

Lexicographic Multi-Objective Linear Programming using Grossone Methodology: Theory and Algorithm

Marco Cococcioni * Massimo Pappalardo †

Yaroslav D. Sergeyev ‡

Abstract

Numerous problems arising in engineering applications can have several objectives to be satisfied. An important class of problems of this kind is lexicographic multi-objective problems where the first objective is incomparably more important than the second one which, in its turn, is incomparably more important than the third one, etc. In this paper, Lexicographic Multi-Objective Linear Programming (LMOLP) problems are considered. To tackle them, traditional approaches either require solution of a series of linear programming problems or apply a scalarization of weighted multiple objectives into a single-objective function. The latter approach requires finding a set of weights that guarantees the equivalence of the original problem and the single-objective one and the search of correct weights can be very time consuming. In this work a new approach for solving LMOLP problems using a recently introduced computational methodology allowing one to work numerically with infinities and infinitesimals is proposed. It is shown that a smart application of infinitesimal weights allows one to construct a single-objective problem avoiding the necessity to determine finite weights. The equivalence between the original multi-objective problem and the new single-objective one is proved. A simplex-based algorithm working with finite and infinitesimal numbers is proposed, implemented, and discussed. Results of some numerical experiments are provided.

Key Words: Multi-Objective Optimization, Lexicographic Problems, Numerical Infinitesimals, Grossone Infinity Computing

*University of Pisa, Dipartimento di Ingegneria dell'Informazione, Largo Lucio Lazzarino, 1 - 56122 Pisa, Italy, marco.cococcioni@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

1 Introduction

Engineering applications often lead to optimization problems where several objectives should be satisfied. An important class of problems of this kind is lexicographic multi-objective problems where the first objective is incomparably more important than the second one which, in its turn, is incomparably more important than the third one, etc. In case each of the objectives is represented by a linear function under linear constraints, Lexicographic Multi-Objective Linear Programming (LMOLP) problems are considered. Traditionally LMOLP problems are solved in two different ways (see, e.g., [7, 14, 36, 37]). The first one called *preemptive approach* consists of solving a sequence of single-objective linear programming (LP) problems, where new constraints are added to the subsequent LP problem once an optimal solution to the previous problem has been found. Clearly, this approach is time consuming. The second approach known as *nonpreemptive* one (see [36]) transforms LMOLP into a single-objective LP problem by using a weighted sum of the objectives. This approach has the difficulty to find the weights which guarantee, a-priori, the equivalence of the resulting single-objective problem with the original multi-objective one. In practice, determining such weights is a tricky task and it can be even more time consuming than the preemptive approach.

In the present paper it is proposed to analyze LMOLP problems using a recently introduced computational methodology allowing one to work *numerically* with infinities and infinitesimals in a handy way (see for a detailed introduction surveys [19, 23, 28, 30, 32] and the book [17] written in a popular way). This computational methodology has already been successfully applied in optimization and numerical differentiation (see [3, 6, 24, 40]) and in a number of other theoretical and computational research areas such as cellular automata (see [4, 5]), Euclidean and hyperbolic geometry (see [11, 12]), percolation (see [8, 9, 38]), fractals (see [18, 20, 26, 31, 38]), infinite series and the Riemann zeta function (see [21, 25, 39]), the first Hilbert problem, Turing machines, and supertasks (see [15, 22, 33, 34]), numerical solution of ordinary differential equations (see [1, 13, 27, 35]), etc.

This methodology uses a numeral system working with an infinite number called *grossone*, expressed by the numeral $\textcircled{1}$, and introduced as the number of elements of the set of natural numbers (the non-contradictory of the methodology has been studied in [10]). This numeral system allows one to express a variety of numbers involving different infinite and infinitesimal parts and to execute operations with all of them in a unique framework. Notice that this numerical approach is not related to the famous non-standard analysis (see [16]) that has a symbolic character and, therefore, allows symbolic computations only, whereas the present text is dedicated to numerical optimization methods.

Following the guidelines traced in [2, 29] for working with problems

involving lexicographic ordering, it is proposed hereinafter to transform LMOLP into a single-objective LP problem by multiplying the most important objective by 1, the second by $\mathbb{1}^{-1}$, the third by $\mathbb{1}^{-2}$ etc., where $\mathbb{1}^{-i}, 1 < i \leq r$, are infinitesimals and r is a finite number of objectives. It is shown then that, after this transformation, the resulting single-objective LP problem formulated with the help of *grossone*-based numbers can be solved only once by using a simplex-like method working with *grossone*-based numbers that can include infinitesimal parts. The overall advantage of this approach consists of the possibility to solve only one LP problem, without the need to look for correct finite weights that should be provided to an algorithm if the traditional nonpreemptive scheme is applied.

The remaining text is structured as follows. In Section 2, the lexicographic multi-objective linear programming problem is stated and the preemptive and nonpreemptive schemes are described. The *grossone* methodology is briefly presented in Section 3 whereas Section 4 introduces the nonpreemptive *grossone*-based scheme. In Section 5, a theoretical analysis of the introduced approach is performed and the equivalence between the original multi-objective problem and the new single-objective one is proved. Section 6 provides the *gross*-simplex algorithm able to solve the resulting problem. Finally, Section 7 presents some promising experimental results.

2 Lexicographic multi-objective linear programming

Given the LMOLP problem:

$$\begin{aligned} \text{LexMax } & \mathbf{c}^1 \cdot \mathbf{x}, \mathbf{c}^2 \cdot \mathbf{x}, \dots, \mathbf{c}^r \cdot \mathbf{x} \\ \text{s.t. } & \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\} \end{aligned} \quad (\text{P1})$$

where $\mathbf{c}^i, i = 1, \dots, r$, are row vectors $\in \mathbb{R}^n$, \mathbf{x} is a column vector $\in \mathbb{R}^n$, \mathbf{A} is a full-rank matrix $\in \mathbb{R}^{m \times n}$, \mathbf{b} is a column vector $\in \mathbb{R}^m$, and \cdot is the standard scalar product between two real vectors. LexMax in (P1) denotes *Lexicographic Maximum* and means that the first objective is much more important than the second, which is, on its turn, much more important than the third one, and so on. Sometimes in literature this is denoted as $\mathbf{c}^1 \cdot \mathbf{x} \gg \mathbf{c}^2 \cdot \mathbf{x} \gg \dots \gg \mathbf{c}^r \cdot \mathbf{x}$. As in any LP problem, the domain of (P1) is a polytope:

$$S \equiv \{x \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}. \quad (1)$$

Notice that the formulation of (P1) makes no use of *gross*-numbers or *gross*-arrays involving $\mathbb{1}$, namely, it involves finite numbers only. Hereinafter we assume that S is bounded and non-empty. In the literature (see, e.g., [7, 14, 36, 37]), there exists two approaches for solving the problem (P1): the *preemptive scheme* and the *nonpreemptive scheme*. They are described in the following two subsections.

2.1 The preemptive scheme

The preemptive scheme introduced in [36] is an iterative method that attacks (P1) by solving a series of single-objective LP problems. It starts by considering the first objective function alone, i.e., by solving the following problem:

$$\begin{aligned} \text{Max } & \mathbf{c}^1 \cdot \mathbf{x} \\ \text{s.t. } & \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\} \end{aligned} \quad (\text{P2.1})$$

Since (P2.1) is a canonical LP problem, it can be solved algorithmically using any standard method (e.g., the simplex algorithm, the interior point algorithm, etc.). Once they have been run, an optimal solution \mathbf{x}^{*1} with the optimal value $\beta_1 = \mathbf{c}^1 \cdot \mathbf{x}^{*1}$ has been obtained. Then the preemptive scheme considers the second objective of (P1) and solves another single-objective LP problem, where the domain has changed, due to the addition of the equality constraint $\mathbf{c}^1 \cdot \mathbf{x} = \beta_1$:

$$\begin{aligned} \text{Max } & \mathbf{c}^2 \cdot \mathbf{x} \\ \text{s.t. } & \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0, \mathbf{c}^1 \cdot \mathbf{x} = \beta_1\} \end{aligned} \quad (\text{P2.2})$$

The algorithm stops either when the last problem has been solved, i.e., after considering the last objective $\mathbf{c}^r \cdot \mathbf{x}$ or when a non-unique solution has been found in the current solved LP problem. Clearly, this approach is time consuming.

2.2 The nonpreemptive scheme based on appropriate finite weights

It has been shown in [36] that there always exists a finite scalar $M \in \mathbb{R}$ such that the solution of the LMOLP problem (P1) can be found by solving only one single-objective LP problem having the following form:

$$\begin{aligned} \text{Max } & \bar{\mathbf{c}} \cdot \mathbf{x} \\ \text{s.t. } & \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\} \end{aligned} \quad (\text{P3})$$

where

$$\bar{\mathbf{c}} = \sum_{i=1}^r \mathbf{c}^i M^{-i+1}. \quad (2)$$

This is a powerful theoretical result. However, from the computational point of view, finding the value of M is not a trivial task. Finding an appropriate value of M and solving the resulting LP problem can be more time consuming than solving the original problem (P1) following the preemptive approach. Indeed, the preemptive scheme requires solving r linear programming problems only in the worst case and, in addition, it does not require the computation of M .

