A Preference-based Evolutionary Algorithm for Multiobjective Optimization

L. Thiele thiele@tik.ee.ethz.ch
ETH-Zurich, Department of Information Technology and Electrical Engineering, Gloriastrasse 35, CH-8092 Zürich, Switzerland

K. Miettinen kaisa.miettinen@jyu.fi
Department of Mathematical Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyväskylä, Finland

P. J. Korhonen pekka.korhonen@hse.fi
Helsinki School of Economics, Department of Business Technology, P.O. Box 1210, FI-00101 Helsinki, Finland

J. Molina julian.molina@uma.es
Department of Applied Economics, University of Malaga, E-29071 Malaga, Spain

Abstract

In this paper, we discuss the idea of incorporating preference information into evolutionary multiobjective optimization and propose a preference-based evolutionary approach that can be used as an integral part of an interactive algorithm. One algorithm is proposed in the paper. At each iteration, the decision maker is asked to give preference information in terms of her/his reference point consisting of desirable aspiration levels for objective functions. The information is used in an evolutionary algorithm to generate a new population by combining the fitness function and an achievement scalarizing function. In multiobjective optimization, achievement scalarizing functions are widely used to project a given reference point into the Pareto optimal set. In our approach, the next population is thus more concentrated in the area where more preferred alternatives are assumed to lie and the whole Pareto optimal set does not have to be generated with equal accuracy. The approach is demonstrated by numerical examples.

Keywords

Multiple objectives, multiple criteria decision making, preference information, reference point, achievement scalarizing function, Pareto optimality, fitness evaluation.

1 Introduction

Most real-life decision and planning situations involve multiple conflicting criteria that should be considered simultaneously. The term *multiple criteria decision making* (MCDM) or *multiobjective optimization* refers to solving such problems. For them, it is characteristic that no unique solution exists but a set of mathematically equally good solutions can be identified. These solutions are known as efficient, nondominated, noninferior or Pareto optimal solutions. In the MCDM literature, the terms are often seen as synonyms.

In the MCDM literature, the idea of solving a multiobjective optimization problem is understood as helping a human *decision maker* (DM) to consider the multiple criteria simultaneously and to find a Pareto optimal solution that pleases him/her most.

Thus, the solution process always requires the involvement of the DM and the final solution is determined by her/his preferences. Usually, decision support systems operate iteratively generating Pareto optimal solutions based on some rules and the DM makes choices and specifies preference information. Those choices are used to lead the algorithm to generate more Pareto optimal solutions until the DM reaches the most satisfactory, that is, the final solution. In other words, not all Pareto optimal solutions are generated but only the ones the DM finds interesting.

On the other hand, evolutionary multiobjective optimization (EMO) methods take a different approach to solving multiobjective optimization problems. It is also important to note that when compared to the MCDM literature, there is a difference in terminology. EMO approaches generate a set of nondominated solutions which is a representation approximating the (unknown) Pareto optimal set. Thus in EMO, Pareto optimality and nondominance are not synonyms. The intervention of the DM is not needed in the process. So far, rather little interest has been paid in the literature to choosing one of the nondominated solutions as the final one. However, there is typically a need to identify such a solution indicating which values the decision variables should have in order to get the best possible values for the conflicting criteria. The difficulty of identifying the best nondominated solutions is even more evident when there are more than two criteria and it, thus, is difficult to display the set of nondominated solutions.

One can say that MCDM and EMO approaches are based on different philosophies even though they are applied to similar problems. In this paper, we combine elements of solution techniques used in MCDM and EMO communities and suggest a way to hybridize them. Because it seems that publications in the literature have mostly concentrated on either MCDM or EMO approaches, we also wish to describe some MCDM developments to those more familiar with EMO approaches.

Helping DMs in solving multiobjective optimization problems has been the subject of intensive studies since the beginning of the 1970's (see, e.g., Benayoun et al. 1971, Geoffrion et al. 1972 and Zionts and Wallenius 1976). However, many theoretical concepts were defined much earlier (see, e.g., Koopmans 1971, Kuhn and Tucker 1951 and Pareto 1906) and, actually, many ideas originated from the theory of mathematical programming.

Surveys of methods developed for multiobjective optimization problems include Chankong and Haimes (1983), Hwang and Masud (1979), Miettinen (1999), Sawaragi et al. (1985) and Steuer (1986). For example, in Hwang and Masud (1979) and Miettinen (1999), multiobjective optimization methods are classified into four classes according to the role of the DM in the solution process. Sometimes, there is no DM available and in this case some neutral compromise solution is to be identified. Such *no-preference methods* must be used if no preference information is available. In *a priori methods*, the DM articulates preference information and one's hopes before the solution process. The difficulty here is that the DM does not necessarily know the limitations and possibilities of the problem and may have too optimistic or pessimistic hopes. Alternatively, a set of Pareto optimal solutions can be generated first and then the DM is supposed to select the most preferred one among them. Typically, evolutionary multiobjective optimization algorithms belong to this class of *a posteriori methods*. If there are more than two criteria in the problem, it may be difficult for the DM to analyze the large amount of information and, on the other hand, generating the set of Pareto optimal or nondominated alternatives may be computationally expensive.

The drawbacks of both a priori and a posteriori methods can be overcome if there is a DM available who is willing to participate in the solution process and direct it according to her/his preferences. So-called *interactive methods* form a solution pattern which is iteratively repeated as long as the DM wants. After each iteration, the DM is provided with one or some Pareto optimal solutions that obey the preferences expressed as well as possible and (s)he can specify more preference information. This can be, for example, in the form of trade-offs, pairwise comparisons, aspiration levels, classification of objective functions, etc. The responses are used to generate presumably improved solutions. In this way, the DM can learn about the problem and fine-tune one's preferences if needed. The ultimate goal is to find the solution that satisfies her/him most. Interactive methods are computationally inexpensive because only such Pareto optimal solutions are generated that are interesting to the DM. In order to support the DM better, the ideas of interactive and a posteriori methods can also be hybridized to utilize advantages of both the approaches see, for example, Klamroth and Miettinen (2008).

Besides using different types of preference information, interactive methods also differ from each other in the way the information is utilized in generating new, improved solutions and what is assumed about the behaviour of the DM. Typically, different methods convert the original multiple objectives as well as the preference information into an optimization problem with a single objective function using a so-called scalarizing function. The resulting problem is then solved with some appropriate single objective solver. When dealing with real-life problems, there may be integer-valued variables or nonconvex or nondifferentiable functions involved, which set requirements on the solvers used.

As discussed above, including DM's preferences is important when dealing with multiobjective optimization problems, as the aim is to help the DM to find the most preferred solutions without exploring the whole set of Pareto optimal solutions and lead her/him to a better knowledge of the problem being solved. However, the number of EMO methods including DM's preferences is relatively small in contrast to the number of interactive approaches found in the MCDM literature. Only some works can be found combining EMO algorithms and interactive methods, although many authors have been asking for such approaches, including Hanne (2006).

Coello (2000) has presented a survey on including preferences when using a multiobjective evolutionary method: Fonseca and Fleming (1993) probably suggested the earliest attempt to incorporate preferences, and the proposal was to use MOGA together with goal information as an additional criterion to assign ranks to the members of a population. Greenwood et al. (1997) used value functions to perform the ranking of attributes, and also incorporated preference information into the survival criteria. Cvetkovic and Parmee (1999, 2002) and Parmee et al. (2000) used binary preference relations (translated into weights) to narrow the search. These weights were used in some different ways to modify the concept of dominance. Rekiek et al. (2000) used the PROMETHEE method (a multiattribute decision analysis method for discrete alternatives) to generate weights for an EMO method. On the other hand, Massebeuf et al. (1999) used PROMETHEE II in an a posteriori form: an EMO algorithm generated nondominated solutions and PROMETHEE II selected some of them based on the DM's preferences. Deb (1999a) used variations of compromise programming to bias the search of an EMO approach. Finally, in Deb (1999b) the DM was required to provide goals for each objective.

More recently, some other approaches have been published. Phelps and Köksalan (2003) used pairwise comparisons to include DM's preferences in the fitness function. In the guided multi-objective evolutionary algorithm (G-MOEA) proposed by Branke et al. (2001) user preferences were taken into account using trade-offs, supplied by the DM, to modify the definition of dominance. In Branke and Deb (2004), two schemes were proposed to include preference information when using an EMO (they used the NSGA-II for testing): modifying the definition of dominance (using the guided dominance principle of G-MOEA) and using a biased crowding distance based on weights.

Sakawa and Kato (2002) used reference points in a traditional approach. One reference point was used to compute a single tentative efficient solution, and at each iteration, the DM was asked to specify a new reference point until satisfaction was reached. Instead of using classical (crisp) reference points, they use a fuzzy approach to represent the DM's preferences. The main contribution of this paper is (besides the coding-decoding method to represent solutions) the way preferences are represented in a fuzzy way.

Finally, in Deb et al. (2005), preferences were included through the use of reference points. Also a guided dominance scheme and a biased crowding scheme were suggested. Deb et al. (2005) are using the concept of the reference point but do not apply an achievement scalarizing function, which is an essential concept in the context of reference points, as suggested by Wierzbicki (1980). An achievement scalarizing function is a "tool" which guarantees that the point found by means of the scalarizing function is Pareto optimal. In addition, an achievement scalarizing function enables finding a solution which satisfies the preferences the DM has specified in the reference point. The main difference of our approach when compared to the one proposed by Deb et al. (2005), as will be shown later, is that we directly use reference point information in fitness evaluation (through an achievement scalarizing function that will also be defined later) in an indicator-based evolutionary algorithm IBEA (see, Zitzler and Kuenzli, 2004).

