IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 4, 2018, accepted July 6, 2018, date of publication July 17, 2018, date of current version August 15, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2856832

Interactive Multiobjective Optimization: A Review of the State-of-the-Art

BIN XIN[1,2,3], (Member, IEEE), LU CHEN[1,2], JIE CHEN[1,2,3], (Senior Member, IEEE), HISAO ISHIBUCHI[4], (Fellow, IEEE), KAORU HIROTA[1], (Life Member, IEEE), AND BO LIU[5]

[1]School of Automation, Beijing Institute of Technology, Beijing 100081, China
[2]State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 100081, China
[3]Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing Institute of Technology, Beijing 100081, China
[4]Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China
[5]Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

Corresponding authors: Bin Xin (brucebin@bit.edu.cn) and Lu Chen (3120140357@bit.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673058 and Grant 71101139, in part by NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization under Grant U1609214, in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China under Grant 61621063, and in part by the Projects of Major International (Regional) Joint Research Program NSFC under Grant 61720106011.

ABSTRACT Interactive multiobjective optimization (IMO) aims at finding the most preferred solution of a decision maker with the guidance of his/her preferences which are provided progressively. During the process, the decision maker can adjust his/her preferences and explore only interested regions of the search space. In recent decades, IMO has gradually become a common interest of two distinct communities, namely, the multiple criteria decision making (MCDM) and the evolutionary multiobjective optimization (EMO). The IMO methods developed by the MCDM community usually use the mathematical programming methodology to search for a single preferred Pareto optimal solution, while those which are rooted in EMO often employ evolutionary algorithms to generate a representative set of solutions in the decision maker's preferred region. This paper aims to give a review of IMO research from both MCDM and EMO perspectives. Taking into account four classification criteria including the interaction pattern, preference information, preference model, and search engine (i.e., optimization algorithm), a taxonomy is established to identify important IMO factors and differentiate various IMO methods. According to the taxonomy, state-of-the-art IMO methods are categorized and reviewed and the design ideas behind them are summarized. A collection of important issues, e.g., the burdens, cognitive biases and preference inconsistency of decision makers, and the performance measures and metrics for evaluating IMO methods, are highlighted and discussed. Several promising directions worthy of future research are also presented.

INDEX TERMS Evolutionary multiobjective optimization, interactive multiobjective optimization, multiple criteria decision making, preference information, preference models.

I. INTRODUCTION

Multiobjective Optimization Problems (MOPs) are ubiquitous in scientific, engineering research, and our social life [1], [2]. An MOP often has several conflicting objectives which are to be optimized simultaneously. Generally, an MOP 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 $f_i(\mathbf{x})$ for $i = 1, 2, \ldots, k$ are objectives to be minimized. The decision vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ belongs to the feasible region S. The objective vector $\mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T$ belongs to the objective space $\mathbf{R}^k$.

Due to the conflicts among objectives, an MOP usually does not have a single optimal solution for all objectives but a set of Pareto optimal solutions. The relevant definitions about the Pareto optimality are as follows [3].

Definition 1 (Pareto Dominance): A decision vector $\mathbf{x}^1$ is said to Pareto dominate $\mathbf{x}^2$ if and only if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$, $\forall i \in \{1, 2, \ldots, k\}$, and there exists at least one index j such that $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$. If $\mathbf{x}^1$ and $\mathbf{x}^2$ do not Pareto dominate each other, they are said to be non-dominated with each other.

Definition 2 (Pareto Optimality): A decision vector $\mathbf{x}^* \in S$ is a Pareto optimal solution if there does not exist any other decision vector $\mathbf{x} \in S$ that Pareto dominates $\mathbf{x}^*$. In this case, $\mathbf{f}(\mathbf{x}^*)$ is called a Pareto optimal objective vector. The set of all Pareto optimal solutions is called the Pareto optimal set. The set of all Pareto optimal objective vectors is named the Pareto front, denoted by PF.

Definition 3 (Weak Pareto Optimality): A decision vector $\mathbf{x}^* \in S$ is weakly Pareto optimal if there does not exist any other decision vector $\mathbf{x} \in S$ that satisfies $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)$, $\forall i \in \{1, 2, \ldots, k\}$.

The ultimate goal of multiobjective optimization is to support a decision maker (DM) to find his/her most preferred solution (MPS). The MPS refers to a Pareto optimal solution which is preferred by the DM to all the other Pareto optimal solutions. Depending on the stage in which the DM takes part in the solution process, multiobjective optimization methods can be divided into the following three classes [3], [4].

- *A priori methods*. The DM provides his/her global preference information first. Then, a Pareto optimal solution which satisfies the preference information is found. This kind of method has been widely used because of the low computational complexity. In reality, however, the DM's global preference information is usually unknown, especially when he/she knows little about the MOP.
- *A posteriori methods*. First, an approximation of the Pareto front is found. Then, the DM chooses the most preferred one. The advantage of a posteriori methods is that the DM can have an overview of the Pareto front. Approximating the whole Pareto front, however, is computationally expensive. Moreover, as the number of objectives increases, the number of non-dominated solutions needed for representing the Pareto front grows exponentially [5], [6], which increases the DM's burdens in selecting the most preferred solution.
- *Interactive methods*. In an interactive multiobjective optimization (IMO) method, the DM specifies preferences progressively during the solution process to guide the search towards his/her preferred regions. The DM does not need to have any global preference structure and he/she can learn from the solution process and adjust his/her preferences. In addition, only one or a small set of solutions which the DM is interested in is found. Thus the computational complexity is reduced and the DM does not need to compare many non-dominated solutions simultaneously. To conclude, interactive methods overcome the weaknesses of both a priori and a posteriori methods [7].

Multiple criteria decision making (MCDM) considers decision problems with multiple conflicting criteria [7]. Its foundations were developed in the 1950s and 1960s and many seminal contributions emerged in the 1970s [8]. MCDM includes multiple attribute decision making which solves discrete problems with a finite set of alternatives and multiple objective decision making which is suitable for continuous problems (i.e., MOPs) [9].

From the 1970s to the 1990s, the MCDM community had developed many classical IMO methods like the step method (STEM) [10], the Geoffrion-Dyer-Feinberg method (GDF) [11], the interactive surrogate worth tradeoff method (ISWT) [12], the reference point method [13], the satisficing tradeoff method (STOM) [14], the GUESS method [15], the Zionts-Wallenius (Z-W) method, and the NIMBUS method [16], [17]. Readers are referred to [17] and [31] for more details of these methods. In general, these methods transform an MOP into a Single-objective Optimization Problem (SOP) at every iteration by using the DM's local preference information. The "iteration" refers to a loop of interaction between the DM and the algorithm. The DM provides preferences for the algorithm, and the algorithm shows the DM one or a set of solutions obtained based on the DM's preferences. The overall process continues until the DM finds his/her MPS. The SOPs are mainly solved by mathematical programming (MP) techniques. MP techniques, however, often require the convexity or the differentiability of SOPs, which may limit the applicability of IMO methods in solving complex MOPs [18].

In the late 1950s, evolutionary computation (EC) sprouted. It has received significant attention since the 1980s [19]. Among a number of developed evolutionary algorithms (EAs), three typical paradigms are genetic algorithm (GA), evolutionary programming (EP), and evolution strategy (ES) [20]. The EAs are population-based optimization algorithms which simulate the biological evolution process by performing a loop of several basic operators such as selection, crossover, and mutation. They have been widely used to solve SOPs since they do not require the continuity, differentiability, or convexity of SOPs. The population-based parallel search of EAs is beneficial to finding multiple Pareto optimal solutions of an MOP in a single run [18].

In the 1980s, Schaffer [21] designed a multi-objective evolutionary algorithm (MOEA) which is called vector evaluated genetic algorithm (VEGA). VEGA is deemed as the first evolutionary multiobjective optimization (EMO) method which uses an EA to solve MOPs [18]. From then on, the field of EMO has developed rapidly. In the early days, the EMO community focused on approximating the whole Pareto front of an MOP. Many EMO algorithms adopt the Pareto dominance to select solutions. However, as the number of objectives grows, the proportion of non-dominated solutions in the population increases. For MOPs with more than three objectives (known as many-objective optimization problems, MaOPs), it is difficult to differentiate solutions by only using the Pareto dominance relation [22], [23].

Realizing the need for collaborations, in 2004, Branke, Deb, Miettinen, and Steuer organized the first Dagstuhl seminar to provide a platform for researchers from MCDM and EMO communities to exchange ideas in solving MOPs [7]. A consensus is reached that it is promising to combine ideas and approaches from the two communities. Since then, the Dagstuhl seminar has been organized every two or three years to enhance the collaboration of the two communities.

In [24], two ways of hybridizing MCDM and EMO methods are identified: "EA in MCDM" and "MCDM in EMO." "EA in MCDM" means solving the SOPs formulated in MCDM methods by EAs. In this way, EAs can help MCDM solve difficult problems (e.g., nonconvex, discontinuous, or non-differentiable problems) that MP techniques are difficult or not able to handle. "EA in MCDM" based approaches can be found in [24]–[26]. Borrowing ideas from MCDM approaches, "MCDM in EMO" based algorithms incorporate the DM's preferences into MOEAs a priori or interactively to guide MOEAs to find solutions which approximate the DM's preferred region of the Pareto front. Approximating only a part of the Pareto front can greatly reduce computational complexity of MOEAs, especially when solving MaOPs. What's more, the DM's preferences can be used to distinguish non-dominated solutions.

Recent reviews on preference-based MOEAs can be found in [27]–[29]. As a part of preference-based MOEAs, interactive MOEAs are included in those reviews. Meignan *et al.* [30] provided a review of interactive optimization methods in the field of operational research. Since the authors focus on a larger field than multiobjective optimization, IMO methods are only one class of the methods that they reviewed. Therefore, state-of-the-art IMO methods, especially interactive MOEAs, are not reviewed sufficiently. Miettinen *et al.* [31] provided an updated overview of IMO methods for solving nonlinear MOPs. The IMO methods developed in the field of EMO are not considered. Focusing on interactive MCDM methods for continuous problems, Kasimoğlu [32] overviewed several classical IMO methods like STEM, GDF, and STOM, while some interactive MOEAs are only briefly mentioned.

To sum up, existing reviews on IMO may concentrate on a larger field than IMO or center mainly on part of IMO methods, i.e., interactive MCDM methods and interactive MOEAs. This paper aims to give an interdisciplinary review of IMO from an "MCDM + EMO" perspective according to a systematic taxonomy which can distinguish IMO methods developed in both fields. Noticing the existence of a large number of IMO methods, especially in the field of MCDM, it is impossible for us to cover all these methods due to space limitations. Thus, only representative IMO methods are reviewed in this paper. The main contributions of this paper are as follows:

- Most of the existing reviews on IMO classify IMO methods based on the type of preference information provided by the DM. A taxonomy of interactive methods in operational research with two views which focus on user's contribution and the components of interactive optimization systems was proposed in [30]. However, the taxonomy is not specific to IMO methods. To better differentiate and characterize different IMO methods, we originally establish a systematic taxonomy for IMO methods which incorporates four essential factors of IMO. The taxonomy facilitates a clear understanding and comparison of the main ideas of various IMO methods and is expected to aid the design of new IMO methods.
- To our best knowledge, for the first time, we made an interdisciplinary review of the state-of-the-art IMO methods from both MCDM and EMO fields by following the established taxonomy with the aim of promoting comprehensive understanding of IMO methods. The main characters of each method which make it different from others are identified.
- We identified and discussed eight crucial issues in IMO from the perspectives of the DM, the algorithm, and the interaction process. They facilitate the understanding of IMO and provide directions for future research.
- Based on the current development of IMO, several research directions worthy of future investigation are provided.

The remainder of this paper is organized as follows. Section II presents the comprehensive taxonomy for IMO methods which consists of four essential design factors including interaction pattern, preference information, preference model, and search engine. Section III is devoted to the review of representative state-of-the-art IMO methods according to the taxonomy. In Section IV, eight crucial issues in IMO are discussed. Conclusions are drawn in Section V and several potential research directions are also provided.

II. TAXONOMY FOR IMO METHODS

Interactive multiobjective optimization comprises two important components, i.e., the DM and the machine (algorithm). Fig. 1 depicts a DM-Machine interaction system which shows the interaction between the DM and the machine in IMO methods. The DM refers to a human DM who wants to find his/her MPS, and the machine is actually an algorithm which involves a preference model and a search engine (optimization algorithm).

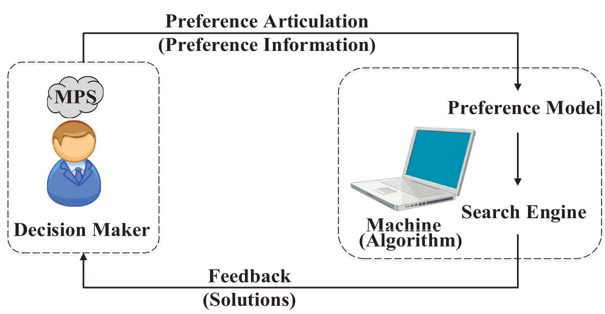


FIGURE 1. The DM-Machine interaction system.

The overall interaction process is as follows. The DM articulates his/her preference information based on his/her knowledge about the problem and the solutions provided by the machine. Based on the provided preferences, the machine builds a preference model which is a bridge between the preference information and the search engine as well as between the DM and the machine. It plays the role of integrating the DM's preferences into the machine and guiding the search engine to find solutions that the DM is interested in.

The obtained solutions are shown to the DM to help him/her provide new preferences. By interacting with the machine, the DM can learn about the problem and adjust his/her preferences to finally find his/her MPS.

According to the DM-Machine interaction system shown in Fig. 1, three essential design factors of IMO methods can be recognized, that is, the preference information, the preference model, and the search engine. Different types of preference information can be provided by the DM and they may cause different cognitive burdens to the DM. As a pivotal element of IMO methods, the preference model determines how the DM's preferences can be utilized by the machine. The DM does not need to know it when interacting with the machine. The search engine, viz., the algorithm, determines the quality of the obtained solutions.

When designing an IMO method, apart from the above three design factors, one should also consider when to ask the DM to interact with the algorithm, after the complete run of the search engine (optimization algorithm) or during its run. Here the complete run refers to running the search engine completely until its stopping criterion is satisfied. We call this design factor the *interaction pattern*. Naturally, two interaction patterns can be identified: *interaction after a complete run (IAR)* and *interaction during the run (IDR)* of the search engine. IAR means that the search engine is allowed to run to end and return the expected solutions according to the most recent preference information provided by the DM. So, no DM-Machine interaction occurs until the termination criterion of the search engine has been satisfied. The DM has to wait for the termination of the search engine before each interaction. The algorithm will be restarted after the DM provides new preferences. In contrast, IDR means that the DM can pause the running of the algorithm as he/she wishes and the algorithm will proceed after the DM gives new preferences. Obviously, the DM-machine interaction is triggered by the machine (algorithm) in IAR, and is more governed by the DM in IDR. Under different interaction patterns, the quality of obtained solutions at each iteration differs and the suitable types of preferences that the DM can provide also differ (details will be presented in Section II-A).

From the above, a systematic taxonomy consisting of the following four essential design factors of IMO methods is built in this section: 1) the *interaction pattern*, 2) the *preference information*, 3) the *preference model*, and 4) the *search engine*. Details of the taxonomy are given as follows.

