Interactive Decomposition Multi-Objective Optimisation via Progressively Learned Value Functions

4 authors, including:

Ke Li
University of Exeter
77 PUBLICATIONS   **3,129** CITATIONS

Dragan Savic
KWR Water Research Institute
557 PUBLICATIONS   **17,049** CITATIONS

Some of the authors of this publication are also working on these related projects:

MSc thesis View project

2015-2017 Flood impact assessment in mega cities under urban sprawl and climate change View project

Interactive Decomposition Multi-Objective Optimisation via Progressively Learned Value Functions*

Ke Li[#1], Renzhi Chen[#2], Dragan Savić[3] and Xin Yao[2,3]

[1]Department of Computer Science, University of Exeter
[2]Department of Computer Science and Engineering, Southern University of Science and Technology
[3]CERCIA, School of Computer Science, University of Birmingham
[4]Department of Engineering, University of Exeter
*Email: {k.li, d.savic}@exeter.ac.uk, rxc332@cs.bham.ac.uk, xiny@sustc.edu.cn
[#]The first two authors make equal contributions to this paper.

Abstract: Decomposition has become an increasingly popular technique for evolutionary multi-objective optimisation (EMO). A decomposition-based EMO algorithm is usually designed to approximate a whole Pareto-optimal front (PF). However, in practice, the decision maker (DM) might only be interested in her/his region of interest (ROI), i.e., a part of the PF. Solutions outside that might be useless or even noisy to the decision-making procedure. Furthermore, there is no guarantee to find the preferred solutions when tackling many-objective problems. This paper develops an interactive framework for the decomposition-based EMO algorithm to lead a DM to the preferred solutions of her/his choice. It consists of three modules, i.e., consultation, preference elicitation and optimisation. Specifically, after every several generations, the DM is asked to score a few candidate solutions in a consultation session. Thereafter, an approximated value function, which models the DM's preference information, is progressively learned from the DM's behavior. In the preference elicitation session, the preference information learned in the consultation module is translated into the form that can be used in a decomposition-based EMO algorithm, i.e., a set of reference points that are biased toward to the ROI. The optimisation module, which can be any decomposition-based EMO algorithm in principle, utilises the biased reference points to direct its search process. Extensive experiments on benchmark problems with three to ten objectives fully demonstrate the effectiveness of our proposed method for finding the DM's preferred solutions.

Keywords: Multi-criterion decision making, interactive multi-objective optimisation, decomposition-based technique, evolutionary computation.

1 Introduction

Multi-objective optimisation problems (MOPs) involve optimising more than one objective function simultaneously. They typically arise in various fields of science (e.g., [1–3]) and engineering (e.g. [4–6]) where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. For example, in portfolio management, maximising the expected value of portfolio returns and minimising the potential risk are two typical conflicting objectives.

Due to the population-based property, evolutionary algorithms (EAs) have been widely recognized as a major approach for multi-objective optimization. Over the last three decades and beyond, much effort has been dedicated to developing evolutionary multi-objective optimization (EMO) algorithms [7–30], such as non-dominated sorting genetic algorithm II (NSGA-II) [31], improved strength Pareto EA (SPEA2) [32] and multi-objective EA based on decomposition (MOEA/D) [33], to find a set of well-converged and well-diversified efficient solutions that approximate the whole Pareto-optimal front (PF). Nevertheless, the ultimate goal of MO is to help the decision maker (DM) find solutions that meet at most her/his preferences. Supplying a DM with a large amount of widely spread trade-off alternatives not only increases her/his workload, but also provides many irrelevant or even noisy

information to the decision-making procedure. Moreover, due to the curse of dimensionality, the performance of EMO algorithms degenerate with the number of objectives. In addition, the number of points to represent a PF grows exponentially with the number of objectives, thereby requiring a large population size to run an EMO algorithm. Besides, there is a severe cognitive obstacle for the DM to comprehend a high-dimensional PF.

