Journal of Global Optimization (2023) 85:757–788
https://doi.org/10.1007/s10898-022-01230-3

Comparing reference point based interactive multiobjective optimization methods without a human decision maker

Lu Chen[1,2,3] · Kaisa Miettinen[4] · Bin Xin[1,2,3] · Vesa Ojalehto[4]

Received: 22 March 2020 / Accepted: 14 August 2022 / Published online: 24 September 2022

Abstract

Interactive multiobjective optimization methods have proven promising in solving optimization problems with conflicting objectives since they iteratively incorporate preference information of a decision maker in the search for the most preferred solution. To find the appropriate interactive method for various needs involves analysis of the strengths and weaknesses. However, extensive analysis with human decision makers may be too costly and for that reason, we propose an artificial decision maker to compare a class of popular interactive multiobjective optimization methods, i.e., reference point based methods. Without involving any human decision makers, the artificial decision maker works automatically to interact with different methods to be compared and evaluate the final results. It makes a difference between a learning phase and a decision phase, that is, learns about the problem based on information acquired to identify a region of interest and refines solutions in that region to find a final solution, respectively. We adopt different types of utility functions to evaluation solutions, present corresponding performance indicators and propose two examples of artificial decision makers. A series of experiments on benchmark test problems and a water resources planning problem is conducted to demonstrate how the proposed artificial decision makers can be used to compare reference point based methods.

✉ Bin Xin
brucebin@bit.edu.cn

Lu Chen
chenlubit@163.com

Kaisa Miettinen
kaisa.miettinen@jyu.fi

Vesa Ojalehto
vesa.ojalehto@gmail.com

[1] School of Automation, Beijing Institute of Technology, Beijing 100081, China

[2] State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing 100081, China

[3] Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing 100081, China

[4] University of Jyvaskyla, Faculty of Information Technology, P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland

Keywords Multicriteria optimization · Interactive multiobjective optimization · Learning phase · Decision phase · Performance comparison · Reference point

1 Introduction

Many real-world optimization problems involve conflicting objectives which are to be optimized simultaneously. These problems are known as multiobjective optimization problems. The ultimate aim of multiobjective optimization is to support a decision maker (DM) to find his/her most preferred solution. Up to now, various multiobjective optimization methods have been proposed by both the Multiple Criteria Decision Making (MCDM) and the Evolutionary Multiobjective Optimization (EMO) communities [7, 20, 30, 56]. They can be divided into no-preference methods, a priori methods, a posteriori methods, and interactive methods according to the role of the decision maker in the solution process [21, 29]. Among them, interactive methods have been widely developed due to several advantages [9, 29, 30, 33, 45]. For example, the DM can learn progressively from the solution process and adjust his/her preferences. In this case, only solutions that the DM is interested in will be obtained. Thus, the computational complexity can be reduced in comparison with approximating the whole set of Pareto optimal solutions.

In the solution process with an interaction method, we can often identify two phases: a learning phase and a decision phase [33]. The learning phase is important for a DM to learn about the possibilities and limitations of the problem. In the decision phase, the DM can focus on the region identified in the learning phase to find better solutions. Different interactive methods require different types of preference information from the DM, such as reference points, desirability of trade-offs, or the classification of objective functions [33]. Among them, reference point based methods are very popular owing to several reasons. For example, it is convenient and intuitive for the DM to specify a reference point consisting of desirable objective function values since the method provides objective function values for the DM, and, thus, no cognitive mapping is necessary [30]. Besides, reference points can be specified without considering the trade-offs among objectives. Thus, the DM's burden is relatively low [24].

The problem of how to compare interactive methods arises naturally with the emergence of diverse interactive methods. A posteriori methods like multiobjective evolutionary algorithms can be compared with some performance metrics (see, e.g., [25, 54]). However, comparing interactive methods is not very straightforward. Only limited studies (see, e.g., [3, 4, 8, 10–12, 26, 28, 35, 37, 39, 44, 58]) have focused on this topic and few of them are valid for reference point based interactive methods. In this paper, we concentrate on the comparison of reference point based methods.

The DM is a key element of applying and also comparing interactive methods. Using human DMs to compare interactive methods has some difficulties. Firstly, it may be rather expensive because human DMs involved in the experiments should have appropriate domain expertise. Furthermore, the order of using interactive methods will influence the comparison results because a human DM learns and what he/she has learnt when interacting with a method will affect his/her decisions on the subsequent methods. To eliminate such effects, a sufficiently large number of human DMs is required so that different groups of DMs would use the methods in different orders. Some research involving human DMs in the experiments has been conducted, and the DMs were asked to express their feelings about interactive methods such as ease of use and the degree of satisfaction on the final solution [3, 8, 11, 12]. However, these experiments were limited in e.g., the range of problems studied and the number of human DMs involved, see [6, 55].

Considering the above difficulties, a good alternative is to develop artificial DMs (ADMs) which can replace human DMs for comparing interactive methods quantitatively whenever human DMs are not available. In such comparisons, there are no real preferences to follow. Instead, ADMs interact with each interactive method and measure the results of the method. Opposed to using human DMs, experimenting with ADMs is cheaper and less time consuming. Besides, a large number of experiments can be conducted without considering human fatigue [6], and quantitative results can be obtained. Most existing ADMs take forms of utility functions (also referred to as value functions). For example, Malakooti and Ravindran [28] used linear utility functions as ADMs to compare multiobjective linear programming methods. Mote et al. [35] assumed a nonlinear utility function. Reeves and Gonzalez [39] adopted linear and multilinear utility functions. Four types of utility functions including quadratic, square-root, exponential and L_4-norm were considered in [44]. López-Ibáñez and Knowles [26] utilized linear scalarizing functions to assess an interactive EMO method.

Nevertheless, utility functions as ADMs cannot compare reference point based methods because they cannot provide reference point information required by these methods. Utility functions are only suitable for comparing *non ad hoc* interactive methods [45] such as trade-off based interactive methods since utility functions can provide the type of preference information required by these methods. To our best knowledge, only a few ADMs have been developed for comparing reference point based methods [1, 2, 4, 37, 38]. The ADMs developed in [4, 37, 38] all consists of a steady part which includes a pre-defined aspiration point or a value function, and a current context which includes the knowledge about the problem gained during the solutions process. In [37], reference points are generated by a decision tree-based approach. The ADM proposed in [4] evolves the reference point through particle swarm optimization. Reference points are generated in [38] by updating the potential region according to the predefined value function. In these ADMs, the search is maintained towards the steady part, which may hinder the exploration of the whole Pareto front. The ADMs proposed in [1, 2] distinguish the learning phase and decision phase. Different ways of generating reference points are developed for the two phases. However, these two ADMs are only suitable for comparing reference point based interactive EMO methods.

In this paper, we propose a new ADM for comparing reference point based methods quantitatively. Similar to the ADMs developed in [1, 2], our ADM also has a learning phase and a decision phase. It should be pointed out that this is the only similarity between our ADM and those two ADMs. Our ADM is designed with a modular structure containing three modules: learning, evaluation, and decision. The learning module and the decision module are used to generate reference points in the two phases, respectively. Specifically, a structured learning approach is designed in the learning module to generate reference points systematically based on Pareto optimal solutions derived so far to explore different regions of the search space, which facilitates identifying a region of interest (ROI). Besides, a polyhedral cone-based method is proposed to update reference points in the decision module so that the solutions in the ROI can be adjusted according to the current preferred solution. The evaluation module has two roles. One is evaluating solutions found so far and identifying a preferred solution at each iteration of the decision phase. The other is to measure the final results of interactive methods at the last iteration.

Compared to the ADMs developed in [4, 37, 38], our ADM does not need to converge to a fixed part at the very beginning. On the contrary, it behaves differently in two phases, reflecting the DM's different needs. In comparison with the two ADMs proposed in [1, 2], our ADM generates reference points in quite different ways. Those two ADMs assume that solution processes of all reference point based EMO methods being compared are conducted simultaneously. The reference point is updated based on the solutions generated by all methods. If a new method is to be compared, the whole comparison will start from the beginning, and all methods have to be run again. Our ADM treats all methods to be compared individually and does not have this limitation. Moreover, our ADM is applicable with any reference point based methods, not only EMO methods. It should be emphasized that our ADM does not enhance reference point based methods, it interacts with different methods and measures the final results of them for the sake of comparing them automatically.

By adopting utility functions with and without noise to evaluate solutions and providing two associated performance indicators to evaluate the final results in the evaluation module, two examples of the proposed ADM are presented in this paper. They are used to show how the proposed ADM can be utilized to compare reference point based methods and capture the differences among those methods. The main contributions are summarized as follows:

- A new modular ADM is proposed for comparing reference point based interactive multiobjective optimization methods automatically and quantitatively. It is not limited to some specific type of methods, only the preference information must be a reference point.
- The proposed ADM takes into account different needs that typically can be seen in the learning and the decision phases, and two different mechanisms of generating reference points are developed according to different needs.
- The proposed ADM's modular structure offers a flexible way of creating different type of ADMs which mimic different interaction processes.
- Two performance indicators are proposed to be used with the new ADM.

The rest of the paper is organized as follows. In Sect. 2, some concepts and notations are given. Section 3 is devoted to presenting the proposed ADM in terms of its actions in each phase and proposing two performance indicators to be used with the ADM. Two examples of the ADM are provided, and how to utilize one of them with a reference point based method is also illustrated. Sections 4 and 5 demonstrate how the ADM can be used to compare different reference point based methods on benchmark test problems and on a real-world problem, respectively. Finally, conclusions are drawn and some future work is discussed briefly in Sect. 6.

2 Concepts and notations

Generally, a multiobjective optimization problem can be defined as follows:

$$\begin{aligned} &\text{minimize } \mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\} \\ &\text{subject to } \ \mathbf{x} \in S \subset \mathbf{R}^n, \end{aligned} \tag{1}$$

where k ($k \geq 2$) objective functions $f_i : S \rightarrow \mathbf{R}$ for $i \in \{1, 2, \ldots, k\}$ are to be minimized simultaneously. The decision vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ belongs to the feasible region $S \subset \mathbf{R}^n$. For each $\mathbf{x}$, the objective vector $\mathbf{z} = \mathbf{f}(\mathbf{x})$ belongs to the objective space $\mathbf{R}^k$. The image $Z = \mathbf{f}(S)$ of the feasible region is called the feasible objective set.

Given two feasible decision vectors $\mathbf{x}^1$ and $\mathbf{x}^2$, $\mathbf{x}^1$ is said to dominate $\mathbf{x}^2$ if $\mathbf{x}^1$ is not worse than $\mathbf{x}^2$ in all objectives and strictly better in at least one objective. A feasible decision vector $\mathbf{x}$ is said to be Pareto optimal if and only if there is no feasible decision vector which dominates it. The corresponding objective vector is called a Pareto optimal objective vector. We denote the set of all Pareto optimal decision vectors as E. The image $\mathbf{f}(E)$ of E is called Pareto front. Note that we use the term Pareto optimal solution to refer to a Pareto optimal objective vector in this paper. The most preferred solution (MPS) is the Pareto optimal solution that the DM is most satisfied with [33].

As stated in the introduction, we concentrate on comparing reference point based methods, rather than proposing or enhancing such kind of methods. In the following, we give a brief introduction on reference point based methods. A reference point $\mathbf{g} = (g_1, g_2, \ldots, g_k)$ is composed of aspiration levels of the DM for all objectives reflecting desirable objective values. It is said to be achievable if its aspiration levels can be achieved or improved simultaneously by a feasible solution; otherwise, it is said to be unachievable [41]. When providing reference points, it is often useful for the DM to know the ranges of the objective values in the Pareto front. We denote $\mathbf{z}^* = (z_1^*, z_2^*, \ldots, z_k^*)$ with $z_i^* = \min_{\mathbf{x}\in S} f_i(\mathbf{x})$ for $i = 1, 2, \ldots, k$ to be the ideal objective vector which gives the minimum value of each objective in the feasible region. The nadir objective vector $\mathbf{z}^{nad} = (z_1^{nad}, z_2^{nad}, \ldots, z_k^{nad})$ is composed of the maximum value of each objective in the Pareto front, i.e., $z_i^{nad} = \max_{\mathbf{x}\in E} f_i(\mathbf{x})$ for $i = 1, 2, \ldots, k$. It is usually difficult to obtain $\mathbf{z}^{nad}$ and one may have to settle for approximations, for example, a payoff table [29] or other special ways [15, 47]. An estimate of $\mathbf{z}^{nad}$ can also be determined by using the information provided by the DM, if available.

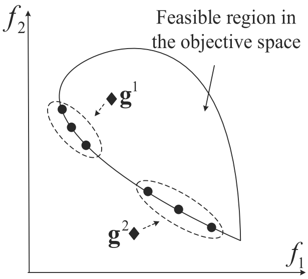
At each iteration, a reference point based method uses the reference point(s) provided by the DM to generate one or more Pareto optimal solutions. If the DM is not satisfied with any solution, he/she is expected to specify a new reference point for the method to generate new solutions. Figure 1 illustrates how an example of a reference point based method works in two iterations. This method produces k+1 Pareto optimal solutions based on one reference point. The solution process will continue until the DM finds a satisfactory solution.

Up to now, many reference point based methods have been proposed. Some methods, such as the reference point method [51], the reference direction (RD) method [23], the light beam search (LBS) [22], and the satisficing trade-off method [36], rely on solving transformed single-objective subproblems to generate Pareto optimal solutions. This means that at each iteration, they formulate a scalarizing function like an achievement scalarizing function (ASF) based on the DM's reference point and minimize it by using an appropriate single-objective optimizer. Various forms of ASFs have been developed (see, e.g., [27, 31, 42, 51]), and a common ASF is the following augmented ASF [52]:

$$s = \max_{i=1,\ldots,k} \{w_i(f_i(\mathbf{x}) - g_i)\} + \rho \sum_{i=1}^{k} w_i(f_i(\mathbf{x}) - g_i), \tag{2}$$

where $\mathbf{w} = [w_1, \ldots, w_k]$ is a weighting vector and ρ is a sufficiently small positive number. One can prove that it generates Pareto optimal solutions for both achievable and unachievable reference points and any Pareto optimal solution with trade-offs between ρ and $1/\rho$ [29].

