An Artificial Decision Maker for Comparing Reference Point Based Interactive Evolutionary Multiobjective Optimization Methods

Bekir Afsar[1] [0000−0003−3643−2342], Kaisa Miettinen[1] [0000−0003−1013−4689], and Ana B. Ruiz[2] [0000−0003−0543−8055]

[1] University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland
bekir.b.afsar@jyu.fi, kaisa.miettinen@jyu.fi

[2] Department of Applied Economics (Mathematics), Universidad de Málaga, C/ Ejido 6, 29071, Málaga, Spain, abruiz@uma.es

Abstract. Comparing interactive evolutionary multiobjective optimization methods is controversial. The main difficulties come from features inherent to interactive solution processes involving real decision makers. The human can be replaced by an artificial decision maker (ADM) to evaluate methods quantitatively. We propose a new ADM to compare reference point based interactive evolutionary methods, where reference points are generated in different ways for the different phases of the solution process. In the learning phase, the ADM explores different parts of the objective space to gain insight about the problem and to identify a region of interest, which is studied more closely in the decision phase. We demonstrate the ADM by comparing interactive versions of RVEA and NSGA-III on benchmark problems with up to 9 objectives. The experiments show that our ADM is efficient and allows repetitive testing to compare interactive evolutionary methods in a meaningful way.

Keywords: Decision making · Aspiration levels · Performance comparison · Many-objective optimization · Interactive methods.

1 Introduction

Many real-world applications involve optimizing a set of conflicting objectives over a set of feasible solutions. This type of problems are known as *multiobjective optimization problems* (MOPs). Typically, no solution exists optimizing all the objectives at the same time, and we look for so-called Pareto optimal solutions, at which an improvement of any objective always implies a sacrifice in, at least, one of the others. Mathematically, Pareto optimal solutions are incomparable, and we need information about the preferences of a *decision maker* (DM), an expert in the problem domain, to identify the *most preferred solution* (MPS).

In interactive methods [15, 16], the DM plays an active role in directing a solution process. (S)he sees information about solutions available, and expresses and possibly changes her/his preferences in an iterative process, which ends once the DM is satisfied. Often, the interactive solution process can be divided into two phases [16]. In the learning phase, the DM explores different solutions to find a *region of interest* (ROI), i.e., solutions suiting her/his preferences. Then, in the decision phase, (s)he fine-tunes the search in this ROI to find her/his MPS.

In most interactive evolutionary multiobjective optimization (EMO) methods, preferences are used to approximate a ROI [2, 14, 21]. Comparing interactive EMO methods is important to reveal the strengths and weaknesses of new proposals against old ones but the quantitative assessment of methods involving DMs is still open [13]. The DM's preferences evolve while (s)he learns about the trade-offs in the problem and the feasibility of one's preferences in the interactive solution process. Therefore, the subjectivity of real DMs, human fatigue, or other limiting factors make it hard to design experiments with human DMs.

Alternatively, the human DM can be replaced somehow. In this regard, interactive methods can be divided into non ad-hoc and ad-hoc methods [19], depending on whether the DM can be replaced by a value function or not, respectively. Here we focus on ad-hoc interactive EMO methods. To compare such methods, we can use an *artificial DM* (ADM) simulating the DM's actions. Comparing methods with ADMs is cheaper and less time consuming than with human DMs. Some literature of ADMs already exists. To generate reference points, the ADMs developed in [1, 18] have a pre-defined steady part that does not change during the solution process and current context that evolves. In [10], a cone is used based on a pre-defined MPS, and the learning of a DM is simulated by narrowing the cone's angle in the solution process. These ADMs need a goal point (initial aspiration levels or a MPS, respectively) to be reached by the ADM. However, this may hinder the exploration of the search and, thus, performances depend on that point. Furthermore, algorithms are run individually, i.e., preferences are generated based on the output of each single algorithm.

We propose an ADM for comparing the performance of interactive EMO methods based on reference points. By performing iterations, our ADM adjusts the preferences according to the insight gained during the solution process. At each iteration, a reference point is generated based on the solutions obtained so far by all compared algorithms. To simulate varying search objectives in different phases, the ADM produces reference points in a different way for the learning and the decision phases. In the learning phase, the reference points simulate exploration to examine the whole set of Pareto optimal solutions in search for a potential ROI. Then, the reference points of the decision phase mimic a progressive convergence towards a MPS inside the previously identified ROI. To compare the performances of the algorithms, the obtained solutions are evaluated according to the reference point generated at each iteration.

