Towards Lexicographic Multi-Objective Linear Programming using Grossone Methodology

Marco Cococcioni[1, a)], Massimo Pappalardo[1, b)] and Yaroslav D. Sergeyev[2, 3, c)]

[1]*University of Pisa, Pisa, Italy*
[2]*University of Calabria, Rende (CS), Italy*
[3]*Lobachevsky State University of Nizhni Novgorod, Russia*

[a)]Corresponding author: marco.cococcioni@unipi.it
[b)]massimo.pappalardo@unipi.it
[c)]yaro@dimes.unical.it

Abstract. Lexicographic Multi-Objective Linear Programming (LMOLP) problems can be solved in two ways: preemptive and nonpreemptive. The preemptive approach requires the solution of a series of LP problems, with changing constraints (each time the next objective is added, a new constraint appears). The nonpreemptive approach is based on a scalarization of the multiple objectives into a single-objective linear function by a weighted combination of the given objectives. It requires the specification of a set of weights, which is not straightforward and can be time consuming. In this work we present both mathematical and software ingredients necessary to solve LMOLP problems using a recently introduced computational methodology (allowing one to work numerically with infinities and infinitesimals) based on the concept of *grossone.* The ultimate goal of such an attempt is an implementation of a simplex-like algorithm, able to solve the original LMOLP problem by solving only one single-objective problem and without the need to specify finite weights. The expected advantages are therefore obvious.

INTRODUCTION

A new approach allowing one to work numerically with infinities and infinitesimals has been recently proposed in [1,2] by introducing the new infinite number *grossone,* being the number of elements of the set of natural numbers. The properties of *grossone* expressed by the symbol ① were studied in these last years and a lot of demonstrations of the usefulness of *grossone* in several fields has already been provided in literature [3-10].

In this paper, we want to analyze multi-objective linear optimization problems (MOLP) using *grossone* and to show that this approach can be particularly fruitful if we study MOLP with the lexicographic order (LMOLP).

Traditionally LMOLP [11-14] are solved in two different ways [11]. One consists of solving a sequence of single-objective LP problems (*preemptive* approach), where new constraints are added to the subsequent LP problem once an optimal solution to the previous problem has been found (this approach is, of course, time consuming).

Another approach (known as *nonpreemptive* approach [11]) 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 with the original problem. In practice, determining such weights is not a trivial task and it can be time consuming.

Our idea consists of transforming LMOLP into a single-objective LP problem by multiplying the most important objective by one, the second by $①^{-1}$, the third by $①^{-2}$, etc., i.e., by infinitesimal coefficients.

After this transformation, the single-objective LP problem (using *grossone*-based numbers) can be solved only once. This new problem can be solved by using a simplex-like method, generalized to the case of *grossone*-based numbers and can be implemented in Matlab [15]. The overall obtained advantage is the need to solve only one LP problem, without the need to look for adequate finite weights to provide to the algorithm, in contrast to the

nonpreemptive method proposed in [11]. Going towards this end, some technical questions arise, which must be solved.

THE GROSSONE METHODOLOGY

As stated above, in [1,2] a new infinite unit of measure (called *grossone* and indicated by the numeral ①) has been introduced as the number of elements of the set of natural numbers. In the new numeral system built upon *grossone* there is the opportunity to treat infinite and infinitesimal numbers as particular cases of a single structure. The new numeral *grossone* can be 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 [1,2] are added to the axioms of real numbers. The new axiom is composed in three parts: Infinity, Identity, and Divisibility:

- *Infinity*. Any finite natural number *n* is less than *grossone*: $n < ①$.
- Identity. The following relationships link ① to the identity elements zero and one:

$$0 \cdot ① = ① \cdot 0 = 0, \quad ① - ① = 0, \quad \frac{①}{①} = 1, \quad ①^0 = 1, \quad 1^{①} = 1, \quad 0^{①} = 0$$

- *Divisibility*. For any finite natural number *n*, the sets $\mathbb{N}_{k,n}$, $1 \le k \le n$,

$$\mathbb{N}_{k,n} = k,\ k+n,\ k+2n,\ k+3n,\ \dots, \quad 1 \le k \le n, \quad \bigcup_{k=1}^{n} \mathbb{N}_{k,n} = \mathbb{N}$$

have the same number of elements indicated by $\frac{①}{n}$.

Thus the axiom states that the infinity number $\frac{①}{n}$ is larger than any finite number. ① behaves as any natural number and the quantities $\frac{①}{n}$ are integers for any natural *n*. Since the axiom is added to the standard axioms of real numbers, all standard properties (commutative, associative, existence of inverse, etc.) also apply to ① and *grossone*-based numerals.

