BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **9**, No 4

Sofia • 2009

In memoriam. *To Professor Vassil Vassilev, who made me feel like home during my visit to the Bulgarian Academy of Sciences on July, 2008.*
F. Ruiz.

Towards a Unified Global Interactive Multiobjective Decision System

Jose Manuel Cabello, Mariano Luque, Francisco Ruiz[1]

Dpt. Applied Economics (Mathematics), University of Málaga, Spain

Abstract: *The wide variety of existing interactive methods brings the need of global interactive systems that enable the decision maker to choose the method that best fits his preferences. Such a system must combine a sufficient variety of methods of different kinds (requiring different types of information to the Decision Maker (DM)), and enable the possibility to change the method any time during the solution process. Besides, a compact mathematical formulation increases the computational efficiency of the implementation. Finally, a good user interface is vital for the practical success of such a system. Professor Vassilev and his research team have been working on this idea for a long time. This paper presents some research results, closely related to this topic.*

Keywords: *Multiobjective Programming, Interactive Methods, Global Decision System, Global Formulation.*

1. Introduction

MultiObjective Programming (MOP) is a branch of Multiple Criteria Decision Making (MCDM) which studies problems where several conflictive objectives are optimized, within a feasible set which is defined by a set of constraints:

$$\begin{array}{ll} \max & f(x) = \left(f_1(x),\ f_2(x), \ldots, f_k(x) \right) \\ \text{s.t.:} & x \in X \end{array} \qquad \text{(MOP)}.$$

[1] Corresponding autor: C/ Ejido, 6. 29071-Málaga, Spain. Tel. +34952131167. E-mail: rua@uma.es

The classification of the techniques that deal with this problem into three main groups is widely accepted by the scientific community (see, for example S t e u e r (1986) or M i e t t i n e n (1999)). The so-called generating techniques or methods without apriori information form the first group. The aim of such methods is to calculate efficient (or non-dominated) solutions for (MOP), or in the best case, to approximate or fully characterize the whole efficient set. For example, in S t e u e r (1986) the program ADBASE (developed originally in 1974) is described, which determines the efficient set for linear multiobjective problems. The problem with this kind of methods is that the amount of information generated is usually too large for the decision maker to manage.

The techniques with apriori information form the second group. In these methods, the solution of the problem is carried out on the basis of some information given by the decision-maker about his preferences, prior to the application of the algorithm. Depending on which information is required, what kind of a final solution is searched, and the solving philosophy used, there are several techniques within this group. Given their importance, Reference Point Methods, Compromise Programming and Goal Programming (with several different variants) should be pointed out. These algorithms have been widely used to solve real MOP problems, and their success and validity are depicted in literature. Besides, there exist several implementations for these techniques, among which GPSYS system for Goal Programming, described in J o n e s (1998) must be considered. In order to apply all these methods, deep knowledge of the problem by the decision-maker is vital, so that the required information is provided with the accuracy needed to trust the final solution. Besides, in some cases the final solution may not be unique and in others, the wishes of the decision-maker are not fully satisfied. So, the decision-maker may wish to carry out an ulterior election of a solution within a more reduced set, in the former case, or to readjust his aspiration levels in the latter. Obviously, these problems are overcome in practice through the iterated application of the techniques. That is, the problem is solved using a determined information, the solution is analyzed by the decision-maker, who may wish to actualize the information given to the method and solve the problem again, and so on.

This is precisely the main idea that underlies the family of methods that form the third block: the interactive techniques. In such methods, the information exchange between the algorithm and the decision maker is carried out in a continuous way along the whole resolution process. This way, the different solutions of the iterative process are progressively adapted to DM’s preference structure. In other words, an interactive algorithm requires periodically information to the decision-maker, in order to readapt its inner search procedure towards the solution that best fits the decision-maker's preferences. Nowadays, a great number of interactive methods can be found in the literature, which vary according to the type and form of the information required by the DM, and to the resolution technique used to solve the intermediate problems. In A s k o y et al. (1996), L u q u e (2000), M i e t t i n e n (1999), S h i n and R a v i n d r a n (1991), among others, several descriptions and classifications of such interactive methods can be found.

There are two main questions that arise when applying interactive methods to a real MOP problem. First, the nature of such algorithms implies the necessity of computational implementations for their correct and comfortable use. Second, the election of the method is not a trivial task at all. The method should depend, among other questions, on which type of solutions the decision-maker wants to obtain, and on which type of information is more comfortable for him to provide along the process. With respect to the first question, there exist several implementations, which apply a single interactive procedure each. Besides, not all the methods do have an available implementation. Among the mentioned implementations, the following must be pointed out:

- VIA (Visual Interactive Approach), developed by K o r h o n e n and L a a k s o (1986) has an implementation, known as "Pareto Race", carried out by K o r h o n e n and W a l l e n i u s (1988).
- The Satisficing Trade-Off Method (STOM) published by N a k a y a m a and S a w a r a g i (1984) has got an implementation, known as MONP-16, whish was developed by V a s s i l e v et al. (1990).
- J a s z k i e w i c z and S l o w i n s k i (1999) developed the Light Beam Search algorithm (LBS), where an implementation is also reported.
- The Nondifferentiable Interactive Multiobjective BUndle-based optimization System (NIMBUS) was published, together with its implementation, by M i e t t i n e n and M ä k e l ä (1995, 2000).

