An Approach to the Automatic Comparison of Reference Point-Based Interactive Methods for Multiobjective Optimization

DMITRY PODKOPAEV[1], **KAISA MIETTINEN**[2], **AND VESA OJALEHTO**[2]
[1]Systems Research Institute, Polish Academy of Sciences, 01-447 Warsaw, Poland
[2]University of Jyvaskyla, Faculty of Information Technology, FI-40014 University of Jyvaskyla, Finland

Corresponding author: Kaisa Miettinen (kaisa.miettinen@jyu.fi)

ABSTRACT Solving multiobjective optimization problems means finding the best balance among multiple conflicting objectives. This needs preference information from a decision maker who is a domain expert. In interactive methods, the decision maker takes part in an iterative process to learn about the interdependencies and can adjust the preferences. We address the need to compare different interactive multiobjective optimization methods, which is essential when selecting the most suited method for solving a particular problem. We concentrate on a class of interactive methods where a decision maker expresses preference information as reference points, i.e., desirable objective function values. Comparison of interactive methods with human decision makers is not a straightforward process due to cost and reliability issues. The lack of suitable behavioral models hampers creating artificial decision makers for automatic experiments. Few approaches to automating testing have been proposed in the literature; however, none are widely used. As a result, empirical performance studies are scarce for this class of methods despite its popularity among researchers and practitioners. We have developed a new approach to replace a decision maker to automatically compare interactive methods based on reference points or similar preference information. Keeping in mind the lack of suitable human behavioral models, we concentrate on evaluating general performance characteristics. Such an evaluation can partly address the absence of any tests and is appropriate for screening methods before more rigorous testing. We have implemented our approach as a ready-to-use Python module and illustrated it with computational examples.

INDEX TERMS Decision making, interactive systems, multiobjective optimization, optimization, optimization methods, testing.

I. INTRODUCTION

Multiobjective optimization problems represent real-life situations where a *decision maker* (DM) needs to find the most preferred solution in the presence of several conflicting objective functions. Because of this conflict, a feasible solution optimizing all objective functions simultaneously does not exist. Instead, one can identify so-called Pareto optimal solutions, i.e., feasible solutions, where improving any of the objectives is impossible without impairing at least one of the others. We need additional information for comparing Pareto optimal solutions with each other and choosing the most

preferred one. This information, referred to as *preference information*, can be obtained from the DM.

The class of interactive multiobjective optimization methods ([16], [22], [25], [31]) has many applications in business and industry (see, e.g., [8], [13], [32], [37]). Their main idea is to iterate between the DM and the method until the most preferred solution has been found. The DM provides some *preference information*; the method utilizes this information to generate one or few Pareto optimal solutions and presents it/them to the DM. Iterations are repeated until a stopping criterion is met. This scheme has two main advantages. First, generating only a small number of solutions in each iteration limits the computational cost compared to deriving a good representation of the whole Pareto optimal set, which may

---

The associate editor coordinating the review of this manuscript and approving it for publication was Genoveffa Tortora.

be impossible in a reasonable time (if the problem is computationally complex). Secondly, the cognitive load set on the DM is low because only a limited amount of information is considered at a time.

In interactive methods, the DM can gradually understand the relation between their preference information and achievable solutions. Observing the DMs' behavior in real-life situations has resulted in distinguishing two phases of interactive solution processes (e.g., [25]). In the *learning phase*, the DM aims at understanding the problem by exploring different parts of the Pareto optimal set to identify a region of interest. In the *decision phase*, the search concentrates on this region to find the final solution.

There are many interactive methods for multiobjective optimization (see, e.g., [16], [22], [23], [31]). Testing their strengths and weaknesses and comparing the performance of particular methods is essential for both theory development and practical applications. By design, testing any interactive method requires the involvement of a DM. Some papers describe experiments with human DMs for evaluating different aspects of interactive methods or their components. In those experiments, one or few test runs for each studied method were performed with multiple (usually dozens of) participants. Comprehensive reviews of such studies can be found in [2] and [28].

Experimental studies of interactive methods involving human DMs are rather fragmented due to several issues. The first one is the high cost of involving human participants in the experiment, limiting the amount of data obtained and the variety of methods and research questions that can be addressed. Secondly, it is even more costly and difficult to hire participants who would be highly qualified experts in the fields where the considered methods are applied. The lack of such participants undermines the results' applicability to real-world cases since non-qualified participants may behave differently during the solution process. In addition, differences in motivation and responsibility of DMs between experimental and real-life settings may affect the results in unpredictable ways. Last but not least, the complexity of human nature, and the variation of human behavior across individuals, time spans, and test environments, make the uniformity of experimental settings hard to control. For example, learning about the problem may significantly change a DM's behavior when using a second method, for which reason test results with the same person when solving similar problems may depend on the order of applying the methods.

A natural way of addressing some of these issues is to simulate the DM's responses to conduct experiments without humans. We refer to this as an *artificial DM* (ADM). This ADM should provide preference information in a format that is appropriate for the tested methods. Thus, different types of methods may require constructing different ADMs. In this regard, we can classify interactive methods into two types [22], [31]: non-ad hoc methods, where the DM can be easily replaced with a utility or value function, and ad hoc methods, where such a replacement is problematic.

We can test non-ad hoc methods using utility or value functions (see, e.g., references in [3] and [36]), but a large class of ad hoc methods has been left behind, among them the important class of reference point-based methods (see, e.g., [39], [41]). Researchers and practitioners use such methods widely (see, e.g., [23], [43]), and this imbalance is this paper's motivation.

Reference point-based interactive methods utilize preference information provided as desirable aspiration levels forming a reference point and possibly reservation levels forming another reference point (or weights of objectives). Expressing preference information in this way is in line with the concept of "satisficing" ([30], [40]), which is regarded as cognitively undemanding for the DM. However, creating an ADM that expresses preference information in terms of reference points is not a trivial task. Furthermore, unlike utility functions or preference relations, which have been studied from the behavioral perspective, reference point-based preference models lack formal theories connecting them with human psychology.

In some works, the interaction between the DM and the method is simulated for testing purposes (see, e.g., [7], [20], [33]). However, only in a small number of papers simulation algorithms are proposed, which are suitable for testing reference point-based methods. In [34], a reference point in each iteration is set relative to the Pareto optimal objective vector obtained in the previous iteration. The new reference point is obtained using the gradient of a value function. In [35], the reference point is selected among modified objective vectors. The value function is used for selecting one of them. In [44], the reference point is obtained as a solution to a computationally complex problem. In [27], reference points are generated by adjusting aspiration levels in each iteration, considering priorities of objectives, and involving randomness. Reference [4] generates reference points based on particle swarm optimization. In [1], an artificial DM is developed for simultaneous comparison of several interactive reference point based evolutionary multiobjective optimization methods. Different mechanisms are proposed for the learning and the decision phases to generate reference points. Furthermore in [15], reference points are generated to test a specific interactive method for solving a bi-objective inventory routing problem.

So far, none of the mentioned approaches has been applied or further elaborated outside the scope of the article where it was first presented. A possible reason may be the lack of a formal model of the ADM per se. Without such a model, it is hard to tell to what extent test results are related to the actual performance of methods in real-life settings. Because of the difficulties with testing reference point-based methods, publications of new methods never include performance evaluations. All mentioned works on method comparisons were conducted post-factum, in some cases years after the methods had been first proposed.

In this paper, we develop a new ADM to test and compare reference point-based interactive methods without human involvement. We derive the ADM's behavior from some

rational assumptions about utilizing information provided by the method. The ADM cannot fully replace human DMs due to the lack of comprehensive behavioral models and difficulties in formalizing the desirable properties of interactive methods [2]. However, it allows evaluating some performance characteristics which are relevant in the decision-making context. Similar to using the Pareto dominance relation for removing inadequate solutions, our ADM can be used as a preliminary filter before the final comparison of interactive methods involving humans.

As mentioned, e.g., in [2], better means for comparing and assessing interactive methods are needed. We address this challenge and summarize our main novel contributions as follows:

- We introduce a new ADM for comparing interactive reference point-based methods. Except for the type of preference information, it does not set any other limitations to the nature of the methods compared. Thus, our ADM enables comparing ad hoc methods and, in this respect, fills a gap in the literature.
- We build the ADM on rational assumptions utilizing information that methods compared provide. This enables evaluating how well methods support the exploration of the objective space.
- We offer an open Python implementation of the ADM, which makes it easily accessible. With it, a) people publishing new interactive methods can compare the new to old ones, and b) people looking for a suitable method for a problem at hand can filter viable candidate methods.

