Available online at www.sciencedirect.com
ScienceDirect
European Journal of Operational Research 180 (2007) 99–115
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/ejor

Discrete Optimization

# A review of interactive methods for multiobjective integer and mixed-integer programming

Maria João Alves *, João Clímaco

*Faculdade de Economia da Universidade de Coimbra, INESCC, Av. Dias da Silva, 165, 3004-512 Coimbra, Portugal*

Received 19 January 2005; accepted 16 February 2006
Available online 5 June 2006

**Abstract**

This paper makes a review of interactive methods devoted to multiobjective integer and mixed-integer programming (MOIP/MOMIP) problems. The basic concepts concerning the characterization of the non-dominated solution set are first introduced, followed by a remark about non-interactive methods vs. interactive methods. Then, we focus on interactive MOIP/MOMIP methods, including their characterization according to the type of preference information required from the decision maker, the computing process used to determine non-dominated solutions and the interactive protocol used to communicate with the decision maker. We try to draw out some contrasts and similarities of the different types of methods.


*Keywords:* Integer programming; Multiple objective programming; Interactive methods

## 1. Introduction

Multiobjective integer and mixed-integer programming (MOIP/MOMIP) is very useful for many areas of application as any model that incorporates discrete phenomena requires the consideration of integer variables (such as, for modeling investment choices, production levels, fixed charges, logical conditions or disjunctive constraints). However, research on methods for the general multiobjective integer/mixed-integer model has been rather limited when compared with multiobjective linear programming with continuous variables (MOLP) or methods (and heuristics) for particular multiobjective combinatorial problems.

The introduction of discrete variables into multiobjective programming problems turns these problems much more difficult to tackle, even if they are linear. The feasible set is no longer convex, and the resulting difficulties go beyond those of changing from mono-objective LP to IP. Thus, in most cases, the problems cannot be handled by adaptations to integer variables of MOLP methods. Further, there are multiobjective approaches designed for all-integer problems that do not apply to the mixed-integer case. Therefore, even for the linear case, techniques for dealing with multiobjective

* Corresponding author.
*E-mail address:* mjalves@fe.uc.pt (M.J. Alves).

0377-2217/$ - see front matter 
doi:10.1016/j.ejor.2006.02.033

integer/mixed-integer programming involve more than the combination of MOLP approaches with integer programming techniques.

In the last years, there have been many developments devoted to particular multiobjective combinatorial problems (such as, location, scheduling, knapsack, shortest-path problems, etc.). The researchers' attention has also focused on the use of meta-heuristics to solve these problems. In fact, there is nowadays a vast list of bibliography on meta-heuristic approaches, namely evolutionary algorithms, to tackle multiobjective combinatorial problems. For space reasons, it is not practicable to include in one review all these topics. We have opted for focusing on interactive methods for the general formulation of the MOIP and MOMIP problem due to their potentialities to solve real multiobjective problems with integer variables for which those special models are not suitable. Without having the pretence of being exhaustive, we present and compare several interactive methods published in the literature of the area, characterizing them according to the protocol of interaction with the decision maker (DM) and the computing process used to obtain non-dominated solutions.

Several survey articles have already been published in this area. Teghem and Kunsch (1986b) presented a survey of interactive methods for multiobjective integer and mixed-integer linear programming published until the final of 1985 (the first method dates from 1980). Covering the same period of time, another review is due to Rasmussen (1986) which focused on multiobjective 0–1 programming, considering both interactive and non-interactive methods. Also, Evans (1984) presented an overview of algorithms for multiobjective mathematical programming developed during the 15 years before. A brief overview of MOIP approaches can be found in Clímaco et al. (1997), and Alves and Clímaco (2001) presented an article on MOMIP. The review we present herein aims at giving an updated state-of-art on interactive MOIP/MOMIP methods. This paper refers to about 20 published interactive methods (although some of them are improvements of previous versions), from which only five were developed until 1985, the period covered by the articles of Teghem and Kunsch (1986b), Rasmussen (1986) and Evans (1984). Also, the more recent articles of Clímaco et al. (1997) and Alves and Clímaco (2001) do not include an extensive description of MOIP and MOMIP interactive methods. The former presents a classification of different algorithm approaches for multicriteria integer programming (but do not describe the functioning of each method), and the latter is a short article specific to the multiobjective mixed-integer case.

We hope this review can contribute to help and motivate researchers to continue the study and the development of this area, with undoubted importance and potentialities, but little active nowadays. The description and the comparison among the proposed methods aim at presenting their advantages and also their weaknesses, relevant issues to researchers and to whom wants to choose a method to tackle a real problem.

The paper is organized as follows. Section 2 presents basic concepts and general results concerning the characterization of the non-dominated solution set. Section 3 discusses non-interactive methods versus interactive methods regarding this type of problems. Section 4 reviews several interactive MOIP/MOMIP methods, presenting a summary of each method, and characterizing them according to the type of preference information required from the DM, the computing process used to determine the non-dominated solutions and the type of interactive protocol. A comparison of the mentioned interactive methods is made in Section 5. The paper ends with some concluding remarks in Section 6.

## 2. Basic concepts and general results

Consider the following multiobjective problem (P):

$$\begin{aligned} &\max z_1 = f_1(x) \\ &\cdots \\ &\max z_k = f_k(x) \\ &\text{s.t. } x \in X, \end{aligned}$$

where $X \subset \Re^n$ denotes the non-convex set of feasible solutions defined by a set of functional constraints, $x \geqslant 0$ and $x_j$ integer $j \in J \subseteq \{1,\ldots,n\}$. It is assumed that $X$ is compact (closed and bounded) and non-empty. The integer variables can either be binary or take on general integer values.

(P) is a MOIP problem if all variables are integer. Otherwise (P) denotes a MOMIP problem.

Models with linear constraints and linear objective functions have been more often considered than nonlinear cases. In linear multiobjective integer or mixed-integer problems (MOILP/MOMILP), the functional constraints can be defined as $Ax \leqslant b$, and the objective functions $f_i(x) = c^i x,\ i = 1, \ldots, k$ where $A$ is a $m \times n$ matrix, $b$ is a $m$-dimensional column vector and $c^i,\ i = 1, \ldots, k$, are $n$-dimensional row vectors.

The concept of efficiency or non-dominance in multiobjective (mixed-)integer programming is defined as usually for multiobjective mathematical programming.

A solution $\bar{x} \in X$ is *efficient* for the problem (P) if and only if there is no $x \in X$ such that $f_i(x) \geqslant f_i(\bar{x})$ for all $i \in \{1, \ldots, k\}$ and $f_i(x) > f_i(\bar{x})$ for at least one $i$.

A solution $\bar{x} \in X$ is *weakly-efficient* for the problem (P) if and only if there is no $x \in X$ such that $f_i(x) > f_i(\bar{x})$ for all $i \in \{1, \ldots, k\}$.

Let $Z \subset \Re^k$ be the image of the feasible region $X$ in the objective functions (criteria) space. A point $\bar{z} \in Z$ corresponding to a (weakly) *efficient* solution $\bar{x} \in X$ is called (weakly) *non-dominated*.

Since the feasible region of (P) is non-convex, *unsupported non-dominated* solutions may exist. A non-dominated point $\bar{z} \in Z$ is called unsupported if it is dominated by a convex combination (which does not belong to $Z$) of other non-dominated criteria points (belonging to $Z$).

Hence, unlike in MOLP, the non-dominated solution set of a problem (P) cannot be fully determined by parameterizing on $\lambda$ the *weighted-sum* program:

$$\max \quad \sum_{i=1}^{k} \lambda_i f_i(x) \qquad (\mathrm{P}_\lambda)$$
$$\text{s.t.} \quad x \in X,$$

where $\lambda \in \Lambda = \{\lambda \in \Re^k | \lambda_i > 0\ \forall i,\ \sum_{i=1}^{k} \lambda_i = 1\}$.

The unsupported non-dominated solutions cannot be reached even if the complete parameterization on $\lambda$ is attempted.

Researchers on multiobjective mathematical programming early recognized this fact and stated other characterizations for the non-dominated set that fit the multiobjective integer/mixed-integer cases.

Basically, two main characterizations can be defined.