For comparability, the ADM assigns the same computational resources (i.e. number of evaluations or generations per iteration) to all algorithms compared. Note that the quality of the results depends on the algorithms, not on the ADM. Since our ADM allows assigning a different pre-fixed number of iterations to the learning and decision phases, it can be applied for different needs. E.g., more iterations can be performed in the learning phase to evaluate the search adaptation capacity when the preferences change drastically. This simulates an exploratory behaviour to gain knowledge about the conflicting objectives or feasible solutions. To compare the ability to search within a specific region to fine-tune solutions, one can increase the number of iterations of the decision phase in the ADM.

In the literature, few works analyze practical properties inherent to interactive solution processes. The reason may be the lack of a framework to simulate the iterative process followed by a DM. Our ADM is aimed at comparing interactive reference point-based EMO methods. However, we do not consider human factors related to decision making, such as cognitive biases like anchoring, which deserve further research. The proposed ADM contributes to existing research by generating reference points in a different manner for the learning and decision phases. To our best knowledge, this is the first ADM to differentiate two phases explicitly. It runs algorithms simultaneously and uses their output to generate the reference points, without requiring a pre-defined goal point, as in [1, 10, 18].

In the following, Section 2 introduces the basic concepts and notation needed. The new ADM is described in Section 3. Next, Section 4 demonstrates the performance of our proposal. Finally, conclusions are drawn in Section 5.

2 Background

A *multiobjective optimization problem* that minimizes k (with $k \geq 2$) conflicting objective functions $f_i : S \to \mathbb{R}$ $(i = 1, \ldots, k)$ can be formulated as follows:

$$\begin{array}{ll} \min & \{f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})\} \\ \text{s.t.} & \mathbf{x} \in S, \end{array} \tag{1}$$

where $S \subset \mathbb{R}^n$ is the *feasible set* of *decision vectors* $\mathbf{x} = (x_1, \ldots, x_n)^T$ in the decision space. Corresponding *objective vectors* of the form $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T$ constitute a *feasible objective region* $Z = \mathbf{f}(S) \subset \mathbb{R}^k$ in the objective space.

Because of the conflicting nature of the objectives, a single solution optimizing all of them at the same time does not exist. On the contrary, there is a set of *Pareto optimal* solutions, at which none of the objectives can be improved without deteriorating, at least, one of the others. Given $\mathbf{z}, \mathbf{z}' \in \mathbb{R}^k$, we say that $\mathbf{z}$ *dominates* $\mathbf{z}'$ if $z_i \leq z'_i$ for all $i = 1, \ldots, k$ and $z_j < z'_j$ for at least one index j. If $\mathbf{z}$ and $\mathbf{z}'$ do not dominate each other, we say that they are *(mutually) nondominated*. Then, $\mathbf{x} \in S$ is *Pareto optimal* if there does not exist any $\mathbf{x}' \in S$ such that $\mathbf{f}(\mathbf{x}')$ dominates $\mathbf{f}(\mathbf{x})$. The corresponding objective vector $\mathbf{f}(\mathbf{x}) \in Z$ is called a *Pareto optimal objective vector*. All Pareto optimal solutions form a *Pareto optimal set* in the decision space, denoted by E. Its image in the objective space is known as a *Pareto optimal front* (PF), denoted by $\mathbf{f}(E)$.

The ranges of the objective functions defined by their worst and best possible values in the PF form so-called nadir and ideal points, respectively. The upper bounds define the *nadir point* $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, \ldots, z_k^{\text{nad}})^T$, i.e., $z_i^{\text{nad}} = \max_{\mathbf{x} \in E} f_i(\mathbf{x})$ $(i = 1, \ldots, k)$, while the lower bounds constitute an *ideal point* $\mathbf{z}^\star = (z_1^\star, \ldots, z_k^\star)^T$ as $z_i^\star = \min_{\mathbf{x} \in E} f_i(\mathbf{x}) = \min_{\mathbf{x} \in S} f_i(\mathbf{x})$ $(i = 1, \ldots, k)$. In practice, the nadir point is difficult to calculate because the set E is unknown and, thus, $\mathbf{z}^{\mathrm{nad}}$ is commonly estimated (see e.g. [15, 20]).