In [1,2] a new way to express infinities and infinitesimals is also provided, using a way similar to traditional positional number system, but with the radix ①. A number $\tilde{c}$ in this new system ($\tilde{c}$ will be called *gross*-scalar from here on) can be constructed by subdividing it into groups of corresponding powers of ① and thus can be represented as:

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

where $m, k \in \mathbb{N}$, $p_i \in \mathbb{N}$ $\forall i \in \{1, 2, \dots, m\}$, $-p_i \in \mathbb{N}$ $\forall i \in \{-1, -2, \dots, -k\}$, $p_0 = 0$ and $c_{p_m}, \dots, c_{p_1}, c_{p_0}, c_{p_{-1}}, \dots, c_{p_{-k}}$ are all finite numbers $(\in \mathbb{R})$.

In this new numeral system, finite numbers are represented by numerals with only one *gross*-power $p_0 = 0$. Infinitesimals are represented by numeral $\tilde{c}$ having only negative finite or infinite *gross*-powers. The simplest infinitesimal is $①^{-1}$, for which $①^{-1} \cdot ① = 1$. We note that all infinitesimals are not equal to zero. In particular, $\frac{1}{①} > 0$.

Table 1 shows the notations used in this work to represent scalars, vectors, *gross*-scalars (numerals involving *grossone*) and *gross*-vectors (vectors of *gross*-scalars).

TABLE 1. Summary of the used notations

Italics lowercase (e.g. *n*) is a real scalar (classical Matlab scalar of class *double**)
Bold lowercase (e.g. **x**) is a real vector (classical Matlab vector of class *double*)
Bold Uppercase (e.g. **A**) is a real matrix (classical Matlab matrix of class *double*)
Italics lowercase with tilde (e.g. $\tilde{c}$) is a *gross*-scalar (Matlab class *GrossScalar*)
Bold lowercase with tilde (e.g. $\tilde{\mathbf{y}}$) is a *gross*-vector (Matlab class *GrossVector*)

* Matlab class that implements the IEEE standard 754-1985 to represent floating point numbers in double precision

LEXICOGRAPHIC MULTI-OBJECTVE LINEAR PROGRAMMING

Consider the LMOLP problem:

$$\begin{aligned} &\text{lexmax } \mathbf{c}^1\mathbf{x}, \mathbf{c}^2\mathbf{x},..., \mathbf{c}^r\mathbf{x} \\ &\text{subject to } \left\{\mathbf{x} \in \mathbb{R}^n, \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\right\} \end{aligned} \tag{1}$$

where *lexmax* means that the objectives are incommensurable, in the sense 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 $\mathbf{c}^1\mathbf{x} \gg \mathbf{c}^2\mathbf{x} \gg ... \gg \mathbf{c}^r\mathbf{x}$.

The preemptive scheme

The preemptive approach starts by solving the single-objective problem:

$$\begin{aligned} &\text{maximize } \mathbf{c}^1\mathbf{x} \\ &\text{subject to } \left\{\mathbf{x} \in \mathbb{R}^n, \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\right\} \end{aligned} \tag{2}$$

By using any algorithm (simplex algorithm, interior point algorithm, etc.) an optimal solution $\mathbf{x}^{*1}$ can be found, associated with the first objective value $\beta_1 = \mathbf{c}^1\mathbf{x}^{*1}$.

Than it solves the second single-objective LP problem:

$$\begin{aligned} &\text{maximize } \mathbf{c}^2\mathbf{x} \\ &\text{subject to } \left\{\mathbf{x} \in \mathbb{R}^n,\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq 0,\ \mathbf{c}^1\mathbf{x} = \beta_1\right\} \end{aligned} \tag{3}$$

The algorithm stops when either we have considered the last objective $\left(\mathbf{c}^r\mathbf{x}\right)$ or when a non-degenerate solution has been found in the last solved LP problem. As anyone can deduce, this approach is time consuming.

The nonpreemptive scheme (using appropriate finite weights)

In [11] it has been demonstrated that there always exists a scalar $M \in \mathbb{R}$ such that the solution to a LMOLP problem can be found by solving only one single-objective LP problem of the form:

$$\begin{aligned} &\text{maximize } \bar{\mathbf{c}}\mathbf{x} \\ &\text{subject to } \left\{\mathbf{x} \in \mathbb{R}^n,\ \mathbf{A}\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq 0\right\} \end{aligned} \tag{4}$$

where $\bar{\mathbf{c}} = \sum_{i=1}^{r} M^{-i+1}\mathbf{c}^i$.

This is a powerful theoretical result. However, from the computational point of view, finding the value of M is not a trivial task. Its computation might make the overall time consumption (computing M and solve one LP problem) more time consuming than solving the original problem 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 *M*.

In the following we will present the idea of using 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 we will call it *gross*-LP problem, to avoid confusion with its standard formulation involving finite numbers only.