### 2.1. Weighted-sum programs with additional constraints

This characterization of the non-dominated solution set consists of introducing additional constraints into the *weighted-sum* program ($\mathrm{P}_\lambda$). Generally, these constraints impose bounds on the objective function values, which can be regarded as a particularization of the general characterization provided by Soland (1979). The introduction of bounds on the objective function values enables the *weighted-sum* program to also compute unsupported non-dominated solutions. This scalarizing program can be stated as follows:

$$\max \quad \sum_{i=1}^{k} \lambda_i f_i(x) \qquad (\mathrm{P}_{\lambda,\mathrm{g}})$$
$$\text{s.t.} \quad x \in X,$$
$$f(x) \geqslant g,$$

where $\lambda \in \Lambda$, $f(x) = (f_1(x), \ldots, f_k(x))$ and $g$ is a row vector of objective bounds.

Besides the fact that every solution obtained by ($\mathrm{P}_{\lambda,\mathrm{g}}$) is non-dominated, there always exists a $g \in \Re^k$ such that ($\mathrm{P}_{\lambda,\mathrm{g}}$) yields a particular non-dominated solution.

### 2.2. Reference point based scalarizing programs

Other characterizations based on reference points can be defined.

The Tchebycheff theory, whose foundation originated from Bowman (1976), integrates this type.

Let us denote by $\|\bar{f} - f(x)\|_w$ the $w$-weighted Tchebycheff metric, i.e., $\max_{1 \leqslant i \leqslant k}\{w_i|\bar{f}_i - f_i(x)|\}$, where $w_i \geqslant 0\ \forall i, \sum_{i=1}^{k} w_i = 1$, and $\bar{f}$ denotes a reference point of the criteria space. Considering $\bar{f} > f(x)$ for all $x \in X$, Bowman (1976) proved that the parameterization on $w$ of $\min_{x \in X}\|\bar{f} - f(x)\|_w$ generates the non-dominated set.

The program $\min_{x \in X}\|\bar{f} - f(x)\|_w$ may yield weakly non-dominated solutions, which can be avoided by considering the *augmented weighted Tchebycheff program*: $\min_{x \in X}\{\|\bar{f} - f(x)\|_w - \rho \sum_{i=1}^{k} f_i(x)\}$, with $\rho$ a small positive value. This program can also be written as follows:

$$\min \quad \left\{ \alpha - \rho \sum_{i=1}^{k} f_i(x) \right\} \qquad (\mathrm{T_w})$$
$$\text{s.t.} \quad w_i(\bar{f}_i - f_i(x)) \leqslant \alpha, \quad 1 \leqslant i \leqslant k,$$
$$x \in X,$$
$$\alpha \geqslant 0.$$

Steuer and Choo (1983) proved that there always exists $\rho$ small enough that enable to reach all the non-dominated set for the finite-discrete and polyhedral feasible region cases. Concerning the mixed-integer case, there may be portions of the non-dominated set (nearby weakly non-dominated solution) that ($T_w$) is unable to compute, even considering $\rho$ very small (an example is shown in Alves and Clímaco, 2001). However, this characterization is still possible in practice, because $\rho$ can be set so small that the DM is unable to discriminate between those solutions and nearby weakly non-dominated solutions.

In Steuer and Choo (1983) and Steuer (1986) a lexicographic weighted Tchebycheff program is proposed for nonlinear and infinite-discrete feasible region cases to overcome this drawback of the augmented weighted Tchebycheff program. The lexicographic approach can also be applied to the mixed-integer (linear or nonlinear) case. However, it is more difficult to implement since two stages of optimization are employed. At the first stage only $\alpha$ is minimized. When the first stage results in alternative optima, a second stage is required. It consists of minimizing $-\sum_{i=1}^{k} f_i(x)$ (i.e., max $\sum_{i=1}^{k} f_i(x)$) over the solutions that minimize $\alpha$ in order to eliminate the weakly non-dominated solutions.

Besides ($T_w$) (either the augmented or the lexicographic forms), there are other approaches based on reference points that allow to characterize the non-dominated set of multiobjective integer/mixed-integer programs. An approach of this type consists in discarding the $w$-vector or fixing it and varying $\bar{f}$, the criteria reference point, which may represent DM's aspiration levels. We shall refer to this scalarizing program as $(T_{\bar{f}})$. There always exist reference points satisfying $\bar{f} > f(x)\ \forall x \in X$, such that $(T_{\bar{f}})$ produces a particular non-dominated solution $\bar{z} = f(\bar{x})$. Reference points that do not satisfy the condition $\bar{f} > f(x)\ \forall x \in X$, can also be considered, provided that the $\alpha$ variable is defined without sign restriction. This corresponds to the minimization of a distance from $Z$ to the reference point if the latter is not attainable and to the maximization of such a distance if the reference point is attainable. If reference or aspiration levels are used as controlling parameters, the (weighted) Tchebycheff metric changes its form of dependence on controlling parameters and should be interpreted as an *achievement scalarizing function* (Lewandowski and Wierzbicki, 1980).

Like the simplest form of ($T_w$), the simplest form of $(T_{\bar{f}})$ may produce weakly non-dominated solutions, but the augmented form is a good substitute in practice and the lexicographic approach guarantees that all non-dominated solutions can be reached. In what follows, we shall use ($T_w$) and $(T_{\bar{f}})$ to denote either the corresponding simplest or augmented forms.

## 3. Non-interactive versus interactive methods

Although providing very important theoretical results, the characterizations of the non-dominated set do not offer explicit means to provide decision support for MOIP/MOMIP problems. Consequently, some researchers have developed methods for handling these types of problems.

Methods may be *non-interactive* – in general, *generating* methods designed to find the whole set or a subset of the non-dominated solutions – or *interactive* – characterized by phases of human intervention alternated with phases of computation.

*Generating* methods that are designed to generate the whole set or a large subset of non-dominated solutions may require an excessive amount of computational resources, namely in processing time, which may be inadequate to deal with large problems. Further, if a large set of alternatives is presented to the DM at the final of the procedure, this will raise additional difficulties to the DM to analyze all the information and make a final choice. However, there are some approaches that attempt to find a representative subset of the non-dominated set – generating methods according to the above definition – that could be easily embodied in an interactive framework. Solanki's biobjective method (Solanki, 1991) may be regarded as an example of such an approach (due to this fact, we include this method in the list of interactive methods presented in the next section). Therefore, the border between a generating method and an interactive method can sometimes be tenuous. A generating method can be used, for instance, to obtain a set of well-distributed non-dominated solutions to be presented to the DM. The DM can then select bounds for the objective functions, and a new MOIP/MOMIP problem is considered including those bounds, for which a generating method will be used again. This process can be repeated by narrowing or relaxing the bounds. This is just an example of how a generating method can be integrated into an interactive process.

Most *generating* methods for MOIP/MOMIP were developed in the decades of 70 and 80. The following list includes some well-known methods of this type (most of them restricted to linear cases):

- Pasternak and Passy (1973), Bitran (1977, 1979), Deckro and Winkofsky (1983), Bitran and Riviera

(1982), Kiziltan and Yucaoglu (1983) and White (1984) – for multiobjective 0–1 programming;
- Klein and Hannan (1982), Villareal and Karwan (1981), Chalmet et al. (1986) and Sylva and Crema (2004) – for multiobjective all-integer programming.
- Mavrotas and Diakoulaki (1998) – for multiobjective mixed 0–1 programming.

We observe that most generating methods are devoted to problems with 0–1 variables, which is understandable because it is easier to use enumeration techniques in 0–1 problems than in general integer or mixed-integer problems. Some methods are specialized to biobjective problems, such as the methods of Pasternak and Passy (1973) and Chalmet et al. (1986), which can profit from graphical representations on the criteria space.

Some generating methods use a constructive process, which add successively new solutions to the efficient/non-dominated set. This is the case of the methods of Pasternak and Passy (1973), Bitran (1977, 1979), Klein and Hannan (1982), White (1984), Chalmet et al. (1986) and Sylva and Crema (2004) (which is a variation of the Klein–Hannan algorithm).

Other methods operate with *potentially* non-dominated solutions in the intermediate phases of the process, and only at the end of the process the true non-dominated set is known – e.g. Villareal and Karwan (1981), Deckro and Winkofsky (1983) and Kiziltan and Yucaoglu (1983).

In the Teghem and Kunsch (1986a) survey of methods to characterize the set of efficient solutions of MOILP, the authors conclude that it might not be wrong to say that none of those methods really copes well with large dimensional problems, and this is not too surprising due to the complexity of the MOILP structure.

It must also be stressed that there are few generating methods devoted to mixed-integer programming. The work of Mavrotas and Diakoulaki (1998) is a good example but it just concerns the mixed 0–1 case. The method consists of generating and saving potentially non-dominated solutions and making pairwise comparisons to successively eliminate the dominated ones. Thus, at the final of the process only the non-dominated solutions remain. The computational results presented in this paper illustrate well the computational effort involved in the generation of all non-dominated solutions. Therefore, the authors suggest the intervention of the DM to reduce the scope of the search, namely by imposing bounds on the objective function values and defining filters for the generating process.