We consider methods that include preference information in the form of a reference point $\mathbf{q} = (q_1, \ldots, q_k)^T$. Here, q_i is a so-called aspiration level, that is, a desirable value for the objective f_i provided by the DM ($i = 1, \ldots, k$).

3 Artificial Decision Maker

We propose an ADM to compare the performances of reference point based interactive EMO algorithms, which runs all algorithms simultaneously. To make a meaningful comparison, the ADM provides the same computational resources (i.e. number of function evaluations or generations per iteration) and produces reference points in a similar manner for all algorithms.

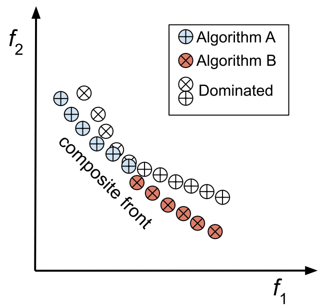
As mentioned before, we consider the two phases of an interactive solution process: the learning and the decision phases. Accordingly, our ADM has two strategies to produce reference points. In the learning phase, the ADM explores the objective space to see the solutions available by providing, at each iteration, a reference point in the least explored area of the PF. Eventually, a ROI is found at the end of the learning phase. In the decision phase, the ADM aims to refine solutions inside this ROI in search for a MPS, so it provides a reference point in this ROI at each iteration of this phase. At the beginning, a pre-fixed number of iterations is assigned to each phase. Respectively, we denote by L and D the number of iterations performed in the learning and in the decision phases.

The ADM uses solutions obtained so far from all algorithms to be compared. Thus, new reference points are generated according to the responses of the algorithms. For this, the ADM first merges the solutions and then eliminates dominated ones to build a *composite front*, as shown in Figure 1a. Thus, the composite front includes only the non-dominated solutions obtained by all algorithms from the first iteration until the current iteration. Note that the ADM does not need any true PF information to generate reference points.

Besides preference generation, the proposed ADM evaluates the responsiveness of the algorithms for the given reference point after each iteration by applying performance metrics that take into account the given reference point. Moreover, it calculates cumulative metric values to evaluate the performances of the algorithms separately in the learning and in the decision phases.

In the literature, decomposition-based EMO algorithms have been proposed to handle problems with $k > 3$. They usually decompose the original MOP into a group of sub-problems by dividing the whole PF into a group of subsets to enhance the performance of algorithms. We have adapted this idea in our ADM to find the exploration degree of the different parts of the whole PF (any other metric giving information about sub-areas of the PF can be also applied for this).

The following approach is used for finding areas of the composite front to be explored at each iteration of the learning phase. First, the composite front is divided equally into a group of subsets by using reference vectors which are uniformly distributed on the PF, as in [3, 4]. Therefore, the ADM creates a set



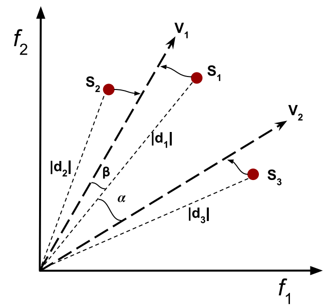


Fig. 1: (a) A composite front. (b) Solution assignment to reference vectors.

of uniformly distributed reference vectors using the canonical simplex-lattice design method [5]. In this, the number of reference vectors is adjusted by a lattice resolution (l), as $\binom{l+k-1}{k-1}$, where l is a pre-fixed parameter in the proposed ADM. With these vectors, the ADM assigns solutions in the composite front to reference vectors according to the angle between them. In the example in Figure 1b, we have two reference vectors (V_1, V_2) and three solutions (S_1, S_2, S_3). Since the angle (β) between S_1 and V_1 is smaller than the angle (α) between S_1 and V_2, solution S_1 is assigned to vector V_1. Similarly, each solution is assigned to the reference vector with the minimum angle (S_2 is assigned to V_1, S_3 is assigned to V_2). The number of assigned solutions to each reference vector gives us information about the exploration degree of different parts of the composite front. In this way, the ADM divides the whole composite front into sub-areas to be able to find the least explored area of the PF. Below, we describe how distances of the solutions to the ideal point are taken into account in generating a reference point. To summarize, the main steps of the proposed ADM are:

Step 0 Initialize all algorithms and provide the first reference point randomly.

