Using Ordinal Regression for Interactive Evolutionary Multiple Objective Optimization with Multiple Decision Makers

Miłosz Kadziński(✉) and Michał Tomczyk

Institute of Computing Science, Poznań University of Technology, Poznań, Poland
milosz.kadzinski@cs.put.poznan.pl, michal.k.tomczyk@gmail.com

Abstract. We present an interactive evolutionary multiple objective optimization (MOO) method incorporating preference information of several decision makers into the evolutionary search. It combines NSGA-II, a well-known evolutionary MOO method, with some interactive value-based approaches based on the principle of ordinal regression. We introduce several variants of the method distinguished by an elitist function indicating a comprehensive value that each solution represents to the group members. The experimental results confirm that all proposed approaches are able to focus the search on the group-preferred solutions, differing, however, with respect to both part of the Pareto front to which they converge as well as the convergence speed measured in terms of a change of utilitarian value of the returned solutions.

Keywords: Evolutionary multiple objective optimization · Interactive method · Group decision · Additive value function · Preference disaggregation · NEMO

1 Introduction

In Multiple Objective Optimization (MOO), several objectives are optimized simultaneously. As goals to be attained usually represent conflicting viewpoints, it is impossible to find a solution for which all objectives reach their individual optima [1]. Instead, we can identify a set of Pareto-optimal (non-dominated) solutions which are considered equivalent in case no additional information is available. A solution is called Pareto-optimal if none of the objective functions can be improved in value without deteriorating some of the other objective values. A possibly infinite set of such solutions forms a Pareto front in the objective space.

Traditionally, in MOO, two separate methodological streams have been developed: evolutionary and interactive ones [1]. On the one hand, the role of Evolutionary MOO (EMO) is to approximate the entire Pareto front. On the other hand, Interactive MOO (IMO) deals with identification of the most preferred solution. IMO techniques require participation of a Decision Maker (DM) who

B. Kamiński et al. (Eds.): GDN 2015, LNBIP 218, pp. 185–198, 2015.
DOI: 10.1007/978-3-319-19515-5_15

is expected to provide her/his subjective preference information. By developing a comprehensive model of such preferences, IMO makes the Pareto optimal solutions more comparable.

The recent trend in MOO consists in merging the interactive and evolutionary approaches (for a review, see [1,2]). This is achieved by integrating preference information into the EMO algorithms already during their optimization runs. The appealing effects of such integration consist in focusing the search on the area of the Pareto front which is most suitable to the DM and in speeding the convergence towards the most preferred region of the objective space.

The existing interactive EMO methods incorporate user preferences into evolutionary algorithms in different ways. In this paper, we focus on Necessary-preference-enhanced Evolutionary Multiobjective Optimizer (NEMO) [2] which combines the evolutionary method, called Non-dominated Sorting Genetic Algorithm II(NSGA-II) [3], with interactive ordinal regression approaches [4,5]. NEMO requires the DM to compare some pairs of solutions from the current population, and constructs a set of compatible general additive value functions. Then, it exploits these functions so that to guide the evolutionary search into regions of the Pareto front which are more desirable from the DM's point of view.

Our interest in NEMO comes from its favorable characteristics in terms of both preference information and preference model it employs. When it comes to pairwise comparisons, they represent indirect preference information whose elicitation is less demanding in terms of cognitive effort of the DM than direct specification of values for some preference model parameters. As far as general additive value functions are concerned, they can be computed efficiently with Linear Programming (LP), at the same time being flexible enough to handle preference information provided by the DMs with different value systems. Moreover, they do not require a pre-defined scaling of the objectives [2].

NEMO, alike other existing interactive EMO methods, was originally designed to deal with preferences expressed by a single DM. However, it is group decision making that is among the most important and frequently encountered processes within companies and organizations. When dealing with multiple DMs in the context of MOO, the main challenge consists in designing the algorithms so that they are able to focus the search on the group consensus solutions.

In this paper, we extend NEMO so that it is capable of dealing with MOO group decision problems. In particular, we propose a few variants of NEMO-GROUP that incorporate preference information of several DMs. Each of these variants is distinguished by a unique elitist function which indicates a comprehensive value that each solution represents to the group members. These values are employed to properly modify NSGA-II so that it promotes the group-preferred solutions in the optimization run. The use of proposed methods is illustrated by examples and experiments revealing the differences with respect to the regions of the Pareto front to which the methods converge, the quality of constructed solutions measured in terms of their utilitarian value, and the convergence speed.

The paper is organized as follows. The next section provides a brief reminder of ordinal regression methods. Section 3 describes the basic concepts of NSGA-II and NEMO. Section 4 presents different variants of our method, NEMO-GROUP. The experimental results are discussed in Sect. 5. The last section concludes.

2 Ordinal Regression

We are considering a multiple criteria decision problem where a set of solutions $A = \{a_1, a_2, \ldots\}$ is evaluated on a family $F = \{g_1, g_2, \ldots, g_n\}$ of n criteria. We assume, without loss of generality, that the smaller $g_j(a)$, $j = 1, \ldots, n$, the better solution a on criterion g_j, for all $a \in A$. Let G_j denote the value set of criterion g_j. Following [2], we assume that $G_j \subseteq \mathbb{R}$, and that the value space on each criterion g_j is bounded, such that $G_j = [\alpha_j, \beta_j]$, $\alpha_j < \beta_j$, where α_j and β_j are, respectively, the worst and the best evaluations. Consequently, $G = G_1 \times G_2 \times \ldots G_n$ represents the evaluation space, and each solution $a \in A$ is associated with an evaluation vector denoted by $g(a) = (g_1(a), g_2(a), \ldots, g_n(a)) \in G$.