From the beginnings of the 1980s decade, the researches have paid more attention to the development of *interactive* methods to deal with MOIP/MOMIP problems (mainly, linear cases) in order to overcome the principal difficulties of generating methods. Interactive methods enable to reduce the computational effort and aid the DM in the decision process. In interactive methods, the set of non-dominated solutions is explored by a progressive articulation of the DM's preferences. This is a shared feature of all interactive methods, but there are different paradigms followed by the authors.

Some authors admit that the DM's preferences can be represented by an *implicit utility function*. Then, the interactive process aims to 'discover' the optimum (or an approximation of it) to that implicit utility function. The convergence to this optimum generally requires no contradictions in the DM's responses given throughout the interactive process.

In contrast with implicit utility function approaches, there are other approaches aiming at a progressive and selective *learning* of the non-dominated solution set, which use an *open communication* protocol to interact with the DM (this terminology is inspired on the concept of open exchange defined by Feyerabend, 1975). Those multiobjective approaches are not intended to converge to any 'best' solution, but to help the DM to avoid the search for non-dominated solutions he/she is not at all interested in, and to help in the identification of satisfactory compromise solution(s). There are no irrevocable decisions during the whole process and the DM is always allowed to go 'backwards' at a later interaction. So, at each interaction, the DM is only asked to give some indications on what direction the search for non-dominated solutions must follow, or occasionally to introduce additional constraints. The process finishes when the DM considers to have gathered sufficient insight into the non-dominated solution set. Using Roy's terminology (Roy, 1987), *convergence* gives place to *creation*. The interactive process is a constructive process, not the search for something 'pre-existent'.

In the next section, we include a tentative classification of each method based on these paradigms: the assumption of an *implicit utility function* vs. an *open communication* protocol.

## 4. Interactive methods for MOIP/MOMIP

In the following review we adopt the following taxonomy to group the methods: *biobjective* and *multiobjective*. We distinguish between these two classes since biobjective methods are naturally less applicable than multiobjective methods.

There are methods that can only handle all-integer problems and others that can also handle mixed-integer problems. For each class (*biobjective/multi-objective*) we first present the *integer* methods and then the *mixed-integer* methods.

Also, a distinction can be made between the methods that can only deal with linear cases and methods that can be applied to nonlinear cases. In this review we have concentrated on methods for linear problems, although some of the presented methods can also tackle nonlinear problems. These cases will be explicitly mentioned below.

We try to classify each method according to the type of procedure used to compute non-dominated solutions – *type* (a), *weighted-sum programs with additional constraints*, or *type* (b), *reference point based scalarizing programs* – and according to the type of protocol used to interact with the DM – the assumption of an *implicit utility function* or an *open communication* protocol. Besides this classification, we describe summarily the general functioning of each method.

Not claiming to be exhaustive, we shall present some methods representative of different approaches, which have been published in well-divulged journals of the area.

### 4.1. Biobjective integer and mixed-integer programming

- Ramesh et al. (1990) proposed an interactive method devoted to biobjective integer linear programming (BILP) which assumes an implicit *utility function* of the DM, pseudo-concave and non-decreasing. The method employs a modified version of the MOLP method of Zionts and Wallenius (1983) within a branch-and-bound framework. The DM's preference structure is assessed using pairwise comparisons. This method can be classified as *type* (a) in what concerns the computation of non-dominated solutions (*weighted-sum programs with additional constraints*).

The method of Ramesh et al. begins by relaxing the integrality conditions of the variables and applies the Zionts–Wallenius method to the linear relaxation of the BILP problem. If the 'optimal' solution to the utility function is integer, then it is also 'optimal' to the original problem. Otherwise, a branch-and-bound search for the integer 'optimum' is conducted, beginning with an initial incumbent integer solution, $z^1 = (z_1^1, z_2^1) \in Z$, obtained using a heuristic procedure. This solution is used to divide the relaxed feasible region into two subregions, $X_1$ and $X_2$, where $X_1$ is obtained by introducing the constraint $f_1(x) \geqslant z_1^1$, and $X_2$ is obtained by introducing ($f_1(x) \leqslant z_1^1 - \varepsilon$ and $f_2(x) \geqslant z_2^1$), with $\varepsilon$ a small positive value. A branch-and-bound search is then performed, separately, in $X_1$ followed by a search in $X_2$. If the search in $X_1$ finds an integer solution that dominates the incumbent solution, the partitions are tightened redefining $X_1$ and $X_2$ as previously but considering $z^2$, the solution that dominates $z^1$, in place of $z^1$. The branch-and-bound search in each partition generates candidate problems for investigation by branching on a fractional variable from a given solution. These biobjective linear relaxation problems are solved using the strategy of the Zionts–Wallenius method. The incumbent solution is updated whenever the 'optimal' solution of a candidate problem is preferred to the current incumbent solution.

- Shin and Allen (1994) presented an interactive method for biobjective integer (linear and nonlinear) programs, with concave objective functions and a convex feasible region (apart from the integrality constraints). This method aims at converging to the best compromise solution of an implicit *utility function* of the DM. The method excludes successively search regions by imposing constraints on the objective function values, which result from pairwise comparisons of non-dominated solutions performed by the DM. At each phase, the method determines the supported non-dominated solution closer to an already known non-dominated solution (to the right or to the left). This is called an ASN (associated supported non-dominated) solution to the previous one, and a *particular technique* is used to compute it. Given a non-dominated point $z^s$, the ASN point to the right of $z^s$ is obtained graphically as follows: a horizontal line is drawn extending from $z^s$ and it is swung downward to the right until it intersects a feasible point – this is the ASN point to the right of $z^s$ (see Fig. 1). Mathematically, an ASN point is obtained by optimizing an auxiliary (nonlinear) single objective problem.

The interactive algorithm begins by determining the non-dominated solution that maximizes $f_2(x)$,

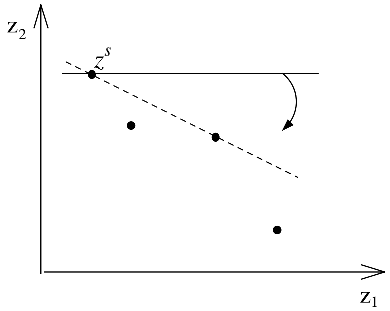


Fig. 1. Locating an ASN point in the method of Shin and Allen.

which is the initial incumbent non-dominated solution. An iterative procedure is then employed with the following steps: (1) Determine an ASN point to the current incumbent solution; if none exists (to the right or to the left), terminate; else, go to the next step. (2) Let $z^s$ and $z^t$ be the incumbent solution and the corresponding ASN, or vice-versa, such that $z^s$ is on the left of $z^t$ (i.e. $z_2^s > z_2^t$). If the DM prefers $z^s$ to $z^t$, impose the constraint $f_2(x) > z_2^t$; if the DM prefers $z^t$ to $z^s$, impose the constraint $f_1(x) > z_1^s$; if the DM is indifferent between them, impose $f_1(x) \geqslant z_1^s$ and $f_2(x) > z_2^t$. Update the incumbent solution with the preferred solution and return to (1).

- Aksoy (1990) developed an interactive method for biobjective mixed-integer programs, which employs a branch-and-bound scheme to divide the subset of non-dominated solutions of each node into two disjointed subsets. The branching process seeks to bisect the range of non-dominated values of $z_2$ at the node under consideration, checking whether a non-dominated point exists whose $z_2$ value is in the middle of the range. So, if $[l_2^j, u_2^j]$ is the range of values of $z_2$ in the node $N^j$, the procedure attempts to obtain a non-dominated solution such as $z_2 = (l_2^j + u_2^j)/2$. This solution is used to divide the non-dominated set of $N^j$ into two disjointed subsets. If no such solution exists, the set is divided using two non-dominated points whose values for $z_2$ are the closest (one up and the other down) to the middle value. These non-dominated solutions are obtained by optimizing one objective function and bounding the other. Hence, this method uses weighted-sums, just considering weights $(0, 1)$ and $(1, 0)$, imposing additional constraints on the criteria values (a particular form of $(P_{\lambda,g})$ – *type* (a)). It considers a lexicographic approach to ensure that just efficient solutions are obtained. The interactive process requires the DM to make pairwise comparisons in order to determine the branching node and to adjust the incumbent solution to the preferred non-dominated solution. It assumes that the DM's preferences are consistent, transitive and invariant over the process aiming to optimize the DM's implicit *utility function*.