Fig. 1 An example of how a reference point based method works



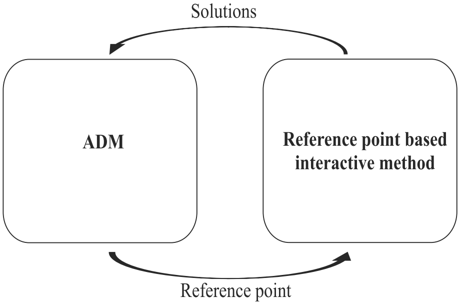


Fig. 2 Interaction between ADM and a reference point based interactive method

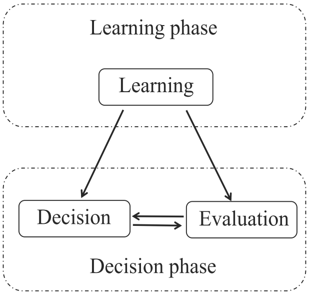


Fig. 3 Framework of the proposed ADM

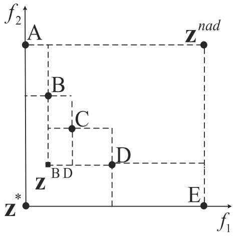
Some reference point based methods solve (1) directly by using a multiobjective evolutionary algorithm at each iteration. They use the DM's reference point to guide the evolution of the population towards the DM's preferred region on the Pareto front. If the DM does not find any satisfactory solution, he/she can supply a new reference point. A new run of the multiobjective evolutionary algorithm will be implemented to find new solutions. Even though evolutionary algorithms cannot usually guarantee Pareto optimality, for simplicity we refer to their solutions as Pareto optimal ones here. Examples of this type of methods include R-NSGA-II [17], RD-NSGA-II [13], LBS-NSGA-II [14], g-NSGA-II [34], r-NSGA-II [43], preference based evolutionary algorithm (PBEA) [49], interactive weighting achievement scalarizing function genetic algorithm (interactive WASF-GA) [40], and others, for surveys, see, e.g., [5, 53].

In the next section, the ADM we build for comparing reference point based methods is introduced.

3 Proposed ADM

To compare reference point based methods, the ADM needs to participate in the solution process. The interaction between the ADM and a method to be considered is shown in Fig. 2. The ADM has solutions provided by the method as input and a new reference point as output. The structure of the proposed ADM is given in Fig. 3. We assume that the ADM will first have a learning phase and then start a decision phase. The learning and decision modules are responsible for generating reference points in the learning phase and the decision phase, respectively. The evaluation module works in the decision phase. We denote the iteration number in the whole solution process by t, and the iteration number in the decision phase by t_d which counts from 0 when this phase starts. In what follows, the ADM is introduced

Fig. 4 Neighboring solutions in a bi-objective case



in terms of its actions in each phase. The steps of utilizing it with a reference point based method are also given.

3.1 ADM's actions in the learning phase

Owing to the modular structure of the proposed ADM, different types of learning modes of ADMs can be developed to simulate different ways of specifying reference points. As stated in the introduction, we focus on a structured learning approach to facilitate a systematic learning of feasible solutions. At each iteration, the solutions which have been generated by the considered interactive method are used to identify a relatively large unexplored region of the objective space. Then, a reference point is determined accordingly with the desire of obtaining new solutions in that region. Here an unexplored region refers to a region inside which no solutions have been found by the considered method yet. The details are as follows.

Firstly, we determine a series of unexplored regions by finding neighboring solutions. We denote an extreme point as the objective vector which has the minimum value of one of the objective functions on the Pareto front [48]. Considering all the extreme points and all the non-dominated solutions generated by the interactive method and passed to the ADM, we give a general definition of neighboring solutions for any number of objectives: For any two solutions $\mathbf{z}^a$ and $\mathbf{z}^b$, denote $\mathbf{z}^{ab} = [z_1^{ab}, \ldots, z_k^{ab}]$ with $z_i^{ab} = \min(z_i^a, z_i^b)$, $i = 1, 2, \ldots, k$. Then, $\mathbf{z}^a$ and $\mathbf{z}^b$ are defined as neighbors if no other solutions are dominated by $\mathbf{z}^{ab}$. Under this definition, the region dominated by $\mathbf{z}^{ab}$ is an unexplored region where the interactive method has not generated solutions inside it yet.

Figure 4 gives an example of a bi-objective case. Considering the five points A, B, C, D, and E, four pairs of neighboring solutions can be found: (A, B), (B, C), (C, D), and (D, E). All other pairs of solutions are not neighbors. For instance, B and D are not neighbors because $\mathbf{z}^{BD}$ dominates C. Figure 5 shows a three-objective case. Any solution in the Pareto front satisfies $f_1 + f_2 + f_3 = 1$. It is easy to verify that (A, D), (B, E), (C, F), (D, F), and (E, F) are neighbors. The gray region, i.e., the region dominated by (0, 0, 0.4), represents the unexplored region determined by (A, D). For problems with more than three objectives, it is also easy to find all neighboring solutions because we only need to determine the dominance relations of solutions. By finding all pairs of neighboring solutions, a series of unexplored regions of the objective space can be obtained.

Secondly, as the learning phase aims to learn about the possibilities of the problem, our ADM is assumed to regard the "largest" unexplored region as promising and want to explore it for finding new solutions. That is, among multiple unexplored regions, the "largest" one is identified as the next region to be explored. Considering that different unexplored regions may overlap when they spread upwards while what we really want to search is the region near the Pareto front, we use the distance between each pair of neighboring solutions to measure the size of each unexplored region. An unexplored region is regarded as the "largest" if

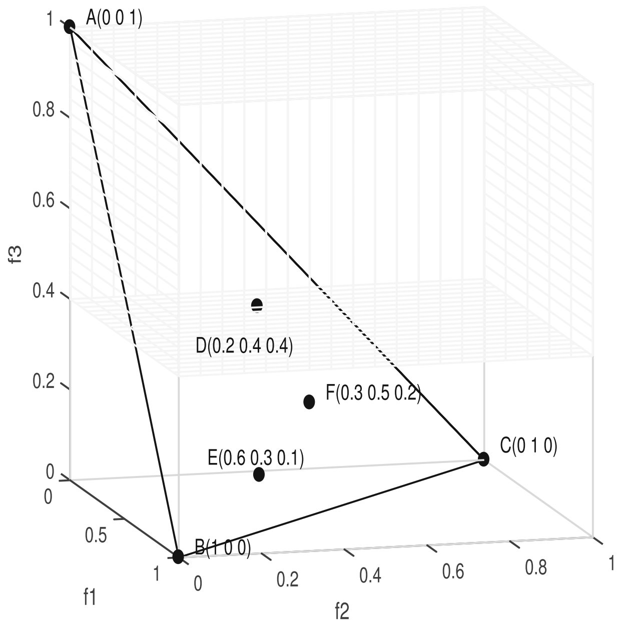


Fig. 5 Neighboring solutions in a three-objective case. The gray region is the unexplored region determined by A and D

the corresponding pair of neighboring solutions has the largest distance among all pairs' distances. Different forms of distance can be used and the normalized Euclidean distance is used in this paper:

$$d(\mathbf{z}^a, \mathbf{z}^b) = \sqrt{\sum_{i=1}^{k} \left(\frac{z_i^a - z_i^b}{z_i^{nad} - z_i^{**}} \right)^2}, \tag{3}$$

where $\mathbf{z}^{**} = [z_1^{**}, z_2^{**}, \ldots, z_k^{**}]$ is a utopian objective vector which is slightly better than $\mathbf{z}^*$. The components of $\mathbf{z}^{**}$ are given by $z_i^{**} = z_i^* - \varepsilon_i$ for all $i = 1, 2, \ldots, k$ where ε_i is a small positive number [45]. The utopian point is used instead of $\mathbf{z}^*$ in case that the denominator in (3) is zero or very small.

Note that when the Pareto front is disconnected or an interactive method poorly responds to the reference point, there may be the case that none of the newly generated solutions locate within the desired largest unexplored region. In this case, the largest unexplored region may always be selected as the largest in the following iterations. To avoid this situation, the same region is allowed to be chosen only once.

Finally, suppose the two neighboring solutions corresponding to the largest unexplored region are $\mathbf{z}^{a*}$ and $\mathbf{z}^{b*}$, the point $\mathbf{z}^{a*b*}$ with $z_i^{a*b*} = \min(z_i^{a*}, z_i^{b*})$ for all $i = 1, \ldots, k$ is taken as the new reference point. With this point, the ADM wants to explore the largest unexplored region for finding solutions in it.

To sum up, the steps of the structured learning approach for generating a new reference point is as follows:

1. Find all pairs of neighboring solutions among all extreme points and all the solutions generated by the interactive method and passed to the ADM.
2. Select the pair with the largest normalized Euclidean distance, denoted by $\mathbf{z}^{a*}$ and $\mathbf{z}^{b*}$.

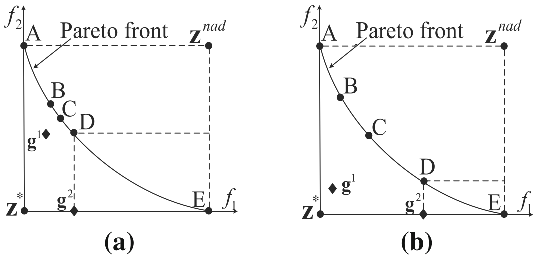


Fig. 6 Impact of the initial reference point on the second one: **a** the initial reference point is close to the Pareto front; **b** the initial reference point is far away from the Pareto front

3. Take $\mathbf{z}^{a*b*}$ with $z_i^{a*b*} = \min(z_i^{a*}, z_i^{b*})$ for all $i = 1, \ldots, k$ as the new reference point $\mathbf{g}^t = [g_1^t, g_2^t, \ldots, g_k^t]$.

When the initial reference point $\mathbf{g}^1$ is close to the Pareto front, the solutions obtained by an interactive method are likely to be close to each other as shown in Fig. 6a where B, C, and D are solutions obtained based on $\mathbf{g}^1$. Then, $\mathbf{g}^2$ will be relatively far away from the Pareto front and a large unknown region will get explored. When $\mathbf{g}^1$ is far away from the Pareto front, the solutions may be far away from each other as exhibited in Fig. 6b. The structured learning approach can lead the method to fill in those large and unexplored regions determined by these solutions step by step. All in all, by exploring the largest unexplored region at each iteration, the proposed ADM can learn about the Pareto front for identifying its ROI which is the region around its preferred solution among all the solutions passed to it in the learning phase.

3.2 ADM's actions in the decision phase

As mentioned, the decision phase contains the evaluation module and the decision module. In what follows, the tasks of the two modules are introduced.

3.2.1 Task of the evaluation module

At each iteration of the decision phase, the evaluation module evaluates all the solutions already generated and passed to the ADM and identifies the ADM's preferred solution at the moment. At the last iteration, it is also responsible for measuring the quality of the ADM's preferred solution. In our implementations, we adopt utility functions and corresponding performance indicators to achieve quantitative evaluation and measurement. Different kinds of utility functions can be used, so that we can experiment with different ADMs to see for example how reference point based methods perform under different styles of providing preferences and how their outcomes differ. In this paper, we consider the following two utility functions as examples:

1. *Deterministic utility function.* We denote a utility function by $U(\mathbf{z}) \in \mathbf{R}$. The preferences of the ADM with this utility function are always stable in the decision phase. To distinguish from a random utility function defined below, we call U a deterministic utility function.
2. *Random utility function.* By introducing noise into U we get a utility function which we call a random utility function with the form $\widetilde{U}(\mathbf{z}, \sigma_{t_d}) = U(\mathbf{z}) + N(0, \sigma_{t_d})$. The preferences of the ADM adopting this utility function are unstable. Its noise follows a normal distribution $N(0, \sigma_{t_d})$ with a mean 0 and a standard deviation σ_{t_d}. There are different choices for setting up σ_{t_d} such as keeping it as a constant or making it decrease with t_d. A decreasing σ_{t_d}

implies that the preferences of the corresponding ADM are getting more and more stable as the decision phase proceeds.

Incorporating each of the above two utility functions, two examples of the proposed ADM can be obtained, and these two ADMs will behave differently in the decision phase. Each ADM can be used to compare reference point based methods separately. In our experiments, we use both of them to compare not only the performance of different reference point based methods, but also their capability under different styles of providing preferences. In fact, other types of utility functions can also be chosen.

Since maximizing a utility function U is equivalent to minimizing the disutility function $U^- = -U$, the following deterministic disutility function U^- and random disutility function $\widetilde{U}^-$ will be used in our experiments to form two different ADMs:

$$U^-(\mathbf{z}) = \underset{1\leq i\leq k}{max}\left\{w_i \frac{(z_i - z_i^{**})}{z_i^{nad} - z_i^{**}}\right\}, \tag{4}$$

$$\widetilde{U}^-(\mathbf{z}) = \underset{1\leq i\leq k}{max}\left\{w_i \frac{(z_i - z_i^{**})}{z_i^{nad} - z_i^{**}}\right\} + N(0, \sigma_{t_d}), \tag{5}$$

where $\mathbf{w} = [w_1, \ldots, w_k]$ is a weighting vector, σ_{t_d} is assumed to decrease gradually and be reduced to zero at the final iteration, which means that the randomness of $\widetilde{U}^-$ is gone in the end and $\widetilde{U}^-$ becomes the same as U^-.