Preference Model. To model the preferences provided by the DM and evaluate a set of solutions, we use an additive value function. It is defined on A as follows [4]:

$$U(a) = \sum_{j=1}^{n} u_j(g_j(a)) = \sum_{j=1}^{n} u_j(a), \tag{1}$$

where $u_j : G_j \to \mathbb{R}$, $j = 1, \ldots, n$, are subject to monotonicity and normalization constraints:

$$\left. \begin{array}{l} u_j(g_j(a)) \geq u_j(g_j(b)), \text{ if } g_j(a) < g_j(b), \\ u_j(\alpha_j) = 0, \ \sum_{j=1}^{n} u_j(\beta_j) = 1. \end{array} \right\} E^{\mathcal{U}} \tag{2}$$

Group Preference Model. We consider a set of DMs (let us denote it by $\mathcal{D} = \{DM_1, \ldots, DM_k, \ldots, DM_s\}$, where s is the number of DMs) cooperating to find a subset of the best consensus solutions. We assume that each DM plays the same role in the committee, so we do not differentiate their weights. Each $DM_k \in \mathcal{D}$ evaluates solutions with her/his individual "true" value function U_k^{TRUE}. The collective utilitarian preference model combines these evaluations into a comprehensive value that solution $a \in A$ represents to the whole committee:

$$U_{\mathcal{D}}(a) = 1/s \sum_{k=1}^{s} U_k^{TRUE}(a). \tag{3}$$

Preference Information. Each $DM_k \in \mathcal{D}$ offers individual preference information which is a set B_k of pairwise comparisons of some reference solutions in A_k^{REF}. In the considered setting, either each DM is allowed to choose the solutions (s)he wishes to compare on her/his own or the pairs to be compared by each DM are drawn randomly. Thus, in general, $A_k^{REF} \neq A_l^{REF}$ for $DM_k, DM_l \in \mathcal{D}$. The comparison of a pair $(a^*, b^*) \in B_k \subseteq A_k^{REF} \times A_k^{REF}$ provided by DM_k

states the strict preference, weak preference, or indifference. These relations are denoted by, $a^* \succ_k b^*$, $a^* \succsim_k b^*$, and $a^* \sim_k b^*$, respectively.

Let each pairwise comparison from B_k be denoted by B_k^t, $t = 1, \ldots, p_k$, where p_k is the number of comparisons contained in B_k. The set of constraints E^k given below translates such a reference pre-order provided by DM_k to a value function:

$$\left.\left.\begin{array}{ll} U(a^*) \geq U(b^*) + \varepsilon, & \text{for } \; B_k^t = (a^* \succ_k b^*) \\ U(a^*) \geq U(b^*), & \text{for } \; B_k^t = (a^* \succsim_k b^*) \\ U(a^*) = U(b^*), & \text{for } \; B_k^t = (a^* \sim_k b^*) \end{array}\right\} \text{ for } t = 1, \ldots, p_k \right\} E^k \qquad (4)$$

where ε is an arbitrarily small positive value.

The pairwise comparisons provided by each $DM_k \in \mathcal{D}$ form the input data for the ordinal regression [4] that finds the whole set of value functions $\mathcal{U}_k$ being able to reconstruct these judgments. It is defined by a set of constraints $E^{\mathcal{U}_k} = E^{\mathcal{U}} \cup E^k$. The set of value functions $\mathcal{U}_{\mathcal{D}}$ compatible with the pairwise comparisons of all DMs is defined with $E^{\mathcal{U}_{\mathcal{D}}} = E^{\mathcal{U}} \cup E^k$, $k = 1, \ldots, s$. Note that $\mathcal{U}_{\mathcal{D}}$ corresponds to the intersection of sets of compatible value functions for all DMs in $\mathcal{D}$.

If $\varepsilon^* = max\ \varepsilon$, s.t. $E^{\mathcal{U}_k}$ $(E^{\mathcal{U}_{\mathcal{D}}})$, is greater than 0 and $E^{\mathcal{U}_k}$ $(E^{\mathcal{U}_{\mathcal{D}}})$ is feasible, the set of compatible value functions $\mathcal{U}_k$ $(\mathcal{U}_{\mathcal{D}})$ is non-empty. Otherwise, the provided preference information is inconsistent with the assumed preference model, which means that there is no value function that would reproduce the pairwise comparisons provided by DM_k (if $\mathcal{U}_k = \emptyset$) or all DMs (if $\mathcal{U}_{\mathcal{D}} = \emptyset$).

Representative Value Function. There is usually more than one compatible value function. The issue of selecting a single representative function has been discussed in detail in [6]. In this paper, we will use the most discriminant value function U_k^R $(U_{\mathcal{D}}^R)$, which is obtained by maximizing ε, subject to $E^{\mathcal{U}_k}$ $(E^{\mathcal{U}_{\mathcal{D}}})$. It discriminates comprehensive values of reference solutions related by the preference in the DM's (DMs') partial ranking.