A. INTERACTION PATTERN

Both IAR and IDR patterns have been widely adopted by IMO methods. Since an IMO method adopting the IAR pattern performs a complete run of the search engine between two adjacent interactions, it can obtain one or a set of (approximate) Pareto optimal solutions at each iteration, as shown in Fig. 2 (a) and (b). If only one solution is generated and shown to the DM, the DM can provide preferences like reference points, weights, tradeoffs, and the classification of objectives. If multiple solutions are shown to the DM,

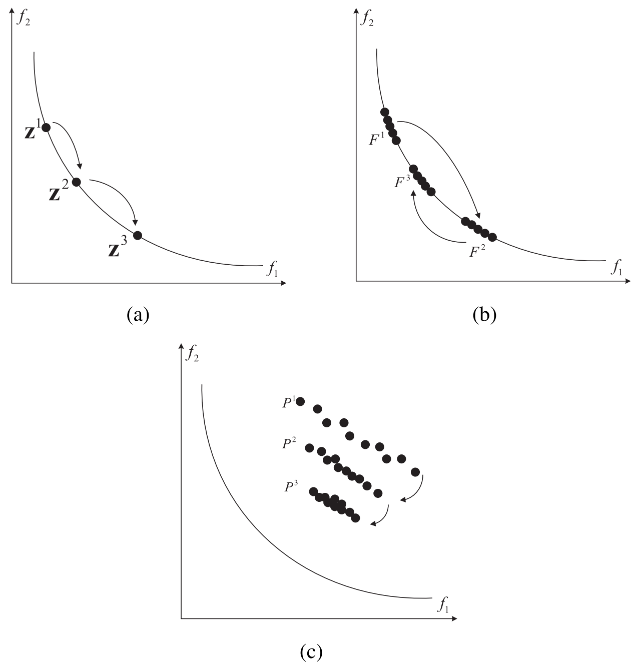


FIGURE 2. The solutions obtained in the first three iterations of IMO methods are demonstrated. The curve in each subfigure represents the Parero front. (a) and (b) show a single solution (**z**) and a set of solutions (*F*) obtained by methods adopting the IAR pattern, respectively. (c) shows the evolution of a population (*P*) corresponding to a sequence of the DM's preferences for methods adopting the IDR pattern.

he/she can specify reference points or weights, provide tradeoffs or classification of objectives on the most preferred solution, compare these solutions, and so on. The IAR pattern is adopted by classical interactive MCDM approaches to obtain one or multiple Pareto optimal solutions at each iteration. It is also used in some interactive MOEAs such as MOEAs based on the reference point information. Details of these IMO methods can be found in Section III.

IMO methods adopting the IDR pattern ask the DM to provide preferences periodically during the run of the search engine to guide the search towards the DM's region of interest (ROI), as shown in Fig. 2 (c). Generally, the population does not get close to the Pareto front until the later period of the optimization. As indicated by Deb *et al.* [33], having more chances to provide new information in this case, the DM is more involved in the overall optimization-cum-decision-making process. So, the IMO process under the IDR pattern is more DM-oriented. It should be noted that preferences like tradeoffs and classification of objectives are more suited to methods employing the IAR pattern since it is more meaningful to consider the tradeoffs among objectives on Pareto optimal solutions.

B. PREFERENCE INFORMATION

According to whether a DM has to make comparisons among objectives or solutions, we divide the preference information into three categories: the *expectation*, the *comparison of objective functions* and the *comparison of solutions*.

The *expectation* refers to goals that the DM wants to achieve. It is often embodied by a reference point which is composed of aspiration levels that the DM wants to achieve for objectives. Specifying reference points has been found a cognitively valid approach of articulating preferences [34]. The aspiration levels for different objectives are independent and can be specified freely by the DM. As a consequence, the reference point can be either achievable or unachievable. To make the DM possess a holistic perception of objectives, the ranges of objectives often need to be known, which demands a prior knowledge or additional computational cost.

The *comparison of objective functions* and the *comparison of solutions* both require the DM to make comparisons when expressing preferences. The difference is that the former reflects the relations among the objectives while the latter actually refers to the comparison of a set of objective vectors which may result in a ranking or classification of them.

The *comparison of objective functions* can be achieved by means of weights, tradeoffs, the classification of objectives, etc. Weights are often used to reflect the relative importance of objectives. They can be specified directly in the form of k weight values, which may not be easy for the DM. Some methods ask the DM to provide the importance grades of objectives through comparing them in pairs so as to induce weights (see, e.g., [37]). In this way, the DM's burdens can be reduced. However, when the number of objectives is large, the DM needs to make a lot of pairwise comparisons of objectives. Note that Roy and Mousseau [35] showed that it is not clear what underlies the notion of importance of objectives. Miettinen [36] stated that controlling the solution process with weights is not necessarily easy for the DM since weights behave in an indirect way. Instead of asking the DM to provide the relative importance of objectives, Luque *et al.* [38] developed an approach to allow the DM to give the relative importance of achieving each aspiration level in reference point based IMO methods.

A tradeoff means sacrificing one objective to gain improvement of another one at a feasible solution. A common form of a tradeoff is the indifference tradeoff (or marginal rate of substitution, MRS) which refers to the amount of increment of one objective to compensate one unit decrement of another one (see [3], [17] for more details). Tradeoffs can provide a precise search direction for the algorithm to find a more desirable solution from the current Pareto optimal solution. However, tradeoff specification often demands heavy cognitive effort from the DM since the DM needs to decide the amount of tradeoffs among objectives. In some IMO methods like the Z-W method and the ISWT method, tradeoff rates are provided for the DM by the methods and what the DM needs to do is specifying his/her desirability on these tradeoffs [12], [17].

The classification of objectives refers to dividing the objectives into several classes at the current Pareto optimal solution according to the types of desirable changes for the objective values of this solution. Miettinen *et al.* summarized five classes of objectives in [17]:

1) $I^{<}$: the set of objectives which should be improved from the current values,
2) $I^{\leq}$: the set of objectives which should be improved till some aspiration levels,
3) $I^{=}$: the set of objectives which are acceptable at the moment,
4) $I^{\geq}$: the set of objectives which can be sacrificed till some upper bounds,
5) $I^{\diamond}$: the set of objectives which can change freely.

Different subsets of the five classes can be adopted. For instance, the STEM method asks the DM to specify satisfactory objective values ($I^{\geq}$) and unsatisfactory ones ($I^{<}$). In the STOM method, the DM needs to classify objectives into three classes, i.e., the classes of objectives which the DM wants to improve more ($I^{\leq}$), relax ($I^{\geq}$), and accept as they are ($I^{=}$), respectively [10].

Classifying the objectives is also found a cognitively valid way of expressing preferences [34]. Compared with providing reference points, the DM can be more in control of the solution process by classifying objectives and specifying the amount of relaxation for the objectives that can be impaired. However, these additional specifications demand more effort from the DM.

The *comparison of solutions* is usually realized by making pairwise comparison on solutions, classifying solutions, or selecting the most preferred one. The pairwise comparison judges the relation between a pair of solutions: one solution is preferred to the other, or the two solutions are incomparable or indifferent. The classification of solutions refers to dividing solutions into multiple classes where solutions in each class are incomparable or indifferent. For example, the DM is asked to classify a sample set of solutions into "relatively good" and "others" in [39]. Selecting the most preferred solution means choosing the best one from a set of solutions. As qualitative preference information, the comparison of solution requires relatively less cognitive burdens from the DM compared with specifying quantitative preferences like aspiration levels and tradeoffs [40]. However, it is important to note that as the number of solutions increases, the DM's burdens are likely to increase, too.

For convenience of an intuitive demonstration of the characters of the above mentioned types of preference information, Table 1 summarizes their features and pros and cons.

C. PREFERENCE MODEL

The following three types of preference models have been frequently used in the literature: *value function (utility function), dominance relation*, and *decision rules* [41]. A value function (VF) is a scalar function of all objectives which evaluates solutions quantitatively. Its parameters are specified by the DM directly or calculated indirectly based on the DM's preferences. Dominance relation describes the DM's preferences in the form of the relation of a pair of solutions. It is often used in place of the Pareto dominance relation in the selection operator of interactive MOEAs. Decision rules model the DM's preferences as a set of "IF-THEN" rules.

TABLE 1. Common types of preference information and their pros and cons.

Category	Preference Information	Features	Pros and Cons
Expectation	Reference point	A reference point consists of k continuous valued aspiration levels representing desirable objective values (quantitative).	The reference point can be specified freely by the DM and it can be either achievable or unachievable. The ranges of objectives often need to be calculated, which demands a prior knowledge or additional computational cost.
Comparison of objective functions	Weights	Weights are used to reflect the relative importance of objectives or achieving certain values. They can be specified by the DM directly in the form of k scalar values (quantitative) or induced from the DM's pairwise comparisons of objectives (qualitative).	It is often believed that weights would represent the relative importance of objectives. However, it is stated in [35] that it is not clear what underlies this notion. In addition, controlling the solution process with weights is not necessarily easy for the DM [36].
	Tradeoffs	In general, taking one objective as the reference objective, the DM needs to give the amount of increment of each of the remaining $k-1$ objectives to compensate one unit decrement of the reference objective (quantitative).	Tradeoffs can provide a precise search direction for finding a more desirable solution. However, providing tradeoffs often lays relatively heavy cognitive burdens on the DM since the DM needs to decide the amount of tradeoffs among objectives.
	Classification of objectives	The DM is asked to classify the objectives into several classes according to the types of desirable changes for objectives (qualitative). Meanwhile, he/she needs to specify aspiration levels or upper bounds for some objectives (quantitative).	Classifying objectives is closely related to specifying reference points. However, the DM can be more in control of the solution process by classifying objectives and specifying the amount of relaxation for the objectives which can be sacrificed. Note that these additional specifications demand more effort from the DM.
Comparison of solutions	Pairwise comparison of solutions	Judging whether one solution is preferred to the other or they are incomparable or indifferent (qualitative).	Comparing solutions requires relatively less cognitive burdens from the DM. However, as the number of solutions increases, the DM's burdens are likely to increase, too.
	Classification of solutions	Dividing a set of solutions into multiple classes where solutions in each class are incomparable or indifferent (qualitative).	
	Selecting the most preferred solution	Selecting the most preferred solution from a set of solutions (qualitative).	

Generally, the premise part of decision rules specifies the conditions that the objectives and/or solutions should satisfy. The decision part specifies relations among solutions or assigns them scores.

1) VALUE FUNCTION

Many IMO methods assume that the DM provides preferences by implicitly referring to an underlying VF which is a scalar function of all objective functions with the form $U = U(f_1, f_2, \ldots, f_k)$. An explicit VF can provide a complete ranking of objective vectors in the objective space. Its optimal solution is the DM's MPS. However, the DM's VF is usually not known explicitly because the DM does not have complete information about an MOP. Many methods model the DM's underlying VF dynamically based on the DM's preferences. In the following, three types of popular VFs are explained.

a: WEIGHTED METRICS

A weighted metric measures the distance between the objective vector and a certain point. The point can be an ideal point $\mathbf{z}^* = [z_1^*, z_2^*, \ldots, z_k^*]^T$ with $z_i^* = \min_{\mathbf{x}\in S} f_i(\mathbf{x})$, $i = 1, 2, \ldots, k$, a nadir point $\mathbf{z}^{nad} = [z_1^{nad}, z_2^{nad}, \ldots, z_k^{nad}]^T$ with $z_i^{nad} = \max_{\mathbf{z}\in PF} z_i$, $i = 1, 2, \ldots, k$, etc. The DM's preferences are reflected by weights. Different L_p-norms with p varying from 1 to ∞ can be adopted.

In interactive MCDM approaches, weighted metrics are formulated as distance optimization problems to find (weakly) Pareto optimal solutions. The *weighted L_p-problem* with $1 \leq p < \infty$ for minimizing the distance to the ideal point is of the following form [3]

$$\begin{aligned} &\text{minimize } \Big\{\sum_{i=1}^{k} w_i \big|f_i(\mathbf{x}) - z_i^*\big|^p\Big\}^{1/p} \\ &\text{subject to } \mathbf{x} \in S \end{aligned} \tag{2}$$

where $w_i \geq 0$, $i = 1, \ldots, k$, and $\sum_{i=1}^{k} w_i = 1$. The *weighted Tchebycheff problem* with $p = \infty$ is of the form

$$\begin{aligned} &\text{minimize } \max_{i=1,\ldots,k} \left\{w_i \left|f_i(\mathbf{x}) - z_i^*\right|\right\} \\ &\text{subject to } \mathbf{x} \in S. \end{aligned} \tag{3}$$

Instead of minimizing the distance to the ideal point, the GUESS method adopts the opposite idea, i.e., maximizing the minimum weighted deviation from the nadir point [3], [15]. Some interactive MOEAs use the weighted metrics directly to guide the evolution of the population. Deb *et al.* [42] utilized the following weighted Euclidean distance metric:

$$d = \sqrt{\sum_{i=1}^{k} w_i \left(\frac{f_i(\mathbf{x}) - q_i}{f_i^{\max} - f_i^{\min}}\right)^2} \tag{4}$$

where $\mathbf{q} = [q_1, \ldots, q_k]^T$ is the reference point provided by the DM, $f_i^{\max}$ and $f_i^{\min}$ are the population maximum and minimum values of the ith objective function, respectively. Solutions close to the reference point under this metric are preferred. It should be noted that preferring solutions close to the reference point may make a solution preferred to another solution which dominates it when the reference point is achievable.

b: ACHIEVEMENT SCALARIZING FUNCTION

The achievement scalarizing function (ASF) was first introduced by Wierzbicki in the reference point method [13]. Wierzbicki regarded it as a modified VF which expresses both the utility of achieving the aspiration levels and the disutility of not achieving them [13]. Up to now, many forms of ASFs have been developed [43]–[46]. Two common ASFs are

$$s(\mathbf{f}(\mathbf{x}), \mathbf{q}, \mathbf{w}) = \max_{i=1,\ldots,k} \{w_i(f_i(\mathbf{x}) - q_i)\} \tag{5}$$

and

$$s_{aug}(\mathbf{f}(\mathbf{x}), \mathbf{q}, \mathbf{w}) = \max_{i=1,\ldots,k} \{w_i(f_i(\mathbf{x}) - q_i)\} + \rho \sum_{i=1}^{k} w_i(f_i(\mathbf{x}) - q_i) \tag{6}$$

where ρ is a small positive number [3], [13]. The solutions obtained by minimizing (5) are weakly Pareto optimal and not necessarily Pareto optimal. It can be proved that (6) generates Pareto optimal solutions with tradeoffs between ρ and $1/\rho$. The weights in (5) and (6) can be kept unaltered or be renewed progressively based on the DM's preferences. For example, three ways of incorporating preferences into the weights of an ASF are introduced in [38].

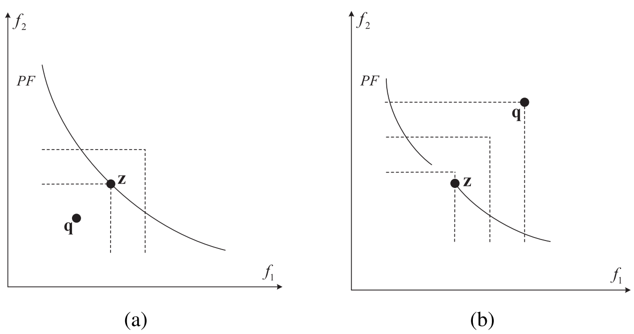


FIGURE 3. The Pareto optimal solution (z) obtained by minimizing (5). Dotted lines are the contour lines of (5). (a) The reference point (q) is unachievable and the obtained Pareto optimal solution is the vertex of a contour line. (b) The reference point is achievable and the Pareto front is disconnected. The obtained Pareto optimal solution lies on the edge of a contour line.