By using a disutility function, solutions generated to the ADM can be compared (and ordered), and the solution with the minimum disutility value at each iteration, denoted by $\mathbf{z}^{best,t-1} = [z_1^{best,t-1}, \ldots, z_k^{best,t-1}]$, can be recognized. Particularly, the final output of the interactive method, namely the solution most preferred by the ADM finally, is called the final solution and is denoted by $\mathbf{z}^{final}$. Note that where $\mathbf{z}^{final}$ is chosen from depends on the interactive methods to be compared. In the literature, many methods like the reference point method [51] only show newly obtained solutions to a DM at each iteration. The final solution will be chosen from the solutions obtained at the final iteration. Few interactive methods like the LBS [22] allow a DM to save a set of desirable solutions and select a satisfactory solution from this set. Naturally, one can keep an archive to store all obtained solutions in real implementations. In our experiments in Sect. 4, the final solution is chosen from solutions obtained at the final iteration.

Denote the solution which minimizes U^- by $\mathbf{z}^{MPS}$. It represents the real MPS of both U^- and $\widetilde{U}^-$. The corresponding disutility value, denoted by U^{-*}, gives the minimum value of U^-. The maximum value of U^- on the Pareto front is denoted by U^{-max}. An example of a performance indicator is the normalized difference between the disutility value of $\mathbf{z}^{final}$ and U^{-*}:

$$difference = \frac{U^-(\mathbf{z}^{final}) - U^{-*}}{U^{-max} - U^{-*}} \times 100\%. \tag{6}$$

The smaller the *difference* value is, the better the final solution is.

As $\mathbf{z}^{final}$ is selected according to the disutility function, it may not be the one closest to $\mathbf{z}^{MPS}$ in the sense of the Euclidean distance. So we use the following normalized Euclidean distance between $\mathbf{z}^{final}$ and $\mathbf{z}^{MPS}$ as the second performance indicator:

$$distance = \sqrt{\sum_{i=1}^{k}\left(\frac{z_i^{final} - z_i^{MPS}}{z_i^{nad} - z_i^{**}}\right)^2}. \tag{7}$$

Remark 1 The reason for the evaluation module to be involved only in the decision phase is to enable learning about the problem in the learning phase. Alternatively, one can assume that it can be adopted in the learning phase and can guide the generation of new reference points. Note that if an ADM uses the evaluation module in the learning phase in the same way as it does in the decision phase, it is actually reduced as an ADM which has no learning phase but only one decision phase.

3.2.2 Task of the decision module

According to the current preferred solution $\mathbf{z}^{best,t-1}$, the decision module generates a new reference point $\mathbf{g}^t$ with the purpose of exploiting the neighborhood of $\mathbf{z}^{best,t-1}$ to find better solutions. The details are as follows.

We denote the set of all the generated solutions and all extreme points by $\{\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^{m^t}\}$ where m^t is the size of this set. The decision module determines a polyhedral cone with $\mathbf{z}^{best,t-1}$ inside it or lying on the boundary and takes the vertex of the cone, denoted by $\mathbf{z}^c = [z_1^c, \ldots, z_k^c]$, as $\mathbf{g}^t$. Specifically, for each $i \in \{1, \ldots, k\}$, if $z_i^{best,t-1} = z_i^*$, then let $z_i^c = z_i^*$; otherwise, take the largest objective value from the objective values less than $z_i^{best,t-1}$ among all the m^t values as z_i^c:

$$z_i^c = \max_{j \in \{1,\ldots,m^t\}} \{z_i^j | z_i^j < z_i^{best,t-1}\}.$$

Figure 7 shows how a new reference point is determined in a bi-objective case. The point C is supposed to be the most preferred one among the five points A, B, C, D, and E. For the first objective, z_1^c is the largest objective value among all the objective values less than $z_1^{best,t-1}$. Similarly, for the second objective, z_2^c is the largest value among all the objective values which are less than $z_2^{best,t-1}$. Hence, $\mathbf{z}^c = [z_1^c, z_2^c]$ is taken as the new reference point.

The vertex $\mathbf{z}^c$ is taken as the new reference point because we assume that the ADM finds the polyhedral cone associated with the current preferred solution to be promising and wants to find better solutions in it. If an interactive method responds to the guidance of the new referent point $\mathbf{g}^t$ well and generates at least one solution in the current polyhedral cone, this cone tends to be narrowed down. For example, in Fig. 8, F is a newly obtained solution at the tth iteration. If C is still the most preferred one found so far, then the new polyhedral cone will be the region dominated by G. If F is preferred to C and suppose it is $\mathbf{z}^{best,t}$, then the new polyhedral cone will be the region dominated by H. In both cases, the polyhedral cone is narrowed down to facilitate a more focused search.

We have now introduced the actions of the proposed ADM in each phase. It is clear that the ADM's actions in the two phases differ a lot. Firstly, the ADM explores different regions in the learning phase to gain more knowledge about the problem, while in the decision phase, the ADM concentrates on a surrounding region of a promising solution to refine solutions.

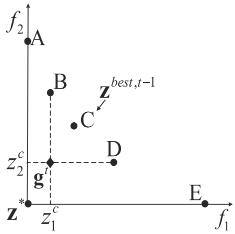


Fig. 7 The new reference point generated in the decision phase

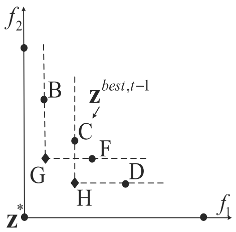


Fig. 8 The possible new polyhedral cone

Secondly, in the learning phase, only the information on the locations of generated solutions is used to generate reference points. In the decision phase, solutions are evaluated and the most preferred one is identified so as to determine new reference points.

By adopting two different types of utility functions, two examples of the ADM can be acquired. We denote them by ADM1 and ADM2. The deterministic utility function reflects the stability of ADM1's preferences, while the random one indicates that ADM2's preferences are unstable but are getting clearer as more solutions are obtained.

Remark 2 Though only one learning phase and one decision phase are considered in our ADM, the two phases can actually be carried out repeatedly and an ADM's actions in a later learning or decision phase can be changed. For example, we can first utilize an ADM which generates reference points randomly in the learning phase and uses a random utility function in the decision phase. Next, we conduct the two phases again with ADM1. This implies that the preferences of the composite ADM become stable in the second decision phase.

Remark 3 It is also possible to switch interactive methods in different phases to judge their performance in a single phase. For instance, one can use interactive methods A and B to find solutions in the learning phase and use method C in the decision phase. In this way, the performance of A and B in the learning phase can be compared. Through identifying the strengths and weaknesses of interactive methods in each phase, new interactive methods can be designed by combining the advantages of existing methods which behave well in different phases.

3.3 The proposed ADM's steps and an example

3.3.1 ADM's steps

Recall that the iteration numbers in the whole solution process and in the decision phase are t and t_d, respectively. Denote the largest allowable numbers of iterations in the learning phase and the decision phase by T_l and T_d, respectively. To evaluate the final solutions obtained by different interactive methods under the same number of iterations, the overall numbers of iterations of the ADM in each phase are assumed to be fixed. In other words, the ADM switches from the learning phase to the decision phase when $t > T_l$ and the decision phase is terminated when $t_d > T_d$. In what follows, the steps of how the ADM interacts with a reference point based method and evaluates the final solution are given. The corresponding flowchart is presented in Fig. 9.

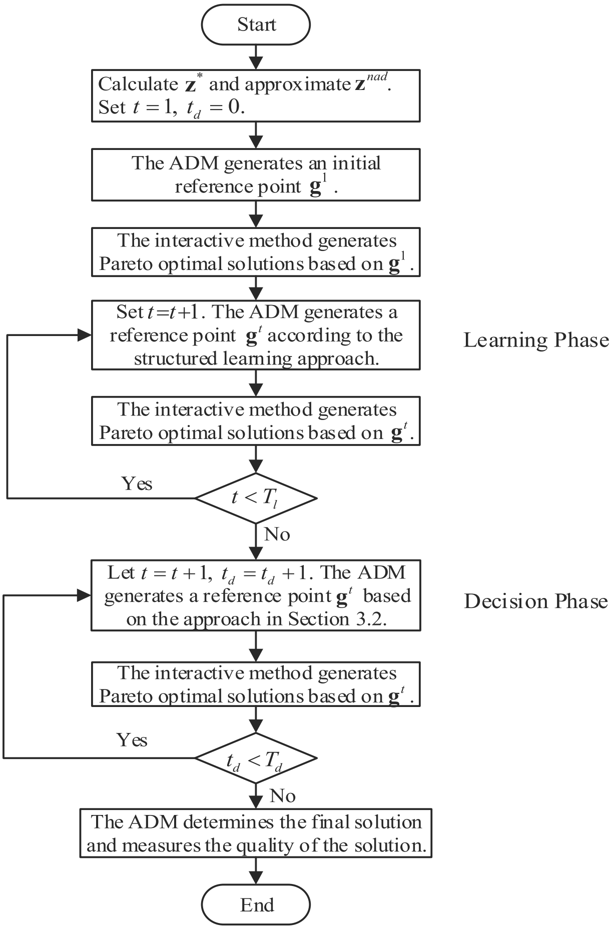

Start
Calculate $\mathbf{z}^*$ and approximate $\mathbf{z}^{nad}$. Set $t = 1$, $t_d = 0$.
The ADM generates an initial reference point $\mathbf{g}^1$.
The interactive method generates Pareto optimal solutions based on $\mathbf{g}^1$.
Set t=t+1. The ADM generates a reference point $\mathbf{g}^t$ according to the structured learning approach.
The interactive method generates Pareto optimal solutions based on $\mathbf{g}^t$.
Yes
$t < T_l$
No
Learning Phase
Let $t = t+1$, $t_d = t_d + 1$. The ADM generates a reference point $\mathbf{g}^t$ based on the approach in Section 3.2.
The interactive method generates Pareto optimal solutions based on $\mathbf{g}^t$.
Yes
$t_d < T_d$
No
Decision Phase
The ADM determines the final solution and measures the quality of the solution.
End

Fig. 9 Flowchart of how the proposed ADM works

Step 1 (Initialization of the learning phase) Set $t = 1, t_d = 0$. Generate an initial reference point $\mathbf{g}^1$ randomly or let the user of the ADM specify it within the hyperbox determined by $\mathbf{z}^*$ and approximate $\mathbf{z}^{nad}$. Let the interactive method generate Pareto optimal solutions based on $\mathbf{g}^1$.

Step 2 (Learning phase) Let $t = t + 1$. The ADM provides a new reference point $\mathbf{g}^t$ according to the structured learning approach in Sect. 3.1. Then, the interactive method generates corresponding Pareto optimal solutions. Repeat this step until $t = T_l$. Then, go to Step 3.

Step 3 (Decision phase) Set $t = t + 1, t_d = t_d + 1$. The ADM generates a new reference point $\mathbf{g}^t$ according to Sect. 3.2. Obtain new Pareto optimal solutions by using the interactive method. Repeat this procedure until $t_d = T_d$. Use the evaluation module to determine a solution as the final output and calculate the performance indicator value of it. Then, the whole process is terminated.

3.3.2 An example of how the ADM works

Now we illustrate how the proposed ADM works with a reference point based method on the bi-objective problem ZDT1 [57]. ADM1 is used as an example and a simple linear disutility function $U^-(f_1, f_2) = f_1 + f_2$ is used to evaluate solutions. Figure 10 shows the contour lines (dotted lines) of the disutility function and the MPS ($\mathbf{z}^{MPS} = [0.25, 0.5]$). The curve represents the Pareto front of ZDT1. The interactive method we use is the reference point method [51] which uses the DM's reference point and k perturbed reference points to generate $k+1$ Pareto optimal solutions by minimizing $k+1$ ASFs at each iteration. The augmented ASF shown by (2) is adopted and is minimized by using the differential evolution algorithm DE/rand/1/bin [46]. Three and two iterations are implemented in the learning phase and the decision phase, respectively.

All the reference points provided by ADM1 and Pareto optimal solutions generated by the reference point method [51] in the learning phase are shown in Fig. 11. We set the initial reference point as $\mathbf{g}^1 = [0.5, 0.1]$. Three Pareto optimal solutions $\mathbf{z}^{1,1}$, $\mathbf{z}^{1,2}$, and $\mathbf{z}^{1,3}$ are generated based on $\mathbf{g}^1$. The new reference point $\mathbf{g}^2$ is generated based on $\mathbf{z}^{1,1}$ and the extreme point $\mathbf{z} = [0, 1]$ which are neighbors and have the largest normalized Euclidean distance among all pairs of neighbors. Then, another three Pareto optimal solutions $\mathbf{z}^{2,1}$, $\mathbf{z}^{2,2}$, and $\mathbf{z}^{2,3}$ are obtained. Similarly, the third reference point $\mathbf{g}^3$ is determined and solutions $\mathbf{z}^{3,1}$, $\mathbf{z}^{3,2}$, and $\mathbf{z}^{3,3}$ are generated.

At the fourth iteration, the decision phase begins. According to the disutility function, $\mathbf{z}^{2,2}$ is the solution with the minimum disutility value found so far, i.e., $\mathbf{z}^{best,3} = \mathbf{z}^{2,2}$ as shown in Fig. 12a. For the first objective, $z_1^{2,1}$ is the largest objective value among all the objective values which are smaller than $z_1^{best,3}$. For the second objective, $z_2^{2,3}$ is the largest objective value among all the objective values less than $z_2^{best,3}$. Thus, $\mathbf{g}^4 = [z_1^{2,1}, z_2^{2,3}]$ is the new reference point. Another three Pareto optimal solutions $\mathbf{z}^{4,1}$, $\mathbf{z}^{4,2}$, and $\mathbf{z}^{4,3}$ corresponding to $\mathbf{g}^4$ are generated and $\mathbf{z}^{4,2}$ has the minimum disutility value now, i.e., $\mathbf{z}^{best,4} = \mathbf{z}^{4,2}$. Figure 12b presents the new reference point $\mathbf{g}^5 = [z_1^{2,2}, z_2^{4,3}]$ and three new solutions $\mathbf{z}^{5,1}$, $\mathbf{z}^{5,2}$, and $\mathbf{z}^{5,3}$. The solution $\mathbf{z}^{5,1}$ overlaps $\mathbf{z}^{MPS}$ and it has the minimum disutility value. Its two performance indicator values are all zero according to (6) and (7), which means that the reference point method behaves well on ZDT1 under the evaluation of ADM1.