This paper is organized as follows. In the next section, we summarize basic concepts used, including a general scheme of interactive methods. In Section III, we present our ADM together with concrete guidelines for connecting it with a method and conducting tests. We present some exemplary results of computational tests in Section IV and conclude in Section V with a discussion.

II. MULTIOBJECTIVE PROBLEM AND REFERENCE POINT-BASED METHODS

We consider *multiobjective optimization problems* formulated as follows:

$$\text{minimize } (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T$$
$$\text{subject to } \mathbf{x} = (x_1, \ldots, x_n)^T, \quad (1)$$

with *decision vectors* $\mathbf{x} \in S \subseteq \mathbb{R}^n$, where S is a nonempty compact set of *feasible solutions*, $k \geq 2$ is the number of objective functions, and $f_i: S \to \mathbb{R}$, $i = 1, \ldots, k$ are the objective functions. For any $\mathbf{x} \in S$, the vector $\mathbf{f}(\mathbf{x}) := (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T \in \mathbb{R}^k$ consisting of objective (function) values is called an *objective vector*, $\mathbb{R}^k$ is referred to as the *objective space*, and $\mathbb{R}^n$ as the *decision space*. The image of S in the objective space is defined by $\mathbf{f}(S) = \{\mathbf{f}(\mathbf{x}): \mathbf{x} \in S\}$.

Solving the problem means finding a feasible solution, which is the most preferred for the DM − a person or a stakeholder who knows the substance of the problem, can

provide preference information, and is responsible for the final solution. The DM compares feasible solutions solely based on their objective vectors, and from the DM's point of view, solving the problem is equivalent to finding the most preferred objective vector in $\mathbf{f}(S)$.

Since the DM prefers smaller objective values to larger ones, the search of the most preferred solution is limited to the set of *Pareto optimal* solutions

$$P = \left\{ \mathbf{x} \in S: \ \left(\mathbf{f}(\mathbf{x}) + \mathbb{R}^k_- \right) \cap \mathbf{f}(S) = \emptyset \right\},$$

where $\mathbb{R}^k_- = \{\mathbf{z} \in \mathbb{R}^k: \mathbf{z} \neq \mathbf{0}, z_i \leq 0 \text{ for all } i = 1, \ldots, k\}$. The objective vector $\mathbf{f}(\mathbf{x})$ corresponding to a Pareto optimal solution $\mathbf{x}$ is called a *Pareto optimal objective vector*, and the set of all such objective vectors is referred to as the *Pareto front*.

Let us also define an *ideal objective vector* and a *nadir objective vector*, respectively, as $\mathbf{z}^\star = (z_1^\star, \ldots, z_k^\star)^T$ where $z_i^\star = \min_{\mathbf{x} \in P} f_i(\mathbf{x})$, $i = 1, \ldots, k$, and $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, \ldots, z_k^{\text{nad}})^T$ where $z_i^{\text{nad}} = \max_{\mathbf{x} \in P} f_i(\mathbf{x})$, $i = 1, \ldots, k$. Note that calculating the ideal objective vector is reduced to separate minimization of each objective function in S. The nadir objective vector is more difficult to obtain, so it is usually estimated (see, e.g., [9], [22] and references therein).

Most reference point-based interactive multiobjective optimization methods[1] fit into the following core structure (e.g., GUESS method [6], an interactive evolutionary approach [10], the NIMBUS method [24], [26]; see also references in [23], [25]).

Step 0. Initialization of the method. Present some information about the problem to the DM.
Step 1. Ask the DM to specify preference information as a reference point.
Step 2. Generate one or several Pareto optimal solutions by minimizing method-specific subproblems in S and present the solution(s) to the DM.
Step 3. Ask the DM if one of the previously generated solutions is satisfactory as a final solution to the problem. If yes, stop; otherwise, go to Step 1.

Steps 1–3 represent one iteration. Step 0 is the initialization step, and the requirement to present problem-related information to the DM varies across methods. Some of them approximate the ideal and the nadir objective vectors to give the DM some idea of the ranges of objectives among Pareto optimal solutions. Other methods derive an initial Pareto optimal solution that the DM needs to express preferences in the first iteration. Without loss of generality, we assume that the information presented to the DM in Step 0 includes at least approximations of the ideal and nadir objective vectors.

Reference point-based methods derive Pareto optimal solution(s) in each iteration by solving scalarized problems. A special class of such problems is based on achievement scalarizing functions (ASFs) [40], [41]. There are many variants of ASFs (see, e.g., [39]–[41]) and different ways of

---

[1]This core structure is also valid for many other interactive multiobjective optimization methods.

expressing preference information in terms of ASF parameters (see, e.g., [19], [21]). All ASFs proposed in the literature can be divided into two types with respect to their number of parameters: either one or two k-dimensional vectors. Both types involve a reference point composed of aspiration levels for k objectives: $\mathbf{z}^{\text{asp}} = \left(z_1^{\text{asp}}, \ldots, z_k^{\text{asp}}\right)$. The aspiration levels represent desirable objective function values. ASFs of the second type incorporate another vector of k components: either positive weights of objectives $\lambda_1, \ldots, \lambda_k$, or reservation levels forming a second reference point $\mathbf{z}^{\text{res}} = \left(z_1^{\text{res}}, \ldots, z_k^{\text{res}}\right)$, where $z_i^{\text{res}} > z_i^{\text{asp}}$, $i = 1, \ldots, k$. Reservation levels represent objective function values the DM prefers to avoid.

We consider the following subproblem involving an ASF:

$$\text{minimize} \max_{i=1,\ldots,k} \lambda_i \left(f_i(\mathbf{x}) - z_i^{\text{asp}}\right) + \rho(\mathbf{x})$$
$$\text{subject to } \mathbf{x} \in S, \tag{2}$$

where $\rho(\mathbf{x})$ is a linear augmentation term for ensuring Pareto optimality of any derived solution.[2] If vector $\mathbf{z}^{\text{res}}$ is provided, then according to [18], [19], weights can be calculated by

$$\lambda_i = 1/\left(z_i^{\text{res}} - z_i^{\text{asp}}\right), \; i = 1, \ldots, k. \tag{3}$$

Another popular approach to specifying weights is by using the formula

$$\lambda_i = 1/\left(z_i^{\text{nad}} - \hat{z}_i\right), \; i = 1, \ldots, k, \tag{4}$$

where $\hat{z}_i = z_i^{\star} - \mu$, $i = 1, \ldots, k$, are components of a so-called utopian objective vector, and μ is a negligibly small positive number. Our approach to replace a DM can provide preference information in any form suitable for either of ASF types described above.

To summarize the interaction between the DM and any reference point-based method focusing on the exchange of information, from the method's point of view, the DM is involved at four different occasions:

- in Step 0, the method passes the initial information about the problem, which includes the ideal and the nadir objective vectors, to the DM;
- in Step 1, the method receives preference information from the DM;
- in Step 2, the method passes to the DM the set of derived Pareto optimal solutions;
- in Step 3, the method receives from the DM a signal to stop the solution process.

It is natural to assume that the DM saves the information about all previously derived Pareto optimal solutions in a *solution pool*. This allows the DM to select the most preferred solution from among those derived in the last iteration and from among previously obtained solutions.[3]

---

[2]A small augmentation term ensures that a solution to (2) is properly Pareto optimal [40], a property stronger than Pareto optimality. For simplicity, we do not use it in this paper. It can be added to any ASF without affecting the validity of the results.

[3]Note that few methods include this option explicitly; among those are, e.g., [17], [24].

III. THE DESIGN OF THE ADM

The ADM is characterized by a choice function, which selects a set of most preferred objective vectors from any given set of Pareto optimal objective vectors. It can be interpreted as a steady, complete model of the ADM's preferences. To make this model operational, we represent it as a value function at the expense of imposing some assumptions. Such a representation is common among other approaches to automated testing of interactive methods ([3], [34]–[36]).

It is realistic to assume that we cannot directly apply the choice function of the ADM to locate the most preferred objective vector (otherwise, there would be no need to use an interactive method). However, the ADM can apply the choice function for selecting the most preferred objective vector among any finite, explicitly given set. Communication with the interactive method is the only means of obtaining information about the problem. The ADM constructs knowledge about the Pareto optimal set to efficiently utilize this information and uses it to guide the search.

