BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 58, No. 3, 2010
DOI: 10.2478/v10175-010-0033-3

Interactive evolutionary multiobjective optimization driven by robust ordinal regression

J. BRANKE[1], S. GRECO[2], R. SŁOWIŃSKI [3,4]*, and P. ZIELNIEWICZ[3]

[1] Warwick Business School, University of Warwick, UK
[2] Faculty of Economics, University of Catania, Italy
[3] Institute of Computing Science, Poznań University of Technology, Poland
[4] Systems Research Institute, Polish Academy of Sciences, Poland

Abstract. This paper presents the Necessary-preference-enhanced Evolutionary Multiobjective Optimizer (NEMO), which combines an evolutionary multiobjective optimization with robust ordinal regression within an interactive procedure. In the course of NEMO, the decision maker is asked to express preferences by simply comparing some pairs of solutions in the current population. The whole set of additive value functions compatible with this preference information is used within a properly modified version of the evolutionary multiobjective optimization technique NSGA-II in order to focus the search towards solutions satisfying the preferences of the decision maker. This allows to speed up convergence to the most preferred region of the Pareto-front.

Key words: evolutionary multiobjective optimization, interactive procedure, robust ordinal regression.

1. Introduction

Real life decision problems usually involve consideration of multiple conflicting objectives. For example, resource-constrained project scheduling involves multiple objectives of the type: project duration, net present value, resource consumption, and so on. As, in general, there does not exist a single solution which optimizes simultaneously all objectives, one has to search for Pareto-optimal solutions. A solution is Pareto-optimal (also called efficient or non-dominated) if there is no other feasible solution which would be at least as good on all objectives while being strictly better on at least one objective. Finding the whole set of Pareto-optimal solutions (also called Pareto-set or Pareto-front) is usually computationally hard. This is why a lot of research has been devoted to heuristic search of an approximation of the Pareto-front. Among the heuristics proposed to this end, Evolutionary Multiobjective Optimization (EMO) procedures appeared to be particularly efficient (see, e.g., [1, 2]).

The underlying reasoning behind the EMO search of an approximation of the Pareto-front is that in the absence of any preference information, all Pareto-optimal solutions have to be considered equivalent.

On the other hand, if the user or decision maker (DM) (alternatively called user) is involved in the multiobjective optimization process, then the preference information provided by the DM can be used to focus the search on the most preferred part of the Pareto-front. This idea stands behind Interactive Multiobjective Optimization (IMO) methods proposed a long time before EMO has emerged (see, e.g., [3–5]).

Recently, it became clear that merging the IMO and EMO methodologies should be beneficial for the multiobjective optimization process [6]. This paper goes in this direction, by proposing the Necessary-preference-enhanced Evolutionary Multiobjective Optimizer (NEMO), which combines an evolutionary multiobjective optimization with an interactive procedure based on so-called Robust Ordinal Regression (ROR) [7].

In ROR, which has been recently implemented in two multiple criteria ranking methods, UTA$^{\mathrm{GMS}}$ [8] and GRIP [9], the DM 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. ROR then identifies the whole set of additive value functions (also called utility functions) compatible with the preference information given by the DM. 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.

NEMO combines NSGA-II [10], 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. The first idea of this interactive method has been presented in [11]. NEMO takes the information about necessary preferences into account during optimization, focusing the search on the most promising parts of the Pareto-front. More specifically, ROR 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

*e-mail: roman.slowinski@cs.put.poznan.pl

all compatible value functions simultaneously. In the context of EMO, the alternatives considered in ROR are solutions of a current population.

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

1. The preference information required by ROR is very basic and easy to provide by the DM. All that the DM is asked for is to compare two non-dominated solutions from a current population, and to reveal whether one is preferred over the other. The preference information is provided with respect to one or several pairs of solutions, every k iterations (k depends on the problem and the willingness of the user to interact with the system. In our studies, k ranges from 10 to 30).
2. The resulting set of compatible value functions implies an appropriate scaling of the objectives, thus, even if the objectives are expressed on very heterogeneous scales, the solutions are automatically scaled to reflect the user's preferences. This issue has been 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 ROR methodology. Then, Sec. 4 presents the basic steps of NEMO. Some empirical results are reported in Sec. 5. The paper concludes with a summary and some ideas for future research.

2. Interactive Evolutionary Multiobjective Optimization

There are various ways in which the user's 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 [12], which gives a higher priority to objectives in which the goal is not fulfilled. [13] suggests to use the distance from the reference point as a secondary criterion following the Pareto ranking. [14] uses an indicator-based evolutionary algorithm, and an achievement scalarizing function to modify the indicator and force the algorithm to focus on the most interesting part of the Pareto-front.

In the guided MOEA proposed in [15], 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.

Deb and Chaudhuri [16] propose 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, Chebycheff 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 – we refer the interested reader to two recent reviews [17, 18]. In the following, we shall restrict our attention to three papers that perhaps come closest to what we propose in this paper, namely [19–21].

Greenwood et al. [19] suggest 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 (weighted sum) of the objectives, while we consider general additive value functions.

The interactive evolutionary algorithm proposed by Phelps and Köksalan [20] 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. The 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 DM's 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 value function).

The method of Jaszkiewicz [21] is based on the Pareto memetic algorithm (PMA). It has been designed for multiobjective combinatorial optimization problems. 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 user's most likely preferences given the preference information gathered so far.

3. Robust Ordinal Regression

3.1. Definitions and notation. To explain ROR, we consider a Multiple Criteria Decision Aiding (MCDA) problem concerning a finite set of alternatives $A = \{x, y, \ldots\}$ ($|A| = m$), evaluated on n criteria (called objectives in optimization) from family $F = \{g_1, \ldots, g_n\}$, with $g_i \colon A \rightarrow \mathbb{R}, i = 1, \ldots, n$. Let $I = \{1, \ldots, n\}$ denote the set of criteria indices, and assume, without loss of generality, that the greater $g_i(x)$, the better alternative x on criterion g_i, for all $i \in I$, $x \in A$. The family of criteria F is supposed to satisfy consistency conditions, i.e., completeness (all relevant criteria are considered), monotonicity (the better the evaluation of an alternative on the considered criteria, the more it is preferable to another), and non-redundancy (no superfluous criteria are considered) [22].