Step 1 Run all algorithms with the same computational budget (number of generations or function evaluations) and the previously obtained reference point.

Step 2 Build (or update) the composite front using the solutions obtained by each algorithm until this iteration.

Step 3 Evaluate the algorithm performances taking the reference point into account.

Step 4 Generate a new reference point for the next iteration based on the composite front and the phase of the solution process:

- **a)** In the learning phase, find the least explored area of the composite front and then, generate the next reference point for that area.
- **b)** In the decision phase, generate the new reference point in the ROI identified at the end of the learning phase to fine-tune solutions.

Step 5 If a termination criterion is met, terminate the process and calculate cumulative metric values for each phase. Else, continue with Step 1.

Preference generation in the learning phase: In this phase, our ADM explores potential regions of the PF to examine those that have been poorly covered so far. Thus, the ADM tries to localize unexplored areas of the composite front and find more solutions in the least explored area by providing the next reference point inside it. As mentioned, a set of uniformly distributed reference vectors is generated on the composite front first. Then, all solutions are assigned to reference vectors as previously described, and the least explored area is identified based on the reference vector which has the minimum number of assigned solutions. Out of the four reference vectors shown in Figure 2a, V_2 is selected.

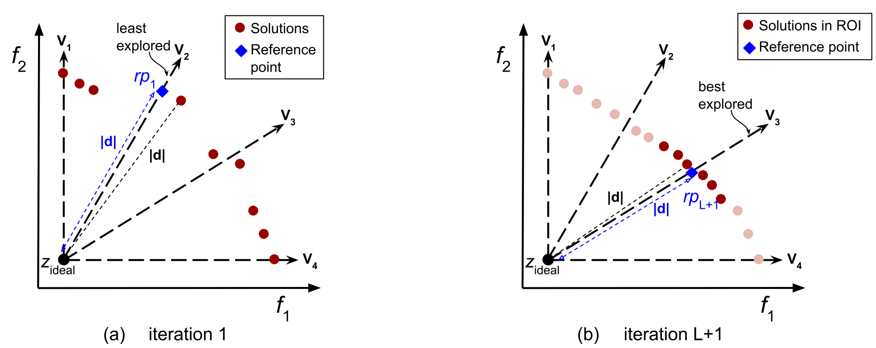


Fig. 2: Preference generation in (a) the learning phase and (b) the decision phase.

The location of the reference point on the selected reference vector is determined by using solutions assigned to it. To this end, the ADM calculates the distances of these solutions to the ideal point of the current composite front, and selects the solution with the minimum distance. This distance $|d|$ sets the next reference point on the selected reference vector, as shown in Figure 2a.

After L iterations, the reference vector that has the maximum number of solutions is identified. We denote it as V_D, and solutions assigned to it constitute the ROI to be further studied in the decision phase.

Preference generation in the decision phase: In the D iterations of the decision phase, the reference vector V_D is employed to generate reference points in order to get progressively closer to the PF by refining solutions in the ROI.

At each iteration, our ADM finds the solution in the ROI with the minimum distance $|d|$ to the ideal point of the composite front, and $|d|$ is used to generate the next reference point as shown in Figure 2b. All the reference points generated in this phase lie on the reference vector V_D. With iterations, the new reference points get closer to the ideal point of the composite front, given that the solutions generated by the algorithms (used to update the composite front) converge to the PF. With this, the ADM performs a finer search in the ROI to find a MPS.

Evaluation: If no preference information is taken into account in the solution process, having a sufficient number of solutions, good closeness to the PF (convergence), good spread over the PF, and good uniformity amongst solutions define the quality of the obtained solutions [12]. To evaluate and compare the performances of reference point based EMO methods, one should redefine the term 'quality' of the solutions. Since preference information is used to guide the solution process, the above features should be measured for the preferred ROI, which is defined in our case by a reference point, instead of for the whole PF. However, to evaluate interactive processes, aspects related to the interaction with the DM should be quantified, but to the best of our knowledge, there do not exist quality indicators for assessing the performance of interactive methods. Therefore, the proposed ADM analyses the performance of each interactive method at each iteration using metrics developed for reference point based EMO methods, where preferences are provided a priori, before the solution process.

