A new cutting plane method for lexicographic multi-objective integer linear programming

Article *in* Communications in Nonlinear Science and Numerical Simulation · November 2023

DOI: 10.1016/j.cnsns.2023.107674

5 authors, including:

Marco Cococcioni
Università di Pisa

Alessandro Cudazzo
Università di Pisa

Lorenzo Fiaschi
Università di Pisa

Massimo Pappalardo
Università di Pisa

A New Cutting Plane Method for Lexicographic Multi-Objective Integer Linear Programming

Marco Cococcioni * Alessandro Cudazzo * Lorenzo Fiaschi *
Massimo Pappalardo † Yaroslav D. Sergeyev ‡

Abstract

This work presents a new cutting plane method for lexicographic multi-objective integer linear programming (LMOILP). The method uses Grossone Methodology to reformulate a LMOILP problem into one having a single non-Archimedean scalar objective function, as done in [1] (but in that case in the absence of the integer constraints). The problem, without the integer constraints, is solved using the GrossSimplex algorithm presented in [1] to find a candidate optimal solution. Here a novel cutting plane is introduced, named Gross-based Objective Function Cutting Plane whenever the optimal value of the Gross-scalar objective function is Gross-fractional. When it happens to be not Gross-fractional, cutting planes are generated using the Fractional Cutting Plane, derived from fractional components of the optimal solution. Moreover, by combining them, we proposed an algorithm that we called Gross-based Cutting Plane (GCP) method. It has been proved that it finds the optimal solution of a LMOILP problem and terminates after a finite number of iterations. To speed-up the GCP, at each iteration subsequent the first one, we re-use the optimal basis of the last continuous relaxation, by running the GrossDualSimplex algorithm. This is the well-known *warm-start* technique, which, however, needs specific attention due to the need to solve a linear problem having a non-Archimedean right-hand side. In the experimental part, we show the efficacy of the proposed approach.

Keywords: Multi-Objective Optimization; Lexicographic Optimization; Integer Linear Programming; Numerical Infinitesimals; Grossone Methodology; Cutting Plane.

1. Introduction

The last decades have witnessed an explosion of literature about multi-objective optimisation, both using deterministic and stochastic approaches. On the contrary, lexicographic multi-objective optimisation, a subfield of multi-objective optimisation, has attracted far less attention by researchers and practitioners, with only a few exceptions [1–5]. We believe that lexicographic optimisation will attract more interests by researchers, engineers and practitioners in proportion to the availability of novel approaches to address them. In [1], the *linear* lexicographic multi-objective optimisation case has been solved by introducing the GrossSimplex algorithm, an algorithm able to deal with infinitesimal/infinite quantities, modelled using Grossone Methodology [6, 7]. The main idea of that work was to transform the multi-objective problem into a single-objective one, where the less important objectives are summed up with an infinitesimal weight, and the order of the infinitesimal weight increases with the objectives. The scalarisation is obtained by using the numeral ① called *Grossone* (see [6,7]). ① is a basic element allowing the construction of a powerful numeral system, which allows to express not only finite but also infinite and infinitesimal quantities.

Many paper studying the consistency of the Grossone Methodology and its connection to the historical panorama of ideas dealing with infinities and infinitesimals have been published so far [8–13]. Specifically,

*University of Pisa, Dipartimento di Ingegneria dell'Informazione, Largo Lucio Lazzarino, 1 – 56122 Pisa, Italy, alessandro.cudazzo@phd.unipi.it, marco.cococcioni@unipi.it, lorenzo.fiaschi@phd.unipi.it

†University of Pisa, Dipartimento di Informatica, Largo Bruno Pontecorvo, 3 – 56127 Pisa, Italy, massimo.pappalardo@unipi.it

‡Corresponding author, University of Calabria, Rende (CS), Italy and Lobachevsky State University of Nizhni Novgorod, Russia, yaro@dimes.unical.it

in [12] it is emphasised that it is not related to non-standard analysis. Grossone has given rise both to a new supercomputer patented in several countries (see [14]), called *Infinity Computer*, and more than sixty journal papers make use of Grossone on fields such as numerical differentiation and ordinary differential equations [15–17], fractals [18, 19], hyperbolic geometry and percolation [20], the first Hilbert problem and Turing machines [21], infinite decision making processes [22], simulation of hybrid systems [23], game theory and probability [24–26]. Moreover, successful applications of this methodology in teaching mathematics should be mentioned (see [27–29]). Concerning optimisation, Grossone has been successfully applied in local, global and multiple criteria optimisation [1, 30–38].

In the present study we consider the Lexicographic Multi-Objective Integer Linear Programming (LMOILP) problem. Differently from [39], where we have solved the problem by extending the Branch-and-Bound method, in this work we aim at devising a solution based on a generalisation of the classical cutting plane method. Then, to efficiently manage the addition of new constraints generated by the cutting planes, we employed the duality theory. In particular, we used the GrossDualSimplex algorithm described in [40]. A remarkable feature of our approach is that it does not require the specification of the scalarising weights, a well-known limitation in classical finite-weight scalarization of lexicographic problems.

The remaining text is structured as follows. In Section 2, the LMOILP problem is introduced and then reformulated by using the Grossone Methodology. In Sections 3 and 4, we present the new Gross-based Objective Function Cutting Plane (GOF-CP) and the Gross-based Cutting Plane (GCP) method, respectively. In Section 5, we show how the addition of new constraints can be handled efficiently through the use of the GrossDualSimplex introduced in [40]. Section 6 presents a few LMOILP test problems and their solutions using the proposed algorithm, while Section 7 is devoted to the conclusions.

2. Lexicographic Multi-Objective Integer Linear Programming with Grossone

The problem tackled in this work is the LMOILP one, which can be formalised as follows:

$$\begin{array}{ll} \text{lexmin} & \mathbf{c}^{1\,T}\mathbf{x},\ \mathbf{c}^{2\,T}\mathbf{x}, ..., \mathbf{c}^{r\,T}\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geqslant 0,\ \mathbf{x} \text{ integer}, \end{array} \qquad P$$

where $\mathbf{c}^i \in \mathbb{Z}^n,\ i = 1, ..., r$, and $\mathbf{x} \in \mathbb{R}^n$ are column vectors, $\mathbf{A} \in \mathbb{Z}^{m\times n}$ is a full-rank matrix, $\mathbf{b} \in \mathbb{Z}^m$ is again a column vector. Superscript T indicates the transpose and the symbol lexmin in P denotes *Lexicographic Minimum* and means that the first objective is infinitely more important than the second, which is, on its turn, infinitely more important than the third one, and so on.

As done in [39, 41], we can reformulate problem P as a single-objective one, by using the Grossone Methodology to obtain a scalarization. In the following subsections, we provide a brief introduction on Grossone methodology, focusing on the key points used in this paper, and the LMOILP reformulation.

2.1 Grossone Methodology

From the foundational point of view, Grossone has been introduced as an infinite unit of measure equal to the number of elements of the set $\mathbb{N}$ of natural numbers. A general way to express infinities and infinitesimals is also provided in [6, 7] by using records similar to traditional positional number systems, but with the radix ①. A number $\tilde{c}$ (called Gross-scalar) in this new numeral system can be constructed by subdividing it into groups of corresponding powers of ① and represented as

$$\tilde{c} = c_{p_m}①^{p_m} + ... + c_{p_1}①^{p_1} + c_{p_0}①^{p_0} + c_{p_{-1}}①^{p_{-1}} + ... + c_{p_{-k}}①^{p_{-k}},$$

where $m, k \in \mathbb{N}$, the exponents p_i are called Gross-powers and $c_{p_i} \in \mathbb{R}$ are called Gross-digits, $i = m, ..., 1, 0, -1, ..., -k$ (in addition, we assume $p_i > p_{i-1}$ and $p_0 = 0$). In general, the Gross-powers p_i can be Gross-scalars on their own, as $\tilde{c}$, but in this work we will assume they are integer values ($p_i \in \mathbb{Z}$).

In this numeral system, finite numbers are represented by numerals with the highest Gross-power equal to zero. Infinitesimals are represented by numerals having negative Gross-powers. We notice that all infinitesimals are not equal to zero, e.g., $①^{-1} > 0$. A number is infinite if it has at least one positive finite or infinite Gross-power.

A Gross-scalar is said *purely finite* iff the coefficient associated with the zeroth power of Grossone is the only one to be different from zero. For instance, the number 3.4 is purely finite and $3.4 - 3.2①^{-5}$ is finite but not purely finite. A Gross-vector can be easily defined as a vector of Gross-scalars.

For the purposes of this paper, we need to define the fractional part of a real number x and when, in our context, a Gross-scalar is Gross-integer or Gross-fractional.

let us now denote $\{x\} \in [0, 1)$ as the fractional part of a real number x, defined as follows:

$$\{x\} = x - \lfloor x \rfloor$$

where, $\lfloor x \rfloor$ denotes the integer part of a real number x:

$$\lfloor x \rfloor = \max\{k \in \mathbb{Z} : k \leq x\}.$$

Definition 1. *A Gross-scalar made of integer Gross-powers is considered Gross-integer iff all its Gross-digits are integer. Otherwise, it is Gross-fractional.*

As an example, $9①^4 + 4.33①^{-3} + 3①^{-5}$ is Gross-fractional because the Gross-digit 4.33 is not integer. On the other hand, $5①^3 + 2①^{-4} - 7①^{-9}$ has integer Gross-digits and hence it is Gross-integer.

2.2 LMOILP reformulation

The reformulation of an LMOILP problem, using Grossone, is as follows:

$$\begin{array}{ll} \min & \tilde{\mathbf{c}}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geqslant 0,\ \mathbf{x} \text{ integer}, \end{array} \qquad \tilde{P}$$

where $\tilde{\mathbf{c}}$ is a column Gross-vector having n Gross-scalar components built using purely finite vectors $\mathbf{c}^i$

$$\tilde{\mathbf{c}} = \sum_{i=1}^{r} \mathbf{c}^i ①^{-i+1} \tag{1}$$

and $\tilde{\mathbf{c}}^{\mathbf{T}}\mathbf{x}$ is the Gross-scalar obtained by multiplying the Gross-vector $\tilde{\mathbf{c}}$ by the purely finite vector $\mathbf{x}$

$$\tilde{\mathbf{c}}^{\mathbf{T}}\mathbf{x} = (\mathbf{c}^{1T}\mathbf{x})①^0 + (\mathbf{c}^{2T}\mathbf{x})①^{-1} + ... + (\mathbf{c}^{rT}\mathbf{x})①^{-r+1}, \tag{2}$$

where (2) can be equivalently written in the extended form as:

$$\tilde{\mathbf{c}}^{\mathbf{T}}\mathbf{x} = (c_1^1 x_1 + ... + c_n^1 x_n)①^0 + (c_1^2 x_1 + ... + c_n^2 x_n)①^{-1} + ... + (c_1^r x_1 + ... + c_n^r x_n)①^{-r+1}.$$

Notice that $\mathbf{c}^{qT}\mathbf{x} = c_1^q x_1 + ... + c_n^q x_n$ is the q-th Gross-digit, $q = 1, ..., r$.