- The method of Solanki (1991), which was designed to biobjective mixed-integer linear programs, is an adaptation of the non-inferior set estimation (NISE) method developed by Cohon for biobjective linear programs. The method seeks to generate a representative subset of non-dominated solutions by combining the NISE's key features with weighted Tchebycheff scalarizing programs (*type* (b)).

The NISE method aims at obtaining an approximate representation of the non-dominated set by computing successively non-dominated solutions that optimize weighted-sums of the objective functions. The segment joining a pair of solutions, say $z^a$ and $z^b$, can be considered a good approximation of that region if the 'error' of the approximation is within a predefined error bound. The 'error' is estimated by the length of the perpendicular of the segment $z^a z^b$ to an upper bound of the non-dominated set. The measure of error used by the NISE method is based on the convexity of the feasible region. Therefore, it is no longer valid in integer or mixed-integer programs. Moreover, the weighted-sum programs, used in NISE to generate non-dominated solutions, cannot capture unsupported solutions. These difficulties led Solanki to adopt the augmented weighted Tchebycheff program, reformulating the measure of the 'error'. At each iteration of the Solanki's method, a new non-dominated solution, say $z^c$, is computed by solving $(T_w)$ for specific $w$ and $\bar{f}$ vectors, which ensure that $z^c$ belongs to the region between a pair of non-dominated criteria points previously determined, say $(z^a, z^b)$. This pair is then replaced by $(z^a, z^c)$ and $(z^c, z^b)$. The approximation of the non-dominated surface is progressively improved, thus decreasing the 'errors' associated with the approximate representation of the pairs. An 'error' is measured by the largest range of the two objectives for the points forming the pair, i.e., the 'error' of the pair $(z^a, z^b)$, $z_1^a > z_1^b$ is given by $\max\{(z_1^a - z_1^b)/R_1, (z_2^b - z_2^a)/R_2\}$, where $R_1$ and $R_2$ are scale factors. The algorithm finishes when the largest 'error' is within a predefined error bound.

This is not an interactive method, since the DM just has to specify the error bound. However, in our opinion, this method could be easily embodied in an interactive framework, in which the DM could choose the pair of solutions to analyze, in each iteration, and decide the continuation or the end of the algorithm. This is the reason why we include the method in this review of interactive methods.

- Ferreira et al. (1996) developed a decision support system for biobjective mixed-integer programs, in which the interactive process follows an *open communication* protocol, aiming at a progressive and selective learning of the non-dominated solution set.

At each interaction, the DM chooses a pair of non-dominated solutions to further explore the non-dominated region between them, starting (in the first interaction) with the pair of non-dominated solutions that optimize individually each objective function. The DM can also specify bounds directly on the objective function values. A weighted-sum program is then optimized in the region selected by the DM, which is reduced in relation to the original feasible region by constraints on the objective function values (method of *type* (a)). The knowledge of new non-dominated solutions enables to eliminate progressively some criteria regions, either by dominance or unfeasibility. Later, Ferreira (1997) proposed the use of the Tchebycheff metric (*type* (b)) instead of weighted-sums. The author observed that this approach has the advantage over the former of enabling to eliminate larger regions by unfeasibility.

### *4.2. Multiobjective integer and mixed-integer programming*

- Marcotte and Soland (1980, 1986) presented an interactive method devoted to multiobjective problems for which the feasible set is either convex or discrete. It is not, however, applicable to the mixed-integer case. The method requires that the DM specifies his/her preferences between pairs of solutions, and assumes that they are stable and follow an appropriate mathematical structure (not requiring the existence of an implicit *utility function*). The algorithm follows a branch-and-bound scheme, splitting the multiobjective problem into sub-problems by introducing constraints on the objective function values. The non-dominated solutions are computed by optimizing weighted-sums of the objective functions in the feasible subset corresponding to each sub-problem. Hence, the method can be classified as *type* (a) in what concerns the computation of non-dominated solutions.

Let $N^j$ be the node $j$ of the branch-and-bound tree corresponding to criteria feasible region $Z^j$ and the ideal point $\beta^j$ (which is computed when $N^j$ is created). To analyze $N^j$, the algorithm first optimizes a weighted-sum of the objective functions to determine a non-dominated solution $z^j \in Z^j$. If $z^j \neq \beta^j$, the node is branched into many nodes as the number of components $i$ such that $z_i^j < \beta_i^j$. In the branch $i$ of $N^j$ the constraint $f_i(x) \geqslant z_i^j + \delta_i$ with $\delta_i > 0$ is added. Hence, the feasible regions of the descendants of $N^j$ are not necessarily disjointed. The ideal points give upper bounds for the criteria values of the corresponding non-dominated sets, and the DM is asked to arrange them in a list of decreasing preferential order. This list defines the order for selecting nodes to be analyzed. Designating by incumbent solution the non-dominated solution preferred by the DM, among all those found thus far, a node will be fathomed if its ideal point is not preferred to the incumbent solution. Therefore, the algorithm terminates if the list is empty, or the ideal point of the first node is feasible, or even if it is not preferred to the incumbent solution.

- White (1985) presented a method based on the method of Marcotte and Soland, which implements an extension of Lagrangean relaxation from scalar problems to vector problems. As the former, this method applies to multiobjective integer problems. The Lagrangean technique aims at finding better bounds for the criteria values, which assist in the elimination of 'non-optimal' solutions to the implicit *utility function* of the DM (a monotonic increasing function).

As mentioned above for the Marcotte and Soland method, the non-dominated incumbent solution is compared with the ideal point of each node of the branch-and-bound tree, since the ideal point gives upper bounds for the criteria values of that branch. The Lagrangean method aims at narrowing those bounds in order to eliminate more nodes of the tree that would be uninteresting to the DM. This method can also be classified as *type* (a) in what concerns the computation of non-dominated solutions.

- A different approach for dealing with multiobjective integer linear problems was proposed by Gonzalez et al. (1985). This method consists of two stages: the first stage only computes supported efficient solutions, by optimizing weighted-sums of the objective functions. The second stage computes unsupported efficient solutions, using weighted-sums of the objective functions with additional constraints that eliminate solutions already known (*type* (a)). The information required from the DM at each interaction consists of selecting the least preferred solution from a reduced set of candidate efficient solutions. The method assumes the existence of an implicit *utility function* of the DM.

In the first stage, the procedure starts by computing the $k$ non-dominated solutions that optimize individually each objective function, forming the set $N^*$ of criteria points. If the DM wants to continue the search, the hyperplane that passes through the $k$ points of $N^*$ is defined and also the corresponding weighted-sum of the objective functions whose gradient is perpendicular to this hyperplane (if the weights are not all positive, a perturbation is made to ensure this condition). The non-dominated point $z^j$ that optimizes this weighted-sum is then determined. If $z^j \neq N^*$ and it is preferred to at least one point of $N^*$, then $z^j$ will take the place of the least preferred point of $N^*$, and the computing procedure is repeated. Otherwise, the stage 1 terminates. The stage 2 computes unsupported efficient solutions. At each iteration, it optimizes the weighted-sum of the objective functions, $F(x)$, whose gradient is perpendicular to the hyperplane that passes through the $k$ points of $N^*$, considering the additional constraint $F(x) \leqslant P$ (where $P$ is obtained by subtracting a fractional amount to the constant of the supporting hyperplane). This procedure can be repeated by diminishing $P$ in order to turn unfeasible the points already generated. The stage 2 terminates whenever the DM desires or the solution obtained is dominated by any other solution.

• Gabbani and Magazine (1986) presented an interactive approach to multiobjective integer linear problems, in which the solutions are obtained heuristically. The method is an adaptation of the Interval Criterion Weights method of Steuer (1977, 1986) for MOLP. The method of Gabbani and Magazine assumes that the DM has an implicit *linear utility function*. It uses (*simple*) *weighted-sum programs*, discarding from consideration unsupported efficient solutions.

The Interval Criterion Weights method of Steuer can be summarized as follows: (i) select $2k+1$ weight vectors from the current weight space (which is $\Lambda$ in the first iteration) according to a selection rule, and solve the corresponding $2k+1$ weighted-sum programs; (ii) ask the DM to select the most preferred solution from the resulting solutions; (iii) contract the weight space around the weight vector which produced the preferred solution. Steps from (i) to (iii) are repeated until a stopping condition is verified. In the proposal of Gabbani and Magazine, the integer programs of (i) are solved heuristically, in order to reduce the computational effort. Consequently, there is no guarantee of computing true efficient solutions or just near-efficient solutions. The computational experience presented by the authors concerns 0–1 multidimensional knapsack problems for which a specific heuristic (developed by Magazine and Oguz) is employed.