With respect to set A, a DM may wish to get recommendation on one of the following questions: (i) what is the subset of best alternatives in A, (ii) how to assign alternatives from A to pre-defined and preference ordered classes, or (iii) how to rank the alternatives from A from the best to the worst.

It is well known that the only objective information coming out from the above problem statement is a dominance relation in set A. Let us recall that according to the *dominance relation* in set A, alternative $x \in A$ is preferred to alternative $y \in A$, $x \succ y$, if and only if $g_i(x) \geq g_i(y)$ for all $i \in I$, with at least one strict inequality. Moreover, x is indifferent to y, $x \sim y$, if and only if $g_i(x) = g_i(y)$ for all $i \in I$. Hence, for any two alternatives $x, y \in A$, one of the four situations may arise: $x \succ y$, $y \succ x$, $x \sim y$ and $x?y$, where the last one means that x and y are incomparable. The dominance relation is a partial preorder (i.e., a reflexive and transitive binary relation) and it is in general very poor, because the most frequent situation is $x?y$. In order to "enrich" this relation, the analyst must learn more about a value system of the DM, so as to be able to construct a DM's preference model. This is why the preference information elicited by the DM is a necessary component of decision aiding. The preference model can finally be used to work out a recommendation with respect to one of the above mentioned questions.

In what follows, the evaluation of each alternative $x \in A$ on each criterion $g_i \in F$ will be denoted either by $g_i(x)$ or x_i. Let G_i denote the value set (scale) of criterion g_i, $i \in I$. Consequently, the Cartesian product of all G_i's,

$$G = \prod_{i=1}^{n} G_i,$$

represents the evaluation space, and $x \in G$ denotes a profile of an alternative in such a space. We consider a weak preference relation $\succsim$ on A which means, for each pair of vectors, $x, y \in G$,

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

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

1) $x \succ y \equiv [x \succsim y$ and not $y \succsim x] \Leftrightarrow$ "x is preferred to y", and
2) $x \sim y \equiv [x \succsim y$ and $y \succsim x] \Leftrightarrow$ "x is indifferent to y".

From a pragmatic point of view, it is reasonable to assume that $G_i \subseteq \mathbb{R}$, for $i = 1, \ldots, n$. More specifically, we shall assume that the evaluation scale 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 alternative $x \in A$ is associated with an evaluation vector denoted by $\underline{g}(x) = (x_1, x_2, \ldots, x_n) \in G$.

Among many preference models considered in the literature, the most popular is an *additive value function* defined on A, such that for each $\underline{g}(x) \in G$,

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

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

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

3.2. Ordinal regression. Like other preference models, the additive value function of a given DM is unknown a priori. A direct elicitation of this function by a DM is counterproductive in real-world decision aiding situations because of a high cognitive effort required. Eliciting indirect preferences in the form of holistic pairwise comparisons of some reference or training alternatives is much less demanding of cognitive effort. This kind of preference information is given as decision examples. Such a reverse search of a preference model from decision examples is done by so-called *ordinal regression* (also called disaggregation-aggregation approach). The preference model found by ordinal regression is *compatible* with the given preference information, i.e., it restores the holistic pairwise comparisons made by the DM. Finally, it is used on the whole set A of alternatives in order to work out a recommendation in terms of the best choice (i), or classification (ii), or ranking (iii). As in NEMO we will use the preference model to recommend a ranking, in the following we concentrate on problem (iii) only.

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 [23]. 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 [24–27], and those using an outranking relation as preference model [28, 29]. This paradigm has also been used since the 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 [30].

3.3. Ordinal regression with robustness considerations. Usually, among the many sets of parameters of a preference model representing the preference information, only one specific set is used to give a recommendation on a set of alternatives. For example, among many value functions representing pairwise comparisons of some alternatives made by the DM, only one value function is finally used to recommend the best choice, or sorting, or ranking of alternatives. Since the choice of one among many sets of parameters compatible with the preference information is rather arbitrary, *robust ordinal regression* (ROR) has been recently proposed with the aim of taking into account all the sets of parameters compatible with the preference information given by the DM [7–9]. The robust ordinal regression approach extends the simple ordinal regression by taking into account not a single instance of the preference model compatible with DM's preference information, but the whole set of compatible instances of the preference model. As a result of considering the whole set of compatible instances of the preference model, one gets two kinds of results with respect to each pair of alternatives $x, y \in A$:

- *necessary preference relation* $x \succsim^N y$, if and only if x is at least as good as y according to all instances of the preference model compatible with the preference information,
- *possible preference relation* $x \succsim^P y$, if and only if x is at least as good as y according to at least one instance of the preference model compatible with the preference information.

The necessary preference relation can be considered as *robust* with respect to the preference information. The robustness of the necessary preference relation refers to the fact that a given pair of alternatives compares in the same way whatever the instance of the preference model compatible with the preference information. Indeed, when no preference information is given, the necessary preference relation boils down to the dominance relation, and the possible preference relation is a complete relation. Every new item of the preference information, e.g., a pairwise comparison of some reference alternatives for which the dominance relation does not hold, is enriching the necessary preference relation and it is impoverishing the possible preference relation, 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 multiobjective optimization. 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.

Computational issues of ROR with respect to the ranking problem (iii) are explained in the next point.

3.4. Computation of a necessary preference ranking on set A, using robust ordinal regression. In NEMO, we will use ROR to set up a necessary preference ranking on a current population of solutions (alternatives). This is why in this point, we explain after [8] and [9], how this ranking is computed.