ASFs have been widely used in both MCDM and EMO fields. Interactive MCDM methods usually use ASFs to formulate minimization problems. Fig. 3 shows the Pareto optimal solution obtained by minimizing (5). The obtained solution is usually the vertex of a contour line of (5) as shown in Fig. 3 (a). It may also lie on the edge of a contour line of (5) for certain shapes of the Pareto front like disconnected ones, as shown in Fig. 3 (b). Interactive EMO methods use ASFs to divide a population into multiple fronts (e.g., [47], [48]) or to define a fitness function which is used to evaluate solutions in the selection operator of MOEAs (e.g., [49]).

c: ADDITIVE VF

Some methods use the following *additive VF*

$$U(\mathbf{f}(\mathbf{x})) = \sum_{i=1}^{k} u_i(f_i(\mathbf{x})) \tag{7}$$

where $u_i(\cdot)$ for $i = 1, \ldots, k$ are non-decreasing marginal VFs [50]–[53]. Marginal VFs are assumed to be piecewise linear in for example [50] and generally non-decreasing in [51]–[53].

A single representative compatible additive VF is used to represent the DM in [53]. Here "compatible" means that an IMO method can construct a preference model which ranks solutions in the same way as the DM does. The whole set of compatible additive VFs is considered in [51] and [52].

There are also other kinds of VFs such as polynomial functions [33] and Gaussian functions [54], [55]. For VFs like weighted metrics, ASFs, and Gaussian functions, their parameters such as weights, reference points, and center vectors are specified by the DM directly at each iteration. Additive VFs and polynomial VFs can be learned from the DM's previous and current preferences by solving an optimization problem with the DM's preferences being constraints [33], [53]. Some IMO methods do not make any a priori assumptions on the form of the DM's underlying VF and use techniques like neural networks and support vector machines to learn a VF from his/her preferences [56]–[58].

2) DOMINANCE RELATION

A dominance relation expresses the DM's preference over a pair of solutions: one dominates (*is preferred over*) the other, or they are indifferent or incomparable. Generally, the binary relation of two solutions $\mathbf{x}$ and $\mathbf{y}$ can be denoted by $\mathbf{x}D\mathbf{y}$ ($\mathbf{x}$ dominates $\mathbf{y}$), $\mathbf{y}D\mathbf{x}$ ($\mathbf{y}$ dominates $\mathbf{x}$), or $\mathbf{x}\bar{D}\mathbf{y}$ (indifferent). Many dominance relations combine the Pareto dominance relation with the DM's preferences, thus non-dominated solutions can also be compared. Here are three examples.

Fonseca and Fleming [59] introduced a "*preferable to*" relation based on aspiration levels provided by the DM. For two solutions $\mathbf{x}$ and $\mathbf{y}$, if $\mathbf{x}$ satisfies all aspiration levels, it is *preferable to* $\mathbf{y}$ if and only if $\mathbf{x}$ Pareto dominates $\mathbf{y}$ or $\mathbf{y}$ does not satisfy all aspiration levels. If $\mathbf{x}$ satisfies none of the aspiration levels, it is *preferable to* $\mathbf{y}$ if and only if $\mathbf{x}$ Pareto dominates $\mathbf{y}$. In the case where $\mathbf{x}$ meets only a part of the aspiration levels, the objectives whose aspiration levels are not met are first considered. If $\mathbf{x}$ Pareto dominates $\mathbf{y}$ with respect to these objectives, $\mathbf{x}$ is *preferable to* $\mathbf{y}$. Besides, if the objective values of $\mathbf{x}$ on these objectives are equal to those of **y**, **x** is *preferable to* **y** provided that **x** Pareto dominates **y** with respect to the remaining objectives, or **y** does not meet all the aspiration levels of the remaining objectives.

Sinha *et al.* [60] constructed a polyhedral cone based on the DM's preferences. If two solutions are both outside or inside the cone, they are compared based on the Pareto dominance relation. Otherwise, the solution inside the cone dominates the solution outside the cone.

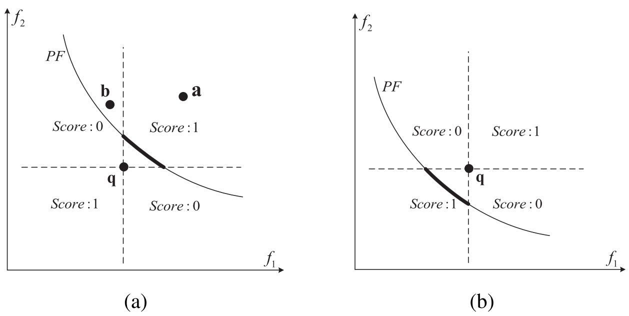


FIGURE 4. Scores based on the reference point q. (a) q is unachievable. (b) q is achievable.

Molina *et al.* [61] defined the *g-dominance* relation by using a reference point. To guide the search towards the local Pareto front around the projection of the reference point (as shown by the bold curves in Fig. 4), solutions satisfying all or none of the aspiration levels in the current population are preferred. Each of them is assigned a score of 1, and the scores of the other solutions are all 0. For two solutions **x** and **y**, **x** *g*-dominates **y** if the score of **x** is larger than that of **y** or their scores are equal but **x** Pareto dominates **y**. Under this definition, solution **a** shown in Fig. 4 (a) is preferred to solution **b** even if **b** Pareto dominates **a**. Therefore, the *g-dominance* relation does not preserve the Pareto dominance relation [62].

3) DECISION RULES

Decision rules usually take the form of "IF-THEN" rules which are induced from the DM's preferences. If certain conditions are satisfied by objectives and/or solutions, decision rules can specify the relations among solutions or assign solutions scores to facilitate the selection of preferred solutions. A preference model in the form of decision rules is more general than the classical functional or relational model and is more understandable for the DM due to its natural syntax [63]. Specifically, as a kind of decision rules, fuzzy rules can handle the DM's qualitative preferences and transform them into quantitative information on objectives or solutions.

In the dominance-based rough set approach (DRSA) [64], decision rules are used to model the DM's preference information which takes the form of exemplary comparison such as a ranking or sorting of a sample set of solutions. The DRSA is used in two interactive MOEAs in [39]. One method uses induced decision rules to provide each solution a score, and the other method utilizes decision rules to define a dominance relation (see Section III for details).

Shen *et al.* [37] constructed a fuzzy inference system to transform the DM's preferences and degrees of improvements between two solutions into strengths of solutions. The DM is asked to express linguistic preferences instead of quantitative preferences, thus his/her cognitive burdens can be reduced.

4) HYBRID PREFERENCE MODEL

Some IMO methods utilize a hybrid preference model. For example, in [62], the weighted Euclidean distance in the form of (4) is used to define the *r-dominance* relation. A solution **x** is said to *r*-dominate a solution **y** if they satisfy one of the following two conditions: 1) **x** Pareto dominates **y**, 2) **x** and **y** are non-dominated, **x** is closer to the reference point than **y** in the sense of the weighted Euclidean distance, and the absolute value of their distance difference is larger than a threshold [62]. In [37], after obtaining the strengths of solutions via the fuzzy inference system, a *strength superior* relation is constructed to define a fitness function.

D. SEARCH ENGINE

The search engine, based on the preference model, searches for solutions that the DM is interested in. Even with the human involvement in IMO, search engine is still a decisive factor for the performance of problem-solving. How to choose or design a competent search engine is always a basic yet crucial issue in IMO, especially for solving hard optimization problems.

We classify the search engines used in IMO methods into MP and non-MP techniques. For different MP branches like linear programming, nonlinear programming, and multiobjective programming, various mature optimization techniques have been developed for solving corresponding problems, e.g., the simplex method for linear programming problems, and the sequential quadratic programming for nonlinear problems. These techniques are often integrated into software platforms in order to aid users to solve optimization problems. IMO methods developed by the MCDM community usually adopt MP techniques to generate Pareto optimal solutions.

Non-MP techniques are mainly heuristic methods including EAs, tabu search, simulated annealing, etc. EAs emphasize versatility and have shown advantages in solving complicated SOPs such as multi-modal, discontinuous, strongly constrained and dynamic problems. EAs feature a population-based search paradigm, which facilitates global search in the solution space as well as parallel implementation. Compared with MP techniques, EAs may be time consuming and they are not guaranteed to find optimal solutions of SOPs. However, they may be able to solve complicated SOPs which MP techniques are hard or impossible to solve. In addition, EAs are able to find a set of (approximate) Pareto optimal solutions of an MOP in a single run, which is very appealing in multiobjective optimization (either a posterior or interactive methods) since the DM can get more information and have more choices. The past decade has witnessed the great success of many EMO algorithms like NSGA-II [65] and MOEA/D [66] in a myriad of

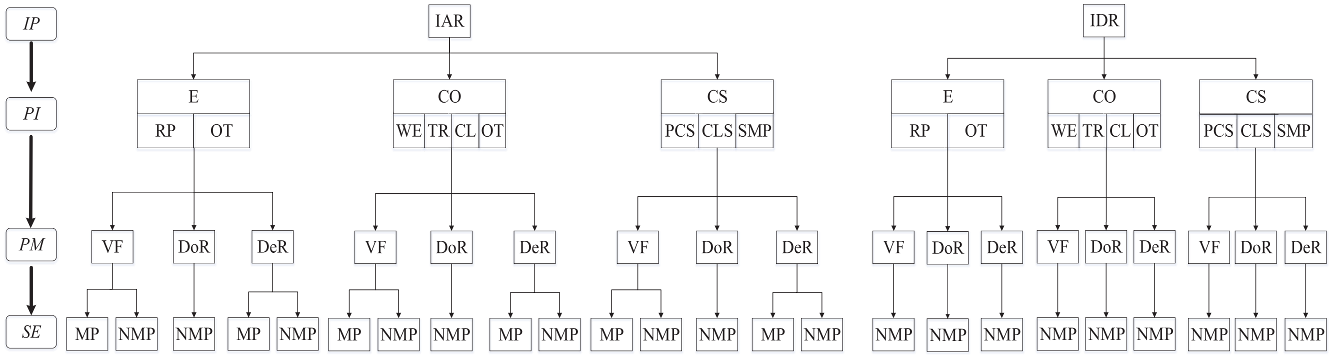


FIGURE 5. Taxonomy for IMO methods. Each string represents a type of IMO methods.

engineering applications [67]. Generally, interactive MOEAs use EAs as search engines. By incorporating the DM's preferences into the selection operator of EAs, their population can move towards the DM's ROI. GAs are the most commonly used EAs in MOEAs.

In recent years, swarm intelligence (SI) has developed rapidly [68], [69]. SI means that a swarm of simple individuals shows intelligent behavior as a whole by interacting with each other and with the environment. Typical SI-based optimization algorithms include ant colony optimization, particle swarm optimization, artificial immune system, bee colony optimization, fish school search, fireworks algorithm, brain storm optimization, etc. Many SI-based multiobjective optimization methods have been proposed [69]–[73].

Memetic algorithms, which combine EAs with local search, have been investigated in many studies [74], [75]. Memetic MOEAs which hybridize MOEAs with local search methods have also been studied recently [76]–[78]. The inclusion of well-directed local search can improve the overall performance of MOEAs [77]. A few IMO methods incorporate the DM's preferences into SI-based optimization algorithms or memetic algorithms [79]–[82]. The advantages of using these algorithms as search engines of IMO methods deserve further investigation.

E. TAXONOMY

According to the above four design factors, we build a taxonomy for IMO methods. The notations of the four design factors and their candidate classes are listed as follows.

1) Interaction Pattern (IP)
   a) Interaction after a complete run (IAR), b) interaction during the run (IDR).
2) Preference Information (PI)
   a) Expectation (E) which includes the reference point (RP) and others (OT), b) the comparison of objective functions (CO) including weights (WE), tradeoffs (TR), the classification of objectives (CL), and others (OT), c) comparison of solutions (CS) comprising pairwise comparisons of solutions (PCS), the classification of solutions (CLS), and selecting the most preferred one (SMP).
3) Preference Model (PM)
   a) Value function (VF), b) dominance relation (DoR), c) decision rules (DeR).
4) Search Engine (SE)
   a) Mathematical programming techniques (MP), b) non-mathematical programming techniques (NMP).

There are special connections among the candidate classes of different design factors. For example, a dominance relation usually acts on a set of solutions, and it is not suitable to employ MP techniques as corresponding SEs. This is because MP techniques handle a single solution instead of a population. In addition, population-based non-MP techniques are good candidates of SEs for the IDR pattern instead of MP techniques. As stated before, when a search engine is interrupted by the DM during its run, the solutions found usually do not reach the Pareto front. The non-MP techniques can obtain multiple non-dominated solutions which provide more information than a single solution obtained by MP techniques. The promising combinations of the candidate classes of the four design factors are given in Fig. 5.

III. REPRESENTATIVE STATE-OF-THE-ART IMO METHODS

This section gives a review of some representative state-of-the-art IMO methods according to the taxonomy built in Section II. The chosen representative IMO methods are summarized in Table 2 with authors, references, names, category features, and main ideas being listed. The category feature of an IMO method is expressed in the form $< IP, PI, PM, SE >$. According to the proposed taxonomy, E, CO, and CS are candidate classes of PI. To render a further description, their candidate subclasses will also be shown at their lower right corner. For example, E_{RP} indicates that the preferences articulated by the DM are reference points which belong to the class *expectation*. Since a few methods allow the DM to change the type of preferences that he/she wants to provide during the solution process, their category features over PI are represented as $PI1_{subclass1/subclass2/\ldots}/PI2_{subclass1/subclass2/\ldots}/\ldots$. For instance, if a method asks the DM to provide preferences in the form of either a reference point or the classification of solutions, its category feature over PI is E_{RP}/CS_{CLS}.

TABLE 2. Representative IMO methods.