In this paper, we suggest a hybrid approach where we combine ideas from both evolutionary and interactive multiobjective optimization (as encouraged e.g. in Branke et al. (2008)). The principle is to incorporate preference information coming from a DM in the evolutionary approach. As justified earlier in this section, we are not interested in approximating the whole Pareto optimal set. Instead, we first give a rough approximation, and then generate a more accurate approximation of the area where the DM's most satisfactory solution lies. In practice, the DM is asked to give preference information in terms of her/his reference point consisting of desirable aspiration levels for objective functions. This information is used in a preference-based evolutionary algorithm that generates a new population by combining the fitness function and a so-called achievement scalarizing function containing the reference point. The next population is more concentrated in the area where more preferred alternatives are assumed to lie. With the new evolutionary approach, the DM can direct the search towards the most satisfactory solution but still learn about the behaviour of the problem, which enables her/him to adjust one's preferences. It is easier to specify the reference point after the DM has seen a rough approximation of Pareto optimal solutions available but the approximation only has to be improved in quality in the interesting parts of the Pareto optimal set. Because evolutionary algorithms set no assumptions on the differentiability, convexity or continuity of the functions involved, the approach can be used in solving complicated real-life problems.

In summary, the suggested approach has two main new elements: first an achievement scalarizing function is adapted for an EMO algorithm and, in particular, fitness evaluation in it, and secondly an interactive solution method is proposed which is based on utilizing this achievement scalarization function in evolutionary optimization. In this way, we modify the traditional aim of EMO algorithms (in generating an approximation of the whole Pareto optimal set) and incorporate ideas (used in MCDM methods) of decision support. Multiobjective optimization is more than biobjective optimization. Thus, the motivation is to create new methods that can conveniently be used when the problem to be solved has more than two criteria and when applying plain EMO ideas is not efficient.

The rest of this paper is organized as follows. In Section 2, we introduce the basic concepts and notations of multiobjective optimization. Section 3 is devoted to discussion on different ways of handling preference information. We pay special attention to reference point based methods and achievement scalarizing functions. We introduce our preference based interactive algorithm in Section 4 and demonstrate how it works with a few examples in Section 5. Finally, we draw some conclusions in Section 6.

2 Multiobjective optimization

A multiobjective optimization problem can be written in the form

$$\begin{array}{ll} \text{minimize} & f(x) = (f_1(x), \ldots, f_k(x)) \\ \text{subject to} & x \in X, \end{array} \tag{1}$$

where $X \subset \Re^n$ is a *feasible set* of decision variables and $f : \Re^n \rightarrow \Re^k$. The n-dimensional space $\Re^n$ is called a *variable space* and the functions f_i, $i = 1, \ldots, k$, are *objective functions* or *criteria*. The k-dimensional space $\Re^k$ is a so-called *criterion space* and its subset, the image of the feasible set, called a *feasible criterion region*, can now be written as $Q = \{q | q = f(x), x \in X\}$. The set Q is of special interest and most considerations in multiobjective optimization are made in the criterion space.

Problem (1) has several mathematically equivalent solutions. They are called efficient, nondominated, noninferior or Pareto optimal (sometimes in the MCDM literature some of these concepts are associated with decision and the others with criterion spaces). Any choice from among the set of Pareto optimal solutions is impossible, unless we have additional information available about the DM's preference structure. To be more specific, we have the following definitions:

Definition 1 *In (1), a vector $f(x)$, $x \in X$, is said to* dominate *another vector $f(y)$, $y \in X$, if $f_i(x) \leq f_i(y)$ for all $i = 1, \ldots, k$, and the inequality is strict for at least one i.*

Definition 2 *In (1), a vector $f(x^*)$, $x^* \in X$, is* nondominated *if there does not exist another $x \in X$ such that $f(x)$ dominates $f(x^*)$.*

The set of all nondominated solutions is called the nondominated frontier or Pareto optimal set. The final ("best") solution of problem (1) is called the *most preferred solution*. It is a nondominated solution preferred by the DM to all other solutions. We also will use the concept of weakly nondominated solutions which is defined by replacing all inequalities by strict inequalities in Definition 1. The set of nondominated solutions is a subset of weakly nondominated solutions.

3 On preference information in different MCDM methods

Several dozens of methods have been developed during the last over 30 years to address multiobjective optimization problems see, for example, the textbooks by Chankong and Haimes (1983), Hwang and Masud (1979), Miettinen (1999), Sawaragi et al. (1985) and Steuer (1986). Typically, they always require the intervention of a DM at some stage in the solution process. A popular way to involve the DM in the solution process is to use interactive approaches as discussed in the introduction. Because the goal is to support the DM, we can refer to the tools used as decision support systems. The ultimate goal is to find the most preferred solution of the DM.

There is no single criterion for evaluating multiple criteria decision support systems. Instead, several relevant criteria can be introduced, for example:

- the system recognizes and generates Pareto optimal solutions and helps the DM to get a "holistic" view of the Pareto optimal set;
- the system helps the DM feel convinced that the final solution is the most preferred one, or at least close enough to that;
- the system provides an intuitive user interface and does not require too much time from the DM to find the final solution.

Provided that the problem is correctly specified, the final solution of a rational DM is always Pareto optimal. Therefore, it is important that the system is able to recognize and generate Pareto optimal solutions. No system can provide a DM with a capability to compare all alternatives simultaneously. However, a good system can provide a holistic view over the alternatives and assist the DM in becoming convinced that her/his final choice is the best or at least close to the best solution. The user interface plays an important role in that aspect.

3.1 Overview of some interactive methods

An example of early interactive methods is the GDF method, see Geoffrion et al. (1972). It assumes that there exists an unknown value function that represents the preferences of the DM and (s)he wants to maximize this function. Even though the function is not explicitly known, information about it is asked from the DM in the form of responses to specific questions involving marginal rates of substitution of pairs of objective functions and, in this way, the DM guides the solution process towards the most preferred solution. This approach assumes consistency on the DM's part as well as some differentiability assumptions.

Alternatively, a small sample of Pareto optimal solutions can be generated and the DM is supposed to select the most preferred one of them. Then, the next sample of Pareto optimal solutions is generated so that it concentrates on the neighbourhood of the selected one, see Steuer (1986).

It has been shown in Larichev (1992) that for a DM, classification of objective functions is a cognitively valid way of expressing preference information. Classification means that the objective function values at the current Pareto optimal solution are shown to the DM and the DM is asked to indicate how the solution should be improved by classifying the functions according to whether their current values are acceptable, should be improved or could be impaired (in order to allow improvements in some others). In addition, desirable amounts of improvements or allowed amounts of impairments may be asked from the DM. Classification-based interactive multiobjective optimization methods include, for example, the Step method (Benayoun et al. 1971), the satisficing trade-off method (STOM, Nakayama and Sawaragi 1984) and the NIMBUS method (Miettinen and Mäkelä 1995, 2006). The methods differ from each other, for example, in the number of classes available, the information asked the DM and how this information is used to generate a new solution.

Closely related to classification is the idea of expressing preference information using reference points. The difference is that while classification assumes that some objective functions must be allowed to get worse values, a reference point can be selected more freely. Reference points consist of aspiration levels reflecting desirable values for the objective functions. This is a natural way of expressing preference information and in this straight-forward way the DM can express hopes about improved solutions and directly see and compare how well they could be attained when the next solution is generated. The reference point is projected to the Pareto optimal set by minimizing a so-called achievement scalarizing function (Wierzbicki 1980, 1986). Here, no specific behavioural assumptions like, for example, transitivity are necessary. Reference points play the main role also, for example, in the 'light beam search' (Jaszkiewicz and Slowinski 1999), visual reference direction approach (Korhonen and Laakso 1986) and its dynamic version Pareto Race (Korhonen and Wallenius 1988). Because of their intuitive nature, in what follows, we concentrate on reference point based approaches and introduce achievement scalarizing functions used with them.

3.2 Achievement scalarizing functions

Many MCDM methods are based on the use of *achievement scalarizing functions* first proposed by Wierzbicki (1980). The achievement (scalarizing) function projects any given (feasible or infeasible) point $\boldsymbol{g} \in \Re^k$ into the set of Pareto optimal solutions. The point $\boldsymbol{g}$ is called a *reference point*, and its components represent the desired values of the objective functions. These values specified by the DM are called *aspiration levels*.

The simplest form of an achievement function to be minimized subject to the original constraints $x \in X$ is:

$$s_g(f(x)) = \max_{i=1,\ldots,k} [w_i(f_i(x) - g_i)], \tag{2}$$

where $w_i > 0$ for all $i = 1, \ldots, k$ are fixed scaling factors and $g \in \Re^k$ is the reference point specified by the DM. We can, e.g., set $w_i = 1/\mathit{range}_i$, where range_i is the subjective (Korhonen and Wallenius 1988) or computed range of the objective function values in the Pareto optimal set, see for further information e.g. Miettinen (1999). Alternatively, it is possible to utilize the ranges of the objective function values in the population approximating the Pareto optimal set. It can be shown that the minimal solution of the achievement function is weakly Pareto optimal (see, e.g., Wierzbicki 1986) independently of how the reference point is chosen. Furthermore, if the solution is unique, it is Pareto optimal. If the reference point $g \in \Re^k$ is feasible for the original multiobjective optimization problem, that is, it belongs to the feasible criterion region, then for the solution of (2), $f(x^*) \in Q$, we have $f_i(x^*) \leq g_i$ for all $i = 1, \ldots, k$. Otherwise, by solving (2) we get a (weakly) Pareto optimal solution that is closest to the Pareto optimal set. To guarantee that only Pareto optimal (instead of weakly Pareto optimal) solutions are generated, a so-called augmented form of the achievement function can be used:

$$s_g(f(x)) = \max_{i=1,\ldots,k} [w_i(f_i(x) - g_i)] + \rho \sum_{i=1}^{k} (f_i(x) - g_i), \tag{3}$$

where $\rho > 0$ is a small *augmentation coefficient* (providing a bound for trade-offs regarded interesting). Let us emphasize that a benefit of achievement functions is that any Pareto optimal solution can be found by altering the reference point only (see, e.g., Wierzbicki 1986).