With respect to the second question, it seems clear that, given that in real practice (despite all the existing convergence proofs of many methods) the final solution is very likely to be different, depending on the method used, two issues should be taken into account. On one hand, the analyst must be provided with a wide variety of algorithms among which he can choose, depending on factors like which kind of information the decision-maker prefers to give; how he wants the information to be presented by the method; what kind of a final solution he wants to obtain. With respect to the type of the information required from the DM, there exist basically four types of interactive methods:

- Trade-off based methods, where the DM must give the tradeoff (or local weights) at the current solution.
- Election methods, where the DM just has to choose, at each iteration, one efficient solution among several ones.
- Reference point based methods, where the DM is asked to give, at each iteration, desirable levels for each objective.
- Classification methods, where the DM, given the current solution, classifies the objectives into a series of groups (objectives to be improved, to leave as they are, to be impaired).

On the other hand, it is probable that, at some stage of the process, the DM gets tired of answering the same kind of questions, or he does not notice any progress during several iterations, or he is unable to be more accurate in his answers. These facts will unavoidably cause incoherent answers, or yield a wrong final solution. One possible way to overcome this problem is to allow the decision-maker to swap to another method during the resolution process. This change should not mean just a restart of the procedure, without taking into account all the iterations already carried out, and the information obtained so far by both the DM and the analyst. In other words, we should wonder whether the method chosen by the analyst is always the best one for both the problem and the decision maker, or it is just the method the analyst himself prefers. In our opinion, the working frame must be wide enough, so as not to force (by the use of a certain method) the DM towards a final solution that is not the one he really prefers. It seems clear that the correct election of the method is vital in the resolution process, and it depends on the problem and on the DM In our opinion, it also depends on the stage of the resolution process. In this sense, it must be pointed out that our aim is not to make the DM jump from one method to another once and again. Rather than that, we think that the change-of-method option can be very useful especially at the end of the resolution process, when the DM finds it very hard to make a significant progress towards his most preferred solution.

Anyway, it is very difficult to find in literature an implementation where the user can choose among several interactive methods. The first approach to this kind of software was described in G a r d i n e r and S t e u e r (1994a, 1994b), where the computational structure of an open architecture which can hold many different interactive methods was proposed. Nevertheless, this idea has not been actually implemented yet. In 2005, V a s s i l e v et al. presented the first such system, called Multidecision-1, which consisted of two separate parts (the systems MKA-1 and MKO-1) and which was designed to support decision makers in solving different multicriteria analysis and multicriteria optimization problems. The second (improved) version of this software is presented in V a s s i l e v et al. (2008). In this paper (Section 2), we present the interactive system PROMOIN (C a b a l l e r o et al., 2002), which was developed by the multicriteria group of the University of Málaga.

For developing such an interactive system, it is also important to formulate a compact global mathematical formulation that can accommodate different methods, with the aim of increasing the computational efficiency of the system. This formulation can be the core of the code, in such a way that it is used to solve all the intermediate problems. M. V a s s i l e v a (2005) proposed the first version of such a formulation (called GENS) which supported several interactive methods. In Section 3, we will present the compact formulation GLIDE (L u q u e, R u i z, M i e t t i n e n, 2009), which has been designed in order to accommodate interactive methods of all the four types of information.

2. PROMOIN system

This program has been implemented in C^{++} language, using the compiler Microsoft Visual C^{++}, and thus, it can be run on a personal computer that works under Windows environment. The first version of the system has been developed for linear multiobjective problems (that is, multiobjective problems where the objective functions and the constraints are all linear). The single objective intermediate problems have been solved using the subroutine library NAG for C, mark 6 (2000), and thus, it is necessary to have the corresponding license to use the program. Nevertheless, it is possible to adapt the system to other linear solvers.

Let us now describe in detail some relevant aspects of the system.

