Scanning integer points with lex-cuts: A finite cutting plane algorithm for integer programming with linear objective

Michele Conforti∗ Marianna De Santis† Marco Di Summa∗ Francesco Rinaldi∗

July 25, 2018

Abstract

We consider the integer points in a box $B$ ordered by a lexicographic rule defined by a lattice basis. To each integer point $x$ in $B$ we associate a cutting plane (lex-cut) that eliminates the integer points in $B$ that are lexicographically smaller than $x$ and is satisfied by the others. The family of lex-cuts contains the Chvátal–Gomory cuts, but does not contain and is not contained in the family of split cuts. This provides a finite cutting plane method to solve the integer program $\min\{cx : x \in S \cap \mathbb{Z}^n\}$, where $S \subset \mathbb{R}^n$ is a compact set and $c \in \mathbb{Z}^n$. We analyze the worst-case behavior of our algorithm and propose some variants.

1 Introduction

The area of nonlinear integer programming is rich in applications but quite challenging from a computational point of view. We refer to the recent articles [5, 7] for comprehensive surveys on these topics. The tools that are mainly used are sophisticated techniques that exploit relaxations, constraint enforcement (e.g., cutting planes) and convexification of the feasible set. Reformulations in an extended space and cutting planes for nonlinear integer programs have been investigated and proposed for some time, see e.g. [8, 11, 15]. This line of research mostly provides a nontrivial extension of the theory of Disjunctive Programming to the nonlinear case. To the best of our knowledge, these results are obtained under some restrictive conditions: typically, convexity of the feasible set $S$, or $S \subseteq \{0, 1\}^n$ (these cases cover some important areas of application).

In this paper we focus on linear inequalities that we use as cuts. As the convex hull of $S \cap \mathbb{Z}^n$ is a polytope when $S \subseteq \mathbb{R}^n$ is compact, a finite number of linear inequalities suffices for its characterization and only $n$ such inequalities determine an optimal point. Furthermore, some relaxations are polyhedral: most notably, Dadush, Dey and Vielma [10] proved that if $S$ is a compact and convex set, then its Chvátal closure is a polytope (whereas this is not the case for the split closure of $S$ [9]).

However, nonlinear inequalities are fundamental in the characterization of the convex hull of some nonlinear sets that strengthen the original formulation. For instance, Burer and

∗Dipartimento di Matematica, Università di Padova, Via Trieste, 63 - 35121 Padova - Italy
†Dipartimento di Ingegneria Informatica Automatica e Gestionale, Sapienza Università di Roma, Via Ariosto, 25 - 00185 Roma - Italy
Kılınç-Karzan [6], extending several results (see, e.g., [1, 3, 4, 14]), show that the convex hull of the intersection of a second-order-cone representable set and a single homogeneous quadratic inequality can be described by adding a single nonlinear inequality, defining an additional second-order-cone representable set.

In this paper we present a finite cutting plane algorithm for problems of the form

$$\min \{cx : x \in S \cap \mathbb{Z}^n\}, \quad (1.1)$$

where $S$ is a compact subset of $\mathbb{R}^n$ (not necessarily convex or connected) and $c \in \mathbb{Z}^n$. This algorithm uses a new family of cutting planes which includes the Chvátal–Gomory cuts, but neither it contains nor is contained in the family of split cuts.

We consider the integer points in a box $B$, ordered by a lexicographic rule, associated with a lattice basis. To each integer point $x$ in $B$, we associate a cutting plane (lex-cut) that is violated by all the integer points in $B$ that are lexicographically smaller than $x$ and is satisfied by the other integer points in $B$.

Our algorithm recursively solves optimization problems of the type

$$\min \{cx : x \in S \cap P\},$$

where $P$ is a polyhedron, and we assume that such an algorithm is available as a black box.

Note that when $S$ is a convex set, this is a convex program that is (in principle) efficiently solvable. To the best of our knowledge, our work represents the first attempt to define a finite cutting plane algorithm for the general problem (1.1) with $S$ compact.

Deriving a finite cutting plane algorithm that uses a well defined family of inequalities does not seem to be straightforward. The oldest and most notable example is Gomory’s finite cutting plane algorithm for bounded integer programs based on fractional cuts [12, 13]. Balas, Ceria and Cornuéjols [2] give a finite cutting plane algorithm for mixed 0/1 problems based on Lift-and-Project cuts. Also in these algorithms, crucial to the detection of a cutting plane is the computation of a lexicographically optimal solution.