The nonpreemptive scheme using *grossone*

Following the lexicographic concept defined in [10], we re-formulate the problem (4) by making use of *gross*-scalars and *gross*-vectors (we call it *gross-primal problem*):

$$\begin{aligned} &\text{maximize } \tilde{\mathbf{c}}\mathbf{x} \\ &\text{subject to } \left\{\mathbf{x}\in\mathbb{R}^n,\ \mathbf{A}\mathbf{x}=\mathbf{b},\ \mathbf{x}\geq 0\right\} \end{aligned} \tag{5}$$

where $\tilde{\mathbf{c}}$ is a row-wise *gross*-vector, defined as $\tilde{\mathbf{c}}=\sum_{i=1}^{r}①^{-i+1}\mathbf{c}^i$ and $\tilde{\mathbf{c}}\mathbf{x}$ is the *gross*-scalar given by:

$$\tilde{\mathbf{c}}\mathbf{x}=(c_1^1x_1+\ldots+c_n^1x_n)①^0+(c_1^2x_1+\ldots+c_n^2x_n)①^{-1}+\ldots+(c_1^rx_1+\ldots+c_n^rx_n)①^{-r+1}.$$

The equivalence of problems (5) and (1) can be found in [15].

The fundamental difference between the definition of $\tilde{\mathbf{c}}$ in (5) with respect to the definition of $\bar{\mathbf{c}}$ in (4) 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 (4). However, this advantage leads to the fact that standard algorithms (like the simplex algorithm or the interior point algorithm) traditionally used for solving (4) can no longer be used in this case. As we will see in the next section, it is still possible to obtain an optimality condition that allows implementing the *gross*-simplex algorithm (a generalization of the traditional simplex algorithm) [15]. The latter is able to solve problem (5) algorithmically.

OPTIMALITY CONDITION

The single-objective *gross*-LP problem formulated in (5) using *grossone* can be solved using the *gross*-simplex algorithm described in [15], provided that the duality theory is extended to the case of *gross*-scalars and *gross*-vectors. First of all let us observe that the dual problem (denoted as *gross-dual problem*) associated with problem (5) is the following one:

$$\begin{aligned} &\text{minimize } \tilde{\mathbf{y}}\mathbf{b} \\ &\text{subject to } \tilde{\mathbf{y}}\mathbf{A}\geq\tilde{\mathbf{c}} \end{aligned} \tag{6}$$

where $\tilde{\mathbf{y}}$ is a row-wise *gross*-vector. Let S be the polyhedron $\left\{\mathbf{x}\in\mathbb{R}^n,\ \mathbf{A}\mathbf{x}=\mathbf{b},\ \mathbf{x}\geq 0\right\}$ and let $\tilde{D}$ the associated dual polyhedron. More precisely, $\tilde{D}$ is the *gross*-polyhedron laying is the span of $\left[\tilde{\mathbf{e}}^0,\tilde{\mathbf{e}}^{-1},\ldots,\tilde{\mathbf{e}}^{-r+1}\right]$ (where $\tilde{\mathbf{e}}^0=\left[1\ \ 0\ \ \ldots\ \ 0\right]$, $\tilde{\mathbf{e}}^{-1}=\left[0\ \ ①^{-1}\ \ 0\ \ \ldots\ \ 0\right]$, …, $\tilde{\mathbf{e}}^{-r+1}=\left[0\ \ \ldots\ \ 0\ \ ①^{-r+1}\right]$) made by all the *gross*-vectors $\tilde{\mathbf{y}}$ such that $\tilde{\mathbf{y}}\mathbf{A}\geq\tilde{\mathbf{c}}$: $\tilde{D}\equiv\left\{\tilde{\mathbf{y}}\in\text{span}\left[\tilde{\mathbf{e}}^0,\ldots,\tilde{\mathbf{e}}^{-r+1}\right],\ \tilde{\mathbf{y}}\mathbf{A}\geq\tilde{\mathbf{c}}\right\}$.

In the following we will assume that both S and $\tilde{D}$ are non-empty.

Definition of optimality

$\mathbf{x}^*\in S$ is an optimal solution for the primal problem $\left(\mathbf{x}^*\in S_{opt}\right)$ iff $\tilde{\mathbf{c}}\mathbf{x}^*\geq\tilde{\mathbf{c}}\mathbf{x}\quad\forall\mathbf{x}\in S$.

$\tilde{\mathbf{y}}^*\in\tilde{D}$ is an optimal solution for the dual problem $\left(\tilde{\mathbf{y}}^*\in\tilde{D}_{opt}\right)$ iff $\tilde{\mathbf{y}}^*\mathbf{b}\leq\tilde{\mathbf{y}}\mathbf{b}\quad\forall\tilde{\mathbf{y}}\in\tilde{D}$.