2.1. Methods implemented

With the aim of providing the user with a wide variety of interactive methods, a group of algorithms, among those that are better known and described in literature, have been implemented in the system. Such methods have been classified according to the kind of information required from the decision-maker, in the following way:

- Tradeoffs (or local weights) based methods:
  - GDF published by G e o f f r i o n, D y e r and F e i n b e r g (1972);
  - IGP (Interactive Goal Programming) published by D y e r (1972);
  - ISWT (Interactive Surrogate Worth Tradeoff) published by C h a n k o n g and H a i m e s (1978);
  - SPOT (Sequential Proxy Optimization Technique) published by S a k a w a (1982);
  - PROJECT method published by L u q u e, Y a n g, W o n g (2009).
- Solutions Generating Methods:
  - Zionts-Wallenius method, published in 1976;
  - Tchebychev method, published by S t e u e r and C h o o (1983);
  - MICA method, published by L u q u e, R u i z and S t e u e r (2009).
- Reference Points Methods:
  - Reference Point Method, published by W i e r z b i c k i (1980);
  - STOM (Satisfacing Trade-Off Method) published by N a k a y a m a and S a w a r a g i (1984);
  - VIA (Visual Interactive Approach) published by K o r h o n e n and L a a k s o (1986).
- Classification Methods:
  - STEM (Step Method) published by B e n a y o u n et al. (1971).

In this way, once a problem has been selected for its resolution, the user can choose among a wide variety of methods, depending on several factors, as it has been explained before. In general, in order to apply a method correctly, not only knowledge of the problem is necessary by the DM, but a, at least, basic knowledge of the algorithm as well. In this sense, it must be pointed out that this system does not substitute the figure of the analyst, who is essential in order to assist the DM. The analyst should be able to inform the DM about the kind of information that each method requires, of the main advantages and disadvantages of each one (efficiency of the solutions, convergence, etc). Anyway, as a support tool, a full help system has been included, where, apart from the traditional help items of Windows based problems, all the interactive methods are described in detail from both the theoretical and algorithmical points of view.

The implementations of all the methods include all their particular options, which can be modified along the resolution process. Moreover, in some methods, a menu of advanced parameters, oriented to more specialized analysts, is provided.

2.2. Data entry

The first task to be carried out when solving a new linear multiobjective problem is to introduce its corresponding data into the system. This is done through the elaboration of a text file (*.txt), very easy to create and with a simple structure. The format of this file for an example problem can be seen in Fig. 1.

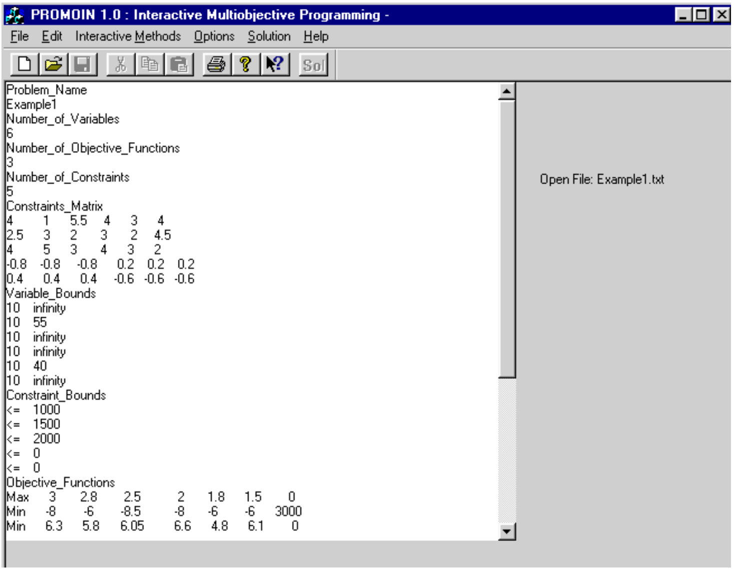

Fig. 1. Example of a txt data file for PROMOIN

This easy format is, in our opinion, the most appropriate one to introduce problems with a large number of variables and constraints, which the user may have stored in another program, like for example a spreadsheet. In these cases, it is not very difficult to export these data into a text file, and afterwards to treat this file in order to put it in the right format for PROMOIN. This is why this text template seems more adequate for large problems. Anyway, the system also offers the possibility of introducing new problems using an assistant. This option may be more adequate for small problems, or examples that are used for teaching purposes.

2.3. Change of a method

Apart form all theoretical considerations, it can be said that, in practice, an interactive method is “good” if it is able to drive quickly the DM towards his most preferred solution (or at least, close enough to it), using the information required. It is a conjunction of these two aspects (information easy-to-provide by the DM and convergence in not many iterations to an acceptable solution) that makes an interactive method a powerful tool within the field of Multiobjective Programming. And, again, despite all theoretical considerations, these properties depend in practice, not only on the method itself, but also on the particular problem and the DM. Using a single method, in determined iterations, the effort that the DM has to make in order to provide the required information, can be relatively small, while later on, it can be much higher. As it has been commented before, the DM can get tired of answering the same type of questions if he does not observe a significant progress, or he can find it hard to be more accurate in his answers. The subsequent imprecision and/or inconsistency in the information provided to the algorithm can endanger the convergence of the method.

