Algorithmic Operations Research

"Binarize and Project" to generate cuts for general mixed-integer programs

Jean-Sébastien Roy

Explore this journal

Cite this article

Article abstract

We consider mixed-integer linear programs with arbitrary bounded integer variables. We first describe a cutting plane approach based on the reformulation of integer variables into binary variables and describe a practical algorithm to compute these cuts for the original problem. We use the term "Binarize and Project" to highlight the similarity to the back onto the original space. Indeed, the interest of this approach lies in taking advantage of cutting plane approaches efficient for mixed-binary problems while keeping the problem in its non-binary form for improving the efficiency of the branch-and-bound procedure.

We then propose a new strengthened reformulation into binary variables that answers some concerns and limitations raised in [9], by ensuring that only one application of the lift-and-project convexification procedure to the binary reformulation of the problem is required to obtain a strengthening of the original problem.

Finally, the method is implemented inside the COIN optimization library and a preliminary experimentation is performed on problems from the MIPLIB library. The computational results confirm that the use of the proposed binary reformulation and cutting plane generation procedure leads to an improved integrality gap reduction albeit with an increased computing time.

"Binarize and Project" to generate cuts for general mixed-integer programs

Jean-Sébastien Roy [a]

[a]1, avenue du Général de Gaulle, 92141, Clamart, Cedex, France.

Abstract

We consider mixed-integer linear programs with arbitrary bounded integer variables. We first describe a cutting plane approach based on the reformulation of integer variables into binary variables and describe a practical algorithm to compute these cuts for the original problem. We use the term "Binarize and Project" to highlight the similarity to the lift-and-project idea of lifting the problem to a higher dimensional space to generate cutting planes which are then projected back onto the original space. Indeed, the interest of this approach lies in taking advantage of cutting plane approaches efficient for mixed-binary problems while keeping the problem in its non-binary form for improving the efficiency of the branch-and-bound procedure.

We then propose a new strengthened reformulation into binary variables that answers some concerns and limitations raised in [9], by ensuring that only one application of the lift-and-project convexification procedure to the binary reformulation of the problem is required to obtain a strengthening of the original problem.

Finally, the method is implemented inside the COIN optimization library and a preliminary experimentation is performed on problems from the MIPLIB library. The computational results confirm that the use of the proposed binary reformulation and cutting plane generation procedure leads to an improved integrality gap reduction albeit with an increased computing time.

Key words: Binary reformulation, Cutting planes generation, Lift-and-project.

1. Introduction

We consider mixed-integer linear programming problems of the form:

$$\min_{(x,y)\in (S\cap \mathbb{Z}^n)\times \mathbb{R}^p} dx + ey \qquad \text{s.t.} \quad Ax + By \geq c \quad (1)$$

where $d \in \mathbb{R}^n$, $e \in \mathbb{R}^p$, $A \in \mathbb{R}^{q\times n}$, $B \in \mathbb{R}^{q\times p}$, $c \in \mathbb{R}^q$, $q \in \mathbb{N}^*$, $n \in \mathbb{N}^*$, $p \in \mathbb{N}^*$ and without loss of generality, we suppose that the variables x are nonnegative and bounded, i.e., $S = [0, b_1] \times \ldots \times [0, b_n]$, with $b \in \mathbb{N}^{*n}$. We denote $P = \{(x, y) \in S \times \mathbb{R}^p | Ax + By \geq c\}$, the solution set of the continuous relaxation of the problem. Elements of P that satisfy the integrity requirements of problem (1) on x will be called mixed-integer solutions.

In order to reduce the size of the tree generated in a branch-and-cut algorithm, cutting planes are used to strengthen the continuous relaxation of these problems. One cutting plane approach, the so-called lift-and-project[1] sequential convexification procedure, consists in calculating the convex hull of the two polyhedra

given by a simple disjunction on any binary variable x_i:

$$\text{conv} \left(\{P \cap \{x_i = 0\}\} \cup \{P \cap \{x_i = 1\}\} \right)$$

This step is repeated on the newly calculated polyhedron as many times as desired. In theory, when all the variables x are binary, this method leads, within a finite number of steps, to the convex hull of the mixed-integer solutions to the problem (see [1]). In practice, only a few facets of this polyhedron are generated to strengthen the problem.

On the contrary, when the variables x are general (bounded) integers, the sequential convexification procedure converges to the convex hull of mixed-integer solutions only in the limit[8], and this is precisely why is it reasonable to expect improved computational efficiency from the use of binary reformulations.

Indeed, when the integer variables x are not restricted to binary values but are bounded, it is possible to reformulate the problem in various ways into an equivalent problem where all integer variables, denoted z, are 0-1 constrained, leading to problems of the form:

$$\min_{(z,y)\in \{0,1\}^m \times \mathbb{R}^p} dTz + ey \;\; s.t. \; ATz + By \geq c, \; Dz \geq 0$$

$$(2)$$

Email: Jean-Sébastien Roy [jean-sebastien.roy@edf.fr].

where $T \in \mathbb{R}^{n \times m}$ is a mapping such that $x = Tz$, and $D \in \mathbb{R}^{n \times q}$ is a constraint matrix involving the z variables only. Various possible choices for T and D will be discussed in section 2. below.

While it is possible, after strengthening the relaxation by applying cutting planes, to use branch-and-bound to solve this transformed mixed-binary problem (2), the discussion found in [9] strongly suggests that this process may be inefficient. It seems therefore interesting, after applying cutting plane approaches to the transformed problem (2), to translate back the resulting inequalities to the initial problem (1) to which branch-and-bound will be hopefully more efficiently applied.

The main goal of this paper is to propose, for general mixed-integer problems, a method to perform this translation in practice and thus take advantage of cutting plane algorithms available and efficient on mixed-binary problems, while still using the branch-and-bound method on the original problem. For this purpose, we present a suitable reformulation into binary variables as well as a procedure that generates cuts on the original problem from a strengthening of the binary reformulation of the problem. The proposed reformulation answers some concerns and limitations raised in [9], by ensuring that only one application of lift-and-project to the binary reformulation of the problem is required to obtain a strengthening of the original problem.

We observe here that our approach is similar in spirit to the main idea of lift-and-project, in which a mixed-binary problem is reformulated in a higher dimensional space where the integrality is more efficiently expressed in terms of a strengthened polyhedron (the term "Binarize and Project" suggests this connection). This polyhedron is then projected back onto the original space by finding inequalities, valid in the higher dimensional space, and at the same time naturally translated to the original space. In our scheme, the higher dimensional space corresponds to a specific binary reformulation of a mixed-integer problem on which any type of cut can be used to express integrality.

Remark that while our goal is to show that the binary reformulation can be used to reduce the integrality gap for some problems, this improvement comes with a price, namely an increase of the size of the problems on which cutting plane algorithms must be applied. In this paper, we do not address the computational efficiency in detail. Some suggestions are provided at the end of section 6., but this question requires extensive work and should be the subject of further research.

This paper is organized as follows. We present in section 2. the reformulations considered and the basic results justifying the use of a subset of the inequalities valid for the reformulated problem to generate inequalities valid for the original problem; we also characterize when strengthening the reformulated problem will lead to separating inequalities in the original one. A practical methodology is described in section 3.. We then present in section 4. the main result highlighting the better efficiency of our proposed binary reformulation as compared with other classical binary reformulations. Two small examples comparing various reformulations into binary variables are presented in section 5.. Computational results obtained from a preliminary implementation of the cut generation method through binary reformulation, applied to MIPLIB problems, is provided in section 6.. Finally, the concluding section provides some comments concerning the interest of the proposed approach.

2. Basis of the proposed method

We are looking for the convex hull of the solutions $(x, y) \in (S \cap \mathbb{Z}^n) \times \mathbb{R}^p$ of the mixed-integer problem (1) with general bounded integer variables x. We describe in this section the proposed reformulation, discuss the extent to which it yields the expected convex hull, and characterize when we will be able to generate cutting planes from a binary reformulation of a problem.

Let $I = (\mathbb{Z}^n \times \mathbb{R}^p) \cap P$ denote the solution set of the mixed-integer problem. We want to describe or at least approximate $P' = \mathrm{conv}\,(I)$. As the convex hull of a finite number of polyhedra, P' is a polyhedron, so $\exists\,(A', B', c')$ such that $P' = \{\,(x, y) \in S \times \mathbb{R}^p \,|\, A'x + B'y \geq c'\,\}$.

We now reformulate the integer variables x of the polyhedron P to generate an equivalent polyhedron P_{01} for which the integer variables are binary. The following definition encompasses the class of reformulations we are interested in. A few examples including our suggestion will follow.

Definition 1. (Reformulation into binary variables). We consider the reformulation of integer variables $x \in S \cap \mathbb{Z}^n$ into binary variables $z \in \{0, 1\}^m$, $m \in \mathbb{N}$, given by the formula $x = Tz$, with the constraints $Dz \geq 0$, where $T \in \mathbb{Z}^{n \times m}$ and $D \in \mathbb{R}^{n \times q}$. The reformulations must satisfy the following rules:

(1) $\forall x \in S$, $\exists z \in [0, 1]^m$ such that $x = Tz$ and $Dz \geq 0$. z is said to be associated with x for the chosen reformulation.