Once the ADM gets the solution sets produced by the algorithms at each iteration, performance metrics (denoted by m_i) are calculated for each set. At the end of the solution process, the ADM finds cumulative metric values for each phase, to evaluate the performances of the algorithms depending on the needs of each phase. For the learning phase, the metric values are considered until iteration L as $\sum_{i=1}^{L} m_i$, and for the decision phase, from iteration $L+1$ until the termination of the algorithm as $\sum_{i=L+1}^{L+D} m_i$.

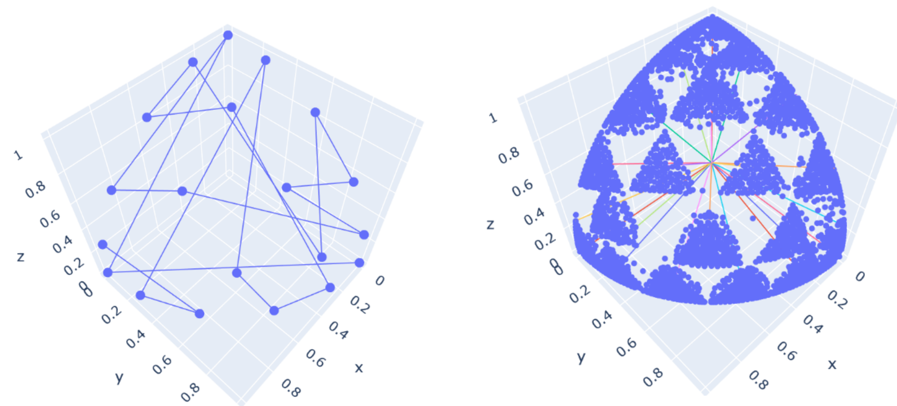
In the literature, some performance metrics for a priori EMO methods have been proposed (see e.g., [9, 11, 17, 22]). Any of these metrics could be utilized here. We use the R-metric [11] to measure the responsiveness of each algorithm for the provided reference points. The R-metric applies regular performance metrics (e.g., IGD) for the solutions in the ROI that is defined by the reference point. The size of the ROI is controlled by a parameter Δ. We employ the R-IGD (using IGD) since it is computationally efficient for a high number of objectives and it measures both convergence and diversity of solutions. The lower the R-IGD value, the better is the quality of the solutions of an algorithm in the ROI.

4 Experimental studies

To demonstrate the applicability and usefulness of the proposed ADM, interactive versions of the EMO algorithms RVEA [3] and NSGA-III [6] are compared and we refer to them as iRVEA and iNSGA-III, respectively. The iRVEA algorithm is described in [8] and iNSGA-III was made interactive in a corresponding way. Their implementations as well as the proposed ADM are available in the open source DESDEO framework (https://desdeo.it.jyu.fi) in Python.

4.1 Search behaviour of the ADM

To have a better understanding of the proposed ADM, we first illustrate its behaviour in the learning and decision phases, respectively. For this purpose, we



(a) Generated reference points. (b) Composite front.

Fig. 3: Search behaviour of the ADM in the learning phase.

conducted two experiments, one for each phase. The ADM iterated iRVEA and iNSGA-III for 20 times, using 50 generations per iteration for both algorithms.

Figure 3a illustrates the generated reference points as dots and Figure 3b contains the composite front obtained by the algorithms after 20 iterations for the three-objective DTLZ2 problem [7]. As shown, the ADM explores the whole objective space. In Figure 3a, the continuous line shows the search path taken by the ADM. After each iteration, the ADM found the least explored area, where the next reference point was generated. Figure 3b shows the composite front obtained by the two algorithms.

In the second experiment, we studied the ADM in the decision phase. After obtaining the first (randomly generated) reference point, the ADM found the most explored area and generated next reference points there. Therefore, it converged in one direction to refine the solutions in the same area. Due to page limitations, we cannot visualize this here. See Figure S1 in the supplementary document (http://www.mit.jyu.fi/optgroup/extramaterial.html).

