Applied Mathematics and Computation 218 (2012) 8029–8038

Contents lists available at SciVerse ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

The use of grossone in Mathematical Programming and Operations Research

Sonia De Cosmis, Renato De Leone *

School of Science and Technology, Università di Camerino, via Madonna delle Carceri 9, Camerino (MC) 62032, Italy

ARTICLE INFO

Keywords:
Linear programming
Simplex Method
Nonlinear programming
Penalty methods

ABSTRACT

The concepts of infinity and infinitesimal in mathematics date back to ancients Greek and have always attracted great attention. Very recently, a new methodology has been proposed by Sergeyev [10] for performing calculations with infinite and infinitesimal quantities, by introducing an infinite unit of measure expressed by the numeral ① (grossone). An important characteristic of this novel approach is its attention to numerical aspects. In this paper we will present some possible applications and use of ① in Operations Research and Mathematical Programming. In particular, we will show how the use of ① can be beneficial in anti-cycling procedure for the well-known Simplex Method for solving Linear Programming problems and in defining exact differentiable penalty functions in Nonlinear Programming.

1. Introduction

A novel approach to infinite and infinitesimal numbers has been recently proposed by Sergeyev in a book and in a series of papers [10–13]. By introducing a new infinite unit of measure (the numeral *grossone*, indicated by ①) as the number of elements of the set of the natural numbers, he shows that it is possible to effectively work with infinite and infinitesimal quantities and to solve many problems connected to them in the field of applied and theoretical mathematics. In this new system, there is the opportunity to treat infinite and infinitesimal numbers as particular cases of a single structure, offering a new view and alternative approaches to important aspects of mathematics such as sums of series (in particular, divergent series), limits, derivatives, etc.

The new numeral *grossone* can be introduced by describing its properties (in a similar way as done in the past with the introduction of zero to switch from natural to integer numbers). The Infinity Unit Axiom postulate (IUA) [11,10] is composed of three parts: Infinity, Identity, and Divisibility:

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

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

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

$$\mathbb{N}_{k,n} = k, k+n, k+2n, k+3n, \ldots, \quad 1 \leqslant k \leqslant n, \quad \bigcup_{k=1}^{n} \mathbb{N}_{k,n} = \mathbb{N} \tag{2}$$

* Corresponding author.
E-mail addresses: sonia.decosmis@unicam.it (S. De Cosmis), renato.deleone@unicam.it (R. De Leone).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2011.07.042

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

The axiom above states that the infinite number ①, greater than any finite number, behaves as any natural number with the elements zero and one. Moreover, the quantities $\frac{①}{n}$ are integers for any natural n. This axiom is added to the standard axioms of real numbers and, therefore, all standard properties (commutative, associative, existence of inverse, etc.) also apply to ①.

Sergeyev [12,13] also defines a new way to express the infinite and infinitesimal numbers using a register similar to traditional positional number system, but with base number ①. A number **C** in this new system can be constructed by subdividing it into groups corresponding to powers of ① and has the following representation:

$$\mathbf{C} = c_{p_m}①^{p_m} + \cdots + c_{p_1}①^{p_1} + c_{p_0}①^{p_0} + c_{p_{-1}}①^{p_{-1}} + \cdots + c_{p_{-k}}①^{p_{-k}}. \tag{3}$$

where the quantities c_i (the *grossdigits*) and p_i (the *grosspowers*) are expressed by the traditional numerical system for representing finite numbers (for example, floating point numbers). The grosspowers are sorted in descending order:

$$p_m > p_{m-1} > \cdots > p_1 > p_0 > p_{-1} > \cdots p_{-(k-1)} > p_{-k}$$

with $p_0 = 0$.

In this new numeral system, finite numbers are represented by numerals with only one grosspower $p_0 = 0$. Infinitesimal numbers are represented by numeral **C** having only negative finite or infinite grosspowers. The simplest infinitesimal number is $①^{-1}$ for which

$$①^{-1}① = ①①^{-1} = 1. \tag{4}$$

We note that infinitesimal numbers are not equal to zero. In particular, $\frac{1}{①} > 0$. Infinite numbers are expressed by numerals having at least one finite or infinite grosspower greater than zero.

A peculiar characteristic of the newly proposed numeral system is its attention to its numerical aspects and to applications. The Infinity Computer proposed by Sergeyev is able to execute computations with infinite, finite, and infinitesimal numbers numerically (not symbolically) in a novel framework.

In this paper we will present two possible uses of this numeral system in Mathematical Programming and Operations Research. In particular, in Section 2 we will show a simple way to implement anti-cycling strategies in the Simplex Method for solving Linear Programming problems. Various anti-cycling procedures have been proposed and implemented in state-of-the-art softwares. The lexicographic strategies has received particular attention since it allows, in contrast to Bland's rule, complete freedom in choosing, at each iteration, the entering variable. In Section 3 we revert our attention to Nonlinear Programming problems and, in particular, to differentiable penalty functions. In the new numeral system it is possible to define an exact, differentiable penalty function and we will show that stationary points of this penalty function are KKT points for the original Nonlinear Programming problem. Two simple examples are also provided showing the effectiveness of this approach. Conclusions and indications for further applications of ① in Mathematical Programming are reported in Section 4.

We briefly describe our notation now. All vectors are column vectors and will be indicated with lower case Latin letter $(x, z, \ldots)$. Subscripts indicate components of a vector, while superscripts are used to identify different vectors. Matrices will be indicated with upper case roman letter $(A, B, \ldots)$. If $A \in \mathbb{R}^{m\times n}$, $A_{.j}$ is the jth column of A; if $B \subseteq \{1, \ldots, n\}$, $A_{.B}$ is the submatrix of A composed by all columns $A_{.j}$ such that $j \in B$. The set of real numbers and the set of nonnegative real numbers will be denoted by $\mathbb{R}$ and $\mathbb{R}_+$ respectively. The rank of a matrix A will be indicated by rank A. The space of the n-dimensional vectors with real components will be indicated by $\mathbb{R}^n$ and $\mathbb{R}^n_+$ is an abbreviation for the nonnegative orthant in $\mathbb{R}^n$. The symbol $\|x\|$ indicates the euclidean norm of a vector x. Superscript ^{T} indicates transpose. The scalar product of two vectors x and y in $\mathbb{R}^n$ will be denoted by x^Ty. Here and throughout the symbols :=and :=denote definition of the term on the left and the right sides of each symbol, respectively. The gradient $\nabla f(x)$ of a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $x \in \mathbb{R}^n$ is assumed to be a column vector. If $F : \mathbb{R}^n \to \mathbb{R}^m$ is a continuously differentiable vector-valued function, then $\nabla F(x)$ denotes the Jacobian matrix of F at $x \in \mathbb{R}^n$.

2. Lexicographic rule and grossone

The Simplex Method, originally proposed by Dantzig [4] more than half a century ago, is still today one of the most used algorithms for solving Linear Programming problems. Finite termination of the method can only be guaranteed if special techniques are employed to eliminate cycling. In this section we will show how in the new numeral system it is very simple to implement such anti-cycling rules.

Given a matrix $A \in \mathbb{R}^{m\times n}$ with rank $A = m$ and vectors $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$, the Linear Programming problem in standard form can be stated as follows:

$$\begin{aligned} \min_x \quad & c^Tx \\ \text{subject to} \quad & Ax = b \\ & x \geqslant 0. \end{aligned} \tag{5}$$