Dealing with Incompatibility of Preference Information. In case of incompatibility, there is no value function compatible with the preference information provided by all DMs. Treating this problem, we will maximize a minimal number of pairwise comparisons of any DM which are consistent, being representable by a single additive value function [7]. It can be achieved by solving the following Mixed Integer Linear Programming (MILP) problem:

$$Maximize\ v, \text{ s.t.} \qquad (5)$$

$$\left.\begin{array}{l} \text{for } k = 1, \ldots, s, \text{ for } t = 1, \ldots, |B_k| : \\ \quad [1 - v_t^k(a^*, b^*)] + U(a^*) \geq U(b^*) + \varepsilon, \\ \quad v_t^k(a^*, b^*) \in \{0, 1\}, \\ \text{for } k = 1, \ldots, s : \\ \quad v \leq \sum_{t=1}^{|B_k|} v_t^k(a^*, b^*), \\ E^{\mathcal{U}}. \end{array}\right\} E'^{\mathcal{U}_{\mathcal{D}}}$$

Apart from providing the minimal number of non-contradictory pairwise comparisons of all DMs (v^*), the solution of the above problem indicates which pairwise comparisons can be reproduced together by an additive value function (they are distinguished with $v_t^{k,*} = 1$). If for all DMs the numbers thereof are imbalanced, we arbitrarily choose the last v^* non-contradictory pairwise comparisons provided by each DM, so that none of them is favored. Then, we determine a representative (most discriminant) value function compatible with thus selected subset of holistic judgments.

3 Reminder on NSGA-II and NEMO

The role of genetic algorithms is to estimate meta-heuristically the Pareto fronts in MOO problems. In particular, NSGA-II [3] incorporates a fast non-dominated sorting algorithm to identify Pareto optimal solutions, and a diversity preservation mechanism for maintaining a well-spread Pareto front. It starts with the initialization of a random parent population P_0 of size N. Then, the offspring Q_0 of the same size is created using the usual selection, recombination and mutation operators. Further, the parents and their offspring ($R_t = P_t \cup Q_t$) are combined to obtain a population of size $2N$.

The new population (P_{t+1}) is filled with the best Pareto fronts from R_t (first $\mathcal{F}_1$ (i.e., non-dominated solutions), then $\mathcal{F}_2$ (i.e., solutions dominated only by some solutions from $\mathcal{F}_1$, etc.), until the size of the next front ($\mathcal{F}_l$) is larger than the number of free slots in P_{t+1}. To have exactly N members in the new population and to maintain diversity, the front $\mathcal{F}_l$ is ordered using the crowded distance comparison operator ($\succ_n$). The total crowding distance of a solution is the sum of its individual objectives' distances which are computed as the absolute normalized differences between the solution and its closest neighbors. Then, the $N - |P_{t+1}|$ solutions with the greatest crowding distance are added to P_{t+1}. The process is iterated until a stopping criterion is met.

NEMO [2] is an interactive evolutionary hybrid which combines NSGA-II with IMO approaches based on the principle of ordinal regression. Alike NSGA-II, NEMO uses the Pareto fronts as a primary criterion to rank individuals. The major innovation consists in asking the DM at regular intervals to compare a single pair of solutions (note that a set A is composed of solutions from the current population). The accumulated preference information is used to select a representative additive general value function U^R [6]. Then, the solutions within each Pareto front are ranked using a representative value comparison operator $\succ_{U^R}$ (the greater $U^R(a)$, the better the solution a). Algorithm 1 describes the use of NEMO for the t-th generation.

4 Using Ordinal Regression for Interactive Evolutionary Multiple Objective Optimization Group Decision

In this section, we propose a few approaches for interactive evolutionary multiple objective optimization incorporating preference information of several DMs.

Algorithm 1. A single NEMO iteration for constructing the t-th generation

```
R_t = P_t ∪ Q_t
if Time to ask the DM {not conducted in NSGA-II} then
   Elicit DM's preferences {present to the DM a pair of non-dominated solutions and
   ask for a preference comparison}
   Determine the representative value function U^R
end if
F = fast-non-dominated-sort(R_t)
Within each non-dominance rank, sort individuals according to representative value
function {U^R replaces the crowding distance in NSGA-II}
P_{t+1} = ∅ and i = 1
while |P_{t+1}| + |F_i| ≤ N do
   representative-value(F_i) {instead of crowding-distance-assignment(F_i)}
   P_{t+1} = P_{t+1} ∪ F_i
   i = i + 1
end while
Sort(F_i, ≻_{U^R}) {instead of Sort(F_i, ≻_n) in NSGA-II}
P_{t+1} = P_{t+1} ∪ F_i[1 : (N − |P_{t+1}|)]
Q_{t+1} = make-new-pop(P_{t+1})

t = t + 1
```

Each of these approaches extends NEMO, originally designed for dealing with preferences of just a single DM. The scheme of this extension is common for all proposed variants with each DM being asked at regular intervals to compare a pair of randomly drawn solutions, and using the Pareto ranking as a primary criterion to rank individuals. The major differences concern a secondary criterion, and treating the population as a whole or evolving its parts individually for each DM. In particular, we propose the following variants of the **NEMO-GROUP** method:

