Interactive Evolutionary Multiobjective Optimization Using Robust Ordinal Regression

Jürgen Branke[1], Salvatore Greco[2], Roman Słowiński[3,4], and Piotr Zielniewicz[3]

[1] Warwick Business School, University of Warwick, Coventry, CV4 7AL, UK
juergen.branke@wbs.ac.uk
[2] Faculty of Economics, University of Catania, 95129 Catania, Italy
salgreco@unict.it
[3] Institute of Computing Science, Poznan University of Technology, 60-965 Poznan, Poland
{roman.slowinski,piotr.zielniewicz}@cs.put.poznan.pl
[4] Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland

Abstract. This paper proposes the Necessary-preference-enhanced Evolutionary Multiobjective Optimizer (NEMO), a combination of an evolutionary multiobjective optimization method, NSGA-II, and an interactive multiobjective optimization method, GRIP. In the course of NEMO, the decision maker is able to introduce preference information in a holistic way, by simply comparing some pairs of solutions and specifying which solution is preferred, or comparing intensities of preferences between pairs of solutions. From this information, the set of all compatible value functions is derived using GRIP, and a properly modified version of NSGA-II is then used to search for a representative set of all Pareto-optimal solutions compatible with this set of derived value functions. As we show, this allows to focus the search on the region most preferred by the decision maker, and thereby speeds up convergence.

1 Introduction

Most of past research on evolutionary multiobjective optimization (EMO) attempts to approximate the complete Pareto-optimal front by a set of well-distributed representatives of Pareto-optimal solutions. The underlying reasoning is that in the absence of any preference information, all Pareto-optimal solutions have to be considered equivalent.

On the other hand, in most practical applications, the decision maker (DM) is eventually interested in only a small subset of good solutions, or even a single most preferred solution. In order to come up with such a result, it is necessary to involve the DM. This is the underlying idea of another multiobjective optimization paradigm: interactive multiobjective optimization (IMO). IMO deals with the identification of the most preferred solution by means of a systematic dialogue with the DM. Only recently, the scientific community has discovered the great potential of combining the two paradigms (for a recent survey, see

M. Ehrgott et al. (Eds.): EMO 2009, LNCS 5467, pp. 554–568, 2009.

[14]). From the point of view of EMO, involving the DM in an interactive procedure will allow to focus the search on the area of the Pareto front which is most relevant to the DM. This, in turn, may allow to find preferred solutions faster. In particular, in the case of many objectives, EMO has difficulties, because the number of Pareto-optimal solutions becomes huge, and Pareto-optimality is not sufficiently discriminative to guide the search into better regions. Integrating user's preferences promises to alleviate these problems, allowing to converge faster to the preferred region of the Pareto-optimal front.

This paper combines NSGA-II [3], a widely used EMO technique, with an IMO methodology from multiple criteria decision aiding (MCDA), originally conceived to deal with a limited number of alternatives. This methodology relies on the Robust Ordinal Regression approach recently implemented in the two methods, UTA^{GMS} [10] and GRIP [7]. In these methods, the user is presented with a small set of alternatives and can state his/her preferences by specifying a holistic preference of one alternative over another, or comparing intensities of preferences between pairs of alternatives. The user can also compare intensities of preferences with respect to single criteria. Robust ordinal regression then identifies the whole set of additive value functions (also called utility functions) compatible with the preference information given by the user. This permits to compare any pair of alternatives x and y in a simple and intuitive way, as follows:

- x is necessarily at least as good as y, if this is true for all compatible value functions,
- x is possibly at least as good as y, if this is true for at least one compatible value function.

The interactive EMO method we are proposing, called NEMO (Necessary-preference-enhanced Evolutionary Multiobjective Optimization), takes the information about necessary preferences into account during optimization, focusing search on the most promising parts of the Pareto-optimal front. More specifically, robust ordinal regression based on information obtained through interaction with the user determines the set of all compatible value functions, and an EMO procedure searches for all non-dominated solutions with respect to all compatible value functions in parallel. In the context of EMO, the alternatives considered in GRIP are solutions of a current population.

We believe that the integration of GRIP into EMO is particularly promising for two reasons:

1. The preference information required by GRIP is very basic and easy to provide by the DM. All that is asked for is to compare two solutions, and to reveal whether one is preferred over the other. Additionally, the DM can compare the intensity of preference between pairs of solutions.
2. The resulting set of compatible value functions implicitly reveals also an appropriate scaling of the criteria, an issue that is largely ignored by the EMO community so far.

The paper is organized as follows. The next section provides a brief overview of existing EMO/IMO hybrids. Section 3 describes the basic concepts of robust ordinal regression, presenting UTA^{GMS} and GRIP. Then, Section 4 presents the basic ideas of our method, NEMO. Preliminary empirical results are reported in Section 5. The paper concludes with a summary and some ideas for future research.