In [39] we have shown that problem $\tilde{P}$ is equivalent to problem P and this new formulation $\tilde{P}$ is attractive because the set of multiple objective functions is mapped into a single (Gross-) scalar function to be optimised. Moreover, the continuous relaxation version is the following Gross-LP problem [1]:

$$\begin{array}{ll} \min & \tilde{\mathbf{c}}^T\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geqslant 0, \end{array} \qquad \tilde{R}$$

which can be effectively solved using *a single run* of the GrossSimplex algorithm [1].

The feasible region of problem $\tilde{R}$ will be denoted by $\mathcal{S}$:

$$\mathcal{S} := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geqslant 0\},$$

while the feasible region of $\tilde{P}$ will be denoted by Ω:

$$\Omega = \mathcal{S} \cap \mathbb{Z}^n.$$

Hereinafter, we assume that $\mathcal{S}$ is bounded and non-empty.

3. Cutting plane with Grossone

In this section we will study and extend cutting planes for problem $\tilde{P}$. A cutting plane is any inequality, that, when added to $\mathcal{S}$, cuts out a given point $\bar{\mathbf{x}}$ in $\mathcal{S}$ but does not exclude any point in Ω. Its formal definition is:

Definition 2. *Given a point $\bar{\mathbf{x}} \in \mathcal{S}$. An inequality $\mathbf{a}^T\mathbf{x} \leqslant a_0$ is a cutting plane for $\mathcal{S}$ if these two conditions hold:*

i) $\mathbf{a}^T\mathbf{x} \leqslant a_0 \quad \forall \mathbf{x} \in \Omega$,

ii) $\mathbf{a}^T\bar{\mathbf{x}} > a_0$.

To generate a new cutting plane, the idea is to solve problem $\tilde{R}$ by using the GrossSimplex. When it ends, it provides an optimal basis $\mathtt{B}$ and the associated optimal solution $\bar{\mathbf{x}}$. If the Gross-scalar $\tilde{\mathbf{c}}_{\mathtt{B}}^T\bar{\mathbf{x}}_{\mathtt{B}}$ is Gross-fractional, then we can generate a series of cutting planes, one for each fractional digit present in $\tilde{\mathbf{c}}_{\mathtt{B}}^T\bar{\mathbf{x}}_{\mathtt{B}}$. We have called this method of generating new cutting planes Gross-based Objective Function Cutting Plane (GOF-CP), and it is described by the following theorem.

Theorem 1 (GOF-CP). *Let $\mathtt{B}$ be an optimal basis for $\tilde{R}$, $\bar{\mathbf{x}}$ the associated optimal solution and $\mathtt{N} := \{1,\dots,n\} \setminus \mathtt{B}$. If the Gross-scalar $\tilde{\mathbf{c}}_{\mathtt{B}}^T\bar{\mathbf{x}}_{\mathtt{B}}$ is Gross-fractional, then for each q such that the Gross-digit $\mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}}$ is not integer, the inequality*

$$\sum_{j\in\mathtt{N}} \{(\mathbf{c}_{\mathtt{N}}^{qT} - \mathbf{c}_{\mathtt{B}}^{qT}\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_j\} x_j \geqslant \{-\mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}}\} \tag{3}$$

is a cutting plane for $\mathcal{S}$.

Proof. Being $\mathtt{B}$ an optimal basis for $\tilde{R}$, an arbitrary $\mathbf{x} \in \Omega$ can always be expressed as $\mathbf{A}\mathbf{x} = \mathbf{A}_{\mathtt{B}}\mathbf{x}_{\mathtt{B}} + \mathbf{A}_{\mathtt{N}}\mathbf{x}_{\mathtt{B}} = \mathbf{b}$, and, for the same reason, we know that $\bar{\mathbf{x}}_{\mathtt{B}} = \mathbf{A}_{\mathtt{B}}^{-1}\mathbf{b}$. Combining them we obtain $\mathbf{x}_{\mathtt{B}} = \bar{\mathbf{x}}_{\mathtt{B}} + \mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}}\mathbf{x}_{\mathtt{N}}$.

Let us now introduce a fictitious variable for each objective function, having a non-positive index (to distinguish it from the input variables x_1,..., x_n). Hence, the fictitious variables are $x_{-r+1} = \mathbf{c}^{1T}\mathbf{x}$,..., $x_{-1} = \mathbf{c}^{(r-1)T}\mathbf{x}$, $x_0 = \mathbf{c}^{rT}\mathbf{x}$, where, in particular, the variable having the most negative index x_{-r+1} is the one associated to the most important objective function.
Now, we know by hypothesis that there exists q such that $\mathbf{c}^{qT}\mathbf{x}$ is not integer. We have that:

$$\begin{aligned} x_{-r+q} &= \mathbf{c}^{qT}\mathbf{x} \qquad\qquad\qquad \text{(by definition of fictitious variable)} \\ &= \mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}} + (\mathbf{c}_{\mathtt{N}}^{qT} - \mathbf{c}_{\mathtt{B}}^{qT}\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})\mathbf{x}_{\mathtt{N}} \\ &= \lfloor \mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}} \rfloor + \{\mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}}\} + \sum_{j\in\mathtt{N}} \Big(\lfloor (\mathbf{c}_{\mathtt{N}}^{qT} - \mathbf{c}_{\mathtt{B}}^{qT}\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_j \rfloor + \{(\mathbf{c}_{\mathtt{N}}^{qT} - \mathbf{c}_{\mathtt{B}}^{qT}\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_j\} \Big) x_j \,. \end{aligned}$$

Then it follows that:

$$\sum_{j\in\mathtt{N}} \{(\mathbf{c}_{\mathtt{N}}^{qT} - \mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_j\} x_j + \{\mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}}\} = x_{-r+q} - \lfloor \mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}} \rfloor - \sum_{j\in\mathtt{N}} \lfloor (\mathbf{c}_{\mathtt{N}}^{qT} - \mathbf{c}_{\mathtt{B}}^{qT}\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_j \rfloor x_j, \tag{4}$$

where the right-hand side is clearly integer.
Moreover:

$$\sum_{j\in\mathtt{N}} \{(\mathbf{c}_{\mathtt{N}}^{qT} - \mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_j\} x_j + \{\mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}}\} \geqslant \{\mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}}\} > 0. \tag{5}$$

This means that, in the feasible points the left-hand side of (4) is strictly positive. On the other hand, its right-hand side is integer, as previously observed. Therefore:

$$\sum_{j\in\mathtt{N}} \{(\mathbf{c}_{\mathtt{N}}^{qT} - \mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_j\} x_j + \{\mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}}\} \geqslant 1.$$

Or, equivalently:

$$\sum_{j\in\mathtt{N}} \{(\mathbf{c}_{\mathtt{N}}^{qT} - \mathbf{c}_{\mathtt{B}}^{qT}\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_j\} x_j \geqslant 1 - \{\mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}}\}. \tag{6}$$

Since the $1 - \{\mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}}\} = \{-\mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}}\}$, (6) can be also alternatively expressed as:

$$\sum_{j\in\mathtt{N}}\{(\mathbf{c}_{\mathtt{N}}^{qT} - \mathbf{c}_{\mathtt{B}}^{qT}\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_j\}x_j \geqslant \{-\mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}}\}. \tag{7}$$

We remark that the obtained equation, if added as a constraint to problem $\tilde{P}$, will not cut out any $\mathbf{x} \in \Omega$, because we have derived it for any $\mathbf{x} \in \Omega$.

Concerning the second requirement of Definition 2, we observe that (3) is not satisfied by $\bar{\mathbf{x}}$. Indeed:

$$0 = \sum_{j\in\mathtt{N}}\{(\mathbf{c}_{\mathtt{N}}^{qT} - \mathbf{c}_{\mathtt{B}}^{qT}\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_j\}x_j < \{-\mathbf{c}_{\mathtt{B}}^{qT}\bar{\mathbf{x}}_{\mathtt{B}}\}.$$

As a consequence, (3) is a cutting plane. ■

Thm. 1 provides us a set of cutting planes arising from each objective function that is fractional.

When $\tilde{\mathbf{c}}_{\mathtt{B}}^{T}\bar{\mathbf{x}}_{\mathtt{B}}$ is not Gross-fractional (and therefore Thm. 1 cannot be used), we can still build the classical Fractional Cutting Plane, starting from a fractional component in the real vector $\mathbf{x}$ (remember that *is not* a vector of Gross-scalars):

Theorem 2 (Fractional Cutting Plane)**.** *Let* $\mathtt{B}$ *be an optimal basis for* $\tilde{R}$, $\bar{\mathbf{x}}$ *the associated optimal solution and* $\mathtt{N} := \{1, \dots, n\} \setminus \mathtt{B}$. *If there exists a component* s *in vector* $\bar{\mathbf{x}}_{\mathtt{B}}$ *such that* $(\bar{\mathbf{x}}_{\mathtt{B}})_s$ *is not integer, then the following inequality*

$$\sum_{j\in\mathtt{N}}\{(\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj}\}x_j \geqslant \{(\bar{\mathbf{x}}_{\mathtt{B}})_s\} \tag{8}$$

is a cutting plane for $\mathcal{S}$.

For the sake of completeness, the well-known proof of Thm. 2 is reported in Appendix. The same appendix also reports the integer formulation of the Fractional Cutting Plane in Thm. 4: this formulation is interesting because it is known to be more numerically stable.
In the next section, we will show how to build an algorithm, based both on GOF-CP and the Fractional Cutting Plane, able to find an optimal solution for problem $\tilde{P}$ in a finite number of iterations.