2 Lexicographic orderings and lex-cuts

A lattice basis of $\mathbb{Z}^n$ is a set of $n$ linearly independent vectors $c^1, \ldots, c^n \in \mathbb{Z}^n$ such that for every $v \in \mathbb{Z}^n$ we have that $\lambda_1, \ldots, \lambda_n \in \mathbb{Z}$ in the unique expression $v = \sum_{i=1}^{n} \lambda_i c^i$.

A lattice basis induces a lexicographic ordering $\prec$ on $\mathbb{R}^n$ defined as follows: given $x^1, x^2 \in \mathbb{R}^n$, we have $x^1 \prec x^2$ if and only if $x_1 \neq x_2$ and $c^i x^1 < c^i x^2$, where $i$ is the smallest index for which $c^i x^1 \neq c^i x^2$. We remark that such an index $i$ exists when $x^1 \neq x^2$, as the vectors $c^1, \ldots, c^n$ are linearly independent. We use $\preceq, \succeq, \succ$ with the obvious meaning. We will make extensive use of the following fact:

**Observation 2.1.** Given $\bar{x} \in \mathbb{R}^n$, $\bar{x} \in \mathbb{Z}^n$ if and only if $c^i \bar{x} \in \mathbb{Z}$ for all $i = 1, \ldots, n$.

**Proof.** Since $\{c^1, \ldots, c^n\}$ is a lattice basis of $\mathbb{Z}^n$, the $n \times n$ matrix $C$ whose rows are $c^1, \ldots, c^n$ is unimodular, i.e., it is an integer matrix with determinant 1 or $-1$. Therefore $C^{-1}$ is unimodular as well and the result follows.

Given a subset $S$ of $\mathbb{R}^n$, a point $\bar{x} \in S$ is extreme if $\bar{x}$ cannot be expressed as a proper convex combination of points in $S$. Given $c \in \mathbb{Z}^n$, an extreme point $\bar{x} \in S$ is an extreme solution if it solves the program $\min \{cx : x \in S\}$. (Extreme solutions exist if $S$ is compact.)
We now assume $S$ to be a nonempty subset of $\mathbb{R}^n$ which is compact, but not necessarily connected or convex. A vector $\bar{x} \in S$ is lexicographically minimum (in $S$) if $\bar{x} < x$ for every $x \in S \setminus \{\bar{x}\}$.

Given $c \in \mathbb{Z}^n$ with relatively prime entries and a lattice basis $\{c^1, \ldots, c^n\}$ with $c^1 = c$, let $\bar{x}$ be a solution of the following programs:

- $c^1 \bar{x} = \min \{c^1 x : x \in S\}$;
- $c^2 \bar{x} = \min \{c^2 x : x \in S, c^1 x = c^1 \bar{x}\}$;
- $c^3 \bar{x} = \min \{c^3 x : x \in S, c^1 x = c^1 \bar{x}, c^2 x = c^2 \bar{x}\}$;
- $\ldots$
- $c^n \bar{x} = \min \{c^n x : x \in S, c^1 x = c^1 \bar{x}, \ldots, c^{n-1} x = c^{n-1} \bar{x}\}$.

Since $S$ is nonempty and compact, the above minima are well-defined. Furthermore these conditions uniquely define $\bar{x}$, as the vectors $c^1, \ldots, c^n$ are linearly independent.

**Observation 2.2.** Let $\bar{x}$ be defined as above. Then $\bar{x}$ is a lexicographically minimum solution of the program $\min \{cx : x \in S\}$ that is an extreme solution.