(2) $\forall x \in S \cap \mathbb{Z}^n$, $\exists z \in \{0,1\}^m$ such that $x = Tz$ and $Dz \geq 0$. z is said to be canonically associated with x for the chosen reformulation.

(3) $\forall z \in \{0,1\}^m$ such that $Dz \geq 0$, $Tz \in S \cap \mathbb{Z}^n$.

Examples of such reformulations [1] include the compact reformulation proposed by Owen-Mehrotra in [9]:

Definition 2. (Owen-Mehrotra, compact reformulation) Let $l_i = \lfloor \log_2 b_i \rfloor$. We reformulate the variables $x_i \in \{0, \dots, b_i\}$ using $x_i = \sum_{j=0}^{l_i} 2^j \cdot z_i^j$ with $\forall j \in \{0, \dots, l_i\}$, $z_i^j \in \{0,1\}$ and the constraints $\sum_{j=0}^{l_i} 2^j \cdot z_i^j \leq b_i$ if $\sum_{j=0}^{l_i} 2^j > b_i$.

And the reformulation proposed by Sherali-Adams in [10]:

Definition 3. (Sherali-Adams, non-compact reformulation) We reformulate variables $x_i \in \{0, \dots, b_i\}$ using $x_i = \sum_{j=1}^{b_i} j \cdot z_i^j$ with $\forall j \in \{1, \dots, b_i\}$, $z_i^j \in \{0,1\}$ and the constraints $\sum_{j=1}^{b_i} z_i^j \leq 1$.

In addition to the above, we suggest an alternative reformulation which will be extensively discussed in the sequel:

Definition 4. (The proposed, strong non-compact reformulation into binary variables) The suggested reformulation defines for each component $x_i \in \{0, \dots, b_i\}$ of x, $i \in \{1, \dots, n\}$, a set of binary variables $z_i^j \in \{0,1\}$ where $j \in \{1, \dots, b_i\}$ such that $x_i = \sum_{j=1}^{b_i} z_i^j$. Moreover, this reformulation is strengthened by including a set of constraints on z: $\forall i \in \{1, \dots, n\}$, $\forall j \in \{1, \dots, b_i - 1\}$, $z_i^j \geq z_i^{j+1}$. Adding such ordering constraints in this reformulation, while not strictly necessary, significantly reduces the feasible space for z by breaking the symmetry and eliminating redundant binary representations of some integers, and will improve the efficiency of the cutting plane algorithms applied to the reformulated problem (see section 4.).

We denote

$$z = \left[z_1^1, \dots, z_1^j, \dots, z_1^{b_1}, z_2^1, \dots, z_i^j, \dots, z_n^{b_n} \right] \in [0,1]^m$$

with $m = \sum_{i=1}^{n} b_i$. With each $x \in S$, we associate $z = \varphi(x) \in [0,1]^m$ such that $z_j^i = 1$ if $j \leq \lfloor x_i \rfloor$, $z_{\lfloor x_i \rfloor + 1}^i = x_i - \lfloor x_i \rfloor$ if $\lfloor x_i \rfloor + 1 \leq b_i$ and $z_j^i = 0$ if $j > \lfloor x_i \rfloor + 1$, i.e., $z = [1, \dots, 1, x_i - \lfloor x_i \rfloor, 0, \dots, 0, 1, \dots]$. Hence $Tz = x$. Moreover, we note that $x \in S \cap \mathbb{Z}^n$ if and only if $z \in \{0,1\}^m$. We remark that for this reformulation, and for all $x \in S \cap \mathbb{Z}^n$, z canonically

---

[1] For all the following reformulations, the structure of matrices T and D is straightforward.

associated with x (see definition 1) is unique and equal to $\varphi(x)$.

Remark 1 *We will call "weak non-compact reformulation", the previous reformulation with the ordering constraints removed, i.e., $D = 0$.*

In the rest of the present section, we consider any reformulation complying with the general definition 1.

Definition 5. (Reformulation of a polyhedron) For $P = \{(x,y) \in S \times \mathbb{R}^p \mid Ax + By \geq c\}$, we denote by $P_{01} = \{(z,y) \in [0,1]^m \times \mathbb{R}^p \mid ATz + By \geq c, Dz \geq 0\}$ the polyhedron of the constraints of the problem in the transformed variables space; $I_{01} = (\mathbb{Z}^m \times \mathbb{R}^p) \cap P_{01}$, the mixed-integer solution set of the transformed problem and $P'_{01} = \text{conv}(I_{01})$ the convex hull of the mixed-integer solutions.

Example 1 *Let $P = \{(x_1, x_2, y) \in \mathbb{R}^3_+ \mid y \leq 3x_1,\ y \leq 3x_2,\ 3x_1 + 3x_2 + 2y \leq 6\}$. Note that the variables x_1 and x_2 are implicitly upper bounded by 2. Then, for the proposed strong non-compact reformulation, P_{01} is defined, for $\left(z_1^1, z_1^2, z_2^1, z_2^2, y\right) \in [0,1]^4 \times \mathbb{R}_+$, by the following equations :*

$$\begin{cases} y \leq 3\left(z_1^1 + z_1^2\right) \\ y \leq 3\left(z_2^1 + z_2^2\right) \\ 3\left(z_1^1 + z_1^2\right) + 3\left(z_2^1 + z_2^2\right) + 2y \leq 6 \\ z_1^1 \geq z_1^2 \\ z_2^1 \geq z_2^2 \end{cases}$$

Let $V = \{(\alpha, \beta, \gamma \in \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R} \mid \alpha Tz + \beta y \geq \gamma$ is an inequality valid for $P'_{01}\}$, the set of the coefficients of inequalities valid for P'_{01} such that the coefficients of the variables z_i^j take the form αT. These inequalities can be translated into inequalities of the form $\alpha x + \beta y \geq \gamma$ defined on the initial space.

Remark 2 *If the strong non-compact reformulation is used, for each i, the coefficients of the variables z_i^j for the inequalities in V, are equal for all $j = 1, \dots, b_i$.*

Let $C = \bigcap_{(\alpha,\beta,\gamma) \in V} \{(x,y) \in S \times \mathbb{R}^p \mid \alpha x + \beta y \geq \gamma\}$, the intersection of all inequalities translated in this way. We now formulate the simple result justifying the use of cutting planes belonging to V to generate cutting planes in the original space.

Proposition 3 *$C = P'$, i.e., the inequalities built from the transformed problem define the convex hull of the mixed-integer solutions to the initial problem.*

Proof: Let us begin by proving that $C \subseteq P'$. Let $\alpha x + \beta y \geq \gamma$ be an inequality valid for P'. We will prove that it is valid for C by showing that $(\alpha, \beta, \gamma) \in V$. By contradiction, suppose $(\alpha, \beta, \gamma) \notin V$. So, $\exists (z', y') \in$

I_{01}, $\alpha T z' + \beta y' < \gamma$. Let $x' = Tz' \in S \cap \mathbb{Z}^n$, hence $Ax' + By' = ATz' + By' \geq c$. Therefore $(x', y') \in I \subseteq P'$ and $\alpha x' + \beta y' < \gamma$ which is inconsistent. Therefore $(\alpha, \beta, \gamma) \in V$ and $\alpha x + \beta y \geq \gamma$ is valid for C.

Let us now prove that $C \supseteq P'$. $\forall (x', y') \in I$, let z' be the 0-1 solution canonically associated with x' for the reformulation under consideration, i.e., $z' \in \{0,1\}^m$ such that $Tz' = x'$ and $Dz' \geq 0$. Hence $Ax' + By' = ATz' + By' \geq c$, and $(z', y') \in I_{01}$. Hence $\forall (\alpha, \beta, \gamma) \in V$, $\alpha x' + \beta y' = \alpha T z' + \beta y' \geq \gamma$, and $(x', y') \in C$. Finally, $I \subseteq C$ and $P' = \mathrm{conv}\,(I) \subseteq \mathrm{conv}\,(C) = C$.

Therefore, we have proved that the set V of inequalities valid for the polyhedron P'_{01}, of the form $\alpha T z + \beta y \geq \gamma$, which can be easily reformulated into inequalities $\alpha x + \beta y \geq \gamma$ valid for the polyhedron P', are sufficient to define P'. We remark that thanks to the lift-and-project sequential convexification procedure, and contrary to the disjunctions discussed in [8], a description of the set V of these inequalities can be obtained within a finite number of steps.

We will use the projection result of proposition 3 to design a cutting plane algorithm. When trying to separate a point (x', y') from P', we will reformulate P into P_{01}, strengthen P_{01} using cutting planes to obtain a polyhedron $\hat{P}_{01}$, and try to find an inequality of the form $\alpha T z + \beta y \geq \gamma$ that separates from $\hat{P}_{01}$ some (z', y') such that $x' = Tz'$ and $Dz' \geq 0$. The following result helps characterize when $\alpha z + \beta y \geq \gamma$ could be an inequality separating (x', y') from P'. Finding such an inequality when it exists will be the subject of the next section, while ensuring that the conditions required by the following proposition are met, by means of a specific binary reformulation, will be the subject of section 4..