In Section 4, we present a nonpreemptive approach based on infinitesimal weights (constructed by using *grossone* integer powers), that overcomes the problem of computing M and still requires the solution of only one single-objective LP problem. Such an LP problem is, however, not a standard one and thus it will be called *gross-LP* problem, to avoid any possible confusion with its standard formulation involving finite numbers only. In the following section we give an information regarding $\textcircled{1}$ methodology that is required to introduce our nonpreemptive approach based on infinitesimal weights.

3 A brief excursus to the grossone methodology

As said before, in [17, 19, 23, 28, 30, 32] a computational methodology working with an infinite unit of measure called *grossone* and indicated by the numeral $\textcircled{1}$ has been introduced as the number of elements of the set of natural numbers \mathbb{N} . On the one hand, this allows one to treat easily many problems related to the traditional set theory operating with Cantor's cardinals by computing the number of elements of infinite sets using $\textcircled{1}$ -based numerals. For instance, the following sets that the traditional cardinalities identify as countable can be measured more precisely. In fact, the set of even numbers \mathbb{E} has $\frac{\textcircled{1}}{2}$ elements (see [22, 28, 30, 32]), namely two times less than the set of natural numbers. The set of integers \mathbb{Z} has $2\textcircled{1}+1$ elements, the set \mathbb{G} of square natural numbers

$$\mathbb{G} = \{x : x = n^2, x \in \mathbb{N}, n \in \mathbb{N}\}$$

has $\lfloor \sqrt{\textcircled{1}} \rfloor$ elements, etc. Analogously, it becomes possible to discern among sets having the traditional cardinality of continuum infinite sets with different number of elements. For instance, it follows that the set of numbers $x \in [0, 1)$ expressed in the binary positional numeral system is equal to $2^{\textcircled{1}}$ and the set of numbers $x \in [0, 1)$ expressed in the decimal positional numeral system has $10^{\textcircled{1}} > 2^{\textcircled{1}}$ elements (for more examples see [17, 22, 23, 28, 30, 32]).

On the other hand, in the numeral system built upon *grossone*, there is the opportunity to treat infinite and infinitesimal numbers in a unique framework and to work with all of them numerically, i.e., by executing arithmetic operations with floating-point numbers and the possibility to assign concrete infinite and infinitesimal values to variables. This is one of the differences with non-standard analysis (see [16]) where non-standard infinite numbers are discussed but, if K is a non-standard infinite integer, there is no possibility to assign a value to K , it always remains just a symbol without any concrete numerical value and only symbolic computations can be executed with it.

The new numeral $\textcircled{1}$ is introduced by describing its properties (following the same approach that lead to the introduction of the zero in the past to switch from natural to integer numbers). To introduce *grossone*, three

methodological postulates and The Infinite Unit Axiom is added to the axioms of real numbers (see [17, 22, 23, 28, 30, 32]). In particular, this axiom states that for any given finite integer n the infinite number $\frac{\mathbb{1}}{n}$ is integer being larger than any finite number. Since the axiom is added to the standard axioms of real numbers, all standard properties (commutative, associative, existence of inverse, etc.) also apply to $\mathbb{1}$ and *grossone*-based numerals. Instead of the usual symbol ∞ different infinite and/or infinitesimal numerals can be used thanks to $\mathbb{1}$. Indeterminate forms are not present and, for example, the following relations hold for infinite numbers $\mathbb{1}$, $\mathbb{1}^2$ and infinitesimals $\mathbb{1}^{-1}$, $\mathbb{1}^{-2}$, as for any other (finite, infinite, or infinitesimal) number expressible in the new numeral system:

$$\begin{aligned} 0 \cdot \mathbb{1} &= \mathbb{1} \cdot 0 = 0, & \mathbb{1} - \mathbb{1} &= 0, & \frac{\mathbb{1}}{\mathbb{1}} &= 1, & \mathbb{1}^0 &= 1, & 1^{\mathbb{1}} &= 1, & 0^{\mathbb{1}} &= 0, \\ 0 \cdot \mathbb{1}^{-1} &= \mathbb{1}^{-1} \cdot 0 = 0, & \mathbb{1}^{-1} &> \mathbb{1}^{-2} > 0, & \mathbb{1}^{-1} - \mathbb{1}^{-1} &= 0, & 2\mathbb{1} - \mathbb{1} &= \mathbb{1}, \\ \frac{\mathbb{1}^{-1}}{\mathbb{1}^{-1}} &= 1, & (\mathbb{1}^{-1})^0 &= 1, & \mathbb{1} \cdot \mathbb{1}^{-1} &= 1, & \mathbb{1} \cdot \mathbb{1}^{-2} &= \mathbb{1}^{-1}, \\ \frac{5\mathbb{1}^{-2}}{\mathbb{1}^{-2}} &= 5, & \frac{6.1\mathbb{1}^2}{\mathbb{1}} &= 6.1\mathbb{1}, & \frac{\mathbb{1}^{-1}}{2\mathbb{1}^{-2}} &= 0.5\mathbb{1}, & \mathbb{1}^2 \cdot \mathbb{1}^{-1} &= \mathbb{1}, & \mathbb{1}^2 \cdot \mathbb{1}^{-2} &= 1. \end{aligned}$$

A general way to express infinities and infinitesimals is also provided in [17, 22, 23, 28, 30, 32] by using records similar to traditional positional number systems, but with the radix $\mathbb{1}$. A number \tilde{c} in this new numeral system (\tilde{c} will be called *gross-scalar* from here on) can be constructed by subdividing it into groups of corresponding powers of $\mathbb{1}$ and thus can be represented as

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

where $m, k \in \mathbb{N}$, exponents p_i are called *gross-powers* (they can be numbers of the type of \tilde{c}) with $p_0 = 0$, and $i = m, \dots, 1, 0, -1, \dots, -k$. Then, $c_{p_i} \neq 0$ called *gross-digits* are finite (positive or negative) numbers, $i = m, \dots, 1, 0, -1, \dots, -k$. In this numeral system, finite numbers are represented by numerals with the highest *gross-power* equal to zero, e.g., $-6.2 = -6.2\mathbb{1}^0$. Infinitesimals are represented by numerals having negative finite or infinite *gross-powers*. The simplest infinitesimal is $\mathbb{1}^{-1}$ for which $\mathbb{1}^{-1} \cdot \mathbb{1} = 1$. We notice that all infinitesimals are not equal to zero, e.g., $\mathbb{1}^{-1} > 0$. A number is infinite if it has at least one positive finite or infinite *gross-power*. For instance, the number $43.6\mathbb{1}^{4.56\mathbb{1}} + 16.7\mathbb{1}^{3.6} - 3.2\mathbb{1}^{-2.1}$ is infinite, it consists of two infinite parts and one infinitesimal part.

In the context of this paper the following definition is important. A *gross-number* (*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 $3.4 - 3.2\mathbb{1}^{-2.1}$ is finite but not purely finite since it has an infinitesimal part.

4 The nonpreemptive *grossone*-based scheme

In order to introduce the scheme we need the following definitions and notations. Hereinafter, an array made of *gross*-scalars is called *gross*-array. In particular, a vector made of *gross*-scalars is called from here on *gross*-vector. The definition of *gross*-matrix is similar. By extension, *gross*-vectors and *gross*-matrices are called *purely finite* iff all of their entries are purely finite *gross*-numbers. Let us recollect our notations now:

- Italics lowercase (e.g. n) is a purely finite real scalar.
- Boldface lowercase (e.g. \mathbf{x}) is a purely finite real vector.
- Boldface uppercase (e.g. \mathbf{A}) is a purely finite real matrix.
- Italics lowercase with tilde (e.g. \tilde{c}) is a *gross*-scalar.
- Boldface lowercase with tilde (e.g. $\tilde{\mathbf{y}}$) is a *gross*-vector.

Let us introduce now the nonpreemptive *grossone*-based scheme following the lexicographic \mathbb{D} -based approach introduced in [29] and in a recent study [2]. It should be stressed that it is supposed hereinafter that the original problem (P1) has been stated using purely finite numbers only. This assumption is not restrictive from the practical point of view since all the LMOLP problems considered traditionally are of this kind. However, since we are now in the framework of \mathbb{D} -based numbers, this assumption should be explicitly stated.

To state the problem (P4), we reformulate (P3) by making use of *gross*-scalars and *gross*-vectors and is defined as follows:

$$\begin{aligned} \text{Max } & \tilde{\mathbf{c}}\mathbf{x} \\ \text{s.t. } & \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}, \end{aligned} \quad (\text{P4})$$

where $\tilde{\mathbf{c}}$ is a row-wise *gross*-vector having n *gross*-scalar components:

$$\tilde{\mathbf{c}} = \sum_{i=1}^r \mathbf{c}^i \mathbb{D}^{-i+1} \quad (3)$$

and $\tilde{\mathbf{c}}\mathbf{x}$ is the *gross*-scalar obtained by multiplying the *gross*-vector $\tilde{\mathbf{c}}$ by purely finite vector \mathbf{x}

$$\tilde{\mathbf{c}}\mathbf{x} = (\mathbf{c}^1 \cdot \mathbf{x})\mathbb{D}^0 + (\mathbf{c}^2 \cdot \mathbf{x})\mathbb{D}^{-1} + \dots + (\mathbf{c}^r \cdot \mathbf{x})\mathbb{D}^{-r+1}, \quad (4)$$

where (4) can be equivalently written in the extended form as

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

The fundamental difference between the definition of $\tilde{\mathbf{c}}$ in (3) with respect to the definition of $\bar{\mathbf{c}}$ in (2) is that $\tilde{\mathbf{c}}$ does not involve any unknown. More precisely, it does not require the specification of a real scalar value, like the value of M in (P3). However, this advantage leads to the fact that standard algorithms (like the simplex algorithm or the interior point algorithm), traditionally used for solving (P3), can no longer be used in this case since \mathbb{D} -based numbers are involved in the definition of the objective function. There will be shown in the next section that it is still possible to obtain optimality conditions and to introduce and implement a *gross-simplex* algorithm (a generalization of the traditional simplex algorithm to the case of \mathbb{D} -based numbers) able to solve problem (P4).

