82	0.72	0.64
rbg035a.2	0.54	0.51	0.53	0.50	0.45	0.39	0.33	0.34
rbg035a	2.33	2.27	2.32	2.25	1.87	1.15	1.21	0.83
rbg038a	0.52	0.52	0.52	0.52	0.37	0.25	0.34	0.23
rbg040a	4.50	4.40	4.48	4.37	3.92	2.97	2.92	2.55
rbg041a	3.87	3.85	3.87	3.85	3.50	3.06	2.76	2.62
rbg042a	2.33	2.24	2.32	2.23	1.86	1.45	1.41	1.22
rbg048a	1.35	1.35	1.35	1.35	1.34	1.31	1.33	1.29
rbg049a	1.22	1.22	1.22	1.22	1.19	1.18	1.16	1.15
rbg050a	0.84	0.82	0.84	0.82	0.76	0.57	0.72	0.49
rbg050b	1.06	1.06	1.06	1.06	1.05	0.98	1.03	0.95
rbg050c	0.69	0.68	0.69	0.68	0.67	0.63	0.62	0.61
rbg055a	0.98	0.98	0.98	0.98	0.95	0.87	0.87	0.74
rbg067a	0.80	0.80	0.80	0.80	0.77	0.70	0.69	0.55
rbg086a	0.94	0.93	0.94	0.92	0.81	0.66	0.72	0.64
rbg092a	0.88	0.86	0.88	0.86	0.81	0.70	0.72	0.66
rbg125a	1.37	1.32	1.37	1.31	1.13	0.96	0.94	0.76
rbg132.2	1.54	1.52	1.54	1.51	1.42	1.21	1.24	1.12
rbg132	2.22	2.13	2.21	2.13	1.87	1.34	1.46	1.09
rbg152.3	0.63	0.62	0.63	0.61	*0.58	*0.52	*0.53	*0.50
rbg152	1.30	1.24	1.28	1.22	*1.05	*0.70	*0.76	*0.50
rbg172a	1.65	1.57	1.64	1.56	1.36	*1.14	1.15	*0.91
rbg193.2	1.68	1.61	1.68	1.60	*1.50	*1.22	*1.23	*1.05
rbg193	1.67	1.55	1.67	1.54	*1.56	*1.20	*1.18	*1.01
rbg201a	2.03	1.93	2.02	1.92	1.88	*1.45	1.48	*1.27
rbg233.2	1.80	1.74	1.80	1.73	*1.63	*1.42	*1.39	*1.37
rbg233	2.04	1.91	2.03	1.88	*1.77	*1.55	*1.51	*1.38
mean	2.81	2.77	2.80	2.76	2.62	2.45	2.42	2.32

Table 4: Percentage gaps of bigM and disjunctive formulations on 50 TSPTW instances [4]; a "*" indicates that the computation of the lift-and-project closure [15] hit the time limit of 2 CPU hours.

6. Conclusions

In this paper we have reviewed the relevant literature on mathematical optimization with logical implications, and we concentrated on the attempt of avoiding the issue of dealing with large NLPs once a disjunctive formulation is used. In particular, we reviewed some existing results that allow to work in the original space of variables for two relevant special cases of single disjunctions with $|\mathcal{J}| = 2$. Then, we significantly extended these special cases by considering pairs of related disjunctions that lead to either single disjunctions with more general sets or to disjunctions with three terms.

Computational experiments comparing disjunctive programming formulations in the original space of variables with bigM ones show that the former are computationally viable and promising. This calls for further investigations in which disjunctive programming formulations should be used, with very hard

MILP problems like scheduling, routing with temporal constraints, etc., in conjunction with sophisticated algorithms involving aggressive cut generation within the branch-and-bound tree and bound reductions.

Acknowledgments

The second author acknowledges the support of MIUR, Italy, under the grant PRIN 2012. The fourth author acknowledges the support of the EU ITN 316647 "Mixed-Integer Nonlinear Optimization" (MINO). We are indebted to four anonymous referees for a careful reading and insightful suggestions.