Preference information. The preference information is given on a subset of reference alternatives $A^R \subseteq A$. The reference alternatives are non-dominated alternatives contained in set A for which the DM is able to express holistic preferences. The DM is expected to make pairwise comparison of few pairs of reference alternatives, i.e., $x \succsim y$ for some pairs $x, y \in A^R$ (in our experiments, we used from 1 to 6 pairs for a population of 32 alternatives).

Additive models. The assumed preference model is an additive value function (2), composed of marginal value functions $u_i(x_i)$, $i \in I$, having one of two forms:

(a) piecewise-linear,
(b) general, non-decreasing.

In case (a), the ranges $[\alpha_i, \beta_i]$ are divided into $\gamma_i \geq 1$ equal sub-intervals $[x_i^0, x_i^1]$, $[x_i^1, x_i^2]$, ..., $[x_i^{\gamma_i - 1}, x_i^{\gamma_i}]$, where $x_i^j = \alpha_i + \frac{j}{\gamma_i}(\beta_i - \alpha_i)$, $j = 0, \ldots, \gamma_i$, and $i \in I$. The marginal value of an alternative $x \in A$ is obtained by linear interpolation,

$$u_i(x_i) = u_i(x_i^j) + \frac{x_i - x_i^j}{x_i^{j+1} - x_i^j}(u_i(x_i^{j+1}) - u_i(x_i^j)), \quad x_i \in [x_i^j, x_i^{j+1}]. \tag{3}$$

The piecewise-linear additive model is completely defined by the marginal values at the breakpoints, i.e., $u_i(x_i^0) = u_i(\alpha_i)$, $u_i(x_i^1)$, $u_i(x_i^2)$, $\cdots$, $u_i(x_i^{\gamma_i}) = u_i(\beta_i)$, $i \in I$. The number of linear pieces γ_i is fixed a priori for each marginal value function u_i, $i \in I$.

In case (b), the characteristic points of marginal value functions $u_i, i \in I$, are fixed in evaluation points of considered alternatives. Let τ_i be the permutation on the set of indices of alternatives from A^R that reorders them according to the increasing evaluation on criterion i, i.e.,

$$\alpha_i \leq x_{\tau_i(1)} \leq x_{\tau_i(2)} \leq \ldots \leq x_{\tau_i(m-1)} \leq x_{\tau_i(m)} \leq \beta_i, \; i \in I.$$

The general, non-decreasing additive model is completely defined by the marginal values at the characteristic points, i.e., $u_i(\alpha_i)$, $u_i(x_{\tau_i(1)})$, $u_i(x_{\tau_i(2)})$, $\ldots$, $u_i(x_{\tau_i(m)})$, $u_i(\beta_i)$. Remark that in this case, no linear interpolation is required to express the marginal value of any reference alternative.

Compatible value functions. Formally, to be compatible with the preference information, the additive value function $U(x) = \sum_{i=1}^{n} u_i(x_i)$ should satisfy the following set of linear programming constraints corresponding to the DM's preference information:

- in case (a) of piecewise-linear marginal value functions, these are

$$\left.\begin{array}{l} \left.\begin{array}{l} U(x) > U(y) \Leftrightarrow\ x \succ y \\ U(x) = U(y) \Leftrightarrow\ x \sim y \end{array}\right\} \text{for all } x, y \in A^R \\ u_i(x_i^{j+1}) - u_i(x_i^j) \geq 0,\ i=1,...,n,\ j=0,...,\gamma_i - 1, \\ u_i(\alpha_i) = 0,\ i = 1,...,n, \\ \sum_{i=1}^{n} u_i(\beta_i) = 1, \end{array}\right\} (E_a^{A^R})$$

- in case (b) of general, non-decreasing marginal value functions, these are

$$\left.\begin{array}{l} \left.\begin{array}{l} U(x) > U(y)\ \Leftrightarrow\ x \succ y \\ U(x) = U(y)\ \Leftrightarrow\ x \sim y \end{array}\right\} \text{for all } x, y \in A^R \\ u_i(x_{\tau_i(j)}) - u_i(x_{\tau_i(j-1)}) \geq 0,\ i=1,...,n,\ j=2,...,m, \\ u_i(x_{\tau_i(1)}) \geq 0,\ u_i(x_{\tau_i(m)}) \leq u_i(\beta_i),\ i = 1,...,n, \\ u_i(\alpha_i) = 0,\ i = 1,...,n, \\ \sum_{i=1}^{n} u_i(\beta_i) = 1. \end{array}\right\} (E_b^{A^R}).$$

Computation of the possible and necessary preference relations in set A. In order to compute the possible preference relation $\succsim^P$ and the necessary preference relation $\succsim^N$ in the complete set of alternatives A, ROR proceeds as follows. For all alternatives $w, z \in A$, let π_i be a permutation of the indices of alternatives from set $A^R \cup \{w, z\}$ that reorders them according to increasing evaluation on criterion i, i.e.,

$$x_{\pi_i(1)} \leq x_{\pi_i(2)} \leq ... \leq x_{\pi_i(\omega-1)} \leq x_{\pi_i(\omega)},$$

where

- if $A^R \cap \{w, z\} = \emptyset$, then $\omega = m + 2$,
- if $A^R \cap \{w, z\} = \{w\}$ or $A^R \cap \{w, z\} = \{z\}$, then $\omega = m + 1$,
- if $A^R \cap \{w, z\} = \{w, z\}$, then $\omega = m$.

Then, one can fix the characteristic points of $u_i, i \in I$, in

$$x_i^0 = \alpha_i, \quad x_i^j = x_{\pi_i(j)} \ \text{ for } j = 1, ..., \omega, \quad x_i^{\omega+1} = \beta_i.$$

Let us consider the following set $E(w, z)$ of ordinal regression constraints, with $u_i(x_i^j)$, $i = 1, ..., n$, $j = 1, ..., \omega + 1$, as variables:

$$\left.\begin{array}{l} \left.\begin{array}{l} U(x) \geq U(y) + \varepsilon\ \Leftrightarrow\ x \succ y \\ U(x) = U(y)\ \Leftrightarrow\ x \sim y \end{array}\right\} \forall x, y \in A^R \\ u_i(x_i^j) - u_i(x_i^{j-1}) \geq 0,\ i=1,...,n,\ j=1,...,\omega+1 \\ u_i(x_i^0) = 0,\ i = 1,...,n \\ \sum_{i=1}^{n} u_i(x_i^{\omega+1}) = 1 \end{array}\right\} (E(w,z))$$

where ε is an arbitrarily small positive value. The set of constraints $E(w, z)$ depends on the pair of alternatives $w, z \in A$ because their evaluations $g_i(w)$ and $g_i(z)$ give coordinates for two of $(\omega + 1)$ characteristic points of marginal value function u_i, for each $i = 1, \ldots, n$.