To illustrate the use of the achievement function, let us consider a problem with two criteria to be minimized as shown in Figure 1. In the figure, the thick solid lines represent the set of Pareto optimal solutions in the criterion space. The points A and B in the criterion space are two different reference points and the resulting Pareto optimal solutions are A′ and B′, respectively. The cones stand for indifference curves when $\rho = 0$ in the achievement scalarizing function and the last point where the cone intersects the feasible criterion region is the solution obtained, that is, the projection of the reference point. As Figure 1 illustrates, different Pareto optimal solutions can be generated by varying the reference point and the method works well for both feasible (e.g., A) and infeasible (e.g., B) reference points. More information about achievement functions is also given, for example, in Miettinen (1999). If so desired, it is possible to adjust the projection direction by modifying the scaling factors w_i using some preference information, see Luque et al. (2009) for details.

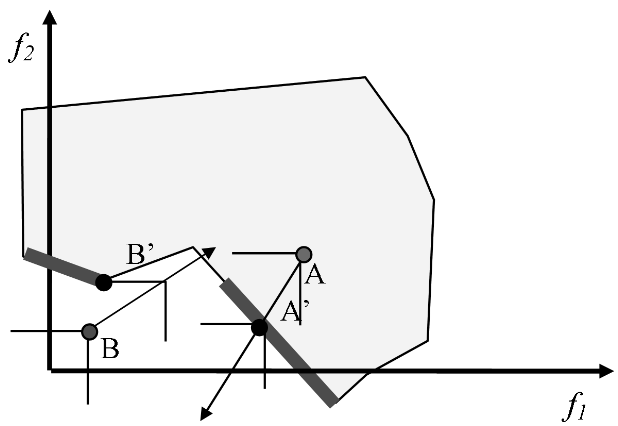

Figure 1: Illustrating the projection of a feasible and infeasible reference point into the Pareto optimal set.

In summary, reference points and achievement functions can be used so that after the DM has specified her/his hopes for a desirable solution as a reference point, (s)he sees the Pareto optimal solution minimizing the corresponding achievement function. In this way, the DM can compare one's hopes (i.e., the reference point) and what was feasible (i.e., the solution obtained) and possibly set a new reference point.

4 An approach to incorporate preference information in EMO

A common point in many EMO methods in the literature is the absence of preference information in the solution process. As mentioned earlier, EMO methods try to generate the whole nondominated frontier (approximating the real Pareto optimal set) assuming that any nondominated solution is desirable. But this is not always the case in a real situation where different areas of the nondominated frontier could be more preferred than some others, and some areas could not be interesting at all. From our point of view, this lack of preference information produces shortcomings in two ways:

- computational effort is wasted in finding undesired solutions; and
- a huge number of solutions is presented to the DM who may be unable to find the most preferred one among them whenever the problem has more than two criteria and, a visual representation of the set of nondominated solutions is not as illustrative or intuitive as with two criteria.

In order to avoid the above-mentioned shortcomings, preference information must be used in the solution process. In this way, we avoid visiting undesired areas and the DM guides the search towards her/his most preferred solution.

In the approach to be introduced, we incorporate preference information given by the DM in the form of a reference point in the evolutionary multiobjective algorithm so that generations gradually concentrate in the neighborhood of those solutions that obey the preferences as well as possible. In the summary of EMO approaches that include some kinds of preferences, given in the introduction, we noted that the approach of Deb et al. (2005) bears some similarity to our approach. For that reason, we describe the differences between these two approaches in more detail. In contrast to Deb et al. (2005), we directly use the reference point based achievement function in the fitness evaluation in an indicator-based evolutionary algorithm IBEA (see, Zitzler and Kuenzli 2004). As the preference information is included in the indicator, the resulting algorithm does not require additional diversity preserving mechanisms, that is, fitness sharing. As a result of this integration of preference information and dominance valuation into a single indicator function, we can show that the consideration of preference information based on reference points is compliant with the Pareto dominance as given in Definitions 1 and 2.

As we have described in Section 3.2 and has also been noticed in Deb et al. (2005), the concepts of reference points and achievement functions are fundamentally related to each other. On the other hand, Deb et al. (2005) rank solutions in a population according to their weighted Euclidian distance from the reference point. Then solutions with the shortest distance get best crowding distance values and then such solutions are preferred in the selection. Thus, that approach is applicable to ranking based methods only.

Our approach is very different because we use an achievement function (2) or (3) (based on a reference point) embedded into a quality indicator. As we do not use the Euclidean distance, we can show compliance with Pareto dominance. In general, one can conclude that the new approach makes direct use of the components of the original reference point ideas (i.e., a reference point together with an achievement function) and therefore provides a strong link between the ideas originating from the MCDM community and the concept of evolutionary multi-objective optimization.

An important additional feature of our approach is the fact that we provide a convenient way to find the 'best' solution in the current population. The DM needs such support in selecting one of many solutions of the current population when dealing with problems involving more than two criteria. Other evolutionary approaches have not suggested any corresponding ideas. To be more specific, a natural way to measure goodness is to select the solution which has the best achievement function value. Using an achievement function (which minimizes maximum deviations) is natural because it has level sets which correspond to the dominance structure of Pareto optimality.

In what follows, we describe a preference-based evolutionary algorithm PBEA that incorporates preference information in IBEA. This algorithm can then be used as a part of an interactive solution method where the DM can iteratively study different solutions and specify different reference points.

4.1 Preference-based evolutionary algorithm PBEA

The basis of the preference-based evolutionary algorithm PBEA is the indicator-based evolutionary algorithm IBEA as described in Zitzler and Kuenzli (2004). The main concept of IBEA is to formalize preferences by a generalization of the dominance relation given in Definition 1. Based on a binary indicator I describing the preference of the DM, a fitness $F(x)$ is computed for each individual x in the current population. The fitness values of the individuals are used to drive the environmental and mating selection. The basic IBEA algorithm can be described as follows:

Basic IBEA Algorithm

Input: population size α; maximum number of generations N; fitness scaling factor κ;

Output: approximation of Pareto optimal set A;

Step 1 (Initialization): Generate an initial set of points P of size α; set the generation counter to $m = 0$;

Step 2 (Fitness Assignment): Calculate fitness values of all points in P, i.e., for all $x \in P$ set

$$F(x) = \sum_{y \in P \setminus \{x\}} (-e^{-I(y,x)/\kappa}). \tag{4}$$

Step 3 (Environmental Selection): Iterate the following three steps until the size of the population does no longer exceed α:

1. choose a point $x^* \in P$ with the smallest fitness value;
2. remove x^* from the population;
3. update the fitness values of the remaining individuals using (4).

Step 4 (Termination): If $m \geq N$ or another termination criterion is satisfied, then set A to the set of points in P that represent the nondominated solutions. Stop.

Step 5 (Mating Selection): Perform binary tournament selection with replacement on P in order to fill the temporary mating pool P'.

Step 6 (Variation): Apply recombination and mutation operators to the mating pool P' and add the resulting points to P. Increment the generation counter m and go to *Step 2*.

In the numerical experiments we are using a slightly improved version of the above algorithm. It scales the objective and indicator values and has been called adaptive IBEA (see Zitzler and Kuenzli 2004).

Obviously, the calculation of the fitness according to (4) using a dominance preserving binary quality indicator I is one of the main concepts in the indicator-based evolutionary algorithm.

Definition 3 *A binary quality indicator I is called dominance preserving if the following relations hold:*

$$f(x) \text{ dominates } f(y) \Rightarrow I(y,x) > I(x,y)$$
$$f(x) \text{ dominates } f(y) \Rightarrow I(v,x) \geq I(v,y) \text{ for all } v \in X.$$

According to the definition above, one can consider the quality indicator I to be a continuous version of the dominance relation given in Definition 1. As we will see, the degree of freedom available can be used to take into account the concept of an achievement function as discussed in Section 3.2.

The environmental selection (Step 3) as well as the mating selection (Step 5) prefer solutions with a high fitness value. The fitness measure $F(x)$ is a measure for the loss in quality if x is removed from the population P. To this end, a given variable x is compared to all other variables y in the current population P, whereas the exponent in expression (4) gives the highest influence to the variable y with the smallest indicator $I(y, x)$. In Zitzler and Kuenzli (2004), it is shown that if the binary quality indicator used in (4) is dominance preserving, then we have

$$f(x) \text{ dominates } f(y) \Rightarrow F(x) > F(y).$$

Therefore, the fitness computation is compliant with the Pareto dominance relation.