**Proof.** As $c = c^1$, the first of the above conditions shows that $\bar{x}$ is an optimal solution of the program $\min \{cx : x \in S\}$. The other conditions guarantee that $\bar{x}$ is lexicographically minimum. Now assume that $\bar{x}$ is a proper convex combination of $x_1, x_2 \in S$ and let $i$ be the lowest index such that $c^i x_1 \neq c^i \bar{x} \neq c^i x_2$. Then $c^i \bar{x} > \min \{c^i x_1, c^i x_2\}$, a contradiction to the fact that $\bar{x}$ is lexicographically minimum. \qed

Given $\ell_i, u_i \in \mathbb{Z}$ for $i = 1, \ldots, n$ with $\ell_i \leq u_i$, we consider the set $B = \{x \in \mathbb{R}^n : \ell_i \leq c^ix \leq u_i, i = 1, \ldots, n\}$. We call $B$ a box as it is a unimodular transformation of a parallelepiped. We assume throughout the paper that $\ell_i = 0, i = 1, \ldots, n$. (This is without loss of generality and simplifies notation and arguments.)

We define $d_n = 1$ and $d_i = \prod_{i < j \leq n}(u_j + 1)$ for $i = 1, \ldots, n - 1$. Since $c^1, \ldots, c^n$ form a lattice basis of $\mathbb{Z}^n$, we have

$$|B \cap \mathbb{Z}^n| = \prod_{i=1}^{n}(u_i + 1). \quad (2.1)$$

Given $\bar{x} \in B \setminus \{0\}$, the leading index $i_\bar{x}$ is the largest index $i$ such that $c^i \bar{x} > 0$. (Note that the leading index is well-defined.) We also define the set $S_\bar{x} = \{x \in B \cap \mathbb{Z}^n : x \prec \bar{x}\}$.

Given $\bar{x} \in B \cap \mathbb{Z}^n \setminus \{0\}$, we have that

$$|S_\bar{x}| = \sum_{j=1}^{i_\bar{x}} d_j c^j \bar{x} = \sum_{j=1}^{n} d_j c^j \bar{x}. \quad (2.2)$$

The lex-cut associated to $\bar{x}$ is the inequality

$$\sum_{j=1}^{i_\bar{x}} d_j c^j x \geq \sum_{j=1}^{i_\bar{x}} d_j c^j \bar{x}. \quad (2.3)$$

**Lemma 2.3.** The lex-cut (2.3) associated to $\bar{x}$ is satisfied by $x^* \in B \cap \mathbb{Z}^n$ if and only if $x^* \in B \setminus S_\bar{x}$. 

3
Proof. Let \( x^* \in (B \cap \mathbb{Z}^n) \setminus S_\bar{x} \) and let \( x' \) satisfy
\[
c^j x' = c^j x^*, \quad j \leq i_\bar{x}; \quad c^j x' = 0, \quad j > i_\bar{x}.
\]
Then \( \bar{x} \preceq x' \preceq x^* \). Therefore \( |S_\bar{x}| \leq |S_{x'}| \) and by (2.2) we have that \( \sum_{j=1}^{i_\bar{x}} d_j c^j \bar{x} \leq \sum_{j=1}^{i_\bar{x}} d_j c^j ' \). Since \( \sum_{j=1}^{i_\bar{x}} d_j x_j' = \sum_{j=1}^{i_\bar{x}} d_j x_j^* \), this shows that \( x^* \) and \( x' \) satisfy (2.3).

Let \( x^* \in S_\bar{x} \) and define \( x' \) as above. Then \( x' \preceq x^* < \bar{x} \). Since \( |S_{x'}| < |S_\bar{x}| \), by (2.2) we have that \( \sum_{j=1}^{i_\bar{x}} d_j c^j x' < \sum_{j=1}^{i_\bar{x}} d_j c^j \bar{x} \). Since \( \sum_{j=1}^{i_\bar{x}} d_j x_j' = \sum_{j=1}^{i_\bar{x}} d_j x_j^* \), we have that \( x^* \) and \( x' \) violate inequality (2.3). \( \square \)

3 An application to Nonlinear Integer Programming

Let \( S \) be a family of compact (again, not necessarily connected or convex) subsets of \( \mathbb{R}^n \) with the following property:

\[ \text{if } S \in S \text{ and } H \text{ is a closed halfspace in } \mathbb{R}^n, \text{ then } S \cap H \in S. \]

Linear optimization over \( S \) is the following problem: given \( S \in S \) and \( c \in \mathbb{Z}^n \), determine an optimal solution to the problem \( \min \{cx : x \in S\} \) or certify that \( S = \emptyset \). (Since \( S \) is compact, either \( S = \emptyset \) or the minimum is well defined.)