2 Interactive Evolutionary Multiobjective Optimization

There are various ways in which user preferences can be incorporated into EMO. Furthermore, there are many IMO techniques, and most of them are suitable for combination with EMO.

A form of preference information often used is a reference point, and various ways to guide the search towards a user-specified reference point have been proposed. Perhaps the earliest such approach has been presented in [8], which gives a higher priority to objectives in which the goal is not fulfilled. [5] suggests to use the distance from the reference point as a secondary criterion following the Pareto ranking. [22] uses an indicator-based evolutionary algorithm, and an achievement scalarizing function to modify the indicator and force the algorithm to focus on the more interesting part of the Pareto front.

In the guided MOEA proposed in [2], the user is allowed to specify preferences in the form of maximally acceptable trade-offs like "one unit improvement in objective i is worth at most a_{ji} units in objective j". The basic idea is to modify the dominance criterion accordingly, so that it reflects the specified maximally acceptable trade-offs.

[4] proposes an interactive decision support system called I-MODE that implements an interactive procedure built over a number of existing EMO and classical decision making methods. The main idea of the interactive procedure is to allow the DM to interactively focus on interesting region(s) of the Pareto front. The DM has options to use several tools for generation of potentially Pareto-optimal solutions concentrated in the desired regions. For example, he/she may use weighted sum approach, utility function based approach, Tchebycheff function approach or trade-off information. The preference information is then used by an EMO to generate new solutions in the most interesting regions.

There are several additional papers that integrate EMO and IMO, but due to space constraints, we refer the interested reader to two recent reviews [14,1]. Instead, in the following, we shall restrict our attention to three papers that perhaps come closest to what we propose in this paper, namely [11], [19], and [13].

[11] suggests a procedure which asks the user to rank a few alternatives, and from this derives constraints for linear weighting of the objectives consistent with the given ordering. Then, these are used within an EMO to check whether there is a feasible linear weighting such that solution x is preferable to solution y. If this is not the case, it is clear that y is preferred to x. The approach differs from ours in two important aspects: first, the interaction with the user is only prior to EMO, while our approach interacts with the user during optimization. Second, the utility function model is only a linear weighting of the objectives, while we consider general additive value functions.

The interactive evolutionary algorithm proposed by [19] allows the user to provide preference information about pairs of solutions during the run. Based on this information, the authors compute the “most compatible” weighted sum of objectives (i.e., a linear achievement scalarizing function) by means of linear programming, and use this as single substitute objective for some generations of the evolutionary algorithm. This concept presented in this paper is truly interactive, as preference information is collected during the run. However, as it reduces the preference information to a single linear weighting of the objectives, the power of EMO, which is capable of simultaneously searching for multiple solutions with different trade-offs, is not exploited. Furthermore, since only partial preference information is available, there is no guarantee that the weight vector obtained by solving the linear programming model defines the DM’s value function, even if the value function has the form of a weighted sum (naturally, the bias may become even more significant when the DM’s preferences cannot be modeled with a linear function).

The method of [13] is based on the Pareto memetic algorithm (PMA). The original PMA samples the set of scalarizing functions drawing a random weight vector for each single iteration and using this for selection and local search. In the proposed interactive version, preference information from pairwise comparisons of solutions is used to reduce the set of possible weight vectors. While this approach is more flexible in terms of the considered value function model, and changes the value function from generation to generation, it still does not make explicit use of the EMO’s capability to search for multiple solutions in parallel.

Furthermore, all of the methods discussed above require a pre-defined scaling of the objectives, while we propose a new way that allows to automatically and continuously adjust the scaling of the objectives to the most likely user preferences given the information gathered so far.

3 Robust Ordinal Regression

In MCDA, the *preference information* may be either direct or indirect, depending whether it specifies directly values of some parameters used in the *preference model* (e.g., trade-off weights, aspiration levels, discrimination thresholds, etc.) or whether it specifies some examples of holistic judgments from which compatible values of the preference model parameters are induced. Eliciting direct preference information from the DM can be counterproductive in real-world decision making situations because of a high cognitive effort required. Eliciting indirect preferences is less demanding in terms of cognitive effort. Indirect preference information is mainly used in the ordinal regression paradigm. According to this paradigm, a holistic preference information on a subset of some reference or training solutions is known first and then a preference model compatible with the information is built and applied to the whole set of solutions in order to rank them.