4. Gross-based Cutting Plane (GCP) method

In this section, we aim to extend the classical cutting plane method for $\tilde{P}$. To do so, we are going to prove that, by adding in a suitable order the cutting planes described in the previous section, it is possible to converge to the optimal integer solution in a finite number of steps.

At the beginning, the input problem is set as the current problem ($\tilde{P}_c := \tilde{P}$). Then, at each iteration, the continuous relaxation $\tilde{R}_c$ of $\tilde{P}_c$ is solved using the GrossSimplex, which returns an optimal basis $\mathtt{B}$ and the associated optimal solution $\bar{\mathbf{x}}$. If $\bar{\mathbf{x}}$ is integer, then we are done and the algorithm terminates. Otherwise, a GOF-CP is added to $\tilde{P}_c$, if $\tilde{\mathbf{c}}^T\bar{\mathbf{x}}$ is Gross-fractional. Alternatively, the Fractional Cutting Plane is added to $\tilde{P}_c$. Then, the relaxed version $\tilde{R}_c$ of the new current problem $\tilde{P}_c$ is solved again, and the procedure repeats. This algorithm, called GCP method, is shown in Alg. 1 and its proof of convergence in finite number of iterations is provided below.

Theorem 3. *The Gross-based Cutting Plane method terminates in a finite number of iterations.*

Proof. At iteration t, let $\bar{\mathbf{x}}^t = (\bar{x}_1^t, \dots, \bar{x}_n^t)$ be the optimal solution of the relaxed problem associated to $\tilde{P}$. Also, recall that we have defined the fictitious variable $\bar{x}_{-r+q}^t := \mathbf{c}^{qT}\bar{\mathbf{x}}^t$, $q \in 1, \dots, r$. Observe that, by construction, the sequence of vectors $(\bar{x}_{-r+1}^t, ..., \bar{x}_{-1}^t, \bar{x}_0^t, \bar{x}_1^t, \dots, \bar{x}_n^t)$ is lexicographically nonincreasing as t increases. We need to show that, for $k = -r+1, \dots, n$, after a finite number of iterations the value of $\bar{x}_k$ is integer and does not change in subsequent iterations. Suppose not, and let k be the smallest index contradicting the claim. In particular, for $i = -r+1, ..., k-1$, after a finite number T of iterations, the value of $\bar{x}_i^t$ are integer and do not change in subsequent iterations, i.e., $\bar{x}_i^t = \bar{x}_i \in \mathbb{Z}$ for all $t \geqslant T$, $i = -r+1, \dots, k-1$.

By construction, the sequence of real numbers $\{\bar{x}_k^t\}_{t>T}$ is nonincreasing and bounded from below because the feasible region of problem $(\tilde{P})$ is bounded. This implies that, for any $\epsilon > 0$, there exists T such that when $\bar{t} > T$, $|\bar{x}_k^{\bar{t}} - l| < \epsilon$, where $l \in \mathbb{R}$. Therefore, we have $\bar{x}_k^{\bar{t}} = \lfloor l \rfloor + f$, where $l - \lfloor l \rfloor \leqslant f < 1$.

Observe that $f > 0$, otherwise l is integer and $\bar{x}^t_k = l$ for all $t \geqslant \bar{t}$ contradicting the choice of k. Since $\bar{x}^{\bar{t}}_i = \bar{x}_i$ is integer for $i = -r+1, \dots, k-1$, while $\bar{x}^t_k$ is not integer by hypothesis, it follows that a cutting plane can be generated. We have to consider two scenarios:

- If $k \leq 0$, a cutting plane can be added according to Thm. 1 by selecting the first fractional Gross-digit $q := r + k$ in $\tilde{\mathbf{c}}\bar{\mathbf{x}}^t$:

$$\sum_{j \in \mathtt{N}} \{(\mathbf{c}^{qT}_{\mathtt{N}} - \mathbf{c}^{qT}_{\mathtt{B}} \mathbf{A}^{-1}_{\mathtt{B}} \mathbf{A}_{\mathtt{N}})_j\} x_j \geqslant \{-\mathbf{c}^{qT}_{\mathtt{B}} \bar{\mathbf{x}}_{\mathtt{B}}\}.$$

  Let us now observe how the GOF-CP, defined in Eq. (3), can be alternatively expressed in an integer form. From (4) and (5) it follows that the right-side of (4) must be greater than or equal to 1 for any $\mathbf{x} \in \Omega$,

$$x_{-r+q} - \lfloor \mathbf{c}^{qT}_{\mathtt{B}} \bar{\mathbf{x}}_{\mathtt{B}} \rfloor - \sum_{j \in \mathtt{N}} \lfloor (\mathbf{c}^{qT}_{\mathtt{N}} - \mathbf{c}^{qT}_{\mathtt{B}} \mathbf{A}^{-1}_{\mathtt{B}} \mathbf{A}_{\mathtt{N}})_j \rfloor x_j \geqslant 1, \tag{9}$$

$$\sum_{j \in \mathtt{N}} \lfloor (\mathbf{c}^{T}_{\mathtt{N}} - \mathbf{c}^{qT}_{\mathtt{B}} \mathbf{A}^{-1}_{\mathtt{B}} \mathbf{A}_{\mathtt{N}})_j \rfloor x_j - x_{-r+q} \leqslant -1 - \lfloor \mathbf{c}^{qT}_{\mathtt{B}} \bar{\mathbf{x}}_{\mathtt{B}} \rfloor. \tag{10}$$

  Since it is well-known that $-1 - \lfloor \mathbf{c}^{qT}_{\mathtt{B}} \bar{\mathbf{x}}_{\mathtt{B}} \rfloor = \lfloor -\mathbf{c}^{qT}_{\mathtt{B}} \bar{\mathbf{x}}_{\mathtt{B}} \rfloor$, the latter can be also expressed as:

$$\sum_{j \in \mathtt{N}} \lfloor (\mathbf{c}^{qT}_{\mathtt{N}} - \mathbf{c}^{qT}_{\mathtt{B}} \mathbf{A}^{-1}_{\mathtt{B}} \mathbf{A}_{\mathtt{N}})_j \rfloor x_j - x_{-r+q} \leqslant \lfloor -\mathbf{c}^{qT}_{\mathtt{B}} \bar{\mathbf{x}}_{\mathtt{B}} \rfloor. \tag{11}$$

  Since $\sum_{j \in \mathtt{N}} \lfloor (\mathbf{c}^{qT}_{\mathtt{N}} - \mathbf{c}^{qT}_{\mathtt{B}} \mathbf{A}^{-1}_{\mathtt{B}} \mathbf{A}_{\mathtt{N}})_j \rfloor x_j \neq 0$ for any $\mathbf{x} \in \Omega$, inequality (11) is valid for an arbitrary point $\mathbf{x} \in \Omega$ and therefore condition i) of Def. 2 holds. Moreover (11) is not satisfied by $\bar{\mathbf{x}}$ because $\sum_{j \in \mathtt{N}} \lfloor (\mathbf{c}^{T}_{\mathtt{N}} - \mathbf{c}^{T}_{\mathtt{B}} \mathbf{A}^{-1}_{\mathtt{B}} \mathbf{A}_{\mathtt{N}})_j \rfloor \bar{x}_j = 0$ and $-x_{-r+q} > \lfloor -\mathbf{c}^{T}_{\mathtt{B}} \bar{\mathbf{x}}_{\mathtt{B}} \rfloor$. Therefore condition ii) of Def. 2 holds and (11) is a cutting plane.
  Since $\mathtt{B}$ is optimal, all the reduced costs are non-negative (i.e., $\mathbf{c}^{qT}_{\mathtt{N}} - \mathbf{c}^{qT}_{\mathtt{B}} \mathbf{A}^{-1}_{\mathtt{B}} \mathbf{A}_{\mathtt{N}})_j \geqslant 0$, $\forall j \in \mathtt{N}$), from (11) it follows that:

$$-x_k \leqslant \lfloor -l \rfloor.$$

  Then, for every $t \geqslant \bar{t} + 1$, $-x^t_k \leqslant \lfloor -l \rfloor \leqslant -l$, thus l is integer and $x^t_k = l$ for all $t > \bar{t}$.

- If $k \geqslant 1$, the following cutting plane will be added (see Thm. 2):

$$\sum_{j \in \mathtt{N}} \{(\mathbf{A}^{-1}_{\mathtt{B}} \mathbf{A}_{\mathtt{N}})_{kj}\} x_j \geqslant \{(\bar{\mathbf{x}}_{\mathtt{B}})_k\}.$$

  To proceed with the proof, it is helpful to use Thm. 4 (see Appendix), which allows to rewrite the above cutting plane in its equivalent integer form:

$$\sum_{j \in \mathtt{N}} \lfloor (\mathbf{A}^{-1}_{\mathtt{B}} \mathbf{A}_{\mathtt{N}})_{kj} \rfloor x_j + x_k \leqslant \lfloor (\bar{\mathbf{x}}_{\mathtt{B}})_k \rfloor = \lfloor l \rfloor.$$

  Using the well-known arguments provided in Thm. 5.19 of [42] on page 215-216, we reach the contradiction that $l = \lfloor l \rfloor$. This concludes the proof.

■

Algorithm 1 The GCP method

procedure GCP($\tilde{P}$)
Input: An LMOILP problem $\tilde{P}$
Output: An optimal basis B and the associated optimal solution $\bar{\mathbf{x}}$
  Set $\tilde{P}_c := \tilde{P}$ ($\tilde{P}_c$ is the current problem) and $t := 0$
  while True **do**
    $\tilde{R}_c \leftarrow$ `integerRelaxation`($\tilde{P}_c$)
    [B, $\bar{\mathbf{x}}$] $\leftarrow$ `GrossLPSolver`($\tilde{R}_c$)
    if $\bar{\mathbf{x}}$ is integer **then** Stop (B is an optimal basis for the given problem $\tilde{P}$)
    else
      if $\tilde{\mathbf{c}}^T\bar{\mathbf{x}}$ is Gross-fractional **then**
        Select the first fractional Gross-digit $\mathbf{c}_{\mathtt{B}}^{q^T}\bar{\mathbf{x}}$ (i.e., the one associated with the highest Gross-power)
        Generate the GOF-CP according to $\mathbf{c}_{\mathtt{B}}^{q^T}\bar{\mathbf{x}}$
      else
        Select the first fractional variable $\bar{x}_k$ with $k \in \{1,\dots,n\}$
        Generate the Fractional Cutting Plane according to $\bar{x}_k$
      end if
      Add the cutting plane to $\tilde{P}_c$
      Let $t := t + 1$
    end if
  end while
end procedure

5. An efficient handling of the addition of new inequalities

In Algorithm 1, we can use the GrossSimplex [30] as GrossLPSolver, which requires solving two problems at each iteration: the auxiliary problem to identify an initial feasible basis and the continuous relaxation of the ILP problem. Nonetheless, the GrossDualSimplex [40] allows for a *warm-start* [43], i.e., it avoids the need to solve the auxiliary problem, by appropriately re-using the previous optimal basis. The remaining of the section is dedicated to provide the technical details of this fact.