Integer linear optimization over \( S \) is defined similarly, but the feasible region is \( S \cap \mathbb{Z}^n \), the set of integer points in \( S \).

We prove that an oracle for solving linear optimization over \( S \) suffices to design a finite cutting plane algorithm that solves integer linear optimization over \( S \).

We now make this statement more precise. Given a subset \( S \) of \( \mathbb{R}^n \) and \( c \in \mathbb{Z}^n \), let \( \bar{x} \in S \) be an extreme solution of the program \( \min \{cx : x \in S\} \). A cutting plane is a linear inequality that is valid for \( S \cap \mathbb{Z}^n \) and is violated by the extreme solution \( \bar{x} \) (which in this case is in \( S \setminus \mathbb{Z}^n \)).

A (pure) cutting plane algorithm for integer linear optimization over \( S \) works as follows.

- Let \( S \in S \) and \( c \in \mathbb{Z}^n \) be given.
- If \( S = \emptyset \), then \( S \cap \mathbb{Z}^n = \emptyset \). Otherwise, find an extreme solution \( \bar{x} \) of \( \min \{cx : x \in S\} \).
- If \( \bar{x} \in S \cap \mathbb{Z}^n \), stop: \( \bar{x} \) is an optimal solution to \( \min \{cx : x \in S \cap \mathbb{Z}^n\} \).
- Otherwise, detect a cutting plane and let \( H \) denote the corresponding half-space. Replace \( S \) with \( S \cap H \) and iterate the procedure.

Algorithm 1 describes the procedure in detail, in particular how to find the extreme solution \( \bar{x} \) and a cutting plane whenever \( \bar{x} \notin S \cap \mathbb{Z}^n \). Note that since \( S \) is compact, numbers \( \ell_1^*, \ldots, \ell_n^*, u_1^*, \ldots, u_n^* \) (as defined in Algorithm 1) exist and can be determined by querying the linear optimization oracle \( 2n \) times. By Observation 2.1, an index \( k \) as in step 3 always exists when \( \bar{x} \notin \mathbb{Z}^n \).

Proposition 3.1. Inequality (3.1) defines a cutting plane. Algorithm 1 terminates after a finite number of iterations.

Proof. Assume \( \bar{x} \notin \mathbb{Z}^n \) and let \( k \) be the smallest index such that \( c^k \bar{x} \notin \mathbb{Z} \). Let \( x^* \) be defined as the unique vector satisfying the following conditions:
\[
c^j x^* = c^j \bar{x}, \quad j < k; \quad c^k x^* = \lfloor c^k \bar{x} \rfloor; \quad c^j x^* = 0, \quad j > k.
\]

4
Algorithm 1: Resolution of integer linear optimization over $S$

Input: $S \subseteq \mathbb{Z}^n \setminus \{0\}$ with relatively prime entries and a lattice basis \( \{e^1, \ldots, e^n\} \) of $\mathbb{Z}^n$, with $e^1 = c$. 

Output: an optimal integer solution $\bar{x}$ for the problem $\min \{cx : x \in S\}$ or a certificate that $S \cap \mathbb{Z}^n = \emptyset$.

1. Compute $\ell_i^* = \min \{c^i x : x \in S\}$, $u_i^* = \max \{c^i x : x \in S\}$, $1 \leq i \leq n$. Define $\ell_i = \lceil \ell_i^* \rceil$, $u_i = \lfloor u_i^* \rfloor$, $1 \leq i \leq n$. Apply a translation so that $\ell_i = 0$, $1 \leq i \leq n$ and let $B = \{x \in \mathbb{R}^n : \ell_i \leq c^i x \leq u_i, i = 1, \ldots, n\}$. Let $d_i = \prod_{j=i+1}^n (u_j + 1)$ for $i = 1, \ldots, n-1$ and $d_n = 1$.

2. If $S \cap B = \emptyset$, the given problem is infeasible: stop. Otherwise compute the lexicographically optimal solution $\bar{x}$ in $S \cap B$ with respect to $\{e^1, \ldots, e^n\}$.

3. If $\bar{x} \in \mathbb{Z}^n$ then return $\bar{x}$. Otherwise let $k$ be the smallest index such that $c^k \bar{x} \notin \mathbb{Z}$.