The *ordinal regression* paradigm emphasizes the discovery of intentions as an interpretation of actions rather than as a priori position, which was called by

March the posterior rationality [16]. It has been known for at least fifty years in the field of multidimensional analysis. It is also concordant with the induction principle used in machine learning. This paradigm has been applied within the two main MCDA approaches: those using a value function as preference model [21,18,12,20], and those using an outranking relation as preference model [15,17]. This paradigm has also been used since mid nineties' in MCDA methods involving a new, third family of preference models - a set of dominance decision rules induced from rough approximations of holistic preference relations [9].

Recently, the ordinal regression paradigm has been revisited with the aim of considering the whole set of value functions compatible with the preference information provided by the DM, instead of a single compatible value function used in UTA-like methods [12,20]. This extension, called *robust ordinal regression*, has been implemented in a method called UTA^{GMS} [10], and further generalized in another method called GRIP [7]. UTA^{GMS} and GRIP are not revealing to the DM one compatible value function, but they are using the whole set of compatible (general, not piecewise-linear only) additive value functions to set up a necessary weak preference relation and a possible weak preference relation in the whole set of considered solutions.

3.1 Concepts: Definitions and Notation

We are considering a multiple criteria decision problem where a finite set of solutions $A = \{x, \ldots, y, \ldots w, \ldots\}$ is evaluated on a family $F = \{g_1, g_2, \ldots, g_n\}$ of n criteria. Let $I = \{1, 2, \ldots, n\}$ denote the set of criteria indices. We assume, without loss of generality, that the smaller $g_i(x)$, the better solution x on criterion g_i, for all $i \in I$, $x \in A$. A DM is willing to rank the solutions of A from the best to the worst, according to his/her preferences. The ranking can be complete or partial, depending on the preference information provided by the DM and on the way of exploiting this information.

Such a decision-making problem statement is called *multiple criteria ranking problem*. It is known that the only information coming out from the formulation of this problem is the dominance ranking. For any pair of solutions $x, y \in A$, one of the four situations may arise in the dominance ranking: x is preferred to y (x dominates y but y does not dominate x), y is preferred to x (y dominates x but x does not dominate y), x is indifferent to y (x and y dominate each other), or x is incomparable to y (neither x dominates y nor y dominates x). Usually, the dominance ranking is very poor, i.e., the most frequent situation is x incomparable to y, in particular if the number of objectives is high.

In order to enrich the dominance ranking, the DM has to provide preference information which is used to construct an aggregation model making the solutions more comparable. Such an aggregation model is called preference model. It induces a preference structure on set A, whose proper exploitation permits to work out a ranking proposed to the DM.

In what follows, the evaluation of each solution $x \in A$ on each criterion $g_i \in F$ will be denoted by $g_i(x)$.

Let G_i denote the value set (scale) of criterion g_i, $i \in I$. Consequently,

$$G = G_1 \times G_2 \times \ldots \times G_n$$

represents the evaluation space. From a pragmatic point of view, it is reasonable to assume that $G_i \subseteq \mathbb{R}$, for $i = 1, \ldots, m$. More specifically, we will assume that the value space on each criterion g_i is bounded, such that $G_i = [\alpha_i, \beta_i]$, where α_i, β_i, $\alpha_i < \beta_i$ are the worst and the best (finite) evaluations, respectively. Thus, $g_i \ : \ A \rightarrow G_i$, $i \in I$. Therefore, each solution $x \in A$ is associated with an evaluation solution denoted by $\underline{g}(x) = (g_1(x), g_2(x), \ldots, g_n(x)) \in G$. For notational simplicity, we will also write x_i instead of $g_i(x)$, so $\underline{g}(x) = (x_1, x_2, \ldots, x_n) \in G$.

We consider a weak preference relation $\succeq$ on A which means, for each pair of solutions $x, y \in A$,

$$x \succeq y \Leftrightarrow \text{“}x \text{ is at least as good as } y\text{”}.$$

This weak preference relation can be decomposed into its asymmetric and symmetric parts, as follows,

1) $x \succ y \equiv [x \succeq y \text{ and } not(y \succeq x)] \Leftrightarrow$ “x is preferred to y”, and
2) $x \sim y \equiv [x \succeq y \text{ and } y \succeq x] \Leftrightarrow$ “x is indifferent to y”.

3.2 The Ordinal Regression Method for Learning the Whole Set of Compatible Value Functions

The additive value function considered in ordinal regression is defined on A such that for each $\underline{g}(x) \in G$,

$$U(\underline{g}(x)) = \sum_{i=1}^{n} u_i(g_i(x_i)), \tag{1}$$

where, u_i are non-increasing marginal value functions, $u_i : G_i \rightarrow \mathbb{R}$, $i \in I$. For the sake of simplicity, we shall write (1) as follows,

$$U(x) = \sum_{i=1}^{n} u_i(x). \tag{2}$$

Recently, two new methods, UTA$^{\text{GMS}}$ [10] and GRIP [7], have generalized the classical ordinal regression approach of the UTA method [12] in several aspects:

– taking into account all additive value functions (1) compatible with the preference information, while UTA is using only one such function,
– considering marginal value functions of (1) as general non-decreasing functions, and not piecewise-linear, as in UTA,
– asking the DM for a ranking of reference solutions which is not necessarily complete (just pairwise comparisons),

- taking into account additional preference information about intensity of preference, expressed both comprehensively and with respect to a single criterion,
- avoiding the use of the exogenous, and not neutral for the result, parameter ε in the modeling of strict preference between solutions.