Consider a generic problem $\tilde{R}$ having B as optimal basis computed using the GrossSimplex algorithm. Then, suppose to add a cutting plane as a constraint for $\tilde{R}$. The resulting $\tilde{R}'$ problem is fully described by the following matrix and vectors

$$\mathbf{A}' := \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{a}^T & 1 \end{bmatrix}, \qquad \tilde{\mathbf{c}}' := \begin{bmatrix} \tilde{\mathbf{c}} \\ 0 \end{bmatrix}, \qquad \mathbf{b}' := \begin{bmatrix} \mathbf{b} \\ b_{m+1} \end{bmatrix}, \tag{12}$$

where $\mathbf{a} := [a_{m+1,1} \dots a_{m+1,n}]^T$ and $\mathbf{0}$ is a column vector of zeros. The dual $\tilde{D}'$, which is a *Gross-Dual-LP* problem, has the form

$$\begin{aligned} \max \quad & \tilde{\mathbf{y}}^T\mathbf{b}' \\ \text{s.t.} \quad & \tilde{\mathbf{y}}^T\mathbf{A}' \leqslant \tilde{\mathbf{c}}'^T \\ & \tilde{\mathbf{y}} \in \tilde{\mathcal{U}}^{m+1}, \end{aligned} \tag*{$\tilde{D}'$}$$

where the vector $\tilde{\mathbf{y}}$ is a Gross-vector and each unknown $\tilde{y}_u$, $u = 1,\dots,m+1$, belongs to the set $\tilde{\mathcal{U}}$, defined as:

$$\tilde{\mathcal{U}} := \{\tilde{y} = y_1①^0 + \dots + y_r①^{r-1},\ y_q \in \mathbb{R},\ q = 1,\dots,r\}.$$

Now, we are ready to state what follows.

Proposition 1. *Let* $\bar{\tilde{\mathbf{y}}} := \bar{\mathbf{y}}^1①^0 + \ldots + \bar{\mathbf{y}}^r①^{r-1}$ *be the optimal solution of the dual problem* $\tilde{D}$ *of a linear program* $\tilde{R}$ *associated to the basis* B*, and consider the modified problem* $\tilde{D}'$ *obtained from* $\tilde{D}$ *adding one constraint to* $\tilde{R}$ *according to* (12)*. Then,* $\tilde{\mathbf{y}}' := \mathbf{y}'^1①^0 + \ldots + \mathbf{y}'^r①^{r-1}$ *is a feasible basic solution for* $\tilde{D}'$ *associated to the basis* $\mathtt{B}' := \mathtt{B} \cup \{n+1\}$*, where each* $\mathbf{y}'^i$ *is defined according to*

$$\mathbf{y}'^i := \begin{bmatrix} \bar{\mathbf{y}}^i \\ 0 \end{bmatrix}. \tag{13}$$

Proof. Since $\bar{\tilde{\mathbf{y}}}$ is optimal for $\tilde{D}$, it is also feasible. Then, it holds that

$$\bar{\tilde{\mathbf{y}}}^T\mathbf{A}_{\mathtt{B}} = \tilde{\mathbf{c}}_{\mathtt{B}}^T \qquad \text{and} \qquad \bar{\tilde{\mathbf{y}}}^T\mathbf{A}_{\mathtt{N}} \leqslant \tilde{\mathbf{c}}_{\mathtt{N}}^T, \tag{14}$$

where $\mathbf{A}_{\mathtt{B}}$ is the submatrix of $\mathbf{A}$ considering the columns indexed by B, $\tilde{\mathbf{c}}_{\mathtt{B}}$ is the subvector of $\tilde{\mathbf{c}}$ considering the entries indexed by B, and $\mathtt{N} := \{1,\dots,n\} \setminus \mathtt{B}$. To prove that $\tilde{\mathbf{y}}'$ is feasible for $\tilde{D}'$, we need to show that

$$\tilde{\mathbf{y}}'^T\mathbf{A}'_{\mathtt{B}'} = \tilde{\mathbf{c}}'^T_{\mathtt{B}'} \qquad \text{and} \qquad \tilde{\mathbf{y}}'^T\mathbf{A}'_{\mathtt{N}} \leqslant \tilde{\mathbf{c}}'^T_{\mathtt{N}}. \tag{15}$$

By construction (13), it holds true that $\tilde{\mathbf{y}}' = [\bar{\tilde{\mathbf{y}}}^T\ 0]^T$. Thus, also leveraging (12), we can rewrite (15) as

$$[\bar{\tilde{\mathbf{y}}}^T\ 0]\begin{bmatrix}\mathbf{A}_{\mathtt{B}} & \mathbf{0}\\ \mathbf{a}_{\mathtt{B}}^T & 1\end{bmatrix} = [\tilde{\mathbf{c}}_{\mathtt{B}}^T\ 0] \qquad \text{and} \qquad [\bar{\tilde{\mathbf{y}}}^T\ 0]\begin{bmatrix}\mathbf{A}_{\mathtt{N}}\\ \mathbf{a}_{\mathtt{N}}^T\end{bmatrix} \leqslant \tilde{\mathbf{c}}_{\mathtt{N}}^T, \tag{16}$$

where $\mathbf{a}_{\mathtt{B}}$ refers to the entries of $\mathbf{a}$ indexed by $\mathtt{B}$ and $\mathbf{a}_{\mathtt{N}}$ to the remaining ones. Executing the block-wise inner products in (16), we achieve the same conditions as in (14), which are satisfied by hypothesis. This fact proves the thesis. ■

For the sake of completeness, we have reported the pseudocode of the GrossDualSimplex in Alg. 2, rewriting it to make the notation compliant with the one used in this paper.

Algorithm 2 The GrossDualSimplex Algorithm

procedure $[\mathbf{x}, \mathtt{B}]$=GrossDualSimplex($\mathbf{A}$, $\mathbf{b}$, $\tilde{\mathbf{c}}$, $\mathtt{B}$) /* *This algorithm solves problem* $\tilde{D}$ */
  Compute the set of non-basic indices $\mathtt{N}$ as: $\mathtt{N} := \{1, \dots, n\} \setminus \mathtt{B}$
  while true **do**
    /* *Compute the dual basis solution* $\tilde{\mathbf{y}}$ *(a Gross-vector)**/
    /* $\boldsymbol{A}_{\mathtt{B}}$ *is the sub-matrix obtained from* $\boldsymbol{A}$ *considering the columns indexed by* $\mathtt{B}$*/
    $\tilde{\mathbf{y}}^T = \tilde{\mathbf{c}}_{\mathtt{B}}^T \mathbf{A}_{\mathtt{B}}^{-1}$
    /* *Compute the primal basis solution* $\mathbf{x}$ *(a real-valued vector)* */
    $\mathbf{x}_{\mathtt{B}} = \mathbf{A}_{\mathtt{B}}^{-1}\mathbf{b}_{\mathtt{B}}$, $\mathbf{x}_{\mathtt{N}} = 0$
    if $\mathbf{x}_{\mathtt{B}} \geqslant 0$ **then**
      return $\mathbf{x}$, $\mathtt{B}$ /**current solution and basis are optimal**/
    else compute the exiting index h
      $h = \min\{i \in \mathtt{B} : \mathbf{x}_i < 0\}$
      $\mathbf{W}^T := -\mathbf{A}_{\mathtt{B}}^{-1}$
      /* $\boldsymbol{W}_i^T$ *is the i-th row of* $\boldsymbol{W}^T$ */
      if $\mathbf{W}_h^T \mathbf{A}_i \leqslant 0\ \forall i \in \mathbf{N}$ **then**
        return NaN /* *we return Not a Number, being the problem unbounded**/
        /* *(This case never occurs when the GrossDualSimplex is used within GCP method, because**/
        /* *polyhedron is not empty by hypothesis. But this scenario must be considered in standalone mode)**/
      else compute the entering index k
        /* *Compute the maximum feasible gain (a Gross-scalar)**/
        $\tilde{\boldsymbol{\theta}} = \min\{\frac{\tilde{\mathbf{c}}_i^T - \tilde{\mathbf{y}}^T\mathbf{A}_i}{\mathbf{W}_h^T\mathbf{A}_i} : i \in \mathbf{N},\ \mathbf{W}_h^T\mathbf{A}_i > 0\}$
        $k = \min\{i \in \mathbf{N} :\ \mathbf{W}_h^T\mathbf{A}_i > 0,\ \frac{\tilde{\mathbf{c}}_i^T - \tilde{\mathbf{y}}^T\mathbf{A}_i}{\mathbf{W}_h^T\mathbf{A}_i} = \tilde{\boldsymbol{\theta}}\}$
        $\mathtt{B} = \mathtt{B} \setminus \{h\} \cup \{k\}$ /* *update the optimal basis* */
      end if
    end if
  end while
end procedure

6. Experimental results

In this section, we introduce four LMOILP test problems having known solution. Then, we verify that the GCP method combined with the GrossDualSimplex is able to successfully solve these problems.

6.1 Test problem 1: the "kite" in 2D

This problem is a slight variation to the 2D problem with 3 objectives described in [4]:

$$\begin{array}{ll} \text{lexmax} & 8x_1 + 12x_2,\ 14x_1 + 10x_2,\ x_1 + x_2 \\ \text{s.t.} & 2x_1 + x_2 \leqslant 120 \\ & 4x_1 + 6x_2 \leqslant 425 \\ & 4x_1 + 3x_2 \leqslant 270 \\ & x_1 + 2x_2 \geqslant 60 \\ & \mathbf{x} \geqslant 0,\ \ \mathbf{x} \in \mathbb{Z}^2. \end{array} \tag{T_1}$$

The polygon $\mathcal{S}$ associated to this problem is shown in Fig. 1 (left sub-figure). The integer points are shown as black spots, while in light grey we have provided the domain of the continuous relaxation.
It can be seen that the first objective vector $\mathbf{c}^1 = [8,\ 12]^T$ is orthogonal to segment $[\alpha,\ \beta]$ ($\alpha = (0,\ 70.83), \beta = (28.75,\ 51.67)$) shown in the same figure. All the nearest integer points parallel to this segment are optimal for the first objective (see the right sub-figure in Fig. 1). Since the solution is not unique, there is the chance to improve the second objective vector ($\mathbf{c}^2 = [14, 10]^T$). The integer point that

maximises the second objective is $\mathbf{x}_{opt} = [28, 52]^T$. Since now the solution is unique, the third objective function has no role in the optimisation. Thus, the lexicographic integer optimal solution for the problem is $\mathbf{x}_{opt} = [28, 52]^T$.