Concerning interactive methods for multiobjective mixed-integer linear programming (MOMILP), a first reference is addressed to the family of the methods that are extensions of the MOLP Zionts–Wallenius method (like the above-mentioned biobjective method of Ramesh et al. (1990)).

• The first version is due to Villarreal et al. (1980). This method received later improvements by Karwan et al. (1985) and Ramesh et al. (1986). Starting by applying the Zionts–Wallenius algorithm to the linear relaxation of the MOMILP problem, the method then employs a branch-and-bound phase until an integer solution that satisfies the DM is achieved. An implicit *utility function* is assumed and the DM's preferences are assessed using pairwise evaluations of decision alternatives and tradeoff analysis. In light of the DM's underlying utility function, decisions on whether (i) to apply again the Zionts–Wallenius procedure to the linear relaxation of a candidate multiobjective sub-problem or (ii) to continue branching by appending a constraint on a variable that does not satisfy the integrality condition, are successively made. A scalarizing program, which consists of the weighted-sum program combined with additional constraints (that delimit the feasible region of each node), is used for computing non-dominated solutions in this interactive branch-and-bound method (*type* (a)).

• Steuer and Choo (1983) developed a general-purpose interactive method, which is applicable to multiobjective integer and mixed-integer programs (including nonlinear cases). The method assumes an implicit DM's *utility function* without any special restriction on shape (except that it must be coordinatewise increasing in criteria space). The strategy of the interactive procedure is to sample series of progressively smaller subsets of non-dominated solutions.

In each interaction, the DM selects his/her preferred solution from a sample of non-dominated solutions. These solutions are obtained by solving the augmented (or the lexicographic) weighted Tchebycheff program $(T_w)$ with several dispersed $w$-vectors over the current weight space (considering the ideal criteria point as the reference point $\bar{f}$) – method of *type* (b). The solution preferred by the DM at the iteration $h$ provides information to define $w^{(h)}$, which is used to reduce the set of $w$-vectors for the following iteration (concentrating the weight space around $w^{(h)}$). The reduction of the weight space depends on a convergence factor defined a priori. The procedure terminates after a predefined number of iterations.

• Another interactive method capable of solving multiobjective integer and mixed-integer linear programming problems was developed by Durso (1992). This method is a modification of the interactive branch-and-bound method of Marcotte and Soland (1986). Unlike the method of Marcote and Soland, this method is able to deal with mixed-integer programs due to the modification on the approach used to compute non-dominated solutions, the main difference between the two methods. The method of Durso uses the augmented weighted Tchebycheff metric – *type* (b).

The method of Durso employs a branching scheme considering progressively smaller portions of the non-dominated set by imposing lower bounds on the criteria values. For each node $N^j$ of the branch-and-bound tree, the $k$ non-dominated solutions that define the ideal point $\beta^j$ are first calculated. The DM is then asked to select the node for analysis by choosing his/her preferred ideal point. The analysis process begins by solving an equally weighted augmented Tchebycheff program to determine a "centralized" non-dominated point for the subset of the node under exploration. Once the DM chooses the most preferred solution of the $k+1$ non-dominated points already known for this node, say $\hat{z}$, up to $k$ new nodes (children) are created, many as the number of elements of the set $\{i|\hat{z}_i < \beta^j - \delta_i\}$ with $\delta_i$ $(i = 1,\ldots,k)$ specified by the DM. Each child inherits its parent's bounding constraints and uses $\hat{z}$ to further restrict the bound for one objective function. Thus, the $i$th child restricts the $i$th criterion by imposing $f_i(x) \geqslant \hat{z}_i + \delta$ with $\delta$ small positive. This approach may be regarded as an *open communication* procedure that terminates when the DM is satisfied with the incumbent solution – the preferred non-dominated solution from those obtained so far.

• Karaivanova et al. (1993) proposed an adaptation of the method of Steuer and Choo (1983) for multiobjective integer and mixed-integer linear programming problems, in which the augmented weighted Tchebycheff scalarizing programs are solved heuristically. Therefore, the obtained solutions may be efficient or just near-efficient solutions. In this paper, the authors aim at comparing the computational results of both methods, defining a *utility function* for each multiobjective problem randomly generated. They observed that the exact procedure (of Steuer and Choo) was faster in small problems (with 3–6 integer variables, 15–30 constraints are 3–6 objective functions) but the proposed heuristic procedure outperformed the former when the number of variables was increased (considering problems with 30 integer variables, three constraints and three objectives). In these tests, a particular heuristic for mixed-integer programming was used.

• Steuer et al. (1993) proposed a multiobjective generic interactive approach. Although it has not been designed specifically to MOIP/MOMIP problems, and the reported computational experience refers to MOLP problems, we can say that it is generic enough to fit this type of problems if an adequate algorithm to solve the integer/mixed-integer scalarizing programs is incorporated.

The proposed approach combines the Tchebycheff method of Steuer and Choo (1983) with the Wierzbicki's aspiration criterion vector method (Wierzbicki, 1982, 1986; Lewandowski and Wierzbicki, 1980; among other references) in order to form an improved procedure for interactive multiobjective programming. Both methods solve the augmented Tchebycheff scalarizing program to compute non-dominated solutions (*type* (b)), but the information inputted by the DM is different in the two methods. The aspiration criterion vector method is oriented by *aspiration vectors* for the objective functions specified by the DM. The authors argue that the Tchebycheff philosophy of Steuer and Choo (1983) is likely to be most useful in the early iterations because of its dispersed sampling, and the aspiration criterion vector philosophy is likely to be most useful in the later iterations when the DM is attempting to pinpoint a final solution.

• L'Hoir and Teghem (1995) presented an interactive method, called MOMIX, which was specially designed to multiobjective mixed-integer linear programming, although a MOLP problem was considered in the application presented in this paper. The underlying principle of the method MOMIX is the use of an interactive branch-and-bound, whose philosophy was introduced by Marcotte and Soland (1986). For each node, a non-dominated solution is determined as in the classical STEM method (Benayoun et al., 1971), i.e. by minimizing a weighted Tchebycheff function to the ideal solution of that node (*type* (b)). The interactive branch-and-bound of MOMIX includes two steps: a "depth first" progression in the tree, which aims at determining a first good compromise solution, and a "backtracking" step to confirm the degree of satisfaction achieved by the DM or to find a better compromise solution. This is, in our opinion, an *open communication* approach.

In the depth first progression, the DM chooses, for each node, the criterion (say $f_p$) he/she wishes to improve in priority. A sub-node is next created by introducing the bound $f_p(x) > \tilde{z}_p$, where $\tilde{z}_p$ is the non-dominated criteria point determined by the weighted Tchebycheff program for that node. The backtracking procedure examines other parts of the tree by generating other sub-nodes (up to more $k-1$). The feasible regions of the children of a given node are disjointed sets (contrariwise to the Marcotte and Soland method), as the following bounds on the objective functions are introduced: 2nd child: $f_{p_2}(x) > \tilde{z}_{p_2} \wedge f_p(x) \leqslant \tilde{z}_p; \ldots;$ $k$th child : $f_{p_k}(x) > \tilde{z}_{p_k} \wedge f_p(x) \leqslant \tilde{z}_p \wedge f_{p_j}(x) \leqslant \tilde{z}_{p_j}$, $j = 2, \ldots, k-1$, where $p$, $p_2, p_3, \ldots, p_k$ is the ordering of the criteria according to the priorities of the DM at that level of the tree.

There are other interactive methods that have been designed to multiobjective integer linear programming but are also applicable to mixed-integer case. Examples of such approaches are those of Vassilev and Narula (1993), Narula and Vassilev (1994) and Karaivanova et al. (1995). In our opinion, they are *open communication* procedures that share some key features, namely the type of information required about the DM's preferences and the concept of projecting a reference direction onto the non-dominated surface (although this concept is employed in different ways). The information of preferences lies fundamentally in the specification of aspiration levels (reference points) and reservation levels for the objective functions. Some of these approaches are continuous/integer (Narula and Vassilev, 1994; Karaivanova et al., 1995) working almost all time with non-dominated continuous solutions (i.e. non-dominated solutions for the linear relaxation of the problem), in order to reduce the computational effort. Whenever the DM finds a satisfactory continuous solution, an integer non-dominated solution close to it is next computed.

The following paragraphs present more details about these methods.