Following the idea given in G a r d i n e r and S t e u e r (1994a, 1994b), an effort must be done to offer the analyst a wide variety of possibilities, so that the DM can decide, at each iteration, what kind of information he prefers to provide. In other words, the DM should be able to change the interactive method at any time of the resolution process. For this reason, this possibility has been included in PROMOIN, using the dialog box that is shown on Fig. 2. It is important to point out that it is the analyst who should decide to change the method along the process in order to adapt the algorithm to the kind of information the DM gives at each time. This "change of method" option does not mean just restarting the process with the current iteration and using another method. Based on the theoretical study presented in L u q u e et al. (2007), the system allows making use of all the information provided so far by both the DM and the algorithm. Namely, as it is depicted in Fig. 3, the program builds the information that the decision-maker would have given, if he had used the new method, in order to obtain the solutions that have been obtained using the previous algorithm. This element can be very useful for the DM in order to provide the new information during the subsequent iterations.

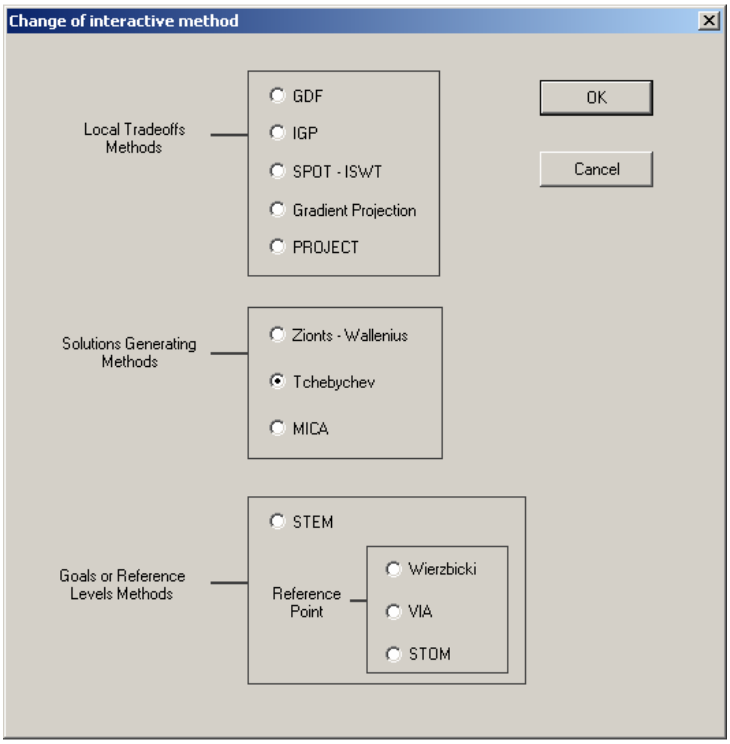

Fig. 2. Change of the interactive method menu

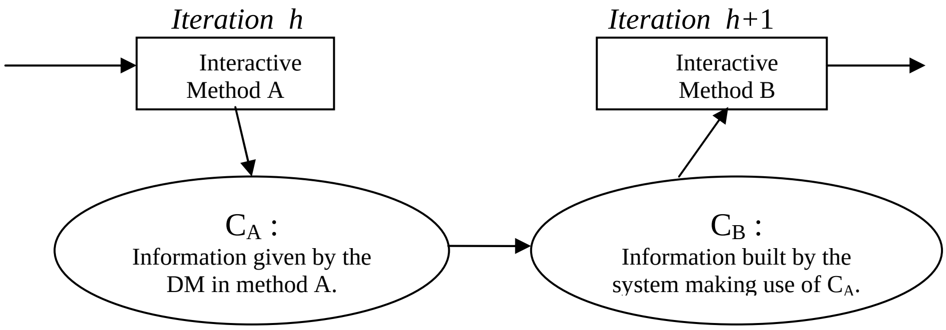

Fig. 3. Graphical scheme of the information transfer process in the change-of-method option

2.4. Options menu

The system has been provided with an options menu where, apart from carrying out different types of changes in the file that has been selected for resolution, a series of useful possibilities are offered to the user. Among them, let us point out the following ones:

- Visualization of the *ideal* ($\mathbf{z}_i$*) and *anti-ideal* (m_i*) values of each objective function.
- Election of the *kind of normalización* of the objective functions, with the following options:
  - — no normalization;
  - — range normalization

$$\tilde{f}_i(x) = \frac{1}{\left|m_i^* - z_i^*\right|} f_i(x), \quad x \in X;$$

  - — L1 normalization

$$\tilde{f}_i(x) = \frac{1}{\sum_{j=1}^{n} \left| c_{ij} \right|} \cdot f_i(x), \quad x \in X;$$

  - — L2 normalization

$$\tilde{f}_i(x) = \frac{1}{\left(\sum_{j=1}^{n} c_{ij}^2\right)^{1/2}} \cdot f_i(x), \quad x \in X;$$

c_{ij} is the j-th coefficient of the i-th objective function.