References

1. Conjunctive Normal Form. In M. Hazewinkel, editor, *Encyclopedia of Mathematics*. Springer, 2001.
2. J. Adams, E. Balas, and D. Zawack. The Shifting Bottleneck Procedure for Job Shop Scheduling. *Management Science*, 34:391–401, 1983.
3. S. Aktürk, A. Atamtürk, and S. Gürel. A strong conic quadratic reformulation for machine-job assignment with controllable processing times. Technical Report BCOL Research Report 07.01, Industrial Engineering & Operations Research, University of California, Berkeley, April 2007.
4. N. Ascheuer. *Hamiltonian path problems in the on-line optimization of flexible manufacturing systems*. PhD thesis, Technische Universität Berlin, 1995.
5. E. Balas. Disjunctive programming. *Annals of Discrete Mathematics*, 5:3–51, 1979.
6. E. Balas. Disjunctive programming: Properties of the convex hull of feasible points. *Discrete Applied Mathematics*, 89:3–44, 1998.
7. E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58:259–324, 1993.
8. E. Balas, S. Ceria, and G. Cornuéjols. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, 42:1229–1246, 1996.
9. E. Balas and R. Jeroslow. Strengthening cuts for mixed integer programs. *European Journal of Operational Research*, 4:224–234, 1980.
10. P. Belotti, P. Bonami, M. Fischetti, A. Lodi, M. Monaci, A. Nogales-Gómez, and D. Salvagnin. On handling indicator constraints in mixed-integer programming. Technical Report OR/13/1, revised OR/14/20, DEI, University of Bologna, 2014.
11. P. Belotti, L. Liberti, A. Lodi, G. Nannicini, and A. Tramontani. Disjunctive inequalities: applications and extensions. In J.J. Cochran, editor, *Wiley Encyclopedia of Operations Research and Management Science*, volume 2, pages 1441–1450. John Wiley & Sons, Inc., 2011.
12. W. Ben-Ameur and A. Ouorou. Mathematical models of the delay-constrained routing problem. *Algorithmic Operations Research*, 1:94–103, 2006.
13. J. Błażewicz, E. Pesch, and M. Sterna. The disjunctive graph machine representation of the job shop scheduling problem. *European Journal of Operational Research*, 127:317–331, 2000.
14. P. Bonami. Lift-and-project cuts for mixed integer convex programs. In O. Günlük and G. Woeginger, editors, *Integer Programming and Combinatorial Optimization*, volume 6655 of *Lecture Notes in Computer Science*, pages 52–64. Springer Berlin / Heidelberg, 2011.
15. P. Bonami. On optimizing over lift-and-project closures. *Mathematical Programming Computation*, 4:151–179, 2012.
16. P. Bonami, J.T. Linderoth, and A. Lodi. Disjunctive cuts for mixed integer nonlinear programming problems. In R. Majoub, editor, *Progress in Combinatorial Optimization*, pages 521–544. Wiley/ISTE, 2011.
17. J. P. Brooks. Support vector machines with the ramp loss and the hard margin loss. *Operations Research*, 59:467–479, 2011.
18. S. Ceria and J Soares. Convex programming for disjunctive convex optimization. *Mathematical Programming*, 86:595–614, 1999.
19. G. Cornuéjols, M.L. Fisher, and G.L. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23:789–810, 1977.
20. S. Dash, O. Günlük, A. Lodi, and A. Tramontani. A time bucket formulation for the traveling salesman problem with time windows. *INFORMS Journal on Computing*, 24:132–147, 2012.
21. S. Elhedhli. Service System Design with Immobile Servers, Stochastic Demand, and Congestion. *Manufacturing & Service Operations Management*, 8:92–97, 2006.
22. M. Fischetti, A. Lodi, and A. Tramontani. On the separation of disjunctive cuts. *Mathematical Programming*, 128:205–230, 2011.
23. H. Fisher and G.L. Thompson. Probabilistic learning combinations of local job-shop scheduling rules. In J.F. Muth and G.L. Thompson, editors, *Industrial Scheduling*, pages 225–251. Prentice-Hall, Englewood Cliffs, 1963.
24. A. Frangioni and C. Gentile. Perspective cuts for a class of convex 0-1 mixed integer programs. *Mathematical Programming*, 106:225–236, 2006.
25. K. Furman, I. E. Grossmann, and N. Sawaya. A useful algebraic representation of disjunctive convex sets using the perspective function. Conference talk MINLP 2014, Pittsburgh.
26. I. E. Grossmann and S. Lee. Generalized Convex Disjunctive Programming: Nonlinear Convex Hull Relaxation. *Computational Optimization and Applications*, 26:83–100, 2003.
27. I. E. Grossmann and F. Trespalacios. Systematic Modeling of Discrete-Continuous Optimization Models through Generalized Disjunctive Programming. *AIChE Journal*, 59(9):3276–3295, 2013.
28. O. Günlük, J. Lee, and R. Weismantel. MINLP strengthening for separaable convex quadratic transportation-cost UFL. Technical Report RC24213 (W0703-042), IBM Research Division, March 2007.