In Zitzler and Kuenzli (2004), one of the dominance preserving indicators used is the additive epsilon indicator defined as

$$I_\epsilon = \min_\epsilon \{ f_i(x) - \epsilon \leq f_i(y) \text{ for } i = 1, \dots, k \}. \tag{5}$$

Its value is the minimal amount ϵ by which one needs to improve each objective, i.e., replace $f_i(x)$ by $f_i(x) - \epsilon$ such that it just dominates $f(y)$, i.e., $f_i(x) - \epsilon \leq f_i(y)$ for all $i = 1, \dots, k$. Using the additive epsilon indicator in (4) results in a diverse approximation of the Pareto optimal solutions.

In order to take preference information into account, we use the achievement function defined in (3). At first, we normalize this function to positive values for a given set of points P

$$s(g, f(x), \delta) = s_g(f(x)) + \delta - \min_{y \in P} \{ s_g(f(y)) \}, \tag{6}$$

where the specificity $\delta > 0$ gives the minimal value of the normalized function. The preference-based quality indicator can now be defined as

$$I_p(y, x) = I_\epsilon(y, x) / s(g, f(x), \delta). \tag{7}$$

This quality indicator can now be used instead of (5) where the base set P used in the normalization (6) is the population P. Because we take preferences into account, we refer to this as a preference-based evolutionary algorithm (PBEA). In this way, we modify the binary indicator with an achievement function based on a reference point. As the achievement function represents directly the preference information contained in a reference point, it is independent of the form of the Pareto front approximation contained in the population P. The configuration follows the idea of some crowding operators but in a reverse way. The specificity $\delta > 0$ now allows to set how large the 'amplification' of the epsilon indicator for solutions close to the reference point should be. In other words, by increasing the value of specificity δ from zero gives us a wider set of solutions surrounding the solution where the reference point was projected and if we set a low value for δ, we get solutions in the close neighborhood of the projected reference point. The parameter δ is just a constant that is added to the achievement function and therefore, its effect does not depend on the shape of the Pareto front approximation.

Figure 2 illustrates the effect of the specificity δ. As in Figure 1, we suppose that we have a reference point A, a Pareto optimal set and the projected reference point A'. The two additional graphs illustrate the functions $s(g, f(y), \delta)$ and $1/s(g, f(y), \delta)$. It can be seen that depending on the relative position to the projected reference point, the points of the Pareto optimal set get different weights. In addition, the smaller the specificity δ, the higher is the relative preference towards points close to the projected reference point.

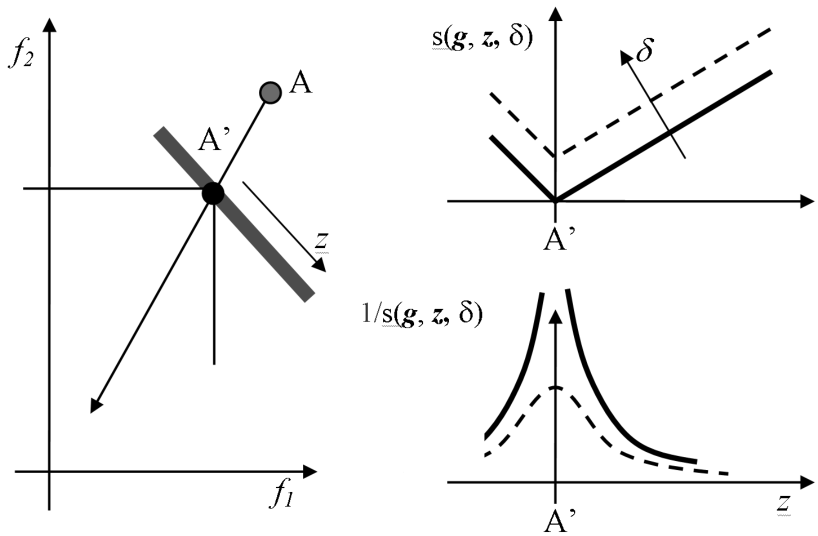

Figure 2: Illustrating the normalized achievement scalarizing function $s(g, f(y), \delta)$ for preferring solutions that are close to a projected reference point.

It remains to be shown that the new preference-based indicator defined in (7) is dominance preserving and, in this case, the resulting fitness evaluation is compliant with the Pareto dominance.

Theorem 1 *The binary quality indicator I_p as defined in (7) is dominance preserving.*

Proof: It has been shown in Zitzler and Kuenzli (2004) that the additive epsilon indicator given in (5) is dominance preserving. Therefore, if $f(x)$ dominates $f(y)$, then we have $I_\epsilon(y, x) > I_\epsilon(x, y)$. From the definition of the scalarizing function (3) and the dominance in Definition 1, we find that if $f(x)$ dominates $f(y)$, then $s_g(f(x)) \leq s_g(f(y))$ which implies $s(g, f(x), \delta) \leq s(g, f(y), \delta)$. As the normalized scalarizing functions $s(g, f(x), \delta)$ and $s(g, f(y), \delta)$ are positive, we find $I_\epsilon(y, x) > I_\epsilon(x, y)$ which implies $I_\epsilon(y, x)/s(g, f(x), \delta) > I_\epsilon(x, y)/s(g, f(y), \delta)$ and therefore $I_p(y, x) > I_p(x, y)$. In a similar way, we can conclude that if $f(x)$ dominates $f(y)$, then we have $I_\epsilon(v, x) > I_\epsilon(v, y)$ and therefore $I_p(v, x) > I_p(v, y)$. ■

In the next section, we show how the preference-based evolutionary algorithm PBEA can be incorporated into an interactive method for multiobjective search.

4.2 Interactive method

The evolutionary algorithm PBEA defined in Section 4.1 can be used in an interactive fashion, for example, in the following way:

- *Step 0 Initialization:* Find a rough approximation of the Pareto optimal set with a small population using the PBEA algorithm without using a specific reference point, that is, with indicator I_ϵ. Select a small set of solutions to characterize the approximation and display the set to the DM for evaluation.

- *Step 1 Reference Point:* Ask the DM to specify desired aspiration level values for the objective functions, that is, a reference point.

- *Step 2 Local Approximation:* Use the reference point information in the preference-based evolutionary algorithm PBEA as described in Section 4.1 to generate a local approximation of the Pareto optimal set.

- *Step 3 Projection of Reference Point:* Among the solutions generated in Step 2, display to the DM the nondominated solution giving the smallest value for the achievement function.

- *Step 4: Termination:* If the DM is willing to continue the search, go to Step 1; otherwise obviously the DM has found a good estimate as the most preferred solution and (s)he stops the search.

In Step 0, the small set of solutions can be selected, for example, using clustering. This step can also be replaced by showing the DM only the best and the worst criterion values found among the nondominated solutions generated. This gives the DM some understanding about the feasible solutions in the problem and helps in specifying the reference point. In the algorithm, and in Step 3 in particular, the idea is to avoid overloading the DM with too much information when the problem in question has more than two objectives and a natural visualization of the solutions on a plane is not possible.

The algorithm offers different possibilities for the DM in directing the search into a desired part of the Pareto optimal set. Some of them will be demonstrated in Section 5 with computational tests. In Step 3, if the DM wants to consider several nondominated solutions, we can display solutions giving next best values for the achievement function or use clustering in the current population. The first-mentioned option is based on the fact that the achievement function offers a natural way to order solutions so that the DM does not have to compare all of them. Naturally, the DM can also use different projections and consider only some of the objectives at a time. In the next step, the DM can select the next reference point according to her/his hopes. Alternatively, some solution in the current population can be selected as the reference point. This means that the DM has found an interesting solution and wishes to explore its surroundings. In the examples in Section 5, we will demonstrate how the approximations of the Pareto optimal set get more accurate from iteration to iteration.

As said, we want to avoid overloading the DM with too much information. However, if the DM wants to consider the current population or a part of it (instead of only one or some solutions in it) a possibility worth consideration is to use some of the tools developed for discrete MCDM problems. By using them, the DM can get visual support in finding the most preferred solution of the current population. As an example we may mention VIMDA (Korhonen 1988), which is a visualization-based system for large-scale discrete multiple criteria problems. Another example is knowCube, a visualization system based on spider web charts (Trinkaus and Hanne 2005). Further methods for dealing with discrete alternatives can be found, for example, in Olson (1996). Widely used value paths are also one possibility for visualization (for further visualizations, see, e.g., Miettinen 1999, 2003). More recent ideas are to use so-called box-indices (Miettinen et al., to appear) or clustering (Aittokoski et al., to appear).

As far as stopping the solution process is concerned, the DM's satisfaction with the current best solution is the main stopping criterion. Alternatively, if the reference point is selected among the members of the previous population and there is no significant change obtained in the best solution of the next iteration, we can say that we have reached a final solution.

Let us point out that our approach can be generalized for several reference points given at the same iteration. This is practical if the DM wishes to study several parts of the Pareto optimal set at the same time. If we have several reference points g^i that should be taken into account simultaneously, we just replace the denominator in (7) by the minimum normalized scalarization for all reference points, i.e.,

$$I_p(x, y) = I_\epsilon(x, y) / \min_i \{s(g^i, f(y), \delta)\}.$$

5 Experimental results

The following experimental results are based on an implementation of PBEA in the framework PISA (http://www.tik.ee.ethz.ch/pisa) that contains implementations of well-known evolutionary algorithms such as NSGA-II, SPEA2 and IBEA as well as various test problems, see Bleuler et al. (2003). In the examples, we use scaling $w_i = 1$ for all $i = 1, \ldots, k$ in the achievement functions because our criteria happen to have similar ranges. Otherwise, scaling factors like $w_i = 1/\mathit{range}_i$ should be used, as discussed in Section 3.2.