– Free Normalization: the decision-maker can choose a positive quantity to divide each objective (which can be provided through the keyboard or in the text file) If these quantities are denoted by $D_1, ..., D_k$, then the normalized functions are:

$$\tilde{f}_i(x) = \frac{1}{D_i} \cdot f_i(x), \quad x \in X.$$

This option can be useful for the DMs when divisors D_j have some practical meaning which allows a better understanding of the values of the normalized function.

- *Historical file and statistic indicators*. The historical file is a file that contains all the relevant information corresponding to all the iterations that have been carried out so far. This information may be useful for the DM in some parts of the process, and it is formed by values (whether real or built by the system) of:
  - used method;
  - solutions in the criterion space;
  - solutions in the decision space;
  - local tradeoffs or weights;
  - tradeoffs corresponding to methods SPOT and ISWT;
  - projection weights from the ideal point to the criteria vector solution;
  - reference points;
  - reference points corresponding to the STOM methods.

The statistic indicators are the mean values and the variances of the previously mentioned values. This data lets the DM observe the dispersion degree of the information provided. During the first iterations, the dispersion degree is likely to be higher, showing that a great part of the feasible set is being explored. On the other hand, in the last iterations this degree should tend to decrease if the DM is close to his/her most preferred solution and his/her answers are consistent.

- *Criteria Structuring*. This option allows the user to establish a comfortable structure of the criteria, with the aim of making it easier for the decision-maker to provide the required information. Namely, the criteria can be grouped in different sublevels, which are organized in a hierarchical way (not in the sense of importance of the objectives, but attending a possible classification of them). Therefore, along the process it is possible to give information in different ways. For example, if the information required are local weights, the user can give these weights for any node of the hierarchical tree, that is, for certain groups of objectives. This process can be carried out at any stage of the process, independently from the algorithm that is being currently used.

3. The GLobal Interactive Decision Environment (GLIDE)

With the cooperation of Prof. K a i s a M i e t t i n e n (University of Jyvaskyla, Finland), we have developed a general formulation that covers thirteen interactive multiobjective optimization methods representing three (or actually four) different method types (L u q u e, R u i z, M i e t t i n e n, 2009). The advantage of this formulation is its simple and compact structure which enables easy implementation. Furthermore, the framework presented allows the DM to conveniently change the style of expressing preference information, that is, changing the method used. However, the DM is not supposed to know different multiobjective optimization methods and their specificities, but can concentrate on the actual problem to be solved and must only decide which kinds of preferences he/she could provide in order to direct the solution process to a desired direction so that he/she can identify the most preferred solution. Based on the preference type used, the general interactive solution scheme will choose the most appropriate method(s) for each case. The flexible possibility of changing the method means that the DM is not restricted to one way of specifying preferences. In different phases of the solution process the DM may wish to approach the problem in different ways and this is now possible. For example, at the early stages, in the so-called learning phase, the DM may wish to get a general overview of the solutions available and later on, once an interesting region of solutions has been identified, the DM may wish to fine-tune one's preferences in a smaller neighborhood. Our framework supports the DM in this and the DM has easy access to methods representing different solution philosophies.

3.1. The global formulation

The mathematical formulation of the GLIDE model takes the following form:

$$
\begin{array}{lll}
\min\limits_{x,\,\alpha} & \alpha + \rho \cdot \sum\limits_{i=1}^{k} \omega_i^h \left(f_i(x) - q_i^h \right) & \\
\text{s.a.:} & \mu_i^h \left(f_i(x) - q_i^h \right) \le \alpha & \forall i \in I_\alpha^h \\
 & f_i(x) \le \varepsilon_i^h + s_\varepsilon \cdot \Delta\varepsilon_i^h & \forall i \in I_\varepsilon^h \\
 & x \in X, &
\end{array}
\qquad \text{(GLIDE)}
$$

where $x \in R^k$ and $\alpha \in R$ are the variables. Besides, there are a series of real parameters ($\rho \ge 0$, $\omega_i^h \ge 0$, q_i^h, $\mu_i^h \ge 0$, ε_i^h, $s_\varepsilon \ge 0$ and $\Delta\varepsilon_i^h$) and two index sets, I_α^h and I_ε^h, which are subsets of $\{1, \ldots, k\}$. By changing the values of the parameters, the GLIDE formulation can be transformed into the (intermediate) single objective problems used by thirteen different interactive methods to generate the next iteration, and thus, the (weak, proper) efficiency of the corresponding optimal solution is guaranteed as in each original method. In general, all the efficient (or properly efficient) solutions of problem (MOP) can be obtained using this global scalarized formulation with adequate values for the parameters (for more details, see Luque, Ruiz, Miettinen, 2009).