To alleviate the above problems associated with the *a posteriori* decision-making procedure in the traditional EMO, it is more practical to incorporate the DM's preference information into the search process. This allows the computational efforts to concentrate on the region of interest (ROI) and thus has a better approximation therein. In general, the preference information can be incorporated *a priori* or *interactively*. If the preference information (in the form of one or more reference points, reference directions or light beams) is elicited *a priori*, it is used to guide the population toward the ROI. For example, the cone-domination based EMO [34], biased niching based EMO [35,36], reference point based EMO [37], the reference direction based EMO [38] and the light beam based EMO [39] are attempts along this direction. Moreover, in [40], Greenwood *et al.* derived a linear value function from a given ranking of a few alternatives to model the DM's preference information. Thereafter, this linear value function is used as the fitness function in an EMO algorithm to guide the population toward the ROI. Note that, in the *a priori* approach, the DM only interacts at the beginning of an EMO run. However, it is non-trivial to faithfully represent the preference information before solving the MOP at hand.

In practice, eliciting the preference information in an interactive manner, which has been studied in the multi-criterion decision-making (MCDM) field for over half a century, seems to be more interesting. This enables the DM to progressively learn and understand the characteristics of the MOP at hand and adjust their elicited preference information. Consequently, solutions are effectively driven toward the ROI. In principle, the above mentioned *a priori* EMO approaches can also be used in an interactive EMO approach in an iterative manner (e.g., [38] and [39]). Specifically, in the first round, the DM can elicit certain preference information (in the form of reference points, reference directions or other means), and it is used in an EMO algorithm to find a set of preferred Pareto-optimal solutions. Thereafter, a few representative solutions can be shown to the DM. If these solutions are satisfied by the DM, they will be used as the outputs and the iterative procedure terminates. Otherwise, the DM will adjust her/his preference information accordingly and it will be used in another EMO run. Alternatively, the DM can be involved to periodically provide preference information as the EMO iterations are underway [41]. In particular, the preference information is progressively learned as value functions with the evolution of solutions. Since the DM gets more frequent chance to provide new information, the overall process is more DM-oriented. Moreover, the DM may feel more in charge and more involved in the overall *optimization-cum-decision-making* process.

During recent years, especially after the developments of MOEA/D and NSGA-III [42], the decomposition-based EMO methods have become increasingly popular for the *a posteriori* MO. Generally speaking, by specifying a set of reference points[1], the decomposition-based EMO methods at first decompose the MOP at hand into multiple subproblems, either with scalar objective or simplified multi-objective. Then, a population-based technique is applied to solve these subproblems in a collaborative manner. Under some mild conditions, the optimal solutions of all subproblems constitute a good approximation to the PF. It is not difficult to understand that the distribution of the reference points is essential for a decomposition-based EMO method. It not only implies *a priori* assumption of the PF's geometrical characteristics, but also determines the distribution of Pareto-optimal solutions. There have been some studies on how to generate desired reference points. For example, [43] suggested a structured method to generate evenly distributed reference points on a canonical simplex. To adapt to the irregular PFs, such as disconnected or mixed shapes and disparately scaled objectives, some adaptive reference point adjustment methods (e.g., [44] and [45]) have been developed to progressively adjust the distribution of reference points on the fly. To integrate the DM's preference information into the decomposition-based EMO methods, a natural idea is to make the distribution of the reference points be biased toward the ROI. Although it sounds intuitive, in practice, how to obtain the

---

[1]In this paper, we use the term reference point without loss of generality, although some other papers, e.g, the original MOEA/D [33], also use the term weight vector interchangeably.

appropriate reference points that accommodate to the DM's preference information is far from trivial. Most recently, there have been some limited initiatives on adjusting the distribution of the reference points according to the DM's preference information (e.g., [46] and [47]). However, most, if not all, of them specify the DM's preference information in *a priori* manner.

This paper develops a simple interactive framework for the decomposition-based EMO algorithms that can progressively learn an approximated value function (AVF) from the DM's behaviour and thus guide the population toward the ROI. This framework consists of three interdependent modules: optimisation, consultation and preference elicitation. Specifically,

- The optimisation module can be any decomposition-based EMO algorithm in principle. It uses the preference information elicited from the preference elicitation module to find the preferred solutions. Periodically, it supplies the consultation module with a few candidates for learning an AVF.