5 Theoretical results

In this section, first, it is proved that problems (P1) and (P4) are equivalent and then, second, optimality conditions are established. Before giving the equivalence theorem, the following three lemmas are required.

The first lemma states that all the optimal solutions of (P4) are vertices or belong to the convex hull of vertices, where the set of all the optimal solutions for a generic problem P is denoted, from hereafter, by the symbol $\Omega(\text{P})$.

Lemma 1. *Each $\mathbf{x}^* \in \Omega(\text{P4})$ is a vertex or lies on the convex hull of the optimal vertices.*

Proof. Since the objective function of problem (P4) is linear in \mathbf{x} , the associated level sets ($\tilde{\mathbf{c}}\mathbf{x} = \tilde{v}$) are hyper-planes. Thus the maximum, for a bounded and non-empty polytope S describing the domain of the problem must be on a vertex (or belong to the convex hull of the optimal vertices), for the same reasons for which the maximum of standard single-objective LP problems is located on a vertex or in the convex hull of the optimal vertices. In this case, similarly to standard LP, it follows that: (i) not all the vertices of (P4) are optimal, and (ii) not all the basic solutions are vertices. □

The next lemma states that all the optimal solutions of (P1) reach the same (*gross-scalar*) objective value for problem (P4).

Lemma 2. *For any $\mathbf{x}^* \in \Omega(\text{P1})$ it follows $\tilde{\mathbf{c}}\mathbf{x}^* = \tilde{v}$ (\tilde{v} being a constant *gross-scalar*).*

Proof. Let \mathbf{x}^* be a generic optimal solution belonging to $\Omega(\text{P1})$, then from (4) we have that the objective value of the problem (P4) associated to it is:

$$\tilde{\mathbf{c}}\mathbf{x}^* = (\mathbf{c}^1 \cdot \mathbf{x}^*)\mathbb{D}^0 + (\mathbf{c}^2 \cdot \mathbf{x}^*)\mathbb{D}^{-1} + \dots + (\mathbf{c}^r \cdot \mathbf{x}^*)\mathbb{D}^{-r+1}.$$

We can observe that:

$$\mathbf{c}^i \cdot \mathbf{x}^* = k^i \in \mathbb{R}, \quad i \in \{1, \dots, r\},$$

due to the fact that \mathbf{x}^* is optimal for (P1) and this problem has been formulated using purely finite numbers only. Thus we have

$$\tilde{\mathbf{c}}\mathbf{x}^* = k^1 \mathbb{1}^0 + k^2 \mathbb{1}^{-1} + \dots + k^r \mathbb{1}^{-r+1} = \tilde{v}.$$

□

The next lemma is the last step in preparing the proof of the equivalence between (P1) and (P4).

Lemma 3. *For all $\mathbf{x}^* \in \Omega(\text{P1})$ and any vertex $\hat{\mathbf{x}}$ of S such that $\hat{\mathbf{x}} \notin \Omega(\text{P1})$, it follows*

$$\tilde{\mathbf{c}}\hat{\mathbf{x}} < \tilde{\mathbf{c}}\mathbf{x}^*.$$

Proof. Let $\hat{\mathbf{x}}$ be a vertex of S which is not optimal for (P1). Thus, there exists an index $q \in \{1, \dots, r\}$ such that

$$\mathbf{c}^i \cdot \hat{\mathbf{x}} = \mathbf{c}^i \cdot \mathbf{x}^*, \quad \forall i \in \{1, \dots, q-1\},$$

but

$$\mathbf{c}^q \cdot \hat{\mathbf{x}} < \mathbf{c}^q \cdot \mathbf{x}^*.$$

This implies that

$$\tilde{\mathbf{c}}\mathbf{x}^* - \tilde{\mathbf{c}}\hat{\mathbf{x}} = \sum_{i=q}^r \mathbb{1}^{-i+1} (\mathbf{c}^i \cdot \mathbf{x}^* - \mathbf{c}^i \cdot \hat{\mathbf{x}}).$$

The expression above can be also expanded as follows

$$\begin{aligned} \tilde{\mathbf{c}}\mathbf{x}^* - \tilde{\mathbf{c}}\hat{\mathbf{x}} &= \mathbb{1}^{-q+1} (\mathbf{c}^q \cdot \mathbf{x}^* - \mathbf{c}^q \cdot \hat{\mathbf{x}}) + \mathbb{1}^{-(q+1)+1} (\mathbf{c}^{q+1} \cdot \mathbf{x}^* - \mathbf{c}^{q+1} \cdot \hat{\mathbf{x}}) + \dots \\ &\quad + \mathbb{1}^{-r+1} (\mathbf{c}^r \cdot \mathbf{x}^* - \mathbf{c}^r \cdot \hat{\mathbf{x}}). \end{aligned}$$

Since $\mathbb{1}^{-q+1} (\mathbf{c}^q \cdot \mathbf{x}^* - \mathbf{c}^q \cdot \hat{\mathbf{x}}) > 0$, it follows that $(\tilde{\mathbf{c}}\mathbf{x}^* - \tilde{\mathbf{c}}\hat{\mathbf{x}})$ is strictly positive too, since r is finite. Indeed, adding a finite number of infinitesimal contributions of orders of $\mathbb{1}$ higher than $-q+1$ will keep the sum strictly positive, even when these contributions are negative in sign, due to the property of *grossone*:

$$\begin{aligned} |\mathbb{1}^{-q+1} (\mathbf{c}^q \cdot \mathbf{x}^* - \mathbf{c}^q \cdot \hat{\mathbf{x}})| &> |\mathbb{1}^{-q} (\mathbf{c}^{q+1} \cdot \mathbf{x}^* - \mathbf{c}^{q+1} \cdot \hat{\mathbf{x}})| + \dots + \\ &\quad |\mathbb{1}^{-r+1} (\mathbf{c}^r \cdot \mathbf{x}^* - \mathbf{c}^r \cdot \hat{\mathbf{x}})|. \end{aligned}$$

□

We are now ready to prove that any solution to (P1) is also a solution to (P4), and vice-versa.

Theorem 1 (Equivalence). $\mathbf{x}^* \in \Omega(\text{P1})$ iff $\mathbf{x}^* \in \Omega(\text{P4})$.

Proof. \Rightarrow If \mathbf{x}^* is optimal for (P1), then $\tilde{\mathbf{c}}\mathbf{x}^* = \tilde{v}$ due to lemma 2. Therefore, according to lemma 1 and 3, \mathbf{x}^* is also optimal for (P4).

\Leftarrow If \mathbf{x}^* is optimal for (P4), then due to lemma 1, it belongs to the convex hull of some vertices. But from lemma 3 it follows that all the vertices of this kind are optimal solutions to problem (P1). \square

Remark 1 (Equivalence of problems (P3) and (P4)). Since we have proved that problems (P4) and (P1) are equivalent, and since in [36] it has been proved that problems (P3) and (P1) are equivalent, it follows that problems (P4) and (P3) are equivalent to each other as well.

6 The *gross*-simplex algorithm

The single-objective *gross*-LP problem formulated in (P4) using *grossone* can be solved using the *gross*-simplex algorithm here described, provided that the duality theory is extended to the case of *gross*-scalars and *gross*-vectors.

The dual problem of (P4) is the following

$$\begin{aligned} \text{Min } & \tilde{\mathbf{y}}\mathbf{b} \\ \text{s.t. } & \{ \tilde{\mathbf{y}} : \tilde{\mathbf{y}}\mathbf{A} \geq \tilde{\mathbf{c}} \}, \end{aligned} \tag{P5}$$

where $\tilde{\mathbf{y}} = [\tilde{y}_1, \dots, \tilde{y}_m]$ is a row-wise m -dimensional *gross*-vector, $\tilde{\mathbf{y}}\mathbf{A}$ an n -dimensional *gross*-vector, and $\tilde{\mathbf{y}}\mathbf{b}$ a *gross*-scalar.

The domain of (P5) is the *gross*-polytope \tilde{D} defined as:

$$\tilde{D} \equiv \{ \tilde{\mathbf{y}} : \tilde{\mathbf{y}}\mathbf{A} \geq \tilde{\mathbf{c}} \}.$$

In the following we will assume that \tilde{D} is bounded and non-empty, as we have assumed for S .

Since optimality conditions are the core of the simplex algorithm, we need to prove optimality conditions in our context.