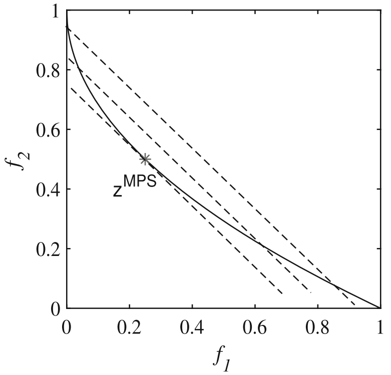
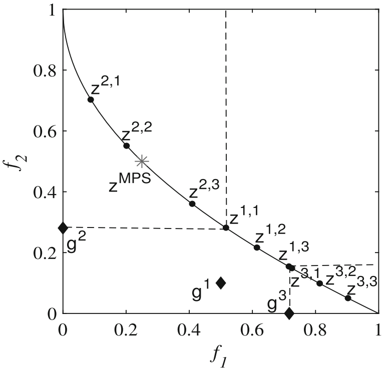
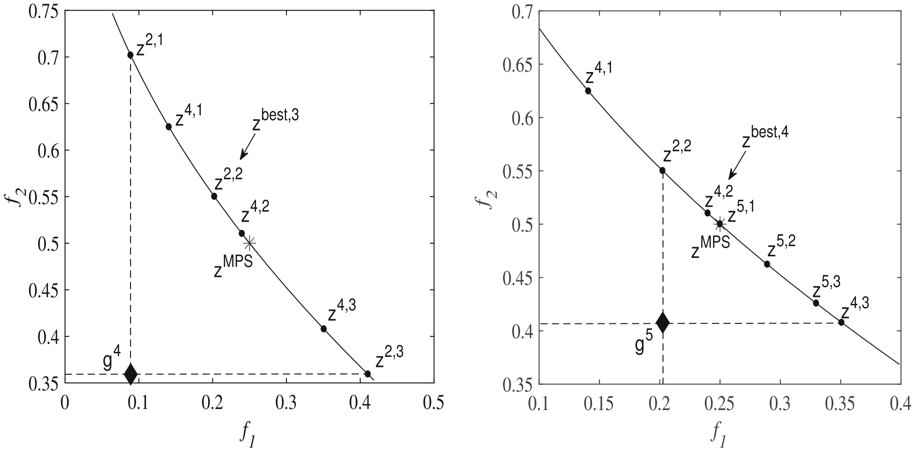


Fig. 10 Contour lines of the disutility function and the MPS

Fig. 11 The reference points and corresponding Pareto optimal solutions in the learning phase





(a) **(b)**

Fig. 12 The reference points and corresponding Pareto optimal solutions in the decision phase: **a** the fourth iteration; **b** the fifth iteration

4 Numerical experiments on benchmark test problems

In this section, we use ADM1 and ADM2 as examples to demonstrate how the proposed ADM can be utilized to compare reference point based methods on benchmark test problems. Four popular methods, i.e., the reference point method [51], R-NSGA-II [17], g-NSGA-II [34], and r-NSGA-II [43] are used as examples of reference point based methods. As introduced in Sect. 3.3.2, the reference point method generates $k+1$ Pareto optimal solutions by minimizing $k+1$ ASFs at each iteration. The other three methods use the DM's reference point to modify NSGA-II [16]. Nevertheless, their ways of modification are different. R-NSGA-II modifies the crowding distance mechanism of NSGA-II to prefer solutions close to the reference point. g-NSGA-II uses *g-dominance* relation to replace the Pareto dominance in NSGA-II to emphasize solutions which satisfy all aspiration levels or achieve none of the aspiration levels. r-NSGA-II substitutes *r-dominance* for the Pareto dominance in NSGA-II to make solutions closer to the reference point more preferred. In the literature, the three methods are usually used as a priori methods. We use them interactively in this paper as follows. At each iteration, the modified NSGA-II is performed for a certain number of generations to find Pareto optimal solutions corresponding to the ADM's current reference point. The procedures of the modified NSGA-II at different iterations are independent. That is to say, the modified NSGA-II is rerun by starting from a newly randomly generated initial population at each iteration. In our experiments, the codes of the latter two methods come from the MATLAB platform PlatEMO [50].

4.1 Experimental settings

4.1.1 Test problems

Three- and five-objective DTLZ1, DTLZ2, DTLZ3, DTLZ4 and DTLZ7 [18] are used as test problems. The Pareto front of DTLZ1 is linear. DTLZ2, DTLZ3, and DTLZ7 have concave Pareto fronts. DTLZ7 has 2^{k-1} disconnected Pareto optimal regions.

4.1.2 Parameter settings of interactive methods

For the reference point method, we adopt the same implementation as used in Sect. 3.3. The parameter values used in DE/rand/1/bin are: (1) population size $NP = 5n$; (2) the scaling factor $F = 0.5$; (3) the crossover probability $CR = 0.5$; and (4) the maximum number of generations $G_{max} = 400$. Note that DE/rand/1/bin is called $k + 1$ times at each iteration of the reference point method. For the other three methods, the population size is 100 and 200 for three-objective and five-objective problems, respectively. The clearing parameter ϵ in R-NSGA-II is set as 0.01 [17] and the non-r-dominance threshold δ in r-NSGA-II is set as 0.3 [43]. Since these three methods generate many solutions at each iteration, we select $k+1$ solutions to the ADM by clustering. This implies that the four methods will present the same number of solutions to the ADM at each iteration. To get comparable results with the four methods, their total numbers of function evaluations are kept the same at each iteration.

4.1.3 Parameter settings of ADMs

It has been reported in the literature that the median number of iterations performed with interactive methods is often between three and eight [19]. Too many iterations will bring a heavy burden to the DM. On the other hand, a reference point based method is likely to have better results when more iterations are adopted since more solutions can be obtained. In order to allow the four methods to have better performance within an acceptable number of iterations, we set the overall number of iterations T as eight. In addition, we set $T_l = 5$, $T_d = 3$. In fact, one can set T, T_d, and T_l the way one wishes.

The formulas of the disutility functions in (4) and (5) are used. Table 1 gives the weighting vectors used in them. For each test problem, the difference between $\mathbf{z}^{nad}$ and $\mathbf{z}^{*}$ in each dimension is large enough, so we use $\mathbf{z}^{*}$ directly instead of $\mathbf{z}^{**}$ (i.e., $\varepsilon_i = 0$, for all $i \in \{1, \ldots, k\}$) in (4), (5), and (7). For ADM2, the standard deviations of the noise of $\widetilde{U}^{-}$ in the decision phase are set as $\sigma_1 = 0.2 \times (U^{-max} - U^{-*})$, $\sigma_2 = \sigma_1/2$, $\sigma_3 = \sigma_2/2$. After the solutions generated at the final iteration are given to the ADM, σ_{t_d} will be reduced to zero, which means that the noise of ADM2's disutility function disappears finally. The MPS on each test problem and corresponding optimal disutility value are listed in Table 1.

Table 1 The weighting vectors, MPSs, and the optimal disutility values

Problem	$\mathbf{w}$	$\mathbf{z}^{MPS}$	U^{-*}
3-obj DTLZ1	[1, 1.2, 1.5]	[0.2, 0.1667, 0.1333]	0.4
3-obj DTLZ2/3/4	[1, 1.2, 1.5]	[0.6838, 0.5698, 0.4558]	0.6838
3-obj DTLZ7	[1, 1.2, 3]	[0.8094, 0.6745, 3.6771]	0.9419
5-obj DTLZ1	[1, 1.2, 1.2, 1.2, 1.5]	[0.12, 0.1, 0.1, 0.1, 0.08]	0.24
5-obj DTLZ2/3/4	[1, 1.2, 1.2, 1.2, 1.5]	[0.5324, 0.4437, 0.4437, 0.4437, 0.3549]	0.5324
5-obj DTLZ7	[4, 4, 1, 1, 2]	[0.1951, 0.1951, 0.7804, 0.7804, 6.3026]	0.9080

4.1.4 Initial reference point

We use four different initial reference points for each test problem, as listed in Table 2. They are denoted by *irp1, irp2, irp3, and irp4* where *irp1* and *irp2* are unachievable while *irp3* and *irp4* are achievable. Besides, *irp1* and *irp3* are close to the Pareto front while *irp2* and *irp4* are far away from the Pareto front. On each test problem, the two ADMs will use the same four initial reference points when applying each interactive method. The approach of generating various initial reference points is given in the supplementary material.

4.2 Experimental results and analysis

4.2.1 Comparison of the four methods by using the proposed ADM

In this subsection, the proposed ADM1 and ADM2 are utilized to compare the four reference point based methods. Since randomness is involved in the solution process, the solutions generated by each method with the same reference point may vary in different runs. Hence, for every ADM, each method is run 21 times independently on each test problem under each initial reference point. This means that each method is applied for a total of $2 \times 4 \times 10 \times 21 = 1680$ times. In each run, the values of the two indicators *difference* and *distance* are calculated by using (6) and (7), respectively.

- *Comparison according to the mean values.* The mean values and the standard deviations of each method over 21 independent runs are listed in Tables 3 and 4. In these two tables, the first three columns show the test problem, the number of objectives, the initial reference point, respectively. The fourth to seventh columns present results of the four methods in terms of *difference*. Note that the reference point method is abbreviated as RPM. The values in parentheses are the standard deviations. Similarly, the eighth to eleventh columns give results in terms of *distance*. In order to facilitate comparison, we ranked the four methods on each instance by using 1, 2, 3, and 4 according to their mean values. The method with rank 1 has the smallest mean value. All the ranks are listed in the fourth to eleventh columns of each table. The average rank of each method over 40 instances is shown in the last row of each table.

According to Tables 3 and 4, no method is absolutely superior to the other methods on all instances. The reference point method ranks the first on more than half of the instances under each indicator, no matter which ADM is utilized. Most of its mean *difference* values are less than 10% and the corresponding standard deviations are less than 5%, which means that it can generally find solutions close to the real MPS and it has a relatively good stability.

Table 2 The initial reference points for each test problem

Problem	$irp1$	$irp2$	$irp3$	$irp4$
3-obj DTLZ1	[0.22, 0.20, 0.01]	[0.1, 0.08, 0.07]	[0.42, 0.12, 0.11]	[0.35, 0.33, 0.32]
3-obj DTLZ2/3/4	[0.72, 0.61, 0.08]	[0.34, 0.28, 0.23]	[0.96, 0.45, 0.36]	[0.84, 0.78, 0.73]
3-obj DTLZ7	[0.15, 0.69, 4.19]	[0.4, 0.34, 3.15]	[0.80, 0.81, 3.41]	[0.53, 0.46, 5.58]
5-obj DTLZ1	[0.19, 0.1, 0.05, 0, 0.14]	[0.06, 0.05, 0.05, 0.05, 0.04]	[0.09, 0.09, 0.08, 0.4, 0.04]	[0.31, 0.3, 0.3, 0.3, 0.29]
5-obj DTLZ2/3/4	[0.56, 0.5, 0.54, 0.06, 0]	[0.27, 0.22, 0.22, 0.22, 0.18]	[0.48, 0.93, 0.31, 0.36, 0.25]	[0.77, 0.72, 0.72, 0.72, 0.68]
5-obj DTLZ7	[0.23, 0.37, 0.2, 0.82, 7.2]	[0.1, 0.1, 0.39, 0.39, 4.77]	[0.82, 0.76, 0.77, 0.74, 6.21]	[0.64, 0.76, 0.76, 0.55, 9.01]