UTA$^{\text{GMS}}$ produces two rankings on the set of solutions A, such that for any pair of solutions $a, b \in A$:

- in the *necessary* ranking, a is ranked at least as good as b if and only if, $U(a) \geq U(b)$ for *all* value functions compatible with the preference information,
- in the *possible* ranking, a is ranked at least as good as b if and only if, $U(a) \geq U(b)$ for *at least one* value function compatible with the preference information.

GRIP produces four more necessary and possible rankings on the set of solutions A, and on the set of pairs of solutions, $A \times A$.

The necessary ranking can be considered as robust with respect to the preference information. Such robustness of the necessary ranking refers to the fact that any pair of solutions is ranked in the same way whatever the additive value function compatible with the preference information. Indeed, when no preference information is given, the necessary ranking boils down to the weak dominance relation (i.e., a is necessarily at least as good as b, if $g_i(a) \leq g_i(b)$ for all $g_i \in F$), and the possible ranking is a complete relation. Every new pairwise comparison of reference solutions, for which the dominance relation does not hold, is enriching the necessary ranking and it is impoverishing the possible ranking, so that they converge with the growth of the preference information.

Moreover, such an approach has another feature which is very appealing in the context of MOO. It stems from the fact that it gives space for interactivity with the DM. Presentation of the necessary ranking, resulting from a preference information provided by the DM, is a good support for generating reactions from the DM. Namely, he/she could wish to enrich the ranking or to contradict a part of it. Such a reaction can be integrated in the preference information considered in the next calculation stage.

The idea of considering the whole set of compatible value functions was originally introduced in UTA$^{\text{GMS}}$. GRIP (Generalized Regression with Intensities of Preference) can be seen as an extension of UTA$^{\text{GMS}}$ permitting to take into account additional preference information in form of comparisons of intensities of preference between some pairs of reference solutions. For solutions $x, y, w, z \in A$, these comparisons are expressed in two possible ways (not exclusive): (i) comprehensively, on all criteria, like "x is preferred to y at least as much as w is preferred to z"; and, (ii) partially, on each criterion, like "x is preferred to y at least as much as w is preferred to z, on criterion $g_i \in F$ ". In the following, we shall use GRIP.

3.3 The Preference Information Provided by the Decision Maker

The DM is expected to provide the following preference information in the dialogue stage of the procedure:

- A partial preorder $\succeq$ on A^R whose meaning is: for some $x, y \in A^R$

  $$x \succeq y \Leftrightarrow \text{ "}x\text{ is at least as good as }y\text{"}.$$

  Moreover, $\succ$ (preference) is the asymmetric part of $\succeq$, and $\sim$ (indifference) is its symmetric part.
- A partial preorder $\succeq^*$ on $A^R \times A^R$, whose meaning is: for some $x, y, w, z \in A^R$,

  $$(x, y)\succeq^*(w, z)\Leftrightarrow\text{"}x\text{ is preferred to }y\text{ at least as much as }w\text{ is preferred to }z\text{"}.$$

  Also in this case, $\succ^*$ is the asymmetric part of $\succeq^*$, and $\sim^*$ is its symmetric part.
- A partial preorder $\succeq_i^*$ on $A^R \times A^R$, whose meaning is: for some $x, y, w, z \in A^R$, $(x, y) \succeq_i^* (w, z) \Leftrightarrow$ "x is preferred to y at least as much as w is preferred to z" on criterion g_i, $i \in I$.

In the following, we also consider the weak preference relation $\succeq_i$ being a complete preorder whose meaning is: for all $x, y \in A$,

$$x \succeq_i y \quad \Leftrightarrow \quad \text{"}x\text{ is at least as good as }y\text{" on criterion } g_i, \quad i \in I.$$

Weak preference relations $\succeq_i$, $i \in I$, are not provided by the DM, but they are obtained directly from the evaluation of solutions x and y on criteria g_i, i.e., $x \succeq_i y \Leftrightarrow g_i(x) \leq g_i(y)$, $i \in I$.

3.4 Linear Programming Constraints

In this subsection, we present a set of constraints that interprets the preference information in terms of conditions on the compatible value functions.