Definition 1 (Definition of optimality for the *gross*-primal problem). A point \mathbf{x}^* is optimal for the problem (P4) iff $\tilde{\mathbf{c}}\mathbf{x}^* \geq \tilde{\mathbf{c}}\mathbf{x} \quad \forall \mathbf{x} \in S$.

Definition 2 (Definition of optimality for the *gross*-dual problem). A point $\tilde{\mathbf{y}}^*$ is optimal for the problem (P5) iff $\tilde{\mathbf{y}}^*\mathbf{b} \leq \tilde{\mathbf{y}}\mathbf{b} \quad \forall \tilde{\mathbf{y}} \in \tilde{D}$.

We need now the following lemma.

Lemma 4 (weak duality). $\forall \mathbf{x} \in S$ and $\forall \tilde{\mathbf{y}} \in \tilde{D}$, it follows that $\tilde{\mathbf{c}}\mathbf{x} \leq \tilde{\mathbf{y}}\mathbf{b}$.

Proof. Since $\mathbf{x} \in S$, $\mathbf{A}\mathbf{x} = \mathbf{b}$. Pre-multiplying both by $\tilde{\mathbf{y}}$, we have

$$\tilde{\mathbf{y}}\mathbf{A}\mathbf{x} = \tilde{\mathbf{y}}\mathbf{b}. \quad (5)$$

Now, since $\tilde{\mathbf{y}} \in \tilde{D}$, we know that $\tilde{\mathbf{y}}\mathbf{A} \geq \tilde{\mathbf{c}}$. By post-multiplying the latter by \mathbf{x} (being $\mathbf{x} \geq 0$), we have:

$$\tilde{\mathbf{y}}\mathbf{A}\mathbf{x} \geq \tilde{\mathbf{c}}\mathbf{x}.$$

By combining it with (5) we obtain $\tilde{\mathbf{y}}\mathbf{b} \geq \tilde{\mathbf{c}}\mathbf{x}$. □

Theorem 2 (Optimality condition). *If $\mathbf{x}^* \in S$, $\tilde{\mathbf{y}}^* \in \tilde{D}$ and $\tilde{\mathbf{c}}\mathbf{x}^* = \tilde{\mathbf{y}}^*\mathbf{b}$, then it follows that $\mathbf{x}^* \in \Omega(\text{P1})$ and $\tilde{\mathbf{y}}^* \in \Omega(\text{P5})$.*

Proof. It is an immediate consequence of the weak duality lemma 4. □

We are now ready to introduce the *gross-simplex* algorithm (see Algorithm 1) which exploits the theoretical results presented above. Notice that the algorithm needs a first feasible basis B . It can be found by solving the standard auxiliary problem analogously to the case where the objective function is only one (no *gross-simplex* required to solve it).

Remark 2. First, let us notice that on Step 2 of Algorithm 1 no divisions between *grossone*-based numerals are required. Second, it could be argued that, in order to limit numerical errors, the inversion of \mathbf{A}_B could be avoided, by using its LU decomposition, as in any other simplex algorithm implementation. However, in this work we do use the inverse, just because we want to make the presentation of the algorithm easier to read. Finally, please, notice that the unboundedness of the problem mentioned in Step 4 of Algorithm 1 cannot happen due to our initial assumptions.

Algorithm 1 The *gross*-simplex algorithm

Step 0. The user has to provide the initial set B of basic indices.

Step 1. Solve the system $\tilde{\mathbf{y}}\mathbf{A}_B = \tilde{\mathbf{c}}_B$ (where \mathbf{A}_B is the sub-matrix obtained by \mathbf{A} by considering the columns indexed by B).

This can be easily calculated as $\tilde{\mathbf{y}} = \mathbf{A}_B^{-T}\tilde{\mathbf{c}}_B$ where \mathbf{A}_B^{-T} is an abbreviation for $(\mathbf{A}_B^{-1})^T$, $\tilde{\mathbf{y}}$ is a *gross*-vector obtained by linearly combining the *gross*-vector $\tilde{\mathbf{c}}_B$ using the purely finite scalar elements in \mathbf{A}_B^{-T} .

Step 2. Compute $\tilde{\mathbf{s}} = \tilde{\mathbf{c}}_N - A_N^T\tilde{\mathbf{y}}$ (where N is the complementary set of B) and then select the maximum (gradient rule). When this maximum is negative (being it finite or infinitesimal), it means that we are done (the current solution is optimal) and thus the algorithm stops. Otherwise, the position of the maximum in the *gross*-vector $\tilde{\mathbf{s}}$ is the index k of the entering variable $N(k)$.

Step 3. Solve the system $\mathbf{A}_B\mathbf{d} = \mathbf{A}_{N(k)}$.

Step 4. Find the largest t such that $\mathbf{x}_B^* - t\mathbf{d} \geq 0$. If there is not such a t , then the problem is unbounded (STOP); otherwise, at least one component of $\mathbf{x}_B^* - t\mathbf{d}$ equals zero and the corresponding variable is the leaving variable. In case of ties, use the Lexicographic Pivoting Rule [3] to break them.

Step 5. Update B and N and return to Step 1.

7 Experimental Results

Example 1. The first example is a little variant to the two-dimensional problem with three objectives described in [37]:

$$\begin{aligned} \text{LexMax} \quad & 8x_1 + 12x_2, \quad 4x_1 + 10x_2, \quad x_1 + x_2 \\ \text{s.t.} \quad & 2x_1 + x_2 \leq 120 \\ & 2x_1 + 3x_2 \leq 210 \\ & 4x_1 + 3x_2 \leq 270 \\ & x_1 + 2x_2 \geq 60 \\ & x_1, x_2 \geq 0 \end{aligned}$$

The polygon S associated to this problem is shown in Figure 1. It can be seen that the first objective vector $\mathbf{c}^1 = [8, 12]$ is orthogonal to segment $[\alpha, \beta]$ ($\alpha = (0, 70)$, $\beta = (30, 50)$) shown in the same figure. Thus all the points laying on this segment are optimal. Since the solution is not unique, there

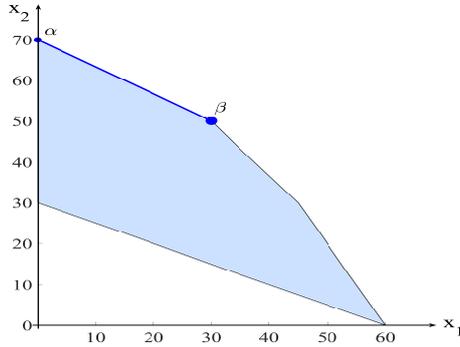


Figure 1: An example in two dimensions with three objectives. All the points in the segment $[\alpha, \beta]$ are optimal for the first objective, while point β is the unique lexicographic optimum for the given problem ($\beta = (30, 50)^T$).

is the chance to try to improve the second objective vector ($\mathbf{c}^2 = [14, 10]$) without deteriorating the first one function. The point that maximizes the second objective is β , associated to solution $\mathbf{x}^* = [30, 50]^T$. Since now the solution is unique, the third objective function can not be taken into account. Thus the lexicographic optimal solution for the problem is $\mathbf{x}^* = [30, 50]^T$.

Before running the *gross*-simplex algorithm we had to transform the problem into the following one, after converting the constraints into equality constraints by adding slack variables x_3, x_4, x_5 , and x_6 :

$$\begin{aligned}
 \text{LexMax} \quad & 8x_1 + 12x_2, \quad 4x_1 + 10x_2, \quad x_1 + x_2 \\
 \text{s.t.} \quad & 2x_1 + x_2 + x_3 = 120 \\
 & 2x_1 + 3x_2 + x_4 = 210 \\
 & 4x_1 + 3x_2 + x_5 = 270 \\
 & -x_1 - 2x_2 + x_6 = -60 \\
 & x_i \geq 0 \quad i = 1, \dots, 6
 \end{aligned} \tag{6}$$

The *gross*-simplex algorithm has been run on the problem (6), by using the initial basis $B = \{2, 3, 4, 5\}$ (N is therefore $\{1, 6\}$). The initial solution associated to the initial basis is: $\mathbf{x} = [0, 30, 90, 120, 180, 0]^T$, which corresponds to the point $(0, 30)$ in Figure 1. Then the algorithm computes $\tilde{\mathbf{s}}$ as $\tilde{\mathbf{c}}_N - A_N^T \tilde{\mathbf{y}}$ giving

$$\tilde{\mathbf{s}} = \begin{bmatrix} 2\mathbb{1}^0 + 9\mathbb{1}^{-1} + 0.5\mathbb{1}^{-2} \\ 6\mathbb{1}^0 + 5\mathbb{1}^{-1} + 0.5\mathbb{1}^{-2} \end{bmatrix}.$$

Thus, according to the gradient rule for choosing the entering variable, the one in N having the second index is selected, i.e., x_6 , which is the one associated to the maximum constraint violation: $6\mathbb{1}^0 + 5\mathbb{1}^{-1} + 0.5\mathbb{1}^{-2}$. The leaving index, computed according to the lexicographic pivoting rule is the third, i.e., variable x_4 . Thus the second base used is $B = \{2, 3, 6, 5\}$. The

gross-objective function is $\tilde{\mathbf{c}}\mathbf{x} = 360\mathbb{1}^0 + 300\mathbb{1}^{-1} + 30\mathbb{1}^{-2}$ and, of course, coincides with that of the dual $\tilde{\mathbf{y}}\mathbf{b}$ (when $\tilde{\mathbf{y}}$ will be also feasible, we will be at the optimum and the algorithm will end).