Proposition 4 *Let $(x', y') \in S \times \mathbb{R}^p$ be such that $(x', y') \notin P'$, and $\hat{P}_{01}$ the polyhedron P_{01} strengthened using cutting planes (hence $P'_{01} \subseteq \hat{P}_{01}$). Let $E = \{(z, y) \mid x' = Tz,\ Dz \geq 0,\ y' = y\}$. If $E \cap \hat{P}_{01} = \emptyset$, then there exists an inequality of the form $\alpha T z + \beta y \geq \gamma$ that separates E from $\hat{P}_{01}$ and the inequality $\alpha x + \beta y \geq \gamma$ separates (x', y') from P'.*

Proof: Let $P^* = \{(Tz, y)\}_{(z,y) \in \hat{P}_{01}}$ denote the projection of $\hat{P}_{01}$ onto the original space, and P^*_{01} its binary reformulation (see definition 5). Assuming $E \cap \hat{P}_{01} = \emptyset$, $(x', y') \notin P^*$, therefore let $\alpha x + \beta y \geq \gamma$ be an inequality separating (x', y') from P^*. Since $P'_{01} \subseteq \hat{P}_{01}$, $P' \subseteq P^*$ and we deduce that $\alpha x + \beta y \geq \gamma$ also separates (x', y') from P'. Hence, $\forall (z, y) \in E, \alpha T z + \beta y = ax' + \beta y' < \gamma$. Moreover, $\forall (z, y) \in P^*_{01}, (Tz, y) \in P^*$

hence $\alpha T z + \beta y \geq \gamma$. Therefore, $\alpha T z + \beta y \geq \gamma$ separates E from P^*_{01}, and since $\hat{P}_{01} \subseteq P^*_{01}$, $\alpha T z + \beta y \geq \gamma$ separates E from $\hat{P}_{01}$.

3. Cut generation through binary reformulation

In this section, we will discuss precisely how to use a binary reformulation to generate cutting planes. Throughout this section, the discussion is intended to remain general, i.e., to apply to any binary reformulation complying with definition 1. Let $(x', y') \in P$ be such that $(x', y') \notin P'$. We assume $P' \neq \emptyset$ and try to build an inequality separating (x', y') from P'.

Let $z' \in [0,1]^m$ associated with x' (see definition 1). Then $x' = Tz'$, and $z' \notin \{0,1\}^m$ because $(x', y') \in P$ and $(x', y') \notin P'$ implies $x' \notin \mathbb{Z}^n$, and since $T \in \mathbb{Z}^{n \times m}$, $z' \in \{0,1\}^m$ would imply $x' \in \mathbb{Z}^n$. Hence $z' \in P_{01}$ and since $z' \notin \{0,1\}^m$, we have $z' \notin P'_{01}$. It is therefore possible to build valid inequalities separating z' from P'_{01}.

After having generated a number of valid inequalities separating (z', y') from P_{01} we add them to P_{01} to obtain a new polyhedron $\hat{P}_{01}$ such that $P_{01} \supsetneq \hat{P}_{01} \supseteq P'_{01}$, and such that $(z', y') \notin \hat{P}_{01}$. Let $\hat{A} \in \mathbb{R}^{k \times m}$, $\hat{B} \in \mathbb{R}^{k \times p}$ and $\hat{c} \in \mathbb{R}^k$ be such that:

$$\hat{P}_{01} = \left\{ (z, y) \in \mathbb{R}^m \times \mathbb{R}^p \,\middle|\, \hat{A}z + \hat{B}y \geq \hat{c} \right\}$$

In order to obtain an inequality separating (x', y') from P', we can, for example, solve the following problem:

$$\min_{(u,v) \in \mathbb{R}^k_+ \times \mathbb{R}^n} u\left(\hat{A}z' + \hat{B}y' - \hat{c}\right) \qquad (3)$$

$$\text{s.t.} \quad u\hat{A} = vT, \ \sum_k u_k \leq 1$$

(Throughout the paper, u and v are row vectors.)

We remark that this problem is well defined since u is bounded, so the objective function is bounded too, and $u = v = 0$ is a solution. This problem consist in finding an inequality valid for $\hat{P}_{01}$ as a nonnegative combination u of the inequalities of the system $\hat{A}z + \hat{B}y \geq \hat{c}$ defining $\hat{P}_{01}$, by maximizing the distance between (z', y') and the half space defined by the valid inequality, in the sense of the distance $-u\left(\hat{A}z' + \hat{B}y' - \hat{c}\right)$, while ensuring, through the constraint $u\hat{A} = vT$, that the inequality can be reformulated as an inequality valid for P', that is, if the strong non-compact reformulation is used, whose coefficients for variables z_i^j are equal for all j. These coefficients define the coefficients of variables x_i for the inequality sought after.

Remark 5 *Multiple formulations of problem (3.) can be obtained by either changing the normalization constraint on u or removing the upper bound on u and using as the solution a primal ray from the now unbounded problem. For simplicity of exposition, we will only consider the L^1 normalization constraint.*

Proposition 6 *Let (u, v) be an optimal solution to problem (3.). Then $vx + u\hat{B}y \geq u\hat{c}$ is an inequality valid for P'. Moreover, if $u\left(\hat{A}z' + \hat{B}y' - \hat{c}\right) < 0$, $vx + u\hat{B}y \geq u\hat{c}$ separates (x', y') from P'.*

Proof: Let $(x'', y'') \in I$, and z'' canonically associated with x''. Therefore $x'' = Tz''$ and $(z'', y'') \in I_{01}$. Since $u\hat{A}z + u\hat{B}y = vTz + u\hat{B}y \geq u\hat{c}$ is a nonnegative combination of the inequalities defining $\hat{P}_{01}$, it is a valid inequality for $\hat{P}_{01}$ and is therefore also valid for P'_{01}. Hence $vx'' + u\hat{B}y'' = vTz'' + u\hat{B}y'' \geq u\hat{c}$ as required.

If $u\left(\hat{A}z' + \hat{B}y' - \hat{c}\right) < 0$, $vx' + u\hat{B}y' = vTz' + u\hat{B}y' = u\hat{A}z' + u\hat{B}y' < u\hat{c}$ and the inequality $vx + u\hat{B}y \geq u\hat{c}$ separates (x', y') from P'.

Algorithm 1 below summarizes the procedure.

Step 1: Let $(x', y') \in P$ assumed to be such that $(x', y') \notin P'$.

Step 2: Compute $z' \in [0, 1]^m$ associated with x' ($z' \notin \{0, 1\}^m$) (see definition 1).

Step 3: Compute one or more cuts separating z' from P'_{01}. By combining them with P_{01}, we obtain $\hat{P}_{01} = \left\{(z, y) \in \mathbb{R}^m \times \mathbb{R}^p \,\middle|\, \hat{A}z + \hat{B}y \geq \hat{c}\right\}$.

Step 4: Solve the cut projection problem (3) and let (u, v) be an optimal solution to it:

$$\min_{(u,v) \in \mathbb{R}^k_+ \times \mathbb{R}^n} u\left(\hat{A}z' + \hat{B}y' - \hat{c}\right)$$

$$\text{s.t.} \quad u\hat{A} = vT, \sum_k u_k \leq 1 \qquad (3)$$

Step 5: If $u\left(\hat{A}z' + \hat{B}y' - \hat{c}\right) < 0$ then, $vx + u\hat{B}y \geq u\hat{c}$ is valid for P' and separates (x', y') from P'.

Algorithm 1. *Generation of an inequality separating (x', y') from $P' = \text{conv}\left((\mathbb{Z}^n \times \mathbb{R}^p) \cap P\right)$*

Remark 7 *In algorithm 1, $u\left(\hat{A}z' + \hat{B}y' - \hat{c}\right)$ is not always strictly negative. For example, as detailed in [9], if either the compact reformulation of Owen-Mehrotra or the non-compact reformulation of Sherali-Adams is used on some polyhedra, performing disjunctions on some binary variables may lead to a strictly smaller polyhedron which nevertheless, once projected back to the original space, is identical to the original polyhedron. Section 5. provides two such examples. In these cases, (x', y') belongs to the projected polyhedron, and therefore $u\left(\hat{A}z' + \hat{B}y' - \hat{c}\right) = 0$.*

Nevertheless, we will show in section 4. that for the strong non-compact reformulation, and under appropriate assumptions, the procedure always generates a separating hyperplane.

The following result characterizes the cases where the inequality generated will be separating.

Proposition 8 *Let $E = \{(z, y) \,|\, x' = Tz, Dz \geq 0, y' = y\}$. The generated inequality $vx + u\hat{B}y \geq u\hat{c}$ separates (x', y') from P' if and only if $E \cap \hat{P}_{01} = \emptyset$.*

Proof: Suppose that the inequality $vx + u\hat{B}y \geq u\hat{c}$ separates (x', y') from P', then $\forall z$ such that $x' = Tz$, $vx' + u\hat{B}y' = vTz + u\hat{B}y' = u\hat{A}z + u\hat{B}y' < u\hat{c}$. Moreover, $u\left(\hat{A}z + \hat{B}y - \hat{c}\right) \geq 0$, being a nonnegative combination of the inequalities defining $\hat{P}_{01}$, is an inequality valid for $\hat{P}_{01}$. Therefore, $u\hat{A}z + u\hat{B}y' < u\hat{c}$ implies $(z, y') \notin \hat{P}_{01}$.