Replace $S$ with $S \cap H$, where $H$ is the halfspace defined by the inequality

$$\sum_{i=1}^k d_i c^i x \geq \sum_{i=1}^{k-1} d_i c^i \bar{x} + d_k \lceil c^k \bar{x} \rceil \quad (3.1)$$

and go to step 2.

Then $x^*$ is the lexicographically minimum vector in $B \cap \mathbb{Z}^n$ such that $\bar{x} \prec x^*$, and inequality (3.1) is the lex-cut (2.3) associated with $x^*$. As $S \cap \mathbb{Z}^n \subseteq B$ and $\bar{x}$ is the lexicographically optimal point in $S \cap B$, $\bar{x} \prec x^* \prec x'$ for every $x' \in S \cap \mathbb{Z}^n \setminus \{x^*\}$. As $S \cap \mathbb{Z}^n \subseteq B \cap \mathbb{Z}^n$, by Lemma 2.3, (3.1) is valid for $S \cap \mathbb{Z}^n$. As $c^k \bar{x} \notin \mathbb{Z}$, the inequality is violated by $\bar{x}$. This shows that (3.1) defines a cutting plane.

Again by Lemma 2.3, the number of possible lex-cuts is equal to the number of integer points in $B$, and by (2.1) this number is $\prod_{i=1}^n (u_i + 1)$. As different iterations of the algorithm use cuts (3.1) associated with lexicographically increasing vectors in $B \cap \mathbb{Z}^n$, this number also bounds the number of iterations of the algorithm.

**Remark 3.2.** Inequality (3.1) is of the form $\sum_{i=1}^k d_i c^i x \geq \delta$ for some $k \in \{1, \ldots, n\}$ and $\delta \in \mathbb{R}$. Therefore, Algorithm 1 produces cuts with only $n$ predetermined normal vectors. In particular, if $e^1, \ldots, e^n$ are nonnegative vectors, as e.g. in the standard basis, all cuts have nonnegative coefficients.

### 4 Lower bound on the number of iterations

In this section we prove that the worst-case behavior of Algorithm 1 is exponential (as expected).

**Proposition 4.1.** For every $n \in \mathbb{N}$, there is a convex subset $S$ of $[0,1]^n$ on which Algorithm 1 takes $2^n - 1$ iterations.

**Proof.** We give a constructive proof of the above proposition. Our lattice basis of $\mathbb{Z}^n$ is the standard basis $\{e^1, \ldots, e^n\}$. Let $\alpha, \beta$ be the vectors with all components equal to 1, except the last, which is 0 in $\alpha$ and 1 in $\beta$. Fix a sufficiently small number $\varepsilon > 0$ and, for every
Given a set \( S \), a Chvátal–Gomory inequality for \( S \) is a linear inequality of the form \( gx \geq \lceil \gamma \rceil \) for some \( g \in \mathbb{Z}^n \) and \( \gamma \in \mathbb{R} \) such that the inequality \( gx \geq \gamma \) is valid for \( S \). We call \( gx \geq \lceil \gamma \rceil \) a proper Chvátal–Gomory inequality if \( gx \geq \lceil \gamma \rceil \) is violated by at least one point in \( S \).

**Proposition 5.1.** Given \( S \in \mathcal{S} \), every proper Chvátal–Gomory inequality for \( S \) is an inequality of the type (3.1).

**Proof.** Let \( gx \geq \lceil \gamma \rceil \) be a proper Chvátal–Gomory inequality for \( S \). Without loss of generality, we assume that the entries of \( g \) are relatively prime integers. Let \( \bar{x} \) be the lexicographically minimum solution found at the first iteration of Algorithm 1 with respect to some lattice basis \( \{c^1, \ldots, c^\alpha\} \), with \( c^1 = g \). Since \( gx \geq \lceil \gamma \rceil \) is a proper Chvátal–Gomory inequality for \( S \), we have \( \gamma \leq g\bar{x} < \lceil \gamma \rceil \). In particular, \( g\bar{x} \notin \mathbb{Z} \). Then the corresponding cut of the type (3.1) is (equivalent to) \( gx \geq \lceil g\bar{x} \rceil = [\gamma] \). □

The converse of the above proposition is false; this will follow from a stronger result.