To be compatible with the provided preference information, the value function $U : A \to [0, 1]$ should satisfy the following constraints corresponding to the DM's preference information:

a) $U(w) > U(z)$ if $w \succ z$
b) $U(w) = U(z)$ if $w \sim z$
c) $U(w) - U(z) > U(x) - U(y)$ if $(w, z) \succ^* (x, y)$
d) $U(w) - U(z) = U(x) - U(y)$ if $(w, z) \sim^* (x, y)$
e) $u_i(w) \geq u_i(z)$ if $w \succeq_i z$, $i \in I$
f) $u_i(w) - u_i(z) > u_i(x) - u_i(y)$ if $(w, z) \succ_i^* (x, y)$, $i \in I$
g) $u_i(w) - u_i(z) = u_i(x) - u_i(y)$ if $(w, z) \sim_i^* (x, y)$, $i \in I$

Moreover, the following normalization constraints should also be taken into account:

h) $u_i(x_i^*) = 0$, where x_i^* is such that $x_i^* = \max\{g_i(x) \ : \ x \in A\}$;
i) $\sum_{i \in I} u_i(y_i^*) = 1$, where y_i^* is such that $y_i^* = \min\{g_i(x) \ : \ x \in A\}$.

For computational details, the reader is referred to [7].

3.5 The Most Representative Value Function

The robust ordinal regression builds a set of additive value functions compatible with preference information provided by the DM and results in two rankings, necessary and possible. Such rankings answer to robustness concerns, since they provide, in general, "more robust" conclusions than a ranking made by an arbitrarily chosen compatible value function. However, in some decision-making situations, it may be desirable to give a score to different solutions, and despite the interest of the rankings provided, some users would like to see, and they indeed need, to know the "most representative" value function among all the compatible ones This allows assigning a score to each solution. Recently, a methodology to identify the "most representative" function in GRIP without losing the advantage of taking into account all compatible value functions has been proposed in [6]. The idea is to select among all compatible value functions the most discriminant value function for consecutive solutions in the necessary ranking, i.e., that value function which maximizes the difference of scores between solutions related by preference in the necessary ranking. To break ties, one can wish to minimize the difference of scores between solutions not related by preference in the necessary ranking. This can be achieved using the following procedure:

1. Determine the necessary preference relations in the considered set of solutions.
2. For all pairs of solutions (a, b), such that a is necessarily preferred to b, add the following constraints to the linear programming constraints of GRIP: $U(a) \geq U(b) + \varepsilon$.
3. Maximize the objective function ε.
4. Add the constraint $\varepsilon = \varepsilon^*$, with ε^* being the resulting maximal ε from point 3), to the linear programming constraints of point 2).
5. For all pairs of solutions (a, b), such that neither a is necessarily preferred to b nor b is necessarily preferred to a, add the following constraints to the linear programming constraints of GRIP and to the constraints considered in above point 4): $U(a) - U(b) \leq \delta$ and $U(b) - U(a) \leq \delta$.
6. Minimize δ.

This procedure maximizes the minimal difference between values of solutions for which the necessary preference holds. If there is more than one such value function, the above procedure selects the most representative compatible value function giving the largest minimal difference between values of solutions for which the necessary preference holds, and the smallest maximal difference between values of solutions for which the possible preference holds.

Notice that the concept of the "most representative" value function thus defined is still based on the necessary and possible preference relations, which remain crucial for GRIP. In a sense, it gives the most faithful representation of these necessary and possible preference relations. Notice also that the above procedure can be simplified by joint maximization of $M\varepsilon - \delta$ where M is a "big value".

In the following, we will use the most representative value function for continuously adapting the scaling of the objectives in a non-linear way.

4 Necessary-Preference-Enhanced Evolutionary Multiobjective Optimization – NEMO

Our main idea is to integrate the concept of GRIP into an EMO approach, in particular NSGA-II [3]. NSGA-II is one of today's most prominent and most successful EMO algorithms. It ranks individuals according to two criteria.

The primary criterion is the so-called dominance-based ranking. This method ranks individuals by iteratively determining the non-dominated solutions in the population (non-dominated front), assigning those individuals the next best rank and removing them from the population. The result is a partial ordering, favoring individuals closer to the Pareto-optimal front.

As secondary criterion, individuals which have the same dominance-rank (primary criterion) are sorted according to crowding distance, which is defined as the sum of distances between a solution's neighbors on either side in each dimension of the objective space. Individuals with a large crowding distance are preferred, as they are in a less crowded region of the objective space, and favoring them aims at preserving diversity in the population.