Conversely, from proposition 4, there exists an inequality of the form $\alpha Tz + \beta y \geq \gamma$ that separates E from $\hat{P}_{01}$. Therefore it exists $u^* \in \mathbb{R}^k_+$, $u^* \neq 0$, such that $(\alpha T, \beta, \gamma) = u^*\left(\hat{A}, \hat{B}, \hat{c}\right)$. Let $\bar{u} = \sum_k u^*_k$, $u' = \frac{u^*}{\bar{u}}$ and $v' = \frac{\alpha}{\bar{u}}$: $(u', v') \in \mathbb{R}^k_+ \times \mathbb{R}^n$, $\sum_k u'_k \leq 1$, $u'\hat{A} = \frac{u^*\hat{A}}{\bar{u}} = \frac{\alpha T}{\bar{u}} = vT$ and $u'\left(\hat{A}z' + \hat{B}y' - \hat{c}\right) = \frac{\alpha Tz' + \beta y' - \gamma}{\bar{u}} < 0$. We conclude using proposition 6.

In optimization problems such as (1), the cutting plane generation algorithm will generally be decomposed in two phases. First the binary expansion P_{01} is generated and strengthened using cutting plane approaches. Then the cutting plane projection problem (3) is repeatedly solved to generate cutting planes to strengthen the relaxation in the original space until the optimum of this relaxation cannot be cut off. Let again $\hat{P}_{01}$ denote the polyhedron obtained after applying valid inequalities to

P_{01} and

$$\min_{(x,y)\in S\times\mathbb{R}^p} dx + ey \tag{4}$$

$$\text{s.t. } Ax + By \geq c$$

the continuous relaxation of problem (1). Ideally, we would like to project $\hat{P}_{01}$ onto the original space but this would be too computationally expensive. As an alternative, we project part of the inequalities defining it and use them to strengthen the polyhedron P. We project as many inequalities as needed to ensure that the optimum over the strengthened P belongs to the projection of $\hat{P}_{01}$. Therefore, upon termination, both the strengthened original and strengthened reformulated problems will have the same optimal value.

More precisely, we try to separate the optimum of the continuous relaxation (4), using a cutting plane generated by solving the cut projection problem (3). If a separating inequality is generated, it is added to the constraints of the continuous relaxation. The procedure is repeated until no separating inequality is generated. Algorithm 2 describes the proposed cutting plane generation procedure in a mixed-integer linear programming optimization context.

If the inequalities generated by solving problem (3) are facets of the projection of $\hat{P}_{01}$ by T (which is ensured by a proper normalization constraint, see for instance [1,4]), then algorithm 2 terminates after a finite number of steps.

4. Using the strong non-compact reformulation in binary variables

In this section we show how the choice of a specific reformulation can provide a decisive advantage to the practical application of the method as compared with other binary reformulations. We will show that contrary to the general case, when using the strong non-compact reformulation (see definition 4), algorithm 1 always generates a separating cutting plane, which makes this algorithm potentially interesting in practice. As shown in the next section on a few examples, this property is not shared by the other suggested binary reformulations.

Let (x', y') be a vertex of P such that $(x', y') \notin P'$. Similarly to the previous section, we try to build an inequality valid for P' separating (x', y') from P'.

Proposition 9 *Assume that the binary reformulation used is the strong non-compact reformulation (definition 4), and the point (x', y') to separate is a vertex of*

Step 1: Strengthen P_{01} using cutting planes to obtain a polyhedron $\hat{P}_{01}$. Usually, P_{01} is strengthened by repeatedly adding cutting planes separating the optimum of the objective function $dTz + ey$ over it from P'_{01}.

Step 2: Let $(x', y') \in P$ be an optimum solution to the continuous relaxation (4).

Step 3: Compute $z' \in [0,1]^m$ associated with x' ($z' \notin \{0,1\}^m$) (see definition 1).

Step 4: Solve the cut projection problem (3) and let (u, v) be an optimal solution to it:

$$\min_{(u,v)\in\mathbb{R}^k_+\times\mathbb{R}^n} u\left(\hat{A}z' + \hat{B}y' - \hat{c}\right)$$

$$\text{s.t. } u\hat{A} = vT, \sum_k u_k \leq 1 \tag{3}$$

Step 5: If $u\left(\hat{A}z' + \hat{B}y' - \hat{c}\right) = 0$, the algorithm terminates.

Step 6: Strengthen P using the inequality $vx + u\hat{B}y \geq u\hat{c}$ valid for P', and go to step 3.. In practice, only the cuts that are tight at the optimum of the strengthened P are kept, which does not deteriorate the value of strengthened relaxation.

Algorithm 2. *Cut generation in a MILP optimization context*

P. Then if the cutting plane method used is lift-and-project applied to a fractional component of $z' = \varphi(x')$ (definition 4), the cutting plane generated by algorithm 1 separates (x', y') from P'.

Proof: Let $z' \in [0,1]^m$ such that $z' = \varphi(x')$. Since $x' \notin \mathbb{Z}^n$, $\exists i'$ such that $x'_{i'} \notin \mathbb{Z}$. Let $j' = \lfloor x_{i'} \rfloor + 1$, therefore $z'^{j'}_{i'} \notin \{0,1\}$. Then, we apply the lift-and-project convexification procedure to variable $z^{j'}_{i'}$ of polyhedron P_{01}, and obtain a new polyhedron denoted $\hat{P}_{01}$ as before. i.e. $\hat{P}_{01} =$

$$\text{conv}\left(\left\{P_{01} \cap \left\{z^{j'}_{i'} = 0\right\}\right\} \cup \left\{P_{01} \cap \left\{z^{j'}_{i'} = 1\right\}\right\}\right)$$

$$= \left\{(z, y) \in \mathbb{R}^m \times \mathbb{R}^p \,\middle|\, \hat{A}z + \hat{B}y \geq \hat{c}\right\}$$

Let $E = \{(z, y) \mid x' = Tz, Dz \geq 0, y' = y\}$ and $P^* = \{(Tz, y)\}_{(z,y)\in\hat{P}_{01}}$, the projection of $\hat{P}_{01}$ through T. We will prove that $(x', y') \notin P^*$, which implies that $E \cap \hat{P}_{01} = \emptyset$.

Let $(x, y) \in P^*$. $\exists (z, y) \in \hat{P}_{01}$ such that $x = Tz$. Since $\hat{P}_{01} \subsetneq P_{01}$, $(z, y) \in P_{01}$ therefore $ATz + By = Ax + By \geq c$ and $(x, y) \in P$. Thus $P^* \subseteq P$. By contradiction, suppose $(x', y') \in P^*$. Since (x', y') is a vertex of P and $(x', y') \in P^*$, (x', y') is a vertex of P^*.

From the definition of $\hat{P}_{01}$, we deduce $\forall (z, y) \in \hat{P}_{01}$, $\exists (z_0, y_0) \in \left\{ P_{01} \cap \left\{ z_{i'}^{j'} = 0 \right\} \right\}$ and $(z_1, y_1) \in \left\{ P_{01} \cap \left\{ z_{i'}^{j'} = 1 \right\} \right\}$ and $\lambda \in [0, 1]$, such that $(z, y) = \lambda (z_0, y_0) + (1 - \lambda) (z_1, y_1)$. In particular, $\forall (z, y) \in \hat{P}_{01}$ such that $(Tz, y) = (x', y')$, (therefore $y = y'$), let $x_0 = Tz_0$ and $x_1 = Tz_1$. We therefore have $(x_0, y_0) \in P^*$ and $(x_1, y_1) \in P^*$. Hence, $\lambda (x_0, y_0) + (1 - \lambda) (x_1, y_1) = (Tz, y') = (x', y') \in P^*$. Since (x', y') is a vertex of P^*, $\lambda \in \{0, 1\}$, and $x' = x_0$ or $x' = x_1$.

Suppose $x' = x_0$. Since $(z_0, y_0) \in \{P_{01} \cap \{z_{i'}^{j'} = 0\}\}$, using the ordering constraints on variables z_i^j in the strong non-compact reformulation, which are therefore verified in $\hat{P}_{01} \subsetneq P_{01}$, in one hand $\forall j > j'$, $z_{0,i'}^j = 0 \leq z_{i'}^{\prime j}$, and in another hand, since $z_{i'}^{\prime j' - 1} = 1$, $\forall j < j'$, $z_{i'}^{\prime j} = 1 \geq z_{0,i'}^j$. Finally, $z_{i'}^{\prime j'} > 0 = z_{0,i'}^{j'}$. Therefore, $x_{0,i'} = (Tz_0)_{i'} = \sum_{j=1}^{b_{i'}} z_{0,i'}^j \leq j' - 1 < \sum_{j=1}^{b_{i'}} z_{i'}^{\prime j} = (Tz')_{i'} = x'_{i'}$, so $x' \neq x_0$, which is inconsistent. Similarly if we suppose $x' = x_1$, we show show that $x_{1,i'} \geq j' > x'_{i'}$ which is also inconsistent. Therefore $(x', y') \notin P^*$, $E \cap \hat{P}_{01} = \emptyset$ and we conclude using proposition 8.