Let see what happens when we solve this problem using the GCP method. At each iteration, the optimal solution of the continuous relaxation, without slack variables, and the value of the objective function will be reported. Notice that the problem is lexmax-formulated, we have to provide $-\tilde{\mathbf{c}}$ to the GrossDualSimplex algorithm if intended as in Algorithm 2.

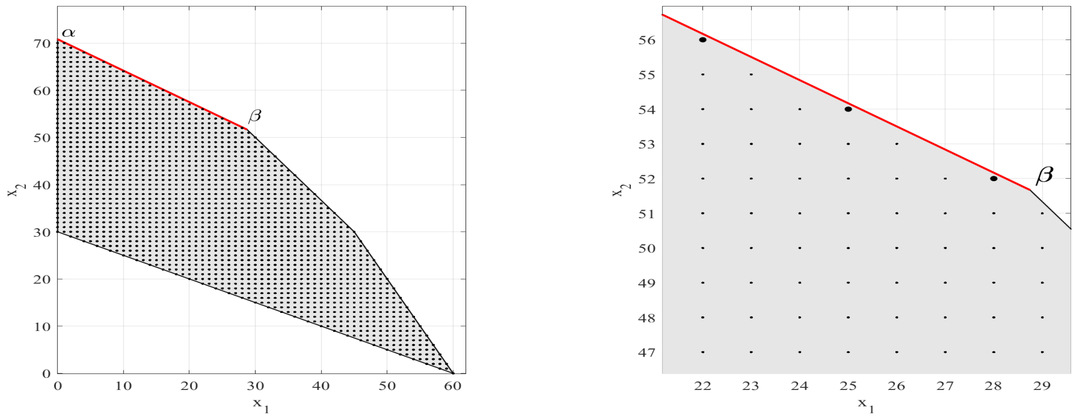

Figure 1: An example in two dimensions with three objectives. The black points on the left figure are all the feasible solutions. All the nearest integer points parallel to the segment $[\alpha,\, \beta]$ (there are many), are optimal for the first objective, while point (28, 52) is the unique lexicographic optimum for the given problem (i.e., considering the second objective too). The third objective plays no role in this case. On the right, a zoom around point β is provided, with some optimal solutions for the first objective highlighted (the ones with a bigger black spot).

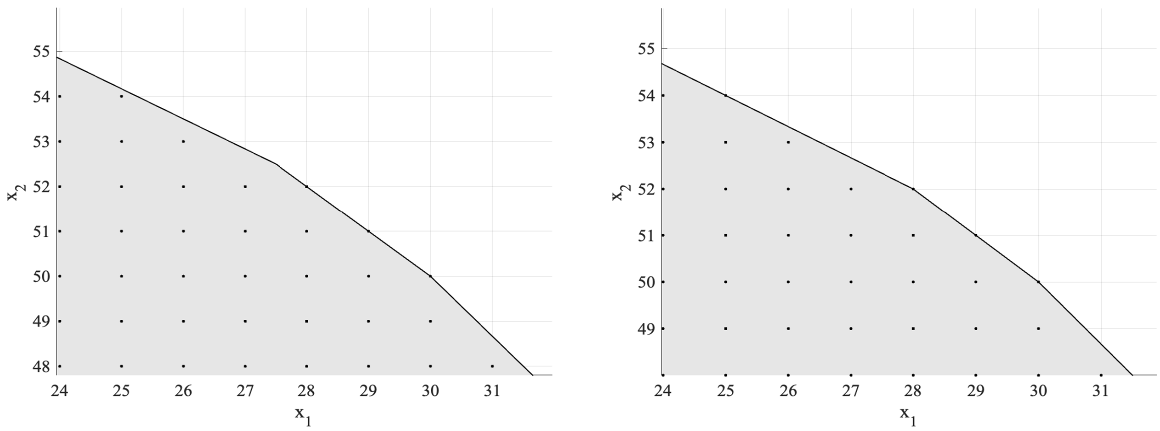

Figure 2: On the left we have the problem $|T_1|$ after iteration 1 and on the right after iteration 3.

Iteration 1 First, we transform our problem in the formulation in which the cutting plane method can be generated, defined as $\tilde{P}_c$ (current problem),

$$\begin{array}{ll} \text{lexmax} & 8x_1 + 12x_2,\ 14x_1 + 10x_2,\ x_1 + x_2 \\ \text{s.t.} & 2x_1 + x_2 + x_3 = 120 \\ & 4x_1 + 6x_2 + x_4 = 425 \\ & 4x_1 + 3x_2 + x_5 = 270 \\ & x_1 + 2x_2 + x_6 = 60 \\ & \mathbf{x} \geqslant 0,\ \ \mathbf{x} \in \mathbb{Z}^6. \end{array}$$

Then, the GrossDualSimplex algorithm solves the continuous relaxation $\tilde{R}_c$, obtaining the solution $\bar{\mathbf{x}} = [28.75,\, 51.66]^T$, which is associated to the following value of the objective function $\tilde{\mathbf{c}}^T\bar{\mathbf{x}}$: $850①^0 + 919.17①^{-1} + 80.42①^{-2}$. Since the latter is Gross-fractional, we lexicographically select its first fractional component, i.e., the one associated to the first objective function, and we generate the following GOF-CP cutting plane (see Th. 1):

$$0.16x_4 + 0.33x_5 \geqslant 0.83.$$

Notice that we can reformulate it according to the original formulation of $|T_1|$ as follows:

$$2x_1 + 2x_2 \leqslant 160.$$

On the left of Fig. 2, one can appreciate the new polyhedron after adding such a constraint.

Iteration 2 The new problem, obtained by adding the above constraint and the corresponding slack variable x_7, is:

$$\begin{array}{ll} \text{lexmax} & 8x_1 + 12x_2,\ 14x_1 + 10x_2,\ x_1 + x_2 \\ \text{s.t.} & 2x_1 + x_2 + x_3 = 120 \\ & 4x_1 + 6x_2 + x_4 = 425 \\ & 4x_1 + 3x_2 + x_5 = 270 \\ & x_1 + 2x_2 + x_6 = 60 \\ & 2x_1 + 2x_2 + x_7 = 160 \\ & \mathbf{x} \geqslant 0,\ \ \mathbf{x} \in \mathbb{Z}^7. \end{array}$$

The new optimal point is $\bar{\mathbf{x}} = [27.5,\ 52.5]^T$, with $\tilde{\mathbf{c}}^T\bar{\mathbf{x}} = 850①^0 + 910①^{-1} + 80①^{-2}$. Since $\tilde{\mathbf{c}}^T\bar{\mathbf{x}}$ is Gross-integer, the next Fractional Cutting Plane is generated by the first entry of the vector $\bar{\mathbf{x}}$. On the left we report the naive representation, while on the right the one according to the original formulation:

$$0.5x_4 + 0.5x_7 \geqslant 0.5, \qquad\qquad 3x_1 + 4x_2 \leqslant 292.$$

Iteration 3 The current problem $\tilde{P}_c$ is

$$\begin{array}{ll} \text{lexmax} & 8x_1 + 12x_2,\ 14x_1 + 10x_2,\ x_1 + x_2 \\ \text{s.t.} & 2x_1 + x_2 + x_3 = 120 \\ & 4x_1 + 6x_2 + x_4 = 425 \\ & 4x_1 + 3x_2 + x_5 = 270 \\ & x_1 + 2x_2 + x_6 = 60 \\ & 2x_1 + 2x_2 + x_7 = 160 \\ & 3x_1 + 4x_2 + x_8 = 292 \\ & \mathbf{x} \geqslant 0,\ \ \mathbf{x} \in \mathbb{Z}^8, \end{array}$$

whose continuous relaxation has $\bar{\mathbf{x}} = [26,\ 53.5]^T$ as optimal solution, with $\tilde{\mathbf{c}}^T\bar{\mathbf{x}} = 850①^0 + 899①^{-1} + 79.5①^{-2}$, which is Gross-fractional. The next GOF-CP will be inspired by the fractional part of the third objective function (left) resulting in the inequality on the right for the original formulation:

$$0.5x_4 \geqslant 0.5, \qquad\qquad 2x_1 + 3x_2 \leqslant 212.$$

The rightmost image of Fig. 2 shows the new polyhedron after having added the cutting plane.

Iteration 4 The optimal solution found by the GrossDualSimplex is now integer and the Gross-ILP problem can be considered correctly optimized: $\bar{\mathbf{x}} = [28,\ 52]^T$ and $\tilde{\mathbf{c}}^T\bar{\mathbf{x}}$: $848①^0 + 912①^{-1} + 80①^{-2}$.

Tab. 1 provides all the GCP iterations solving $|T_1|$.