- The consultation module is the interface by which the DM interacts with the optimisation module. It simulates the DM that assigns a score to each candidate supplied by the optimisation module. Then, by using the scored candidates found so far as the training data, a machine learning algorithm is applied to find an AVF that models the DM's preference information.

- The preference elicitation module aims at translating the preference information learned from the consultation module in the form that can be used in a decomposition-based EMO algorithm. In particular, it changes the distribution of reference points to be focused in the ROI.

In empirical studies, our proposed interactive framework is embedded in two widely used decomposition-based EMO algorithms, i.e., MOEA/D and NSGA-III. Their effectiveness for finding preferred Pareto-optimal solutions are validated on several benchmark problems with three to ten objectives.

The rest of this paper is organised as follows. Section 2 provides some preliminaries of this paper. Section 3 describes the technical details of the interactive framework step by step. Afterwards, in Section 4 and 5, the effectiveness of the proposed method is empirically investigated on various benchmark problems with three to ten objectives. Section 6 concludes this paper and provides some future directions.

2 Preliminaries

In this section, we first provide some basic definitions of MO. Then, to facilitate the descriptions of our proposed interactive framework for the decomposition-based EMO algorithms, we start from describing the working mechanisms of two widely used decomposition-based EMO algorithms, i.e., MOEA/D and NSGA-III. At the end, we briefly overview the past studies of interactive MO.

2.1 Basic Definitions

The MOP considered in this paper is formulated as:

$$
\begin{aligned}
\text{minimize} \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\
\text{subject to} \quad & \mathbf{x} \in \Omega
\end{aligned}, \tag{1}
$$

where $\mathbf{x} = (x_1, \cdots, x_n)^T$ is a n-dimensional decision vector and $\mathbf{F}(\mathbf{x})$ is an m-dimensional objective vector. Ω is the feasible set in the decision space $\mathbb{R}^n$ and $\mathbf{F} : \Omega \to \mathbb{R}^m$ is the corresponding attainable set in the objective space $\mathbb{R}^m$. Without considering the DM's preference information, given two solutions $\mathbf{x}^1, \mathbf{x}^2 \in \Omega$, $\mathbf{x}^1$ is said to dominate $\mathbf{x}^2$ if and only if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for all $i \in \{1, \cdots, m\}$ and $\mathbf{F}(\mathbf{x}^1) \neq \mathbf{F}(\mathbf{x}^2)$. A solution $\mathbf{x} \in \Omega$ is said to be Pareto-optimal if and only if there is no solution $\mathbf{x}' \in \Omega$ that dominates it. The set of all Pareto-optimal solutions is called the Pareto-optimal set (PS) and their corresponding objective vectors form the PF. Accordingly, the ideal point is defined as $\mathbf{z}^* = (z_1^*, \cdots, z_m^*)^T$, where $z_i^* = \min_{\mathbf{x} \in PS} f_i(\mathbf{x})$, and the nadir point is defined as $\mathbf{z}^{nd} = (z_1^{nd}, \cdots, z_m^{nd})^T$, where $z_i^{nd} = \max_{\mathbf{x} \in PS} f_i(\mathbf{x})$, $\forall i \in \{1, \cdots, m\}$.

2.2 Decomposition-based EMO algorithms

2.2.1 MOEA/D

The basic idea of MOEA/D is to decompose the original MOP into several subproblems and it uses a population-based technique to solve these subproblems in a collaborative manner. In particular, with respect to a reference point $\mathbf{w}$, this paper uses the Tchebycheff function [13,48] as a subproblem which is defined as:

$$\text{minimize } g(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) = \max_{1 \leq i \leq m} |f_i(\mathbf{x}) - z_i^*|/w_i$$

$$\text{subject to} \quad \mathbf{x} \in \Omega \tag{2}$$

where $\mathbf{z}^*$ is the ideal point. The general working mechanisms of MOEA/D is given as the following three-step process.

Step 1: Initialize a population of solutions $P := \{\mathbf{x}^i\}_{i=1}^N$, a set of reference points $W := \{\mathbf{w}^i\}_{i=1}^N$ and their neighbourhood structure. Randomly assign each solution to a reference point.

Step 2: For $i = 1, \cdots, N$, do

Step 2.1: Randomly selects a required number of mating parents from $\mathbf{w}^i$'s neighbourhood.