Suppose that the polyhedron defined by the set of constraints $E(w, z)$ is not empty. In this case, we have that:

$$w \succsim^P z \ \Leftrightarrow\ D(w, z) \geq 0$$

where

$$\begin{array}{rcl} D(w, z) & = & \max\{U(w) - U(z)\} \\ & & \text{s.t. set } E(w, z) \text{ of constraints} \end{array} \tag{4}$$

and

$$w \succsim^N z \ \Leftrightarrow\ d(w, z) \geq 0$$

where

$$\begin{array}{rcl} d(w, z) & = & \min\{U(w) - U(z)\} \\ & & \text{s.t. set } E(w, z) \text{ of constraints.} \end{array} \tag{5}$$

In order to make the possible preference relation $\succsim^P$ and the necessary preference relation $\succsim^N$ independent of a fixed value of ε in $E(w, z)$, we have to solve the following linear programming problems instead of solving (4) and (5):

$$\begin{array}{rcl} \varepsilon^P(w, z) & = & \max \varepsilon \\ & & \text{s.t. set } E(w, z) \text{ of constraints,} \\ & & \text{plus the constraint } U(w) \geq U(z). \end{array} \tag{6}$$

and

$$\begin{array}{rcl} \varepsilon^N(w, z) & = & \max \varepsilon \\ & & \text{s.t. set } E(w, z) \text{ of constraints,} \\ & & \text{plus the constraint } U(z) \geq U(w) + \varepsilon. \end{array} \tag{7}$$

Then, one can conclude:

$$w \succsim^P z \Leftrightarrow \varepsilon^P(w, z) > 0$$

and

$$w \succsim^N z \Leftrightarrow \varepsilon^N(w, z) \leq 0.$$

As demonstrated in [8], the possible preference relation $\succsim^P$ is strongly complete and negatively transitive, while the necessary preference relation $\succsim^N$ is inducing a partial preorder on set A.

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: the necessary ranking and the possible ranking. 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 alternatives (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 alternative. Recently, a methodology to identify the "most representative" function in ROR without loosing the advantage of taking into account all compatible value functions has been proposed in [31]. The idea is to select among all compatible value functions the most discriminant value function for consecutive alternatives in the necessary ranking, i.e., that value function which maximizes the difference of scores between alternatives related by preference in the necessary ranking. To break ties, one can wish to minimize the difference of scores between alternatives 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 alternatives.
2. For all pairs of alternatives (x, y), such that x is necessarily preferred to y, add the following constraints to the linear programming constraints of ROR: $U(x) \geq U(y) + \theta$.
3. Maximize the objective function θ.
4. Add the constraint $\theta = \theta^*$, with θ^* being the resulting maximal θ from point 3), to the linear programming constraints of point 2).
5. For all pairs of alternatives (x, y), such that neither x is necessarily preferred to y nor y is necessarily preferred to x, add the following constraints to the linear programming constraints of ROR and to the constraints considered in above point 4): $U(x) - U(y) \leq \delta$ and $U(y) - U(x) \leq \delta$.
6. Minimize δ.

This procedure maximizes the minimal difference between values of alternatives 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 alternatives for which the necessary preference holds, and the smallest maximal difference between values of alternatives 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 ROR. 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 $M\theta - \delta$, 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 nonlinear way.

4. Necessary-preference-enhanced Evolutionary Multiobjective Optimization – NEMO

Our main idea is to integrate the concept of ROR into an EMO approach, in particular NSGA-II [10]. NSGA-II is one of today's most prominent and most successful EMO algorithms. It ranks individuals (solutions in a population) 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-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 relation. 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 Subsec. 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:

$$\text{Crowding distance}(x) = \sum_{i=1}^{n} |u_i(y_i^x) - u_i(z_i^x)| - |U(y^x) - U(z^x)|,$$

where U is the "most representative value function", u_i are its marginal value functions, y_i^x and z_i^x are left and right neighbors of x in the dimension of marginal value u_i, and y^x and z^x are vectors composed of y_i^x and z_i^x, respectively, $i = 1, \ldots, n$. 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.[1]

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.

[1]Our first attempt was to simply calculate the crowding distance in marginal utility space, i.e., as $\sum_{i=1}^{n} |u_i(y_i^x) - u_i(z_i^x)|$. However, we found that the formulation specified above, which additionally takes into account the overall utility of the vectors of neighbors, works even better in practice, so this is what we use in the experiments. More work is necessary to get a better understanding of various possible ways of calculating the crowding distance.

Algorithm 1 Basic NEMO

Generate initial solutions randomly
Elicit the user's preferences {Present to user some pairs 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 the user's preferences
  end if
  Determine *necessary* ranking
  Determine *secondary* ranking
  Environmental selection
until Stopping criterion met
Return all preferred solutions according to necessary ranking

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 k 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 pair of solutions for which no necessary preference relation holds. This prevents the user from specifying inconsistent information.

5. Experimental results

We tested NEMO on two simple test functions from [32], each having two objectives and 30 real-valued decision variables. One test function, ZDT1, has a convex Pareto-front, while the other (ZDT2) has a concave Pareto-front.