At first, we give simulation results using well-known two-dimensional benchmark functions, namely ZDT1 and ZDT3, see Zitzler et al. (2000). As can be seen in Figure 3, a run of the multiobjective optimizer IBEA for ZDT1 without preference information yields a Pareto approximation containing points that are almost equally spaced (denoted by triangles). When using the reference point (0.6, 1.0) (denoted by a big star) with a high specificity of $\delta = 0.1$, a run of PBEA (denoted by stars) results in an approximation that (a) dominates a part of the previous run without preferences and (b) is concentrated around the projected reference point (which has a circle around it). Remember that by a projected reference point we mean a solution giving the minimum value for the achievement function used. In order to demonstrate the effect of decreasing specificity, we set $\delta = 0.02$, and then, as discussed earlier, the concentration is even more visible (the points are denoted by boxes). Let us point out here that the optimal front was not yet achieved because we used on purpose a small population size of 20 in order to make the spacing of the solutions and development of different populations more visible. In the figure, one can see that when compared to the population obtained using IBEA (without preference information), using PBEA (with the same population sizes and generation numbers) we get better solutions, that is, solutions that are closer to the actual Pareto optimal set. In other words, by including preference information we get a better approximation with the same computational effort, and from iteration to iteration the approximation gets more accurate, as will be seen in what follows.

Figure 4 represents a possible interaction with a DM using the preference based algorithm PBEA in ZDT1 with population size 20. At first, the DM performs a run without preference information with 300 generations and selects a point in the population as the reference point for the next iteration (with specificity $\delta = 0.05$ and 400 generations). This process is repeated again with a new reference point and a new run is performed with 500 generations. Now, a preferred solution from the last run is chosen as a reference point for a final run (with specificity $\delta = 0.02$ and 500 generations) in order to focus the search even more.

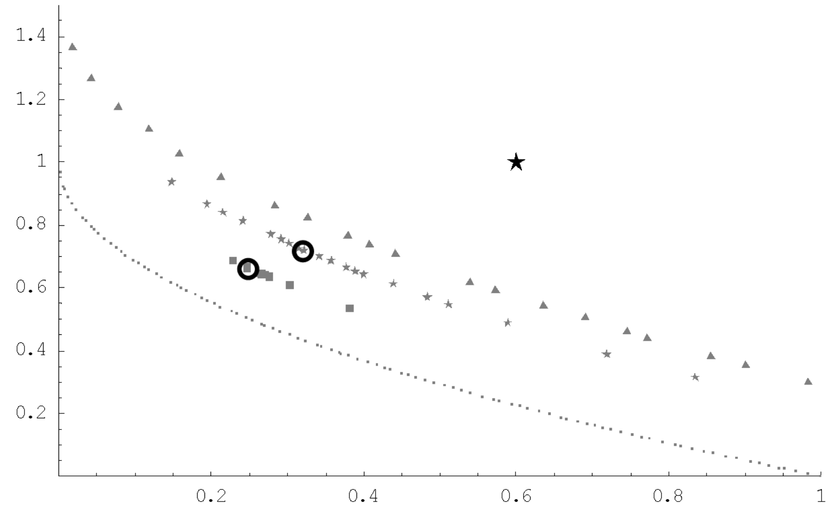

Figure 3: Results of three optimization runs for the benchmark problem ZDT1 with 100 decision variables and population size of 20 using 500 generations. *Bottom Curve*: Approximation of the Pareto optimal set. *Triangle*: optimization without preference information. *Star*: Preference-based search with reference point (0.6, 1.0) (indicated by a big black star) and specificity $\delta = 0.1$. *Box*: reference point (0.6, 1.0) as indicated and specificity $\delta = 0.02$. The circles point to solutions with the best achievement scalarizing function values.

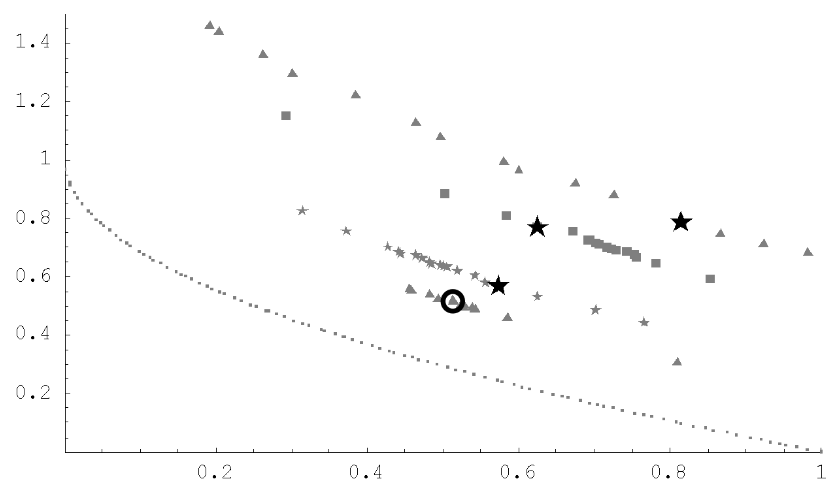

Figure 4: Possible interaction of a DM with the preference-based optimization tool in ZDT1 with population size 20. *Triangle:* Search using IBEA (300 generations). *Box:* Preference-based PBEA using reference point (see black star) from the first iteration ($\delta = 0.05$, 400 generations). *Star:* Preference-based search using reference point (see black star) from the second iteration ($\delta = 0.05$, 500 generations). *Triangle:* Preference-based search using reference point from the third iteration ($\delta = 0.03$, 500 generations). The circle denotes the optimal solution of the achievement function, that is, projected reference point.

Here, specificity (and the number of generations) was varied to demonstrate that this is possible but it is not necessary to vary the value between iterations. Let us again point out that we used small amount of generations and population sizes in order to clarify evolvement of the solution process. In practice, the approach can easily reach the Pareto optimal set if the usual parameter settings for the population size and number of generations are used.

The next three Figures 5, 6 and 7 show the effect of different locations of reference points, i.e., optimistic or pessimistic ones. To this end, we use another benchmark function ZDT3, see Zitzler et al. (2000), which is characterized by a discontinuous Pareto optimal set. A run with IBEA without any preference information yields the set of points shown in Figure 5 as triangles. It can be guessed that the Pareto optimal set consists of 5 disconnected subsets. A PBEA optimization using the pessimistic reference point (0.7, 2.5) (denoted by a black star) (with specificity $\delta = 0.03$) yields the points shown as boxes. Again, they dominate points that have been determined using optimization without preference information and are concentrated around the projection of the reference point (i.e., a solution with a circle). Similar results are obtained if an optimistic reference point (0.4, 2.7) (with specificity $\delta = 0.02$) is chosen, see Figure 6. The larger the distance between the reference point and the Pareto approximation, the smaller is the effect of concentrating the search around the projection of the reference point. This can clearly be seen in Figure 7 where the optimistic reference point (0.3, 2.6) with (specificity $\delta = 0.01$) is chosen. In all the figures, circles denote solutions with the best achievement function value in the current population.

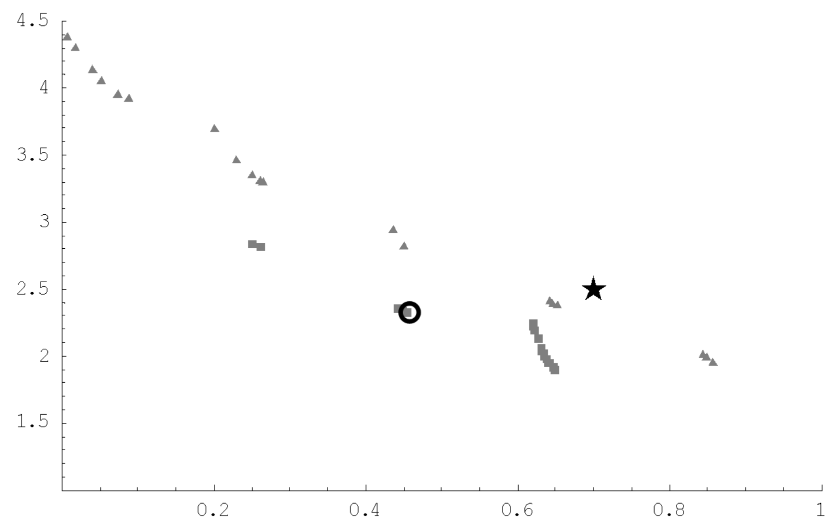

Figure 5: Results of two optimization runs for the benchmark problem ZDT3 with 100 decision variables, population size 20 and 100 generations. *Triangle*: IBEA optimization without preference information. *Box*: Preference-based PBEA with pessimistic reference point (0.7, 2.5) (indicated by a big star) and specificity $\delta = 0.03$. Circled solution is the projected reference point.