The solution associated to this second base is $\mathbf{x} = [0, 70, 50, 0, 60, 80]^T$, while the new value for $\tilde{\mathbf{s}}$ is

$$\tilde{\mathbf{s}} = \begin{bmatrix} 7.33\mathbb{1}^{-1} + 0.33\mathbb{1}^{-2} \\ -4\mathbb{1}^0 - 3.33\mathbb{1}^{-1} - 0.33\mathbb{1}^{-2} \end{bmatrix}.$$

Again, according to the gradient rule, the next entering index is the first, i.e., variable x_1 . It is interesting to note that in this case the constraint violation (a positive entry in $\tilde{\mathbf{s}}$) is not finite, as in previous case, but infinitesimal: $7.33\mathbb{1}^{-1} + 0.33\mathbb{1}^{-2}$. The leaving index is the fourth and thus the leaving variable is x_5 . Therefore, the next basis (the third) is $B = \{2, 3, 6, 1\}$. The *gross*-objective function now equals to $840\mathbb{1}^0 + 700\mathbb{1}^{-1} + 70\mathbb{1}^{-2}$.

The solution associated to the third base is $\mathbf{x} = [30, 50, 10, 0, 0, 70]^T$, while the new value for $\tilde{\mathbf{s}}$ is

$$\tilde{\mathbf{s}} = \begin{bmatrix} -3.67\mathbb{1}^{-1} - 0.17\mathbb{1}^{-2} \\ -4\mathbb{1}^0 + 0.33\mathbb{1}^{-1} - 0.17\mathbb{1}^{-2} \end{bmatrix}.$$

Since all the entries of $\tilde{\mathbf{s}}$ are negative (one is infinitesimal and one is finite), we are done. In fact, the solution found (once the slack variables are discarded), is the correct one, i.e., $\mathbf{x}^* = [30, 50]^T$. Furthermore, the *gross*-objective function at the end equals to $\tilde{\mathbf{c}}\mathbf{x} = \tilde{\mathbf{y}}\mathbf{b} = 840\mathbb{1}^0 + 920\mathbb{1}^{-1} + 80\mathbb{1}^{-2}$.

Example 2. This illustrative example is in three dimensions with three objectives:

$$\begin{aligned} &\text{LexMax } x_1, x_2, x_3 \\ &\text{s.t. } \{\mathbf{x} \in \mathbb{R}^3 : 0 \leq x_i \leq 1, i = 1, \dots, 3\} \end{aligned} \tag{7}$$

with the domain being the cube shown in Figure 2.

Let us understand first by a simple reasoning where the solution is located. It can be immediately seen that by considering the first objective alone (maximize x_1), the maximum is the face of the cube determined by the vertices $\alpha, \beta, \gamma, \delta$ (see Figure 2). Since the optimum is not unique, the second objective function can be considered in order to improve it without deteriorating the first objective. Then, all the points within the segment $[\beta, \gamma]$ are all optimal for the second objective. Again, the optimum is not unique, and thus we can consider the third objective, which allows us to select the point γ as the unique solution that maximizes all three objectives, i.e., the point γ is the lexicographic optimum to the given problem.

Before solving the problem, we had to change its form in order to convert the constraints from inequalities to equalities, by adding three slack variables x_4, x_5 and x_6 . The *gross*-simplex algorithm has been run on the problem

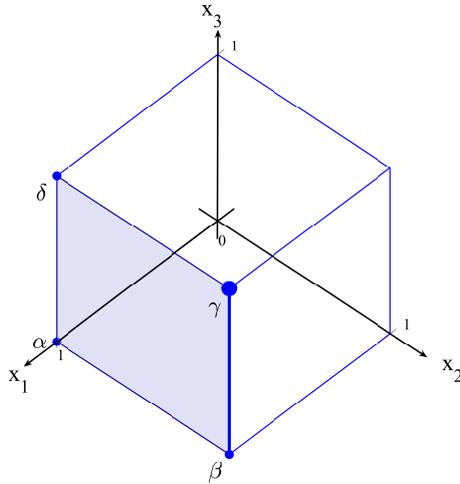


Figure 2: In the example (7) all the points belonging to the face of the cube determined by the vertices α , β , γ , δ are optimal according to the first objective (max of x_1). The segment $[\beta, \gamma]$ contains all the points that optimize the second objective (max of x_2) without deteriorating the first. Finally, the point γ is the lexicographic optimum ($\gamma=(1, 1, 1)$).

(7), by using the initial basis $B = \{4, 5, 6\}$ (N is therefore $\{1, 2, 3\}$). The corresponding initial solution is $\mathbf{x} = [0, 0, 0, 1, 1, 1]^T$, which corresponds to the origin of Figure 2. The first $\tilde{\mathbf{s}}$ gross-vector computed is

$$\tilde{\mathbf{s}} = \begin{bmatrix} 1\mathbb{1}^0 \\ 1\mathbb{1}^{-1} \\ 1\mathbb{1}^{-2} \end{bmatrix}$$

from which the algorithm has selected the first index, meaning that x_1 is the entering variable (according to the gradient rule). The leaving variable in this case is x_4 . Thus the second base is $B = \{1, 5, 6\}$.

The next solution (associated to the second base) is $\mathbf{x} = [1, 0, 0, 0, 1, 1]^T$ (point α of Figure 2) and the next $\tilde{\mathbf{s}}$ is:

$$\tilde{\mathbf{s}} = \begin{bmatrix} -1\mathbb{1}^0 \\ 1\mathbb{1}^{-1} \\ 1\mathbb{1}^{-2} \end{bmatrix}.$$

This time the most positive entry is the second, and thus the entering variable is x_2 and the leaving one is x_5 . The next base is $\{1, 2, 6\}$ and the new candidate solution is $\mathbf{x} = [1, 1, 0, 0, 0, 1]^T$ (point β of Figure 2), while the next $\tilde{\mathbf{s}}$ is:

$$\tilde{\mathbf{s}} = \begin{bmatrix} -1\mathbb{1}^0 \\ -1\mathbb{1}^{-1} \\ 1\mathbb{1}^{-2} \end{bmatrix}.$$

Algorithm 2 Generation of a randomly rotated hypercube in n dimensions

Step 1. Let $\mathbf{Ax} \leq \mathbf{b}$ be the initial, unrotated hypercube, in n -dimensions.
Let \mathbf{C} be the matrix containing the objective vectors, row by row:

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}^1 \\ \mathbf{c}^2 \\ \dots \\ \mathbf{c}^n \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Step 2. Generate a random rotation matrix \mathbf{Q} . It can be computed using a QR factorization utility, applied to a random matrix \mathbf{T} . In particular, \mathbf{T} must be an n -by- n matrix having entries randomly generated according to the normal distribution (zero mean and unitary variance). In Matlab the matrix \mathbf{Q} can be obtained in this way:

$$\begin{aligned} \mathbf{T} &= \text{randn}(n); \\ [\mathbf{Q}, \mathbf{R}] &= \text{qr}(\mathbf{T}); \end{aligned}$$

Step 3. Rotate the polytope: $\mathbf{A}' = \mathbf{AQ}$ (\mathbf{b} does not change under rotations: $\mathbf{b}' = \mathbf{b}$) $\mathbf{C}' = \mathbf{CQ}$ ($\mathbf{C}' = \mathbf{Q}$, since \mathbf{C} is an identity matrix);

Step 4. Transform the polytope from the form $\mathbf{A}'\mathbf{x}' \leq \mathbf{b}'$ to the form $\{\mathbf{A}''\mathbf{x}'' = \mathbf{b}'', \mathbf{x}'' \geq 0\}$

Step 5. Compute the lexicographic optimum. It can be obtained starting from the lexicographic optimum of the unrotated problem (which is known to be $\mathbf{x}^* = [1, 1, \dots, 1]^T$, a vector of n ones), multiplied by the same rotation matrix: $\mathbf{x}'^* = \mathbf{Q}^T \mathbf{x}^*$.

In this situation, there exists still one positive component, meaning that there is the chance to improve the objective function. The entering variable is x_3 , while the leaving one is x_6 . Eventually we get $\mathbf{x} = [1, 1, 1, 0, 0, 0]^T$, which is the optimal one, since now $\tilde{\mathbf{s}}$ has no positive entries:

$$\tilde{\mathbf{s}} = \begin{bmatrix} -1\mathbb{1}^0 \\ -1\mathbb{1}^{-1} \\ -1\mathbb{1}^{-2} \end{bmatrix}.$$

The final solution corresponds to the point γ of Figure 2 and the final *gross*-objective function value is $1\mathbb{1}^0 + 1\mathbb{1}^{-1} + 1\mathbb{1}^{-2}$.