Step 2.2: Use crossover and mutation to reproduce offspring $\mathbf{x}^c$.

Step 2.3: Update the subproblems within the neighbourhood of $\mathbf{w}^i$ by $\mathbf{x}^c$.

Step 3: If the stopping criteria is met, then stop and output the population. Otherwise, go to Step 2.

We would like to make some remarks on some important ingredients of the above MOEA/D procedure.

- In Step 1, we use the classic method developed by Das and Dennis [43] to initialise a set of evenly distributed reference points from a canonical simplex. Furthermore, the neighbourhood structure $B(i)$ of each reference point $\mathbf{w}^i$, $i \in \{1, \cdots, N\}$, contains its T closest reference points, where $T = 20$ as suggested in [49]. Fig. 1 gives two examples of reference point distribution and the neighbourhood of a reference point in the two- and three-objective cases.
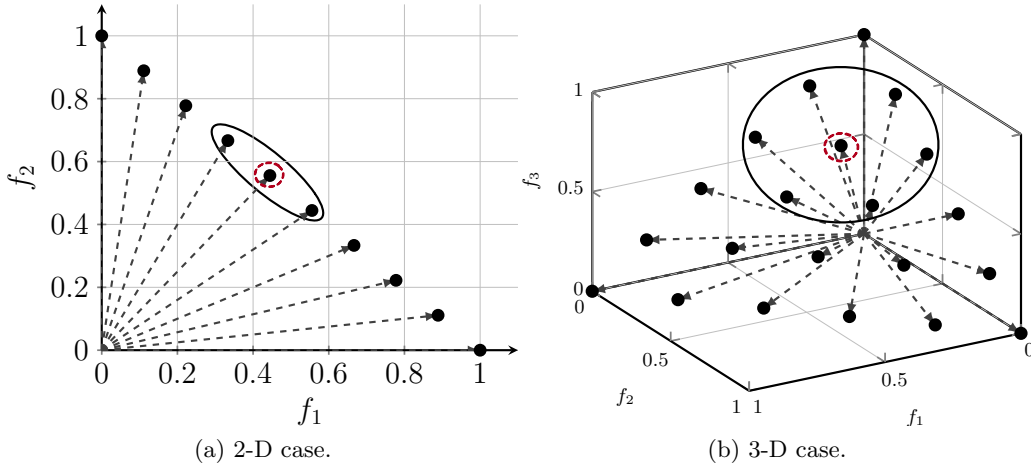

(a) 2-D case. (b) 3-D case.

Figure 1: Illustration of reference points generated by the Das and Dennis' method [43]. The black circle represents the neighbourhood of a particular reference point.

- In Step 2.1, to improve the exploration ability, there is a small probability $\delta = 0.1$ to select mating parents from the whole population as suggested in [49].

- In Step 2.3, $\mathbf{x}^c$ can update a particular reference point $\mathbf{w}$ if and only if $g(\mathbf{x}^c|\mathbf{w}, \mathbf{z}^*) < g(\mathbf{x}|\mathbf{w}, \mathbf{z}^*)$, where $\mathbf{x}$ is the solution originally associated with $\mathbf{w}$.

- In Step 2.3, $\mathbf{x}^c$ also has a small probability $\delta = 0.1$ to update a subproblem from W, rather than merely in $B(i)$.

2.2.2 NSGA-III

This is an extension of NSGA-II for handling many-objective optimisation problems. The subproblem in NSGA-III is to optimise the local crowdedness of the region associated with its corresponding reference point. In particular, it replaces the crowding distance with a reference point based density estimation. The general working mechanisms of NSGA-III is given as follows.

Step 1: Initialise a population of solutions $P := \{\mathbf{x}^i\}_{i=1}^N$, a set of reference points $W := \{\mathbf{w}^i\}_{i=1}^N$.

Step 2: Use crossover and mutation to generate a population of offspring Q.

Step 3: Use non-dominated sorting [31] to divide $R := P \bigcup Q$ into several non-domination fronts $F_1, F_2, \cdots$.