4.2 Experimental settings

We conducted experiments on the benchmark problems DTLZ1-4 [7] with the number of objectives (k) ranging from 3 to 9, resulting in 28 different problems. The number of variables was set as $10+k-1$ [7]. For each algorithm, we executed 50 and 100 generations per iteration. All other parameters of iRVEA and iNSGA-III were set as in [3] and [6], respectively. We set the ADM parameters as follows: the number of iterations L for the learning phase 4, the number of iterations D for the decision phase 3, the lattice resolution 5 and the parameter Δ for the R-metric 0.3 for the learning phase and 0.2 for the decision phase. We made 21 independent runs with the proposed ADM by using the same experimental settings. We assumed that the solution process starts with a learning phase and continues with a decision phase. However, this could be easily adapted if one prefers several phases in different orders, instead.

4.3 Numerical results

In Table 1, we give a concrete example of reference points generated by the ADM for one run with the three-objective DTLZ1 problem. As can be seen, the switch from one phase to another takes place between iterations 4 and 5. Additionally, the R-IGD metric values are listed per iteration to show the responses of algorithms after each iteration. As mentioned before, the cumulative metric values are eventually considered for both phases.

Table 1: Reference points and R-IGD values at each iteration for DTLZ1.

Iteration	Reference point	iRVEA	iNSGA-III
1	[0.9475, 0.9475, 2.8426]	6.00E-01	5.70E-01
2	[0.0000, 0.9417, 0.6278]	7.39E-01	7.38E-01
3	[0.0000, 0.0000, 0.5090]	5.52E-01	5.52E-01
4	[0.0000, 0.5067, 0.0000]	5.52E-01	5.52E-01
5	[0.0000, 0.2843, 0.1895]	7.64E-01	7.63E-01
6	[0.0000, 0.2684, 0.1789]	7.65E-01	7.62E-01
7	[0.0000, 0.2680, 0.1786]	7.64E-01	7.63E-01

Table 2 summarizes the cumulative R-IGD values of iRVEA and iNSGA-III for the problems considered with 4, 7 and 9 objectives and 50 generations per iteration. We report the phase of the solution process (learning, iterations 1–4, or decision, iterations 5–7), and the cumulative R-IGD metric values. The mean and the standard deviation of the metric values of 21 independent runs are presented and the best results are highlighted in bold.

An interesting observation can be made for DTLZ1 in Table 2. The R-IGD values of iNSGA-III are better than those of iRVEA in the learning phases for all numbers of objectives. This means that iNSGA-III responded better to the drastic changes of the provided preference by the ADM. In contrast, iRVEA outperformed iNSGA-III in the decision phases as the number of objectives increased. We conclude that iRVEA exploited better than iNSGA-III on DTLZ1.

For problems DTLZ2 and DTLZ4, iNSGA-III performed better than iRVEA for all numbers of objectives in both phases. A special case was seen for DTLZ3 with 4 objectives when iRVEA responded better in the learning phase, and iNSGA-III refined solutions better in the decision phase. The situation was the opposite for 7 objectives. Moreover, iRVEA results were better for both the learning and the decision phases for the 9-objective DTLZ3 problem. The results for other numbers of objectives can be found in Table S1 in the supplementary document. As mentioned, we also made the experiments by increasing the computational budget to 100 generations per iteration for both algorithms. We provide these results in Table S2 in the above-mentioned supplementary document.

Table 2: Cumulative R-IGD values for test problems with 4, 7 and 9 objectives.