- **NEMO-G1** determines a representative value function U_k^R for each $DM_k \in \mathcal{D}$ based on her/his pairwise comparisons only, and then ranks subsets of Pareto fronts according to their comprehensive values $U_{G1}^R(a) = \sum_{k=1}^{s} U_k^R(a)$;
- **NEMO-G2** determines U_k^R for each $DM_k \in \mathcal{D}$, and ranks subsets of Pareto fronts using $U_{G2}^R(a) = min_{k=1,\ldots,s} U_k^R(a)$;
- **NEMO-G3** maximizes the minimal number of pairwise comparisons of any $DM_k \in \mathcal{D}$ that can be represented together by an additive value function, determines the representative value function $U_{G3}^R(a)$ compatible with the consistent pairwise comparisons of all DMs (equal number of comparisons provided by each DM), and uses it to rank subsets of Pareto fronts;
- **NEMO-G4** divides a population into $1/s$ equal sub-populations, one for each $DM_k \in \mathcal{D}$, and evolves them separately using a representative value function $U_k^R(a)$ of each $DM_k \in \mathcal{D}$; a final population is obtained by combining together sub-populations of all DMs.

milosz.kadzinski@cs.put.poznan.pl

5 Experimental Results

To study the performance of different variants of NEMO-GROUP, we use ZDT1 and DTLZ2 with two (2D) and five (5D) objectives, respectively. We use artificial DMs who apply a pre-defined individual value functions for comparing pairs of solution whenever preference elicitation is conducted. Precisely, we use the linear functions, so the goal of each $DM_k \in \mathcal{D}$ is to minimize $U_k^{TRUE-LIN}(a) = \sum_{j=1}^{n} w_j^k g_j(a)$, where w_j^k, $j = 1, \ldots, n$, are weights of the n cost-type objectives. Thus, the whole group aims at minimizing $U_{\mathcal{D}}(a) = 1/s \sum_{k=1}^{s} U_k^{TRUE-LIN}(a)$. All these individual functions are unknown to the NEMO-GROUP algorithms, which instead use an additive value function defined in Sect. 2 as an internal preference model.

In our tests, we use a real-valued representation. We generate offspring by simulated binary crossover with probability of 0.9 and $\kappa_c = 1$, whereas mating selection is performed by tournament selection. We also apply Gaussian mutation with probability of 1/30. The population size is set to 60, and all methods are run for 500 generations.

5.1 Illustrative Examples

In this subsection, we use ZDT1-2D for an initial graphical comparison of the proposed approaches. We assume that preference elicitation is performed every 10 generation, and that DMs' value functions are parameterized with the following weights (w_1^k, w_2^k) for $k = 1, 2, 3, 4$: $DM_1 - (0.7, 0.3)$, $DM_2 - (0.4, 0.6)$, $DM_3 - (0.5, 0.5)$, and $DM_4 - (0.3, 0.7)$. Intuitively, the greater the weight, the more important it is to minimize the respective objective.

Figure 1 shows the results for different variants of NEMO-GROUP for three DMs ($DM_1 - DM_3$). To demonstrate the convergence to the Pareto front, for NEMO-G1, NEMO-G4, and NSGA-II, we depict populations obtained after 100, 300, and 500 generations. For clarity, for NEMO-G2 and NEMO-G3, we provide results only after 500 generations.

As can be seen, NSGA-II approximates the whole Pareto front, whereas all variants of NEMO-GROUP are focused on the solutions preferred to the DMs. For algorithms using an aggregated group value function as a secondary criterion (i.e., NEMO-G1, NEMO-G2, and NEMO-G3), the final population is narrowed to a single small part of the Pareto front composed of solutions which can be seen as the best compromise for all DMs. However, each of these approaches convergences to a slightly different region of the Pareto front (see the top-right part of Fig. 1). Typically, the populations constructed by NEMO-G1 and NEMO-G2 are closer to each other when compared with the population constructed by NEMO-G3. Indeed, NEMO-G1 and NEMO-G2 employ the group value functions aggregating the same DMs' individual representative value functions, though in a slightly different way, whereas NEMO-G3 constructs a single representative value function which is common for all DMs. Finally, since NEMO-G4 evolves a separate sub-population for each DM, the final population in our illustrative study