It thus follows from the above proposition 9 that when our proposed strong non-compact reformulation is used, and when we are looking to separate a vertex of P not belonging to P', only one application of the lift-and-project convexification procedure to the binary reformulation of the polyhedron (see definition 5) is required to obtain a valid inequality separating the chosen vertex of P from P'. As illustrated in the next section through examples, this property is a key advantage of the strong non-compact reformulation, and is not shared by the other suggested reformulations (definitions 2 and 3). To a large extent, this property alleviates one of the main criticisms of the idea of using binary reformulations stated in [9], namely that the lift-and-project convexification procedure must be applied multiple times in sequence before obtaining a valid inequality separating the chosen vertex of P from P'.

Remark 10 *We remark that, when using the strong non-compact reformulation, applying the lift-and-project convexification procedure once to variable z_i^j,*

i.e., considering the disjunction: $z_i^j = 0$ or $z_i^j = 1$, corresponds to the disjunction $x_i \leq j - 1$ or $x_i \geq j$ on variable x_i described by Owen and Mehrotra in [8]. The sequential application of such disjunctions on x cannot systematically permit to obtain the convex hull P' of integer points of P in a finite number of steps: the disjunctions must be performed simultaneously to achieve this goal. On the contrary, the sequential application of the lift-and-project procedure to all variables of P_{01} leads, after projection, to valid inequalities defining P' (see proposition 3). Naturally, a more realistic sequential application of the lift-and-project procedure to a few binary variables before projecting, produces an intermediate result.

As an illustration, let L be a sequence of pairs $(i_k, j_k)_{k \in \{1, \dots, K\}}$ with $i_k \in \{1, \dots, n\}$ and $j_k \in \{1, \dots, b_i\}$, describing disjunctions of the form $x_{i_k} \leq j_k - 1$ or $x_{i_k} \geq j_k$. Let

$$P^* = \mathrm{conv}(\bigcap_{(i,j) \in L} (\{(x,y) \in P | x_i \leq j - 1\} \cup \{(x,y) \in P | x_i \geq j\}))$$

denote the polyhedron obtained by performing all these disjunctions simultaneously on P. In one hand, P^* can be described with approximately 2^K times more constraints and variables than P, which is often intractable. On the other hand, while P^* cannot be obtained by applying the disjunctions sequentially, it can be obtained using the strong non-compact reformulation, sequential lift-and-project and projection. More precisely, let P_{01} be the strong non-compact reformulation of P. Let us sequentially apply lift-and-project to variables z_i^j, for $(i, j) \in L$. Let $Q_0 = P_{01}$ and $\forall k \in \{1, \dots, K\}$, $Q_k =$

$$\mathrm{conv}(\{(z, y) \in Q_{k-1} | z_i^j = 0\}$$
$$\cup \{(z, y) \in Q_{k-1} | z_i^j = 1\})$$

In accordance with the beginning of this remark, the projection of Q_K onto the original space is P^*. In the context of cutting plane generation (using problem (3.) to perform the projection, and only generating a few facets of the polyhedra involved), this formulation might be more tractable.

5. Comparison of the strong non-compact reformulation against other binary reformulations

In this section, we compare the strong non-compact reformulation to other classical reformulations introduced in section 2.. On two examples, we show how

cutting plane generation through reformulation progressively yields the convex hull of mixed-integer solutions, notably in a case where mixed-integer Gomory cuts fail to do so.

5.1. *First example*

We consider here the following example due to Cook, Kannan and Schrijver[6]:

Let P be the polyhedron in variables $(x_1, x_2, y) \in \mathbb{R}^3_+$ defined by the constraints:

$$P = \{(x_1, x_2, y) \in \mathbb{R}^3_+ | y \leq 3x_1, \ y \leq 3x_2,$$
$$3x_1 + 3x_2 + 2y \leq 6\}$$

Note that the variables x_1 and x_2 are implicitly upper bounded by 2.

We are looking for the convex hull P' of points I such that x_1 and x_2 are integers, i.e., $I = (\mathbb{Z}^2 \times \mathbb{R}) \cap P$ and $P' = \text{conv} \, I$.

The vertices of polyhedron P (see figure 1) are the points:

$$(x_1, x_2, y) \in \{(0,0,0), \, (2,0,0), \, (0,2,0), \, (\tfrac{1}{2}, \tfrac{1}{2}, \tfrac{3}{2})\}$$
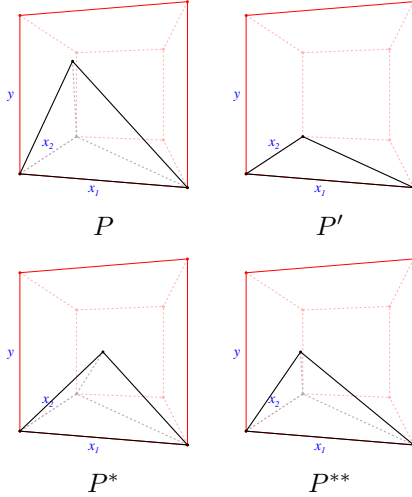

Fig. 1. Polyhedra P, P', P^* and P^{**}, in cubes 2 units wide

The vertices of polyhedron P' (see figure 1) are the points:
$$(x_1, x_2, y) \in \{(0,0,0), \, (2,0,0), \, (0,2,0)\}$$

5.1.1. *Methodology*

To obtain P', a first approach would be to apply cutting plane methods such as mixed-integer Gomory cuts. It can be shown that it is not possible to obtain the convex hull P' through the application of a finite number of mixed-integer Gomory cuts (see [6]).

Alternatively, it is possible to reformulate integer variables x_1 and x_2 into binary variables in order to obtain a polyhedron P_{01} such that the convex hull of its integer points can be obtained by applying a finite number of lift-and-project cuts. We then obtain the polyhedron P' through a projection.

We will apply this method with various binary reformulations, including the strong non-compact one proposed here. For each of these reformulations, we will apply the lift-and-project convexification procedure (see section 1.) to some or all of the variables, and then project the polyhedron onto the original space. We will describe the polyhedra by listing their vertices. In the case of the application of the lift-and-project convexification over all the variables, the result will systematically be P' (see section 2.). We are therefore interested in applying it on a limited subset of the variables. We remark that the result of the sequential convexification does not depend on the order of application on the variables (see for example proposition 4 in [9]).

Let

$$P^* = \text{conv}(\{(0,0,0), \, (2,0,0), \, (0,2,0), \, (1, \tfrac{1}{3}, 1)\})$$

and

$$P^{**} = \text{conv}(\{(0,0,0), \, (2,0,0), \, (0,2,0), \, (\tfrac{1}{3}, 1, 1)\})$$

(see figure 1).

5.1.2. *Compact reformulation (Owen-Mehrotra)*

In the case of the compact reformulation (see definition 2), a disjunction on a variable z_i^j: $z_i^j = 0$ or $z_i^j = 1$, corresponds to a disjunction on a variable x_i: $x_i \leq 2^{j-1}$ or $x_i \geq 2^j$ if $j = l_i$ and $x_i \leq b_i$, or $x_i \geq 2^j$ if not, which is obviously not very interesting if $j \neq l_i$. We observe here the behavior described by Owen and Mehrotra in [9]. In the following, we list the variables on which the disjunction-convexification procedure is applied, followed by an arrow ($\rightarrow$) and the polyhedron obtained after projection.

For example, $\left[z_1^0, z_1^1\right] \rightarrow P^*$ means that the polyhedron P was reformulated using the compact reformulation into a polyhedron P_{01}. Disjunctions on the variables z_1^0 and z_1^1 are applied on P_{01} and lead to

a new polyhedron $\hat{P}_{01}$, i.e., let $Q = \mathrm{conv}(\{(z, y) \in P_{01} | z_1^0 = 0\} \cup \{(z, y) \in P_{01} | z_1^0 = 1\})$, then $\hat{P}_{01} = \mathrm{conv}\left(\left\{(z, y) \in Q \,|\, z_1^1 = 0\right\} \cup \left\{(z, y) \in Q \,|\, z_1^1 = 1\right\}\right)$. As observed earlier, the resulting polyhedron does not depend on the order according to which the disjunctions are considered. Similarly, applying the disjunctions sequentially or simultaneously lead to the same result. The resulting polyhedron $\hat{P}_{01}$ is then projected back to the original (x_1, x_2) space, and in this case, this projection leads to P^* as defined above.

Examining all possible simple disjunctions on variables z_i^j, we obtain:

$$\left[z_1^0\right] \to P, \left[z_1^1\right] \to P ;$$

$$\left[z_1^0, z_1^1\right] \;\to\; P^*, \; \left[z_1^0, z_2^0\right] \;\to\; P, \; \left[z_1^0, z_2^1\right] \;\to\; P, \left[z_1^1, z_2^0\right] \to P, \left[z_1^1, z_2^1\right] \to P ;$$

$$\left[z_1^0, z_1^1, z_2^0\right] \to P^*, \left[z_1^0, z_1^1, z_2^1\right] \to P^*, \left[z_1^0, z_2^0, z_2^1\right] \to P^{**}, \left[z_1^1, z_2^0, z_2^1\right] \to P^{**};$$

and obviously

$$\left[z_1^0, z_1^1, z_2^0, z_2^1\right] \to P'.$$

It is therefore not possible to obtain P' without applying lift-and-project on all binary variables. Moreover, as written in [9], to get any reduction of P, the application of lift-and-project must be carried out, before projection, with respect to all the binary variables z_i^j associated with a given variable x_i.