Table 3 Results of the four methods when using ADM1 to compare them

Problem	k	*irp*	*Difference*				*Distance*			
			RPM	R-NSGA-II	g-NSGA-II	r-NSGA-II	RPM	R-NSGA-II	g-NSGA-II	r-NSGA-II
DTLZ1	3	*irp*1	3.906(2.809)^{1}	7.632(3.521)^{2}	3296(2514)^{4}	14.79(8.204)^{3}	0.052(0.040)^{1}	0.152(0.079)^{2}	32.67(31.51)^{4}	0.180(0.090)^{3}
		irp2	0.429(0.388)^{1}	3.693(3.574)^{2}	4316(3342)^{4}	17.63(9.682)^{3}	0.005(0.006)^{1}	0.043(0.035)^{2}	47.96(38.01)^{4}	0.228(0.118)^{3}
		irp3	3.349(1.875)^{1}	6.756(4.409)^{2}	4580(3941)^{4}	16.75(8.619)^{3}	0.047(0.027)^{1}	0.091(0.067)^{2}	45.98(38.73)^{4}	0.210(0.108)^{3}
		irp4	1.422(0.773)^{1}	1.788(1.881)^{2}	5362(4280)^{4}	19.22(8.854)^{3}	0.022(0.014)^{1}	0.030(0.034)^{2}	54.27(43.50)^{4}	0.238(0.123)^{3}
	5	*irp1*	2.914(1.175)^{1}	15.17(3.395)^{2}	26257(7450)^{4}	76.46(41.03)^{3}	0.045(0.013)^{1}	0.286(0.026)^{2}	377.1(61.07)^{4}	1.000(0.523)^{3}
		irp2	0.956(0.368)^{1}	13.97(4.783)^{2}	25853(6222)^{4}	92.55(47.04)^{3}	0.020(0.005)^{1}	0.241(0.094)^{2}	357.3(46.57)^{4}	1.204(0.616)^{3}
		irp3	5.587(1.756)^{1}	14.74(4.773)^{2}	26430(6424)^{4}	80.62(51.01)^{3}	0.104(0.031)^{1}	0.274(0.068)^{2}	378.0(70.19)^{4}	0.973(0.581)^{3}
		irp4	1.802(0.561)^{1}	10.54(4.449)^{2}	26552(8138)^{4}	102.4(60.35)^{3}	0.025(0.006)^{1}	0.218(0.077)^{2}	369.2(89.57)^{4}	1.292(0.768)^{3}
DTLZ2	3	*irp1*	6.941(0.000)^{2}	9.955(2.459)^{4}	5.134(1.710)^{1}	9.739(2.486)^{3}	0.180(0.000)^{2}	0.344(0.106)^{4}	0.083(0.038)^{1}	0.339(0.113)^{3}
		irp2	0.411(0.000)^{1}	10.15(7.317)^{4}	5.400(5.961)^{3}	3.894(3.975)^{2}	0.006(0.000)^{1}	0.227(0.226)^{4}	0.052(0.051)^{2}	0.065(0.120)^{3}
		irp3	2.666(0.000)^{1}	10.25(7.718)^{4}	7.401(3.510)^{3}	4.491(5.152)^{2}	0.022(0.000)^{1}	0.130(0.090)^{4}	0.090(0.046)^{3}	0.055(0.073)^{2}
		irp4	1.590(0.000)^{1}	7.758(11.35)^{3}	6.066(3.758)^{2}	19.61(3.567)^{4}	0.019(0.000)^{1}	0.129(0.208)^{3}	0.091(0.071)^{2}	0.514(0.125)^{4}
	5	*irp1*	11.72(0.000)^{1}	20.97(6.935)^{3}	149.8(26.52)^{4}	13.19(1.892)^{2}	0.444(0.000)^{1}	0.636(0.082)^{3}	1.759(0.226)^{4}	0.515(0.069)^{2}
		irp2	5.308(0.648)^{1}	25.69(5.980)^{3}	153.8(23.68)^{4}	11.72(9.191)^{2}	0.074(0.014)^{1}	0.646(0.146)^{3}	1.774(0.169)^{4}	0.300(0.243)^{2}
		irp3	3.317(0.040)^{1}	32.51(7.910)^{3}	152.5(22.70)^{4}	18.07(5.770)^{2}	0.054(0.000)^{1}	0.789(0.090)^{3}	1.843(0.206)^{4}	0.538(0.123)^{2}
		irp4	3.878(0.000)^{1}	12.11(5.415)^{2}	157.3(30.93)^{4}	21.85(5.707)^{3}	0.043(0.000)^{1}	0.445(0.164)^{2}	1.745(0.194)^{4}	0.493(0.209)^{3}

Table 3 continued

Problem	k	*irp*	*Difference*				*Distance*			
			RPM	R-NSGA-II	g-NSGA-II	r-NSGA-II	RPM	R-NSGA-II	g-NSGA-II	r-NSGA-II
DTLZ3	3	*irp1*	6.225(2.786)^{1}	8.431(3.884)^{2}	5497(3598)^{4}	20.52(7.009)^{3}	0.127(0.086)^{1}	0.244(0.144)^{2}	40.95(25.41)^{4}	0.280(0.081)^{3}
		irp2	4.687(3.733)^{1}	6.114(7.005)^{2}	6373(5582)^{4}	24.17(5.204)^{3}	0.053(0.047)^{1}	0.093(0.138)^{2}	44.18(30.19)^{4}	0.334(0.089)^{3}
		irp3	2.878(0.885)^{1}	8.956(6.379)^{2}	5761(3978)^{4}	21.42(6.080)^{3}	0.042(0.028)^{1}	0.123(0.117)^{2}	41.92(27.46)^{4}	0.304(0.105)^{3}
		irp4	3.840(3.441)^{1}	5.347(4.752)^{2}	4755(3621)^{4}	23.85(4.047)^{3}	0.036(0.025)^{1}	0.065(0.064)^{2}	35.15(25.51)^{4}	0.330(0.089)^{3}
	5	*irp1*	10.26(4.590)^{1}	17.69(5.127)^{2}	91875(32154)^{4}	43.73(43.91)^{3}	0.255(0.124)^{1}	0.558(0.063)^{3}	1012(329.9)^{4}	0.538(0.460)^{2}
		irp2	7.792(1.938)^{1}	27.27(9.837)^{2}	100039(22665)^{4}	49.77(70.71)^{3}	0.139(0.052)^{1}	0.608(0.193)^{2}	1051(226.3)^{4}	0.680(1.044)^{3}
		irp3	10.25(3.005)^{1}	35.46(7.867)^{2}	100121(17770)^{4}	59.73(46.90)^{3}	0.218(0.098)^{1}	0.780(0.114)^{3}	1067(185.0)^{4}	0.678(0.517)^{2}
		irp4	6.935(2.092)^{1}	13.41(7.478)^{2}	102039(18981)^{4}	35.24(36.73)^{3}	0.130(0.094)^{1}	0.440(0.138)^{2}	1098(194.6)^{4}	0.493(0.432)^{3}
DTLZ4	3	*irp1*	7.433(2.155)^{2}	17.17(8.346)^{4}	4.709(1.574)^{1}	9.660(2.855)^{3}	0.191(0.097)^{2}	0.488(0.116)^{4}	0.087(0.051)^{1}	0.196(0.100)^{3}
		irp2	5.597(3.541)^{1}	13.14(7.981)^{4}	5.738(3.600)^{2}	8.574(3.223)^{3}	0.070(0.048)^{2}	0.405(0.210)^{4}	0.066(0.047)^{1}	0.111(0.064)^{3}
		irp3	2.276(2.440)^{1}	8.411(6.360)^{3}	7.260(4.905)^{2}	9.091(3.194)^{4}	0.033(0.035)^{1}	0.126(0.118)^{4}	0.113(0.137)^{2}	0.116(0.049)^{3}
		irp4	3.686(2.809)^{1}	7.257(4.904)^{3}	7.127(3.615)^{2}	8.967(3.323)^{4}	0.045(0.034)^{1}	0.160(0.152)^{4}	0.113(0.075)^{2}	0.141(0.100)^{3}
	5	*irp1*	12.06(1.248)^{1}	22.22(6.094)^{3}	29.53(15.45)^{4}	13.56(3.784)^{2}	0.473(0.073)^{3}	0.636(0.075)^{4}	0.457(0.161)^{1}	0.292(0.093)^{2}
		irp2	10.60(5.805)^{1}	24.66(7.845)^{3}	32.88(18.01)^{4}	13.22(4.034)^{2}	0.290(0.174)^{2}	0.694(0.159)^{4}	0.430(0.220)^{3}	0.228(0.063)^{1}
		irp3	14.93(4.936)^{2}	18.41(5.926)^{3}	30.02(12.77)^{4}	14.07(4.692)^{1}	0.484(0.080)^{3}	0.512(0.160)^{4}	0.436(0.150)^{2}	0.247(0.092)^{1}
		irp4	11.13(5.915)^{2}	10.05(6.992)^{1}	27.25(13.13)^{4}	17.76(5.382)^{3}	0.225(0.176)^{1}	0.332(0.195)^{3}	0.339(0.137)^{4}	0.313(0.129)^{2}
DTLZ7	3	*irp1*	8.653(2.336)^{2}	10.71(2.684)^{3}	7.787(1.566)^{1}	34.58(7.567)^{4}	0.294(0.053)^{2}	0.557(0.220)^{3}	0.203(0.052)^{1}	0.711(0.076)^{4}
		irp2	1.124(0.510)^{1}	1.625(1.742)^{3}	1.543(1.116)^{2}	39.09(13.94)^{4}	0.050(0.027)^{3}	0.049(0.049)^{2}	0.045(0.029)^{1}	0.739(0.083)^{4}
		irp3	2.882(0.005)^{2}	8.312(2.937)^{3}	1.832(0.955)^{1}	38.16(11.31)^{4}	0.065(0.007)^{2}	0.448(0.188)^{3}	0.047(0.026)^{1}	0.715(0.090)^{4}
		irp4	6.184(3.809)^{3}	5.480(4.598)^{2}	2.417(1.522)^{1}	39.04(10.96)^{4}	0.218(0.101)^{2}	0.238(0.251)^{3}	0.075(0.051)^{1}	0.713(0.078)^{4}
	5	*irp1*	6.255(2.704)^{1}	12.50(3.639)^{3}	38.63(12.49)^{4}	9.730(2.277)^{2}	0.387(0.247)^{1}	0.756(0.102)^{4}	0.694(0.217)^{3}	0.670(0.115)^{2}
		irp2	4.118(2.366)^{3}	2.039(0.606)^{2}	49.37(12.29)^{4}	1.224(1.004)^{1}	0.246(0.144)^{3}	0.089(0.019)^{2}	0.830(0.229)^{4}	0.061(0.043)^{1}
		irp3	2.525(0.594)^{1}	78.34(24.63)^{4}	46.84(20.87)^{2}	76.59(24.07)^{3}	0.156(0.032)^{1}	0.751(0.224)^{3}	0.812(0.277)^{4}	0.724(0.233)^{2}
		irp4	8.263(2.989)^{1}	32.80(18.07)^{2}	46.18(24.30)^{4}	36.12(23.24)^{3}	0.532(0.204)^{1}	0.836(0.289)^{3}	0.856(0.347)^{4}	0.727(0.305)^{2}
Average rank			1.25	2.6	3.275	2.875	1.375	2.85	3.075	2.7

Table 4 Results of the four methods when using ADM2 to compare them

Problem	k	*irp*	*Difference*				*Distance*			
			RPM	R-NSGA-II	g-NSGA-II	r-NSGA-II	RPM	R-NSGA-II	g-NSGA-II	r-NSGA-II
DTLZ1	3	*irp1*	7.448(3.965)^{1}	10.14(3.975)^{2}	4364(4158)^{4}	15.08(7.673)^{3}	0.135(0.086)^{1}	0.206(0.098)^{3}	41.18(38.76)^{4}	0.204(0.107)^{2}
		irp2	2.292(2.079)^{1}	9.780(6.291)^{2}	6360(5281)^{4}	14.92(8.040)^{3}	0.031(0.034)^{1}	0.125(0.092)^{2}	66.91(65.26)^{4}	0.194(0.107)^{3}
		irp3	5.864(4.218)^{1}	14.33(5.915)^{2}	3054(2330)^{4}	17.10(6.936)^{3}	0.094(0.081)^{1}	0.178(0.072)^{2}	28.01(20.14)^{4}	0.215(0.086)^{3}
		irp4	5.462(3.816)^{2}	2.849(2.022)^{1}	3475(3492)^{4}	17.01(6.650)^{3}	0.084(0.059)^{2}	0.039(0.026)^{1}	33.86(31.01)^{4}	0.220(0.081)^{3}
	5	*irp1*	6.647(3.566)^{1}	20.00(6.093)^{2}	25960(7819)^{4}	68.24(39.00)^{3}	0.126(0.076)^{1}	0.335(0.048)^{2}	364.8(56.89)^{4}	0.890(0.470)^{3}
		irp2	3.344(3.579)^{1}	17.84(5.221)^{2}	28451(7976)^{4}	119.0(61.82)^{3}	0.049(0.044)^{1}	0.270(0.089)^{2}	389.8(67.88)^{4}	1.558(0.899)^{3}
		irp3	8.391(2.258)^{1}	19.99(7.230)^{2}	25372(6442)^{4}	93.16(40.69)^{3}	0.144(0.042)^{1}	0.331(0.085)^{2}	363.5(51.77)^{4}	1.218(0.532)^{3}
		irp4	3.396(2.621)^{1}	12.88(4.980)^{2}	28505(7419)^{4}	106.7(47.97)^{3}	0.056(0.044)^{1}	0.227(0.069)^{2}	384.7(57.93)^{4}	1.370(0.677)^{3}
DTLZ2	3	*irp1*	11.02(2.209)^{3}	11.72(2.967)^{4}	8.398(2.182)^{1}	10.61(1.936)^{2}	0.299(0.102)^{2}	0.365(0.095)^{4}	0.181(0.079)^{1}	0.351(0.102)^{3}
		irp2	7.367(6.026)^{2}	15.21(7.737)^{4}	6.685(3.701)^{1}	7.858(6.193)^{3}	0.108(0.103)^{3}	0.309(0.226)^{4}	0.073(0.049)^{1}	0.099(0.121)^{2}
		irp3	4.252(2.468)^{1}	20.98(9.284)^{4}	11.45(7.515)^{3}	7.383(7.678)^{2}	0.063(0.045)^{1}	0.318(0.151)^{4}	0.141(0.086)^{3}	0.100(0.129)^{2}
		irp4	4.360(4.996)^{1}	5.626(6.465)^{2}	7.824(4.101)^{3}	22.72(8.860)^{4}	0.044(0.044)^{1}	0.073(0.135)^{2}	0.122(0.091)^{3}	0.437(0.204)^{4}
	5	*irp1*	14.87(2.660)^{1}	21.05(5.596)^{3}	149.6(31.45)^{4}	15.60(3.284)^{2}	0.413(0.089)^{1}	0.604(0.076)^{3}	1.672(0.255)^{4}	0.536(0.056)^{2}
		irp2	6.539(3.265)^{1}	32.09(9.009)^{3}	149.2(21.04)^{4}	12.99(8.607)^{2}	0.100(0.069)^{1}	0.691(0.128)^{3}	1.724(0.241)^{4}	0.201(0.182)^{2}
		irp3	4.177(1.110)^{1}	35.86(7.674)^{3}	152.7(25.34)^{4}	20.61(5.650)^{2}	0.071(0.012)^{1}	0.808(0.095)^{3}	1.807(0.223)^{4}	0.569(0.128)^{2}
		irp4	6.716(2.247)^{1}	11.88(5.468)^{2}	149.4(20.79)^{4}	25.89(6.056)^{3}	0.093(0.042)^{1}	0.360(0.195)^{2}	1.762(0.193)^{4}	0.478(0.190)^{3}
DTLZ3	3	*irp1*	10.35(5.313)^{1}	10.36(2.826)^{2}	7123(4675)^{4}	21.06(6.440)^{3}	0.237(0.130)^{1}	0.294(0.134)^{3}	49.94(28.90)^{4}	0.274(0.101)^{2}
		irp2	8.769(3.583)^{1}	11.08(7.427)^{2}	5880(2451)^{4}	24.18(6.470)^{3}	0.099(0.037)^{1}	0.156(0.134)^{2}	41.02(13.51)^{4}	0.347(0.108)^{3}
		irp3	4.742(2.349)^{1}	12.17(8.390)^{2}	6236(4704)^{4}	21.47(7.421)^{3}	0.079(0.051)^{1}	0.154(0.128)^{2}	46.87(35.13)^{4}	0.318(0.126)^{3}
		irp4	8.520(5.052)^{2}	4.671(2.823)^{1}	6544(9890)^{4}	22.51(5.238)^{3}	0.098(0.066)^{2}	0.053(0.038)^{1}	43.95(53.84)^{4}	0.308(0.090)^{3}
	5	*irp1*	12.82(4.487)^{1}	17.93(5.621)^{2}	99566(22136)^{4}	36.22(27.31)^{3}	0.323(0.132)^{1}	0.559(0.087)^{3}	1064(243.9)^{4}	0.475(0.330)^{2}
		irp2	10.16(3.808)^{1}	29.14(6.047)^{2}	89091(17358)^{4}	68.08(61.60)^{3}	0.157(0.071)^{1}	0.654(0.194)^{2}	978.9(157.1)^{4}	0.784(0.689)^{3}
		irp3	13.19(4.363)^{1}	37.63(9.944)^{3}	88988(20326)^{4}	33.94(32.36)^{2}	0.325(0.159)^{1}	0.781(0.142)^{3}	1001(194.3)^{4}	0.477(0.438)^{2}
		irp4	11.01(3.314)^{2}	9.095(3.472)^{1}	99941(19039)^{4}	43.67(35.16)^{3}	0.272(0.140)^{1}	0.301(0.159)^{2}	1131(220.3)^{4}	0.551(0.344)^{3}