Figure 8 is again based on runs for the benchmark problem ZDT3. Here, we use two reference points (0.25, 3.3) and (0.85, 1.8) (with $\delta = 0.03$ each). This example models a DM who intends to concentrate her/his search on two areas of the Pareto approxima-

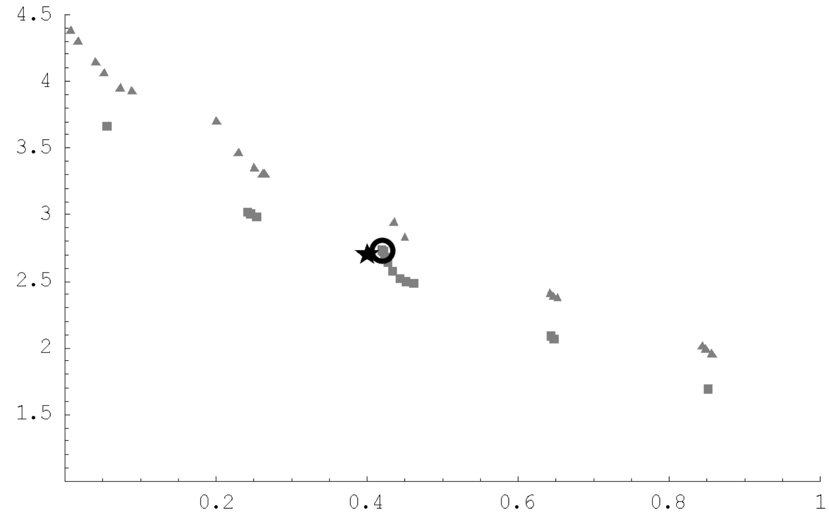

Figure 6: Results of two optimization runs for the benchmark problem ZDT3 with 100 decision variables, population size 20 and 100 generations. *Triangle*: IBEA optimization without preference information. *Box*: Preference-based PBEA with optimistic reference point (0.4, 2.7) (indicated by a big star) and specificity $\delta = 0.02$. Circled solution is the projected reference point.

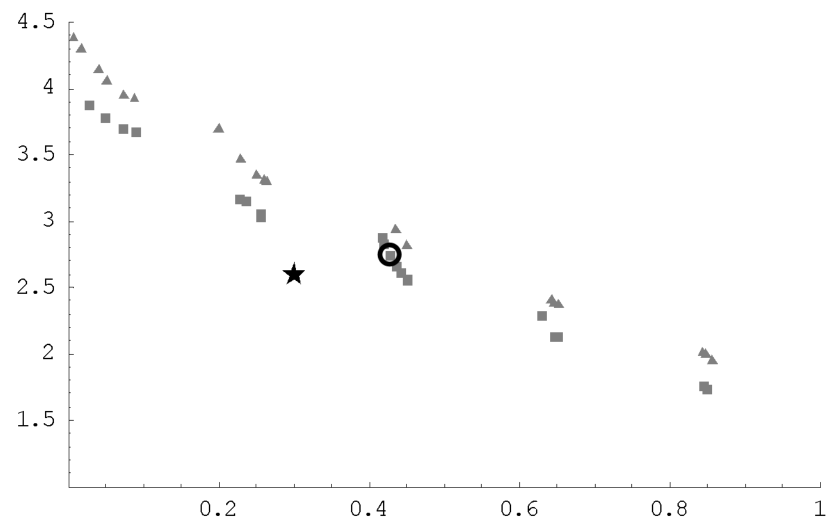

Figure 7: Results of two optimization runs for the benchmark problem ZDT3 with 100 decision variables, population size 20 and 100 generations. *Triangle*: IBEA optimization without preference information. *Box*: Preference-based PBEA with optimistic reference point (0.3, 2.6) (indicated by a big star) and specificity $\delta = 0.01$. Circled solution is the projected reference point.

tion simultaneously. As can be seen, the search concentrates on the projections of the two reference points as expected.

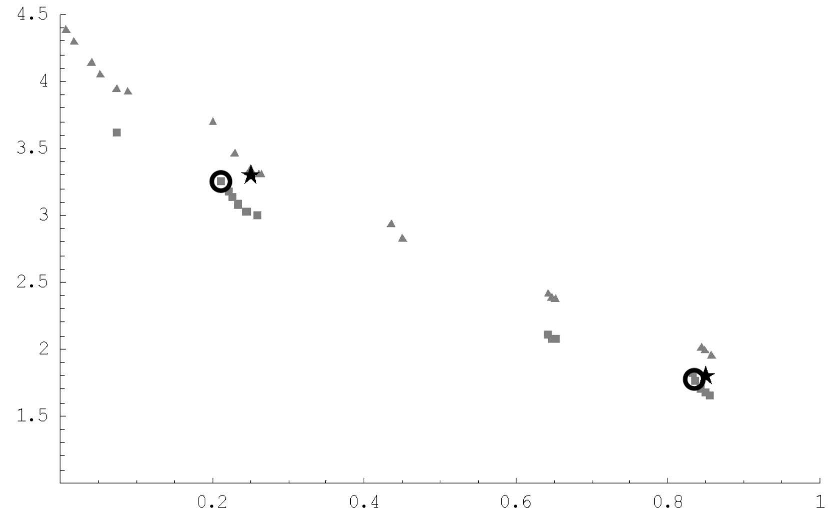

Figure 8: Results of two optimization runs for the benchmark problem ZDT3 with 100 decision variables, population size 20 and 100 generations. *Triangle*: IBEA optimization without preference information. *Box*: Preference-based PBEA with two reference points (0.25, 3.3) and (0.85, 1.8) (indicated by big stars) and specificity $\delta = 0.03$. Circled solutions are the projected reference points.

Finally, we consider one more problem with five criteria, see Miettinen et al. (2003). The problem is related to locating a pollution monitoring station in a two-dimensional decision space. The five criteria correspond to the expected information loss as estimated by five different experts. Therefore, the DM needs to find a location that balances the five possible losses. The problem formulation is as follows:

The decision variables have box constraints $x_1 \in [-4.9, 3.2]$, $x_2 \in [-3.5, 6]$. The criteria are based on the function

$$f(x_1, x_2) = -u_1(x_1, x_2) - u_2(x_1, x_2) - u_3(x_1, x_2) + 10$$

where

$$\begin{aligned} u_1(x_1, x_2) &= 3(1 - x_1)^2 e^{-x_1^2 - (x_2+1)^2} \\ u_2(x_1, x_2) &= -10(x_1/4 - x_1^3 - x_2^5) e^{-x_1^2 - x_2^2} \\ u_3(x_1, x_2) &= 1/3 \cdot e^{-(x_1+1)^2 - x_2^2}. \end{aligned}$$

The actual objective functions to be minimized are

$$\begin{aligned} f_1(x_1, x_2) &= f(x_1, x_2) \\ f_2(x_1, x_2) &= f(x_1 - 1.2, x_2 - 1.5) \\ f_3(x_1, x_2) &= f(x_1 + 0.3, x_2 - 3.0) \\ f_4(x_1, x_2) &= f(x_1 - 1.0, x_2 + 0.5) \\ f_5(x_1, x_2) &= f(x_1 - 0.5, x_2 - 1.7). \end{aligned}$$

In order to get a rough overview about the complexity of the problem, the following two Figures 9 and 10 represent a projected scan of the Pareto optimal set. They have been produced simply by probing the decision variables on a regular equidistant mesh and selecting the Pareto optimal points from the set of solutions received. These two figures demonstrate the need of preference based approaches and show how difficult it is to study approximations of Pareto optimal sets when the problem has more than two objectives.

Figure 9 shows solutions projected to the second and third dimension of the criterion space. The grey level of the points corresponds to the first criterion. In a similar way, Figure 10 shows the projection on the fourth and fifth criterion (and the grey level again indicates values of the first criterion). It can be observed that the optimization problem is highly non-linear and the Pareto optimal set is discontinuous.

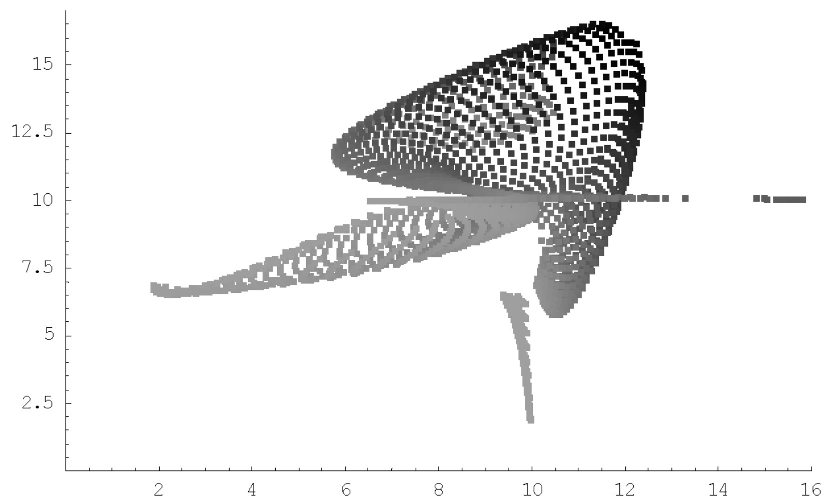

Figure 9: Approximated Pareto optimal set of a multiobjective optimization problem with 2 decision variables and 5 objective functions, see Miettinen et al. (2003). The projection on objectives f_2 and f_3 is shown where the grey levels of the points correspond to values of f_1.

Usually, in evolutionary approaches it is assumed that the graphical representation of the Pareto front is self-explanatory and the DM can easily select her/his most preferred solution from there. Figures 9 and 10 demonstrate very clearly that this is not necessarily the case with more criteria. It simply gets too difficult for the DM to analyze the Pareto optimal solutions generated because (s)he can see only different projections of the 5-dimensional criterion space. When using the preference-based approach, we do not need to illustrate the whole Pareto front but it is enough to provide a rough approximation of it. The DM then directs the search by varying reference points and we can easily identify the best solution of the current population with the help of the achievement function. Thus, projections and other cognitively difficult visualizations are not needed at all. In Table 1, we illustrate how our approach finds solutions corresponding to different reference points, that is, their projections. (In all the runs, population size of 200 was used with 100 generations.)