5.1.3. *Sherali-Adams non-compact reformulation*

In the case of the Sherali-Adams non-compact reformulation (see definition 3), a disjunction on variable z_i^j: $z_i^j = 0$ or $z_i^j = 1$, corresponds to a disjunction on variable x_i: $x_i \le b_i - 1$ or $x_i = b_i$ if $j = b_i$ and $x_i \le b_i$ or $x_i = j$ otherwise, which is obviously not very interesting if $j \ne b_i$.

$$\left[z_1^1\right] \to P, \left[z_1^2\right] \to P ;$$
$$\left[z_1^1, z_1^2\right] \;\to\; P^*, \; \left[z_1^1, z_2^1\right] \;\to\; P, \; \left[z_1^1, z_2^2\right] \;\to\; P, \left[z_1^2, z_2^1\right] \to P, \left[z_1^2, z_2^2\right] \to P ;$$
$$\left[z_1^1, z_1^2, z_2^1\right] \to P^*, \left[z_1^1, z_1^2, z_2^2\right] \to P^*, \left[z_1^1, z_2^1, z_2^2\right] \to P^{**}, \left[z_1^2, z_2^1, z_2^2\right] \to P^{**}.$$

We observe here the same behavior as with the compact reformulation as these are more or less the same reformulations in our case.

5.1.4. *Weak non-compact reformulation*

In the case of the weak non-compact reformulation (see remark 1), a disjunction on variable z_i^j: $z_i^j = 0$ or $z_i^j = 1$, corresponds to a disjunction on variable x_i: $x_i \le b_i - 1$ or $x_i \ge 1$, which does not depend on j and is obviously not very interesting.

$$\left[z_1^1\right] \to P, \left[z_1^2\right] \to P ;$$
$$\left[z_1^1, z_1^2\right] \;\to\; P^*, \; \left[z_1^1, z_2^1\right] \;\to\; P, \; \left[z_1^1, z_2^2\right] \;\to\; P, \left[z_1^2, z_2^2\right] \to P, \left[z_1^2, z_2^2\right] \to P ;$$
$$\left[z_1^1, z_2^2, z_2^1\right] \to P^*, \left[z_1^1, z_2^1, z_2^2\right] \to P^*, \left[z_1^1, z_2^2, z_2^2\right] \to P^{**}, \left[z_1^2, z_2^1, z_2^2\right] \to P^{**}.$$

We observe here the same behavior as with the Sherali-Adams reformulation.

5.1.5. *Strong non-compact reformulation*

As explained in section 4., in the case of the strong non-compact reformulation (see definition 4), a disjunction on variable z_i^j: $z_i^j = 0$ or $z_i^j = 1$, corresponds to a disjunction on variable x_i: $x_i \le j - 1$ or $x_i \ge j$, which corresponds to the disjunctions described by Owen and Mehrotra in [8].

$$\left[z_1^1\right] \to P^*, \left[z_1^2\right] \to P ;$$
$$\left[z_1^1, z_1^2\right] \;\to\; P^*, \; \left[z_1^1, z_2^1\right] \;\to\; P', \; \left[z_1^1, z_2^2\right] \;\to\; P^*, \left[z_1^2, z_2^2\right] \to P^{**}, \left[z_1^2, z_2^2\right] \to P ;$$
$$\left[z_1^1, z_2^2, z_2^1\right] \to P', \left[z_1^1, z_1^2, z_2^2\right] \to P^*, \left[z_1^1, z_2^1, z_2^2\right] \to P', \left[z_1^2, z_2^1, z_2^2\right] \to P^{**}.$$

We remark that in the practical case of linear programming, since the only vertex with fractional components is vertex $\left(\frac{1}{2}, \frac{1}{2}, \frac{3}{2}\right)$, lift-and-project would be applied to variable z_1^1 (or symmetrically z_2^1). Afterwards, by projecting the vertices of the polyhedron obtained and by keeping only those not in I, the only remaining vertex is $\left(1, \frac{1}{3}, 1\right)$. Thus, lift-and-project would be applied to variable z_2^1. We would therefore obtain P' after two convexification steps only.

5.1.6. *Remark on disjunctions applied to integer variables*

As explained in remark 10, the sequential application of the lift-and-project procedure to the polyhedron P_{01} provides a better strengthening than the sequential application of the disjunctions proposed by Owen and Mehrotra. As an illustration, on our example, four disjunctions are possible: $(x_1 \le 0) \vee (x_1 \ge 1)$, $(x_1 \le 1) \vee (x_1 \ge 2)$, and similarly for x_2. The sequen-

tial application of these four disjunctions, whatever the order, repeated any number of times, does not lead to the convex hull of mixed-integer solutions to the problem. On the contrary, the sequential application of the disjunctions of the form $\left(z_i^j = 0\right) \vee \left(z_i^j = 1\right)$ in the binary reformulation corresponds to the simultaneous application of the disjunctions $(x_i \leq j - 1) \vee (x_i \geq j)$ in the original space, which naturally lead to the convex hull of the mixed-integer solutions to the problem. I.e., in our case,

$$P' = \text{conv}[(\{(x_1, x_2, y) \in P | x_1 \leq 0\} \cup \{(x_1, x_2, y)$$
$$\in P | x_1 \geq 1\}) \cap (\{(x_1, x_2, y) \in P | x_2 \leq 0\}$$
$$\cup \{(x_1, x_2, y) \in P | x_2 \geq 1\})]$$

Whereas, let $\Pi[\cdot]$ denote the projection onto the original space, and let: $Q =$

$$\text{conv}(\{(z, y) \in P_{01} \| z_1^1 = 0\} \cup \{(z, y) \in P_{01} | z_1^1 = 1\})$$

Then $P' =$

$$\Pi[\text{conv}(\{(z, y) \in Q | z_2^1 = 0\} \cup \{(z, y) \in Q \| z_2^1 = 1\})]$$

5.2. *Second example*

This second example is the one discussed in [9].

We are interested in building the convex hull P' of the integer points of the polyhedron $P = \left\{ x_1, x_2 \in [0, 4]^2 \middle| 3x_1 + 5x_2 \leq 20, \ 5x_1 + 3x_2 \leq 20 \right\}$. The case of the compact reformulation is detailed in [9]. The authors show that, firstly, lift-and-project must be applied on all 6 binary variables of the reformulation to obtain P', and secondly that it is necessary to apply it to at least three binary variables to obtain a polyhedron whose projection into the original space is smaller than P.

In the case of the strong non-compact reformulation (see 5.1.5.), the application of the lift-and-project procedure to the various variables results in the following polyhedra (we apply each time lift-and-project on a vertex of the polyhedron such that its projection into the initial space does not satisfy integrality constraints):

Let: $S' = \{(0, 0), (4, 0), (4, 0)\}$. We have $P' = \text{conv}(S')$.

$$\left[z_1^3\right] \to \text{conv}\left(S' \cup \left\{\left(3, \tfrac{5}{3}\right), \left(2, \tfrac{14}{5}\right)\right\}\right)$$
$$\left[z_1^3, z_2^2\right] \to \text{conv}\left(S' \cup \left\{\left(\tfrac{17}{5}, 1\right), \left(2, \tfrac{14}{5}\right)\right\}\right)$$
$$\left[z_1^3, z_2^2, z_1^4\right] \to \text{conv}\left(S' \cup \left\{\left(2, \tfrac{14}{5}\right)\right\}\right)$$
$$\left[z_1^3, z_2^2, z_1^4, z_2^3\right] \to \text{conv}\left(S' \cup \left\{\left(\tfrac{5}{3}, 3\right)\right\}\right)$$

$$\left[z_1^3, z_2^2, z_1^4, z_2^3, z_1^2\right] \to \text{conv}\left(S' \cup \left\{\left(1, \tfrac{17}{5}\right)\right\}\right)$$
$$\left[z_1^3, z_2^2, z_1^4, z_2^3, z_1^2, z_2^4\right] \to P'$$

Initial polyhedron P After one cut

After two cuts   After three cuts

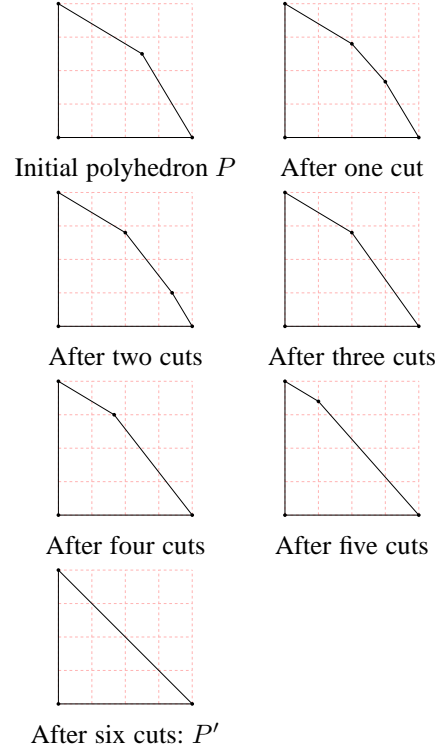After four cuts   After five cuts

After six cuts: P'

Fig. 2. All stages of the application of the method on the example from [9]

We observe (see figure 2) that while, to obtain P', it is necessary to apply the method to as many variables as in the compact reformulation case, each application leads to a projection nearer to the convex hull than the previous one, which was not the case with the compact reformulation. Therefore, in accordance with the methodology described in section 4., it is possible to stop at any stage of the application of lift-and-project on the binary variables and go back to the initial formulation while still taking advantage of a strengthening not necessarily obtainable by applying existing methods (such as mixed-integer Gomory cuts) on the initial polyhedron.