We structure the ADM in three parts and describe them in the following three subsections. The parts are the value function characterizing the ADM's preferences (the steady part), the representation of knowledge about the Pareto optimal set (the current context), and the mechanism for generating preference information based on these two (the preference generator). This structure is borrowed from [27], which fact constitutes the only similarity with the reference.

A. THE STEADY PART

The choice function characterizing an ADM is (see, e.g., [29]):

$$C \colon 2^{\mathbb{R}^k} \to 2^{\mathbb{R}^k}. \tag{5}$$

Since it is only applied to choosing from finite sets, we redefine the function domain as the set of all finite subsets of $\mathbb{R}^k$ without loss of generality:

$$C \colon \mathcal{R} \to \mathcal{R}, \tag{6}$$

where $\mathcal{R} = \left\{Y \subset \mathbb{R}^k : |Y| < \infty\right\}$, Y are sets of objective vectors and $|Y|$ denotes the cardinality of Y.

Let us assume that the choice function satisfies the weak axiom of revealed preferences (or any other equivalent rationality axiom in [29]):

$$\left(\mathbf{z}, \mathbf{z}' \in A \cap B \;\&\; \mathbf{z} \in C(A) \;\&\; \mathbf{z}' \in C(B)\right) \implies \mathbf{z} \in C(B).$$

In other words, if two elements are both most preferred in some set, then there are no circumstances where one element is preferred over another. As a direct consequence, we have the following statement [29]:

Statement 1. There exists a linear order (a complete, transitive and antisymmetric binary relation) $\succeq$ on $\mathbb{R}^k$, which represents (6):

$$C(Y) = \left\{\mathbf{z} \in Y : \mathbf{z} \succeq \mathbf{z}' \text{ for any } \mathbf{z}' \in Y \setminus \{\mathbf{z}\}\right\},$$

where $Y \subset \mathbb{R}^k$, $|Y| < \infty$.

Furthermore, let us assume that the ADM can be applied to solve only a countable set of problems, which implies that the binary relation is defined on a countable set of objective vectors. This technical assumption is required for applying another theoretical result of the rational choice theory [5], stating that *any complete and transitive binary relation on a countable set has a value representation*. Thus, we have the following statement:

Statement 2. There exists a value function[4] $v : \mathbb{R}^k \to \mathbb{R}$ such that for any $y, y' \in \mathbb{R}^k$ we have $y \succeq y' \Leftrightarrow v(y) \geq v(y')$.

Summing up, from reasonable assumptions we obtain Statements 1 and 2, saying that the model of steady preferences initially defined as the choice function C can be represented as a value function $v : \mathbb{R}^k \to \mathbb{R}$:

$$C(Z) = \arg\max\{v(\mathbf{z}) : \mathbf{z} \in Z\} \text{ for any } Z \subset \mathbb{R}^k.$$

B. THE CURRENT CONTEXT
By the current context, we mean the knowledge about the Pareto front, which the ADM constructs from information accumulated during the solution process. As said before, the ADM receives the following information from the interactive method: the ideal and nadir objective vectors at the beginning of the solution process and a set of Pareto optimal objective vectors derived at each iteration. This information is included in the current context and used by the ADM for deriving information about the potential location of new Pareto optimal objective vectors that have not been generated yet. We shall now describe this in detail.

Given two objective vectors $\mathbf{z}$ and $\mathbf{z}'$ such that $z_i \leq z_i'$ for any $i = 1, \ldots, k$, we define a box as the Cartesian product of closed intervals:

$$\beta\left[\mathbf{z}, \mathbf{z}'\right] = \prod_{i=1,\ldots,k} [z_i, z_i']$$

and call $\mathbf{z}$ and $\mathbf{z}'$ its *minimal and maximal points*, respectively. Using this notation, we define the *initial region* as follows:

$$B = \beta\left[\mathbf{z}^\star, \mathbf{z}^{\text{nad}}\right].$$

It is the box in the objective space containing the whole Pareto front. The role of the initial region in constructing the current context is to limit the area of the potential location of new Pareto optimal objective vectors.

Let j be an iteration number. After completing the iteration, the *solution pool* $\hat{Y}^j$ is defined by

$$\hat{Y}^j = \bigcup_{l=1,\ldots,j} Y^l, \tag{7}$$

where Y^l is the set of Pareto optimal objective vectors derived in iteration l. In addition, we set $\hat{Y}^0 = \emptyset$.

Now we introduce a set of objective vectors called the *potential region* R^j:

$$R^j = B \setminus \bigcup_{\mathbf{z} \in \hat{Y}^j} \left(\mathbf{z} + \mathbb{R}^k_< \cup \mathbf{z} - \mathbb{R}^k_<\right),$$

[4]Note that the term *utility function* is often used in the rational choice theory instead of a *value function*.

where $\mathbb{R}^k_< = \left\{\mathbf{z} \in \mathbb{R}^k : z_i < 0, i = 1, \ldots, k\right\}$. It is easy to see that if a Pareto optimal objective vector derived in the initial region during the iteration $j + 1$ is new (i.e. it does not belong to $\hat{Y}^j$), then it belongs to R^j: otherwise it would either dominate or be dominated by one of the previously derived Pareto optimal solutions. Therefore, information about the potential region is relevant when searching for new Pareto optimal solutions.

We represent the potential region in each iteration as a collection of boxes in the objective space. Once the ideal and nadir objective vectors have been obtained in Step 0, the ADM *initializes* the potential region as one box, namely the initial region: $R^0 = \{B\}$. After having received the set of Pareto optimal objective vectors Y^j derived in Step 2 of the j-th iteration, the ADM *updates* the potential region. It subtracts the set of vectors which dominate or are dominated by the elements of Y^j from the area comprised of all the boxes.

Let us define a set of 2^k orthants of the objective space:

$$\Omega = \left\{\Lambda(\mathbf{b}) : \mathbf{b} = (b_1, \ldots, b_k) \in \{-1, 1\}^k\right\},$$
$$\Lambda(\mathbf{b}) = \left\{(y_1, \ldots, y_k)^T \in \mathbb{R}^k : b_i y_i \leq 0, i = 1, \ldots, k\right\}.$$

Algorithm 1 can be used for updating the potential region, where we denote any box included in the potential region by β and the current iteration number by j.

It is important to note that if $Y^j \subseteq \hat{Y}^{j-1}$ (no new Pareto optimal objective vectors have been derived in iteration $j, j > 1$), the current context does not change according to Algorithm 1. On the other hand, the information about the absence of new solutions should be taken into account by the ADM. To address that, we introduce a small modification to the potential region. It is described at the end of Subsection III-C since its justification refers to an element of the preference generator.

C. THE PREFERENCE GENERATOR
In each iteration j, the search for the most preferred solution is naturally narrowed down to the potential region. We determine a box in R^j, where such a solution can most likely be located, and to explore this box, we ask the method to derive Pareto optimal solution(s) in it.

Without complete knowledge about the Pareto optimal set, determining which box is most promising in the above sense is not possible. We propose considering the objective vector in the center of a box (called the midpoint) as a neutral guess about its content and applying the ADM's steady preference model for selecting one among the midpoints of all boxes in R^j. Then the ADM instructs the method to derive Pareto optimal solution(s) in the box by setting the reference point at the midpoint of the selected box. We set the vector of reservation levels at the maximal point of the box, reflecting the fact that it represents the upper bounds on components of Pareto optimal objective vectors, which can be possibly derived in this box. We formalize this procedure of generating preference information as follows.

Algorithm 1: Updating the Potential Region

Initialize $R^j := R^{j-1}$.