In our approach, we will

1. Replace the dominance-based ranking by the *necessary* ranking. The necessary ranking is calculated analogously to the dominance-based ranking, but taking into account the preference information by the user through the necessary preference relations. More specifically, first put in the best rank those solutions which have no competitor which would be necessarily preferred, remove them from the population, etc.
2. Replace the crowding-distance by a distance calculated taking into account the multidimensional scaling given by the "most representative value function" among the whole set of compatible value functions (see sub-section 3.5). While in NSGA-II the crowding distance is calculated in the space of objective functions, in NEMO it is calculated in the space of marginal value functions which are components of the "most representative" value function. Given a solution x, its crowding distance is calculated according to the following formula:

$$\text{Crowding distance}(x) = \sum_{i=1}^{n} \left|u_i(y^i) - u_i(z^i)\right| - \left|\sum_{i=1}^{n} \left[U(y^i) - U(z^i)\right]\right|,$$

   where U is the "most representative value function", u_i are its marginal value functions, and y^i and z^i are left and right neighbors of x in dimension of marginal value u_i. Remark that for a given n, we can have up to $2n$ different neighbors of x in all dimensions, due to non-univocal selection of solutions with equal marginal values. In fact, we select the neighbors such as to diversify them as much as possible.

Algorithm 1. Basic NEMO

Generate initial solutions randomly
Elicit user preferences {Present to user a pair of solutions and ask for a preference information}
Determine *necessary* ranking {Will replace dominance ranking in NSGA-II}
Determine *secondary* ranking {Order solutions within a front, based on crowding distance measured in terms of the "most representative value function"}
repeat
  Mating selection and offspring generation
  if Time to ask DM **then**
    Elicit user preferences
  end if
  Determine *necessary* ranking
  Determine *secondary* ranking
  Environmental selection
until Stopping criterion met
Return all preferred solutions according to necessary ranking

Preferences are elicited by asking the DM to compare some pairs of solutions, and specify a preference relation between them. This is done during the run of the NSGA-II.

The overall algorithm is outlined in Algorithm 1. Although the general procedure is rather straightforward, there are several issues that need to be considered:

1. How many pairs of solutions are shown to the DM, and when? Here, we decide to ask for one preference relation every k generations, i.e., every x generations, NSGA-II is stopped, and the user is asked to provide preference information about one given pair of individuals.
2. Which pairs of solutions shall be shown to the DM for comparison? Here, we randomly pick a small set of non-dominated solutions (according to the necessary ranking). This also prevents the user from specifying inconsistent information.

5 Experimental Results

An empirical evaluation of interactive EMO methods is difficult, because the test environment has to include a model of the user behavior. For testing, we use the simple 30-dimensional ZDT1 test function. We assume that our artificial user makes decisions with respect to a simple predefined value function $U(x) = -(0.6f_1(x) + 0.4f_2(x))$. This function is unknown to NEMO, but is used to simulate user's comparisons of solutions when preferences are elicited. In every k-th generation, NEMO randomly selects two individuals from the non-dominated solutions according to the necessary ranking, and receives as feedback the solution preferred by the DM according to the predefined value function. In particular, only pairwise comparisons of solutions are considered here, while intensities of preferences between pairs of solutions are not (yet) considered. The population size has been set to 32.

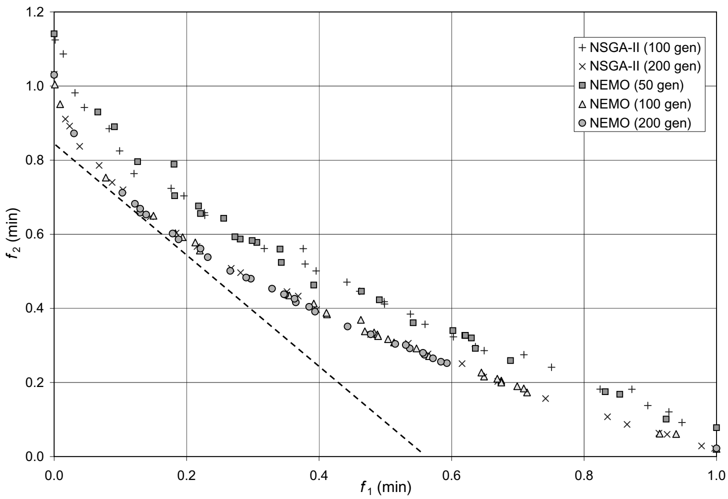

Fig. 1. Results of NEMO and NSGA-II on ZDT1 after 50, 100 and 200 generations, with preference elicitation every 20 generations. The dashed line indicates the artificial user's value function.