5.2.1. *Remark on disjunctions applied to integer variables*

The remarks of subsection 5.1.6. also apply here. For example, the disjunction on z_1^3 is equivalent to the disjunction $(x_1 \leq 2) \vee (x_1 \geq 3)$; the sequential disjunc-

tions on z_1^3 then z_2^2 are equivalent to the simultaneous disjunctions $(x_1 \leq 2) \vee (x_1 \geq 3)$ and $(x_2 \leq 1) \vee (x_2 \geq 2)$; while the sequential disjunctions $(x_1 \leq 2) \vee (x_1 \geq 3)$ then $(x_2 \leq 1) \vee (x_2 \geq 2)$ lead to a larger polyhedron $\operatorname{conv}\left(S' \cup \left\{\left(\frac{17}{5}, 1\right), \left(\frac{46}{17}, 2\right), \left(2, \frac{14}{5}\right)\right\}\right)$ as illustrated on figure 3.

Simultaneous disjunctions   Sequential disjunctions

Fig. 3. Difference between the simultaneous and sequential application of disjunctions $(x_1 \leq 2) \vee (x_1 \geq 3)$ and $(x_2 \leq 1) \vee (x_2 \geq 2)$ on P.

Again, the better efficiency of the sequential disjunctions applied to the binary reformulation comes from the property that sequential disjunctions on mixed-binary problems are equivalent to simultaneous disjunctions, which is not the case for general mixed-integer problems.

6. Practical experiments

6.1. *Implementation in COIN/Cgl*

We have implemented the proposed cut generation through strong non-compact binary reformulation as a cut generation method for the Cgl library of COIN[7]. This implementation corresponds to the reformulation into binary variables, then the repeated application of cut generation methods to the resulting mixed-binary problem and the application of algorithm 2 to generate inequalities valid for the original problem. Algorithm 3 provides details on the procedure implemented while algorithm 4 provides details on the lift-and-project procedure used by algorithm 3.

At this stage, some implementation problems remain unsolved and left for future research. Notably, when the integer variables are unbounded, a preliminary computation of implied bounds is necessary, and even when bounds are known, if the allowed range of integer variables is very large, the number of binary variables in the reformulation may be too large to be handled easily. A few suggestions regarding the computational efficiency are given at the end of this section.

Step 1: Let (x_0, y_0) be an optimal vertex of the continuous relaxation of problem (1), and v_0 its value. We suppose that x_0 has at least one fractional component.
Step 2: Reformulate into binary variables the integer variables of the problem to obtain a problem such as:

$$\min_{(z,y) \in \mathbb{R}^m \times \mathbb{R}^p} dTz + ey \qquad \text{s.t.} \quad \hat{A}z + \hat{B}y \geq \hat{c} \quad (5)$$

Step 3: Let z_0 associated with x_0.
Step 4: (Cut generation on the transformed problem.)
  (1) Initialization: $i \leftarrow 0$
  (2) Apply lift-and-project to separate (z_i, y_i) (i.e., algorithm 4 below with $z' = z_i$ and $y' = y_i$). Strengthen the polyhedron $\hat{P}_{01} = \left\{(z,y) \in \mathbb{R}^m \times \mathbb{R}^p \,\middle|\, \hat{A}z + \hat{B}y \geq \hat{c}\right\}$ by modifying matrices $\hat{A}$, $\hat{B}$, and vector $\hat{c}$ using the generated cuts.
  (3) $i \leftarrow i + 1$
  (4) If cuts have been generated, compute an optimal vertex of the modified problem (5). Let (z_i, y_i) be the vertex and v_i the optimal value.
  (5) As long as z_i has at least one fractional component, that cuts have been generated during this step, and that v_i has sufficiently decreased relatively to v_{i-1}, i.e., $|v_{i-1} - v_i| \leq 0.001 \times (|v_i| + 1)$, and/or that the maximum number of rounds has not been exceeded, go back to step 4 (2).
Step 5: The cuts generated so far where used to strengthen P_{01} into $\hat{P}_{01}$. To finally generate cuts to strengthen P, using the strengthened reformulation $\hat{P}_{01}$, apply algorithm 2 on polyhedron $\hat{P}_{01}$, starting from step 2, with $x' = x_0$ and $y' = y_0$.

Algorithm 3. *Implementation of the cut generation using binary reformulation*

6.2. *Application to MIPLIB problems using a lift-and-project cut generator*

We have applied the methodology described above to MIPLIB[3] problems for which at least one integer variable is not binary. These problems have already been studied in the context of reformulation into binary variables in [9].

We have compared the reduction of the integrality gap obtained through the use of a lift-and-project cut generator on either the original problem or on its strong non-

(1) Let (z', y') an optimal vertex of P_{01} to be
separated from P'_{01}. We suppose that z' has
at least one fractional component. Let $J =$
$\{i \in 1, \ldots, m \,|\, z'_i \notin \{0, 1\}\}$, i.e., the set of indices
of binary variables that are fractional at the opti-
mal vertex considered.
(2) For all $i \in J$:
  (a) Consider the problem $\min_{(\alpha, \beta, \gamma)} \alpha z' + \beta y' -$
γ subject to $\alpha z + \beta y \geq \gamma$ is valid for
$\mathrm{conv}(\{(z, y) \in P_{01} | z_i = 0\} \cup \{(z, y) \in$
$P_{01} | z_i = 1\})$. See for example [1] for a prac-
tical formulation.
  (b) If the minimum of this problem is 0, then
no cut is generated. Otherwise, the problem
is unbounded. Let (α, β, γ) be an unbounded
ray. Then $\alpha z + \beta y \geq \gamma$ is a cut separating
(z', y') from P'_{01}.
(3) Return the list of cuts generated during the previ-
ous step. Remark that P_{01} is kept unchanged dur-
ing the course of step 2. All the cuts generated by
the above procedure are applied to P_{01} upon ter-
mination of this step.

Algorithm 4. One round of lift-and-project cut generation

compact reformulation. Lift-and-project is used in the
comparison since it is the only cutting plane approach
for which we can guarantee the generation of separating
inequalities using the proposed procedure (see proposi-
tion 9). In both cases, 5 rounds of sequential lift-and-
project cut generation are applied to all fractional bi-
nary variables. I.e., algorithm 4 is applied to an opti-
mum vertex of a polyhedron P_{01}; the generated cuts are
added to P_{01}; an optimal vertex of the new polyhedron
is obtained; then algorithm 4 is applied again, and so
on, five times successively. In the binary reformulation
case, the resulting cuts are then projected using the algo-
rithm 2. The problem `noswot` has been omitted since
it does not present any integrality gap. Table 1 presents
the characteristics of the problems in terms of number
of continuous, binary and general integer variables, as
well as the number of binary variables added by the
reformulation of general integer variables. In this im-
plementation, we choose to avoid reformulating general
integer variables whose definition interval is larger than
500. Such variables would create a considerable num-
ber of binary variables (32000 in the case of `bell3a`
and 94600 in the case of `bell5`), which should be bet-
ter handled by either reformulating a subset of the defi-
nition interval, or some implied bounds computation to

reduce the definition interval. Moreover, it seems intu-
itively that in most problems, given the finite precision
of the computational tools involved, integer variables
with large definition intervals behave quite similarly to
continuous variables.

Table 1

Problem	Variables			
	Continuous	Binary	Integer	Integer as binary
`arki001`	850	442	96	608
`bell3a`	62	39	32	0
`bell5`	46	30	28	600
`blend2`	89	239	25	66
`flugpl`	7	0	11	198
`gen`	720	144	6	144
`gesa2`	816	240	168	432
`gesa2_o`	504	384	336	864
`gesa3`	768	216	168	504
`gesa3_o`	480	336	336	1008
`gt2`	0	24	164	1148
`qnet1`	124	1288	129	675
`qnet1_o`	124	1288	129	675
`rout`	241	300	15	30

*Number of continuous, binary, general integer non bi-
nary variables of MIPLIB 3.0 problems, as well as the
number of binary variables added by the reformulation
of the general integer variables.*

Since no suitable lift-and-project cutting plane gen-
erator is available in Cgl, we developed our own im-
plementation. This implementation is not optimized so
that the only significant performance measurement is
the reduction of the integrality gap at the root node of
the problem. [5] provide the basis for a much more effi-
cient, but highly complex implementation, suitable for
branch-and-cut.