For each Pareto optimal objective vector $\mathbf{y} \in Y^j \setminus \hat{Y}^{j-1}$:

> **For** each box $\beta \in R^j$:
>
>> Check the relation between β and the set $\mathbf{y} + \mathbb{R}^k_< \cup \mathbf{y} - \mathbb{R}^k_<$.
>>
>> **If** $\beta \subset \left(\mathbf{y} + \mathbb{R}^k_< \cup \mathbf{y} - \mathbb{R}^k_<\right)$, then β cannot contain any new Pareto optimal objective vectors. In this case,
>>
>>> remove β from R^j.
>>
>> **If** $\beta \cap \left(\mathbf{y} + \mathbb{R}^k_< \cup \mathbf{y} - \mathbb{R}^k_<\right) = \emptyset$, then β can still contain new Pareto optimal objective vectors. In this case,
>>
>>> keep β in R^j.
>>
>> **If** $\beta \cap \left(\mathbf{y} + \mathbb{R}^k_< \cup \mathbf{y} - \mathbb{R}^k_<\right)$ is neither empty nor coinciding with β, then a part of β can still contain new Pareto optimal objective vectors, and the rest of β cannot. In this case, modify R^j in order to get rid of vectors belonging to $\beta \cap \left(\mathbf{y} + \mathbb{R}^k_< \cup \mathbf{y} - \mathbb{R}^k_<\right)$ as follows:
>>
>>> Split β into a set of boxes, where each box is the intersection of β with one of the 2^k orthants from Ω anchored at $\mathbf{y}$.
>>>
>>> Include in R^j all newly obtained boxes except the following two: $\beta \cap \left(\mathbf{y} + \mathbb{R}^k_\leq\right)$ and $\beta \cap \left(\mathbf{y} - \mathbb{R}^k_\leq\right)$, where $\mathbb{R}^k_\leq = \left\{\mathbf{z} \in \mathbb{R}^k : z_i \leq 0,\ i = 1, \ldots, k\right\}$.
>>>
>>> Remove the initial box β from R^j.

- For all boxes in the potential region, the ADM determines their midpoints forming the set

$$\Phi^j = \left\{ \mathrm{Mid}(\beta) : \ \beta := \beta\left[\mathbf{z}, \mathbf{z}'\right] \in R^j \right\}, \qquad (8)$$

where

$$\mathrm{Mid}(\beta) = \left(\frac{z_1 + z_1'}{2}, \frac{z_2 + z_2'}{2}, \ldots, \frac{z_k + z_k'}{2} \right).$$

- Using the choice function (5), the ADM selects the subset of the most preferred midpoints:

$$\Xi = C(\Phi^j). \qquad (9)$$

- The ADM chooses any box $\beta^\star$ whose midpoint belongs to Ξ, and generates preference information in terms of aspiration and reservation levels: $\mathbf{z}^{\mathrm{asp}} = \mathrm{Mid}(\beta^\star)$, $\mathbf{z}^{\mathrm{res}} = \mathbf{z}'$ where $\mathbf{z}'$ is the maximal point of $\beta^\star$.

After generating the preference information, the ADM expresses it depending on the requirements of the interactive method:

- if the method requires one reference point, then the ADM returns $\mathbf{z}^{\mathrm{asp}}$;
- if the method requires two reference points, then the ADM returns $\mathbf{z}^{\mathrm{asp}}$ and $\mathbf{z}^{\mathrm{res}}$;

- if the method requires a reference point and weights, then the ADM returns $\mathbf{z}^{\mathrm{asp}}$ and the vector of weights $(\lambda_1, \ldots, \lambda_k)$ calculated according to (3).

Since the derivation of Pareto optimal solutions is confined to the most interesting box selected by the ADM, we can address the case where $Y^j \subseteq \hat{Y}^{j-1}$ for some $j, j > 1$, i.e., no new Pareto optimal objective vectors have been derived in iteration j. We slightly amend Algorithm 1 for updating the current context described in Subsection III-B based on the following reasoning. The lack of new Pareto optimal objective vectors implies that the interior of the most interesting box selected by the ADM contains no solutions. Since this box has been explored, we should not repeat an attempt to derive solutions in it. Therefore, after completing each iteration, *the box $\beta^\star$ that was selected by the ADM is removed from the potential region*.

We illustrate the process of generating preference information described above in Fig. 1 in the case of $k = 2$. Let us assume that in some iteration, the solution pool consists of three Pareto optimal objective vectors $\mathbf{z}^1$, $\mathbf{z}^2$ and $\mathbf{z}^3$. We show the boxes comprising the potential region in gray color. The dashed curves depict isoquants of the value function representing steady preferences. Small crossed circles in the centers of the boxes mark their midpoints. It is easy to notice that the midpoint of the box between solutions $\mathbf{z}^2$ and $\mathbf{z}^3$ has the highest value of the value function. After the ADM selects this box and generates a reference point in it, the method derives a new Pareto optimal objective vector $\mathbf{z}^4$. The ADM adds this vector to the solution pool and updates the potential region by removing from it the areas $\mathbf{z}^4 + \mathbb{R}^2_<$ and $\mathbf{z}^4 - \mathbb{R}^2_<$. It divides the box under consideration into four boxes and deletes two of them as described in Subsection III-B.

D. THE SCHEME OF IMPLEMENTING TESTS USING THE ADM

Let us first describe the aspects of interactive methods which can be evaluated using the ADM. The central question is how well the method finds the most preferred solution, referred to as the *performance aspect*. We can measure it by the following two means, which are dual to each other.

- "Accuracy": given the maximum number of iterations, how close to the most preferred solution can a solution be found.
- "Convergence speed": given desirable accuracy in the above sense, how many iterations are needed to reach this accuracy.

To evaluate these means for a given problem and an ADM's steady preference model, we must first derive the most preferred solution regarding the ADM's steady preferences. We should introduce a measure of closeness of any Pareto optimal solution and the most preferred solution. We must perform a predefined number of iterations to calculate the accuracy and the minimum proximity between the derived solutions and the most preferred solution. To calculate the convergence speed, we must perform iterations until the
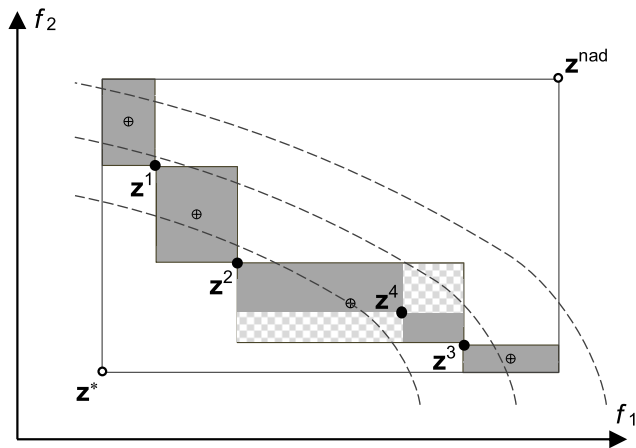
FIGURE 1. The operation of the ADM in the case $k = 2$.

proximity of one of the derived solutions falls below the desirable accuracy. The number of iterations serves as the result of calculations. It is practical to set the threshold as the maximal number of iterations to limit the time needed for conducting experiments in the latter case. If the desired accuracy has not been achieved, the convergence speed can be described as "failed to converge in the allotted number of iterations."

Another critical aspect is how well the method explores the objective space regarding the accuracy of overall information about the Pareto front. We can quantify this so-called *exploration aspect* as the hypervolume of the potential region. The smaller is the hypervolume of R^j, the less uncertainty the ADM has about the shape of the Pareto front after the i-th iteration.

We propose the following scenario of conducting experiments. We formulate the main research question by the two means describing the performance. After each run, we accompany the performance analysis with the exploration analysis and consider the hypervolume of the potential region after each iteration. The stopping criterion is formulated based on the experiment design. Depending on the performance analysis, the stopping criterion can be, for instance, a certain number of iterations or a certain closeness to the most preferred solution.

Let us now present the scheme of implementing experiments. In testing, the ADM is a black box replacing the DM in the interaction introduced in Section II. During experiments, the ADM and the method are called in turns, exchanging information: the output from the ADM serves as the input for the method and vice versa until a stopping criterion is met.

Fig. 2 summarizes how one can use an ADM consisting of the three described parts to test an interactive reference point-based method. Below, we specify the details of the interaction.

- In Step 0, the method passes initial information about the problem ($\mathbf{z}^\star$ and $\mathbf{z}^{\text{nad}}$) to the ADM. As described in Subsection III-B, current context is initialized: the

potential region R^0 as $\{\beta\left[\mathbf{z}^\star, \mathbf{z}^{\text{nad}}\right]\}$ and the solution pool $\hat{Y}^0$ as $\emptyset$.
- In Step 1, the method asks for preference information as parameters of the ASF. The preference generator creates such information based on the steady part and the current context as described in Section III-C, and passes this information to the method.
- In Step 2, the method passes to the ADM the set of Pareto optimal solutions Y^j derived in iteration j. The ADM uses it to update the current context: it is included in the solution pool and used for updating the potential region as described in Section III-B.
- In Step 3, the method asks for information related to stopping the solution process.