A linear inequality is a split cut for \( S \) if there exist \( \pi \in \mathbb{Z}^n \) and \( \pi_0 \in \mathbb{Z} \) such that the inequality is valid for both \( \{x \in S : \pi x \leq \pi_0\} \) and \( \{x \in S : \pi x \geq \pi_0 + 1\} \). It is known that every Chvátal–Gomory inequality is a split cut but not vice versa.

The next result shows that our family of cuts is not included in and does not include the family of split cuts. Combined with the previous proposition, this implies that our family of cuts strictly contains the Chvátal–Gomory inequalities.

**Proposition 5.2.** There exist a bounded polyhedron \( S \) and a split cut for \( S \) that cannot be obtained as (and is not implied by) an inequality of the type (3.1). Conversely, there exist a bounded polyhedron \( S \) and an inequality of the type (3.1) that is not a split cut for \( S \).
Proof. Let $S \subseteq \mathbb{R}^2$ be the triangle with vertices $(0,0)$, $(1,0)$ and $(1/2,-1)$. (See Figure 1 to follow the proof.) The inequality $x_2 \geq 0$ is a split cut for $S$, as it is valid for both sets $\{x \in S : x_1 \leq 0\}$ and $\{x \in S : x_1 \geq 1\}$. Note that after the application of the cut, the continuous relaxation becomes the segment with endpoints $(0,0)$ and $(1,0)$, which is the convex hull of the integer points in $S$.

Assume that the cut $x_2 \geq 0$ can be obtained via an iteration of Algorithm 1 for some lattice basis $\{c_1, c_2\}$ and some choice of the bounds $\ell_1, \ell_2, u_1, u_2 \in \mathbb{Z}$. In the following, we will write $c_1 = (c_1^1, c_1^2)$ and $c_2 = (c_2^1, c_2^2)$.

Recall that in Algorithm 1 a translation is applied such that $\ell_i = 0$ for every $i$. However, in this proof it is more convenient to work without applying the translation. It is easy to see that in this case the form of the lex-cut is still (3.1), but now $d_i = \prod_{j=i+1}^n (u_j - \ell_j + 1)$ for $i = 1, \ldots, n-1$. Thus, in our two-dimensional example, $d_1 = u_2 - \ell_2 + 1$ and $d_2 = 1$.

Since the point $(1/2,-1)$ is the only fractional vertex of $S$, we must have $\bar{x} = (1/2,-1)$, otherwise no cut is generated. Suppose $k = 1$, i.e., $c_1^1 \notin \mathbb{Z}$. Then the inequality generated by the algorithm is equivalent to $c_1^1 x \geq \lceil c_1^1 \bar{x} \rceil$. Since this inequality must be equivalent to $x_2 \geq 0$ and the entries of $c_1$ are relatively prime integers, we necessarily have $c_1^1 = (0,1)$. But then $c_1^1 \bar{x} = -1$, a contradiction to the assumption $c_1^1 \bar{x} \notin \mathbb{Z}$.

Suppose now $k = 2$, i.e., $c_1^1 \bar{x} \in \mathbb{Z}$ and $c_2^1 \bar{x} \notin \mathbb{Z}$. Then the inequality given by the algorithm is

$$d_1 (c_1^1 x - c_1^1 \bar{x}) + c_2^1 x - \lceil c_2^1 \bar{x} \rceil \geq 0. \tag{5.1}$$

We claim that $c_1^1 \neq 0$. If this is not the case, then $c_1^1 = 0$ and $c_2^1 \neq 0$ (as $\{c_1, c_2\}$ is a basis), and inequality (5.1) does not reduce to the desired cut $x_2 \geq 0$, as $d_1 c_1^1 + c_2^1 \neq c_2^1 \neq 0$. Thus $c_1^1 \neq 0$. This implies that either the point $(0,-1)$ or the point $(1,-1)$ satisfies the strict inequality $c_1^1 x > c_1^1 \bar{x}$. We assume that this holds for $\hat{x} := (0,-1)$ (the other case is similar).

Note that $c_1^1 \hat{x} + 1$, as $c_1^1 \bar{x} \in \mathbb{Z}$ and $c_1^1 \bar{x} \notin \mathbb{Z}$. Furthermore, the slope of the line defined by the equation $c_1^1 x + c_1^1 \bar{x}$ is positive.