Table 2 presents the results. *LP value* corresponds to
the value of the continuous relaxation ; *MIP value* to
the optimal value (or in the case of `arki001`, the best
integer value known) ; *L&P value* to the value of the
relaxation after applying lift-and-project (5 rounds) ;
B&P value to the value of the relaxation after applying
lift-and-project (5 rounds) to the strong non-compact
reformulation (i.e., the result of algorithm 3). Finally,
the improvement corresponds to the difference in value
between the two methods expressed as a percentage
of the integrality gap that remains to be closed after
applying lift-and-project to the original problem, i.e.,
$\frac{B\&P\,value - L\&P\,value}{MIP\,value - L\&P\,value}$. Three problems exhibit no or a

small improvement, all other problems exhibit significant improvements.

Table 3 presents the number of cuts generated by each method that are tight at the optimum of the strengthened relaxation. We observe that, given the projection performed after the cut generation on the binary expanded space, the number of cuts kept in the original space is comparable with the number of cuts generated without reformulation.

Table 3

Problem	Improvement (%)	L&P cuts	B&P cuts
arki001	18.7	58	65
bell3a	0	10	8
bell5	0.4	13	6
blend2	20.8	15	7
flugpl	13.1	0	7
gen	4.1	64	33
gesa2	24.3	54	79
gesa2_o	47.8	65	90
gesa3	94.0	34	87
gesa3_o	94.1	30	86
gt2	92.7	0	14
qnet1	34.1	30	25
qnet1_o	69.4	0	31
rout	8.8	37	38

Comparison of the number of cuts tight at the optimum of the strengthened relaxation, between L&P and B&P, using 5 rounds of lift-and-project on the MIPLIB 3.0 problems (column "Improvement" reproduced from table 2).

Finally, table 4 presents the computing times of each method obtained on a 1280 MHz Sun UltraSPARC III, using CPLEX 9.0 as the underlying linear solver. We observe that, the computing times are most of the time significantly higher for B&P than for L&P, which is a consequence of the much larger programs generated by the reformulation in our preliminary implementation. Nevertheless, we believe these computing times are not representative of a real-world implementation. Indeed, many methods can be devised to improve the numerical efficiency:

(1) Sophisticated implementations of lift-and-project[1,2] ignore all variables that are at a bound in the continuous relaxation solution, greatly reducing the cutting plane generation problem size. In our case, the solution to the reformulated problem associated (see definition 4) with the optimal solution to the continuous relaxation will have the same number of fractional variables as the the optimal solution to the continuous relaxation. The cutting plane generation problem will therefore initially be of the same size as without reformulation. Subsequent rounds of lift-and-project do not share this property, and the number of fractional variables will increase: care should be taken to ensure the optimal solutions of the strengthened continuous relaxation of the reformulated problem are chosen among the many possible equivalent ones as having as much binary variables at bounds as possible, for example by introducing a bias in the objective such as $\epsilon \sum_{i=1}^{n} \sum_{j=1}^{b_i} j z_i^j$, for ϵ small enough or by solving an additional small optimization problem. I.e., let (z', y) an optimal solution to the strengthened continuous relaxation of the reformulated problem. Before performing another lift-and-project round, we might want to choose another solution (z, y) that minimize $\sum_{i=1}^{n} \sum_{j=1}^{b_i} j z_i^j$ subject to the cutting planes generated so far, and the constraint $Tz = Tz'$.

(2) Only a subset of the general integer variables could be reformulated, and, for variables with large definition intervals, only a subset of the definition interval, around the solution to the continuous relaxation, could be reformulated.

(3) And finally, computing times would be significantly reduced and probably also much less impacted by the problem size increase induced by the reformulation, if an efficient lift-and-project implementation, such as the one described in [5] was used [2].

To conclude this experimentation, the use of the proposed cut generation procedure, while computationally heavy, avoids applying branch-and-bound to the binary problem, and does not lead to larger programs. On the contrary, as observed in [9], the use of binary reformulation alone creates a considerable amount of variables negatively impacting the time required to solve the problem.

This limited experiment demonstrates the ability to take advantage of cutting plane approaches dedicated to binary problems, such as lift-and-project, in the context of general integer variables. Nevertheless, much work remains to create an efficient implementation of this technique.

---

[2] No such implementation is currently available to the author.

Table 2

Problem	LP value	MIP value	L&P value	B&P value	Improvement (%)
arki001	7579599.8	7580813*	7579599.8	867726.15	18.7
bell3a	862578.64	878430.32	867726.15	867726.13	0
bell5	8608417.9	8966406.5	8906426.9	8906662.9	0.4
blend2	6.9156751	7.598985	7.1681526	7.2577345	20.8
flugpl	1167185.7	1201500	1167185.7	1171670	13.1
gen	112130.04	112313	112248.4	112251.02	4.1
gesa2	25476490	25779856	25591179	25637000	24.3
gesa2_o	25476490	25779856	25591200	25681342	47.8
gesa3	27833632	27991043	27851253	27982605	94.0
gesa3_o	27833632	27991043	27851253	27982741	94.1
gt2	13460.233	21166	13460.233	20603.861	92.7
qnet1	14274.103	16029.693	14458.939	14993.899	34.1
qnet1_o	12095.572	16029.693	12095.572	14824.597	69.4
rout	981.86429	1077.56	996.478	1003.6282	8.8

Comparison of the strengthened relaxation value, between L&P and B&P, using 5 rounds of lift-and-project on the MIPLIB 3.0 problems.

Table 4

Problem	Improvement (%)	L&P time (sec.)	B&P time (sec.)
arki001	18.7	1237	16877
bell3a	0	1	1
bell5	0.4	1	37
blend2	20.8	7	10
flugpl	13.1	1	13
gen	4.1	50	143
gesa2	24.3	27	261
gesa2_o	47.8	28	354
gesa3	94.0	13	900
gesa3_o	94.1	11	976
gt2	92.7	1	42
qnet1	34.1	1768	6691
qnet1_o	69.4	1	991
rout	8.8	146	135

Comparison of the computing times, in seconds, between L&P and B&P, using 5 rounds of lift-and-project on the MIPLIB 3.0 problems (column "Improvement" reproduced from table 2).

7. Conclusion

We have proposed a practical approach of cut generation through binary expansion using a specific reformulation into binary variables, called here the "strong non-compact reformulation", that answers some theoretical concerns and limitations raised in [9], namely the necessity to apply the lift-and-project convexification procedure to all binary variables associated with an integer variable before obtaining a strengthening of the original problem, a concern answered by proposition 9. Also, instead of applying branch-and-bound to the binary reformulation of the problem (which would lead to poor performance) our approach only uses the binary reformulation of the problem to find cutting planes to strengthen the original problem (in general integer variables) to which branch-and-bound is subsequently applied.

The implementation of our approach, combined with sequential lift-and-project, results in reduced integrality gaps for some problems, with no significant increase in the number of generated cutting planes. This suggests that the generation of cuts from the strong non-compact reformulation of the problem might be an interesting method in some cases.

While lift-and-project is the only cutting planes approach for which we can guarantee the generation of separating inequalities, it might be interesting to combine our methodology with other cutting planes approaches, particularly those dedicated to binary problems, to somehow extend them to the general integer variables case.

Other possible research directions include studying the relationship between the cuts resulting from applying disjunctive programming to the original (non binary) problem and those resulting from applying lift-and-project on the strong non-compact reformulation. A comparison between these two cutting plane generation approaches, both from a theoretical and a numeri-

cal point of view, will be the subject of future research.

References

[1] E. Balas, S. Ceria, and G. Cornujols, *A lift-and-project cutting plane algorithm for mixed 0-1 programs*, Mathematical Programming **58** (1993), pp. 295–323.

[2] E. Balas, S. Ceria, and G. Cornujols, *Mixed 0-1 programming by lift-and-project in a branch-and-cut framework*, Management Science **42** (1996), no. 9, 1229–1246.

[3] R. E. Bixby, S. Ceria, C. M. McZeal, and M. W. P. Savelsbergh, *An updated mixed integer programming library: MIPLIB 3.0*, Technical Report TR98-03, Department of Computational and Applied Mathematics, Rice University, 1996.

[4] P. Bonami, *Étude et mise en œuvre d'approches polydriques pour la rsolution de programmes en nombre entiers ou mixtes gnraux*, Ph.D. thesis, Universit Paris VI, 2003.

[5] 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*, Math. Program., Ser. B **94** (2003), 221–245.

[6] W. J. Cook, R. Kannan, and A. Schrijver, *Chvtal closures for mixed integer programming problems*, Mathematical Programming **47** (1990), 155–174.

[7] R. Lougee-Heimer, F. Barahona, B. Dietrich, J. P. Fasano, J. Forrest, R. Harder, L. Ladanyi, T. Pfender, T. Ralphs, M. Saltzman, and K. Schienberger, *The COIN-OR initiative: Open-source software accelerates operations research progress*, ORMS Today **28** (2001), no. 5, 20–22.

[8] J. Owen and S. Mehrotra, *A disjunctive cutting plane procedure for general mixed-integer linear programs*, Mathematical Programming **89** (2001), no. 3, pp. 437–448.

[9] ———, *On the value of binary expansions for general mixed-integer linear programs*, Operations Research **50** (2002), no. 5, pp. 810–819.

[10] H. D. Sherali and W. P. Adams, *A reformulation linearization technique for solving discrete and continuous nonconvex problems*, ch. 4, Kluwer, Boston, MA, 1999.