Table 4 continued

Problem	k	*irp*	*Difference*				*Distance*			
			RPM	R-NSGA-II	g-NSGA-II	r-NSGA-II	RPM	R-NSGA-II	g-NSGA-II	r-NSGA-II
DTLZ4	3	*irp1*	11.46(2.682)^{2}	15.84(7.303)^{4}	10.25(6.227)^{1}	11.56(2.783)^{3}	0.293(0.126)^{3}	0.475(0.104)^{4}	0.239(0.161)^{1}	0.290(0.120)^{2}
		irp2	9.462(5.975)^{2}	20.72(11.03)^{4}	6.553(4.204)^{1}	10.94(6.328)^{3}	0.121(0.097)^{2}	0.442(0.167)^{4}	0.091(0.095)^{1}	0.169(0.126)^{3}
		irp3	5.500(3.969)^{1}	17.26(7.572)^{4}	9.466(7.111)^{2}	10.17(5.487)^{3}	0.076(0.057)^{1}	0.300(0.170)^{4}	0.112(0.089)^{2}	0.179(0.123)^{3}
		irp4	6.842(3.559)^{1}	9.285(5.935)^{2}	10.10(4.993)^{3}	10.77(6.271)^{4}	0.081(0.051)^{1}	0.213(0.180)^{4}	0.172(0.095)^{3}	0.161(0.129)^{2}
	5	*irp1*	14.74(1.643)^{1}	19.92(6.157)^{3}	34.21(20.55)^{4}	14.88(4.338)^{2}	0.520(0.047)^{3}	0.600(0.078)^{4}	0.518(0.183)^{2}	0.322(0.091)^{1}
		irp2	12.07(6.294)^{1}	33.80(14.96)^{4}	28.32(15.14)^{3}	12.82(3.744)^{2}	0.314(0.193)^{2}	0.715(0.156)^{4}	0.357(0.189)^{3}	0.225(0.076)^{1}
		irp3	17.79(6.135)^{2}	25.32(11.16)^{3}	39.56(20.36)^{4}	14.47(4.808)^{1}	0.499(0.108)^{2}	0.525(0.131)^{4}	0.508(0.268)^{3}	0.287(0.133)^{1}
		irp4	10.61(3.242)^{1}	11.99(5.213)^{2}	31.13(16.76)^{4}	17.99(5.432)^{3}	0.263(0.174)^{1}	0.406(0.165)^{3}	0.412(0.160)^{4}	0.333(0.138)^{2}
DTLZ7	3	*irp1*	8.653(2.336)^{2}	11.99(2.066)^{3}	7.787(1.566)^{1}	34.58(7.567)^{4}	0.294(0.053)^{2}	0.537(0.214)^{3}	0.203(0.052)^{1}	0.711(0.076)^{4}
		irp2	1.124(0.510)^{1}	4.923(3.009)^{3}	1.543(1.116)^{2}	39.09(13.94)^{4}	0.050(0.027)^{2}	0.186(0.149)^{3}	0.045(0.029)^{1}	0.739(0.083)^{4}
		irp3	2.882(0.005)^{2}	8.071(3.302)^{3}	1.832(0.955)^{1}	38.16(11.31)^{4}	0.065(0.007)^{2}	0.282(0.157)^{3}	0.047(0.026)^{1}	0.715(0.090)^{4}
		irp4	6.184(3.809)^{2}	8.587(5.035)^{3}	2.417(1.522)^{1}	39.04(10.96)^{4}	0.218(0.101)^{2}	0.368(0.248)^{3}	0.075(0.051)^{1}	0.713(0.078)^{4}
	5	*irp1*	8.329(0.967)^{1}	14.22(4.030)^{3}	38.63(12.49)^{4}	13.32(3.572)^{2}	0.597(0.178)^{1}	0.747(0.078)^{4}	0.694(0.217)^{2}	0.726(0.055)^{3}
		irp2	6.165(2.691)^{3}	3.527(1.884)^{2}	49.37(12.29)^{4}	3.116(2.870)^{1}	0.237(0.176)^{3}	0.111(0.045)^{2}	0.830(0.229)^{4}	0.096(0.055)^{1}
		irp3	45.33(40.40)^{1}	91.00(2.716)^{4}	46.84(20.87)^{2}	88.80(3.046)^{3}	0.548(0.373)^{1}	1.028(0.196)^{4}	0.812(0.277)^{2}	0.904(0.156)^{3}
		irp4	9.317(3.379)^{1}	63.76(15.03)^{3}	46.18(24.30)^{2}	72.52(15.66)^{4}	0.579(0.171)^{1}	0.899(0.172)^{4}	0.856(0.347)^{2}	0.889(0.197)^{3}
Average rank			1.46	2.54	3.04	2.96	1.38	2.63	3.17	2.83

R-NSGA-II usually ranks the last on three-objective DTLZ2 and DTLZ4, and ranks the second or the third on the other test problems. Similarly, r-NSGA-II ranks the second or the third in most cases. From the average rank, R-NSGA-II and r-NSGA-II rank the second or the third among the four methods. When ADM2 is utilized, R-NSGA-II is slightly better than r-NSGA-II in terms of each indicator. When ADM1 is used, R-NSGA-II has a better *difference* value while r-NSGA-II is better under the *distance* indicator. This shows that a better disutility value and a closer distance are not always consistent. It is meaningful to use different indicators to measure the methods' performance.

Although ranking the first or second on three-objective DTLZ7, g-NSGA-II ranks the last on most of the other instances, which makes it the worst one among the four methods. Meanwhile, its mean *difference* values on DTLZ1 and DTLZ3 are very large. When checking the solutions obtained by g-NSGA-II, we found that the final population of g-NSGA-II at each iteration is usually far away from the Pareto front, especially on DTLZ1 and DTLZ3. This shows that g-NSGA-II has a relatively weak capability to converge to the Pareto front. In fact, g-NSGA-II has a drawback that it may prefer a solution $\mathbf{a}$ to another solution which dominates $\mathbf{a}$ in the Pareto sense, which can hinder its convergence.

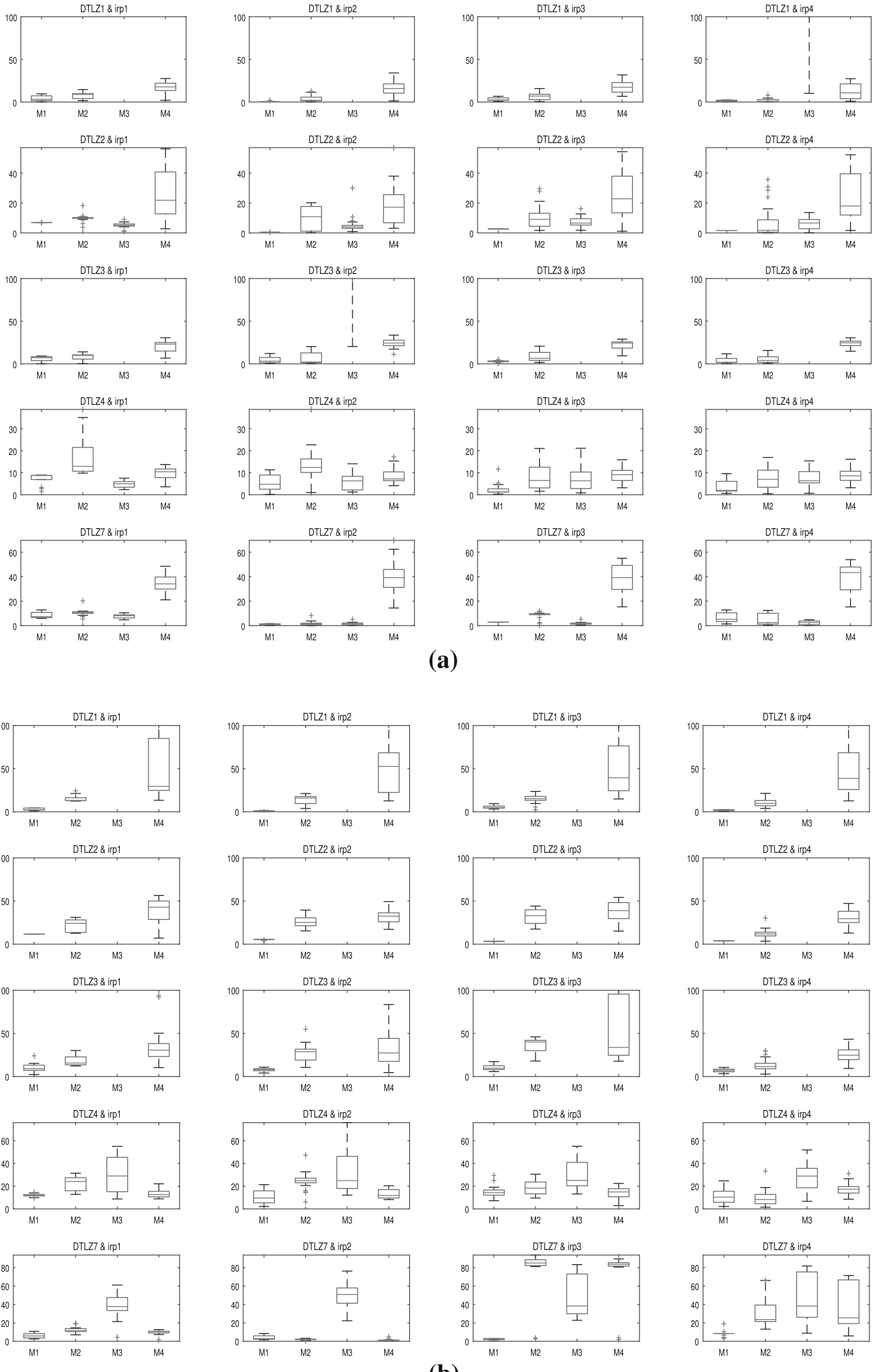
From the differences between the mean values on three- and five-objective problems, it can be observed that the reference point method has no significant differences while the other three methods generally get better mean values on three-objective problems. This reflects the degradation of the three methods' performance with the increase of the number of objectives.

• *Comparison according to the boxplots.* For the sake of an intuitive comparison of the four methods, the boxplots of the *difference/distance* values over 21 independent runs of each method are drawn. Here we only present the boxplots of the *difference* values when ADM1 is used, as shown in Fig. 13. All the boxplots can be found in the supplementary material and they give similar results to Fig. 13 does. In Fig. 13, M1, M2, M3, and M4 represent the reference point method, R-NSGA-II, g-NSGA-II, and r-NSGA-II, respectively. As the *difference* values of g-NSGA-II are much larger than those of the other methods on several problems, in order to see the differences among the other methods, the maximum *difference* value of the boxes is restricted to be no more than 100%. Values larger than 100% are not shown.

From Fig. 13, we can see that the *difference* values of the reference point method are usually smaller and the lengths of the boxes are usually shorter than the other methods on most test problems. This means that the reference point method generally finds better solutions and is more stable than the others. g-NSGA-II performs rather badly on three- and five-objective DTLZ1, DTLZ3. However, it has relatively smaller boxes which are closer to zero than the other three methods on three-objective DTLZ7. R-NSGA-II and r-NSGA-II usually rank in the middle among the four methods. These observations are similar to what we have derived according to the mean values and standard deviations of the four methods.