Authors and References	Names	Category Features	Main Ideas
Deb et al. [42]	R-NSGA-II	$< IAR, E_{RP}, VF, NMP >$	The reference point method and NSGA-II are combined. Solutions with short weighted Euclidean distance from the reference point are preferred.
Deb and kumar [47]	RD-NSGA-II	$< IAR, E_{RP}, VF, NMP >$	The reference direction method and NSGA-II are combined together. Multiple ASFs are used to classify solutions into different fronts.
Deb and kumar [83]	LBS-NSGA-II	$< IAR, E_{RP}, VF, NMP >$	The light beam search is combined with NSGA-II. An ASF is used to modify the crowding scheme of NSGA-II.
Thiele et al. [49]	PBEA	$< IAR, E_{RP}, VF, NMP >$	The reference point method is combined with an indicator-based evolutionary algorithm [84]. An ASF is used to modify the fitness function of evolutionary algorithm.
Molina et al. [61]	g-dominance	$< IAR, E_{RP}, DoR, NMP >$	The *g-dominance* relation is proposed. Solutions satisfying all aspiration levels or none of the aspiration levels are preferred.
Said et al. [62]	r-NSGA-II	$< IAR, E_{RP}, VF + DoR, NMP >$	The *r-dominance* relation is proposed. Solutions close to the reference point in the sense of weighted Euclidean distance are preferred.
López-Jaimes and Coello [85]	Chebyshev preference relation	$< IAR, E_{RP}, VF + DoR, NMP >$	The *Chebyshev preference* relation is developed by combining an ASF with the Pareto dominance relation.
Ruiz et al. [48]	Interactive WASF-GA	$< IAR, E_{RP}, VF, NMP >$	A set of ASFs is formulated with a reference point and a predefined set of weighting vectors to rank solutions.
Narukawa et al. [55]	P-NSGA-II	$< IAR, E_{OT}, VF, NMP >$	The DM's preferences are represented by Gaussian functions. The center and spread vectors of Gaussian functions are provided by the DM.
Shen et al. [37]	FLMOEA	$< IDR, CO_{WE}, VF{+}DoR{+}DeR, NMP >$	A *strength superior* relation is constructed based on the DM's preferences via a fuzzy inference system.
Yang [86]	GRIST	$< IAR, CO_{TR}, VF, MP >$	The projection of the marginal rates of substitution is used to obtain a decent direction of the underlying disutility function.
Luque [87]	PROJECT	$< IAR, CO_{TR}, VF, MP >$	The GRIST method is combined with the reference point method.
Chen et al. [26]	T-IMO-EA	$< IAR, CO_{TR}, VF, NMP >$	An EA is used as the search engine to generate approximate Pareto optimal solutions under the framework of the GRIST method.
Miettinen and Mäkelä [16]	NIMBUS	$< IAR, CO_{CL}, VF, MP >$	The DM is asked to classify objectives into up to five classes to obtain a more desirable Pareto optimal solution at each iteration.
Miettinen and Mäkelä [88]	Synchronous NIMBUS	$< IAR, CO_{CL}, VF, MP >$	Multiple scalarizing functions are utilized to generate multiple Pareto optimal solutions simultaneously at each iteration.
Miettinen et al. [89]	NAUTILUS	$< IAR, CO_{OT}, VF, MP >$	All objective values are improved simultaneously step by step by starting from the nadir point.
Sindhya et al. [24]	PIE	$< IAR, CO_{OT}, VF, NMP >$	Nearly the same philosophy as the NAUTILUS method is followed. However, an EA is employed as the search engine.
Phelps et al. [90]	IEM	$< IDR, CS_{PCS}, VF, NMP >$	A linear VF is utilized to model the DM's pairwise comparisons of solutions to evaluate the population.
Deb et al. [33]	PI-EMO-VF	$< IDR, CS_{PCS}, VF + DoR, NMP >$	A dominance relation which combines the Pareto dominance relation with a polynomial VF compatible with the DM's preferences is defined.
Sinha et al. [91]	Advanced PI-EMO-VF	$< IDR, CS_{PCS}, VF + DoR, NMP >$	A generalized polynomial VF which fits any kind of convex preference is adopted to advance PI-EMO-VF.
Branke et al. [52]	NEMO-I	$< IDR, CS_{PCS}, VF + DoR, NMP >$	A new dominance relation is established by using the whole set of additive VFs compatible with the DM's preferences.
Branke et al. [53]	NEMO-0	$< IDR, CS_{PCS}, VF, NMP >$	A representative additive VF is used to calculate a new distance instead of the crowding distance in NSGA-II.
Brattiti and Passerini [57]	BC-EMO	$< IDR, CS_{PCS}, VF, NMP >$	Support vector machines are utilized to learn the DM's underlying VF.
Pedro and Takahashi [58]	INSPM	$< IDR, CS_{PCS}, VF, NMP >$	A radial basis function network is used to approximate the DM's underlying VF.
Greco et al. [39]	DRSA-EMO-PCT	$< IDR, CS_{PCS}, DoR + DeR, NMP >$	Decision rules are induced from the DM's pairwise comparison of solutions to define a preference relation and a diversity measure.
Greco et al. [39]	DRSA-EMO	$< IAR, CS_{CLS}, DeR, NMP >$	Decision rules are induced from the DM's classification of solutions to rank solutions.

TABLE 2. ***(Continued.)*** **Representative IMO methods.**

Authors and References	Names	Category Features	Main Ideas
Chugh et al. [92]	I-SIBEA	$< IDR, CS_{CLS}, VF, NMP >$	A weight distribution function is defined by asking the DM to specify preferred and non-preferred solutions. Solutions are selected to the next generation based on their contributions to the weighted hypervolume.
Sinha et al. [60]	PI-EMO-PC	$< IDR, CS_{SMP}, DoR, NMP >$	A polyhedral cone is built by using the best solution selected by the DM and the end points of the current population to modify the Pareto dominance relation.
Fowler et al. [93]	Cone dominance relation	$< IDR, CS_{SMP}, DoR, NMP >$	The best and worst solutions specified by the DM are utilized to construct convex preference cones to define the cone dominance relation.
Gong et al. [94]	iMOEA/D	$< IDR, CS_{SMP}, VF, NMP >$	The weighting vector of the best solution selected by the DM is used to renew the preferred weight region.
Köksalan and Karahan [95]	iTDEA	$< IDR, CS_{SMP}, VF, NMP >$	A territory is defined for each solution. Smaller territories are defined around preferred solutions to obtain denser solutions in the DM's ROI(s).
Wang et al. [96]	iPICEA-g	$< IDR, E_{RP}/CO_{WE}, VF, NMP >$	Preferences either as aspiration levels or weights can be handled. Candidate solutions and specially generated goal vectors are co-evolved to guide candidate solutions towards the ROI(s).
Hakanen et al. [97]	interactive RVEA	$< IDR, E_{RP/OT}/CS_{CLS}, VF, NMP >$	Different types of preferences including reference points, preferred ranges of objective values, and preferred or non-preferred solutions are transformed into reference vectors to guide the search of the population.

For an IMO method using a hybrid preference model, its category feature over PM is represented in the form $PM1 + PM2 + \ldots$ where PMi represents a single preference model. In what follows, the chosen IMO methods are briefly introduced according to the following four categories, i.e., the *expectation*-based methods, methods based on the *comparison of objective functions*, methods based on the *comparison of solutions*, and methods *catering to different types of preferences*.

A. EXPECTATION-BASED METHODS

1) REFERENCE POINT BASED METHODS

In the 1980s, Wierzbicki [13] proposed the reference point method in which the reference point specified by the DM and k perturbed reference points are projected onto the Pareto fronts by using ASFs as shown in Fig. 6. Then, in 1993,

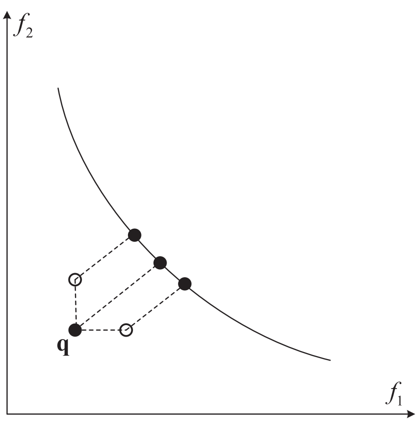


FIGURE 6. **The reference point and perturbed reference points (the hollow circles) are projected onto the Pareto front by ASFs.**

Fonseca and Fleming [59] perhaps proposed the first interactive reference point based MOEA [27]. The idea is to use the DM's preferences to evaluate solutions in a population of GA at each generation. The main disadvantage is that the DM's workload is very high as he/she has to provide preferences at every generation of GA. In the past decade, the reference point based MOEAs have become popular in IMO.

Deb et al. [42] proposed the R-NSGA-II which combines the reference point method with NSGA-II. The DM can provide several reference points simultaneously. For each reference point, the weighted Euclidean distance from it to every individual in the current population is calculated using (4). Individuals close to all reference points are preferred. To control the extent of obtained solutions, an ϵ-clearing strategy is adopted. It is important to note that in the case of a single reference point, the diversity of the population of R-NSGA-II is likely to reduce [42]. Said *et al.* [62] remarked that more than one reference point must be handled by R-NSGA-II to achieve satisfactory results.

In [47], the RD-NSGA-II which hybridizes the reference direction approach with NSGA-II was developed. In the reference direction approach [98], the reference direction is a vector from the current iteration point **p** to the reference point **q** specified by the DM [99]. At every iteration, a number of points lying on the reference direction are projected onto the Pareto front by using ASFs to obtain multiple Pareto optimal solutions as shown in Fig. 7. RD-NSGA-II borrows the concept of the reference direction approach. It projects a set of points on the reference direction onto the Pareto front by an EMO procedure. ASFs formulated based on these points are used to sort the population into multiple fronts. This sorting procedure can be easily generalized to handle the case where multiple reference directions are

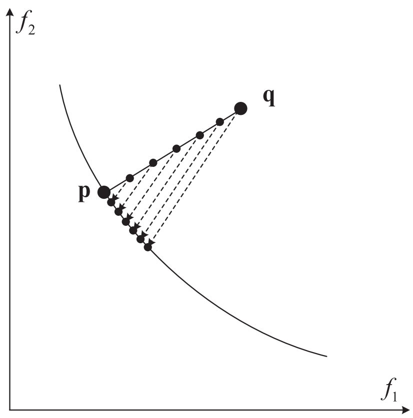


FIGURE 7. An example of projecting a reference direction onto the Pareto front.

considered simultaneously. However, RD-NSGA-II may also face the problem of a degradation of the population diversity when using a single reference direction since it does not have a clearing mechanism [28].

By borrowing the concept of light beam search (LBS) [100], the LBS-NSGA-II was proposed in [83]. The LBS approach combines the reference point idea and the tools of multi-attribute decision analysis. The DM needs to provide a reference point and a reservation point to determine a search direction. By using an augmented ASF, the reference point is projected onto the Pareto front. An outranking relation is used to find neighboring solutions of the current solution. To define the outranking relation, the DM also needs to specify indifference, preference, and veto thresholds for every objective. To reduce the DM's burdens, only veto thresholds are used in LBS-NSGA-II and the outranking relation used in LBS is modified accordingly. After the non-dominance ranking, each solution in each front is assigned a crowding rank by using an ASF and the modified outranking relation. LBS-NSGA-II can be easily modified to find multiple ROIs at each iteration. It is worth noting that providing veto thresholds may still be not easy for the DM.

Combining the reference point method and the indicator-based evolutionary algorithm (IBEA) [84], Thiele *et al.* [49] presented the preference-based evolutionary algorithm (PBEA). PBEA uses an ASF to modify the binary quality indicator of IBEA which is used for fitness calculation. In this way, the DM's preferences are incorporated into the fitness evaluation. Note that the modified binary quality indicator is Pareto dominance preserving. PBEA can also be easily generalized to handle the case where more than one reference point is provided. The extent of the ROIs can be adjusted by a parameter. However, the authors noted that adjusting the parameter is a topic for future research.

In Section II-C, we have described the *g-dominance* relation and the *r-dominance* relation. The *g-dominance* relation can be coupled easily into any Pareto-based MOEA without having to modify the main architecture of the MOEA. It was integrated into two metaheuristics in [61] and results showed that each of them can find approximate Pareto optimal solutions adapted to the reference points. One drawback is that it may contradict the order of solutions induced by the Pareto dominance.

The *r-dominance* relation is able to create a strict partial order between non-dominated solutions by modifying the Pareto dominance relation and preferring solutions closer to the reference point while preserving the Pareto dominance. It is integrated into NSGA-II (the new algorithm is referred to as r-NSGA-II) in which the non-dominated sorting procedure is replaced by the non-r-dominated sorting. A parameter is used to control the selection pressure and thus control the breadth of the ROI. However, difficulties arise when solving highly multimodal problems and thus strategies allowing escaping from local optima are necessary [62]. Note that r-NSGA-II could be extended to explore multiple ROIs corresponding to multiple reference points at the same time [62]. It could also be used with other metaheuristics like ant colony optimization, particle swarm optimization, and so on.

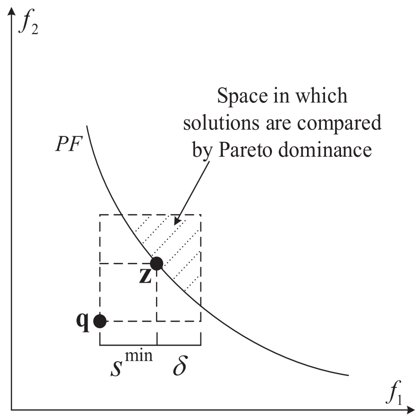


FIGURE 8. Illustration of how the feasible objective space is divided by the Chebyshev preference relation. Solutions whose ASF values are not larger than the minimal value of the ASF (s^{min}) plus a threshold (δ) are compared by the Pareto dominance relation. Solutions in the remaining part are compared by the ASF.

Another dominance relation called the *Chebyshev preference* relation was developed in [85]. It divides the feasible objective space into two parts based on the minimal value of the ASF shown in (6), denoted by s^{min}, and a threshold δ as shown in Fig. 8. The Pareto dominance relation is used to compare solutions whose ASF values are not larger than $s^{min} + \delta$ and the ASF is used to compare solutions in the remaining part. The Chebyshev preference relation can also be incorporated into any Pareto-based MOEA requiring only slight modifications of its structure. An inconvenience is that the setting of the threshold δ is not trivial in practice.

Aiming at obtaining a well-distributed set of non-dominated solutions approximating the ROI defined by the reference point, the weighting achievement scalarizing function genetic algorithm (WASF-GA) uses the reference point and a pre-defined set of weighting vectors to formulate a set of ASFs. These ASFs are used to classify the population of an MOEA into multiple fronts. An interactive version of WASF-GA was proposed in [48]. Besides a reference point, the DM is also allowed to specify the number of solutions he/she wants to see at each iteration. This number is expected to be much lower than the population size used by other methods. Therefore, the interactive WASF-GA has a much lower computational cost than that of other methods. A shortcoming is that the distribution of obtained solutions is influenced by the distribution of the weighting vectors.

To sum up, most of the above reference point based interactive MOEAs can explore multiple ROIs simultaneously. Some of them allow adjusting the spread of obtained solutions by using parameters. However, the setting of these parameters is not trivial when solving real-world problems.

2) OTHER METHODS

Narukawa *et al.* [54] proposed the preference-based NSGA-II (P-NSGA-II) which represents the DM's preferences by Gaussian functions on a hyperplane. The DM is asked to specify the center and spread vectors of Gaussian functions. A preference function is calculated by using Gaussian functions and is used to replace the crowding distance to compare solutions in NSGA-II. To obtain dense solutions around center vectors, Narukawa et al. [55] extended P-NSGA-II by taking into account an allowable radius and a shortest distance. The DM can change the center and spread vectors during the optimization to change the distribution of current solutions. Experiments showed that the extended P-NSGA-II can easily adapt its search to the preferred region(s) of the DM. It can also deal with the case where multiple preferences are provided by the DM.

In several preference-based MOEAs which integrate desirability functions with MOEAs, the DM is asked to specify parameters of a desirability function a priori for each objective to represent his/her desires of objective values [101]–[103]. Each objective is mapped to the domain $[0, 1]$ by a desirability function which is monotonically decreasing with the objective function and is to be maximized. In [101] and [102], the original MOP is transformed into a new MOP in which all desirability functions are to be maximized simultaneously. An MOEA is used to solve the new MOP instead of the original one. Due to the monotonicity of the desirability functions, the dominance relations between solutions in the objective space will not be violated in the space of desirability functions. In [103], the proposed algorithm operates in the original objective space and the desirability index which scalarizes all the desirability functions into a univariate quality index is used to select solutions after the non-dominated sorting. These algorithms are all a priori methods and they may be used in an interactive manner by asking the DM to change the parameters to form new desirability functions which focus on new objective regions if he/she does not find a satisfactory solution currently.