An empirical evaluation of interactive EMO methods is challenging, because the test environment has to include a model of the user behavior. We therefore use an artificial user which applies a pre-specified value function for decision making. Obviously, this value function is not known to NEMO, but only used by the artificial user (DM) for comparing two solutions when preferences are elicited. Two value functions are used in our tests:

- **Linear** – assumes the artificial user has a value function that is a linear weighting of the objectives. More specifically, we used $U(x) = -(0.6f_1(x) + 0.4f_2(x))$, i.e., his/her goal is to minimize $0.6f_1(x) + 0.4f_2(x)$.
- **Chebycheff** – assumes the artificial user has a Chebycheff-like value function. More specifically, his/her goal is to minimize $\max\{0.6f_1(x), 0.4f_2(x)\}$.

For the internal preference model learned by NEMO, we also tested two options:

- **Linear** – simply learns a linear weighting of the objectives. As mentioned in Sec. 2, NEMO with this preference model is almost equivalent to the approach by Greenwood et al. [19], although used in an interactive fashion rather than eliciting all user preferences before the optimization starts.
- **General additive** – allows for arbitrary non-decreasing marginal value functions as outlined above.

NEMO is set up such that in every k-th generation, it randomly selects two individuals which are incomparable in the the necessary ranking (no necessary preference relation holds for them), and receives as feedback the solution preferred by the artificial user according to the assumed value function. The information from the DM is then used to update the internal preference model. We used a population size of 32, simulated binary crossover with crossover probability of 0.9 and $\eta_c = 1$, and Gaussian mutation with probability 0.03 and standard deviation $\sigma = 0.01$.

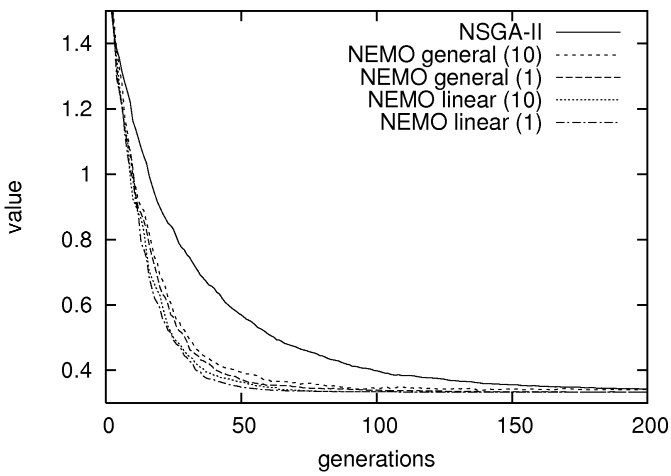
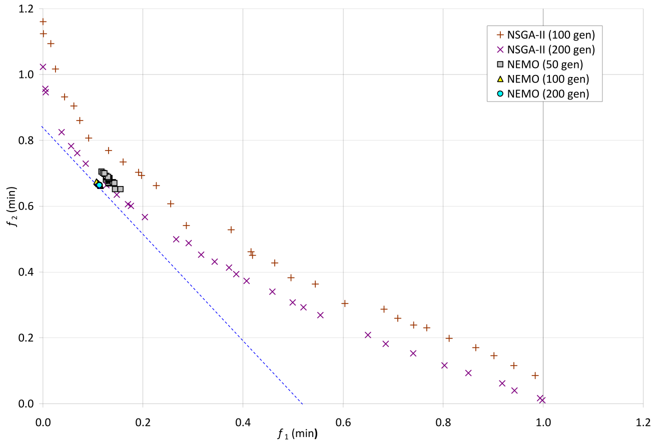
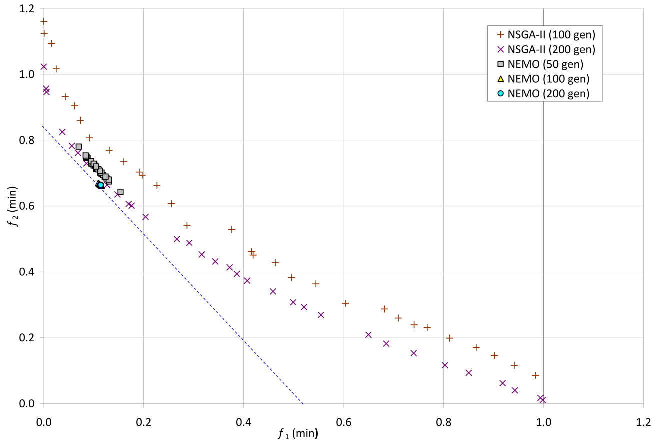


Fig. 1. Value of best solution in population depending on the number of generations for problem ZDT1. Numbers in parentheses denote number of generations between preference elicitation steps

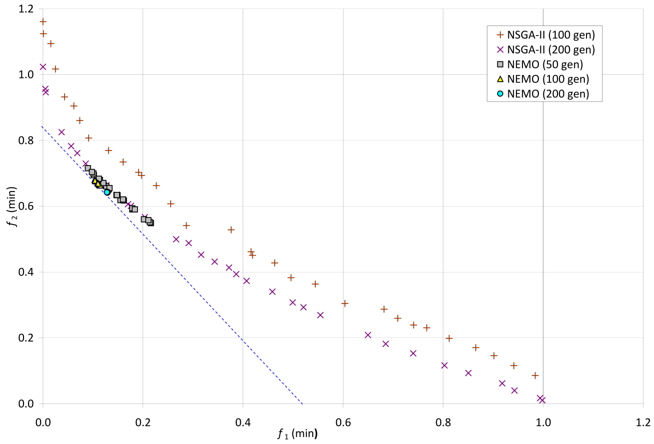
a) ZDT1, linear preference model, elicitation every generation