IV. TEST EXPERIMENTS

We have implemented the ADM in Python 3 and published the code under Mozilla Public License 2.0 on GitHub: https://github.com/industrial-optimization-group/desdeo-adm.

Our purpose is to demonstrate what meaningful information and valuable insights we can gain about interactive methods. We define basic experimental settings and, during the runs, collect information on the performance and exploration aspects introduced in Subsection III-D.

We solve one problem using two popular methods: the reference point method (RPM) [39], [40] and the synchronous NIMBUS method [24], [26]. The latter uses the classification of objectives as preference information indicating how a Pareto optimal solution (called a current solution) should be improved. We can obtain this information from a reference point if the latter does not dominate or is not dominated by the current solution. We show that the ADM can test a broader class of interactive methods by considering a classification-based method.

Below, we briefly summarize the two methods and describe how one can convert a reference point into preference information for NIMBUS. After that, we define the value function representing the steady part of the ADM, formulate an example problem and present the results of experiments.

A. REFERENCE POINT AND NIMBUS METHODS

Both methods fit into the core structure described in Section II. In Step 1 of the RPM, the DM specifies a reference point $\mathbf{z}^{\text{asp}}$. In Step 2, the method derives $k + 1$ Pareto optimal solutions as follows. First, the method solves (2) with the given reference point and fixed weights defined by (4), obtaining a Pareto optimal objective vector $\tilde{\mathbf{z}}$ (the current solution). Then, k more reference points are created by perturbing $\mathbf{z}^{\text{asp}}$, where the i-th perturbation, $i = 1, \ldots, k$, consists of adding the value $\|\mathbf{z}^{\text{asp}} - \tilde{\mathbf{z}}\|$ to the i-th component of $\mathbf{z}^{\text{asp}}$. For each of these reference points, the method solves problem (2). In this way, the DM gets a better idea of the surroundings of the current solution in the objective space. Thus, it is enough to pass the first reference point to the RPM as the preference information.
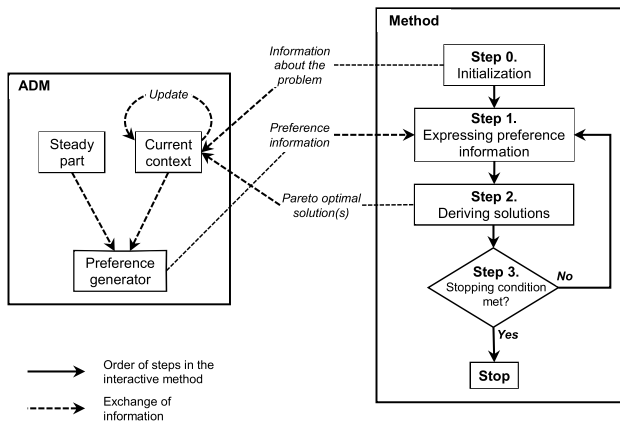
FIGURE 2. The scheme of interaction between an ADM and an interactive method.

In Step 0 of the synchronous NIMBUS method, in addition to the ideal and nadir objective vectors, a Pareto optimal solution is generated as the first current solution. It is needed for expressing preference information in the first step. Since this Pareto optimal objective vector belongs to the solution pool, in terms of the core structure of interactive methods (Section II), it should be treated as a result of an iteration rather than a part of the initialization step. Thus, in our experiments we assume that in the first iteration, the ADM provided the reference point $z^{asp} := \text{Mid}(B)$ with the components $\frac{z_i^{nad} + z_i^\star}{2}$, $i = 1, \ldots, k$, to get a neutral compromise solution [24] by solving problem (2).

In Step 1 of NIMBUS, the DM expresses preference information for the current solution $\tilde{z}$ by classifying the objective functions into up to five classes. This classification shows the desirable way of changing the objective vector component-wise to get a more preferred solution. The classes are subsets of functions f_i, $i = 1, \ldots, k$, whose values should be decreased ($i \in I^<$), should be decreased till a desirable level $\gamma_i < \tilde{z}_i$ ($i \in I^\le$), are satisfactory at the moment ($i \in I^=$), are allowed to increase till an upper bound $\varepsilon_i > \tilde{z}_i$ ($i \in I^\ge$), and are allowed to change freely ($i \in I^\diamond$).

Some of the classes may be empty; however, the following conditions should be satisfied for a classification to be feasible: $I^< \cup I^\le \ne \emptyset$ and $I^> \cup I^\diamond \ne \emptyset$. In Step 2, the method derives up to four Pareto optimal solutions by solving four different scalarized subproblems, reflecting the diversity of interpretations of the preference information. We give a simple way of converting reference points into classification following [24] in Appendix I.

B. THE STEADY PART OF THE CONSIDERED ADM

As a value function, we use the Cobb-Douglas function (see, e.g., [14], [38]). It is defined for non-negative objective values to be maximized:

$$\omega^{\bar{\alpha}}(\mathbf{z}) = \prod_{i=1,\ldots,k} z_i^{\alpha_i},$$

where $\bar{\alpha} = (\alpha_1, \ldots, \alpha_k)$ and α_i, $i = 1, \ldots, k$, are non-negative parameters. To use this function, we map the objective space of problem (1) to the objective space of a maximization problem with non-negative objective values:

$$g_i(z_i) := \frac{z_i^{nad} - z_i}{z_i^{nad} - \hat{z}_i} + 0.01, \quad i = 1, \ldots, k.$$

In other words, we normalize each component of the objective vector to the interval from 0 (the worst value) to 1 (the best value), and add a small term to avoid the marginal rates of substitution diminishing to zero. The value function representing the steady ADM's preferences is

$$v^{\bar{\alpha}}(\mathbf{z}) = \omega^{\bar{\alpha}}(\mathbf{g}(\mathbf{z})) \tag{10}$$

where $\mathbf{g}(\mathbf{z}) = (g_1(z_1), \ldots, g_k(z_k))$.

C. EXAMPLE PROBLEM

We tested the two interactive methods on a three-objective optimization problem. This number of objectives ensures that both the RPM and NIMBUS solve the same number of subproblems in each iteration. We considered the following modification of the problem formulated in [12]:

$$\begin{aligned}
\text{minimize } & (f_1(\mathbf{x}), f_2(\mathbf{x}), f_k(\mathbf{x}))^T \\
\text{subject to } & \mathbf{x} = (x_1, x_2)^T \in [0, 1]^2, \\
& f_1(x) \le 1,
\end{aligned} \tag{11}$$

where $f_1(\mathbf{x}) = \phi(x_1, x_2)$, $f_2(\mathbf{x}) = \phi(x_1, x_2 - 1)$, $f_3(\mathbf{x}) = \phi(x_1 - 1, x_2)$, $\phi(x_1, x_2) = \psi(x_1, x_2) - e^{-50\psi(x_1,x_2)}$ and $\psi(x_1, x_2) = x_1^2 + x_2^2$. For this problem, we have $\mathbf{z}^\star = (-1, -1, -1)$ and $\mathbf{z}^{nad} = (1, 2, 2)$. We have chosen this particular problem because it has a nonconvex Pareto front of a complex shape, making the solution process more challenging.

It is a nonlinear problem. The scalarized subproblems were solved using the simplicial homology global optimization (SHGO) algorithm [11] published in Github https://github.com/Stefan-Endres/shgo and available as a part of the SciPy library. It is well suited for problems of this type and has an easy-to-use interface.

D. EXPERIMENT DESIGN

We characterize an instance of the ADM by a vector $\bar{\alpha} = (\alpha_1, \alpha_2, \alpha_3)$ of parameters of the value function v defined by (10), which represents the steady part. We used each instance for solving the example problem independently by both considered methods. We performed 100 experiments with randomly generated $\bar{\alpha}$. We ran the solution process with each instance for up to 25 iterations, as this number exceeds the typical duration of real-life solution processes reported in the literature.

At the end of each iteration i, $i > 0$, after the ADM updated the current context taking the derived solutions into account, we collected the following indicators characterizing the solution process.