If $c_2^1 \hat{x} \geq \ell_2$, then $\hat{x}$ satisfies inequality (5.1), as $c_1^1 \hat{x} - c_1^1 \bar{x} \geq 1$ and $c_2^1 \hat{x} - c_2^1 \bar{x} \geq \ell_2 - c_2^1 \bar{x} \geq -d_1$. Since the point $(1,0)$ also satisfies (5.1) (as it is an integer point in $S$), the middle point of $\hat{x}$ and $(1,0)$ satisfies (5.1). However, the middle point is $(1/2, -1/2)$, which is in $S$. This shows that in this case (5.1) is not equivalent to $x_2 \geq 0$.

Therefore we assume $c_2^1 \hat{x} < \ell_2$. Since $c_2^1 \hat{x} \geq \ell_2$, the line defined by the equation $c_2^1 x = \ell_2$ intersects the line segments $[\hat{x}, \bar{x}]$ in a point distinct from $\bar{x}$. Then, because $(0,0)$ satisfies the inequality $c_2^1 x \geq \ell_2$ (as it is in $S$), the slope of the line defined by the equation $c_2^1 x = \ell_2$ is negative. Furthermore, since $c_2^1, \bar{x} \in \mathbb{Z}^2$, we have $c_2^1 \bar{x} \leq \lceil \ell_2 \rceil$, and thus the line defined by the equation $c_2^1 x = \lceil \ell_2 \rceil$ intersects $[\hat{x}, \bar{x}]$ in some point $x^*$.

Now consider the system $c_1^1 x = c_1^1 \bar{x}$, $c_2^1 x = \lceil \ell_2 \rceil$. Since the constraint matrix is unimodular (as $\{c_1, c_2\}$ is a lattice basis of $\mathbb{Z}^2$) and the right-hand sides are integer, the unique solution to this system is an integer point. However, the first equation defines a line with positive slope containing $\bar{x}$ and the second equation defines a line with negative slope containing $x^*$. From this we see that the intersection of the two lines is a point satisfying $0 < x_1 \leq 1/2$ and therefore cannot be an integer point, a contradiction. This concludes the proof that there is a split cut that cannot be obtained via an iteration of Algorithm 1.

For the converse, let $S \subseteq \mathbb{R}^2$ be the quadrilateral with vertices $(0,9/2)$, $(1,5)$, $(1/12,0)$ and $(1,0)$. If we take $c_1^1, c_2^1$ to be the vectors in the standard basis of $\mathbb{R}^2$, and $\ell_1 = \ell_2 = 0$, $u_1 = 1$ and $u_2 = 5$, then Algorithm 1 yields the cut $6x_1 + x_2 \geq 5$. Let $Q$ be the topological closure of the subset of $S$ cut off by this inequality, i.e., $Q := \{x \in S : 6x_1 + x_2 \leq 5\}$. 7
Figure 1: Illustration of the first part of the proof of Proposition 5.2. The inequality \( x_2 \geq 0 \) is a split cut for the shadowed triangle, but is not of the type (3.1).

\( Q \) is the quadrilateral with vertices \((0,9/2), (1/12,0), (5/6,0), (1/13,59/13)\). Note that \( Q \) contains a horizontal and a vertical segment each of length at least \( 3/4 \) (there are longer vertical segments, but this will suffice; see figure 2).