B. METHODS BASED ON THE COMPARISON OF OBJECTIVE FUNCTIONS

1) WEIGHTS-BASED METHODS

Firstly, we introduced two a priori weights-based methods. Cvetković and Parmee [104] proposed an approach of transforming the DM's qualitative preferences into weights of objectives. The relative importance of objectives is divided into several grades such as "much more important," "more important," and "equally important." The DM is asked to compare pairs of objectives and specify one of the grades. Transitive relations are used to resolve the problem of preference contradictoriness and reduce the number of pairwise comparisons required from the DM. The weights of objectives are calculated according to a preference matrix. Based on the weighting vector $\mathbf{w}$, a modified Pareto dominance relation, namely the weighted-dominance relation, is defined: for two solutions $\mathbf{x}$, $\mathbf{y}$ and a parameter $0 < \tau < 1$, $\mathbf{x}$ weighted-dominates $\mathbf{y}$ if and only if

$$\sum_{\forall i: f_i(\mathbf{x}) \leq f_i(\mathbf{y})} w_i \geq \tau. \tag{8}$$

The main weakness of the weighted-dominance relation is that only the number of improvements of one solution over another is considered while the amount of improvements are ignored [28].

Pointing out that converting fuzzy preferences into single-valued weights will result in loss of information, Jin and Sendhoff [105] developed an approach to convert a DM's fuzzy preferences into weight intervals. The DM is also asked to make pairwise comparisons on objectives. The induced weight intervals are incorporated into an MOEA by using random weighted aggregation and dynamic weighted aggregation approaches. However, the authors noted that the incorporation of fuzzy preferences using the dynamic weighted aggregation method is applicable to convex Pareto fronts only.

Considering both the importance factors of objectives and the improvements of solutions, Shen *et al.* [37] proposed the interactive MOEA based on fuzzy logic (FLMOEA). The DM is expected to specify relative importance between pairs of objectives by using linguistic terms when the generation counter satisfies certain conditions. The importance factors of objectives are computed accordingly and they together with the degrees of improvements between two solutions are used to construct a *strength superior* relation via a fuzzy inference system. Using this relation, a fitness function is defined in order to guide the population towards the ROI. Experiments showed that when the DM's preferences are changed, FLMOEA is able to adjust the search direction and the range of solutions. The comparative experiment with the random weighted aggregation based algorithm in [105] showed that FLMOEA can find solutions more centralized in the desired region corresponding to the DM's preferences. However, the time complexity of FLMOEA is larger.

2) TRADEOFF-BASED METHODS

In [86] and [106], the gradient-based interactive step tradeoff (GRIST) method was proposed. It estimates the gradient of an underlying VF by using the indifference tradeoffs provided by the DM. The projection of the gradient onto the tangent hyperplane of the Pareto front provides a new search direction along which the DM's utility can be improved. A unique feature of the GRIST method is that the established necessary optimality conditions can facilitate the elicitation of the DM's preferences and hence reduce the DM's burdens. Besides, its convergence is proved. The GRIST method can be applied to both linear and nonlinear MOPs where the Pareto fronts could be nonconvex or nonsmooth at finite points. However, the framework of the gradient projection is not suited to be used on MOPs with disconnected Pareto fronts because it may lead to a local optimum of the underlying VF instead of the real MPS.

Indicating that the convexity conditions in an MOP are assumed in most interactive tradeoff-based methods, Luque *et al.* [87] developed the PROJECT method using local tradeoffs. The PROJECT method combines the gradient projection framework of the GRIST method with the reference point method, which facilitates generating Pareto optimal solutions that can reflect the DM's local tradeoffs. To be specific, after calculating the projection of the DM's local indifference tradeoffs, a new reference point is defined and projected onto the Pareto front to obtain a new solution which maintains the local tradeoffs. In this way, the PROJECT method provides a better way of searching for a new solution by using the reference point method compared to the GRIST method.

Since the GRIST method uses MP techniques to generate Pareto optimal solutions while MP techniques often require the convexity or differentiability of optimization problems, the applicability of the GRIST method in solving MOPs may be limited. Chen *et al.* [26] developed the tradeoff-based interactive multiobjective optimization method driven by evolutionary algorithms (T-IMO-EA) to enhance the versatility of the GRIST method by combining it with EAs. EAs are used as the search engine to generate approximate Pareto optimal solutions. An approach of approximating the normal vector of the tangent hyperplane of the Pareto front was proposed for determining the tradeoff direction. Experiments showed that T-IMO-EA has a better convergence than the GRIST method on the used test MOPs.

3) METHODS BASED ON THE CLASSIFICATION OF OBJECTIVES

A classical IMO method based on the classification of objectives is the nondifferentiable interactive multiobjective bundle-based optimization system named NIMBUS [16]. At every iteration, it allows the DM to classify objectives into up to five classes at the current solution so as to find a more desirable solution. A function combining a weighted distance metric with an ASF is used to transform the MOP into a constrained SOP. NIMBUS is applicable to both linear and nonlinear problems involving continuous or integer-valued variables. It has been implemented as a WWW-NIMBUS software system which is a World-Wide Web based interactive optimization software on the Internet [107].

Since different scalarizing functions may result in different Pareto optimal solutions under the same preference information, a synchronous NIMBUS was developed in [88] with the basic idea of using several scalarizing functions based on the same preference information to generate multiple Pareto optimal solutions at each iteration. As a consequence, the DM is provided with several solutions and he/she can judge which one is the most preferred at each iteration. The synchronous NIMBUS was implemented as the version 4.0 of the WWW-NIMBUS system containing four different scalarizing functions [88].

4) OTHER METHODS

Aiming at avoiding undesired anchoring effects which means that the DM fixes his/her thinking on some information even though the information may be irrelevant, Miettinen *et al.* [89] proposed the NAUTILUS method. This method starts from the nadir point and improves all objectives simultaneously step by step as demonstrated in Fig. 9. The DM is asked to specify the number of iterations that he/she wants to carry out first to control the speed of approaching the Pareto front. At each iteration, he/she is also asked to rank the relative importance of improving each objective or specify percentages how he/she would like to improve the current objective values at the current iteration solution which lies on the segment joining the previous iteration solution and its corresponding projection on the Pareto front. Note that the DM is allowed to change the number of remaining iterations and take a step backwards. A significance of the NAUTILUS method is that the DM does not need to make tradeoffs among objectives since all the objectives values are always improved.

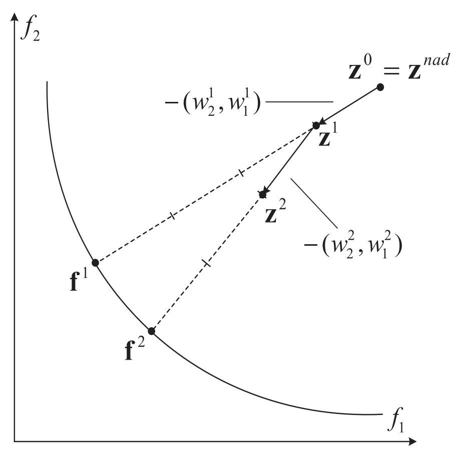


FIGURE 9. **The idea of the NAUTILUS method.**

Following this research, Sindhya *et al.* [24] developed a preference-based interactive evolutionary algorithm (PIE) which follows nearly the same philosophy as the NAUTILUS method. The main difference is that PIE is an "EA in MCDM" approach in which an EA is used as the search engine instead of an MP technique to minimize an ASF at each iteration. What's more, at each generation of the EA, the solution with the smallest ASF value and the entire population are both stored in archive sets so that the DM can get access to previous solutions easily. The authors noted that storing many solutions is not a disadvantage nowadays due to the high performance and large memory of current computers.

C. METHODS BASED ON THE COMPARISON OF SOLUTIONS

In this subsection, we divide methods based on the comparison of solutions into three categories, i.e., methods based on pairwise comparisons, methods based on the classification of solutions, and methods based on selecting the most preferred solution.

1) METHODS BASED ON PAIRWISE COMPARISONS

Asking the DM to compare a set of solutions pairwise is popular in interactive MOEAs. In the following, several representative interactive MOEAs based on pairwise comparisons are briefly described. They all ask for the DM's pairwise comparisons on solutions periodically during the evolution process of an EMO procedure to guide the population towards the DM's ROI. The main differences among them lie in the construction of the preference model and the usage of the preference model to guide the search.

In the interactive evolutionary metaheuristic algorithm (IEM), the DM's pairwise comparisons of solutions are turned into constraints of a linear programming (LP) problem whose optimal solution is the weights of an estimated VF which has the form of a linearly weighted sum of objectives [90]. The estimated VF is used as the fitness function to evaluate the population. Note that the LP problem may be infeasible because the DM's underlying VF may be far from the form of the estimated VF. In this case, the preference constraints will be iteratively removed from the oldest until a feasible solution is found. The IEM is computationally efficient due to the use of LP to derive the VF. However, the restriction to a linear VF is a limiting factor [53].

The progressively interactive EMO approach using value functions (PI-EMO-VF) asks the DM to give a complete ranking of several solutions (five in the paper) in order to determine a polynomial VF by solving a VF optimization problem [33]. Note that the VF optimization problem is nonlinear and the authors propose to use sequential quadratic programming to solve it. The derived polynomial VF is used to modify the Pareto dominance relation to distinguish solutions. A termination criterion was developed with the principle that solutions more preferred than the current best solution can be found along the gradient of the VF from the best solution. If the current best solution cannot be significantly improved, the algorithm will be terminated. The drawbacks of PI-EMO-VF are that it does not handle the case where the DM finds some solutions incomparable and high computational cost is required in sequential quadratic programming.

Sinha *et al.* [91] advanced PI-EMO-VF by using a generalized polynomial VF which fits any kind of convex preference information. Moreover, a partial ordering of the given points is allowed by considering the case where the DM thinks that some solutions are incomparable.

In [52], the necessary-preference-enhanced evolutionary multiobjective optimizer (NEMO) was developed by combining an interactive procedure based on robust ordinal regression (ROR) [108] with NSGA-II. The whole set of general additive VFs compatible with the DM's pairwise comparisons of solutions is considered, which results in two rankings: the necessary ranking and the possible ranking. The necessary ranking is used to replace the Pareto dominance relation and the most representative VF recognized based on the two rankings is used to define a new crowding distance. A shortcoming is that $O(m^2)$ LP problems have to be solved to rank m solutions, which is computationally expensive [53].

Branke *et al.* [53] categorized NEMO as NEMO-I approach and proposed a variant of the NEMO framework named NEMO-0. The DM only compares one pair of solutions shown to him/her at each iteration. Only a representative additive monotonic VF compatible with all preferences of the DM is considered. It is used to calculate a distance to replace the crowding distance of NSGA-II. Compared with NEMO-I, NEMO-0 requires less preference information and reduces the computational complexity.

Being able to trade off the model accuracy and complexity, support vector machines have been trained in the brain-computer EMO algorithm (BC-EMO) to learn the DM's VF from his/her pairwise comparisons of solutions [57]. The predicted VF is used to sort solutions and the sorting procedure can be embedded into any MOEA to guide the selection of solutions. The authors have implemented BC-EMO on top of NSGA-II and the crowding distance mechanism is switched off after the preference model is trained with the goal of directing the generating of new solutions in the ROI. The main contribution of BC-EMO is that it does not make any a priori assumptions on the shape of the DM's VF, which is more practical than assuming that the DM's VF has a specific form. However, turning off the crowing mechanism after certain generations can result in a significant reduction of the population diversity and the premature convergence especially on multimodal MOPs [28].

A radial basis function network is used in the interactive non-dominated sorting algorithm with a preference model (INSPM) to approximate the DM's underlying VF [58]. A dynamic crowding distance is employed instead of the original crowding distance in NSGA-II to allow denser solutions in the DM's preferred regions and coarser in the non-preferred regions. A Kendall-tau distance (KTD) value is calculated at each generation to measure the accuracy of the approximated VF. If the KTD value does not satisfy the tolerance, the preference model will be updated by asking for the DM's new preferences. The INSPM has a parameter to control the sampling density along the Pareto front. Higher parameter values result in a more concentrated sampling near the DM's ROI. However, as the parameter influences the final results, how to set the parameter remains a problem in practice.

The above interactive MOEAs all model the DM's preferences as VFs. In the dominance-based rough set approach to EMO using pairwise comparison tables (DRSA-EMO-PCT) algorithm, the DRSA is applied in interactive EMO to model the DM's pairwise comparisons of solutions as decision rules [39]. The induced decision rules are used to define a preference relation and compute a diversity measure for selecting solutions. As stated in Section II-B, a preference model represented as decision rules is more general and comprehensible than a functional or relational model.

2) METHODS BASED ON THE CLASSIFICATION OF SOLUTIONS

In [39], another algorithm named DRSA-EMO which also uses decision rules as the preference model was developed. Different from DRSA-EMO-PCT, DRSA-EMO asks the DM to classify a set of solutions into "relatively good" and "others." At each iteration, a complete run of an EMO procedure guided by the induced decision rules is performed. During the evolution process, solutions are ranked using a primary score based on the number of DRSA rules matching each of them and a secondary score based on the crowding distance. In the first interaction, the solutions presented to the DM for classification are generated using Monte Carlo method. In other interactions, the presented solutions are from the last population of the EMO procedure. Since DRSA can handle the problem of decision under uncertainty, both DRSA-EMO-PCT and DRSA-EMO can take into account robustness concerns [39].

Chugh *et al.* [92] proposed the interactive simple indicator-based evolutionary algorithm (I-SIBEA) which allows the DM to specify the number of times he/she wants to interact with an EMO procedure. The number of generations which should be carried out between two adjacent interactions is calculated accordingly. The DM is expected to select preferred and non-preferred solutions from a set of solutions. Based on these preferences, the feasible region is partitioned into three parts and a weight distribution function is defined to calculate the weighted hypervolume in the subsequent generations. Solutions are selected to the next generation based on their contributions to the hypervolume. An advantage of I-SIBEA is that the use of both preferred and non-preferred solutions offers the DM more flexibility to guide the search. Experiments have shown that the information of non-preferred solutions makes the algorithm converge faster to the DM's ROI.

3) METHODS BASED ON SELECTING THE MOST PREFERRED SOLUTION

In this part, we introduce four interactive MOEAs which ask the DM to select the most preferred solution from a set of solutions periodically to guide the search of an MOEA towards the DM's ROI.

In the progressively interactive EMO approach based on polyhedral cones (PI-EMO-PC) [60], the DM selects the most preferred solution from the archive set periodically by using an approach named VIMDA [109]. The best solution and the k end points of the non-dominated front from the current population are used to construct a polyhedral cone for modifying the Pareto dominance relation. Each end point has the best function value for the corresponding objective. The polyhedral cone is further utilized to get a search direction which is used to determine whether the algorithm should be terminated or not.

Fowler *et al.* [93] also developed an interactive MOEA based on convex preference cones. Periodically, the DM specifies the best and worst solutions from several solutions presented to him/her. The two solutions are used to construct convex preference cones with which inferior solutions are defined. All preference cones are retained throughout all generations whether or not the solutions from which the cones are derived are still surviving. The defined cone dominance is used to order the population. Note that this algorithm assumes that the DM's underlying VF is quasi-concave and it is guaranteed to produce partial orders consistent with the preferences induced from quasi-concave functions. However, the authors stated that investigating the effect of the DM's inconsistencies is worth future research.