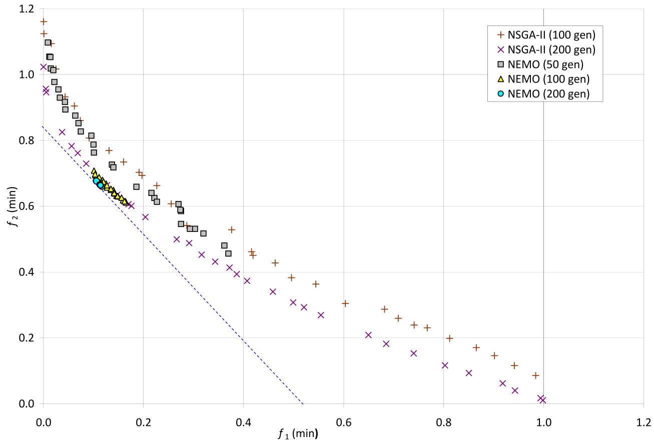
b) ZDT1, linear preference model, elicitation every 10 generations



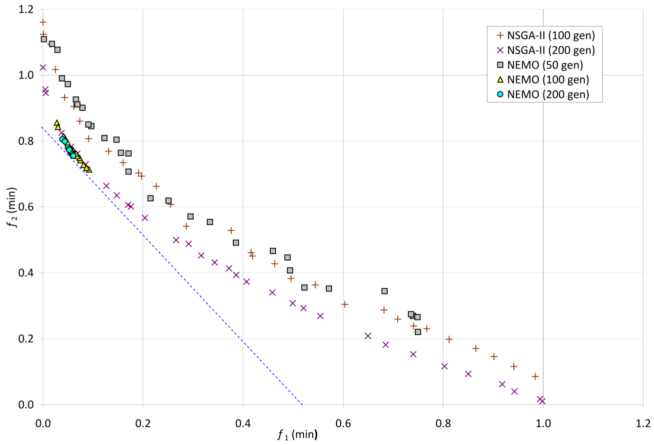
c) ZDT1, linear preference model, elicitation every 20 generations



d) ZDT1, general additive preference model, elicitation every generation



e) ZDT1, general additive preference model, elicitation every 10 generations



f) ZDT1, general additive preference model, elicitation every 20 generations

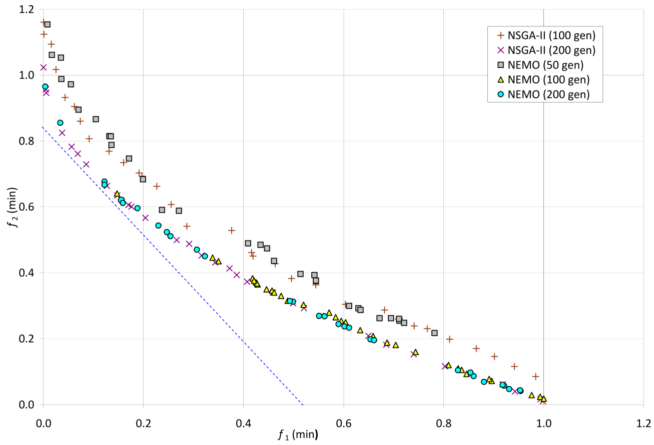


Fig. 2. Some exemplary populations for test problem ZDT1, with different internal preference models and different frequency of preference elicitation.