• *Comparison according to statistical tests.* To compare the four methods pairwise, the Wilcoxon rank sum test at a significance level of 0.05 is conducted on each pair of methods. We use three symbols $+$, $\approx$, and $-$ to represent that method i performs statistically better than, equal to, and worse than method j, respectively. The numbers of these symbols over 40 instances with respect to each ADM and each indicator are given in Tables 5, 6, 7 and 8.

In Tables 5, 6, 7 and 8, both the indicators give rather similar results. The reference point method performs statistically better than or equal to the other three methods in most cases, no matter which ADM is used. On the contrary, g-NSGA-II is inferior to the other methods in more than a half of the instances. These observations are consistent with what we found from the mean values in Tables 3 and 4. R-NSGA-II is a little better than r-NSGA-II when ADM1


DTLZ1 & irp1 DTLZ1 & irp2 DTLZ1 & irp3 DTLZ1 & irp4

DTLZ2 & irp1 DTLZ2 & irp2 DTLZ2 & irp3 DTLZ2 & irp4

DTLZ3 & irp1 DTLZ3 & irp2 DTLZ3 & irp3 DTLZ3 & irp4

DTLZ4 & irp1 DTLZ4 & irp2 DTLZ4 & irp3 DTLZ4 & irp4

DTLZ7 & irp1 DTLZ7 & irp2 DTLZ7 & irp3 DTLZ7 & irp4

(a)

DTLZ1 & irp1 DTLZ1 & irp2 DTLZ1 & irp3 DTLZ1 & irp4

DTLZ2 & irp1 DTLZ2 & irp2 DTLZ2 & irp3 DTLZ2 & irp4

DTLZ3 & irp1 DTLZ3 & irp2 DTLZ3 & irp3 DTLZ3 & irp4

DTLZ4 & irp1 DTLZ4 & irp2 DTLZ4 & irp3 DTLZ4 & irp4

DTLZ7 & irp1 DTLZ7 & irp2 DTLZ7 & irp3 DTLZ7 & irp4

(b)

Fig. 13 Boxplots of *difference* values (%) over 21 independent runs for each method when using ADM1 to compare methods: **a** boxplots on 3-objective test problems; **b** boxplots on 5-objective test problems

Table 5 Numbers of $+/\approx/-$ in terms of *difference* when using ADM1 to compare method i with method j

Method i	Method j		
	R-NSGA-II	g-NSGA-II	r-NSGA-II
RPM	31/8/1	33/3/4	34/5/1
R-NSGA-II	–	25/9/6	21/8/11
g-NSGA-II	–	–	9/3/28

Table 6 Numbers of $+/\approx/-$ in terms of *distance* when using ADM1 to compare method i with method j

Method i	Method j		
	R-NSGA-II	g-NSGA-II	r-NSGA-II
RPM	29/8/3	30/7/3	30/9/1
R-NSGA-II	–	25/4/11	21/6/13
g-NSGA-II	–	–	10/3/27

Table 7 Numbers of $+/\approx/-$ in terms of *difference* when using ADM2 to compare method i with method j

Method i	Method j		
	R-NSGA-II	g-NSGA-II	r-NSGA-II
RPM	31/8/1	30/5/5	33/4/3
R-NSGA-II	–	21/12/7	15/12/13
g-NSGA-II	–	–	9/6/25

Table 8 Numbers of $+/\approx/-$ in terms of *distance* when using ADM2 to compare method i with method j

Method i	Method j		
	R-NSGA-II	g-NSGA-II	r-NSGA-II
RPM	31/6/3	26/10/4	27/10/3
R-NSGA-II	–	22/6/12	15/12/13
g-NSGA-II	–	–	8/7/25

is used, and the two methods perform similarly under ADM2. This is slightly different from the comparison result of them according to their average ranks in Tables 3 and 4.

4.2.2 Influence of the initial reference point

According to Tables 3 and 4, the mean *difference/distance* values of some methods when using different initial reference points differ on some problems. To see whether the initial reference point influences the results of each method, we conducted the Wilcoxon rank sum test at a significance level of 0.05 to compare the results of each method under the following pairs of initial reference points: (1) *irp1* and *irp2*, (2) *irp1* and *irp3*, (3) *irp2* and *irp4*, and (4) *irp3* and *irp4*. Tables 9, 10, 11 and 12 present the numbers of $+/\approx/-$ over the ten test problems for each pair of initial reference points. Each table corresponds to a method.

According to Table 9, the reference point method performs better on more than a half of the instances when using *irp2* than using *irp1*. Under other pairs of initial reference points, the reference point method behaves inversely on different problems. Generally, *irp3* leads to better results on more problems than *irp1* does, while it is slightly inferior to *irp4* in terms of *distance*; *irp2* is slightly better than *irp4*. Hence, the reference point method usually gets

Table 9 Numbers of $+/\approx/-$ over ten problems for the reference point method when using different pairs of initial reference points

	ADM1		ADM2	
	Difference	*Distance*	*Difference*	*Distance*
irp1 versus *irp2*	0/4/6	0/0/10	0/2/8	0/0/10
irp1 versus *irp3*	2/2/6	1/3/6	1/4/5	0/5/5
irp2 versus *irp4*	5/4/1	5/4/1	3/7/0	4/5/1
irp3 versus *irp4*	4/2/4	2/2/6	3/4/3	1/6/3

Table 10 Numbers of $+/\approx/-$ over ten problems for R-NSGA-II when using different pairs of initial reference points

	ADM1		ADM2	
	Difference	*Distance*	*Difference*	*Distance*
irp1 versus *irp2*	1/6/3	1/5/4	3/5/2	3/2/5
irp1 versus *irp3*	3/5/2	2/2/6	5/4/1	3/3/4
irp2 versus *irp4*	2/2/6	2/4/4	2/0/8	2/1/7
irp3 versus *irp4*	0/2/8	0/2/8	0/1/9	0/3/7

Table 11 Numbers of $+/\approx/-$ over ten problems for g-NSGA-II when using different pairs of initial reference points

	ADM1		ADM2	
	Difference	*Distance*	*Difference*	*Distance*
irp1 versus *irp2*	1/7/2	1/7/2	1/7/2	1/5/4
irp1 versus *irp3*	1/8/1	0/9/1	0/9/1	0/8/2
irp2 versus *irp4*	0/10/0	1/9/0	1/9/0	2/8/0
irp3 versus *irp4*	0/10/0	0/10/0	0/10/0	1/9/0

Table 12 Numbers of $+/\approx/-$ over ten problems for r-NSGA-II when using different pairs of initial reference points

	ADM1		ADM2	
	Difference	*Distance*	*Difference*	*Distance*
irp1 versus *irp2*	0/8/2	0/5/5	1/6/3	1/4/5
irp1 versus *irp3*	2/7/1	0/8/2	3/6/1	2/6/2
irp2 versus *irp4*	4/6/0	4/6/0	4/6/0	4/6/0
irp3 versus *irp4*	1/7/2	1/9/0	2/7/1	1/8/1

better results when using *irp2* as compared with using other initial reference points, which implies that this method is more likely to obtain a final solution closer to the MPS when the ADM's initial reference point is unachievable and far away from the Pareto front on the ten test problems.

Table 10 shows that R-NSGA-II usually has opposite performance on different problems under *irp1*, *irp2*, and *irp3*. In most cases, R-NSGA-II performs better under *irp4* than *irp3*, which means that between the two achievable initial reference points, R-NSGA-II tends to behave better when the initial reference point far away from the Pareto front is used.

From Tables 11 and 12, g-NSGA-II and r-NSGA-II are insensitive to the location of the initial reference point. On more than a half of the problems, the results under different initial reference points are statistically indifferent.

5 Numerical experiments on a water resources planning problem

This section illustrates how to use the proposed ADM to compare reference point based methods on a real-world problem, namely, a water resources planning problem [32]:

$$\begin{aligned} \text{minimize } f_1(\mathbf{x}) &= e^{0.01x_1} x_1^{0.02} x_2^2 \\ f_2(\mathbf{x}) &= 0.5x_2^2 \\ f_3(\mathbf{x}) &= -e^{0.005x_1} x_1^{0.001} x_2^2 \\ \text{subject to } & 0.01 \le x_1 \le 1.3 \\ & 0.01 \le x_2 \le 10, \end{aligned} \tag{8}$$

where x_1 and x_2 are the total man-hours devoted to building a dam and the mean radius of the lake impounded (in miles), respectively. The three objectives f_1, f_2, and f_3 represent the cost of construction, the water loss, and the total storage capacity of the reservoir, respectively.

We compare four reference point based methods, i.e., the reference point method [51], R-NSGA-II [17], g-NSGA-II [34], and r-NSGA-II [43]. We consider an ADM with the following disutility function

$$U^-(\mathbf{z}) = \max_{1 \le i \le 3} \left\{ \frac{(z_i - z_i^*)}{z_i^{nad} - z_i^*} \right\}, \tag{9}$$

where $\mathbf{z}^*$ and $\mathbf{z}^{nad}$ are the ideal point and the nadir point of the water resources planning problem, respectively. The MPS of this ADM is $\mathbf{z}^{MPS} = [50.92, 25, -50.34]$. The minimum disutility value is 0.5.

Similar to Sect. 3.3, DE/rand/1/bin is adopted as a solver to minimize the augmented ASF formed by the reference point method. The parameter values of DE/rand/1/bin are set as: (1) population size $NP = 20$; (2) the scaling factor $F = 0.5$; (3) the crossover probability $CR = 0.5$; and (4) the maximum number of generations $G_{max} = 200$. For the other three methods, we set $NP = 40$ and $G_{max} = 400$. The overall numbers of function evaluations of the four methods are the same. With regard to the ADM, the numbers of iterations in the learning phase and the decision phase are both set as 3. Since the reference point method obtains four solutions at each iteration, the final population of each of the other three methods are clustered into four groups and four representative solutions are passed to the ADM at each iteration for fair comparison.

We have an unachievable initial reference point $p1 = [30, 15, -80]$. Each method was run 20 times. The *difference* and *distance* indicator values were calculated in each run. Table 13 shows the mean values and the standard deviations (in parentheses) of the indicators for the four methods over 20 independent runs. According to the mean *difference* values in Table 13, the order of the four methods from the best to the worst was g-NSGA-II, RPM, R-NSGA-II, and r-NSGA-II. In terms of the mean *distance* values, g-NSGA-II and RPM still ranked the first and the second, respectively, while r-NSGA-II performed better than R-NSGA-II. It

Table 13 Mean and standard deviation (in parentheses) of indicator values for the four methods

	RPM	R-NSGA-II	g-NSGA-II	r-NSGA-II
Difference	1.814(0.000)	1.946(0.744)	0.752(0.503)	3.288(2.435)
Distance	0.016(0.000)	0.182(0.057)	0.008(0.005)	0.050(0.068)

should be noted that the deviations of the two indicator values for the reference point method are both zero. This means that the reference point method found the same final solution in all the 20 runs and it is rather stable when solving this water resources planning problem.

In each of the 20 independent runs, the 24 solutions obtained by each reference point based method in the solution process were recorded. Figure 14 shows the MPS and all the 24 solutions obtained by each method in the run when the method has the best *distance* indicator value among 20 runs. It can be seen that for each method, as expected, the solutions generated in the learning phase have a wide spread in the objective space, and the solutions obtained in the decision phase are more densely distributed in a small region. This implies that the ADM realized a wide exploration in the learning phase and a focused exploitation in the decision phase when it worked with each reference point based method. Now let us focus on the ability of each method in converging to the MPS. We can find in Fig. 14 that the solutions obtained by the reference point method, g-NSGA-II, and r-NSGA-II in the decision phase locate around the MPS, while the solutions generated by R-NSGA-II deviate from the MPS as a whole. This helps explain why R-NSGA-II ranked the last in terms of the *distance* indicator in Table 13. In order to see more clearly the closeness of solutions to the MPS through visualization, Fig. 15 displays only the solutions constrained in the neighbor of the MPS among the 24 solutions. Generally, g-NSGA-II found more solutions close to the MPS than the other methods, while R-NSGA-II seldom obtained solutions near the MPS.