Lemma (weak duality)

$\forall\mathbf{x}_0\in S$ and $\forall\tilde{\mathbf{y}}_0\in\tilde{D}$, $\quad\tilde{\mathbf{c}}\mathbf{x}_0\leq\tilde{\mathbf{y}}_0\mathbf{b}$ holds.
For the proof, see [15].

Optimality condition

If $\mathbf{x}^* \in S$, $\tilde{\mathbf{y}}^* \in \tilde{D}$ and furthermore $\tilde{\mathbf{c}}\mathbf{x}^* = \tilde{\mathbf{y}}^*\mathbf{b}$, then $\mathbf{x}^* \in S_{opt}$ and $\tilde{\mathbf{y}}^* \in \tilde{D}_{opt}$.

It is an immediate consequence of the weak duality lemma.

MATLAB IMPLEMENTATION: THE GROSSONE LIGHT TOOLBOX

We have already sketched a Matlab implementation of the proposed approach, allowing us to obtain preliminary results that confirm the feasibility of the approach. In this section we document this effort very briefly. Keeping in mind that the final goal was the derivation of a simplex algorithm able to solve the LP problem involving *grossone* in (5), we had to represent *grossone*-based numerals in Matlab. To do so, we have followed an object-programming paradigm and we have implemented the *GrossScalar* class. That class comes with basic algebraic operations needed by our *gross*-simplex algorithm: addition, subtraction between two *gross*-scalars, multiplication of a *GrossScalar* by a real scalar, etc. Another class proved to be useful: the *GrossVector* class, which is a handy container of *gross*-scalars that allows us to work with vectors of *gross*-scalars more efficiently. The speedup can be significant, especially when the code is written in a vectorized fashion and a GPGPU (General Purpose Graphics Processing Units) is available on the machine running the toolbox [16]. Note that in the *gross*-simplex algorithm we had to pre-multiply the inverse of a scalar matrix by a *GrossVector*: this is a linear combination of *gross*-scalars and thus it does not require the division between *gross*-scalars (a more complex operation w.r.t. addition and multiplication), if the inverse of the real matrix is computed before the multiplication. The toolbox has been named *light* since it does not support the division between *gross*-scalars and *grossone*-based numerals and can only handle integer powers of *grossone*.

ACKNOWLEDGMENTS

The research of Ya.D. Sergeyev was supported by the Russian Science Foundation, project No 15-11-30022 “Global optimization, supercomputing computations, and applications”.

REFERENCES

1. Ya.D. Sergeyev, *Arithmetic of Infinity* (Edizioni Orizzonti Meridionali , CS, Italy, 2003, 2d el. ed. 2013).
2. Ya.D. Sergeyev, Informatica **19(4)**, 567–596 (2008).
3. M. Margenstern, Applied Mathematics and Computation, **278**, 45–53 (2016).
4. Ya.D. Sergeyev, Communications in Nonlinear Science and Numerical Simulation **31(1–3)**, 21–29 (2016).
5. L. D’Alotto, Applied Mathematics and Computation **255**, 15–24 (2015).
6. Ya.D. Sergeyev, Applied Mathematics and Computation **219(22)**, 10668–10681 (2013).
7. S. De Cosmis and R. De Leone, Applied Mathematics and Computation **218(16)**, 8029–8038, (2012).
8. Ya.D. Sergeyev, M.S. Mukhametzhanov, F. Mazzia, F. Iavernaro, and P. Amodio, Int. J. of Unconventional Computing **12(1)**, 3–23 (2016).
9. P. Amodio, F. Iavernaro, F. Mazzia, M.S. Mukhametzhanov, and Ya.D. Sergeyev, Mathematics and Computers in Simulation, doi:10.1016/j.matcom.2016.03.007, (*in press*).
10. Ya.D. Sergeyev, The Mathematical Intelligencer **37(2)**, 4–8 (2015).
11. H.D. Sherali and A.L. Soyster, Journal of Optimization Theory and Applications **39(2)**, 173–186 (1983).
12. H. Isermann, OR Spektrum **4**, 223–228 (1982).
13. L. Pourkarimi and M. Zarepisheh, European Journal of Operational Research **176**, 1348–1356 (2007).
14. I.P. Stanimirović, Facta universitatis - series: Mathematics and Informatics **27(1)**, 55–66 (2012).
15. M. Cococcioni, M. Pappalardo and Ya.D. Sergeyev, “Lexicographic Multi-Objective Linear Programming using Grossone Methodology: Theory and Algorithm”, (*under preparation*).
16. M. Cococcioni, M. Rixen and R. Grasso, “Rapid prototyping of high performance fuzzy computing applications using high level GPU programming for maritime operations support,” in *Proc. of 2011 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA'11)*, Paris (France), 2011, pp. 17–23.