3.2. Methods supported

The formulation described in Section 3.1 supports the following interactive multiobjective methods:

- Tradeoffs (or local weights) based methods:
  - ISWT (The Interactive Surrogate Worth Tradeoff, Chankong and Haimes, 1978).
  - SPOT (Sequential Proxy Optimization Technique, Sakawa, 1982).
  - PROJECT (Luque, Yang, Wong, 2009).
- Solutions Generating Methods:
  - Tchebychev method, (Steuer and Choo, 1983).
  - MICA method (Luque, Ruiz and Steuer, 2009).

- Reference Points Methods:
  - Reference Point Method, (W i e r z b i c k i, 1980).
  - VIA (Visual Interactive Approach, K o r h o n e n and L a a k s o, 1986).
  - GUESS method (B u c h a n a n, 1997).
- Classification Methods:
  - STEM (Step Method, B e n a y o u n et al., 1971).
  - STOM (Satisfacing Trade-Off Method, N a k a y a m a and S a w a r a - g i, 1984).
  - Reference Direction Algorithm (V a s s i l e v and N a r u l a, 1993).
  - NIMBUS (M i e t t i n e n and M ä k e l ä, 1995).
  - Modified Reference Point Method (V a s s i l e v et al., 2001).

In order to solve the intermediate single optimization problems used by these methods, the parameters of the (GLIDE) formulation have to be set accordingly. For example, Table 1 shows the values of the parameters for the Reference Direction Algorithm.

Table 1. Parameters in GLIDE formulation for the Reference Direction Algorithm

Index Sets	$I_\alpha^h = I_h^{\le}$	$I_\varepsilon^h = I_h^{=} \cup I_h^{\ge}$	
Weights	$\omega_i^h = 0 \ (i = 1, \ldots, k)$	$\mu_i^h = \dfrac{1}{f_i^h - \hat{q}_i^h}$ for $i \in I_h^{\le}$	$\rho = 0$
Reference levels	$q_i^h = \hat{q}_i^h$ for $i \in I_h^{\le}$		
Objective	$\varepsilon_i^h = f_i^h$ for $i \in I_h^{=}$	$\Delta\varepsilon_i^h = 0$ for $i \in I_h^{=}$	s_ε
bounds	$\varepsilon_i^h = \hat{q}_i^h$ for $i \in I_h^{\ge}$	$\Delta\varepsilon_i^h = f_i^h - \hat{q}_i^h$ for $i \in I_h^{\ge}$	

As it can be seen, the GLIDE formulation is simple and compact, and together with the tables of parameters for each method, it makes it possible to implement a global interactive system without the need of calling different subroutines for each method. This causes a greater computational efficiency of the system.

4. Concluding remarks

When designing a global interactive system, two issues have to be taken into account. First, the system must contain a sufficient number of interactive techniques, which allow the decision maker to choose the type of information he wishes to provide to the system at any moment during the solution process. This is the main feature of the interactive system PROMOIN, which has been designed in order to solve multiobjective programming problems using interactive methods. PROMOIN includes several interactive techniques, which are well known in literature, and which have been discussed with real applications. The change-of-method option can be very useful to solve real problems, given that it lets the DM give the information in a different way when he feels unable to keep on with the initial scheme. Besides, the system is able to build information for any method, based on the information give by the DM to other methods. This means that the information used for a given method is not completely lost when the user decides to change it. Finally, other options, like the historical information, the hierarchical objective structuring, the normalization schemes, etc., complete this implementation.

Second, the inner form of the interactive system is also important. Following this idea, a global interactive formulation (GLIDE) has been built, which can accommodate several interactive methods. The compact structure of this formulation takes the form of a general optimization problem with a set of parameters that have to be changed in order to obtain the different interactive methods supported. From the point of view of the programmer, the global formulation is complemented with tables with the values of the parameters of GLIDE for each of the methods considered. This provides a simple implementation framework that makes it easier to create an interactive system based on the GLIDE formulation.

Last but not least, it is important to point out that the success of an interactive method (or system) also lies on its user interface. A friendly, supportive and easy to understand interface is always a vital complement for a good interactive method. For example, Fig. 4 gives an example of user friendly interface for assigning the relative importance of the different criteria in the Multidecision1 interactive system (V a s s i l e v et al., 2005), and Fig. 5 shows one possible interface of www-NIMBUS (M i e t t i n e n and M ä k e l ä, 2000), which allows the user to compare the solutions obtained at several iterations.