1) The *iteration value* defined as $\frac{v^i}{v^\star} \cdot 100\%$, where v^i is the maximum value of the value function v among the

solutions derived in iteration *i*, and $v^\star$ is the optimum value of the value function for problem (11). This indicator reflects the quality of obtained solutions and can be interpreted in terms of the proximity of the best solution among the obtained ones to the most preferred solution.

2) The volume of the potential region, which characterizes the exploration aspect.

3) The total number of Pareto optimal solutions accumulated in the solution pool. As explained later, we consider this indicator for controlling the fairness of method comparisons.

Fig. 3 illustrates the results of running the solution process for 25 iterations using both methods with the ADM characterized by $\bar{\alpha} = (1, 1, 1)$. It includes the plots of the iteration value (black lines, the left-hand axis) and volume of the potential region (gray lines, the right-hand axis) along with iterations. The indicators of the RPM and NIMBUS are depicted by solid and dashed lines, respectively. The small rhombuses mark the maximum iteration values among 25 iterations for both methods.

For the RPM, 99.2% of the optimum was achieved in the 21st iteration, and in the case of NIMBUS, 99.7% of the optimum was achieved in the 18th iteration. The ADM demonstrated the same pattern for both methods: the iteration value gradually stabilized near the highest value while the potential region volume decreased. This observation indicates that an improvement in the quality of the derived solutions goes hand in hand with the accumulation of knowledge about the Pareto front.

E. RESULTS OF EXPERIMENTS

In each of the 100 experiments, we generated the coefficients α_1, α_2, and α_3 as independent random variables uniformly distributed in the interval [1, 2]. During the test of each of the methods with the same ADM, we calculated the following indicators.

1) The maximum among iteration values achieved during the first ten iterations. This characterized accuracy, where we interpreted the obtained value in terms of the proximity to the most preferred solution. We chose ten iterations as a specific number of iterations to be completed, taking into account our experience with real-life interactive solution processes.

2) The smallest iteration number where the iteration value of 95% was reached, referred to as the iteration number of 95% level. This indicator characterized the convergence speed. Not getting a 95% level in 25 iterations was considered a failure.

3) The volume of the potential region after ten iterations characterizing the exploration power of the method.

4) The size of the solution pool after ten iterations used for controlling the fairness of experiments.

Table 1 summarizes the results of 100 experiments. The table contains the average value and the standard deviation
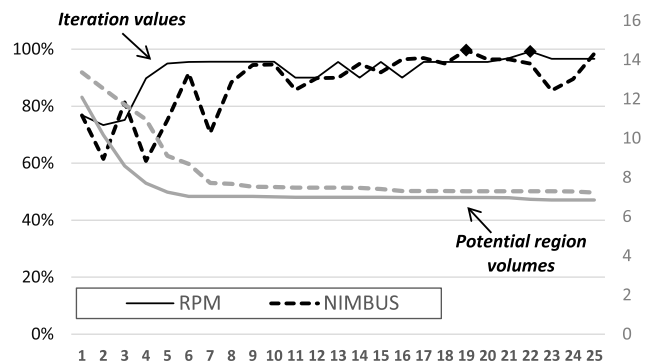

FIGURE 3. Iteration value and potential region volume during the solution process with $\bar{\alpha} = (1, 1, 1)$.

in parentheses for each of the mentioned indicators. The last column summarizes the differences in the indicators values between the RPM and NIMBUS in each experiment. On average, the RPM performed similarly to NIMBUS in terms of accuracy[5] and 21.6% faster in terms of convergence speed. The RPM also reduced the average potential region volume after the 10th iteration to 7.12. It is 10.1% less than what was done by NIMBUS (note that the volume of the initial region is equal to 18). We can view this as a better exploration of the objective space.

On the other hand, the RPM provided the ADM with, on average, 2.9 (8.6%) more solutions in 10 iterations, which can be partially explained by the special treatment of the first iteration in NIMBUS (where only one solution was derived, see Subsection IV-A). This advantage partially explains the better performance of the RPM.

Another important observation from the experiments was that the differences between the methods' performance varied significantly across ADM instances. For the maximum iteration value and the iteration number of 95% level (see the first two cells of the third column), the standard deviation of the difference was 5.7 and 2.5 times higher than the absolute average difference, respectively. In addition, in all the experiments where the RPM failed to achieve 95% iteration value, the NIMBUS method succeeded, and vice versa. In other words, the differences in those methods' performance are sensitive to the parameters of the ADM's steady preference model.

The above observation sheds light on the issue of selecting an interactive method. It appears that even if a problem instance and a class of steady preference models are given, there is not enough information to conclude which method is better. The recommendation on method selection may depend on the parameters of the preference model. This dependence undermines the possibility of universal recommendations,

---

[5]Since the maximum iteration value in ten iterations was, on average, similar for both methods, we needed to test if this difference was significant. We have tested the hypothesis that this indicator has the same statistical distribution for both methods using the nonparametric test by [42]. This hypothesis was rejected with the confidence level of 98.9%. For the remaining indicators, the confidence of rejecting analogous hypotheses was much higher (the values of *p*-statistics for this test were lower by orders of magnitude).

TABLE 1. Comparison of the reference point and NIMBUS methods in 100 experiments.

	RPM	NIMBUS	Difference
Maximum iteration value in 10 iterations	95.4% (2.9%)	94.7% (3.6%)	0.7% (4.1%)
Iteration number of 95% level	8.8 (5.4)	11.7 (5.3)	-2.5 (6.4)
Number of failures to achieve 95%	5	7	
Potential region volume after 10 iterations	7.12 (0.57)	7.92 (1.22)	-0.8 (1.2)
Size of the solution pool after 10 iterations	37.0 (2.4)	34.1 (1.4)	2.9 (2.7)

suggesting that it may be necessary to select an interactive method for each DM.

One can look for patterns in the dependence between the preference model parameters and methods' performance by conducting experiments, where parameters change systematically. As an example, we set[6] the value of α_3 to 1, and varied α_1 and α_2 independently from 1.5 to 2.5 with step size 0.2. For each of the 36 resulting parameter vectors, we ran the solution process with both the RPM and NIMBUS in the same way as in the former series of experiments. The tables in Appendix II present the main performance indicators depending on the parameter values: the maximum iteration value achieved in the first ten iterations (accuracy) and the iteration number, where a 95% iteration value has been achieved (convergence speed). We can notice that better performance of both methods in terms of both indicators was achieved for the lowest value of α_1 as well as low ratios between α_1 and α_2. However, the combinations of parameter values where each method demonstrated good performance did not completely coincide. In terms of accuracy, good performance of the RPM was also achieved for some combinations where $\alpha_2 \in \{0.7, 0.9\}$, and good performance of the NIMBUS method was also achieved for $\alpha_1 = 1.5$ and $0.7 \leq \alpha_2 \leq 1.3$. Interestingly, for the combinations $\alpha_1 \in \{1.3, 1.5\}$ and $\alpha_2 = 0.5$, despite the RPM had better performance in terms of accuracy, it had a bad performance in terms of convergence speed as the iteration value of 95% was not reached in 25 iterations. Summing up, the influence of parameters of the value function on methods' performance does not look arbitrary but demonstrates some patterns. The analysis of such patterns could be utilized for predicting methods' performance based on parameter values.

Let us note that we must judge the fairness of comparing methods individually for various experimental settings. For example, one must decide whether it is fair to conduct the same number of iterations (knowing that different methods may need different amounts of computations per iteration). Alternatively, one can apply the same number of function

[6]We do not need to vary the parameter α_3 since the value function with parameters $(\alpha_1, \alpha_2, \alpha_3)$ is equivalent to the function with parameters $(\alpha_1 + \delta, \alpha_2 + \delta, 1)$ where $\delta = 1 - \alpha_3$.

evaluations or require a similar amount of information from the ADM. For example, due to the special treatment of the initialization phase, the NIMBUS method derived fewer solutions in the first iteration than the RPM. We can consider the amount of cognitive load set on the DM as a "common denominator" for establishing the equality of experimental settings for human DMs. In the case of ADMs, we can estimate corresponding information by the size of the solution pool.

V. CONCLUSIONS

The concept of reference points is widely used for representing preference information in interactive multiobjective optimization methods. The class of reference point-based methods is important from theoretical and practical points of view but lacks comparative studies. We have proposed an approach called an ADM to enable comparing interactive reference point-based methods. One can apply it for preliminary comparison of methods of this class before involving a human DM. The transparent description of the principles underlying the ADM's operation and a ready-to-use implementation in a widely used language (Python) should make our development easy and attractive to apply.