If the inequality \( 6x_1 + x_2 \geq 5 \) is a split cut for \( S \), then there exist \( \pi \in \mathbb{Z}^2 \) and \( \pi_0 \in \mathbb{Z} \) such that every point in \( \{x \in S : \pi x \leq \pi_0\} \cup \{x \in S : \pi x \geq \pi_0 + 1\} \) satisfies \( 6x_1 + x_2 \geq 5 \). In particular, \( Q \subseteq \{x \in \mathbb{R}^n : \pi_0 \leq \pi x \leq \pi_0 + 1\} \) must contain two orthogonal segments of length \( 3/4 \). This is possible only if the Euclidean distance between the lines \( \{x \in \mathbb{R}^2 : \pi x = \pi_0\} \) and \( \{x \in \mathbb{R}^2 : \pi x = \pi_0 + 1\} \) is at least \( \frac{3}{4\sqrt{2}} \). Therefore \( \|\pi\|^2 \leq \left(\frac{4\sqrt{2}}{3}\right)^2 = \frac{32}{9} < 4 \). Since \( \pi \) is an integer vector, we deduce that \( \pi_1, \pi_2 \in \{0,1,-1\} \). It can be verified that if \( |\pi_1|=|\pi_2|=1 \) then \( Q \) is not contained in the strip. Therefore one entry of \( \pi \) is 0 and the other is 1 or \(-1\). It can be checked that the only strip of this type containing \( Q \) is \( \{x \in \mathbb{R}^2 : 0 \leq x_1 \leq 1\} \). However, the inequality \( 6x_1 + x_2 \geq 5 \) is not valid for all the points in \( \{x \in S : x_1 \leq 0\} \cup \{x \in S : x_1 \geq 1\} \), as the point \((0,9/2)\) is in this set but violates the inequality.

6  Strengthening the algorithm?

An obvious variant of Algorithm 1 is the following: instead of being computed only once at the beginning of the procedure, the box \( B \) can be updated at every iteration or whenever it seems convenient. Though this requires some additional computational effort at every iteration, it seems reasonable to hope that this approach provides better cuts. In the following, after formalizing this idea, we illustrate that this intuition is not correct.

Given a set \( S \), consider two boxes \( B \) and \( B' \) such that \( S \subseteq B \subseteq B' \), and denote by \( C \) (resp., \( C' \)) the cut (3.1) yielded by Algorithm 1 when the box \( B \) (resp., \( B' \)) is chosen. Since \( C \) has smaller coefficients than \( C' \), one may expect that \( C \) dominates \( C' \), i.e., every point in \( S \) that satisfies \( C' \) also satisfies \( C \). Alternatively, one may at least expect the weaker property that \( C \) yields a better relaxation than \( C' \) with respect to the objective function, i.e., the next solution \( \bar{x} \) obtained after the introduction of the cut \( C' \) has larger objective value than the solution obtained after the introduction of the cut \( C \). However, we now show that this is not always the case. This means that, although the upper bound on the number of iterations given in the proof of Proposition 3.1 depends on the size of the box, it is not true that tighter
Figure 2: Illustration of the second part of the proof of Proposition 5.2. The inequality $6x_1 + x_2 \geq 5$ is a cut of the type (3.1) for the shadowed quadrilateral, but is not a split cut.

boxes give better cuts.

**Proposition 6.1.** There are instances in which choosing weaker bounds $\ell_i, u_i$ in Algorithm 1 yields a stronger cut with respect to the objective function.

**Proof.** Let $S \subseteq \mathbb{R}^3$ be the triangle with vertices $(0, 1, 1/3), (1/3, 0, 4)$ and $(1, 0, 0)$. The objective function is $x_1$ and we choose $c^1, \ldots, c^n$ to be the vectors in the standard basis of $\mathbb{R}^n$. If we take $\ell_1 = \ell_2 = \ell_3 = 0$, $u_1 = u_2 = 1$ and $u_3 = 4$, the corresponding inequality (3.1) is $10x_1 + 5x_2 + x_3 \geq 6$. After the introduction of this cut, the new optimal value is $1/9$.

On the other hand, if we compute (3.1) considering the weaker bound $u_3 = 5$, the cut is $12x_1 + 6x_2 + x_3 \geq 7$ and the new optimal value is $2/17$, which is larger than $1/9$.

It is easy to construct examples (already in dimension 2) for which choosing stronger bounds $\ell_i, u_i$ yields a stronger cut with respect to the objective function.

We also remark that the lower bound on the number of iterations shown in Proposition 4.1 also holds for this variant of the algorithm: the proof is the same.

Finally, it is easy to see that different choices of the lattice basis (or different choices of the ordering of the elements of the same lattice basis) may result in a different number of iterations of the algorithm. For instance, for the example in Proposition 4.1 Algorithm 1 would terminate in a single iteration if the lattice basis vectors were sorted differently (more precisely, if $c^1$ were chosen to be the last vector of the standard basis of $\mathbb{R}^n$). It is not clear how the lattice basis and the bounds should be chosen in order to minimize the number of iterations.

**References**