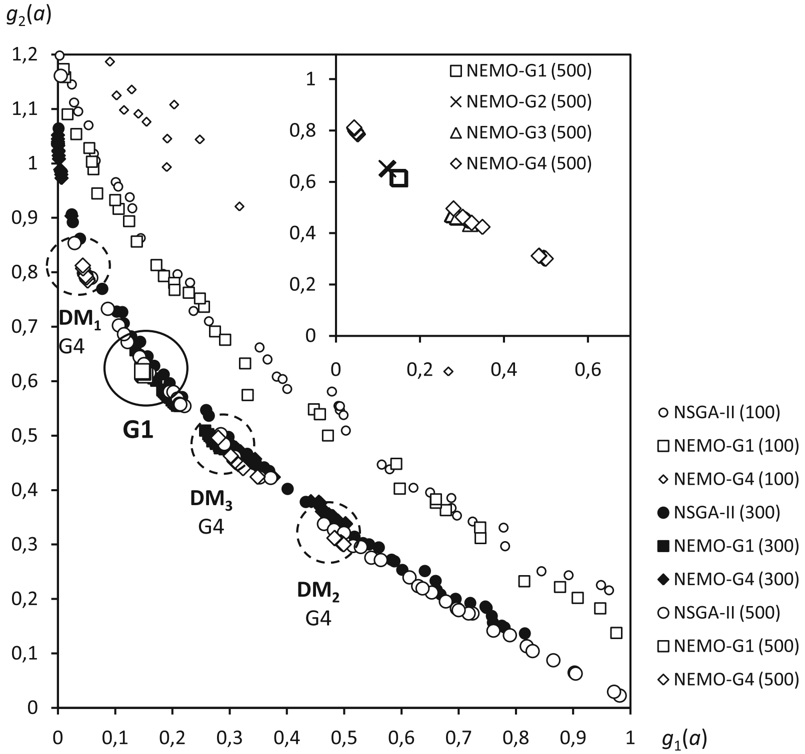

Fig. 1. Exemplary results of NEMO-G1, NEMO-G4, and NSGA-II on ZDT1-2D with three decision makers after 100, 300, and 500 generations. The results for NEMO-G2 and NEMO-G3 after 500 generation are provided in the top-right corner.

is composed of three clearly disjoint sets of solutions. These sub-populations contain solutions which are most preferred to a particular DM.

Such characteristic performance of different variants of NEMO-GROUP is confirmed by analogous results in the context of four DMs ($DM_1 - DM_4$; see Fig. 2). Obviously, NEMO-G1, NEMO-G2, and NEMO-G3 converge to different regions of the Pareto front than in case of three DMs, whereas NEMO-G4 approximates fourth sub-population with solutions more oriented to minimization of the second objective (as indicated by the weights for DM_4 - $(0.3, 0.7)$).

5.2 Convergence in Terms of a Utilitarian Value of the Solutions

In this subsection, we study the evolution of a utilitarian value for the best-of-population and average-in-population solutions in successive generations. These convergence factors permit to assess the performance of different variants of NEMO-GROUP from the point of view of a whole group of DMs. On the one hand, the best solution in the returned population may be perceived as a default outcome of the method that is most likely to be accepted by the DMs. On the

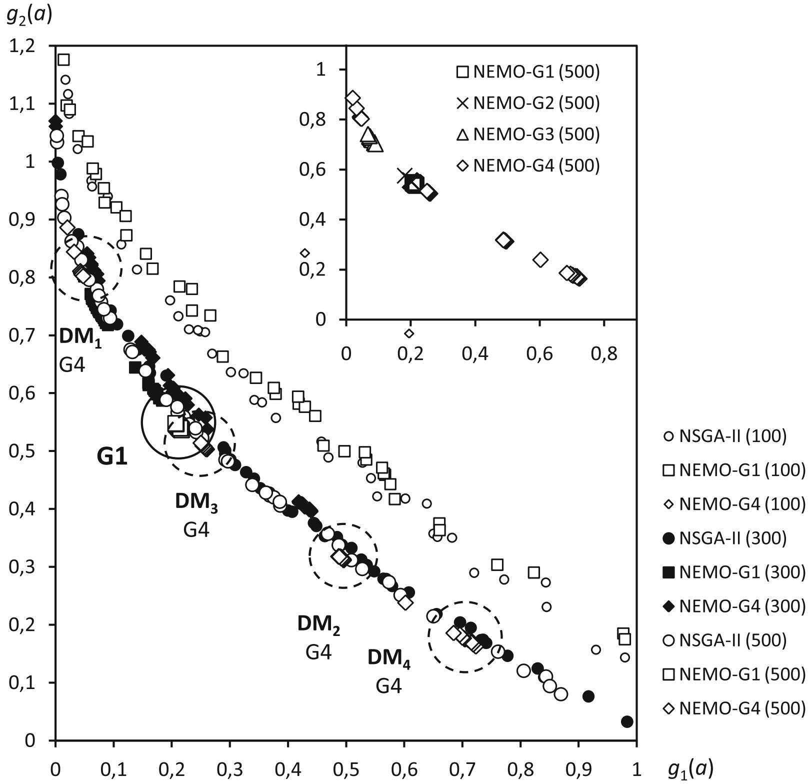

Fig. 2. Exemplary results of NEMO-G1, NEMO-G4, and NSGA-II on ZDT1-2D with four decision makers after 100, 300, and 500 generations. The results for NEMO-G2 and NEMO-G3 after 500 generations are provided in the top-right corner.

other hand, an average quality of the individuals contained in the population reveals if the search has been appropriately focused on the group consensus solutions [2].

All results presented in this section have been averaged over 100 independent runs, each for different weight vectors for the DMs' value functions. Figures 3 and 4 present the convergence plots for, respectively, value of the best solution and average value of all solutions in the population for ZDT1-2D with three DMs. The top-right parts of the figures depict the convergence plots for NEMO-G1 and NEMO-G4 starting from the first generation, whereas the main parts of the figures demonstrate the convergence between 100 and 500 generation for all considered algorithms. In this way, we can better illustrate when different approaches start to converge towards the Pareto front, what is their convergence speed measured in terms of a change of a utilitarian value, and what is value of the solution(s) at which their performance stabilizes.