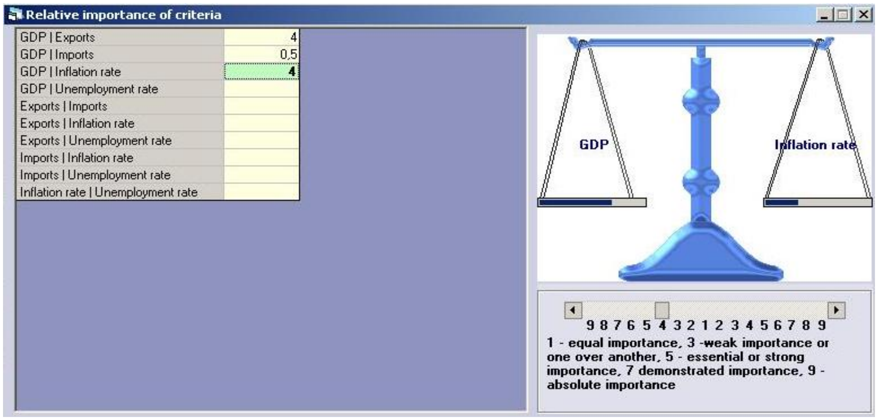

Fig. 4. Screen shot of the user interface of Multidecision1

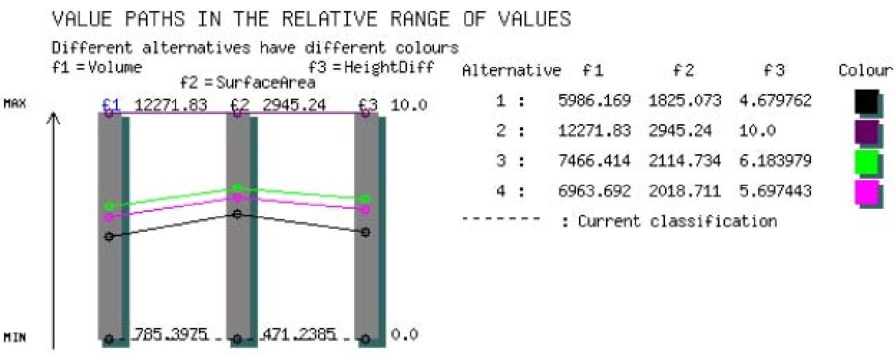

Fig. 5. Screen shot of the user interface of www-NIMBUS

In this paper, we have described some of the research results of the Multicriteria and sustainability group of the University of Málaga (Spain), some of which have been carried out in cooperation with Prof. K. Miettinen (University of Jyvaskyla, Finland), that have been carried out with the aim of building a global interactive decision system.

References

1. Aksoy, Y., T. W. Butler, E. D. Minor. Comparative Studies in Interactive Multiple Objective Mathematicas Programming. – European Journal of Operational Research, **89**, 1996, 408-422.
2. Benayoun, R., J. Montgolfier, J. Tergny, O. Laritchev. Linear Programming with Multiple Objective Functions: Step Method (STEM). – Mathematical Programming, **1**, 1971, 366-375.
3. Buchanan, J. T. A Naïve Approach for Solving MCDM Problems: The GUESS Method. – Journal of the Operational Research Society, **48**, 1997, 202-206.
4. Caballero, R., M. Luque, J. Molina, F. Ruiz. PROMOIN: An Interactive System for Multiobjective Programming. – International Journal of Information Technology and Decision Making, **1**, 2002, 635-656.
5. Chankong, V., Y. Haimes. The Interactive Surrogate Worth Trade-off (ISWT) Method for Multiobjective Decision-Making. – In: S. Zionts, Ed. Multiple Criteria Problem Solving. Lecture Notes in Economics and Mathematical Systems. Vol. **155**. Berlin, Heidelberg, Springer-Verlag, 1978, 42-67.
6. Dyer, J. S. Interactive Goal Programming (IGP). – Management Science, **19**, 1972, No 1, 62-70.
7. Gardiner, L., R. E. Steuer (1994a). Unified Interactive Multiple Objective Programming. – European Journal of Operational Research, **74**, 1994, No 3, 391-406.
8. Gardiner, L., R. E. Steuer (1994b). Unified Interactive Multiple Objective Programming: An Open Architecture for Accommodating New Procedures. – Journal of the Operational Research Society, **45**, 1994, No 12, 1456-1466.
9. Geoffrion, A. M., J. S. Dyer, A. Feinberg. An Interactive Approach for Multi-Criterion Optimization, with an Application to the Operation of an Academic Department. – Management Science, **19**, 1972, No 4, Part I, 357-368.
10. Jaszkiewicz, A., R. Slowinski. The "Light Beam Search" Approach (LBS) – An Overview of Methodology and Applications. – European Journal of Operational Research, **113**, 1999, 300-314.
11. Jones, D. F. Intelligent Solution and Analysis of Goal Programmes: The GPSYS System. – Decision Support Systems, **23**, 1998, No 4, 329-332.
12. Korhonen, P., J. Laakso. A Visual Interactive Method for Solving the Multiple Criteria Problem (VIA). – European Journal of Operational Research, **24**, 1986, No 2, 277-287.
13. Korhonen, P., J. Wallenius. A Pareto Race. – Naval Research Logistics, **35**, 1988, 615-623.
14. Luque, M. Multicriteria Decision Making with Interactive Methods. PhD Thesis, University of Málaga, Spain, 2000 (in Spanish).
15. Luque, M., R. Caballero, J. Molina, F. Ruiz. Equivalent Information for Multiobjective Interactive Procedures. – Management Science, **53**, 2007, No 1, 125-134.
16. Luque, M., F. Ruiz, K. Miettinen. Global Formulation for Interactive Multiobjective Optimization. OR Spectrum, accepted (doi:10.1007/s00291-008-0154-3), 2009.
17. Luque, M., F. Ruiz, R. E. Steuer. Modified Interactive Chebyshev Algorithm (MICA) for Convex Multiobjective Programming. – European Journal of Operational Research, accepted (doi:10.1016/j.ejor.2009.11.011), 2009.
18. Luque, M., J. B. Yang, B. Y. H. Wong. PROJECT Method for Multiobjective Optimization Based on the Gradient Projection and Reference Points. – IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans, Vol. **39**, 2009, No 4, 864-879.