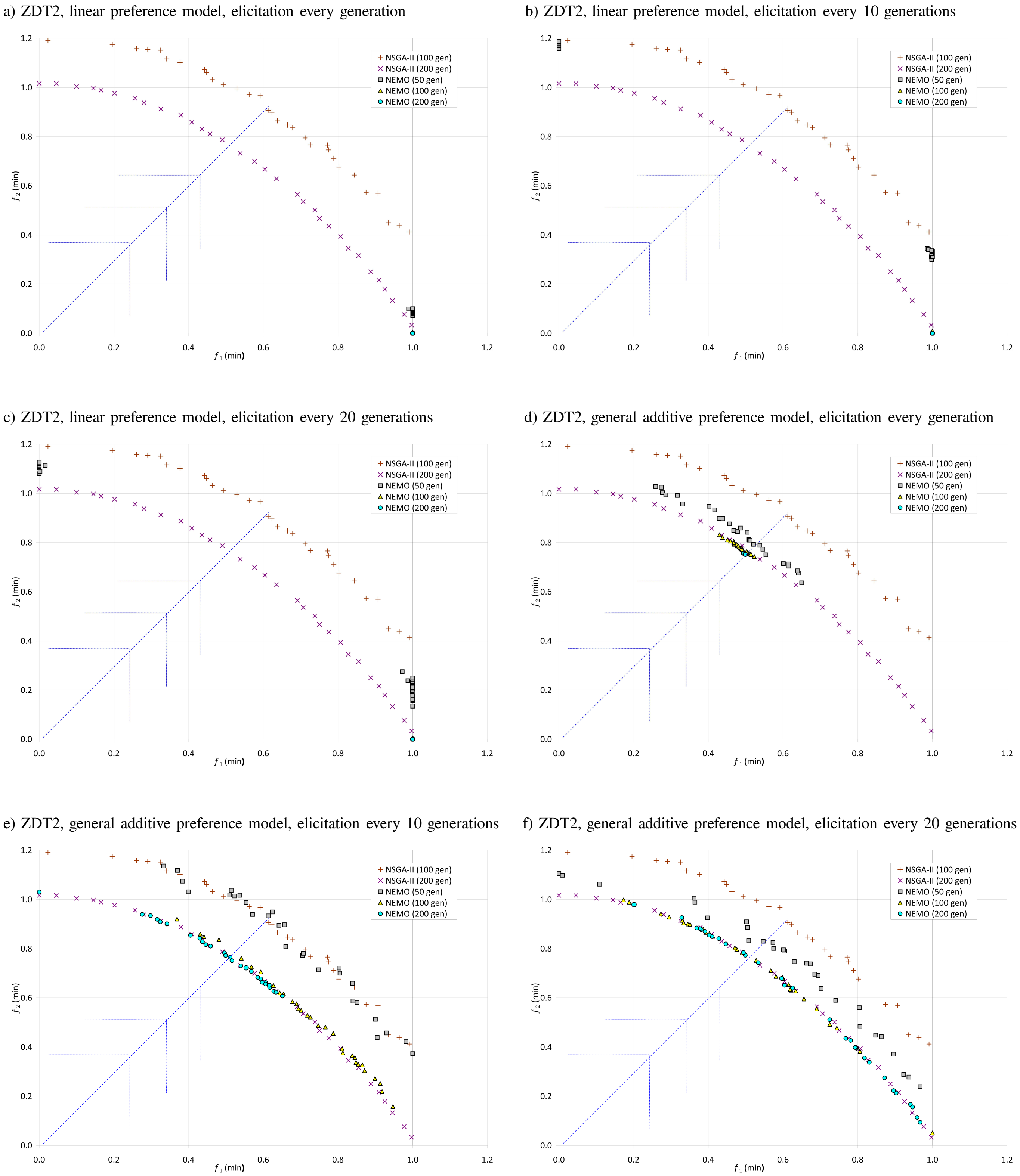


Fig. 3. Some exemplary populations for test problem ZDT1, with different internal preference models and different frequency of preference elicitation.

Figure 2 shows the results of NEMO on test function ZDT1 and an artificial DM with a linear utility function. Let us first consider panels (a)-(c), where the NEMO preference model is also linear. Clearly, while NSGA-II attempts to approximate the whole front, NEMO learns to focus the search on the area most interesting for the artificial DM. This allows NEMO to find more interesting solutions quicker, as can be seen by the fact that the solutions found by NSGA-II after 200 generations are found by NEMO already after 100 generations. For panel (a), a preference elicitation step is performed every generation, for panel (b) every 10 generations and panel (c) only every 20 generations, corresponding to 100, 20, and 10 pairwise comparisons during the run of 200 generations. We can observe that the more frequent the preference elicitation step, the stronger NEMO can focus the search. The reason is that with more preference information, the set of compatible value functions is narrowed down, and NEMO has a more precise notion of what the user wants. Note, however, that this is not just a user parameter to influence the focusing. NEMO will, by design, narrow down the search to a maximally possible extent given the preference information provided, without excluding any solution that might still be preferred.

The same plots but for NEMO with a general additive preference model are shown in Fig. 2 (d)-(f). Again, we see a focus on the most interesting region, and stronger focus for more provided information. When preferences are elicited only every 20 generations, there is barely any focusing effect remaining. When comparing the overall performance of NEMO with general additive model to NEMO with linear model, it seems the general additive preference model performs worse. This is not surprising, as given an artificial user with linear value function, a linear preference model is sufficient to capture the user's preferences, while it has fewer degrees of freedom and thus requires less information for learning a practically useful model.

Similar insights can be gained from the convergence plot (Fig. 1) which shows the user's value for the most preferred solution in the population over time. These figures also clearly show that NEMO not only correctly focuses on the most relevant solution, but also finds these solutions more quickly than the standard NSGA-II. For example, to reach the same solution quality that NSGA-II reaches after 95 generations, NEMO requires only around 40 generations. What kind of preference model is used seems to have, at least for this test problem, a larger impact than the number of preference elicitations. The disadvantage of using the general additive preference model in NEMO can not be compensated by eliciting 10 times as much information (compare NEMO general (1) with NEMO linear (10))[2].

Now let us look at the ZDT2 problem with a concave Pareto-front and with an artificial DM having a Chebycheff value function (Fig. 3). Here, a linear preference model (panels (a)-(c)) is not sufficient to capture the user's preferences, as there is no linear combination of objectives that would yield the most preferred solution in the concave region of the Pareto-front. So it is not surprising that the model eventually converges to a solution which is at the lower right corner of the Pareto-front, away from the truly most preferred solution. When NEMO uses the general additive preference model, however, (panels (d)-(f)) it does converge correctly to the area of the most preferred solution. Again, more preference elicitation steps allow a stronger focus of the search, and NEMO is generally faster as the solutions after 100 generations roughly correspond to the NSGA-II solutions after 200 generations.

The convergence plots support our observations (see Fig. 4). NEMO with linear preference model converges quickly in the beginning, as the model has few parameters to learn. But eventually, it can not capture the complexity of the artificial DM's Chebycheff value function and solution quality actually worsens again and converges to a clearly sub-optimal level. With the general additive preference model, the Chebycheff value function can be learned and the algorithm converges to the right area of the Pareto-front. Convergence is slightly slower in the beginning than NEMO with linear preference model, as more parameters need to be learned, but it is still significantly faster than not using any preferences as in standard NSGA-II.

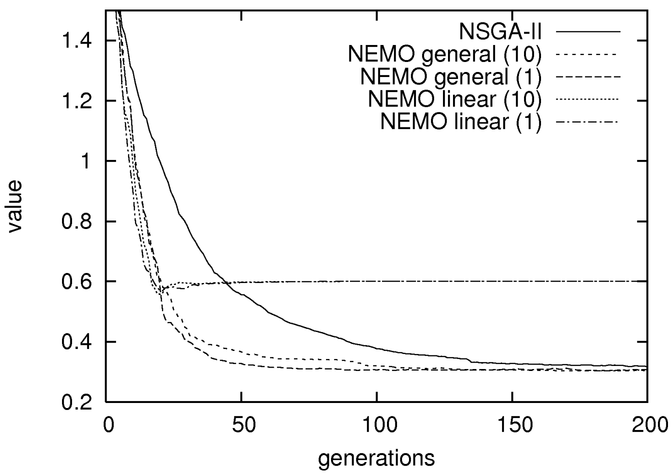


Fig. 4. Value of best solution in population depending on the number of generations for problem ZDT2. Numbers in parentheses denote number of generations between preference elicitation steps

Overall, the experiments demonstrate that the simple linear preference model from [19] is not always sufficient to capture the user preferences. NEMO, on the other hand, allows to choose from a range of preference models and thus can accommodate a much larger variety of user value functions.

6. Conclusions

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

[2]Results for NSGA-II have been averaged over 10 runs, results for NEMO have been averaged over 4 runs.

1. It models the user's preferences in terms of very general additive value functions,
2. It requires a preference information expressed in a simple and intuitive way (comparisons of solutions),
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.