Example 3a. This example is a generalization of previous example to the case of five objectives in \mathbb{R}^5 . To make the problem less obvious, a random rotation to the hypercube has been added. The hypercube considered is now centered at the origin, to better handle its rotation

$$S \equiv \{x \in \mathbb{R}^5 : -1 \leq x_i \leq 1, i = 1, \dots, 5\}.$$

The problem consists of the lexicographic optimization of x_1, x_2, \dots, x_5 , but the objectives will be affected by the rotation, as well, making the objective functions more appealing. The method applied to generate a randomly rotated hypercube and the corresponding objective vectors is shown above as Algorithm 2.

The Algorithm 2 has been applied: the generated rotation matrix \mathbf{Q} is reported in Appendix.

Using this matrix, the following problem $(\mathbf{A}', \mathbf{b}', \mathbf{C}')$ has been generated:

$$\begin{aligned} \text{LexMax} \quad & -0.37x_1 - 0.03x_2 + 0.41x_3 - 0.83x_4 - 0.08x_5, \\ & -0.48x_1 - 0.77x_2 + 0.23x_3 + 0.36x_4 - 0.04x_5, \\ & +0.34x_1 - 0.38x_2 - 0.33x_3 - 0.23x_4 - 0.76x_5, \\ & +0.43x_1 + 0.10x_2 + 0.82x_3 + 0.24x_4 - 0.28x_5, \\ & +0.58x_1 - 0.51x_2 + 0.04x_3 - 0.27x_4 + 0.58x_5 \\ \text{s.t.} \quad & \{\mathbf{x}' \in \mathbb{R}^5 : \mathbf{A}'\mathbf{x}' \leq \mathbf{b}'\} \end{aligned} \tag{8}$$

where \mathbf{A}' and vector \mathbf{b}' are reported in Appendix as well (\mathbf{C}' is not reported, being it equal to \mathbf{Q}).

The lexicographic optimum for this problem is:

$$\mathbf{x}'^* = \mathbf{Q}^T \mathbf{x}^* = [0.50, -1.58, 1.16, -0.74, -0.59]^T.$$

Then the problem has been transformed in the form $\{\mathbf{A}''\mathbf{x}'' = \mathbf{b}'', \mathbf{x}'' \geq 0\}$ and the *gross*-simplex algorithm has been run. After 13 steps, the algorithm has found the correct lexicographic optimum.

The list of the visited vertices, along with the objective function, is shown in Table 1.

Table 1: Iterations performed by the *gross*-simplex algorithm while solving problem (8) with five objectives.

Iter.	\mathbf{x}'	<i>gross</i> -objective function $\tilde{\mathbf{c}}\mathbf{x}'$
1	$[-0.50, 1.58, -1.16, 0.74, 0.59]^T$	$-1\mathbb{1}^0 - 1\mathbb{1}^{-1} - 1\mathbb{1}^{-2} - 1\mathbb{1}^{-3} - 1\mathbb{1}^{-4}$
2	$[-0.83, 1.56, -0.80, 0.00, 0.52]^T$	$-0.11\mathbb{1}^0 - 1\mathbb{1}^{-1} - 1\mathbb{1}^{-2} - 1\mathbb{1}^{-3} - 1\mathbb{1}^{-4}$
3	$[-1.23, 1.52, -0.35, -0.93, 0.43]^T$	$1\mathbb{1}^0 - 1\mathbb{1}^{-1} - 1\mathbb{1}^{-2} - 1\mathbb{1}^{-3} - 1\mathbb{1}^{-4}$
4	$[-1.96, 0.36, 0.00, -0.39, 0.37]^T$	$1\mathbb{1}^0 + 0.52\mathbb{1}^{-1} - 1\mathbb{1}^{-2} - 1\mathbb{1}^{-3} - 1\mathbb{1}^{-4}$
5	$[-2.19, 0.00, 0.11, -0.22, 0.35]^T$	$1\mathbb{1}^0 + 0.98\mathbb{1}^{-1} - 1\mathbb{1}^{-2} - 1\mathbb{1}^{-3} - 1\mathbb{1}^{-4}$
6	$[-2.20, -0.01, 0.11, -0.22, 0.35]^T$	$1\mathbb{1}^0 + 1\mathbb{1}^{-1} - 1\mathbb{1}^{-2} - 1\mathbb{1}^{-3} - 1\mathbb{1}^{-4}$
7	$[-2.08, -0.14, 0.00, -0.29, 0.10]^T$	$1\mathbb{1}^0 + 1\mathbb{1}^{-1} - 0.67\mathbb{1}^{-2} - 1\mathbb{1}^{-3} - 1\mathbb{1}^{-4}$
8	$[-2.04, -0.19, -0.04, -0.32, 0.00]^T$	$1\mathbb{1}^0 + 1\mathbb{1}^{-1} - 0.54\mathbb{1}^{-2} - 1\mathbb{1}^{-3} - 1\mathbb{1}^{-4}$
9	$[-1.51, -0.77, -0.55, -0.67, -1.17]^T$	$1\mathbb{1}^0 + 1\mathbb{1}^{-1} + 1\mathbb{1}^{-2} - 1\mathbb{1}^{-3} - 1\mathbb{1}^{-4}$
10	$[-1.22, -0.70, 0.00, -0.51, -1.36]^T$	$1\mathbb{1}^0 + 1\mathbb{1}^{-1} + 1\mathbb{1}^{-2} - 0.33\mathbb{1}^{-3} - 1\mathbb{1}^{-4}$
11	$[-0.66, -0.57, 1.09, -0.19, -1.74]^T$	$1\mathbb{1}^0 + 1\mathbb{1}^{-1} + 1\mathbb{1}^{-2} + 1\mathbb{1}^{-3} - 1\mathbb{1}^{-4}$
12	$[0.00, -1.14, 1.13, -0.50, -1.09]^T$	$1\mathbb{1}^0 + 1\mathbb{1}^{-1} + 1\mathbb{1}^{-2} + 1\mathbb{1}^{-3} + 0.13\mathbb{1}^{-4}$
13	$[0.50, -1.58, 1.16, -0.74, -0.59]^T$	$1\mathbb{1}^0 + 1\mathbb{1}^{-1} + 1\mathbb{1}^{-2} + 1\mathbb{1}^{-3} + 1\mathbb{1}^{-4}$

Example 3b. This example considers ten objectives in \mathbb{R}^{10} and applies the rotations described above in order to construct more interesting problems. The corresponding rotation matrix \mathbf{Q} , matrix \mathbf{A}' , and vector \mathbf{b}' are reported in Appendix. The resulting problem (9) can be written as follows:

$$\begin{aligned} \text{LexMax } & \mathbf{c}'^1 \cdot \mathbf{x}', \mathbf{c}'^2 \cdot \mathbf{x}', \dots, \mathbf{c}'^{10} \cdot \mathbf{x}', \\ \text{s.t. } & \{ \mathbf{x}' \in \mathbb{R}^{10} : \mathbf{A}'\mathbf{x}' \leq \mathbf{b}' \}. \end{aligned} \quad (9)$$

where \mathbf{c}'^1 is the first row of \mathbf{C}' , \mathbf{c}'^2 is its second row, and so on (\mathbf{C}' being equal to \mathbf{Q}).

The lexicographic optimum for this problem is:

$$\mathbf{x}'^* = [-1.62, 1.27, 1.57, -1.05, -0.40, -0.57, -0.55, -0.61, -0.99, 0.20]^T.$$

The *gross*-simplex algorithm has been applied and the optimum has been obtained after 29 iterations, as shown in the following Table 2.

Table 2: Iterations performed by *gross*-simplex algorithm for problem (9).