In Table 1, the rows correspond to different runs of the evolutionary multiobjective optimization algorithms, and the columns correspond to the two reference points that

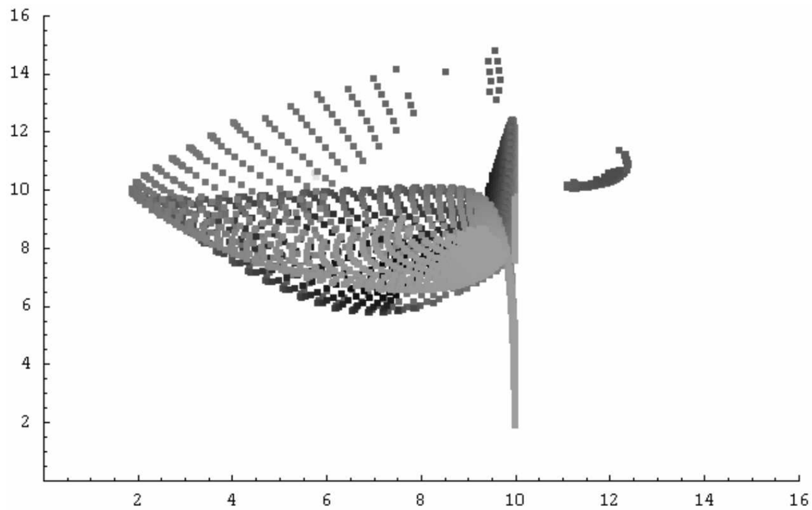

Figure 10: For the same problem as in Figure 9, the projection on objectives f_4 and f_5 is shown where the grey levels of the points correspond to values of f_1.

have been used to evaluate the achievement function. The objective vectors in the table give the best value for the achievement function in question (as mentioned in Step 3 of the interactive method). For comparative reasons, also the corresponding achievement function values are recorded even though they are not normally shown to the DM (as before, the smaller the value, the better the solution).

		g *ref1* = (10, 10, 10, 10, 10)	*ref2* = (12, 11, 10, 9, 8)
IBEA	$f(x)$	(8.55, 7.54, 8,17, 9.61, 9.03)	(9.72, 8.4, 9.94, 7.93, 6.83)
	$s_g(f(x))$	-0.39	-0.06
PBEA	$f(x)$	(8.87, 8.99, 8,65, 8.84, 9.05)	(9.21, 9.93, 9.29, 8.39, 7.64)
$\delta = 0.1$	$s_g(f(x))$	-0.95	-0.36
PBEA	$f(x)$	(8.95, 8.96, 8,60, 8.92, 8.96)	(9.31, 10.0, 9.37, 8.39, 7.37)
$\delta = 0.02$	$s_g(f(x))$	-1.04	-0.61

Table 1: Solutions giving minimal achievement function values for different reference points and values of achievement function (negative numbers).

The reference point *ref1* = (10, 10, 10, 10, 10) represents preferred equal losses in the different objectives of the problem. When the components of the weight vector w in (3) are equal, the algorithm is seeking for a solution where all values of objective functions are as equal as possible. The other reference point *ref2* = = (12, 11, 10, 9, 8) demonstrates how preferring decreasing values is reflected in the solutions produced.

In Table 1, we can see that the preference-based evolutionary algorithm PBEA gives better solutions in terms of minimizing the achievement function than the basic evolutionary approach IBEA. It means that the values of the objective functions are more equal in the solution provided by PBEA than IBEA. In case of the first reference point *ref1* we find the achievement values -0.39 for IBEA and -1.04 for PBEA (with specificity 0.02). In addition, a lower specificity leads to a better approximation ($-1.04 < -0.95$), as explained earlier. Similar observations hold for the second reference point as well. The DM may continue by specifying a new reference point or by selecting the final solution as described in the interactive method in Section 4.

Out of curiosity, let us suppose, for example, that we look at the PBEA population based on reference point *ref1* (and $\delta = 0.02$) and try to find in this population a point that is closest to the other reference point *ref*2. Then the best point has an achievement function value of -0.05 which is much worse than the best value of -0.61 given in the table. If we use the plain IBEA algorithm, i.e., no preference information is provided, then the point closest to the reference point *ref2* has an achievement function value of -0.06 which again is much worse then -0.61. As a result we can say that it makes a substantial difference whether we optimize with respect to a reference point used in fitness evaluation, or some other reference point, as expected.

Our small illustrative example just demonstrates that the preference-based evolutionary approach is very helpful for the DM when (s)he wants to find the most preferred solution for her/his multiobjective optimization problem involving more than two objectives. The DM does not have to study different projections (if (s)he does not want to) because we can conveniently identify the best solution of the current population with the help of the achievement function.

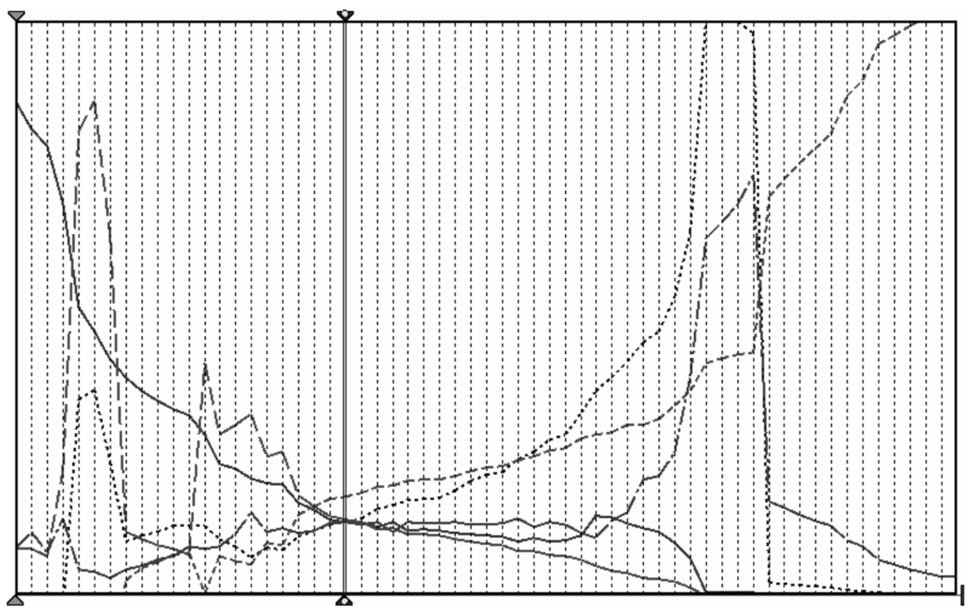

Figure 11: Illustration of 70 solutions of the final population using value paths. The example screen is adopted from the implementation of the recent unpublished version of VIMDA (Korhonen 1988).

If the DM, after all, wants to compare different solutions of the population, it is possible to use, for example, value paths as mentioned in Section 4.2. In Figure 11, we have 70 solutions of the last PBEA run with reference point *ref1* and $\delta = 0.02$. Each criterion is represented by a path and vertical lines correspond to solutions. The solution denoted by a bold vertical line is the best solution listed in Table 1. As mentioned earlier, this solution is balanced in terms of objective function values as the corresponding reference point is and this fact can be easily seen in the figure. In Figure 11, lower criterion values are in the top part of the figure since they stand for more preferred values.

We have here demonstrated how PBEA can be used in the first iterations of our interactive algorithm. From here, the DM can continue by specifying a reference point according to her/his preferences. Even from these examples one can see how concentration on a subspace of the Pareto optimal set means that the quality of the approximation improves and the population sizes can be kept relatively small, which means savings in computational cost.

Let us point out that if computational cost is an important factor to consider, there is no need to restart the optimization whenever a new reference point has been specified. Instead, it is possible to include the nondominated solutions of the previous population into the temporary mating pool of the next iteration of the interactive algorithm. In this way, the diversity of an initial mating pool is combined with the knowledge about interesting regions of the search space. The search can now benefit from the previous iteration if the new reference point is in the same subspace of the Pareto optimal set as the previous was. Otherwise, the old solutions will be removed in the environmental selection and new, better solutions will be generated in the region of interest.

6 Conclusions

We have introduced a new preference-based evolutionary algorithm that incorporates preference information coming from a DM. By setting desirable values for objective functions as a reference point, the DM can conveniently study such parts of the Pareto optimal set that (s)he finds interesting and the whole Pareto optimal set does not have to be generated with equal accuracy and population sizes can be kept rather small.

In multiobjective optimization, reference points are projected to the Pareto optimal set with the help of achievement functions. Our innovative idea is to include an achievement function in the fitness evaluation of an evolutionary algorithm. Our preference-based evolutionary algorithm can be used as an integral part of an interactive multiobjective optimization algorithm. In this way, we get solutions in the neighborhood of the projected reference point (i.e., solutions concentrated around the reference point projected to the Pareto optimal set) and we do not waste effort in computing solutions in uninteresting areas. We can adjust how wide a neighborhood we are interested in by setting a value for a specificity parameter. Adjusting the specificity parameter value is a topic for further research. (Naturally, using an interactive method necessitates that the DM has time and interest in taking part in the solution process.)

Our computational experiments indicate that the approximations produced utilizing reference point information give the DM reliable information on the solutions available, the approximations get more accurate than without preference information as well as from iteration to iteration and the DM can find the most preferred solution as the final one conveniently. In addition, we have an intuitive way available for finding good solutions in a population, that is, the achievement function used helps the DM in ordering solutions and we are, thus, able to solve problems with more than two criteria.