Table 1: Summary of the iterations of the GCP method on problem $|T_1|$

Iter.	$\tilde{\mathbf{c}}^T\bar{\mathbf{x}}^T$	$\bar{\mathbf{x}}^T$	**Cutting Plane**
1	$850①^0 + 919.17①^{-1} + 80.42①^{-2}$	[28.75, 51.66]	GOF-CP obj. 2
2	$850①^0 + 910①^{-1} + 80①^{-2}$	[27.5, 52.5]	Fractional Cutting Plane on $\bar{x}_1$
3	$850①^0 + 899①^{-1} + 79.5①^{-2}$	[26, 53.5]	GOF-CP obj. 3
4	$848①^0 + 912①^{-1} + 80①^{-2}$	[28, 52]	None

6.2 Test problems 2 and 3: the "house" in 3D and 5D

This illustrative example is in three dimensions with three objectives

$$\begin{array}{ll} \text{lexmax} & x_1, -x_2, -x_3 \\ \text{s.t.} & \quad 4x_i \leqslant \ \ 81 \quad i = 1, ..., 3 \\ & -4x_i \leqslant -41 \quad i = 1, ..., 3 \\ & -x_1 - x_2 \leqslant -26 \\ & -x_1 + x_2 \leqslant \ \ 5 \\ & \mathbf{x} \geqslant 0,\ \ \mathbf{x} \in \mathbb{Z}^3 \end{array} \qquad |T_2|$$

and the domain being the polyhedron shown in Fig. 3. Considering the first objective alone (maximization of x_1), all the nearest integer points parallel to the square having vertices α, β, γ, δ (see Fig. 3) are optimal. Since the optimum is not unique, the use of the second objective refines the optimization to the integer points near to the segment $[\beta, \gamma]$ (see Fig. 4, which provides the plant-view of Fig. 3 with $x_3 = 10.25$). Again, the optimum in not unique. This means that also the third objective can improve the choice of the optimal solution, selecting the nearest integer point to γ as the unique, Such a point, namely $[20, 11, 11]^T$ is the lexicographic integer optimum of the problem.

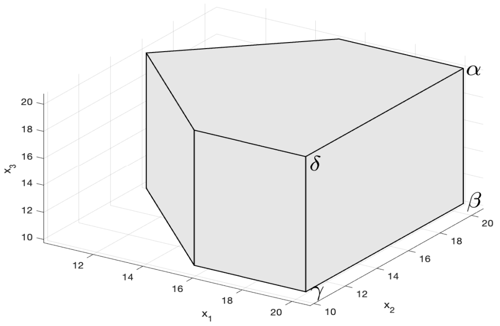

Figure 3: Polyhedron related to problem $|T_2|$

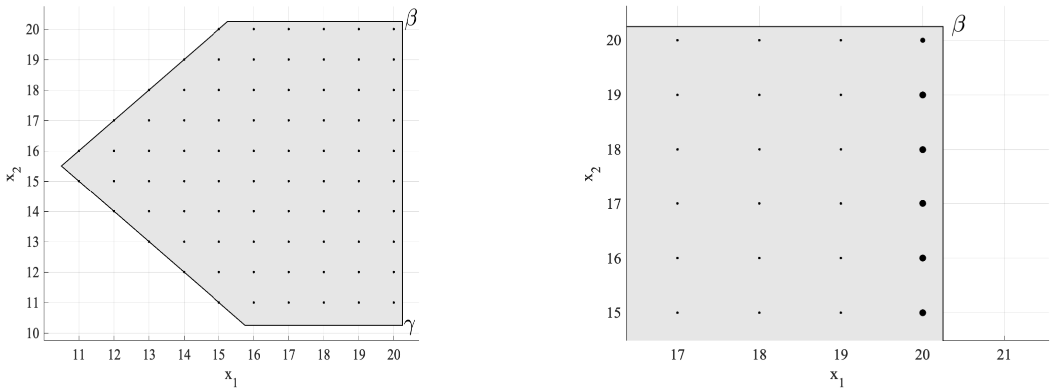

Figure 4: Section view of Fig. 3 with $x_3 = -10$ (left) and its top-right zoom (right).

Tab. 2 shows the seven iterations needed by the GCP method to solve the problem. The optimal value of the objective function can be computed as $\tilde{\mathbf{c}}^T \mathbf{x}_{opt} = 20①^0 - 11①^{-1} - 11①^{-2}$.

Table 2: Iterations of the GCP method on test problem $|T_2|$

It.	$\tilde{\mathbf{c}}^T\bar{\mathbf{x}}$	$\bar{\mathbf{x}}^T$	**Cutting Plane**
1	$20.25①^0 - 10.25①^{-1} - 10.25①^{-2}$	[20.25, 10.25, 10.25]	GOF-CP obj. 1
2	$20①^0 - 10.25①^{-1} - 10.25①^{-2}$	[20, 10.25, 10.25]	GOF-CP obj. 2
3	$20①^0 - 10.33①^{-1} - 10.25①^{-2}$	[20, 10.3333, 10.25]	GOF-CP obj. 2
4	$20①^0 - 10.5①^{-1} - 10.25①^{-2}$	[20, 10.5, 10.25]	GOF-CP obj. 2
5	$20①^0 - 11①^{-1} - 10.25①^{-2}$	[20, 11, 10.25]	GOF-CP obj. 3
6	$20①^0 - 11①^{-1} - 10.33①^{-2}$	[20, 1110.3333]	GOF-CP obj. 3
7	$20①^0 - 11①^{-1} - 10.5①^{-2}$	[20, 11, 10.5]	GOF-CP obj. 3
8	$20①^0 - 11①^{-1} - 11①^{-2}$	[20, 11, 11]	None

Based on the 3D house problem, we developed an algorithm (see Alg. 3) capable of generating house problems in n dimensions.
By means of Alg. 3, we have generate the five dimension problem $|T_3|$:

$$\begin{aligned} \text{lexmax} \quad & x_1, -x_2, ..., -x_5 \\ \text{s.t.} \quad & \left\{\mathbf{x} \in \mathbb{Z}^5 : \mathbf{A}\mathbf{x} \leqslant \mathbf{b}\right\} \end{aligned} \qquad |T_3|$$

Algorithm 3 Generation of a "house" problem in $\mathbb{R}^n$ dimensions

Step 1. The problems is formulated as follow ($\rho \in \mathbb{N}$ is a parameter that controls the size of the house):

$$\begin{aligned} \text{lexmax} \quad & x_1,\ -x_2,\ ...,\ -x_n \\ \text{s.t.} \quad & 4x_i \leqslant 4 \cdot (\rho + 10.25),\ i = 1, ..., n \\ & -4x_i \leqslant -41,\ i = 1, ..., n \\ & -x_1 - x_2 \leqslant -20 - \lceil \rho/2 + 0.5 \rceil \\ & -x_1 + x_2 \leqslant \lceil \rho/2 - 0.25 \rceil \\ & \mathbf{x} \geqslant 0,\ \mathbf{x} \in \mathbb{Z}^n. \end{aligned}$$

Step 2. Optimal integer solution is $\mathbf{x}_{opt} = [\rho + 10,\ \rho + 1,\ \rho + 1,\ \rho + 1,\ ...,\ \rho + 1]^T$.
The optimal value is computed as: $\tilde{f}_{opt} = \tilde{\mathbf{c}}^T \mathbf{x}_{opt}$, where $\tilde{\mathbf{c}}$ is derived from $\mathbf{C}$.

where

$$\mathbf{A} = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ -4 & 0 & 0 & 0 & 0 \\ 0 & -4 & 0 & 0 & 0 \\ 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & 0 & -4 \\ -1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 81 \\ 81 \\ 81 \\ 81 \\ 81 \\ -41 \\ -41 \\ -41 \\ -41 \\ -41 \\ 5 \\ -26 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{c}^{1\,T} \\ \mathbf{c}^{2\,T} \\ ... \\ \mathbf{c}^{5\,T} \end{bmatrix} = \begin{bmatrix} 1 & 0 & .. & 0 \\ 0 & -1 & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & -1 \end{bmatrix} \in \mathbb{R}^{5 \times 5}.$$

Problem $|T_3|$ has the known lexicographic optimum $\mathbf{x}_{opt} = [20,\ 11,\ 11,\ 11,\ 11]^T$. The GCP method solves this third test problem in 13 iterations, as reported in Tab. 3.

Table 3: Iterations of the GCP method on test problem $|T_3|$

It.	$\tilde{\mathbf{c}}^T \bar{\mathbf{x}}^T$	$\bar{\mathbf{x}}^T$	Cutting Plane
1	$-20.25①^0 + 10.25①^{-1} + 10.25①^{-2} + 10.25①^{-3} + 10.25①^{-4}$	[20.25, 10.25, 10.25, 10.25, 10.25]	GOF-CP obj. 1
2	$-20①^0 + 10.25①^{-1} + 10.25①^{-2} + 10.25①^{-3} + 10.25①^{-4}$	[20, 10.25, 10.25, 10.25, 10.25]	GOF-CP obj. 2
3	$-20①^0 + 10.33①^{-1} + 10.25①^{-2} + 10.25①^{-3} + 10.25①^{-4}$	[20 10.33, 10.25, 10.25, 10.25]	GOF-CP obj. 2
4	$-20①^0 + 10.5①^{-1} + 10.25①^{-2} + 10.25①^{-3} + 10.25①^{-4}$	[20, 10.5, 10.25, 10.25, 10.25]	GOF-CP obj. 2
5	$-20①^0 + 11①^{-1} + 10.25①^{-2} + 10.25①^{-3} + 10.25①^{-4}$	[20, 11, 10.25, 10.25, 10.25]	GOF-CP obj. 3
6	$-20①^0 + 11①^{-1} + 10.33①^{-2} + 10.25①^{-3} + 10.25①^{-4}$	[20, 11, 10.33, 10.25, 10.25]	GOF-CP obj. 3
7	$-20①^0 + 11①^{-1} + 10.5①^{-2} + 10.25①^{-3} + 10.25①^{-4}$	[20, 11, 10.5, 10.25, 10.25]	GOF-CP obj. 3
8	$-20①^0 + 11①^{-1} + 11①^{-2} + 10.25①^{-3} + 10.25①^{-4}$	[20, 11, 11, 10.25, 10.25]	GOF-CP obj. 4
9	$-20①^0 + 11①^{-1} + 11①^{-2} + 10.33①^{-3} + 10.25①^{-4}$	[20, 11, 1110.3333, 10.25]	GOF-CP obj. 4
10	$-20①^0 + 11①^{-1} + 11①^{-2} + 10.5①^{-3} + 10.25①^{-4}$	[20, 11, 11, 10.5, 10.25]	GOF-CP obj. 4
11	$-20①^0 + 11①^{-1} + 11①^{-2} + 11①^{-3} + 10.25①^{-4}$	[20, 11, 11, 11, 10.25]	GOF-CP obj. 5
12	$-20①^0 + 11①^{-1} + 11①^{-2} + 11①^{-3} + 10.33①^{-4}$	[20, 11, 11, 11, 10.33]	GOF-CP obj. 5
13	$-20①^0 + 11①^{-1} + 11①^{-2} + 11①^{-3} + 10.5①^{-4}$	[20, 11, 11, 11, 10.5]	GOF-CP obj. 5
14	$-20①^0 + 11①^{-1} + 11①^{-2} + 11①^{-3} + 11①^{-4}$	[20, 11, 11, 11, 11]	None

6.3 Test problem 4: the randomly rotated hypercube in 50D

As a stress test for the GCP method, we have also considered a problem having 50 objectives in $\mathbb{R}^{50}$. The problem, generated thanks to Alg. 4, possesses the following geometry: an external randomly rotated hypercube which has no role in the optimization but adds complexity to the problem, and an inner hypercube which constitutes the truly feasible region of the polyhedron.