Problem	k	Phase	iRVEA		iNSGA-III	
			mean	std. dev.	mean	std. dev.
DTLZ1	4	learning	2.74E+00	3.49E-01	**2.62E+00**	1.66E-01
		decision	2.03E+00	3.76E-01	**1.98E+00**	2.69E-01
	7	learning	3.30E+00	4.49E-01	**2.73E+00**	2.13E-01
		decision	**2.27E+00**	3.01E-01	2.38E+00	3.89E-01
	9	learning	3.07E+00	2.59E-01	**2.80E+00**	2.76E-01
		decision	**2.16E+00**	2.99E-01	2.46E+00	6.23E-01
DTLZ2	4	learning	4.71E-01	3.43E-01	**2.21E-01**	1.21E-01
		decision	5.73E-01	7.67E-01	**2.41E-01**	2.64E-01
	7	learning	8.28E-01	5.28E-01	**4.15E-01**	2.10E-01
		decision	1.47E+00	1.57E+00	**6.10E-01**	5.00E-01
	9	learning	9.68E-01	5.32E-01	**4.93E-01**	1.78E-01
		decision	1.23E+00	1.53E+00	**5.01E-01**	3.36E-01
DTLZ3	4	learning	**4.47E-01**	3.09E-01	5.66E-01	2.31E-01
		decision	1.42E-01	9.69E-02	**9.20E-02**	5.82E-02
	7	learning	8.91E-01	5.14E-01	**8.26E-01**	2.95E-01
		decision	**3.43E-01**	1.39E-01	7.50E-01	5.05E-01
	9	learning	**7.03E-01**	3.34E-01	1.02E+00	3.70E-01
		decision	**3.75E-01**	2.20E-01	9.87E-01	7.22E-01
DTLZ4	4	learning	3.60E-01	4.43E-01	**2.19E-01**	1.92E-01
		decision	3.68E-01	3.91E-01	**3.43E-01**	3.95E-01
	7	learning	3.01E+00	2.19E+00	**7.29E-01**	3.70E-01
		decision	1.18E+00	6.74E-01	**6.69E-01**	4.48E-01
	9	learning	8.25E-01	3.11E-01	**7.60E-01**	2.83E-01
		decision	8.26E-01	4.48E-01	**6.90E-01**	4.20E-01

We have shown that the proposed ADM enables us to study the suitability of each algorithm in the learning phase or in the decision phase. This type of information is hard, or even impossible, to get by just using a performance metric, specially if the metric is not formulated for interactive methods. Observe that the ADM has been designed to evaluate the search conducted by the algorithms distinguishing both phases, and it does not change the algorithms themselves. One should note that the findings of the analysis should not be generalized. The objective of this consideration was to demonstrate how the ADM can be applied, not to find any winner among the algorithms compared.

5 Conclusions

In this paper, an ADM was proposed for comparing reference point based interactive EMO methods. The ADM provides reference points in a different manner for the learning and the decision phases to reflect various objectives in the interactive solution process. In the learning phase, the ADM generates reference points to find more solutions in the least explored areas, while in the decision phase, it refines solutions inside the ROI identified. During the solution process, the ADM evaluates the performances of the algorithms by applying performance metrics that take into account the preference information and, at the end of the solution process, it calculates the metric values for each phase separately.

Experiments on benchmark problems were conducted to demonstrate how the ADM can be used to compare interactive EMO methods. To make the comparison meaningful, the same computational budgets were given to the algorithms compared and the R-IGD was used to evaluate their performances.

The proposed ADM has a modular structure and can be adapted for different needs. The comparison is meaningful since all algorithms to be compared apply the same preference information. In our future research, we intend to make it adaptive to decide automatically when to switch from the learning phase to the decision phase. We also plan to consider different types of preference information and develop quality indicators dedicated for interactive methods.

Acknowledgements

This research was partly funded by the Academy of Finland (grants 322221 and 311877) and is related to the thematic research area Decision Analytics utilizing Causal Models and Multiobjective Optimization (DEMO), jyu.fi/demo, at the University of Jyvaskyla. Ana B. Ruiz would like to thank the support of the Spanish government (project ECO2017-88883-R) and the Andalusian regional government (PAI group SEJ-532 and project UMA18-FEDERJA-024).

References

1. Barba-González, C., Ojalehto, V., García-Nieto, J.M., Nebro, A.J., Miettinen, K., Aldana-Montes, J.F.: Artificial decision maker driven by PSO: An approach for testing reference point based interactive methods. In: Auger, A., Fonseca, C.M., Lourenço, N., Machado, P., Paquete, L., Whitley, D. (eds.) Parallel Problem Solving from Nature - PPSN XV, 15th International Conference, Proceedings, Part I. pp. 274–285. Springer (2018)
2. Branke, J., Deb, K., Miettinen, K., Slowinski, R. (eds.): Multiobjective Optimization. Interactive and Evolutionary Approaches. Springer, Berlin, Heidelberg (2008)
3. Cheng, R., Jin, Y., Olhofer, M., Sendhoff, B.: A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Transactions on Evolutionary Computation **20**(5), 773–791 (2016)
4. Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., Sindhya, K.: A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. IEEE Transactions on Evolutionary Computation **22**(1), 129–142 (2018)
5. Cornell, J.A.: Experiments with Mixtures: Designs, Models, and the Analysis of Mixture Data. John Wiley & Sons (2011)
6. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. IEEE Transactions on Evolutionary Computation **18**(4), 577–601 (2013)

7. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Proceedings of the 2002 Congress on Evolutionary Computation. pp. 825–830 (2002)
8. Hakanen, J., Chugh, T., Sindhya, K., Jin, Y., Miettinen, K.: Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms. In: 2016 IEEE Symposium Series on Computational Intelligence, Proceedings. IEEE (2016)
9. Hou, Z., Yang, S., Zou, J., Zheng, J., Yu, G., Ruan, G.: A performance indicator for reference-point-based multiobjective evolutionary optimization. In: 2018 IEEE Symposium Series on Computational Intelligence, Proceedings. pp. 1571–1578. IEEE (2018)
10. Huber, S., Geiger, M.J., Sevaux, M.: Simulation of preference information in an interactive reference point-based method for the bi-objective inventory routing problem. Journal of Multi-Criteria Decision Analysis **22**(1-2), 17–35 (2015)
11. Li, K., Deb, K., Yao, X.: R-metric: Evaluating the performance of preference-based evolutionary multiobjective optimization using reference points. IEEE Transactions on Evolutionary Computation **22**(6), 821–835 (2018)
12. Li, M., Yao, X.: Quality evaluation of solution sets in multiobjective optimisation: A survey. ACM Computing Surveys **52**(2), 1–38 (2019)
13. López-Ibáñez, M., Knowles, J.: Machine decision makers as a laboratory for interactive EMO. In: Gaspar-Cunha, A., Henggeler Antunes, C., Coello, C.C. (eds.) Evolutionary Multi-Criterion Optimization, 8th International Conference, Proceedings, Part II. pp. 295–309. Springer (2015)
14. Meignan, D., Knust, S., Frayret, J.M., Pesant, G., Gaud, N.: A review and taxonomy of interactive optimization methods in operations research. ACM Transactions on Interactive Intelligent Systems **5**(3), 17:1–17:43 (2015)
15. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1999)
16. Miettinen, K., Ruiz, F., Wierzbicki, A.P.: Introduction to multiobjective optimization: Interactive approaches. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization: Interactive and Evolutionary Approaches, pp. 27–57. Springer, Berlin, Heidelberg (2008)
17. Mohammadi, A., Omidvar, M.N., Li, X.: A new performance metric for user-preference based multi-objective evolutionary algorithms. In: 2013 IEEE Congress on Evolutionary Computation, Proceedings. pp. 2825–2832. IEEE (2013)
18. Ojalehto, V., Podkopaev, D., Miettinen, K.: Towards automatic testing of reference point based interactive methods. In: Handl, J., Hart, E., Lewis, P.R., López-Ibáñez, M., Ochoa, G., Paechter, B. (eds.) Parallel Problem Solving from Nature - PPSN XIV, 14th International Conference, Proceedings. pp. 483–492. Springer (2016)
19. Steuer, R.E.: Multiple Criteria Optimization: Theory, Computation and Application. John Wiley & Sons (1986)
20. Szczepanski, M., Wierzbicki, A.P.: Application of multiple criteria evolutionary algorithm to vector optimization, decision support and reference-point approaches. Journal of Telecommunications and Information Technology **3**(3), 16–33 (2003)
21. Xin, B., Chen, L., Chen, J., Ishibuchi, H., Hirota, K., Liu, B.: Interactive multiobjective optimization: A review of the state-of-the-art. IEEE Access **6**, 41256–41279 (2018)
22. Yu, G., Zheng, J., Li, X.: An improved performance metric for multiobjective evolutionary algorithms with user preferences. In: 2015 IEEE Congress on Evolutionary Computation, Proceedings. pp. 908–915. IEEE (2015)