Step 4: Starting from F_1, solutions are stored in a temporary archive $\overline{P}$ till its size for the first time equals or exceeds N, where $\overline{P} := \bigcup_{i=1}^{\ell} F_i$. In particular, F_ℓ is the last acceptable non-domination front. If the size of $\overline{P}$ equals N, then let $P := \overline{P}$ and go to Step 7; otherwise go to Step 5.

Step 5: Let $P := \bigcup_{i=1}^{\ell-1} F_i$. Associate each member of F_ℓ with its closest reference point.

Step 6: A randomly chosen solution in the least crowded reference point is added into P. This process iterates till the size of P equals N.

Step 7: If the stopping criteria is met, then stop and output P. Otherwise, go to Step 2.

We would like to make some remarks on some important ingredients of the above NSGA-III procedure.

- In Step 5, the association of a solution with a reference point is according to the shortest perpendicular distance between this solution and the reference line, starting from the origin and passing through the corresponding reference point. Fig. 2 gives a simple illustration of association in a two-dimensional scenario.
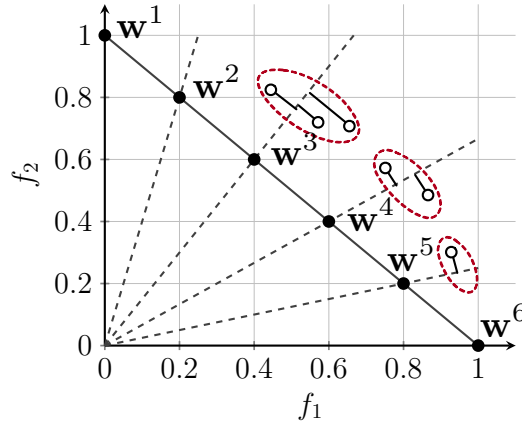

Figure 2: Illustration of solution association of NSGA-III.

- In Step 6, the crowdedness of a reference point is counted as the number of solutions associated with it. For example, as shown in Fig. 2, the crowdedness of $\mathbf{w}^3$ is 3. Note that the crowdedness information is updated after choosing a solution from F_ℓ and add it to P.

2.3 Past Studies on Progressively Interactive Methods

As mentioned in Section 1, there have been a plethora of studies to approximate DM's preferred solutions *a priori*, *posteriori* or *interactively*. Since this paper mainly investigates the frequent involvements of a DM with an EMO algorithm, we do not intend to review *a priori* and *posteriori* approaches, except to encourage the interested readers to look at some recent survey papers [50–52].

Some recent studies periodically asked the DM to provide her/his preference information upon one or more pairs of alternative points found by an EMO algorithm. The information is then used to derive a value function that represents the DM's preference information. For example, Phelps and Köksalan [41] proposed an interactive EA that progressively constructs a linear value function, which is a weighted sum of objectives, by periodically asking the DM to rank pairs of solutions. Thereafter, the resulting value function is then used as the selection criteria of an EA to rank solutions. However, due to the use of a linear model, it might not be effective when the DM's "golden" value function is nonlinear. In [53], Fowler *et al.* proposed to use convex preference cones to model the DM's preference information. In their developed interactive EMO algorithm, such cones are used to partially rank the population members and thus facilitate the fitness assignment. Instead of merely using a single value function, Jaszkiewicz [54] proposed to use a set of linear value functions, each of which is a weighted sum of objectives, chosen from several randomly generated value functions to represent DM's preference information. Due to the use linear value function, it remains the same hallmark as [41] in handling nonlinear problems, especially when the DM's preferred solutions lie on a non-convex part of the PF.

In [55], Deb *et al.* developed a polynomial value function model that is expected to be useful for both linear and nonlinear problems. Specifically, to obtain the preference information, the DM is asked to rank a set of well distributed candidates periodically. Based on this order information, a polynomial value function model is fitted by solving a computationally intensive sequential quadratic programming procedure. Once a most discriminating value function has been identified, it is used to modify the Pareto dominance principle in NSGA-II in order to emphasise the reproduction and survival of preferred solutions. Moreover, the polynomial value function is also used to determine whether the overall optimisation procedure should be terminated or not by performing a local search procedure.