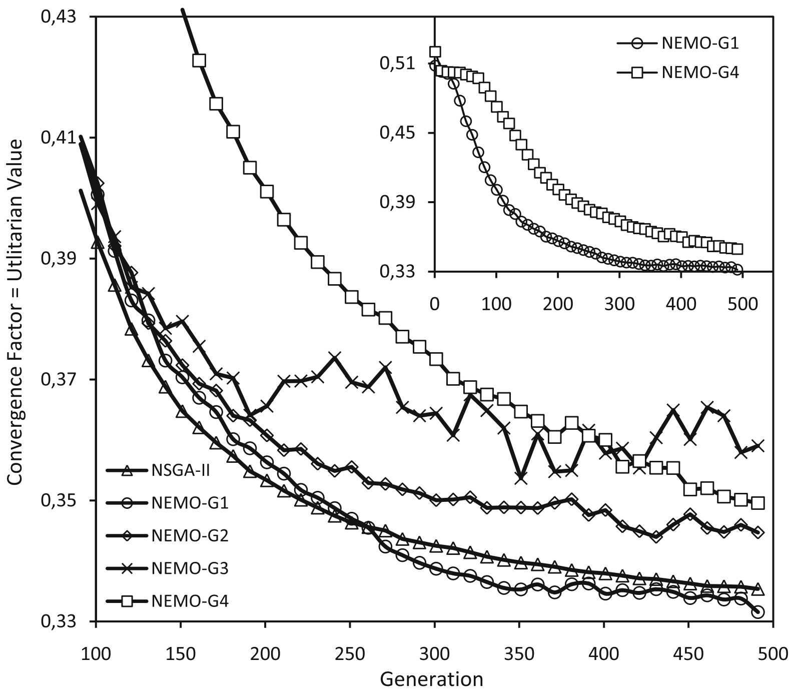

Fig. 3. Utilitarian value of the best-of-population solution in successive generations of NEMO-G1, NEMO-G2, NEMO-G3, and NEMO-G4 for three decision makers, and NSGA-II applied to ZDT1-2D (results averaged over 100 runs).

To compare the performance of proposed approaches for various benchmark problems and different numbers of DMs, we focus on the precise measures derived from the convergence plots. In this section, when presenting the experimental results in a tabular form, the text in bold and italics indicates the best performing algorithm across all 100 optimization runs. Additionally, we indicate in bold these approaches whose distance from the best performer proved to be statistically insignificant according to a Mann-Whitney-U test with 5 % significance level.

First, we refer to the minimal values obtained throughout the 500 generations (see Table 1). When it comes to the best utilitarian solution obtained during the optimization run (i.e., the minimal best-of-population value) for ZDT1-2D, NEMO-G1 and NEMO-G2 are most advantageous, while NEMO-G3 and NEMO-G4 are significantly worse than other approaches. Surprisingly, the value of the best solution discovered by NSGA-II is only slightly worse than that of the best performing variants of NEMO-GROUP. For DTLZ2-5D, NEMO-G4 and NEMO-G1 outperform other algorithms (except for the case with 2 DMs where the differences are statistically insignificant). Moreover, when moving to five dimensions, the best solution discovered by NSGA-II is much worse than in case of all NEMO-GROUP variants.

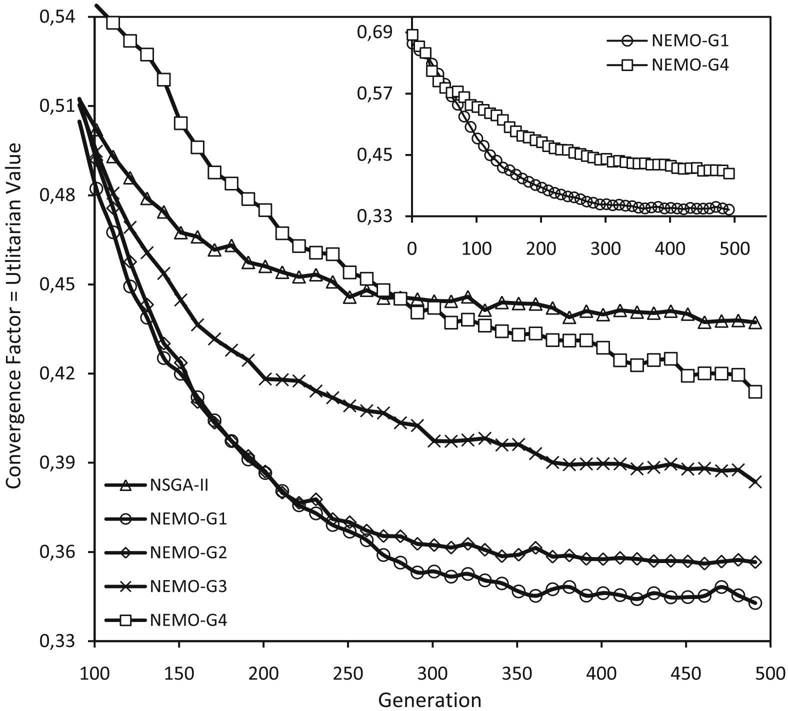

Fig. 4. Utilitarian value of the average-in-population solution in successive generations of NEMO-G1, NEMO-G2, NEMO-G3, and NEMO-G4 for three decision makers, and NSGA-II applied to ZDT1-2D (results averaged over 100 runs).