In summary, our solution philosophy is different from the conventional EMO approaches and it has the benefits of incorporating preference information (expressed in the form easily understandable for the DM as a reference point), saving computational cost and providing a convenient way of identifying the best solution of the current population (best reflecting the preferences expressed in the form of a reference point).

This last point is especially important as, when solving problems with more than two objectives, simple visualization is not enough to help the DM find the most preferred solution. The idea of including preference information in fitness evaluation can be used in other EMO approaches besides IBEA as well.

Acknowledgement

The research was partly supported by the Academy of Finland and the Jenny and Antti Wihuri Foundation. The authors wish to thank Prof. Roman Slowinski for his valuable ideas.

References

Aittokoski T., Äyrämö S., Miettinen K. (2008): Clustering aided approach for decision making in computationally expensive multiobjective optimization. Optimization Methods and Software (to appear).

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

Bleuler S., Laumanns M., Thiele L., Zitzler E. (2003): PISA - A Platform and Programming Language Independent Interface for Search Algorithms, in: C.M. Fonseca, P.J. Fleming, E. Zitzler, K. Deb, L. Thiele (Eds.), Evolutionary Multi-Criterion Optimization (EMO 2003), Springer-Verlag, Faro, Portugal, 494–508.

Branke J., Deb K. (2004): Integrating User Preferences into Evolutionary Multi-Objective Optimization, KanGal Report 2004, Indian Institute of Technology. Kanpur, India.

Branke J., Kaussler T., Schmeck H. (2001): Guidance in evolutionary multi-objective optimization. Advances in Engineering Software, 32, 499–507.

Branke J., Deb K., Miettinen K., Slowinski R. Eds. (2008): Multiobjective Optimization: Interactive and Evolutionary Approaches. Springer-Verlag, Berlin, Heidelberg.

Cvetkovic D., Parmee I. (1999): Genetic Algorithm-based Multi-objective Optimization and Conceptual Engineering Design, in: Proceedings of the Congress on Evolutionary Computation (CEC '99), vol. 1, IEEE Press, 29–36.

Cvetkovic D., Parmee I. (2002): Preferences and their application in evolutionary multiobjective optimization. IEEE Transactions in Evolutionary Computation, 6(1), 42–57.

Chankong V., Haimes Y.Y. (1983): Multiobjective Decision Making Theory and Methodology, Elsevier Science Publishing, New York.

Coello C.A.C (2000): Handling Preferences in Evolutionary Multiobjective Optimization: A Survey, in: Proceedings of the 2000 Congress on Evolutionary Computation, IEEE Service Center, Piscataway, NJ, 30–37.

Deb K. (1999a): Multi-Objective Evolutionary Algorithms: Introducing Bias Among Pareto-Optimal Solutions. KanGAL Report 99002, Indian Institute of Technology, Kanpur, India.

Deb K. (1999b): Solving goal programming problems using multi-objective genetic algorithms. In Proceedings of the Congress on Evolutionary Computation (CEC '99), IEEE Press, 77–84.

Deb K., Sundar J., Uday B.R.N. (2005): Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. KanGAL Report 2005012, Indian Institute of Technology. Kanpur, India. And Proceedings of the 8th annual conference on Genetic and evolutionary computation, 2006, Seattle, 635–642.

Fonseca C., Fleming P. (1993): Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization, in: S. Forrest (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kauffman Publishers, 416–423.

Geoffrion A.M., Dyer J.S., Feinberg A. (1972): An interactive approach for multi-criterion optimization, with an application to the operation of an academic department. *Management Science*, 19, 357–368.

Greenwood G., Hu X., D'Ambrosio J. (1997): Fitness Functions for Multiple Objective Optimization Problems: Combining Preferences with Pareto Rankings, in: R. K. Belew, M. D. Vose (Eds). Foundations of Genetic Algorithms 4, Morgan Kauffman Publishers, 437–455.

Hanne, T. (2006): Interactive Decision Support Based on Multiobjective Evolutionary Algorithms, in: H.-D. Haasis, H. Kopfer, J. Schönberger (Eds.). Operations Research Proceedings 2005, Springer-Verlag, 761–766.

Hwang C.-L., Masud A.S.M. (1979): Multiple Objective Decision Making - Methods and Applications, Springer-Verlag, Berlin, Heidelberg.

Jaszkiewicz A., Slowinski R. (1999): The 'Light Beam Search' approach - an overview of methodology and applications, *European Journal of Operational Research*, 113, 300–314.

Klamroth K., Miettinen K. (2008): Integrating Approximation and Interactive Decision Making in Multicriteria Optimization, Operations Research, 56, 222234.

Koopmans T.C. (1971): Analysis and Production as an Efficient Combination of Activities, in: T.C. Koopmans (Ed.), Activity Analysis of Production and Allocation (originally published in 1951), Yale University Press, New Haven, London, 33–97.

Korhonen, P. (1988): A Visual Reference Direction Approach to Solving Discrete Multiple Criteria Problems, *European Journal of Operational Research*, 34, 152–159.

Korhonen P., Laakso J. (1986): A visual interactive method for solving the multiple criteria problem, *European Journal of Operational Research*, 24, 277–287.

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

Kuhn H.W., Tucker A.W. (1951): Nonlinear Programming, in: J. Neyman (Ed.), Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, Berkeley, Los Angeles, 481–492.

Larichev O.I. (1992): Cognitive validity in design of decision-aiding techniques. *Journal of Multi-Criteria Decision Analysis*, 1, 127–138.

Luque M., Miettinen K., Eskelinen P., Ruiz F. (2009): Incorporating preference information in interactive reference point methods for multiobjective optimization. Omega, 37, 450-462.

Massebeuf S., Fonteix C., Kiss L., Marc StateI., Pla F., Zaras K. (1999): Multicriteria Optimization and Decision Engineering of an Extrusion Process Aided by a Diploid Genetic Algorithm, in: Proceedings of the Congress on Evolutionary Computation (CEC '99), IEEE Press, 14–21.

Miettinen K. (1999): Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, Boston.

Miettinen K. (2003): Graphical Illustration of Pareto Optimal Solutions, in: T. Tanino, T. Tanaka, M. Inuiguchi (Eds.), Multi- Objective Programming and Goal Programming: Theory and Applications, Springer-Verlag, Berlin, Heidelberg, 197–202.

Miettinen K, Mäkelä M.M. (1995): Interactive bundle-based method for nondifferentiable multiobjective optimization: NIMBUS. *Optimization*, 34(3), 231–246.

Miettinen K, Mäkelä M.M. (2006): Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research,* 170, 909–922.

Miettinen K, Lotov A.V., Kamenev G.K., Berezkin V.E. (2003): Integration of two multiobjective optimization methods for nonlinear problems. *Optimization Methods and Software,* 18(1), 63–80.

Miettinen, K., Molina, J., Gonzlez, M., Hernndez-Daz, A., Caballero, R.: Using box indices in supporting comparison in multiobjective optimization. European Journal of Operational Research. To appear (2008).

Nakayama H., Sawaragi Y. (1984): Satisficing trade-off method for multiobjective programming, in: M. Grauer, A.P. Wierzbicki (Eds.), Interactive Decision Analysis. Springer-Verlag, StateBerlin, 113–122.

Olson D. (1996): Decision Aids for Selection Problems. Springer-Verlag, New York.

Pareto V. (1906): Manuale di Economia Politica. Piccola Biblioteca Scientifica, Milan, Note: Translated into English by Ann S. Schwier (1971), Manual of Political Economy, MacMillan, London.

Parmee I., Cvetkovic D., Watson A., Bonham C. (2000): Multi-Objective satisfaction within an interactive evolutionary design environment, *Journal of Evolutionary Computation*, 8 (2), 197–222.

Phelps S., Köksalan M. (2003): An interactive evolutionary metaheuristic for multiobjective combinatorial optimization, *Management Science*, 49(12), 1726–1738.

Rekiek B., De Lit P., Pellichero F., LEglise T., Falkenauer E. and Delchambre A. (2000): Dealing with User's Preferences in Hybrid Assembly Lines Design. In Proceedings of the MCPL2000 Conference.

Sakawa M., Kato K., (2002) An interactive fuzzy satisficing method for general multiobjective 0-1 programming problems through genetic algorithms with double strings based on a reference solution, *Fuzzy Sets and Systems*, 125(3), 289–300.

Sawaragi Y., Nakayama H., Tanino T. (1985): Theory of Multiobjective Optimization, Academic Press, Orlando, Florida.

Steuer R.E. (1986): Multiple Criteria Optimization: Theory, Computation, and Application, John Wiley & Sons.

Trinkaus H.L., Hanne T. (2005): knowCube: A visual and interactive support for multicriteria decision making, *Computers & Operations Research* 32, 1289–1309.

Wierzbicki A. (1980): The Use of Reference Objectives in Multiobjective Optimization, in: G. Fandel, T. Gal (Eds.), Multiple Objective Decision Making, Theory and Application, Springer-Verlag, 468–486.

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

Zionts S., Wallenius J. (1976): An interactive programming method for solving the multiple criteria problem, *Management Science*, 22(6), 652–663.

Zitzler E., Deb K., Thiele L. (2000): Comparison of multiobjective evolutionary algorithms: empirical results, *Evolutionary Computation*, 8(2), 173–195.

Zitzler E., Kuenzli S. (2004): Indicator-based Selection in Multiobjective Search, in: X. Yao, E. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J. Rowe, P. Tino, A. Kabán, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature - PPSN VIII, 8th International Conference, Proceedings, Springer-Verlag, Berlin, 832–842.