The above two methods both construct preference cones based on the DM's preferences for defining new dominance relations. In the next two methods, solutions are corresponding to weights and the best solution selected by the DM is used to update the preferred weight region. In the interactive MOEA/D algorithm (iMOEA/D) proposed in [94], the weighting vector of the selected best solution is used to renew the preferred weight region which is a hyper-sphere with the preferred weighting vector being the center. The weighting vectors outside the preferred weight region are relocated inside this region to focus the search around the DM's preferred solutions. The authors noted that by operating on weighting vectors, the DM learns more direct and precise knowledge about the possible Pareto front and determines the optimizing direction better.

In the interactive territory defining evolutionary algorithm (iTDEA) [95], a territory is defined around each individual and the favorable weights of the best solution selected by the DM, i.e., weights which minimize the weighted Tchebycheff distance of the solution to the ideal point, are identified to determine a new preferred weight region. A smaller territory size parameter is assigned to this region to obtain denser solutions around the preferred solution. It is also possible for the DM to select several best solutions to allow the algorithm to concentrate on several regions synchronously. For practical considerations, filtering out a representative set of solutions from the population to present to the DM at every iteration is necessary. However, the authors pointed out that filtering may mislead the algorithm in the early interactions. Thus, a better filtering mechanism is needed to ensure a desirable representation of the population.

D. METHODS CATERING TO DIFFERENT TYPES OF PREFERENCES

Compared to other methods, methods allowing the DM to provide different types of preferences offer the DM more choices, which is very helpful if the DM is not good at specifying a special type of preferences.

In [96], an interactive MOEA named iPICEA-g which is an interactive version of the preference-inspired co-evolutionary algorithm using goal vectors (PICEA-g) [110] was proposed. It can handle the DM's preferences either as aspiration levels or as weights of objectives. In fact, instead of specifying preferences by using any numeric values, the DM can simply brush his/her preferred regions in the objective space and iPICEA-g can configure the required parameters like aspiration levels and weights automatically based on the brushed regions. The population of candidate solutions and specially generated goal vectors corresponding to the DM's ROI(s) are co-evolved to guide candidate solutions towards the ROI(s). The major benefits of iPICEA-g are that it caters to preferences in the form of either aspiration levels or weights and multiple ROIs can be explored simultaneously.

The interactive RVEA which is based on the reference vector guided evolutionary algorithm (RVEA) [111] has also been developed to handle various types of the DM's preferences [97]. The types of preferences considered include specifying a reference point, providing preferred ranges for each objective, and selecting preferred solution(s) or non-preferred solution(s). The interactive RVEA is able to transform them into a single format, i.e., reference vectors, to guide the population towards the DM's preferred region by using an angle penalized distance scalarization.

E. SUMMARY

From Table 2 and the review of representative IMO methods, it can be seen that IMO methods adopting the IAR pattern often ask the DM to provide reference points, the classification of objectives, or tradeoffs. IMO methods employing the IDR pattern usually require the DM to compare objective functions in pairs or make comparisons on solutions.

The IMO methods developed in the last decade are mainly hybrid methods combining MCDM and EMO techniques in which "MCDM in EMO" based approaches predominate. They incorporate the DM's preferences into MOEAs mainly by defining a preference-based fitness function or replacing the Pareto dominance relation with a new dominance relation. A preferred region instead of a single solution is found, which can provide the DM with more information.

Some interactive MOEAs use the IAR pattern to find the DM's ROI at each iteration. Most of them combine the reference point method with MOEAs, which may be because the reference point can be specified freely by the DM and it is easy to be used interactively. Multiple ROIs can be found simultaneously by specifying multiple reference points. This is beneficial when the DM does not have an exact preferred reference point or he/she wants to search for several ROIs. Interactive MOEAs under the IDR pattern allow the DM to interrupt an MOEA to compare objective functions or a set of solutions so as to guide the population of the MOEA towards the DM's preferred region. Because only a single complete run of an MOEA is performed, interactive MOEAs with the IDR pattern is less time-consuming than those with the IAR pattern. Moreover, the optimization process is more DM-oriented because the DM can provide preferences more frequently [33]. One problem is that the solutions shown to the DM at intermediate generations may not give the DM a good idea about the real tradeoffs among objectives [48].

Several IMO methods which transform MOPs into SOPs at each iteration have also been proposed in recent years. Among them, two "EA in MCDM" based IMO methods, namely PIE and T-IMO-EA, combine the NAUTILUS method and the GRIST method with EAs, respectively. They use EAs as search engines in an attempt to handle problems with nonconvexity, discontinuity, and so on. The other methods use MP techniques to generate Pareto optimal solutions at each iteration.

IV. CRUCIAL ISSUES IN IMO

According to the review of representative state-of-the-art IMO methods, we summarize eight crucial issues which need deliberations in IMO. Based on the DM-Machine interaction system presented in Section II, these issues in the sequel are divided into three classes, i.e., the DM-related issues, the machine-related issues, and issues related to the whole interaction system.

A. DM-RELATED ISSUES

1) HOW TO REDUCE THE DM'S BURDENS? (*DM1*)

The DM's burdens may be different when the DM interacts with different IMO methods. High burdens may cause the DM to make mistakes when he/she is providing preferences or to terminate an IMO method prematurely. So, reducing the DM's burdens is an important issue.

There are mainly two kinds of burdens. One is the cognitive burden of the DM when providing preferences. As stated in Section II-E, specifying different types of preferences tends to demand different cognitive burdens from the DM. Besides, different DMs may prefer providing different types of preferences. Thus, their cognitive burdens for providing the same type of preferences are likely to be different. Asking the DM to provide preferences which he/she prefers providing can reduce his/her burdens. In this sense, offering the DM different options of expressing preferences in the solution process is a good way of reducing the DM's burdens. Moreover, visualizing solutions graphically can promote the

DM's understanding and analysis of the solutions shown to him/her and thus reduce his/her burdens. Common visualization techniques include bar charts, scatter plots, value paths, spider-web charts, and so on (see [112], [113] for comprehensive surveys).

The other kind of burden is the number of interactions between the DM and the machine. Too many interactions may make the DM feel tired and more likely to make mistakes. Some IMO methods aim at finding the DM's MPS within a limited number of interactions [89], [92], [114]. For example, in [114], only when enough progress has been made along the direction from the initial point to the ideal point, the DM is asked to provide new preferences.

2) HOW TO HANDLE COGNITIVE BIASES OF THE DM? (*DM2*)

Due to mental or physical factors, a DM cannot be fully rational. A cognitive bias refers to the difference between the DM's decision and the rational decision. Handling the DM's cognitive biases properly may reduce the possibility of finding a final solution far from the real MPS.

Anchoring is one kind of bias relevant to IMO. Buchanan and Corner [115] studied the effects of anchoring in two IMO methods and developed two measures of the anchoring bias. They concluded that the starting point matters and a more directed and structured solution method is more likely to support anchoring than methods based on free search. Besides, since any starting point is likely to bias the DM, starting at a point which reflects the initial preferences of the DM should be considered. The NAUTILUS method described in Section III avoids anchoring by starting from the nadir point so that each objective can be improved and the DM can reach any Pareto optimal solution [89]. It can be used in other IMO methods to find a non-anchored starting point.

Another relevant bias is loss aversion which means that people prefer avoiding losses to obtaining equivalent gains [116], [117]. It may hinder a DM from shifting from a Pareto optimal solution to another because he/she has to sacrifice one or more objectives. In the NAUTILUS method, since all objectives can be improved simultaneously, the DM always attains gains and does not have to make tradeoffs between objectives.

3) HOW TO HANDLE DIFFERENT TYPES OF PREFERENCES? (*DM3*)

Since most existing IMO methods ask for one specific type of preferences from the DM, if the DM is not good at providing such preferences, he/she will probably make mistakes. So, it is necessary to develop IMO methods which are able to handle different types of preference information. Some researchers have developed general interaction systems which integrate several IMO methods together and allow the DM to choose freely the type of preference information he/she wants to provide at each iteration (see [118]–[121]).

Luque *et al.* [122] studied the relations among different types of preferences. If different types of preferences produce the same solution when they are used in their corresponding IMO methods, they are called equivalent information. By studying the equivalency of different types of preferences, the previous preferences provided by the DM can be transferred into the same type of preferences that he/she wants to provide at the current iteration, which can assist the DM to provide new preferences. However, the IMO methods considered in [122] are all MCDM methods and ask for quantitative preferences. For some interactive MOEAs, it is difficult to study the relations among different types of preferences because the preferences may be qualitative and the preference models may not be functions.

4) HOW TO HANDLE INCONSISTENT PREFERENCES? (*DM4*)

Inconsistent preferences mean that the preferences provided by the DM contain contradictory information or are incompatible with the preference model of an IMO method. Contradictory information means that one or more loops exist in the DM's ranking or sorting of a set of solutions. In other words, there are at least three solutions **a**, **b**, and **c** which satisfy that **a** is preferred to **b**, **b** is preferred to **c**, but **c** is preferred to **a**. Contradictory information may be produced because of mistakes or the evolution of the DM's preferences. It can be eliminated by deleting or changing the preference relations which cause the contradiction. Some IMO methods delete the oldest or inconsistent preferences to obtain a compatible preference model [53], [90], [123]. However, Greco *et al.* [64] pointed out that inconsistencies cannot be simply removed because they may contain important information to build preference models.

B. MACHINE-RELATED ISSUES

1) HOW TO CHOOSE/CONSTRUCT AN APPROPRIATE PREFERENCE MODEL? (*MA1*)

As various preference models have been developed to model the DM's preferences, the applicability and effectiveness of preference models deserves in-depth investigation. One important concern is the relation among different preference models with respect to their ability to model different types of preferences. If existing preference models can be divided into equivalence classes, an appropriate preference model for one or several certain types of preferences can be selected from related equivalence classes according to some criterion like ease of use, high accuracy of modelling the DM's preferences, and low computational cost. In order to adapt to more types of preferences, an ensemble of preference models can be constructed by integrating appropriate preference models from different equivalence classes.

2) HOW TO FIND THE DM'S MPS QUICKLY? (*MA2*)

For IMO, it is necessary to find the DM's MPS quickly because it is impractical for the DM to spend a long time participating in the solution process. Firstly, the interaction time spent by the DM per iteration should be reduced. This can be done by providing the DM with lucid information whenever possible. Although more information may increase the DM's confidence in the solution obtained, the percentage of the information used decreases and the solution may be worse [3], [124]. Secondly, the DM may not be willing to wait a long time for the output of the machine. So efficient search engines are very appealing. Finally, as mentioned in the issue *Dm1*, the number of interactions should not be too large.

For methods that generate one or more Pareto optimal solutions per iteration, the above mentioned three items (i.e., the interaction time per iteration, the computation time per iteration, and the number of interactions) can be reduced simultaneously. However, for some interactive MOEAs which progress towards the DM's ROI with the guidance of a sequence of the DM's preferences, reducing the number of interactions may slow down the population's speed of moving towards the ROI. For example, Deb *et al.* [33] noted that providing preferences more frequently can speed up the optimization of their proposed interactive MOEA. In this case, the tradeoff between the computation time and the number of interactions has to be considered.

C. ISSUES RELATED TO THE DM-MACHINE INTERACTION SYSTEM

1) WHEN SHOULD AN IMO METHOD STOP? (*D-M1*)

The termination of an IMO method is an important issue. Miettinen [3] summarized three main stopping criteria. The first one is when the DM is satisfied with one solution and wants to stop. The second one is when the DM feels tired and does not want to continue. The last one is when a mathematical stopping criterion is met. Vanderpooten and Vincke [125] stated that the interaction procedure should be stopped only if the DM is satisfied with a solution. Korhonen and Wallenius [126] also stated that behavioral convergence is more important than mathematical convergence. Mathematical convergence, however, is not trivial because it contributes to the performance evaluation of IMO methods and can assist the DM to make decisions.

2) HOW TO EVALUATE IMO METHODS? (*D-M2*)

Since the ultimate goal of IMO is to find the DM's MPS, testing with human DMs is essential for verifying the practicability of IMO methods. We can test the degree of satisfaction of the DM on the final solution. Since it may be hard for the DM to give a quantitative value, he/she can be asked to express his/her satisfaction in natural language like ''satisfactory'' and ''dissatisfactory.'' Moreover, the DM's feeling matters because it determines whether an IMO method can be widely accepted and used in reality. It is influenced by the interaction interface, the information shown to the DM, the required preferences, the DM's confidence about the final solution, etc. In some research, IMO methods were compared in experiments involving human DMs [127]–[129]. Obviously, the human involvement in IMO renders the evaluation of IMO methods subjective. The evaluation results may vary with human DMs. Even with the same human DM, the results may change because of the randomness of human DMs. Furthermore, it is difficult to repeat such experiments because of human DMs' randomness and their memories about previous experiments.

As an alternative, a virtual DM can be designed to play the role of a human DM. A common virtual DM is a VF whose optimal solution in the search space serves as the MPS [33], [53]. The distance between the optimal solution and the final solution of an IMO method can be used to evaluate the quality of the final solution. Chen *et al.* [130] proposed a virtual-DM library consisting of four different VFs for comprehensive and fair comparison of IMO methods. As VFs cannot provide preferences like reference points, Ojalehto *et al.* [131] developed an artificial DM specially for testing reference point based IMO methods. Compared with human DMs, virtual DMs are much cheaper and more convenient and they also facilitate repetitive experiments owing to the elimination of biases and fatigue of human DMs [132].

In [133] and [134], two performance metrics were developed to quantitatively evaluate the quality of a set of solutions obtained by prior-preference-based MOEAs using reference points. A user-supplied reference point is used to pre-process the obtained solutions before the performance assessment. Since the two metrics are valid for a pre-specified reference point and the change of preferences are not considered, they are not yet suitable for testing IMO methods.

V. CONCLUSION AND FUTURE RESEARCH

Reliable IMO methods are expected to adapt to different DMs and support them to find their MPSs with low human burdens and small computational cost. As many IMO methods have been developed by MCDM and EMO communities, we have built a comprehensive taxonomy in this paper to identify IMO methods. The taxonomy includes four classification criteria in which each criterion represents one design factor of IMO methods. According to the taxonomy, a review of the state-of-the-art IMO methods from the fields of MCDM and EMO has been provided to assist in understanding the details of IMO methods. Besides, eights important issues worthy of consideration in IMO have been summarized. Among them, four issues are related to the DM, two issues are connected with the machine (algorithm), and the rest two are related to the DM-Machine interaction system. We expect that this paper can promote the development of IMO with the joint effort of both MCDM and EMO communities.

From the review of representative IMO methods and the discussion of the eight crucial issues, it is clear that some IMO-related research directions deserve further investigation. Some of them include the following.

A. HUMAN BEHAVIOR

The DM plays an essential role in IMO. How different DMs' behaviors might influence the final solution of an IMO method is, as yet, not so clear. It is necessary to understand the behaviors of different DMs in order to give a DM suitable support to find his/her MPS. Besides, the research on human behavior is beneficial to solving issues *Dm1* and *Dm2*.

B. FURTHER HYBRIDIZATION OF MCDM AND EMO METHODS

By hybridizing MCDM and EMO methods, their advantages can be synergized. Existing "MCDM in EMO" based IMO methods usually ask the DM to provide reference points or compare solutions. New interactive MOEAs which consider other types of preferences are expected. Compared to "MCDM in EMO" based methods, "EA in MCDM" based IMO methods seem to get less attention. Since many classical IMO methods like STEM and NIMBUS have been successfully applied to solve real-world MOPs [135]–[137], it is of interest to research the possibility of integrating EAs into them to enhance their capability for solving more problems.