Iter.	\mathbf{x}' (<i>gross</i> -objective function $\tilde{\mathbf{c}}\mathbf{x}'$)
1	$[1.62, -1.27, -1.57, 1.05, 0.40, 0.57, 0.55, 0.61, 0.99, -0.20]^T$, ($-1\mathbb{0}^0 - 1\mathbb{0}^{-1} - 1\mathbb{0}^{-2} - 1\mathbb{0}^{-3} - 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
2	$[0.17, -2.04, -1.04, 1.49, 0.44, 1.24, 0.43, 0.08, 0.76, -0.04]^T$, ($1\mathbb{0}^0 - 1\mathbb{0}^{-1} - 1\mathbb{0}^{-2} - 1\mathbb{0}^{-3} - 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
3	$[0.20, -2.10, -1.00, 1.42, 0.35, 1.17, 0.45, 0.00, 0.65, -0.09]^T$, ($1\mathbb{0}^0 - 0.8\mathbb{0}^{-1} - 1\mathbb{0}^{-2} - 1\mathbb{0}^{-3} - 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
4	$[0.31, -2.37, -0.87, 1.13, 0.00, 0.87, 0.52, -0.32, 0.23, -0.29]^T$, ($1\mathbb{0}^0 + 0.1\mathbb{0}^{-1} - 1\mathbb{0}^{-2} - 1\mathbb{0}^{-3} - 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
5	$[0.36, -2.52, -0.80, 0.97, -0.20, 0.70, 0.56, -0.50, 0.00, -0.40]^T$, ($1\mathbb{0}^0 + 0.5\mathbb{0}^{-1} - 1\mathbb{0}^{-2} - 1\mathbb{0}^{-3} - 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
6	$[0.42, -2.66, -0.73, 0.82, -0.39, 0.54, 0.59, -0.67, -0.23, -0.51]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} - 1\mathbb{0}^{-2} - 1\mathbb{0}^{-3} - 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
7	$[0.43, -1.88, -0.87, 0.44, 0.00, 0.95, 0.53, -1.70, -0.30, -0.63]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 0.5\mathbb{0}^{-2} - 1\mathbb{0}^{-3} - 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
8	$[0.43, -1.60, -0.92, 0.31, 0.14, 1.10, 0.51, -2.07, -0.32, -0.67]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} - 1\mathbb{0}^{-3} - 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
9	$[0.35, -1.63, -0.92, 0.00, 0.17, 1.13, 0.44, -1.97, -0.31, -0.51]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} - 0.6\mathbb{0}^{-3} - 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
10	$[0.10, -1.74, -0.94, -0.99, 0.26, 1.20, 0.21, -1.67, -0.29, 0.00]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 0.6\mathbb{0}^{-3} - 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
11	$[0.02, -1.78, -0.95, -1.32, 0.29, 1.23, 0.13, -1.57, -0.28, 0.17]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} - 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
12	$[0.00, -1.77, -0.95, -1.33, 0.29, 1.22, 0.14, -1.56, -0.27, 0.15]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} - 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
13	$[-0.10, -1.71, -0.94, -1.38, 0.30, 1.19, 0.17, -1.49, -0.26, 0.00]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} - 0.8\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
14	$[-0.96, -1.22, -0.90, -1.77, 0.32, 0.96, 0.45, -0.96, -0.12, -1.23]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} - 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
15	$[-1.33, -0.84, -0.82, -1.50, 1.19, 0.00, 0.85, -0.89, -0.91, -0.66]^T$, ($1.0\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 0.8\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
16	$[-1.38, -0.79, -0.81, -1.47, 1.30, -0.12, 0.90, -0.88, -1.02, -0.59]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} - 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
17	$[-1.64, -0.51, -0.23, -1.47, 0.86, -0.88, 0.94, -1.33, 0.00, -0.20]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 0.6\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
18	$[-1.70, -0.44, -0.10, -1.47, 0.76, -1.06, 0.95, -1.43, 0.24, -0.11]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 1\mathbb{0}^{-6} - 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
19	$[-1.68, -0.40, 0.00, -1.47, 0.75, -1.02, 0.99, -1.40, 0.21, -0.10]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 1\mathbb{0}^{-6} - 0.9\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
20	$[-1.47, -0.09, 0.81, -1.52, 0.62, -0.64, 1.34, -1.16, 0.01, 0.00]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 1\mathbb{0}^{-6} + 0.2\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
21	$[-1.46, -0.08, 0.83, -1.52, 0.62, -0.62, 1.35, -1.15, 0.00, 0.00]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 1\mathbb{0}^{-6} + 0.3\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
22	$[-1.41, 0.00, 1.02, -1.53, 0.59, -0.53, 1.43, -1.10, -0.05, 0.03]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 1\mathbb{0}^{-6} + 0.5\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
23	$[-1.31, 0.14, 1.38, -1.55, 0.53, -0.36, 1.59, -0.99, -0.14, 0.07]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 1\mathbb{0}^{-6} + 1\mathbb{0}^{-7} - 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
24	$[-1.28, 0.18, 1.53, -1.52, 0.60, -0.42, 1.24, -0.96, -0.19, 0.00]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 1\mathbb{0}^{-6} + 1\mathbb{0}^{-7} - 0.6\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
25	$[-1.14, 0.32, 2.08, -1.40, 0.84, -0.64, 0.00, -0.86, -0.37, -0.26]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 1\mathbb{0}^{-6} + 1\mathbb{0}^{-7} + 0.9\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
26	$[-1.13, 0.33, 2.13, -1.38, 0.87, -0.66, -0.12, -0.85, -0.39, -0.29]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 1\mathbb{0}^{-6} + 1\mathbb{0}^{-7} + 1\mathbb{0}^{-8} - 1\mathbb{0}^{-9}$)
27	$[-1.42, 0.88, 1.80, -1.19, 0.12, -0.61, -0.37, -0.71, -0.74, 0.00]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 1\mathbb{0}^{-6} + 1\mathbb{0}^{-7} + 1\mathbb{0}^{-8} + 0.2\mathbb{0}^{-9}$)
28	$[-1.47, 0.97, 1.75, -1.16, 0.00, -0.60, -0.41, -0.69, -0.80, 0.05]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 1\mathbb{0}^{-6} + 1\mathbb{0}^{-7} + 1\mathbb{0}^{-8} + 0.4\mathbb{0}^{-9}$)
29	$[-1.62, 1.27, 1.57, -1.05, -0.40, -0.57, -0.55, -0.61, -0.99, 0.20]^T$, ($1\mathbb{0}^0 + 1\mathbb{0}^{-1} + 1\mathbb{0}^{-2} + 1\mathbb{0}^{-3} + 1\mathbb{0}^{-4} + 1\mathbb{0}^{-5} + 1\mathbb{0}^{-6} + 1\mathbb{0}^{-7} + 1\mathbb{0}^{-8} + 1\mathbb{0}^{-9}$)

Acknowledgement. The research of Ya.D. Sergeyev was supported by the Russian Science Foundation, project No 15-11-30022 “Global optimization, supercomputing computations, and applications”. The authors are grateful to the anonymous reviewers for their insightful comments and suggestions.

Appendix

This appendix reports information related to Example 3a and 3b. In order to construct the problem (8) of experiment 3a (the one with 5 dimensions) the following rotation matrix \mathbf{Q} , matrix \mathbf{A}' , and vector \mathbf{b}' have been used:

$$\mathbf{Q} = \begin{bmatrix} -0.37, & -0.03, & 0.41, & -0.83, & -0.08 \\ -0.48, & -0.77, & 0.23, & 0.36, & -0.04 \\ 0.34, & -0.38, & -0.33, & -0.23, & -0.76 \\ 0.43, & 0.10, & 0.82, & 0.24, & -0.28 \\ 0.58, & -0.51, & 0.04, & -0.27, & 0.58 \end{bmatrix},$$

$$\mathbf{A}' = \begin{bmatrix} -0.37, & -0.03, & 0.41, & -0.83, & -0.08 \\ -0.48, & -0.77, & 0.23, & 0.36, & -0.04 \\ 0.34, & -0.38, & -0.33, & -0.23, & -0.76 \\ 0.43, & 0.10, & 0.82, & 0.24, & -0.28 \\ 0.58, & -0.51, & 0.04, & -0.27, & 0.58 \\ 0.37, & 0.03, & -0.41, & 0.83, & 0.08 \\ 0.48, & 0.77, & -0.23, & -0.36, & 0.04 \\ -0.34, & 0.38, & 0.33, & 0.23, & 0.76 \\ -0.43, & -0.10, & -0.82, & -0.24, & 0.28 \\ -0.58, & 0.51, & -0.04, & 0.27, & -0.58 \end{bmatrix},$$

$$\mathbf{b}' = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]^T.$$

The following matrices and vector are the ones related to problem (9) of experiment 3b (the one with 10 dimensions):

$$\mathbf{Q} = \begin{bmatrix} -0.72, & -0.38, & 0.27, & 0.22, & 0.02, & 0.34, & -0.06, & -0.26, & -0.11, & 0.08 \\ 0.12, & -0.31, & 0.15, & -0.34, & -0.42, & -0.35, & 0.08, & -0.38, & -0.49, & -0.24 \\ 0.00, & 0.53, & -0.09, & -0.26, & 0.26, & 0.28, & -0.04, & -0.70, & -0.05, & -0.08 \\ -0.21, & -0.09, & -0.02, & -0.81, & 0.08, & 0.06, & -0.19, & 0.25, & 0.02, & 0.42 \\ -0.49, & 0.28, & 0.02, & -0.22, & 0.01, & -0.14, & 0.16, & 0.30, & 0.08, & -0.70 \\ -0.21, & 0.21, & 0.05, & 0.15, & 0.49, & -0.54, & 0.23, & 0.04, & -0.45, & 0.32 \\ -0.16, & 0.18, & 0.36, & 0.00, & -0.27, & -0.47, & 0.03, & -0.28, & 0.63, & 0.24 \\ 0.20, & 0.29, & 0.74, & -0.04, & -0.12, & 0.35, & 0.32, & 0.22, & -0.19, & 0.09 \\ 0.09, & 0.10, & 0.38, & 0.08, & 0.17, & -0.15, & -0.85, & 0.07, & -0.12, & -0.18 \\ -0.25, & 0.47, & -0.28, & 0.17, & -0.63, & 0.05, & -0.21, & 0.12, & -0.30, & 0.25 \end{bmatrix},$$