Algorithm 4 Generation of a randomly rotated benchmark in n dimensions

---

Step 1. Let $\{\mathbf{A}\mathbf{x} \leqslant \mathbf{b},\ \mathbf{x} \in \mathbb{Z}^n\}$ be the initial unrotated hypercube problem in n-dimensions, formulated as follow:

$$\begin{aligned} \text{lexmax} \quad & x_1, x_2, ..., x_n \\ \text{s.t.} \quad & -1000 \leqslant x_i \leqslant 1000,\ i = 1, ..., n \\ & \mathbf{x} \in \mathbb{Z}^n. \end{aligned}$$

Step 2. Generate a random rotation matrix $\mathbf{Q}$, computed by a QR factorization applied to a random matrix $\mathbf{T}$. Where, $\mathbf{T}$ is a n-by-n matrix of normally distributed random numbers (zero mean and unitary variance). In Matlab:

```
T = randn(n);
[Q, ~] = qr(T);
```

Step 3. Rotate the polytope: $\mathbf{A}' = \lfloor \mathbf{AQ} * 10^2 \rfloor$, $\mathbf{b}' = \mathbf{b} * 10^2$ and $\mathbf{C}' = \mathbf{C}$) and then add the inner hypercube by adding these constraints to $\mathbf{A}'$:

$$0 \leqslant x_i \leqslant (100 + 0.2),\ i = 1, ..., n.$$

Step 4. Compute the LMOILP optimum:
$\mathbf{x}_{opt} = [100, 100, \ldots, 100]^T$ and $\tilde{f}_{opt} = \tilde{\mathbf{c}}^T \mathbf{x}_{opt}$, where $\tilde{\mathbf{c}}$ is derived from $\mathbf{C}'$.

The GCP method is capable to find the optimal integer solution after 101 iterations. Two cutting planes are generated for each objective function $(50 \cdot 2)$ plus the last iteration, in which the optimal integer solution is found.

The significance of this investigation comes from the performance comparison we realized between the GrossSimplex and the GrossDualSimplex. As discussed in Section 5, the use of a dual solver allows one to avoid the search for a new feasible basic solution to restart the optimization after the addition of each cutting plane to the relaxed program. The 50-dimension problem in analysis is a good test case to assess the impact the warm-start has on the optimization, measuring the speedup experienced by the GrossDualSimplex against the GrossSimplex. Tab. 4 resumes this study, show evidences of a significant improvement from a computational perspective. Actually, the warm-start is able to reduce the GCP execution time from 32 seconds to only 1.8 seconds, almost 18 times faster. This is also reflected in the total number of solving iterations, which drastically decreases from 25452 of the GrossSimplex to the only 402 of the GrossDualSimplex. Notice that this computational unburdening comes keeping fixed the number of cutting planes used to solve the problem. This fact is of extreme significance from a practical perspective, giving a strong dignity to the existence and implementation of the GrossDualSimplex, and indicating how the GrossDualSimplex is the algorithm to use for solving integer linear programs.

Table 4: GCP method results on the randomly rotated hypercube in 50D (fourth test problem). GCP was performed without and with warm-start, and using the GrossSimplex and GrossDualSimplex as GrossLPsolver, respectively.

Method	GCP Iter.	Execution time (sec)	Total GrossLPSolver Iter.
GCP without warm-start	101	32s	25452
GCP with warm-start	101	1.8s	402

7. Conclusions

In a previous work [1], we have considered the Lexicographic Multi-Objective Linear Programming problem and, to solve it, we have designed the GrossSimplex algorithm, a generalisation of the well-known simplex algorithm able to work with infinitesimal/infinite quantities using the Grossone methodology. Then, we solved the same problem, but with the addition of the integrality constraint, by generalising the Branch-and-Bound method [39].In this paper we have provided a solution based on a cutting plane method, called GCP method.

First, a new set of cutting planes for LMOILP problems, which we called GOF-CP, has been presented. This cutting plane is inspired by each fractional objective function that composes the multi-objective problem. Then, we have also proved that the GCP method converges to the optimal integer solution by applying the GOF-CP and the Fractional Cutting Plane in an appropriate manner. After that, we efficiently handled the addition of new constraints by exploiting the duality theory and the GrossDual-Simplex introduced in [40]. Finally, through several LMOILP test problems having known solutions, we have verified that the proposed GCP method was able to find the optimal solution to all the proposed problems.

Acknowledgements

Authors Cococcioni, Pappalardo and Sergeyev wish to thank the Gruppo Nazionale per il Calcolo Scientifico (GCNS), for funding the 2022 GNCS project titled "Numerical algorithms in optimization, IDEs, and applications" (grant number CUP_E55F2200270001). Authors Cococcioni, Cudazzo and Fiaschi wish to thank the Italian Ministry of Education, Universities and Research, for funding the FoReLab project (Departments of Excellence, 2023-2027).

Appendix. The Fractional Cutting Plane

The following well-known theorem involves only real numbers, neither Gross-scalars nor Gross-vectors. We provide its proof here for the sake of completeness and of consistency with the notation used in this paper.

Theorem 2 (Fractional Cutting Plane)**.** *Let* $\mathtt{B}$ *be an optimal basis for problem* $\tilde{R}$*,* $\bar{\mathbf{x}}$ *the associated optimal solution, and* $\mathtt{N} := \{1, \ldots, n\} \setminus \mathtt{B}$*. If there exists a component* s *in vector* $\bar{\mathbf{x}}_{\mathtt{B}}$ *such that* $(\bar{\mathbf{x}}_{\mathtt{B}})_s$ *is not integer, then the following inequality*

$$\sum_{j \in \mathtt{N}} \{(\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj}\} x_j \geqslant \{(\bar{\mathbf{x}}_{\mathtt{B}})_s\} \tag{17}$$

is a cutting plane for $\mathcal{S}$*.*

Proof. Let us consider an arbitrary $\mathbf{x} \in \Omega$, we have that:

$$\begin{aligned}(\mathbf{x}_{\mathtt{B}})_s &= (\bar{\mathbf{x}}_{\mathtt{B}})_s - \sum_{j \in \mathtt{N}} (\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj} x_j \\ &= \lfloor (\bar{\mathbf{x}}_{\mathtt{B}})_s \rfloor + \{(\bar{\mathbf{x}}_{\mathtt{B}})_s\} - \sum_{j \in \mathtt{N}} \lfloor (\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj} \rfloor x_j - \sum_{j \in \mathtt{N}} \{(\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj}\} x_j.\end{aligned}$$

Then it follows that:

$$\sum_{j \in \mathtt{N}} \{(\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj}\} x_j - \{(\bar{\mathbf{x}}_{\mathtt{B}})_s\} = \lfloor (\bar{\mathbf{x}}_{\mathtt{B}})_s \rfloor - \sum_{j \in \mathtt{N}} \lfloor (\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj} \rfloor x_j - (\mathbf{x}_{\mathtt{B}})_s, \tag{18}$$

where the right-hand side is integer:

$$\lfloor (\bar{\mathbf{x}}_{\mathtt{B}})_s \rfloor - \sum_{j \in \mathtt{N}} \lfloor (\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj} \rfloor x_j - (\mathbf{x}_{\mathtt{B}})_s \in \mathbb{Z}. \tag{19}$$

Indeed, both x_j and $(\mathbf{x}_{\mathtt{B}})_s$ are integers, since $\mathbf{x} \in \Omega$ by hypothesis.

Let us make an observation concerning the left-hand side of (18):

$$\sum_{j \in \mathtt{N}} \{(\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj}\} x_j - \{(\bar{\mathbf{x}}_{\mathtt{B}})_s\} \geqslant -\{(\bar{\mathbf{x}}_{\mathtt{B}})_s\} > -1 \tag{20}$$

Indeed, $\sum_{j \in \mathtt{N}} \{(\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj}\} x_j \geqslant 0$ because $\{(\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj}\}$ are strictly positive values, and x_j are $\geqslant 0$ by hypothesis. Instead, $\{(\bar{\mathbf{x}}_{\mathtt{B}})_s\}$ is the fractional part of $(\bar{\mathbf{x}}_{\mathtt{B}})_s$ and, by definition, it will fall within 0 and 1 (more precisely, $0 \leqslant \{(\bar{\mathbf{x}}_{\mathtt{B}})_s\} < 1$). Being the right hand side of (18) integer for any $\mathbf{x} \in \Omega$ (as shown in (19)), we can conclude that:

$$\sum_{j \in \mathtt{N}} \{(\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj}\} x_j - \{(\bar{\mathbf{x}}_{\mathtt{B}})_s\} \geqslant 0, \tag{21}$$

which can be rewritten as:

$$\sum_{j\in\mathtt{N}}\{(\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj}\}x_j \geqslant \{(\bar{\mathbf{x}}_{\mathtt{B}})_s\}.$$

So, the last inequality holds for an arbitrary point $\mathbf{x} \in \Omega$ (condition i) of Def. 2. Instead, (17) satisfy condition ii) of Def. 2:

$$\sum_{j\in\mathtt{N}}\{(\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj}\}\bar{x}_j \ngeqslant \{(\bar{\mathbf{x}}_{\mathtt{B}})_s\},$$

being $\bar{x}_j = 0 \;\; \forall j \in \mathtt{N}$ and 0 is not greater or equal to a strictly positive number:

$$0 \ngeqslant \{(\bar{\mathbf{x}}_{\mathtt{B}})_s\}.$$

Therefore, (17) is a cutting plane. ■

Let us observe now how the Fractional Cutting Plane, defined in (17), can be alternatively (and equivalently) expressed in integer form, as shown in Thm. 4.