When comprehensively judging the returned population of solutions being most favorable from the point of view of the whole group (i.e., the best average-in-population value), NEMO-G1 performs the best for all considered problems and numbers of DMs except DTLZ2-5D and 3 DMs. However, when compared with NEMO-G2 or NEMO-G3, its advantage is statistically insignificant for some tested configurations. Furthermore, among all variants of NEMO-GROUP, NEMO-G4 proves to be the worst for all considered settings except DTLZ2-5D and 5 DMs. Finally, NSGA-II is significantly worse than NEMO-G1, NEMO-G2, and NEMO-G3, which construct only solutions which are relevant from the point of view of the whole group. This difference is particularly visible for a higher dimensional DTLZ2-5D.

As the other set of measures derived from the convergence plots we consider the average group utilitarian values observed throughout 500 generations. In this way, we are able to judge the overall performance of the algorithms from the point of view of either the best solution or a complete population returned after each generation. Since the value to which the algorithms converge highly affects the overall performance, the conclusions about the best and worst performing

Table 1. Minimal (best) value throughout 500 generations (results averaged over 100 runs; SD = standard deviation).

	2 DMs		3 DMs		4 DMs		5 DMs	
	$U_{\mathcal{D}}$	SD	$U_{\mathcal{D}}$	SD	$U_{\mathcal{D}}$	SD	$U_{\mathcal{D}}$	SD
Approach	Best-of-population value for ZDT1-2D							
NSGA-II	0.3009	0.0978	0.3352	0.0662	0.3574	0.0491	0.3616	0.0435
NEMO-G1	***0.2971***	0.1006	***0.3324***	0.0703	***0.3572***	0.0569	***0.3582***	0.0455
NEMO-G2	**0.2994**	0.1008	**0.3451**	0.0742	0.3661	0.0559	0.3691	0.0484
NEMO-G3	0.3121	0.1126	0.3567	0.0888	0.3892	0.0691	0.3991	0.0701
NEMO-G4	0.3052	0.1059	0.3467	0.0753	0.3816	0.0600	0.3979	0.0574
	Average-in-population value for ZDT1-2D							
NSGA-II	0.4372	0.0189	0.4372	0.0154	0.4392	0.0107	0.4370	0.0109
NEMO-G1	***0.3130***	0.1025	***0.3410***	0.0709	***0.3666***	0.0543	***0.3686***	0.0456
NEMO-G2	**0.3148**	0.0950	0.3551	0.0752	0.3802	0.0527	0.3816	0.0519
NEMO-G3	0.3321	0.1118	0.3838	0.1044	0.4096	0.0788	0.4232	0.0782
NEMO-G4	0.3528	0.1124	0.4150	0.0842	0.4490	0.0579	0.4659	0.0500
	Best-of-population value for DTLZ2-5D							
NSGA-II	0.1392	0.0531	0.1547	0.0552	0.1619	0.0451	0.1702	0.0463
NEMO-G1	**0.0943**	0.0528	**0.1097**	0.0520	**0.1279**	0.0614	**0.1239**	0.0442
NEMO-G2	**0.0919**	0.0608	0.1199	0.0731	0.1338	0.0664	0.1349	0.0543
NEMO-G3	**0.0954**	0.0600	0.1163	0.0546	0.1282	0.0538	0.1516	0.0693
NEMO-G4	***0.0887***	0.0533	***0.1045***	0.0458	***0.1102***	0.0366	***0.1194***	0.0388
	Average-in-population value for DTLZ2-5D							
NSGA-II	0.3932	0.0375	0.3906	0.0268	0.3915	0.0258	0.3872	0.0288
NEMO-G1	***0.1259***	0.0742	**0.1603**	0.0769	***0.1646***	0.0734	***0.1607***	0.0539
NEMO-G2	***0.1259***	0.0754	**0.1548**	0.0803	**0.1723**	0.0748	**0.1749**	0.0699
NEMO-G3	0.1346	0.0743	***0.1509***	0.0657	**0.1721**	0.0699	0.1951	0.0747
NEMO-G4	0.1411	0.0648	0.1683	0.0543	0.1769	0.0411	0.1863	0.0423

algorithms are analogous to the case of considering only the best results. The important differences are the following:

- NEMO-G4 performs poorly for ZDT1-2D, because it starts to converge later than other algorithms.
- NSGA-II and NEMO-G3 are even less advantageous for DTLZ2-5D than in case of considering the best results only, because their convergence curves are more erratic than the others, deteriorating several times in the phase when performance of other algorithms stabilizes or still slightly improves.
- NEMO-G2 shows clear advantages with respect to NEMO-G1 in terms of average-in-population value, because in the initial generations the performance of both algorithms is very similar and NEMO-G1 derives its comprehensive superiority from exploiting more favorable search directions only after 250 generation (Table 2).

Table 2. Average value throughout 500 generations (results averaged over 100 runs; SD = standard deviation).