$$\mathbf{A}' = \begin{bmatrix} -0.72, & -0.38, & 0.27, & 0.22, & 0.02, & 0.34, & -0.06, & -0.26, & -0.11, & 0.08 \\ 0.12, & -0.31, & 0.15, & -0.34, & -0.42, & -0.35, & 0.08, & -0.38, & -0.49, & -0.24 \\ 0.00, & 0.53, & -0.09, & -0.26, & 0.26, & 0.28, & -0.04, & -0.70, & -0.05, & -0.08 \\ -0.21, & -0.09, & -0.02, & -0.81, & 0.08, & 0.06, & -0.19, & 0.25, & 0.02, & 0.42 \\ -0.49, & 0.28, & 0.02, & -0.22, & 0.01, & -0.14, & 0.16, & 0.30, & 0.08, & -0.70 \\ -0.21, & 0.21, & 0.05, & 0.15, & 0.49, & -0.54, & 0.23, & 0.04, & -0.45, & 0.32 \\ -0.16, & 0.18, & 0.36, & 0.00, & -0.27, & -0.47, & 0.03, & -0.28, & 0.63, & 0.24 \\ 0.20, & 0.29, & 0.74, & -0.04, & -0.12, & 0.35, & 0.32, & 0.22, & -0.19, & 0.09 \\ 0.09, & 0.10, & 0.38, & 0.08, & 0.17, & -0.15, & -0.85, & 0.07, & -0.12, & -0.18 \\ -0.25, & 0.47, & -0.28, & 0.17, & -0.63, & 0.05, & -0.21, & 0.12, & -0.30, & 0.25 \\ 0.72, & 0.38, & -0.27, & -0.22, & -0.02, & -0.34, & 0.06, & 0.26, & 0.11, & -0.08 \\ -0.12, & 0.31, & -0.15, & 0.34, & 0.42, & 0.35, & -0.08, & 0.38, & 0.49, & 0.24 \\ -0.00, & -0.53, & 0.09, & 0.26, & -0.26, & -0.28, & 0.04, & 0.70, & 0.05, & 0.08 \\ 0.21, & 0.09, & 0.02, & 0.81, & -0.08, & -0.06, & 0.19, & -0.25, & -0.02, & -0.42 \\ 0.49, & -0.28, & -0.02, & 0.22, & -0.01, & 0.14, & -0.16, & -0.30, & -0.08, & 0.70 \\ 0.21, & -0.21, & -0.05, & -0.15, & -0.49, & 0.54, & -0.23, & -0.04, & 0.45, & -0.32 \\ 0.16, & -0.18, & -0.36, & -0.00, & 0.27, & 0.47, & -0.03, & 0.28, & -0.63, & -0.24 \\ -0.20, & -0.29, & -0.74, & 0.04, & 0.12, & -0.35, & -0.32, & -0.22, & 0.19, & -0.09 \\ -0.09, & -0.10, & -0.38, & -0.08, & -0.17, & 0.15, & 0.85, & -0.07, & 0.12, & 0.18 \\ 0.25, & -0.47, & 0.28, & -0.17, & 0.63, & -0.05, & 0.21, & -0.12, & 0.30, & -0.25 \end{bmatrix},$$

$$\mathbf{b}' = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]^T.$$

References

- [1] 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*, 2017. in Press.
- [2] 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.
- [3] 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.
- [4] L. D’Alotto. Cellular automata using infinite computations. *Applied Mathematics and Computation*, 218(16):8077–8082, 2012.
- [5] L. D’Alotto. A classification of one-dimensional cellular automata using infinite computations. *Applied Mathematics and Computation*, 255:15–24, 2015.
- [6] R. De Leone. Nonlinear programming and grossone: Quadratic programming and the role of constraint qualifications. *Applied Mathematics and Computation*, published online 15 May 2017.

- [7] H. Isermann. Linear lexicographic optimization. *OR Spektrum*, 4:223–228, 1982.
- [8] D.I. Iudin, Ya.D. Sergeyev, and M. Hayakawa. Interpretation of percolation in terms of infinity computations. *Applied Mathematics and Computation*, 218(16):8099–8111, 2012.
- [9] 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.
- [10] G. Lolli. Metamathematical investigations on the theory of grossone. *Applied Mathematics and Computation*, 255:3–14, 2015.
- [11] M. Margenstern. An application of grossone to the study of a family of tilings of the hyperbolic plane. *Applied Mathematics and Computation*, 218(16):8005–8018, 2012.
- [12] M. Margenstern. Fibonacci words, hyperbolic tilings and grossone. *Communications in Nonlinear Science and Numerical Simulation*, 21(1–3):3–11, 2015.
- [13] F. Mazzia, Ya.D. Sergeyev, F. Iavernaro, P. Amodio, and M.S. Mukhametzhanov. Numerical methods for solving ODEs on the Infinity Computer. 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 090033. AIP Publishing, New York, 2016.
- [14] L. Pourkarimi and M. Zarepisheh. A dual-based algorithm for solving lexicographic multiple objective programs. *European Journal of Operational Research*, 176:1348–1356, 2007.
- [15] D. Rizza. Supertasks and numeral systems. 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 090005. AIP Publishing, New York, 2016.
- [16] A. Robinson. *Non-standard Analysis*. Princeton Univ. Press, Princeton, 1996.
- [17] Ya.D. Sergeyev. *Arithmetic of Infinity*. Edizioni Orizzonti Meridionali, CS, 2003, 2nd ed. 2013.
- [18] Ya.D. Sergeyev. Blinking fractals and their quantitative analysis using infinite and infinitesimal numbers. *Chaos, Solitons & Fractals*, 33(1):50–75, 2007.

- [19] Ya.D. Sergeyev. A new applied approach for executing computations with infinite and infinitesimal quantities. *Informatica*, 19(4):567–596, 2008.
- [20] Ya.D. Sergeyev. Evaluating the exact infinitesimal values of area of Sierpinski’s carpet and volume of Menger’s sponge. *Chaos, Solitons & Fractals*, 42(5):3042–3046, 2009.
- [21] Ya.D. Sergeyev. Numerical point of view on Calculus for functions assuming finite, infinite, and infinitesimal values over finite, infinite, and infinitesimal domains. *Nonlinear Analysis Series A: Theory, Methods & Applications*, 71(12):e1688–e1707, 2009.
- [22] Ya.D. Sergeyev. Counting systems and the First Hilbert problem. *Nonlinear Analysis Series A: Theory, Methods & Applications*, 72(3-4):1701–1708, 2010.
- [23] Ya.D. Sergeyev. Lagrange Lecture: Methodology of numerical computations with infinities and infinitesimals. *Rendiconti del Seminario Matematico dell’Università e del Politecnico di Torino*, 68(2):95–113, 2010.
- [24] Ya.D. Sergeyev. Higher order numerical differentiation on the Infinity Computer. *Optimization Letters*, 5(4):575–585, 2011.
- [25] Ya.D. Sergeyev. On accuracy of mathematical languages used to deal with the Riemann zeta function and the Dirichlet eta function. *p-Adic Numbers, Ultrametric Analysis and Applications*, 3(2):129–148, 2011.
- [26] Ya.D. Sergeyev. Using blinking fractals for mathematical modelling of processes of growth in biological systems. *Informatica*, 22(4):559–576, 2011.
- [27] Ya.D. Sergeyev. Solving ordinary differential equations by working with infinitesimals numerically on the Infinity Computer. *Applied Mathematics and Computation*, 219(22):10668–10681, 2013.
- [28] Ya.D. Sergeyev. Computations with grossone-based infinities. In Calude C.S. and Dinneen M.J., editors, *Unconventional Computation and Natural Computation: Proc. of the 14th International Conference UCNC 2015*, volume LNCS 9252, pages 89–106. Springer, New York, 2015.
- [29] Ya.D. Sergeyev. The olympic medals ranks, lexicographic ordering, and numerical infinities. *The Mathematical Intelligencer*, 37(2):4–8, 2015.
- [30] Ya.D. Sergeyev. Un semplice modo per trattare le grandezze infinite ed infinitesime. *Matematica nella Società e nella Cultura: Rivista della Unione Matematica Italiana*, 8(1):111–147, 2015.

- [31] 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.
- [32] Ya.D. Sergeyev. Numerical infinities and infinitesimals: Methodology, applications, and repercussions on two Hilbert problems. *EMS Surveys in Mathematical Sciences*, to appear.
- [33] Ya.D. Sergeyev and A. Garro. Observability of Turing machines: A refinement of the theory of computation. *Informatica*, 21(3):425–454, 2010.
- [34] 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.
- [35] Ya.D. Sergeyev, M.S. Mukhametzhanov, F. Mazzia, F. Iavernaro, and P. Amodio. Numerical methods for solving initial value problems on the Infinity Computer. *International Journal of Unconventional Computing*, 12(1):3–23, 2016.
- [36] H.D. Sherali and A.L. Soyster. Preemptive and nonpreemptive multi-objective programming: Relationship and counterexamples. *Journal of Optimization Theory and Applications*, 39(2):173–186, 1983.
- [37] I.P. Stanimirovic. Compendious lexicographic method for multi-objective optimization. *Facta universitatis - series: Mathematics and Informatics*, 27(1):55–66, 2012.
- [38] M.C. Vita, S. De Bartolo, C. Fallico, and M. Veltri. Usage of infinitesimals in the Menger’s Sponge model of porosity. *Applied Mathematics and Computation*, 218(16):8187–8196, 2012.
- [39] A.A. Zhigljavsky. Computing sums of conditionally convergent and divergent series using the concept of grossone. *Applied Mathematics and Computation*, 218(16):8064–8076, 2012.
- [40] A. Žilinskas. On strong homogeneity of two global optimization algorithms based on statistical models of multimodal objective functions. *Applied Mathematics and Computation*, 218(16):8131–8136, 2012.