19. M i e t t i n e n, K. Nonlinear Multiobjective Optimization. Massachusetts, Kluwer Academic Publishers, 1999.
20. M i e t t i n e n, K., M. M. M ä k e l ä. Interactive Bundle-Based Method for Nondifferentiable Multiobjective Optimization: NIMBUS. – Optimization, **34**, 1995, No 3, 231-246.
21. M i e t t i n e n, K., M. M. M ä k e l ä. Interactive Multiobjective Optimization System www-NIMBUS on the Internet. – Computers & Operations Research, **27**, 2000, 709-723.
22. NAG. Numerical Algorithm Group Limited: NAG C Library Manual. Mark 6. NAG Ltd., Oxford, 2000.
23. N a k a y a m a, H., Y. S a w a r a g i. Satisficing Trade-off Method for Multiobjective Programming. – In: M. Grauer, and A. P. Wierzbicki, Eds. Interactive Decision Analisys. Lecture Notes in Economics and Mathematical Systems. Vol. **229**. Laxenburg, Springer-Verlag, 1984, 113-122.
24. S a k a w a, M. Interactive Multiobjective Decision Making by the Sequential Proxy Optimization Technique: SPOT. – European Journal of Operational Research, **9**, 1982, No 4, 386-396.
25. S h i n, W. S., A. R a v i n d r a n. Interactive Multiple Objective Optimization: Survey I – Continuous Case. – Computers and Operations Research, **18**, 1991, No 1, 97-114.
26. S t e u e r, R. E., E. U. C h o o. An Interactive Weighted Tchebycheff Procedure for Multiple Objective Programming. – Mathematical Programming, **26**, 1983, 326-344.
27. S t e u e r, R. E. Multiple Criteria Optimization: Theory, Computation and Application. Wiley Series in Probability and Mathematical Statistics-Applied. New York, 1986.
28. V a s s i l e v, V., S. C. N a r u l a. A Reference Direction Algorithm for Solving Multiple Objective Integer Linear Programming Problems. – Journal of the Operational Research Society, **44**, 1993, No 12, 1201-1209.
29. V a s s i l e v, V., V. S g u r e v, A. A t a n a s s o v, A. D e i a n o v, M. K i c h o v i c h, L. K i r i l o v, V. D j a m b o v, G. V a c h k o v. Software Product for Multiobjective Nonlinear Programming: MONP-16. Version 1.1. General Description. User's Guide. Software Products and Systems Corporation (Institute of Industrial Cybernetics and Robotics, Sofia, 1990.
30. V a s s i l e v, V., S. C. N a r u l a, V. G. G o u l j a s h k i. An Interactive Reference Direction Algorithm for Solving Multi-Objective Convex Nonlinear Integer Programming Problems. – International Transactions in Operational Research, **8**, 2001, No 4, 367-380.
31. V a s s i l e v, V., K. G e n o v a, M. V a s s i l e v a. A Multicriteria Decision Support System Multidecision-1. – International Journal Information Theories & Applications, **13**, 2005, 103-111.
32. V a s s i l e v, V., M. V a s s i l e v a, B. S t a y k o v, K. G e n o v a, F. A n d o n o v, P. C h o n g o v a. MultiDecision-2: A Multicriteria Decision Support System. – International Journal "Information Technologies and Knowledge", **2**, 2008, No 1, 203-211.
33. V a s s i l e v a, M. Generalized Scalarizing Problems GENS and GENSLEX of Multicriteria Optimization. – International Journal Information Theories & Applications, **13**, 2005, 111-120.
34. W i e r z b i c k i, A. P. The Use of Reference Objectives in Multiobjective Optimization. – In: G. Fandel, T. Gal, Eds., Multiple Criteria Decision Making Theory and Application. Lecture Notes in Economics and Mathematical Systems. Vol. **177**. Heidelberg, Springer-Verlag, 1980, 468-486.
35. Z i o n t s, S., J. W a l l e n i u s. An Interactive Programming Method for Solving the Multiple Criteria Problem. – Management Science, **22**, 1976, No 6, 652-663.