Our 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 the user's 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 simple pairwise comparisons, we plan to integrate into ROR intensities of preferences, and 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 investigate 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 calculations and the convergence to the most interesting part of the Pareto-front.

REFERENCES

[1] C.A. Coello Coello, D.A. Van Veldhuizen, and G.B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer, Dordrecht, 2002.

[2] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, Chichester, 2001.

[3] K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer, Dordrecht, 1999.

[4] R. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application*, Wiley, New York, 1986.

[5] Ph. Vincke, *Multicriteria Decision Aid*, Wiley, Chichester, 1992.

[6] J. Branke, K. Deb, K. Miettinen, and R. Słowiński, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, Springer, Berlin, 2008.

[7] S. Greco, R. Słowiński, J. Figueira, and V. Mousseau, "Robust ordinal regression", in *Trends in Multiple Criteria Decision Analysis*, pp. 241–283, ed. M. Ehrgott, J. Figueira, and S. Greco, Springer, Berlin, 2010.

[8] S. Greco, V. Mousseau, and R. Słowiński, "Ordinal regression revisited: multiple criteria ranking with a set of additive value functions", *Eur. J. Operational Research* 191 (2), 415–435 (2008).

[9] J. Figueira, S. Greco, and R. Słowiński, "Building a set of additive value functions representing a reference preorder and intensities of preference: GRIP method", *Eur. J. Operational Research* 195 (2), 460–486 (2009).

[10] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm", *NSGA-II. IEEE Transactions on Evolutionary Computation* 6 (2), 182–197 (2002).

[11] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz, "Interactive evolutionary multiobjective optimization using robust ordinal regression", in *Evolutionary Multi-Criterion Optimization (EMO'09)*, vol. 5467, pp. 554–568, ed. M. Ehrgott, C.M. Fonseca, X. Gandibleux, J.-K. Hao, M. Sevaux, Springer, Berlin, 2009.

[12] C.M. Fonseca and P.J. Fleming, "Genetic algorithms for multiobjective optimization: formulation, discussion, and generalization", *Proc. Fifth Int. Conf. on Genetic Algorithms* 1, 416–423 (1993).

[13] K. Deb, J. Sundar, U.B. Rao N., and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms", *Int. J. Computational Intelligence Research* 2 (3), 273–286 (2006).

[14] L. Thiele, K. Miettinen, P.J. Korhonen, and J. Molina, "A preference-based interactive evolutionary algorithm for multiobjective optimization", *Evolutionary Computation Journal* 17 (3), 411–436 (2009).

[15] J. Branke, T. Kaußler, and H. Schmeck, "Guidance in evolutionary multi-objective optimization", *Advances in Engineering Software* 32, 499–507 (2001).

[16] K. Deb and S. Chaudhuri, "I-MODE: an interactive multiobjective optimization and decision-making using evolutionary methods", *Applied Soft Computing* 10, 496–511 (2010).

[17] A. Jaszkiewicz and J. Branke, "Interactive multiobjective evolutionary algorithms", in *Multiobjective Optimization – Interactive and Evolutionary Approaches*, vol. LNCS 5252, pp. 179–193, ed. J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Springer, Berlin, 2008.

[18] J. Branke, "Consideration of partial user preferences in evolutionary multiobjective optimization", in *Multiobjective Optimization – Interactive and Evolutionary Approaches*, vol. LNCS 5252, pp. 157–178, ed. J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Springer, Berlin, 2008.

[19] G.W. Greenwood, X.S. Hu, and J.G. D'Ambrosio, "Fitness functions for multiple objective optimization problems: combining preferences with Pareto rankings", in *Foundations of Genetic Algorithms*, pp. 437–455, ed. R.K. Belew and M.D. Vose, Morgan Kaufmann, Berlin, 1997.

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

[21] A. Jaszkiewicz, "Interactive multiobjective optimization with the Pareto memetic algorithm", *Foundations of Computing and Decision Sciences* 32 (1), 15–32 (2007).

[22] B. Roy and D. Bouyssou, *Aide Multicritère à la Décision: Méthodes et Cas*, Economica, Paris, 1993.

[23] J. March, "Bounded rationality, ambiguity and the engineering of choice", *Bell J. Economics* 9, 587–608 (1978).

[24] V. Srinivasan and A.D. Shocker, "Estimating the weights for multiple attributes in a composite criterion using pairwise judgments", *Psychometrika* 38, 473–493 (1973).

[25] D. Pekelman and S.K. Sen, "Mathematical programming models for the determination of attribute weights", *Management Science* 20, 1217–1229 (1974).

[26] E. Jacquet-Lagrèze and Y. Siskos, "Assessing a set of additive utility functions for multicriteria decision making: the UTA method", *Eur. J. Operational Research* 10, 151–164 (1982).

[27] Y. Siskos, V. Grigoroudis, and N. Matsatsinis, "UTA methods", in *Multiple Criteria Decision Analysis: State-of-the-Art Surveys*, pp. 297–343, ed. F. Figueira, S. Greco and M. Ehrgott, Springer, Berlin, 2005.

[28] L. Kiss, J.M. Martel, and R. Nadeau, "ELECCALC – an interactive software for modelling the decision maker's preferences", *Decision Support Systems* 12, 757–777 (1994).

[29] V. Mousseau and R. Słowiński, "Inferring an ELECTRE TRI model from assignment examples", *J. Global Optimization* 12, 157–174 (1998).

[30] S. Greco, B. Matarazzo, and R. Słowiński, "Rough sets theory for multicriteria decision analysis", *Eur. J. Operational Research* 129 (1), 1–47 (2001).

[31] J. Figueira, S. Greco, and R. Słowiński, "Identifying the "most representative" value function among all compatible value functions in the GRIP method", *68th Meeting of the EURO Working Group on MCDA* 1, CD-ROM (2008).

[32] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results", *Evolutionary Computation Journal* 8 (2), 125–148 (2000).