Theorem 4 (Fractional Cutting Plane in integer form)**.** *Let* $\mathtt{B}$ *be an optimal basis for problem* $\tilde{R}$*,* $\bar{\mathbf{x}}$ *the associated optimal solution, and* $\mathtt{N} := \{1, \dots, n\} \setminus \mathtt{B}$*. If there exists a component* s *in vector* $\bar{\mathbf{x}}_{\mathtt{B}}$ *such that* $(\bar{\mathbf{x}}_{\mathtt{B}})_s$ *is not integer, then the following inequality*

$$\sum_{j\in\mathtt{N}}\lfloor(\mathbf{A}_{\mathtt{B}}^{-1}\mathbf{A}_{\mathtt{N}})_{sj}\rfloor x_j + (\mathbf{x}_{\mathtt{B}})_s \;\leqslant \lfloor(\bar{\mathbf{x}}_{\mathtt{B}})_s\rfloor \tag{22}$$

is a cutting plane for problem $\mathcal{S}$*.*

Proof. The proof is omitted, since it is standard and it can be found on most books devoted to integer linear programming. ■

Finally, observe how this integer form is very useful to avoid numerical problems caused by rounding, making this formulation of the cutting plane more numerically stable.

References

[1] M. Cococcioni, M. Pappalardo, and Ya. D. Sergeyev. Lexicographic multi-objective linear programming using grossone methodology: Theory and algorithm. *Applied Mathematics and Computation*, 318:298–311, 2018.

[2] S. Khosravani, M. Jalali, A. Khajepour, A. Kasaiezadeh, S. K. Chen, and B. Litkouhi. Application of lexicographic optimization method to integrated vehicle control systems. *IEEE Transactions on Industrial Electronics*, 65(12):9677–9686, 2018.

[3] E. Weber, A.E. Rizzoli, R. Soncini-Sessa, and A. Castelletti. A lexicographic optimization in water resource planning: the case of lake verbano, italy. In *Proc. 1st Biennial Meeting of the International Environmental Modelling and Software Society (IEMSS)*, 2002.

[4] I.P. Stanimirovic. Compendious lexicographic method for multi-objective optimization. *Facta universitatis - series: Mathematics and Informatics*, 27(1):55–66, 2012.

[5] J. Marques-Silva, J. Argelich, A. Graça, and I. Lynce. Boolean lexicographic optimization: algorithms & applications. *Annals of Mathematics and Artificial Intelligence*, 62(3):317–343, 2011.

[6] Ya. D. Sergeyev. Numerical infinities and infinitesimals: Methodology, applications, and repercussions on two hilbert problems. *EMS Surveys in Mathematical Sciences*, 4:219–320, 2017.

[7] Ya. D. Sergeyev and R. De Leone, editors. *Numerical Infinities and Infinitesimals in Optimization*. Springer, 2022.

[8] Rocco Gangle, Gianluca Caterina, and Fernando Tohmé. A constructive sequence algebra for the calculus of indications. *Soft Computing*, 24(23):17621–17629, Dec 2020.

[9] G. Lolli. Infinitesimals and infinites in the history of mathematics: A brief survey. *Applied Mathematics and Computation*, 218(16):7979–7988, 2012.

[10] G. Lolli. Metamathematical investigations on the theory of grossone. *Applied Mathematics and Computation*, 255:3–14, 2015.

[11] F. Montagna, G. Simi, and A. Sorbi. Taking the Pirahã seriously. *Communications in Nonlinear Science and Numerical Simulation*, 21(1–3):52–69, 2015.

[12] Ya. D. Sergeyev. Independence of the grossone-based infinity methodology from non-standard analysis and comments upon logical fallacies in some texts asserting the opposite. *Foundations of Science*, 24(1):153–170, Mar 2019.

[13] Ya. D. Sergeyev. Some paradoxes of infinity revisited. *Mediterranean Journal of Mathematics*, 19(3):143, 2022.

[14] Ya. D. Sergeyev. *Computer system for storing infinite, infinitesimal, and finite quantities and executing arithmetical operations with them.* USA patent 7,860,914, 2010.

[15] P. Amodio, F. Iavernaro, F. Mazzia, M.S. Mukhametzhanov, and Ya. D. Sergeyev. A generalized Taylor method of order three for the solution of initial value problems in standard and infinity floating-point arithmetic. *Mathematics and Computers in Simulation*, 141:24–39, 2017.

[16] Ya. D. Sergeyev, M.S. Mukhametzhanov, F. Mazzia, F. Iavernaro, and P. Amodio. Numerical methods for solving initial value problems on the Infinity Computer. *Int. Journal of Unconventional Computing*, 12(1):3–23, 2016.

[17] A. Falcone, A. Garro, M. S. Mukhametzhanov, and Ya. D. Sergeyev. A simulink-based software solution using the infinity computer methodology for higher order differentiation. *Applied Mathematics and Computation*, 409:125606, 2021.

[18] Ya. D. Sergeyev. The exact (up to infinitesimals) infinite perimeter of the Koch snowflake and its finite area. *Communications in Nonlinear Science and Numerical Simulation*, 31(1–3):21–29, 2016.

[19] F. Caldarola and M. Maiolo. On the topological convergence of multi-rule sequences of sets and fractal patterns. *Soft Computing*, 24(23):17737–17749, 2020.

[20] D. I. Iudin, Ya. D. Sergeyev, and M. Hayakawa. Infinity computations in cellular automaton forest-fire model. *Communications in Nonlinear Science and Numerical Simulation*, 20(3):861–870, 2015.

[21] Ya. D. Sergeyev and A. Garro. Single-tape and multi-tape turing machines through the lens of the grossone methodology. *Journal of Supercomputing*, 65(2):645–663, 2013.

[22] D. Rizza. Numerical methods for infinite decision-making processes. *Int. Journal of Unconventional Computing*, 14(2):139–158, 2019.

[23] A. Falcone, A. Garro, M. S. Mukhametzhanov, and Ya. D. Sergeyev. Simulation of hybrid systems under Zeno behavior using numerical infinitesimals. *Communications in Nonlinear Science and Numerical Simulation*, 111:106443, 2022.

[24] L. D'Alotto. Infinite games on finite graphs using grossone. *Soft Computing*, 24(23):17509–17515, 2020.

[25] L. Fiaschi and M. Cococcioni. Non-archimedean game theory: A numerical approach. *Applied Mathematics and Computation*, 409:125356, 2021.

[26] C. S. Calude and M. Dumitrescu. Infinitesimal probabilities based on grossone. *SN Computer Science*, 1(1):36, 2019.

[27] L. Antoniotti, F. Caldarola, G. d'Atri, and M. Pellegrini. New approaches to basic calculus: An experimentation via numerical computation. In Ya. D. Sergeyev and D. E. Kvasov, editors, *Numerical Computations: Theory and Algorithms*, pages 329–342. Springer International Publishing, 2020.

[28] P. Iannone, D. Rizza, and A. Thoma. Investigating secondary school students' epistemologies through a class activity concerning infinity. In E. Bergqvist, M. Österholm, C. Granberg, and L. Sumpter, editors, *Proceedings of the 42nd Conference of the International Group for the Psychology of Mathematics Education*, volume 3, pages 131–138, 2018.

[29] F. Ingarozza, M. T. Adamo, M. Martino, and A. Piscitelli. A grossone-based numerical model for computations with infinity: A case study in an italian high school. In Ya. D. Sergeyev and D. E. Kvasov, editors, *Numerical Computations: Theory and Algorithms*, pages 451–462. Springer International Publishing, 2020.

[30] M. Cococcioni, M. Pappalardo, and Ya. D. Sergeyev. Towards lexicographic multi-objective linear programming using grossone methodology. In Sergeyev Ya. D., Kvasov D. E., Dell'Accio F., and Mukhametzhanov M.S., editors, *Proc. of*

the 2nd Intern. Conf. "Numerical Computations: Theory and Algorithms", volume 1776, page 090040. AIP Publishing, New York, 2016.

[31] L. Lai, L. Fiaschi, M. Cococcioni, and K. Deb. Solving mixed pareto-lexicographic multiobjective optimization problems: The case of priority levels. *IEEE Transactions on Evolutionary Computation*, 25(5):971–985, 2021.

[32] R. De Leone, G. Fasano, and Ya.D. Sergeyev. Planar methods and grossone for the conjugate gradient breakdown in nonlinear programming. *Computational Optimization and Applications*, 71:73–93, 2018.

[33] S. De Cosmis and R. De Leone. The use of grossone in mathematical programming and operations research. *Applied Mathematics and Computation*, 218(16):8029–8038, 2012.

[34] R. De Leone. Nonlinear programming and grossone: Quadratic programming and the role of constraint qualifications. *Applied Mathematics and Computation*, 318:290–297, 2018.

[35] M. Gaudioso, G. Giallombardo, and M. S. Mukhametzhanov. Numerical infinitesimals in a variable metric method for convex nonsmooth optimization. *Applied Mathematics and Computation*, 318:312–320, 2018.

[36] Ya. D. Sergeyev, Kvasov D. E., and Mukhametzhanov M.S. On strong homogeneity of a class of global optimization algorithms working with infinite and infinitesimal scales. *Communications in Nonlinear Science and Numerical Simulation*, 59:319–330, 2018.

[37] R. De Leone, G. Fasano, M. Roma, and Ya. D. Sergeyev. Iterative grossone-based computation of negative curvature directions in large-scale optimization. *Journal of Optimization Theory and Applications*, 186(2):554–589, 2020.

[38] A. Astorino and A. Fuduli. Spherical separation with infinitely far center. *Soft Comput.*, 24(23):17751–17759, 2020.

[39] M. Cococcioni, A. Cudazzo, M. Pappalardo, and Ya. D. Sergeyev. Solving the lexicographic multi-objective mixed-integer linear programming problem using branch-and-bound and grossone methodology. *Communications in Nonlinear Science and Numerical Simulation*, 84:105177, 2020.

[40] M. Cococcioni and L. Fiaschi. A Simplex Algorithm for Linear Programming with Non-Archimedean Right-Hand sides. *submitted to an international journal (pre-print available at https://doi.org/10.21203/rs.3.rs-2772714/v1).*

[41] M. Cococcioni, A. Cudazzo, M. Pappalardo, and Ya. D. Sergeyev. *Multi-objective Lexicographic Mixed-Integer Linear Programming: An Infinity Computer Approach*, pages 119–149. Springer International Publishing, 2022.

[42] M. Conforti, G. Cornuejols, and G. Zambelli. *Integer Programming*. Springer Publishing Company, 2014.

[43] V. Chvátal. *Linear programming*. A Series of books in the mathematical sciences. W.H. Freeman, 1983.