Available online at www.sciencedirect.com

SCIENCE DIRECT®

Decision Support Systems 41 (2005) 160–175

Decision Support Systems

www.elsevier.com/locate/dsw

MOPEN: A computational package for Linear Multiobjective and Goal Programming problems

R. Caballero*, M. Luque, J. Molina, F. Ruiz

Department of Applied Economics (Mathematics), Faculty of Economics, University of Málaga, Campus El Ejido s/n. Málaga, 29071, Spain

Received 30 November 2002; accepted 1 June 2004
Available online 14 August 2004

Abstract

MOPEN is a computational package designed as a global tool for Linear Multiobjective and Goal Programming problems with continuous and/or integer variables. The main existing techniques for these problems have been included in this package. That is, it is possible to generate or approximate the efficient set using Generating Methods, to obtain Compromise solutions or to use Goal Programming or reference Point approaches. As will be described, many advanced options have been implemented with every method. MOPEN has been implemented under a Windows environment; thus, it is easy to build and handle the data entry files and the result layout files. The behavior of MOPEN—in terms of CPU time used to solve large problems—can be considered as good; therefore, this package is a powerful tool to handle the previously mentioned problems.

Keywords: Multiobjective Programming; Computational package; Goal Programming; Integer programming

1. Introduction

1.1. Basic Definitions

In this paper, the general Linear Multiobjective problem

$$\begin{array}{ll} & \mathrm{Min}(f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_p(\mathbf{x})) = \left(\mathbf{c}_1^t\mathbf{x}, \mathbf{c}_2^t\mathbf{x}, \cdots, \mathbf{c}_p^t\mathbf{x}\right) \\ (LMOP) & \text{s.t.}: \mathbf{x}\in X = \{\mathbf{x}\in R^n : A\mathbf{x}\leq\mathbf{b}\} \\ & A\in M_{s\times n}(R),\ \mathbf{b}\in R^s \end{array}$$

will be considered. The following classical concepts will be used throughout the paper:

- Let $\mathbf{x}_i$* be the optimum value of f_i: $f_i(\mathbf{x}_i*)$=min $\{f_i(\mathbf{x})/\mathbf{x}\in R^{\mathrm{n}},\ A\cdot\mathbf{x}\leq\mathbf{b}\}$. $\mathbf{x}_i$* is called the Ideal Solution of f_i, and $f_i*=f(\mathbf{x}_i*)$ is its Ideal Value.
- The pay-off matrix is formed by the values of all the functions f_i in all the ideal solutions:

$$\begin{pmatrix} f_1(\mathbf{x}_1*) & f_1(\mathbf{x}_2*) & \cdots & f_1(\mathbf{x}_n*) \\ f_2(\mathbf{x}_1*) & f_2(\mathbf{x}_2*) & \cdots & f_2(\mathbf{x}_n*) \\ \vdots & \vdots & & \vdots \\ f_n(\mathbf{x}_1*) & f_n(\mathbf{x}_2*) & \cdots & f_n(\mathbf{x}_n*) \end{pmatrix}$$

* Corresponding author. Tel.: +34 95 213 1168; fax: +34 95 213 2061.
E-mail address: r_caballero@uma.es (R. Caballero).

0167-9236/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2004.06.002

The elements of the main diagonal of the pay-off matrix are the ideal values of each function. The maximum value per column, $\max_{j=1,\cdots,p}\left\{f_i\left(\mathbf{x}_j*\right)\right\}$, is called the *Anti-ideal Value* of f_i and is denoted by $\mathbf{f}^{(i)}$. The corresponding solution is called the *Anti-ideal Solution* of f_i and is denoted by $\mathbf{x}^{(i)}$.

- A feasible solution $\mathbf{x}^*\in X$ is said to be efficient for (LMOP) (or Pareto optimal) if there does not exist any other solution $\mathbf{x}\in X$, such that:

$$f_i(\mathbf{x})\leq f_i(\mathbf{x}^*) \qquad \forall i=1,\cdots,p$$

with at least one $j\in\{1,\ldots,p\}$ such that $f_j(\mathbf{x})<f_j(\mathbf{x}^*)$.

- A feasible solution $x^*\in X$ is said to be weakly efficient for (LMOP) (or weakly Pareto optimal) if there does not exist any other solution $\mathbf{x}\in X$, such that:

$$f_i(\mathbf{x})<f_i(\mathbf{x}^*) \qquad \forall i=1,\cdots,p.$$

The most widely accepted classification of the existing techniques to solve these problems depends on the information flow between the decision maker and the analyst. The first possibility is that the decision maker provides his/her preferences only by stating which objectives have to be minimized and which ones have to be maximized. In this case, the aim of the methods is to show the Pareto efficient set (or an approximation of it). The techniques corresponding to this scheme are called *Generating Techniques* (see, for example, Ref. [16]). Second, the decision maker may want to overcome the conflict among the objectives, without having to state a clear preference towards a specific one. In this case, an efficient solution has to be found, characterized by creating a compromise or equilibrium among the objectives. This is the basic idea of *Compromise Programming* [23,24,15]. On the other hand, if, prior to the resolution process, the decision maker provides information in the form of target values, α_i, for each objective, and possibly preference levels among them, then this constitutes the *Goal Programming* scheme (see Refs. [5,6,8]). Similarly, the *Reference Point* method [22] lets the decision maker establish aspiration levels for the objectives, without having to renounce the efficiency of the solutions. Finally, in the *Interactive Techniques* (see, for example, [9]), there exists a continuous flow of information between the analyst and the decision maker, throughout the whole resolution process).

In this paper, the software MOPEN is described, in which the main algorithms belonging to the two first groups have been implemented, while the interactive techniques are implemented in PROMOIN (see Ref. [4]).

1.2. Existing implementations

Two main implementations can be found within the previously described framework: ADBASE and the systems GPSYS and IGPSYS. ADBASE was developed by Ralph Steuer in 1974, in the FORTRAN language, although, since then, it has undergone several revisions and improvements. This software determines all the efficient vertices and edges of a Linear Multiobjective problem. Besides this, it also has the option to solve a Lexicographic Goal Programming problem. This program runs under an MSDOS environment and uses a specific format for the data entry files. The original idea of the author was to create a code that was general enough to be implemented under any operating system, in a personal computer, and at a time when the use of Windows was not very widespread. These data entry files can only be edited by an MSDOS program, and this fact can be inconvenient when using Windows. On the other hand, GPSYS and IGPSYS, developed by M. Tamiz and D.F. Jones, are practically the only implementations for Goal Programming problems available at the moment. GPSYS solves Linear Goal Programming problems, and IGPSYS solves Integer Linear Goal Programming problems. Both programs have been implemented in the FORTRAN language and run under MSDOS. The implementations include all the normalization possibilities for Goal Programming as well as several options to detect and restore the efficiency of the final solutions. Nevertheless, the difficulties encountered regarding editing the data entry file are similar to those described for ADBASE. Thus, both implementations are highly efficient from a computational point of view, although they do not take advantage of the benefits offered by the implementation under a Windows environment.