C. TRADEOFF BETWEEN EXPLORATION AND EXPLOITATION

Exploration and exploitation in search and optimization have been defined and discussed in [138]. The tradeoff between them influences the computation cost and the convergence quality. In interactive MOEAs, the relationship between the exploration of the search space and the exploitation of the ROI is also worth studying. Emphasizing exploration can provide the DM with more information; however, it may reduce the convergence speed. Emphasizing exploitation may narrow the DM's view of the problem and lead to the premature convergence. Some IMO methods have utilized a parameter to weigh the exploration and exploitation [62], [85]. Initially, the parameter is set as a relatively large number to obtain a wide range of solutions. Later, it is reduced to concentrate the search towards the DM's ROI.

D. HANDLING LINGUISTIC PREFERENCES

Usually, a human DM is more likely to provide qualitative preferences rather than quantitative preferences. In view of this fact, fuzzy logic is an appropriate tool to handle the linguistic preferences of humans. It can convert fuzzy preferences into quantitative information as it does in [37], [104], and [105]. Fuzzy logic or fuzzy inference systems for the purpose of preference extraction and transformation deserve further investigation.

E. INTERACTION SYSTEMS

As stated in the issue *Dm3*, some general interaction systems which allow the DM to select the type of preferences that he/she wants to provide and change it freely at each iteration have been developed. In order to cater to more types of preferences, it is necessary to design new interaction systems which consist of multiple IMO methods with different preference models. These preference models should be capable of modelling different types of preferences.

REFERENCES

[1] K. Deb, "Multi-objective optimization," in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, 2nd ed. New York, NY, USA: Springer, 2014.

[2] K. Miettinen and F. Ruiz, "NAUTILUS framework: Towards trade-off-free interaction in multiobjective optimization," *J. Bus. Econ.*, vol. 86, nos. 1–2, pp. 5–21, 2016.

[3] K. Miettinen, *Nonlinear Multiobjective Optimization*. Norwell, MA, USA: Kluwer, 1999.

[4] C. L. Hwang and A. S. M. Masud, *Multiple Objective Decision Making Methods and Applications*. Berlin, Germany: Springer, 1979.

[5] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2008, pp. 2424–2431.

[6] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 48, no. 1, 2015, Art. no. 13.

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

[8] M. M. Köksalan, J. Wallenius, and S. Zionts, *Multiple Criteria Decision Making: From Early History to the 21st Century*. Singapore: World Scientific, 2011.

[9] C. L. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications: A State-of-the-Art Survey*. Berlin, Germany: Springer, 1981.

[10] R. Benayoun, J. de Montgolfier, J. Tergny, and O. Laritchev, "Linear programming with multiple objective functions: Step method (STEM)," *Math. Program.*, vol. 1, no. 1, pp. 366–375, 1971.

[11] A. M. Geoffrion, J. S. Dyer, and A. Feinberg, "An interactive approach for multi-criterion optimization, with an application to the operation of an academic department," *Manage. Sci.*, vol. 19, no. 4, pp. 357–368, 1972.

[12] V. Chankong and Y. Y. Haimes, "The interactive surrogate worth trade-off (ISWT) method for multiobjective decision-making," in *Multiple Criteria Problem Solving*. Berlin, Germany: Springer, 1978, pp. 42–67.

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

[14] H. Nakayama and Y. Sawaragi, "Satisficing trade-off method for multiobjective programming," in *Interactive Decision Analysis*. Berlin, Germany: Springer, 1984, pp. 113–122.

[15] J. T. Buchanan, "A Naïve approach for solving MCDM problems: The GUESS method," *J. Oper. Res. Soc.*, vol. 48, no. 2, pp. 202–206, 1997.

[16] K. Miettinen and M. M. Mäkelä, "Comparing two versions of NIMBUS optimization system," Dept. Math., Lab. Sci. Comput., Univ. Jyväskylä, Jyväskylä, Finland, Tech. Rep. 23/1996, 1996.

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

[18] C. A. C. Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.

[19] T. Bäck, U. Hammel, and H. P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 3–17, Apr. 1997.

[20] T. Bäck, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. London, U.K.: Oxford Univ. Press, 1996.

[21] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genet. Algorithms*, 1985, pp. 93–100.

[22] A. López-Jaimes and C. A. C. Coello, "Study of preference relations in many-objective optimization," in *Proc. 11th Annu. Conf. Genet. Evol. Comput.*, 2009, pp. 611–618.

[23] C. von Lücken, B. Barán, and C. Brizuela, "A survey on multi-objective evolutionary algorithms for many-objective problems," *Comput. Optim. Appl.*, vol. 58, no. 3, pp. 707–756, 2014.

[24] K. Sindhya, A. B. Ruiz, and K. Miettinen, "A preference based interactive evolutionary algorithm for multi-objective optimization: PIE," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Berlin, Germany: Springer, 2011, pp. 212–225.

[25] K. Miettinen, "Using interactive multiobjective optimization in continuous casting of steel," *Mater. Manuf. Processes*, vol. 22, no. 5, pp. 585–593, 2007.

[26] L. Chen, B. Xin, and J. Chen, "A trade-off based interactive multi-objective optimization method driven by evolutionary algorithms," *J. Adv. Comput. Intell. Intell. Inform.*, vol. 21, no. 2, pp. 284–292, 2017.

[27] R. C. Purshouse, K. Deb, M. M. Mañsor, S. Mostaghim, and R. Wang, "A review of hybrid evolutionary multiple criteria decision making methods," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2014, pp. 1147–1154.
[28] S. Bechikh, M. Kessentini, L. B. Said, and K. Ghédira, "Preference incorporation in evolutionary multiobjective optimization: A survey of the state-of-the-art," in *Advances in Computers*, vol. 98. Amsterdam, The Netherlands: Elsevier, 2015, pp. 141–207.
[29] J. Branke, "MCDA and multiobjective evolutionary algorithms," in *Multiple Criteria Decision Analysis: State of the Art Surveys*, 2nd ed. New York, NY, USA: Springer, 2016, pp. 977–1008.
[30] D. Meignan, S. Knust, J.-M. Frayret, G. Pesant, and N. Gaud, "A review and taxonomy of interactive optimization methods in operations research," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 3, 2015, Art. no. 17.
[31] K. Miettinen, J. Hakanen, and D. Podkopaev, "Interactive nonlinear multiobjective optimization methods," in *Multiple Criteria Decision Analysis: State of the Art Surveys*, 2nd ed. New York, NY, USA: Springer, 2016, pp. 927–976.
[32] F. Kasimoğlu, "A survey on interactive approaches for multi-objective decision making problems," *J. Defense Sci.*, vol. 15, no. 1, pp. 231–255, 2016.
[33] K. Deb, A. Sinha, P. J. Korhonen, and J. Wallenius, "An interactive evolutionary multiobjective optimization method based on progressively approximated value functions," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 723–739, Oct. 2010.
[34] O. I. Larichev, "Cognitive validity in design of decision-aiding techniques," *J. Multi-Criteria Decis. Anal.*, vol. 1, no. 3, pp. 127–138, 1992.
[35] B. Roy and V. Mousseau, "A theoretical framework for analysing the notion of relative importance of criteria," *J. Multi-Criteria Decis. Anal.*, vol. 5, no. 2, pp. 145–159, 1996.
[36] K. Miettinen, "Introduction to multiobjective optimization: Noninteractive approaches," in *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Berlin, Germany: Springer, 2008, pp. 1–26.
[37] X. Shen, Y. Guo, Q. Chen, and W. Hu, "A multi-objective optimization evolutionary algorithm incorporating preference information based on fuzzy logic," *Comput. Optim. Appl.*, vol. 46, no. 1, pp. 159–188, 2010.
[38] M. Luque, K. Miettinen, P. Eskelinen, and F. Ruiz, "Incorporating preference information in interactive reference point methods for multiobjective optimization," *Omega*, vol. 37, no. 2, pp. 450–462, 2009.
[39] S. Greco, B. Matarazzo, and R. Słowiński, "Interactive evolutionary multiobjective optimization using dominance-based rough set approach," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.
[40] S. Greco, V. Mousseau, and R. Słowiński, "Ordinal regression revisited: Multiple criteria ranking using a set of additive value functions," *Eur. J. Oper. Res.*, vol. 191, no. 2, pp. 416–436, 2008.
[41] R. Słowiński, S. Greco, and B. Matarazzo, "Axiomatization of utility, outranking and decision rule preference models for multiple-criteria classification problems under partial inconsistency with the dominance principle," *Control Cybern.*, vol. 31, no. 4, pp. 1005–1035, 2002.
[42] K. Deb, J. Sundar, N. Udaya Bhaskara Rao, 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.
[43] K. Miettinen and M. M. Mäkelä, "On scalarizing functions in multiobjective optimization," *OR Spectr.*, vol. 24, no. 2, pp. 193–213, 2002.
[44] F. Ruiz, M. Luque, F. Miguel, and M. del Mar Muñoz, "An additive achievement scalarizing function for multiobjective programming problems," *Eur. J. Oper. Res.*, vol. 188, no. 3, pp. 683–694, 2008.
[45] M. Luque, K. Miettinen, A. B. Ruiz, and F. Ruiz, "A two-slope achievement scalarizing function for interactive multiobjective optimization," *Comput. Oper. Res.*, vol. 39, no. 7, pp. 1673–1681, 2012.
[46] Y. Nikulin, K. Miettinen, and M. M. Mäkelä, "A new achievement scalarizing function based on parameterization in multiobjective optimization," *OR Spectr.*, vol. 34, no. 1, pp. 69–87, 2012.
[47] K. Deb and A. Kumar, "Interactive evolutionary multi-objective optimization and decision-making using reference direction method," in *Proc. 9th Annu. Conf. Genet. Evol. Comput.*, 2007, pp. 781–788.
[48] A. B. Ruiz, M. Luque, K. Miettinen, and R. Saborido, "An interactive evolutionary multiobjective optimization method: Interactive WASF-GA," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Cham, Switzerland: Springer, 2015, pp. 249–263.
[49] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evol. Comput.*, vol. 17, no. 3, pp. 411–436, 2009.
[50] Y. Siskos, E. Grigoroudis, and N. F. Matsatsinis, "UTA methods," in *Multiple Criteria Decision Analysis: State of the Art Surveys*. New York, NY, USA: Springer, 2005, pp. 297–334.
[51] J. R. Figueira, S. Greco, V. Mousseau, and R. Słowiński, "Interactive multiobjective optimization using a set of additive value functions," in *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Berlin, Germany: Springer, 2008, pp. 97–119.
[52] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz, "Interactive evolutionary multiobjective optimization driven by robust ordinal regression," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 58, no. 3, pp. 347–358, 2010.
[53] J. Branke, S. Greco, R. Słowiński, and P. Zielniewicz, "Learning value functions in interactive evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 1, pp. 88–102, Feb. 2015.
[54] K. Narukawa, Y. Tanigaki, and H. Ishibuchi, "Evolutionary many-objective optimization using preference on hyperplane," in *Proc. 16th Annu. Conf. Genet. Evol. Comput.*, 2014, pp. 91–92.
[55] K. Narukawa, Y. Setoguchi, Y. Tanigaki, M. Olhofer, B. Sendhoff, and H. Ishibuchi, "Preference representation using Gaussian functions on a hyperplane in evolutionary multi-objective optimization," *Soft Comput.*, vol. 20, no. 7, pp. 2733–2757, 2016.
[56] D. S. Todd and P. Sen, "Directed multiple objective search of design spaces using genetic algorithms and neural networks," in *Proc. 1st Annu. Conf. Genet. Evol. Comput.*, 1999, pp. 1738–1743.
[57] R. Battiti and A. Passerini, "Brain–computer evolutionary multiobjective optimization: A genetic algorithm adapting to the decision maker," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 671–687, Oct. 2010.
[58] L. R. Pedro and R. H. C. Takahashi, "INSPM: An interactive evolutionary multi-objective algorithm with preference model," *Inf. Sci.*, vol. 268, pp. 202–219, Jun. 2014.
[59] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Proc. 5th Int. Conf. Genet. Algorithms*, 1993, pp. 416–423.
[60] A. Sinha, P. Korhonen, J. Wallenius, and K. Deb, "An interactive evolutionary multi-objective optimization method based on polyhedral cones," in *Learning and Intelligent Optimization*. Berlin, Germany: Springer, 2010.
[61] J. Molina, L. V. Santana, A. G. Hernández-Díaz, C. A. C. Coello, and R. Caballero, "g-dominance: Reference point based dominance for multiobjective metaheuristics," *Eur. J. Oper. Res.*, vol. 197, no. 2, pp. 685–692, 2009.
[62] L. B. Said, S. Bechikh, and K. Ghédira, "The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 801–818, Oct. 2010.
[63] S. Greco, B. Matarazzo, and R. Słowiński, "Rough sets theory for multicriteria decision analysis," *Eur. J. Oper. Res.*, vol. 129, no. 1, pp. 1–47, 2001.
[64] S. Greco, B. Matarazzo, and R. Słowiński, "Dominance-based rough set approach to interactive multiobjective optimization," *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Berlin, Germany: Springer, 2008, pp. 121–155.
[65] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[66] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
[67] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "An overview of population-based algorithms for multi-objective optimisation," *Int. J. Syst. Sci.*, vol. 46, no. 9, pp. 1572–1599, 2015.
[68] C. Blum and D. Merkle, *Swarm Intelligence: Introduction and Applications*. Berlin, Germany: Springer, 2008.
[69] Y. Tan, Y. Shi, F. Buarque, A. Gelbukh, S. Das, and A. Engelbrecht, *Advances in Swarm and Computational Intelligence*. Cham, Switzerland: Springer, 2015.
[70] M. Reyes-Sierra and C. A. C. Coello, "Multi-objective particle swarm optimizers: State-of-the-art," *Int. J. Comput. Intell. Res.*, vol. 2, no. 3, pp. 287–308, 2006.
[71] I. Alaya, C. Solnon, and K. Ghédira, "Ant colony optimization for multi-objective optimization problems," in *Proc. 19th IEEE Int. Conf. Tools Artif. Intell.*, Oct. 2007, pp. 450–457.
[72] F. Campelo, F. G. Guimarães, and H. Igarashi, "Overview of artificial immune systems for multi-objective optimization," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Berlin, Germany: Springer, 2007, pp. 937–951.
[73] R. Akbari, R. Hedayatzadeh, K. Ziarati, and B. Hassanizadeh, "A multi-objective artificial bee colony algorithm," *Swarm Evol. Comput.*, vol. 2, pp. 39–52, Feb. 2012.

[74] Y.-S. Ong, M. H. Lim, and X. Chen, "Memetic computation—Past, present & future," *IEEE Comput. Intell. Mag.*, vol. 5, no. 2, pp. 24–31, May 2010.

[75] B. Xin, J. Chen, J. Zhang, H. Fang, and Z.-H. Peng, "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: A review and taxonomy," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 5, pp. 744–767, Sep. 2012.

[76] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 204–223, Apr. 2003.

[77] A. Lara, G. Sanchez, C. A. C. Coello, and O. Schutze, "HCS: A new local search strategy for memetic multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 14, no. 1, pp. 112–132, Feb. 2010.

[78] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.