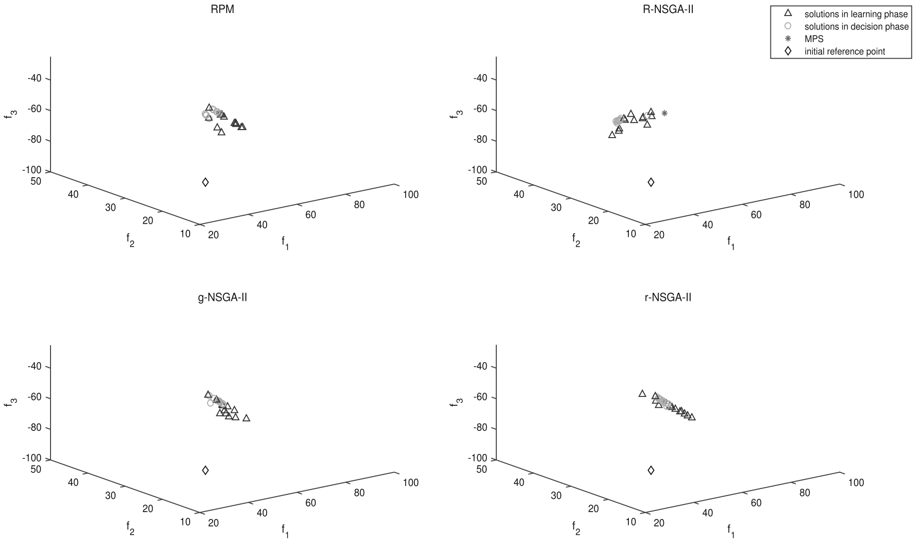


Fig. 14 Solutions obtained by each reference point based method in a single run

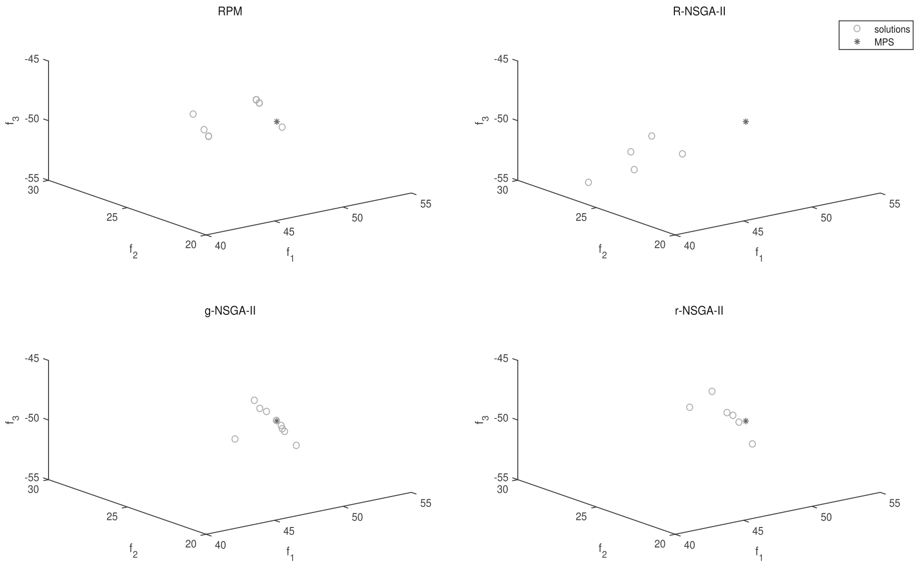


Fig. 15 Solutions obtained by each method around the MPS in a single run

6 Conclusions

In this paper, we have proposed an ADM to compare reference point based interactive methods quantitatively. Owning to a modular structure, the ADM can achieve automatic interaction with the methods to be compared and automatic evaluation of the final results. No human DMs are involved in the process. Conducting a series of experiments on multiple benchmark test problems and a water resources planning problem, we have demonstrated how the proposed ADM can be used to compare reference point based methods. The performance differences among different methods have been captured by analyzing the experimental results.

In addition to the ADM proposed in this paper, the modular structure we present can also be utilized to design other types of ADMs. Our future research will focus on building more ADMs with different abilities for a more comprehensive comparison of reference point based methods. For example, we intend to build ADMs which are able to generate multiple reference points at the same time.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10898-022-01230-3.

Acknowledgements We would like to thank the International Graduate Exchange Program of Beijing Institute of Technology, the National Outstanding Youth Talents Support Program (Grant 61822304), the Basic Science Center Programs of NSFC (Grant 62088101), the Shanghai Municipal Science and Technology Major Project (2021SHZDZX0100), the Shanghai Municipal Commission of Science and Technology Project (19511132101), and the Academy of Finland (Grant 287496) for the financial support. This research is related to the thematic research area DEMO jyu.fi/demo of the University of Jyväskylä.

Data availability The datasets generated during the current study are available from the corresponding author on reasonable request.

References

1. Afsar, B., Miettinen, K., Ruiz, A.B.: Artificial decision maker for comparing reference point based interactive evolutionary multiobjective optimization methods. In: Ishibuchi, H., Zhang, Q., Cheng, R., Li, K., Li, H., Wang, H., Zhou, A.: (eds.) Evolutionary Multi-Criterion Optimization, 11th International Conference, EMO 2021, Proceedings, pp. 619–631 (2021)
2. Afsar, B., Ruiz, A.B., Miettinen, K.: Comparing interactive evolutionary multiobjective optimization methods with an artificial decision maker. Complex Intell. Syst. https://doi.org/10.1007/s40747-021-00586-5
3. Aksoy, Y., Butler, T.W., Minor, E.D.: Comparative studies in interactive multiple objective mathematical programming. Eur. J. Oper. Res. **89**(2), 408–422 (1996)
4. Barba-González, C., Ojalehto, V., García-Nieto, J., 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., et al. (eds.) Proceedings of the International Conference on Parallel Problem Solving from Nature, pp. 274–285. Springer, Berlin (2018)
5. Bechikh, S., Kessentini, M., Said, L.B., Ghédira, K.: Preference incorporation in evolutionary multiobjective optimization: a survey of the state-of-the-art. Adv. Comput. **98**, 141–207 (2015)
6. Belton, V., Branke, J., Eskelinen, P., Greco, S., Molina, J., Ruiz, F., Słowiński, R.: Interactive multiobjective optimization from a learning perspective. In: Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.) Multiobjective Optimization: Interactive and Evolutionary Approaches, pp. 405–433. Springer, Berlin (2008)
7. Branke, J., Deb, K., Miettinen, K., Słowiński, R. (eds.): Multiobjective Optimization: Interactive and Evolutionary Approaches. Springer (2008)
8. Buchanan, J.T.: An experimental evaluation of interactive MCDM methods and the decision making process. J. Oper. Res. Soc. **45**(9), 1050–1059 (1994)
9. Chen, L., Xin, B., Chen, J.: Interactive multiobjective evolutionary algorithm based on decomposition and compression. SCIENCE CHINA Inf. Sci. **64**(10), 202201 (2021)
10. Chen, L., Xin, B., Chen, J., Li, J.: A virtual-decision-maker library considering personalities and dynamically changing preference structures for interactive multiobjective optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 636–641. IEEE (2017)
11. Corner, J.L., Buchanan, J.T.: Experimental consideration of preference in decision making under certainty. J. Multi-Criteria Decis. Anal. **4**(2), 107–121 (1995)
12. Corner, J.L., Buchanan, J.T.: Capturing decision maker preference: experimental comparison of decision analysis and MCDM techniques. Eur. J. Oper. Res. **98**(1), 85–97 (1997)
13. Deb, K., Kumar, A.: Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, pp. 781–788. ACM (2007)
14. Deb, K., Kumar, A.: Light beam search based multi-objective optimization using evolutionary algorithms. In: Proceedings of the IEEE Congress on Evolutionary Computation, pp. 2125–2132. IEEE (2007)
15. Deb, K., Miettinen, K., Chaudhuri, S.: Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. IEEE Trans. Evol. Comput. **14**(6), 821–841 (2010)
16. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
17. Deb, K., Sundar, J., Udaya Bhaskara Rao, N., Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithms. Int. J. Comput. Intell. Res. **2**(3), 273–286 (2006)
18. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization, pp. 105–145. Springer, Berlin (2005)
19. Gardiner, L.R., Vanderpooten, D.: Interactive multiple criteria procedures: some reflections. In: Clímaco, J. (ed.) Multicriteria Analysis, pp. 290–301. Springer, Berlin (1997)

20. Giagkiozis, I., Purshouse, R.C., Fleming, P.J.: An overview of population-based algorithms for multi-objective optimisation. Int. J. Syst. Sci. **46**(9), 1572–1599 (2015)
21. Hwang, C.L., Masud, A.S.M.: Multiple Objective Decision Making—Methods and Applications: A State-of-the-Art Survey. Springer, Berlin (1979)
22. Jaszkiewicz, A., Słowiński, R.: The 'Light Beam Search' approach—an overview of methodology applications. Eur. J. Oper. Res. **113**(2), 300–314 (1999)
23. Korhonen, P.J., Laakso, J.: A visual interactive method for solving the multiple criteria problem. Eur. J. Oper. Res. **24**(2), 277–287 (1986)
24. Larichev, O.I.: Cognitive validity in design of decision-aiding techniques. J. Multi-Criteria Decis. Anal. **1**(3), 127–138 (1992)
25. Li, M., Yang, S., Liu, X.: Diversity comparison of Pareto front approximations in many-objective optimization. IEEE Trans. Cybern. **44**(12), 2568–2584 (2014)
26. López-Ibáñez, M., Knowles, J.: Machine decision makers as a laboratory for interactive EMO. In: Gaspar-Cunha, A., Henggeler, A., Coello, C.A.C. (eds.) Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015, Proceedings, pp. 295–309. Springer, Berlin (2015)
27. Luque, M., Miettinen, K., Ruiz, A.B., Ruiz, F.: A two-slope achievement scalarizing function for interactive multiobjective optimization. Comput. Oper. Res. **39**(7), 1673–1681 (2012)
28. Malakooti, B., Ravindran, A.: Experiments with an interactive paired comparison simplex method for MOLP problems. Ann. Oper. Res. **5**(3), 575–597 (1985)
29. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer, Dordrecht (1999)
30. Miettinen, K., Hakanen, J., Podkopaev, D.: Interactive nonlinear multiobjective optimization methods. In: Greco, S., Ehrgott, M., Figueira, J.R. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, 2nd edn., pp. 927–976. Springer, Berlin (2016)
31. Miettinen, K., Mäkelä, M.M.: On scalarizing functions in multiobjective optimization. OR Spectrum **24**(2), 193–213 (2002)
32. Miettinen, K., Mäkelä, M.M., Kaario, K.: Experiments with classification-based scalarizing functions in interactive multiobjective optimization. Eur. J. Oper. Res. **175**(2), 931–947 (2006)
33. 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 (2008)
34. Molina, J., Santana, L.V., Hernández-Díaz, A.G., Coello, C.A.C., Caballero, R.: g-dominance: Reference point based dominance for multiobjective metaheuristics. Eur. J. Oper. Res. **197**(2), 685–692 (2009)
35. Mote, J., Olson, D.L., Venkataramanan, M.A.: A comparative multiobjective programming study. Math. Comput. Model. **10**(10), 719–729 (1988)
36. Nakayama, H.: Aspiration level approach to interactive multi-objective programming and its applications. In: Pardalos, P., Siskos, Y., Zopounidis, C. (eds.) Advances in Multicriteria Analysis, pp. 147–174. Springer, Berlin (1995)
37. Ojalehto, V., Podkopaev, D., Miettinen, K.: Towards automatic testing of reference point based interactive methods. In: Handl, J., et al. (eds.) Proceedings of the International Conference on Parallel Problem Solving from Nature, pp. 483–492. Springer, Berlin (2016)
38. Podkopaev, D., Miettinen, K., Ojalehto, V.: An approach to the automatic comparison of reference point-based interactive methods for multiobjective optimization. IEEE Access **9**, 150037–150048 (2021)
39. Reeves, G.R., Gonzalez, J.J.: A comparison of two interactive MCDM procedures. Eur. J. Oper. Res. **41**(2), 203–209 (1989)
40. Ruiz, A.B., Luque, M., Miettinen, K., Saborido, R.: An interactive evolutionary multiobjective optimization method: Interactive WASF-GA. In: Gaspar-Cunha, A., Antunes, C.H., Coello, C.A.C. (eds.) Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015, Proceedings, Part II, pp. 249–263. Springer, Berlin (2015)
41. Ruiz, A.B., Saborido, R., Luque, M.: A preference-based evolutionary algorithm for multiobjective optimization: the weighting achievement scalarizing function genetic algorithm. J. Global Optim. **62**(1), 101–129 (2015)
42. Ruiz, F., Luque, M., Miguel, F., del Mar Muñoz, M.: An additive achievement scalarizing function for multiobjective programming problems. Eur. J. Oper. Res. **188**(3), 683–694 (2008)
43. Said, L.B., Bechikh, S., Ghédira, K.: The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making. IEEE Trans. Evol. Comput. **14**(5), 801–818 (2010)
44. Shin, W.S., Ravindran, A.: A comparative study of interactive tradeoff cutting plane methods for MOMP. Eur. J. Oper. Res. **56**(3), 380–393 (1992)
45. Steuer, R.E.: Multiple Criteria Optimization: Theory, Computation, and Applications. Wiley, New York (1986)

46. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
47. Szczepański, M., Wierzbicki, A.P.: Application of multiple criteria evolutionary algorithms to vector optimisation, decision support and reference point approaches. J. Telecommun. Inf. Technol. **3**, 16–33 (2003)
48. Tabatabaei, M., Hakanen, J., Hartikainen, M., Miettinen, K., Sindhya, K.: A survey on handling computationally expensive multiobjective optimization problems using surrogates: non-nature inspired methods. Struct. Multidiscip. Optim. **52**(1), 1–25 (2015)
49. Thiele, L., Miettinen, K., Korhonen, P.J., Molina, J.: A preference-based evolutionary algorithm for multi-objective optimization. Evol. Comput. **17**(3), 411–436 (2009)
50. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: PlatEMO: a MATLAB platform for evolutionary multi-objective optimization. IEEE Comput. Intell. Mag. **12**(4), 73–87 (2017)
51. Wierzbicki, A.P.: A mathematical basis for satisficing decision making. Math. Model. **3**(5), 391–405 (1982)
52. Wierzbicki, A.P.: On the completeness and constructiveness of parametric characterizations to vector optimization problems. OR Spectrum **8**(2), 73–87 (1986)
53. 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**(1), 41256–41279 (2018)
54. Yen, G.G., He, Z.: Performance metric ensemble for multiobjective evolutionary algorithms. IEEE Trans. Evol. Comput. **18**(1), 131–144 (2014)
55. Zanakis, S.H., Solomon, A., Wishart, N., Dublish, S.: Multi-attribute decision making: a simulation comparison of select methods. Eur. J. Oper. Res. **107**(3), 507–529 (1998)
56. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: a survey of the state of the art. Swarm Evol. Comput. **1**(1), 32–49 (2011)
57. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: empirical results. Evol. Comput. **8**(2), 173–195 (2000)
58. Zujevs, A., Eiduks, J.: New decision maker model for multiobjective optimization interactive methods. In: Butleris, R., et al. (eds.) Proceedings of the International Conference on Information and Software Technologies, pp. 51–58. Kaunas Technologija, Kaunas (2011)