• The interactive algorithm of Vassilev and Narula (1993) can be summarized as follows: (i) compute an initial non-dominated solution, say $\hat{z}$. (ii) If the DM is satisfied with $\hat{z}$, stop; otherwise, ask the DM to specify a new reference point $q$ such as $q_i > \hat{z}_i$ for the objective functions $i$ the DM wishes to improve ($i \in H$), $q_i < \hat{z}_i$ for the objective functions $i$ the DM accepts to deteriorate ($i \in L$) and $q_i = \hat{z}_i$ for the objective functions $i$ the DM would maintain equal ($i \in E$). (iii) Based on $q$ and $\hat{z}$, a scalarizing program is solved and a new (weakly) non-dominated solution is obtained; this solution is assigned to $\hat{z}$, returning to (ii). The scalarizing program maximizes the smallest standardized difference to the last solution ($\hat{z}$) for all objective functions $i \in H$, in order to move as far as possible from $\hat{z}$, but imposing constraints for the other $i \in L \cup E$. The formulation of the scalarizing program is the following, where $\theta$ is a non-negative parameter.

$$
\begin{aligned}
\max \quad & \alpha \\
\text{s.t.} \quad & f_i(x) - (q_i - \hat{z}_i)\alpha \geqslant \hat{z}_i, \quad i \in H, \\
& f_i(x) \geqslant q_i - \theta(\hat{z}_i - q_i), \quad i \in L, \\
& f_i(x) = \hat{z}_i, \quad i \in E, \\
& x \in X, \quad \alpha \geqslant 0.
\end{aligned}
$$

Narula and Vassilev (1994) proposed a modification of this algorithm, which consists of computing continuous solutions in phases (i) and (iii). These are (weakly) non-dominated solutions for the linear relaxation of the multiobjective integer problem. In (iii) one or more solutions are computed for different values of $\theta$. The DM may decide to continue the search of continuous solutions or require the computation of the non-dominated integer solution closer (in the sense *min–max*) to a continuous solution that the DM finds interesting.

• Karaivanova et al. (1995) presented two methods based on the projection of reference points onto the non-dominated set. The first method is pure integer and its underlying principle is close to the method of Vassilev and Narula (1993), but implemented in a different way. The scalarizing program used by Vassilev and Narula maximizes the smallest standardized difference to the last solution for the objective functions the DM wants to improve. Instead, Karaivanova et al. minimizes the largest standardized difference to the reference point for the same objective functions ($i \in H$). For the other objective functions ($i \neq H$), the constraints $f_i(x) \geqslant q_i$ are imposed.

The second method, designated by continuous/integer method, uses the MOLP Pareto Race method of Korhonen and Wallenius (1988) to move around the continuous non-dominated frontier. When the DM finds the most preferred solution for the continuous problem, the integer solution closest to it (in terms of the achievement scalarizing function) is computed.

The authors pointed out the disadvantages of each method. The pure integer one is time consuming but the continuous/integer method operates most of the time in the continuous space, which may be unsatisfactory to the DM. Therefore, the authors proposed a combination of both integrated into a decision support system.

• Alves and Clímaco (2000a) developed an interactive method for multiobjective integer and mixed-integer linear programming. This method follows the same principles as a previous all-integer method presented in Alves and Clímaco (1999). The two methods differ basically in the techniques used to solve the reference point scalarizing programs and the methodology employed for sensitivity analysis. While the first method uses cutting plane techniques, the method of Alves and Clímaco (2000a) uses branch-and-bound techniques. They are *open communication* approaches, which enable a free exploration of the non-dominated solution set. These methods are mainly devoted to perform directional searches by solving scalarizing programs ($T_{\bar{f}}$) parameterized on $\bar{f}$ – *type* (b).

Basically, the method of Alves and Clímaco (2000a) works as follows. At each interaction, the DM can assess directly a new reference point ($\bar{f}$) or just select an objective function, say $f_p$, he wants to improve in relation to the non-dominated solution determined in the previous calculation. In the latter case, the reference point is adjusted automatically by increasing the $p$th component of $\bar{f}$, in order to produce new non-dominated solutions – directional search – more suited to the DM's preferences. This involves an iterative process of sensitivity analysis and operations to update the branch-and-bound tree. The sensitivity analysis procedure returns a value $\bar{\theta}_p \geqslant 0$ such that the structure of the previous branch-and-bound tree remains unchanged for variations in $\bar{f}_p$ up to $\bar{f}_p + \bar{\theta}_p$. Therefore, reference points from $\bar{f}$ to $(\bar{f}_1, \ldots, \bar{f}_p + \bar{\theta}_p, \ldots, \bar{f}_k)$ lead to non-dominated solutions that can be obtained in a straightforward way. To continue the search in the same direction, a slight increase over $\bar{\theta}_p$ is first considered. The previous branch-and-bound tree is used to proceed to the next computations. Thus, the tree structure and some information on the terminal nodes are preserved from one computation to the next, which are used to determine $\bar{\theta}_p$, also providing a starting structure for the next computations. The previous branch-and-bound tree is firstly simplified and then expanded if new branching is required until a new non-dominated solution is reached. This procedure enables to save time in the computational phases of *directional searches*. Besides choosing the objective function to be improved at each moment, the DM has also the possibility of imposing bounds on the objective functions in order to have more control over the directional searches. These constraints may be revised whenever the DM wants, by relaxing or tightening the bounds.

## 5. Comparison of the MOIP/MOMIP interactive methods

Despite the difficulties of comparing interactive methods, some comments and a critical judgement on the research on this area should be made, drawing out some advantages and disadvantages of each method, similarities and differences among them, applicability and computational experience.

Interactive methods aim at overcoming the main difficulties of the generating methods, namely the computational effort and the cognitive burden to the DM. Nevertheless, some interactive approaches still require a significant computational effort. Moreover, there are approaches that put too many questions to the DM, or only tackle biobjective problems or pure integer problems. Some of them are not *complete* in the sense that any efficient solution can be generated. In the following paragraphs we try to particularize some of these points.

The computational effort involved in each iteration is, in general, higher in methods that require the resolution of several integer (or mixed-integer) independent programs at each iteration/interaction. Examples of such methods are those from Steuer and Choo (1983), Marcotte and Soland (1986), White (1985) and Durso (1992). The first one suggests the resolution of $2k$ scalarizing programs ($k$ being the number of objective functions) per iteration. In the other methods, $k+1$ programs are solved for each investigated node of the branch-and-bound tree.

Also, the computational effort is considerable in the extensions of the MOLP Zionts–Wallenius method to problems with integer variables (Ramesh et al., 1990; Villarreal et al., 1980; Karwan et al., 1985; Ramesh et al., 1986). Besides the computational issues, these approaches put many questions to the DM, some of them being difficult to answer, such as the evaluation of *tradeoff* vectors corresponding to relaxed sub-problems.

More recent methods (Vassilev and Narula, 1993; Narula and Vassilev, 1994; Karaivanova et al., 1995; Alves and Clímaco, 2000a) attempt to reduce the computational effort and to not place too many demands on the DM. Vassilev and Narula (1993) started by solving only one scalarizing program at each interaction, but they conclude that even one (mixed) integer program each time may by excessive. Therefore, Narula and Vassilev (1994), followed by Karaivanova et al. (1995) (in their second method), developed continuous–integer approaches, in which most of the time is spent in computing non-dominated solutions for the linear relaxation of the multiobjective (mixed) integer problem. However, since it is not known a priori whether the continuous solutions are near or far from the closest integer solutions, a high waste of time can exist in searching for information that can be uninteresting for the DM. This is, in our opinion, the main disadvantage of this type of approaches. Alves and Clímaco (2000a) reduce the computational effort by profiting from previous computations to solve the following scalarizing programs, employing sensitivity analysis and post-optimality techniques. This is, however, only applicable when consecutive solutions throughout a directional search are computed. If the DM wants to search for dispersed solutions, the scalarizing programs might be solved independently.

Also attempting to reduce the computational burden, some interactive approaches use heuristic techniques to solve the scalarizing programs. Examples of such proposals are the method of Gabbani and Magazine (1986), which is an adaptation of the contraction method of Steuer (1986), and the method of Karaivanova et al. (1993), which is an adaptation of the method of Steuer and Choo (1983). Although this type of approaches can be an interesting way of overcoming the main difficulties of the exact techniques, it is also essential to define adequate performance measures to evaluate the quality of the approximate solutions and the efficiency of the algorithms.

Besides, extensive experimentation must be carried out. The above-mentioned papers do not refer to quality measures, and just Karaivanova et al. (1993) present computational times for the proposed approach. Indeed, heuristic approaches, namely evolutionary algorithms and other meta-heuristics devoted to particular multiobjective integer problems have been a worthwhile stream of research in the last years. Most of these approaches are non-interactive, aiming at generating a representative set of potential efficient solutions. Also for these approaches there has been an increasing concern in creating indices to measure the quality of the solutions.