29. O. Günlük and J. Linderoth. Perspective Reformulation and Applications. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 61–89. Springer New York, 2012.
30. H. Hijazi. *Mixed-Integer NonLinear Optimization approaches for Network Design in Telecommunications*. PhD thesis, Université d'Aix-Marseille II, 2010.
31. H. Hijazi, P. Bonami, G. Cornuéjols, and A. Ouorou. Mixed Integer NonLinear Programs featuring "On/Off" Constraints. *Computational Optimization and Applications*, 52:537–558, 2012.
32. H. Hijazi and L. Liberti. Constraint Qualification Failure in Second-Order Cone Formulations of Uunbounded Disjunctions. Technical report, NICTA, Canberra ACT Australia, 09 2014.
33. R. Jeroslow and J. Lowe. Modelling with integer varibales. In B. Korte and K. Ritter, editors, *Mathematical Programming at Oberwolfach II*, volume 22 of *Mathematical Programming Studies*, pages 167–184. Springer Berlin Heidelberg, 1984.
34. M. Kılınç. *Disjunctive Cutting Planes and Algorithms for Convex Mixed Integer Nonlinear Programming*. PhD thesis, University of Wisconsin-Madison, 2011.
35. M. Kılınç, J. Linderoth, and J. Luedtke. Effective separation of disjunctive cuts for convex mixed integer nonlinear programs. Technical Report TR1681, Computer Sciences Department, University of Wisconsin-Madison, 2010.
36. 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, D. Salvagnin, D.E. Steffy, and Wolter K. Miplib 2010. *Mathematical Programming Computation*, 3:103–163, 2011.
37. T. Koch, B. Hiller, M.E. Pfetsch, and L. Schewe, editors. *Evaluating Gas Network Capacities*. SIAM-MOS series on Optimization. SIAM, 2014.
38. S. Lawrence. Resource Constrained Project Scheduling: An Experimental Investigation of Heuristic Scheduling Techniques. GSIA, Carnegie Mellon University.
39. S. Lee and I. E. Grossmann. New algorithms for nonlinear generalized disjunctive programming. *Computers and Chemical Engineering*, 24:2125–2141, 2000.
40. A. Lodi. Indicator constraints in mixed-integer programming. Conference Talk MIP 2014, Columbus.
41. A. Lodi. Mixed integer programming computation. In M. Jünger, T.M. Liebling, D. Naddef, G.L. Nemhauser, W.R. Pulleyblank, G. Reinelt, G. Rinaldi, and L.A. Wolsey, editors, *50 Years of Integer Programming 1958-2008*, pages 619–645. Springer Berlin Heidelberg, 2010.
42. R. A. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86:515–532, 1999.
43. A. Tramontani. Lift-and-project cuts in CPLEX 12.5.1. Conference Talk INFORMS 2013, Minneapolis.