	2 DMs		3 DMs		4 DMs		5 DMs	
	$U_{\mathcal{D}}$	SD	$U_{\mathcal{D}}$	SD	$U_{\mathcal{D}}$	SD	$U_{\mathcal{D}}$	SD
Approach	Best-of-population value for ZDT1-2D							
NSGA-II	0.3370	0.0997	***0.3693***	0.0706	***0.3910***	0.0530	***0.3951***	0.0479
NEMO-G1	***0.3367***	0.1032	**0.3701**	0.0733	**0.3913**	0.0558	**0.3959**	0.0495
NEMO-G2	**0.3368**	0.1029	**0.3772**	0.0765	**0.3991**	0.0578	**0.4027**	0.0513
NEMO-G3	0.3459	0.1081	0.3858	0.0799	0.4108	0.0513	0.4189	0.0453
NEMO-G4	0.3608	0.1079	0.4069	0.0840	0.4399	0.0677	0.4566	0.0675
	Average-in-population value for ZDT1-2D							
NSGA-II	0.4780	0.0320	0.4779	0.0239	0.4783	0.0182	0.4785	0.0171
NEMO-G1	0.3882	0.0897	***0.4142***	0.0634	***0.4323***	0.0501	***0.4359***	0.0436
NEMO-G2	***0.3846***	0.0866	**0.4207**	0.0696	**0.4394**	0.0576	**0.4409**	0.0521
NEMO-G3	0.3995	0.0973	0.4455	0.0815	0.4649	0.0528	0.4773	0.0426
NEMO-G4	0.4315	0.1016	0.4832	0.0778	0.5097	0.0570	0.5233	0.0551
	Best-of-population value for DTLZ2-5D							
NSGA-II	0.1357	0.0499	0.1551	0.0418	0.1625	0.0372	0.1695	0.0390
NEMO-G1	***0.0966***	0.0477	***0.1128***	0.0444	**0.1263**	0.0416	***0.1271***	0.0368
NEMO-G2	**0.0974**	0.0511	0.1173	0.0448	**0.1269**	0.0405	0.1325	0.0375
NEMO-G3	**0.0984**	0.0488	0.1235	0.0413	0.1343	0.0397	0.1468	0.0432
NEMO-G4	**0.0995**	0.0503	**0.1140**	0.0416	***0.1227***	0.0335	0.1331	0.0365
	Average-in-population value for DTLZ2-5D							
NSGA-II	0.3935	0.0177	0.3940	0.0135	0.3940	0.0128	0.3944	0.0118
NEMO-G1	***0.1449***	0.0503	***0.1640***	0.0484	***0.1755***	0.0406	***0.1741***	0.0343
NEMO-G2	**0.1460**	0.0517	**0.1682**	0.0444	**0.1799**	0.0407	0.1828	0.0389
NEMO-G3	0.1573	0.0527	0.1883	0.0477	0.2050	0.0482	0.2195	0.0504
NEMO-G4	0.1681	0.0580	0.1818	0.0480	0.1938	0.0399	0.2016	0.0415

6 Conclusions and Future Research

In this paper, we presented an interactive evolutionary multiple objective optimization method incorporating preference information of several decision makers into the evolutionary search. After recalling NEMO designed for interaction with a single decision maker, we extended it in different ways to group decision. In all proposed variants of NEMO-GROUP, the user interaction is based on ordinal regression. The main differences between these variants concern construction of an elitist function indicating values that solutions represent to the group members.

The experimental results confirm that the proposed approaches are able to focus the search on the group-preferred solutions. Nevertheless, they indicate that the proposed approaches differ with respect to both part of the Pareto front to which they converge as well as the convergence speed measured in terms of a change of a utilitarian value of the returned solutions.

We envisage the following developments:

- employing different procedures for selecting a single representative value function within ordinal regression (e.g., optimizing the misranking error ε in case of incompatibility) and using methods derived from Robust Ordinal Regression [5] for preserving elitism;
- verifying how different values of elicitation interval influence the convergence speed of the algorithms;
- testing the proposed approaches more thoroughly on a set of benchmark problems with different dimensions, various forms of the assumed DMs' true value functions, and different group value functions (e.g., egalitarian or elitist ones).

Acknowledgments. The authors acknowledge financial support from the Poznan University of Technology, grant DS-MLODA KADRA (2015).

References

1. Poles, S., Vassileva, M., Sasaki, D.: Multiobjective Optimization Software. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization. LNCS, vol. 5252, pp. 329–348. Springer, Heidelberg (2008)
2. Branke, J., Greco, S., Słowiński, R., Zielniewicz, P.: Learning value functions in interactive evolutionary multiobjective optimization. IEEE Trans. Evol. Comput. **19**(1), 88–102 (2015)
3. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
4. Jacquet-Lagrèze, E., Siskos, Y.: Preference disaggregation: 20 years of MCDA experience. Eur. J. Oper. Res. **130**(2), 233–245 (2001)
5. Corrente, S., Greco, S., Kadziński, M., Słowiński, R.: Robust ordinal regression in preference learning and ranking. Mach. Learn. **93**(2–3), 381–422 (2013)
6. Kadziński, M., Greco, S., Słowiński, R.: Selection of a representative value function in robust multiple criteria ranking and choice. Eur. J. Oper. Res. **217**(3), 541–553 (2012)
7. Kadziński, M., Greco, S., Słowiński, R.: Selection of a representative value function for robust ordinal regression in group decision making. Group Decis. Negot. **22**(3), 429–462 (2013)