We demonstrated how one can apply the ADM with some preliminary experiments and that it can exhibit complex operations and help draw nontrivial conclusions. For example, we were able to quantify how sensitive differences are between methods' performance to parameters of the ADM. It appears that the recommendations on method selection may require adjustment in each case. We can establish the fairness of method comparison in terms of the number of solutions provided to the ADM in the same number of iterations.

The current implementation of our approach is not suitable for comparing interactive methods where Pareto optimal solutions are approximated. If different elements of the solution pool can dominate each other, then the premise that any newly derived objective vector belongs to the potential region is violated. Algorithm 1 of updating the potential region should be revised to adapt our approach for comparing approximate methods. In the long run, it is desirable to compare methods from different classes (using different preference types), employing other ADM approaches calibrated to ensure fairness of comparing methods.

As said in the introduction, we have not intended the ADM to replace human DMs in tests of interactive methods fully. Due to the broad spectrum of human behaviors, evaluating the accuracy of the ADM should be done in the framework of a multidisciplinary study involving many experiments with real DMs in various settings. That would pave the way for further improvement of the proposed approach.

APPENDIX I
GENERATING NIMBUS-RELATED PREFERENCE INFORMATION FROM A GIVEN REFERENCE POINT z^{asp}

First, it is natural to select the current solution in iteration j from among those members of the solution pool which are

TABLE 2. Maximum iteration value in 10 iterations (values that are greater than 98% are highlighted in bold).

	0.5	0.7	0.9	1.1	1.3	1.5
0.5	**99.1%**	**100.0%**	**99.7%**	**99.8%**	**99.3%**	**98.4%**
	97.8%	**98.3%**	**98.1%**	**98.4%**	**99.1%**	**99.6%**
0.7	97.6%	**98.2%**	97.5%	**99.5%**	**99.9%**	**99.9%**
	94.9%	94.7%	94.2%	96.3%	**98.5%**	**99.3%**
0.9	97.6%	**99.5%**	96.2%	96.5%	97.5%	**99.3%**
	97.0%	95.9%	93.7%	93.9%	88.0%	95.2%
1.1	95.9%	**98.9%**	**99.0%**	95.7%	96.3%	97.1%
	91.8%	**99.7%**	93.5%	95.5%	94.9%	95.5%
1.3	93.9%	98.0%	**98.1%**	94.6%	90.8%	96.8%
	89.1%	**98.6%**	96.7%	96.5%	91.3%	94.4%
1.5	92.1%	96.8%	97.1%	94.0%	95.3%	90.2%
	86.4%	**98.7%**	**98.7%**	**99.1%**	**99.4%**	94.2%

TABLE 3. Iteration number of 95% level (the dash indicates that 95% level has not been reached in 25 iterations).

	0.5	0.7	0.9	1.1	1.3	1.5
0.5	1	1	1	1	1	1
	3	1	3	3	3	3
0.7	4	3	4	1	1	1
	11	16	13	1	3	3
0.9	3	4	5	5	5	2
	10	7	12	15	18	4
1.1	3	5	6	9	4	4
	12	8	12	9	15	8
1.3	–	5	5	16	11	8
	12	8	8	8	12	12
1.5	–	5	5	12	10	11
	12	9	7	7	9	13

the most preferred (according to the steady part):

$$\tilde{\mathbf{z}} \in C(\hat{Y}^j).$$

Secondly, each objective function is assigned to one of the five classes by consecutively applying the following rules (once an assignment is done, the remaining rules do not apply):

- if z_i^{asp} is close to $z_i^{\star}$, then assign i to $I^{<}$,
- if z_i^{asp} is close to z_i^{nad}, then assign i to $I^{\diamond}$,
- if $z_i^{\mathrm{asp}} = \tilde{z}_i$, then assign i to $I^{=}$,
- if $z_i^{\mathrm{asp}} < \tilde{z}_i$, then assign i to $I^{\leq}$ and set $\gamma_i = z_i^{\mathrm{asp}}$,
- if $z_i^{\mathrm{asp}} > \tilde{z}_i$, then assign i to $I^{\geq}$ and set $\varepsilon_i = z_i^{\mathrm{asp}}$.

The proximity of z_i^{asp} to $z_i^{\star}$ or z_i^{nad} mentioned in the first two rules means that the absolute difference between z_i^{asp} and the corresponding value is less than 1% of the difference $z_i^{\mathrm{nad}} - z_i^{\star}$.

Observe that due to the method of defining preference information (see Subsection III-C), a reference point $\mathbf{z}^{\mathrm{asp}}$ generated by the ADM always belongs to the interior of the potential region. It follows that $\mathbf{z}^{\mathrm{asp}}$ never dominates nor is dominated by any Pareto optimal objective vector. In this case, it is easy to see that the classification obtained as above is always feasible.

APPENDIX II
INDICATORS OF METHOD PERFORMANCE DEPENDING ON ADM PARAMETERS

In both Tables 2 and 3, the row and column headers represent the values of α_1 and α_2, respectively. Each cell contains two values of the performance indicator obtained in the experiment with the corresponding parameter vector $(\alpha_1, \alpha_2, 1)$: the value for the RPM above the value for the NIMBUS method.

REFERENCES

[1] B. Afsar, K. Miettinen, and A. B. Ruiz, "An artificial decision maker for comparing reference point based interactive evolutionary multiobjective optimization methods," in *Proc. 11th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, H. Ishibuchi, Q. Zhang, R. Cheng, K. Li, H. Li, H. Wang, and A. Zhou, Eds. Cham, Switzerland: Springer, 2021, pp. 619–631.

[2] B. Afsar, K. Miettinen, and F. Ruiz, "Assessing the performance of interactive multiobjective optimization methods: A survey," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–27, 2021, Art. no. 85.

[3] Y. Aksoy, T. W. Butler, and E. D. Minor, "Comparative studies in interactive multiple objective mathematical programming," *Eur. J. Oper. Res.*, vol. 89, no. 2, pp. 408–422, 1996.

[4] C. Barba-González, V. Ojalehto, J. García-Nieto, A. J. Nebro, K. Miettinen, and J. F. Aldana-Montes, "Artificial decision maker driven by PSO: An approach for testing reference point based interactive methods," in *Proc. 15th Int. Conf. Parallel Problem Solving Nature, Part I*, A. Auger, C. M. Fonseca, N. Lourenço, P. Machado, L. Paquete, and D. Whitley, Eds. Cham, Switzerland: Springer, 2018, pp. 274–285.

[5] G. Birkhoff, *Lattice Theory*. Providence, RI, USA: American Mathematical Society, 1948.

[6] J. T. Buchanan, "A naïve approach for solving MCDM problems: The GUESS method," *J. Oper. Res. Soc.*, vol. 48, pp. 202–206, 1997.

[7] L. Chen, B. Xin, J. Chen, and J. Li, "A virtual-decision-maker library considering personalities and dynamically changing preference structures for interactive multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2017, pp. 636–641.

[8] Y. Cui, Z. Geng, Q. Zhu, and Y. Han, "Review: Multi-objective optimization methods and application in energy saving," *Energy*, vol. 125, no. C, pp. 681–704, 2017.

[9] K. Deb, K. Miettinen, and S. Chaudhuri, "Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 821–841, 2010.

[10] K. Deb, J. Sundar, U. B. Rao N, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," *Int. J. Comput. Intell. Res.*, vol. 2, no. 3, pp. 273–286, 2006.

[11] S. C. Endres, C. Sandrock, and W. W. Focke, "A simplicial homology algorithm for Lipschitz optimisation," *J. Global Optim.*, vol. 72, no. 2, pp. 181–217, 2018.

[12] P. Eskelinen and K. Miettinen, "Trade-off analysis approach for interactive nonlinear multiobjective optimization," *OR Spectr.*, vol. 34, no. 4, pp. 803–816, 2012.

[13] S. Greco, M. Ehrgott, J. R. Figueira, Eds., *Multiple Criteria Decision Analysis: State Surveys*. New York, NY, USA: Springer, 2016.

[14] A. Guerraggio and E. Molho, "The origins of quasi-concavity: A development between mathematics and economics," *Historia Math.*, vol. 31, no. 1, pp. 62–75, 2004.