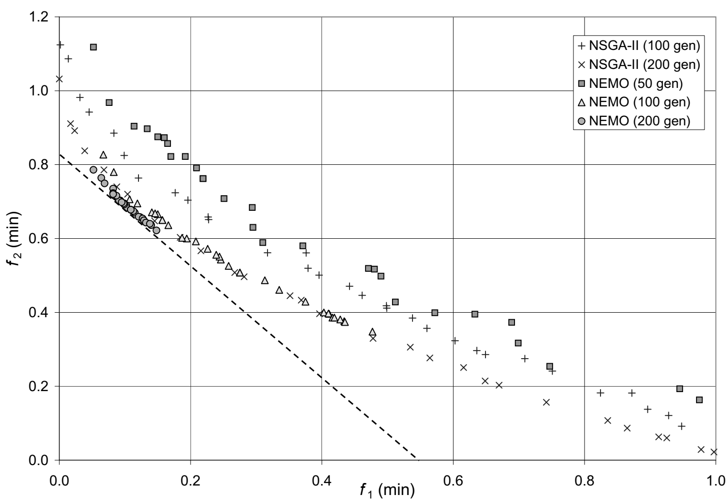

Fig. 2. Results of NEMO and NSGA-II on ZDT1 after 50, 100 and 200 generations, with preference elicitation in every generation. The dashed line indicates the artificial user's value function.

Figure 1 shows results of NEMO and NSGA-II after 50, 100 and 200 generations, when the preference information concerning one pairwise comparison is gathered every 20 generations. As can be seen, NEMO converges faster than

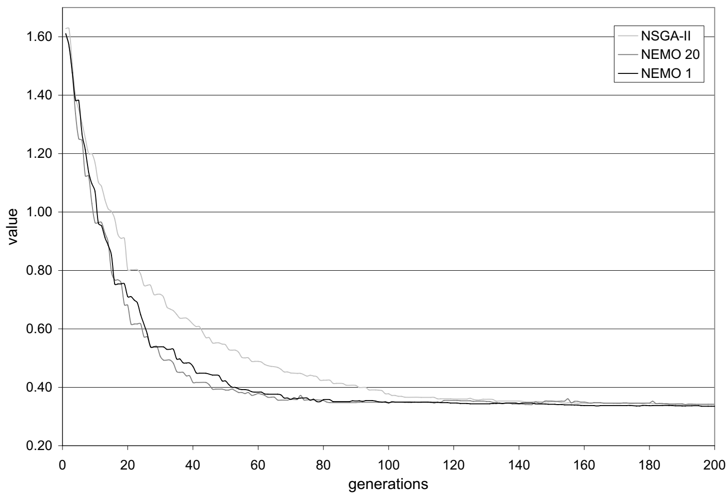

Fig. 3. Value of the most preferred solutions in successive generations of NEMO and NSGA-II

NSGA-II. After 50 generations, the solutions obtained by NEMO are as good as the solutions obtained by NSGA-II after 100 generations. Moreover, in the course of generations, the population of solutions obtained by NEMO is narrowed to a smaller part of the Pareto front than the population of solutions obtained by NSGA-II. This is because NEMO concentrates on on the user-preferred solutions on the Pareto front, while NSGA-II attempts to approximate the whole front.

The tendency observed in Figure 1 is reinforced when the preferences are gathered more often. Figure 2 shows results of NEMO and NSGA-II after 50, 100 and 200 generations, when the preference information concerning one pairwise comparison is gathered in every generation. After 100 generations NEMO is reaching equally good solutions as NSGA-II after 200 generations. Moreover, due to the richer preference information than in the previous case, the solutions obtained by NEMO are focused on a smaller part of the Pareto front.

Figure 3 shows the evolution of the value of the artificial user's value function for the most preferred solution in successive generations. It permits to observe the convergence speed of NEMO and NSGA-II. "NEMO 20" corresponds to the case presented in Figure 1, and "NEMO 1" to the case presented in Figure 2.

6 Conclusion

We presented an interactive EMO method called NEMO. It combines the advantages of the well known EMO method NSGA-II with an MCDA method GRIP enabling the user interaction based on robust ordinal regression. The main advantages of the proposed methodology are the following:

1. It models the user's preferences in terms of very general value functions,
2. It requires a preference information expressed in a simple and intuitive way (comparisons of solutions or comparisons of intensities of preferences),
3. It considers all value functions compatible with the user's preferences, with the goal to generate a representative approximation of all Pareto-optimal solutions compatible with any of these value functions,
4. With respect to crowding distance, it permits to calculate distances in utility space, rather than objective space, thereby alleviating the need of scaling the objectives.

Preliminary empirical results show that the proposed NEMO method works as expected and is able to converge faster to the user-preferred solutions than NSGA-II without taking user preferences into account.