Natural limitations on the applicability exist for the approaches that only address two objective functions. In our opinion, the methods of Aksoy (1990), Solanki (1991) and Ferreira et al. (1996) are attractive approaches from the computational and cognitive points of view, but their applicability is limited to biobjective problems. In fact, some methodological difficulties can be easier overcome in the biobjective case than in the multiobjective one. For instance, both Aksoy's method and the method of Marcotte and Soland (1986) employ a similar branch-and-bound structure, but the former can be applied to integer and mixed-integer problems and the latter just addresses integer problems.

Some methods cannot capture unsupported efficient solutions. This is the case of Gabbani and Magazine (1986) method, which is based on weighted-sums of the objective functions.

Many approaches, in particular the most recent ones, are based on criteria *reference points*, using Tchebycheff functions or more general *achievement scalarizing functions*. Some advantages of this type of approach can be pointed out. Remember the method of Marcotte and Soland, which uses weighted-sums with additional constraints on the objective functions. This method cannot be applied to mixed-integer problems, but the replacement of weighted-sums by the Tchebycheff metric, proposed by Durso (1992), has overcome this weakness. Also, the Tchebycheff metric introduced in the second approach proposed by Ferreira (1997) has improved the previous method of the author because this change enables the elimination of larger regions by unfeasibility. Therefore, it is noteworthy to observe that there has been an increasing focus on reference point approaches to deal with multiobjective integer and mixed-integer problems.

Finally, a brief comment must be made to the computational experience of the above-mentioned methods. Few papers related to computational experience, the most recent ones being Karaivanova et al. (1993, 1995) and Alves and Clímaco (1999, 2000a). Although the fast evolution of computers enhances undoubtedly the results, it should be stressed that more efforts are still necessary to turn the methods applicable to large problems.

## 6. Concluding remarks

In this paper we made a revision of interactive methods devoted to the general case of multiobjective integer and mixed-integer programming. It was paid particular attention to interactive approaches because, in our opinion, they generally are more adequate to face the difficulties of complex decision problems. Nonetheless, generating methods can also be promising if they are appropriately used, and further investigation on this field can provide important findings to the development of more effective interactive methods.

On the other extreme of the spectrum, some approaches making an a priori articulation of the decision markers preferences, the utility/value function approaches, were proposed in scientific literature. They are the most popular in classical operations research, in which the preferences of the decision makers are modeled a priori, assuming that the model is, in Platonic sense, a faithful description of reality. It is supposed that, in the construction of the model, the analyst has full information and is *rational.* This type of approach is the most attractive in terms of computational effort. However, the descriptive models of optimization are per se, in our opinion, insufficient to prescribe decisions in most of the practical problems.

We believe that interactive approaches rooted in constructivism (*open communication* protocols) represent one of the most promising ways of research to develop adequate MOIP/MOMIP tools for decision aiding in many complex practical situations.

Other type of interactive tools has been reported, which assumes the existence of an *implicit utility function* of the decision maker. The interactive protocols are built in order to *discover* the optimal solution of that function. These methods constitute an evolution of those based on the *paradigm of optimality* referred to above, providing more flexibility to the decision makers.

It must be recognized that he research effort in the development of MOIP/MOMIP methods has been limited. However, we believe that this is a very promising area because the interest of including explicitly multiple objectives in different real world application areas of integer programming models is undoubted.

We will exemplify with a short reference to one of these areas, the telecommunication planning, design and management. Recently, an increasing number of scientific papers has appeared in the literature (see Clímaco and Craveirinha, 2004). We emphasize those related to QoS (quality of service) routing problems. In fact, the new paradigm of routing taking into account different metrics related to the quality of service is very important in many situations, particularly in Internet problems. Some routing problems are among those situations where the decisions must be taken automatically, in *real time*, and revised periodically taking into account the state of certain parameters of the networks also monitored in real time. This opens new fronts of research, namely developing rule based approaches to aggregate the preferences and the use of heuristics to get solutions in short time. It must also be remarked that approaches devoted to special cases of MOIP/MOMIP problems may be very useful in many practical applications. For example, in routing problems, the multiobjective shortest-path algorithms are especially useful because they are much more efficient in computational terms than the general-purpose algorithms. As this paper just addresses to the general case, it is suggested the reading of other sources, for instance Ehrgott and Gandibleux (2000), for those interested in special case approaches.

It must be also stressed that uncertainty is a key issue in fixing the parameters in MOIP/MOMIP models as well as to fix the parameters required by the methods to aggregate the preferences of the decision makers. In most situations the model cannot integrate all the interesting issues of the practical problem regarding the interests of the decision maker. Therefore, it is a challenge to work on the following directions: first, the implementation of procedures enabling the evaluation of the robustness of the results when some parameters are known in an imprecise manner, besides the improvement of sensitivity analysis procedures; secondly, a posteriori detailed analysis of some satisfactory solutions, taking into account issues not included in the model, but relevant to choose among more or less *indifferent* satisfactory solutions previously selected by the decision maker using an interactive approach.

Finally, a brief reference should be made to meta-heuristic approaches. In the last years, many research efforts have been assigned to multiobjective meta-heuristic approaches, namely multiobjective evolutionary algorithms. A vast list of papers can be found on evolutionary multiobjective optimization, almost published in the last decade (http://www.lania.mx/~ccoello/EMOO/EMOObib.html). Most of them apply to particular combinatorial problems and are non-interactive, aiming at generating a good approximation of the whole non-dominated solution set. Nevertheless, examples of interactive approaches to multiobjective 0–1 or integer problems can be given: Kato and Sakawa (1998) and Alves and Clímaco (2000b) for multiobjective 0–1 programming, and Sakawa et al. (1994) for multiobjective integer programming. As this type of approaches is out of the scope of this review, they have not been comprised in this paper, even though this is a remarkable research area.

## References


Aksoy, Y., 1990. An interactive branch-and-bound algorithm for bicriterion nonconvex/mixed integer programming. Naval Research Logistics 37, 403–417.

Alves, M.J., Clímaco, J., 1999. Using cutting planes in an interactive reference point approach for multiobjective integer linear programming problems. European Journal of Operational Research 117 (3), 565–577.

Alves, M.J., Clímaco, J., 2000a. An interactive reference point approach for multiobjective mixed-integer programming using branch-and-bound. European Journal of Operational Research 124 (3), 478–494.

Alves, M.J., Clímaco, J., 2000b. An interactive method for 0–1 multiobjective problems using simulated annealing and tabu search. Journal of Heuristics 6 (3), 385–403.

Alves, M.J., Clímaco, J., 2001. Multiobjective mixed-integer programming. In: Pardalos, P.M., Floudas, C.A. (Eds.), Encyclopedia of Optimization, vol. III. Kluwer Academic Publishers, pp. 466–472.

Benayoun, R., Montgolfier, J., Tergny, J., Larichev, O., 1971. Linear programming with multiple objective functions: Step method (STEM). Mathematical Programming 1, 366–375.

Bitran, G.R., 1977. Linear multiple objective programs with zero–one variables. Mathematical Programming 13, 121–139.

Bitran, G.R., 1979. Theory and algorithms for linear multiple objective programs with zero–one variables. Mathematical Programming 17 (3), 362–389.

Bitran, G.R., Riviera, J., 1982. A combined approach to solve binary multicriteria problems. Naval Research Logistics Quarterly 29 (2), 181–201.

Bowman Jr, V.J., 1976. On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In: Thiriez, H., Zionts, S. (Eds.), Multiple Criteria Decision Making, Lecture Notes in Economics and Mathematical Systems, vol. 130. Springer-Verlag, Berlin, pp. 76–86.

Chalmet, L.G., Lemonidis, L., Elzinga, D.J., 1986. An algorithm for the bi-criterion integer programming problem. European Journal of Operational Research 25, 292–300.

Clímaco, J., Craveirinha, J., 2004. Multicriteria analysis in telecommunication network planning and design – problems and issues. In: Figueira, J., Greco, S., Ehrgott, M. (Eds.), Multiple Criteria Decision Analysis: The State of the Art Surveys. Kluwer Academic Publishers, pp. 899–952 (Chapter 22).

Clímaco, J., Ferreira, C., Captivo, M.E., 1997. Multicriteria integer programming: An overview of the different algorithmic approaches. In: Clímaco, J. (Ed.), Multicriteria Analysis. Springer-Verlag, Berlin, pp. 248–258.