[79] R. Allmendinger, X. Li, and J. Branke, "Reference point-based particle swarm optimization using a steady-state approach," in *Proc. Asia–Pacific Conf. Simulated Evol. Learn.* Berlin, Germany: Springer, 2008, pp. 200–209.

[80] U. K. Wickramasinghe and X. Li, "Integrating user preferences with particle swarms for multi-objective optimization," in *Proc. 10th Annu. Conf. Genet. Evol. Comput.*, 2008, pp. 745–752.

[81] M. Chica, Ó. Cordón, S. Damas, and J. Bautista, "Interactive preferences in multiobjective ant colony optimisation for assembly line balancing," *Soft Comput.*, vol. 19, no. 10, pp. 2891–2903, 2015.

[82] J. Dias, M. E. Captivo, and J. Clímaco, "A memetic algorithm for multiobjective dynamic location problems," *J. Global Optim.*, vol. 42, no. 2, pp. 221–253, 2008.

[83] K. Deb and A. Kumar, "Light beam search based multi-objective optimization using evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 2125–2132.

[84] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Berlin, Germany: Springer, 2004, pp. 832–842.

[85] A. López-Jaimes and C. A. C. Coello, "Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems," *Inf. Sci.*, vol. 277, pp. 1–20, Sep. 2014.

[86] J.-B. Yang, "Gradient projection and local region search for multiobjective optimisation," *Eur. J. Oper. Res.*, vol. 112, no. 2, pp. 432–459, 1999.

[87] M. Luque, J. B. Yang, and B. Y. H. Wong, "PROJECT method for multiobjective optimization based on gradient projection and reference points," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 4, pp. 864–879, Jul. 2009.

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

[89] K. Miettinen, P. Eskelinen, F. Ruiz, and M. Luque, "NAUTILUS method: An interactive technique in multiobjective optimization based on the nadir point," *Eur. J. Oper. Res.*, vol. 206, no. 2, pp. 426–434, 2010.

[90] S. Phelps and M. Köksalan, "An interactive evolutionary metaheuristic for multiobjective combinatorial optimization," *Manage. Sci.*, vol. 49, no. 12, pp. 1726–1738, 2003.

[91] A. Sinha, K. Deb, P. Korhonen, and J. Wallenius, "Progressively interactive evolutionary multi-objective optimization method using generalized polynomial value functions," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.

[92] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "An interactive simple indicator-based evolutionary algorithm (I-SIBEA) for multiobjective optimization problems," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Cham, Switzerland: Springer, 2015, pp. 277–291.

[93] J. W. Fowler, E. S. Gel, M. M. Köksalan, P. Korhonen, J. L. Marquis, and J. Wallenius, "Interactive evolutionary multi-objective optimization for quasi-concave preference functions," *Eur. J. Oper. Res.*, vol. 206, no. 2, pp. 417–425, 2010.

[94] M. Gong, F. Liu, W. Zhang, L. Jiao, and Q. Zhang, "Interactive MOEA/D for multi-objective decision making," in *Proc. 13th Annu. Conf. Genet. Evol. Comput.*, 2011, pp. 721–728.

[95] M. Köksalan and I. Karahan, "An interactive territory defining evolutionary algorithm: iTDEA," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 702–722, Oct. 2010.

[96] R. Wang, R. C. Purshouse, and P. J. Fleming, "'Whatever works best for you'—A new method for a priori and progressive multi-objective optimisation," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.* Berlin, Germany: Springer, 2013, pp. 337–351.

[97] J. Hakanen, T. Chugh, K. Sindhya, Y. Jin, and K. Miettinen, "Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Dec. 2016, pp. 1–8.

[98] P. J. Korhonen and J. Laakso, "A visual interactive method for solving the multiple criteria problem," *Eur. J. Oper. Res.*, vol. 24, no. 2, pp. 277–287, 1986.

[99] P. Korhonen and J. Laakso, "On developing a visual interactive multiple criteria method—An outline," in *Decision Making with Multiple Objectives*. Berlin, Germany: Springer, 1985, pp. 272–281.

[100] A. Jaszkiewicz and R. Słowiński, "The 'light beam search' approach—An overview of methodology applications," *Eur. J. Oper. Res.*, vol. 113, no. 2, pp. 300–314, 1999.

[101] T. Wagner and H. Trautmann, "Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 688–701, Oct. 2010.

[102] Z. Li and H.-L. Liu, "Integrating preference by means of desirability function with evolutionary multi-objective optimization," *Intell. Automat. Soft Comput.*, vol. 21, no. 2, pp. 197–209, 2015.

[103] H. Trautmann, T. Wagner, D. Biermann, and C. Weihs, "Indicator-based selection in evolutionary multiobjective optimization algorithms based on the desirability index," *J. Multi-Criteria Decis. Anal.*, vol. 20, nos. 5–6, pp. 319–337, 2013.

[104] D. Cvetković and I. C. Parmee, "Preferences and their application in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 42–57, Feb. 2002.

[105] Y. Jin and B. Sendhoff, "Incorporation of fuzzy preferences into evolutionary multiobjective optimization," in *Proc. 4th Asia–Pacific Conf. Simulated Evol. Learn.*, 2002, pp. 26–30.

[106] J.-B. Yang and D. Li, "Normal vector identification and interactive tradeoff analysis using minimax formulation in multiobjective optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 32, no. 3, pp. 305–319, May 2002.

[107] K. Miettinen and M. M. Mäkelä, "Interactive multiobjective optimization system WWW-NIMBUS on the Internet," *Comput. Oper. Res.*, vol. 27, nos. 7–8, pp. 709–723, 2000.

[108] J. R. Figueira, S. Greco, and R. Słowiński, "Building a set of additive value functions representing a reference preorder and intensities of preference: GRIP method," *Eur. J. Oper. Res.*, vol. 195, no. 2, pp. 460–486, 2009.

[109] P. Korhonen and J. Karaivanova, "An algorithm for projecting a reference direction onto the nondominated set of given points," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 29, no. 5, pp. 429–435, Sep. 1999.

[110] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474–494, Aug. 2013.

[111] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[112] P. Korhonen and J. Wallenius, "Visualization in the multiple objective decision-making framework," *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Berlin, Germany: Springer, 2008, pp. 195–212.

[113] K. Miettinen, "Survey of methods to visualize alternatives in multiple criteria decision making problems," *OR Spectr.*, vol. 36, no. 1, pp. 3–37, 2014.

[114] A. Sinha, P. Korhonen, J. Wallenius, and K. Deb, "An interactive evolutionary multi-objective optimization algorithm with a limited number of decision maker calls," *Eur. J. Oper. Res.*, vol. 233, no. 3, pp. 674–688, 2014.

[115] J. T. Buchanan and J. Corner, "The effects of anchoring in interactive MCDM solution methods," *Comput. Oper. Res.*, vol. 24, no. 10, pp. 907–918, 1997.

[116] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," *Econ., J. Econ. Soc.*, vol. 47, no. 2, pp. 263–291, 1979.

[117] A. Tversky and D. Kahneman, "Loss aversion in riskless choice: A reference-dependent model," *Quart. J. Econ.*, vol. 106, no. 4, pp. 1039–1061, 1991.

[118] S. Chaudhuri and K. Deb, "An interactive evolutionary multi-objective optimization and decision making procedure," *Appl. Soft Comput.*, vol. 10, no. 2, pp. 496–511, 2010.

[119] M. Luque, F. Ruiz, and K. Miettinen, "Global formulation for interactive multiobjective optimization," *OR Spectr.*, vol. 33, no. 1, pp. 27–48, 2011.

[120] F. Ruiz, M. Luque, and K. Miettinen, "Improving the computational efficiency in a global formulation (GLIDE) for interactive multiobjective optimization," *Ann. Oper. Res.*, vol. 197, no. 1, pp. 47–70, 2012.

[121] M. Ghaznavi, M. Ilati, and E. Khorram, "An interactive algorithm for solving multiobjective optimization problems based on a general scalarization technique," *Iranian J. Numer. Anal. Optim.*, vol. 6, no. 1, pp. 79–101, 2016.

[122] 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.

[123] V. Mousseau, J. Figueira, L. Dias, C. G. da Silva, and J. Clímaco, "Resolving inconsistencies among constraints on the parameters of an MCDA model," *Eur. J. Oper. Res.*, vol. 147, no. 1, pp. 72–93, 2003.

[124] M. Kok, "The interface with decision makers and some experimental results in interactive multiple objective programming methods," *Eur. J. Oper. Res.*, vol. 26, no. 1, pp. 96–107, 1986.

[125] D. Vanderpooten and P. Vincke, "Description and analysis of some representative interactive multicriteria procedures," *Math. Comput. Model.*, vol. 12, nos. 10–11, pp. 1221–1238, 1989.

[126] P. Korhonen and J. Wallenius, "Behavioral issues in MCDM: Neglected research questions," in *Multicriteria Analysis*. Berlin, Germany: Springer, 1997, pp. 412–422.

[127] J. L. Corner and J. T. Buchanan, "Experimental consideration of preference in decision making under certainty," *J. Multi-Criteria Decis. Anal.*, vol. 4, no. 2, pp. 107–121, 1995.

[128] J. L. Corner and J. T. Buchanan, "Capturing decision maker preference: Experimental comparison of decision analysis and MCDM techniques," *Eur. J. Oper. Res.*, vol. 98, no. 1, pp. 85–97, 1997.

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

[130] 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.*, Jun. 2017, pp. 636–641.

[131] V. Ojalehto, D. Podkopaev, and K. Miettinen, "Towards automatic testing of reference point based interactive methods," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Cham, Switzerland: Springer, 2016, pp. 483–492.

[132] V. Belton *et al.*, "Interactive multiobjective optimization from a learning perspective," in *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Berlin, Germany: Springer, 2008, pp. 405–433.

[133] A. Mohammadi, M. N. Omidvar, and X. Li, "A new performance metric for user-preference based multi-objective evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 2825–2832.

[134] K. Li and K. Deb, "Performance assessment for preference-based evolutionary multi-objective optimization using reference points," Comput. Optim. Innov. Lab., Michigan State Univ., East Lansing, MI, USA, Tech. Rep. 2016001, 2016.

[135] R. Rodríguez, M. Luque, and M. González, "Portfolio selection in the Spanish stock market by interactive multiobjective programming," *Top*, vol. 19, no. 1, pp. 213–231, 2011.

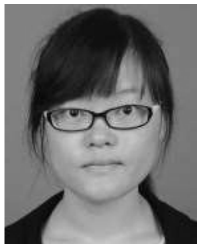
[136] H. Ruotsalainen, K. Miettinen, J.-E. Palmgren, and T. Lahtinen, "Interactive multiobjective optimization for anatomy-based three-dimensional HDR brachytherapy," *Phys. Med. Biol.*, vol. 55, no. 16, pp. 4703–4719, 2010.

[137] S. Taras and A. Woinaroschy, "An interactive multi-objective optimization framework for sustainable design of bioprocesses," *Comput. Chem. Eng.*, vol. 43, pp. 10–22, Aug. 2012.

[138] J. Chen, B. Xin, Z. Peng, L. Dou, and J. Zhang, "Optimal contraction theorem for exploration–exploitation tradeoff in search and optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 3, pp. 680–691, May 2009.

BIN XIN (S'09–M'10) received the B.S. degree in information engineering and the Ph.D. degree in control science and engineering from the Beijing Institute of Technology, Beijing, China, in 2004 and 2012, respectively. He was an Academic Visitor with the Decision and Cognitive Sciences Research Centre, The University of Manchester, from 2011 to 2012.

He is currently an Associate Professor with the School of Automation, Beijing Institute of Technology. His current research interests include search and optimization, evolutionary computation, combinatorial optimization, and multi-agent systems. He is an Associate Editor of the *Journal of Advanced Computational Intelligence and Intelligent Informatics* and the journal *Unmanned Systems*.



LU CHEN received the bachelor's degree from the Beijing Institute of Technology, Beijing, China, in 2014. She is currently pursuing the Ph.D. degree with the School of Automation, Beijing Institute of Technology. She was a Visiting Doctoral Student with the Faculty of Information Technology, University of Jyväskylä, from 2017 to 2018.

Her current research interests include interactive multiobjective optimization and evolutionary multiobjective optimization.

JIE CHEN (M'09–SM'12) received the B.S., M.S., and Ph.D. degrees in control theory and control engineering from the Beijing Institute of Technology, China, in 1986, 1996, and 2001, respectively. From 1989 to 1990, he was a Visiting Scholar with California State University, USA. From 1996 to 1997, he was a Research Fellow with the School of E&E, University of Birmingham, U.K.

He is currently a Professor of control science and engineering with the Beijing Institute of Technology. He is also an Academician of the Chinese Academy of Engineering. He is also the Head of the State Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology.

He has co-authored four books and over 200 research papers. His main research interests include intelligent control and decision in complex systems, multi-agent systems, and optimization methods. He serves as the Managing Editor for the *Journal of Systems Science and Complexity* (2014–2017) and an Associate Editor for the IEEE TRANSACTIONS ON CYBERNETICS (2016–2018) and many other international journals.

HISAO ISHIBUCHI (M'93–SM'10–F'14) received the B.S. and M.S. degrees in precision mechanics from Kyoto University, Kyoto, Japan, in 1985 and 1987, respectively, and the Ph.D. degree in computer science from Osaka Prefecture University, Sakai, Osaka, Japan, in 1992. He was with Osaka Prefecture University from 1987 to 2017. Since 2017, he has been a Chair Professor with the Southern University of Science and Technology, China. His research interests include fuzzy rule-based classifiers, evolutionary multi-objective and many-objective optimization, memetic algorithms, and evolutionary games.

Dr. Ishibuchi was the IEEE Computational Intelligence Society Vice-President for Technical Activities from 2010 to 2013. He is currently an AdCom Member of the IEEE CIS (2014–2019), and the Editor-in-Chief of the *IEEE Computational Intelligence Magazine* (2014–2019).

KAORU HIROTA (M'83–LM'17) received the Dr.E. degree from the Tokyo Institute of Technology in 1979. After his career at the Sagami Institute of Technology from 1979 to 1982 and Hosei University from 1982 to 1995, he had been with the Tokyo Institute of Technology from 1995 to 2015.

He is currently a Professor of the Beijing Institute of Technology, framework of 1000 Global Experts Program, Chinese Government, in 2015. He is also a Professor Emeritus with the Tokyo Institute of Technology and the Director of the Japan Society for the Promotion of Science Beijing Office. He has published over 290 journal papers, 50 books, and 550 conference papers. His research interests include fuzzy systems, intelligent robot, and image understanding. He was an experienced President and fellow of the International Fuzzy Systems Association, and a President of the Japan Society for Fuzzy Theory and Systems. He is a Chief Editor of the *Journal of Advanced Computational Intelligence and Intelligent Informatics*. He organized over 10 international conferences/symposiums as a founding/general/program chair.

BO LIU received the B.S. degree in electrical engineering from Xi'an Jiaotong University, China, in 2001, and the Ph.D. degree in control science and engineering from the Department of Automation, Tsinghua University, China, in 2007.

He was a Post-Doctoral Research Staff on economic system analysis with the Centre for World Food Studies (SOW-VU), Faculty of Economics and Business Administration, Vrije Universiteit, Amsterdam, The Netherlands, from 2007 to 2009. He is currently an Associate Professor with the Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China. His research interests include logistic and scheduling systems, combinatorial optimization, and data mining for healthcare. He is the Technical Committee Member of Task Force on Mimetic Computing in the IEEE Computational Intelligence Society.