Clearly, a more thorough empirical analysis on a variety of test functions and value functions is necessary. Also, we are currently elaborating and extending the approach in various directions. In particular, we are implementing improved interaction mechanisms, with adaptive methods to determine when a DM should be asked for preference information, and what individuals to present for comparison. We will also extend the current interaction to allow additional preference information to be incorporated. Apart from the above mentioned intensities of preferences, we plan to integrate into GRIP maximum/minimum trade-off information, e.g., one unit improvement in objective f_1 is worth at most w units worsening in objective f_2.

Finally, we plan to elaborate a slightly different approach: instead of calculating the necessary preference relation in the population of solutions, we could look for solutions that are the best for at least one compatible value function. The expected advantages of this new approach are speeding up of calculations and of the convergence to the most interesting part of the Pareto front.

References

1. Branke, J.: Consideration of partial user preferences in evolutionary multiobjective optimization. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization - Interactive and Evolutionary Approaches. LNCS, vol. 5252, pp. 157–178. Springer, Heidelberg (2008)
2. Branke, J., Kaußler, T., Schmeck, H.: Guidance in evolutionary multi-objective optimization. Advances in Engineering Software 32, 499–507 (2001)
3. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
4. Deb, K., Chaudhuri, S.: I-MODE: An interactive multi-objective optimization and decision-making using evolutionary methods. Technical Report KanGAL Report No. 2007003, Indian Institute of Technology Kanpur (2007)
5. Deb, K., Sundar, J., Udaya Bhaskara Rao, N., Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithms. International Journal of Computational Intelligence Research 2(3), 273–286 (2006)

6. Figueira, J., Greco, S., Słowiński, R.: Identifying the "most representative" value function among all compatible value functions in the GRIP method. In: 68th Meeting of the EURO Working Group on MCDA, Chania, October 2-3 (2008)
7. Figueira, J., Greco, S., Słowiński, R.: Building a set of additive value functions representing a reference preorder and intensities of preference: GRIP method. European Journal of Operational Research 195(2), 460–486 (2009)
8. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: Formulation, discussion, and generalization. In: Proceedings of the Fifth International Conference on Genetic Algorithms, pp. 416–423 (1993)
9. Greco, S., Matarazzo, B., Słowiński, R.: Rough sets theory for multicriteria decision analysis. European Journal of Operational Research 129(1), 1–47 (2001)
10. Greco, S., Mousseau, V., Słowiński, R.: Ordinal regression revisited: Multiple criteria ranking with a set of additive value functions. European Journal of Operational Research 191(2), 415–435 (2008)
11. Greenwood, G.W., Hu, X.S., D'Ambrosio, J.G.: Fitness functions for multiple objective optimization problems: combining preferences with Pareto rankings. In: Belew, R.K., Vose, M.D. (eds.) Foundations of Genetic Algorithms, pp. 437–455. Morgan Kaufmann, San Francisco (1997)
12. Jacquet-Lagrèze, E., Siskos, Y.: Assessing a set of additive utility functions for multicriteria decision making: The UTA method. European Journal of Operational Research 10, 151–164 (1982)
13. Jaszkiewicz, A.: Interactive multiobjective optimization with the Pareto memetic algorithm. Foundations of Computing and Decision Sciences 32(1), 15–32 (2007)
14. Jaszkiewicz, A., Branke, J.: Interactive multiobjective evolutionary algorithms. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization - Interactive and Evolutionary Approaches. LNCS, vol. 5252, pp. 179–193. Springer, Heidelberg (2008)
15. Kiss, L., Martel, J.M., Nadeau, R.: ELECCALC - an interactive software for modelling the decision maker's preferences. Decision Support Systems 12, 757–777 (1994)
16. March, J.: Bounded rationality, ambiguity and the engineering of choice. Bell Journal of Economics 9, 587–608 (1978)
17. Mousseau, V., Słowiński, R.: Inferring an ELECTRE TRI model from assignment examples. Journal of Global Optimization 12, 157–174 (1998)
18. Pekelman, D., Sen, S.K.: Mathematical programming models for the determination of attribute weights. Management Science 20, 1217–1229 (1974)
19. Phelps, S., Köksalan, M.: An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. Management Science 49(12), 1726–1738 (2003)
20. Siskos, Y., Grigoroudis, V., Matsatsinis, N.: UTA methods. In: Figueira, F., Greco, S., Ehrgott, M. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, pp. 297–343. Springer, Heidelberg (2005)
21. Srinivasan, V., Shocker, A.D.: Estimating the weights for multiple attributes in a composite criterion using pairwise judgments. Psychometrika 38, 473–493 (1973)
22. Thiele, L., Miettinen, K., Korhonen, P.J., Molina, J.: A preference-based interactive evolutionary algorithm for multiobjective optimization. Working papers w-412, Helsinki School of Economics, Helsinki (2007)