Deckro, R.F., Winkofsky, E.P., 1983. Solving zero–one multiple objective programs through implicit enumeration. European Journal of Operational Research 12, 362–374.

Durso, A., 1992. An interactive combined branch-and-bound/Tchebycheff algorithm for multiple criteria optimization. In: Goicoechea, A., Duckstein, L., Zionts, S. (Eds.), Multiple Criteria Decision Making: Theory and Applications in Business, Industry and Government. Springer-Verlag, New York, pp. 107–122.

Ehrgott, M., Gandibleux, X., 2000. A survey and annotated bibliography of multiobjective combinatorial optimization. OR Spektrum 22 (4), 425–460.

Evans, G.W., 1984. An overview of techniques for solving multiobjective mathematical programs. Management Science 30 (11), 1268–1282.

Ferreira, C. Problemas de localização e distribuição multicritério – aproximações e estudo de alguns casos com implicações ambientais. PhD Thesis (In Portuguese). University of Aveiro, Portugal, 1997.

Ferreira, C., Santos, B.S., Captivo, M.E., Clímaco, J., Silva, C.C., 1996. Multiobjective location of unwelcome or central facilities involving environmental aspects: A prototype of a decision support system. Belgian Journal of Operations Research, Statistics and Computer Science 36 (2–3), 159–172.

Feyerabend, P., 1975. Against the Method. New Left Books, London.

Gabbani, D., Magazine, M., 1986. An interactive heuristic approach for multi-objective integer-programming problems. Journal of the Operational Research Society 37 (3), 285–291.

Gonzalez, J.J., Reeves, G.R., Franz, L.S., 1985. An interactive procedure for solving multiple objective integer linear programming problems. In: Haimes, Y.Y., Chankong, V. (Eds.), Decision Making with Multiple Objectives, Lecture Notes in Economics and Mathematical Systems, vol. 242. Springer-Verlag, Berlin, pp. 250–260.

Karaivanova, J.N., Narula, S.C., Vassilev, V., 1993. An interactive procedure for multiple objective integer linear programming problems. European Journal of Operational Research 68, 344–351.

Karaivanova, J., Korhonen, P., Narula, S., Wallenius, J., Vassilev, V., 1995. A reference direction approach to multiple objective integer linear programming. European Journal of Operational Research 81, 176–187.

Karwan, M.H., Zionts, S., Villarreal, B., Ramesh, R., 1985. An improved interactive multicriteria integer programming algorithm. In: Haimes, Y.Y., Chankong, V. (Eds.), Decision Making with Multiple Objectives, Lecture Notes in Economics and Mathematical Systems, vol. 242. Springer-Verlag, Berlin, pp. 261–271.

Kato, K., Sakawa, M., 1998. An interactive fuzzy satisficing method for large scale multiobjective 0–1 programming problems with fuzzy parameters through genetic algorithms. European Journal of Operational Research 107 (3), 590–598.

Kiziltan, G., Yucaoglu, E., 1983. An algorithm for multiobjective zero–one linear programming. Management Science 29 (12), 1444–1453.

Klein, D., Hannan, E., 1982. An algorithm for the multiple objective integer linear programming problem. European Journal of Operational Research 9, 378–385.

Korhonen, P., Wallenius, J., 1988. A Pareto Race. Naval Research Logistics 35, 615–623.

Lewandowski, A., Wierzbicki, A. Aspiration based decision analysis and support. Part I: Theoretical and methodological backgrounds. Working paper WP-88-03, IIASA, Laxenburg, Austria, 1988.

L'Hoir, H., Teghem, J. Portfolio selection by MOLP using an interactive branch and bound. Foundations of Computing and Decision Sciences 1995; 20 (3); Institute of Computing Science, Technical University of Poznan, Poland.

Marcotte, O., Soland, R.M., 1980. Branch and bound algorithm for multiple criteria optimization. University of Cornell and George Washington.

Marcotte, O., Soland, R.M., 1986. An interactive branch-and-bound algorithm for multiple criteria optimization. Management Science 32 (1), 61–75.

Mavrotas, G., Diakoulaki, D., 1998. A branch and bound algorithm for mixed zero–one multiple objective linear programming. European Journal of Operational Research 107, 530–541.

Narula, S.C., Vassilev, V., 1994. An interactive algorithm for solving multiple objective integer linear programming problems. European Journal of Operational Research 79, 443–450.

Pasternak, H., Passy, U., 1973. Bicriterion mathematical programs with Boolean variables. In: Cochrane, J.L., Zeleny, M. (Eds.), Multiple Criteria Decision Making. University of South Carolina Press, pp. 327–348.

Ramesh, R., Karwan, M.H., Zionts, S., 1990. An interactive method for bicriteria integer programming. IEEE Transactions on Systems, Man, and Cybernetics 20 (2), 395–403.

Ramesh, R., Zionts, S., Karwan, M.H., 1986. A class of practical interactive branch and bound algorithms for multicriteria integer programming. European Journal of Operational Research 26, 161–172.

Rasmussen, L.M., 1986. Zero–one programming with multiple criteria. European Journal of Operational Research 26, 83–95.

Roy, B., 1987. Meaning and validity of interactive procedures as tools for decision making. European Journal of Operational Research 31, 297–303.

Sakawa, M., Shibano, T., Kato, K., 1994. An interactive fuzzy satisficing method for multiobjective integer programming problems through genetic algorithms. In: Jain, L.C., Jain, R.K. (Eds.), Second International Conference on Knowledge-Based Intelligent Electronic Systems. IEEE, Adelaide, Australia, pp. 94–100.

Shin, W.S., Allen, D.B., 1994. An interactive paired comparison method for bicriterion integer programming. Naval Research Logistics 41, 423–434.

Soland, R.M., 1979. Multicriteria optimization: A general characterization of efficient solutions. Decision Sciences 10, 26–38.

Solanki, R., 1991. Generating the noninferior set in mixed integer biobjective linear programs: An application to a location problem. Computers and Operations Research 18 (1), 1–15.

Steuer, R.E., 1977. An interactive multiple objective linear programming procedure. TIMS Studies in the Mangement Sciences 6, 225–239.

Steuer, R.E., 1986. Multiple Criteria Optimization: Theory, Computation and Application. Wiley, New York.

Steuer, R.E., Choo, E.-U., 1983. An interactive weighted Tchebycheff procedure for multiple objective programming. Mathematical Programming 26, 326–344.

Steuer, R.E., Silverman, J., Whisman, A.W., 1993. A combined Tchebycheff/aspiration criterion vector interactive multiobjective programming procedure. Management Science 39 (10), 1255–1260.

Sylva, J., Crema, A., 2004. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. European Journal of Operational Research 158, 46–55.

Teghem, J., Kunsch, P.L., 1986a. A survey of techniques for finding efficient solutions to multi-objective integer linear programming. Asia-Pacific Journal of Operational Research 3, 95–108.

Teghem, J., Kunsch, P.L., 1986b. Interactive methods for multi-objective integer linear programming. In: Fandel, G., Grauer, A., Kurzhanski, A., Wierzbicki, A.P. (Eds.), Large-Scale Modelling and Interactive Decision Analysis, Lecture Notes in Economics and Mathematical Systems, vol. 273. Springer-Verlag, Berlin, pp. 75–87.

Vassilev, V., Narula, S.C., 1993. A reference direction algorithm for solving multiple objective integer linear programming problems. Journal of the Operational Research Society 44 (12), 1201–1209.

Villareal, B., Karwan, M.H., 1981. Multicriteria integer programming: A (hybrid) dynamic programming recursive approach. Mathematical Programming 21, 204–223.

Villarreal, B., Karwan, M.H., Zionts, S., 1980. An interactive branch and bound procedure for multicriterion integer linear programming. In: Fandel, G., Gal, T. (Eds.), Multiple Criteria Decision Making Theory and Application, Lecture

Notes in Economics and Mathematical Systems, vol. 177. Springer-Verlag, Berlin, pp. 448–467.

White, D.J., 1984. A branch and bound method for multi-objective boolean problems. European Journal of Operational Research 15, 126–130.

White, D.J., 1985. A multiple objective interactive Lagrangean relaxation approach. European Journal of Operational Research 19, 82–90.

Wierzbicki, A.P., 1982. A mathematical basis for satisficing decision making. Mathematical Modelling 3, 391–405.

Wierzbicki, A.P., 1986. On the completeness and constructiveness of parametric characterizations to vector optimization problems. OR Spectrum 8, 73–87.

Zionts, S., Wallenius, J., 1983. An interactive multiple objective linear programming method for a class of underlying nonlinear utility functions. Management Science 29 (5), 519–529.