[15] S. Huber, M. J. Geiger, and M. Sevaux, "Simulation of preference information in an interactive reference point-based method for the bi-objective inventory routing problem," *J. Multi-Criteria Decis. Anal.*, vol. 22, nos. 1–2, pp. 17–35, 2015.

[16] C.-L. Hwang and A. Masud, *Multiple Objective Decision Making—Methods and Applications: A State-of-the-Art Survey*. Berlin, Germany: Springer, 1979.

[17] A. Jaszkiewicz and R. Slowinski, "The 'Light Beam Search' approach—An overview of methodology applications," *Eur. J. Oper. Res.*, vol. 113, no. 2, pp. 300–314, 1999.

[18] I. Kaliszewski, "Out of the mist—-Towards decision-maker-friendly multiple criteria decision making support," *Eur. J. Oper. Res.*, vol. 158, no. 2, pp. 293–307, 2004.

[19] I. Kaliszewski, "On variant selection mechanisms in interactive MCDA–Engineering versus reverse engineering," *J. Multi-Criteria Decis. Anal.*, vol. 23, nos. 1–2, pp. 40–48, 2016.

[20] M. Lopez-Ibanez and J. Knowles, "Machine decision makers as a laboratory for interactive EMO," in *Proc. 8th Int. Conf. Evol. Multi-Criterion Optim.*, A. Gaspar-Cunha, C. H. Antunes, and C. Coello, Eds. Cham, Switzerland: Springer, 2015, pp. 295–309.

[21] M. Luque, R. Caballero, J. Molina, and F. Ruiz, "Equivalent information for multiobjective interactive procedures," *Manage. Sci.*, vol. 53, no. 1, pp. 125–134, 2007.

[22] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA, USA: Kluwer, 1999.

[23] K. Miettinen, J. Hakanen, and D. Podkopaev, "Interactive nonlinear multiobjective optimization methods," in *Multiple Criteria Decision Analysis: State Art Surveys*, S. Greco, M. Ehrgott, and J. R. Figueira, Eds. New York, NY, USA: Springer, 2016, pp. 927–976.

[24] K. Miettinen and M. M. Mäkelä, "Synchronous approach in interactive multiobjective optimization," *Eur. J. Oper. Res.*, vol. 170, no. 3, pp. 909–922, 2006.

[25] K. Miettinen, F. Ruiz, and A. P. Wierzbicki, "Introduction to multiobjective optimization: Interactive approaches," in *Multiobjective Optimization: Interactive and Evolutionary Approaches*, J. Branke, K. Deb, K. Miettinen, R. Słowiński, Eds. Berlin, Germany: Springer, 2008, pp. 27–57.

[26] V. Ojalehto, K. Miettinen, and T. Laukkanen, "Implementation aspects of interactive multiobjective optimization for modeling environments: The case of GAMS-NIMBUS," *Comput. Optim. Appl.*, vol. 58, no. 3, pp. 757–779, 2014.

[27] V. Ojalehto, D. Podkopaev, and K. Miettinen, "Towards automatic testing of reference point based interactive methods," in *Proc. 14th Int. Conf. Parallel Problem Solving Nature*, J. Handl, E. Hart, R. Lewis, M. López-Ibáñez, G. Ochoa, and B. Paechter, Eds. Cham, Switzerland: Springer, 2016, pp. 483–492.

[28] D. L. Olson, "Review of empirical studies in multiobjective mathematical programming: Subject reflection of nonlinear utility and learning," *Decis. Sci.*, vol. 23, no. 1, pp. 1–20, 1992.

[29] A. K. Sen, "Choice functions and revealed preference," *Rev. Econ. Stud.*, vol. 38, no. 3, pp. 307–317, 1971.

[30] H. A. Simon, "Rational choice and the structure of the environment," *Psychol. Rev.*, vol. 63, no. 2, pp. 129–138, 1956.

[31] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*. Hoboken, NJ, USA: Wiley, 1986.

[32] T. Stewart, O. Bandte, H. Braun, N. Chakraborti, M. Ehrgott, M. Göbelt, Y. Jin, H. Nakayama, S. Poles, and D. D. Stefano, "Real-world applications of multiobjective optimization," in *Multiobjective Optimization: Interactive and Evolutionary Approaches*, J. Branke, K. Deb, K. Miettinen, and R. Słowiński, Eds. Berlin, Germany: Springer, 2008, pp. 285–327.

[33] T. J. Stewart, "Robustness of additive value function methods in MCDM," *J. Multi-Criteria Decis. Anal.*, vol. 5, no. 4, pp. 301–309, 1996.

[34] T. J. Stewart, "Evaluation and refinement of aspiration-based methods in MCDM," *Eur. J. Oper. Res.*, vol. 113, no. 3, pp. 643–652, 1999.

[35] T. J. Stewart, "Goal programming and cognitive biases in decision-making," *J. Oper. Res. Soc.*, vol. 56, no. 10, pp. 1166–1175, 2005.

[36] M. Sun, "Some issues in measuring and reporting solution quality of interactive multiple objective programming procedures," *Eur. J. Oper. Res.*, vol. 162, no. 2, pp. 468–483, 2005.

[37] M. Vallerio, J. Hufkens, J. Van Impe, and F. Logist, "An interactive decision-support system for multi-objective optimization of nonlinear dynamic processes with uncertainty," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7710–7731, 2015.

[38] H. Varian, *Microeconomic Analysis*, 3rd ed. Mountain View, CA, USA: Norton, 1992.

[39] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization," in *Multiple Criteria Decision Making Theory and Application*, G. Fandel and G. Gal, Eds. Berlin, Germany: Springer, 1980, pp. 468–486.

[40] A. P. Wierzbicki, "A mathematical basis for satisficing decision making," *Math. Model.*, vol. 3, no. 5, pp. 391–405, 1982.

[41] A. P. Wierzbicki, "Reference point approaches," in *Multicriteria Decision Making*, T. Gal, T. Stewart, and T. Hanne, Eds. Boston, MA, USA: Kluwer, 1999, pp. 237–275.

[42] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

[43] B. Xin, L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu, "Interactive multiobjective optimization: A review of the state-of-the-art," *IEEE Access*, vol. 6, pp. 41256–41279, 2018.

[44] A. Zujevs and J. Eiduks, "New decision maker model for multiobjective optimization interactive methods," in *Proc. 17th Int. Conf. Inf. Softw. Technol.* Kaunas, Lithuania: Technologija, 2011, pp. 51–58.

DMITRY PODKOPAEV received the Ph.D. degree in mathematics and physics from the Institute of Mathematics, National Academy of Sciences of Belarus. He currently works as an Assistant Professor with the System Research Institute, Polish Academy of Sciences. His research interest includes multiobjective optimization as a methodology for decision making in complex systems. He develops optimization methods, preference models, and interactive decision support systems, and has applied multiobjective optimization methods in various application domains, such as medicine, finance, computer systems, and supply chain management. He has the title of Docent in multiobjective optimization from the University of Jyväskylä, Finland.

KAISA MIETTINEN received the Ph.D. degree in mathematical information technology from the University of Jyväskylä (JYU), Finland.

She has worked with IIASA, Austria; the KTH Royal Institute of Technology, Stockholm, Sweden; and the Helsinki School of Economics, Finland. She is currently a Professor of industrial optimization with JYU. She also heads the Research Group on Multiobjective Optimization and is the Director of the thematic research area "Decision Analytics Utilizing Causal Models and Multiobjective Optimization" at JYU. She has authored about 190 refereed journals, proceedings, and collection papers; edited 17 proceedings, collections, and special issues; and written a monograph on nonlinear multiobjective optimization. Her research interests include theory, methods, applications, and software of nonlinear multiobjective optimization. She is a member of the Finnish Academy of Science and Letters, Section of Science, and has been the President of the International Society on Multiple Criteria Decision Making (MCDM). She has received the Georg Cantor Award of the International Society on MCDM for an independent inquiry in developing innovative ideas in the theory and methodology of MCDM. She belongs to the Editorial Board of seven international journals and the steering committee of *Evolutionary Multi-Criterion Optimization*.

VESA OJALEHTO received the M.Sc. and Ph.D. degrees in mathematical information technology from the University of Jyväskylä, Finland, in 2006 and 2015, respectively. From 2015 to 2019, he was a Postdoctoral Researcher with the University of Jyväskylä. His research interests include multiobjective optimization and machine learning techniques for anomaly detection.

• • •