In [56], Battiti and Passerini developed an interactive EMO algorithm that uses the support vector machine (SVM) for ranking [57] to represent complex value functions. Specifically, the DM is asked to rank (at least partially) some selected alternatives during the interaction session. This ranking information is then used to train a SVM, and the derived value function is used to replace the crowding distance in NSGA-II. The empirical results suggested that the training of a SVM requires a relatively large number of solutions, whereas a small number of interactions seem to be sufficient to approximate the DM's golden value function.

In [58] and [59], Branke *et al.* proposed an interactive EMO algorithm by ordinal regression which is able to build preference models compatible with preference information from holistic comparisons of solutions. During the interaction session, the DM is asked to rank a single pair of solutions. This information is used to update the additive value function model that is used in subsequent generations to rank incomparable solutions in terms of the Pareto dominance principle.

In [60], Korhonen *et al.* developed an interactive MO algorithm that progressively learns the DM's preference information by asking the DM to make a set of binary comparisons among several solutions. Specifically, a class of value functions are identified by solving a linear programming problem upon the preference information obtained from the interaction. In particular, they considered three classes of value functions, i.e., linear, quasi-concave and no pre-assumed forms. Based on this classification, they defined a dominance structure and determined the expected probabilities of finding new and better solutions either by search or choosing from several samples. Note that this algorithm terminates if the probability of finding better solutions is low and thus just outputs the currently found most preferred solution. As an extension, [61] developed a sampling-based method to calculate the expected probabilities of finding better solutions.

3 Proposed Method

As shown in Fig. 3, our proposed interactive framework consists of three interdependent modules: consultation, preference elicitation and optimisation. In principle, the optimisation module can be any decomposition-based EMO algorithm. It uses the preference information provided by the preference elicitation module to find the DM's preferred solutions. In addition, it periodically supplies the

consultation module with a few incumbent candidates for scoring. The consultation module is the interface by which the DM interacts with the optimisation procedure. It progressively learns an AVF, which represents the DM's preference information, from the DM's behaviour. The preference elicitation module translates the preference information, learned from the consultation module, into the form that can be used in the optimisation module. In the following paragraphs, we will introduce the technical details of each module step by step.
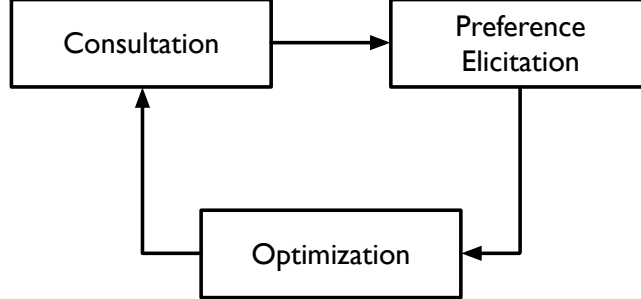

Figure 3: Flowchart of the interactive framework.

3.1 Consultation Module

The consultation module is the interface where the DM interacts with, and expresses her/his preference information to the optimisation module. In principle, there are various ways to represent the DM's preference information. In this paper, we assume that the DM's preference information is represented as a value function. It assigns a solution a score that represents its desirability to the DM. The consultation module mainly aims to progressively learn an AVF that approximates the DM's golden value function, which is unknown *a priori*, by asking the DM to score a few incumbent candidates. We argue that it is labor-intensive to consult the DM every generation. Furthermore, as discussed in [56], consulting the DM at the early stage of the evolution might be detrimental to the decision-making procedure, since the DM can hardly make a reasonable judgement on poorly converged solutions. In this paper, we fix the number of consultations. Before the first consultation session, the EMO algorithm runs as usual without considering any DM's preference information. Afterwards, the consultation session happens every $\tau > 1$ generations.

To approximate the DM's preference information, we need to address two major questions: 1) which solutions can be used for scoring? and 2) how to learn an appropriate AVF?

3.1.1 Scoring

A naïve strategy is to ask the DM to score all solutions in a population. In this case, the search is completely driven by the DM. This obviously increases her/his cognitive load thus has a high risk to cause her/his fatigue. Instead, during each consultation session, we only ask the DM to score a limited number (say $1 \leq \mu \ll N$) of incumbent candidates chosen from the current population.

If it is at the first consultation session, we first initialise another μ "seed" reference points, which can either be generated by the Das and Dennis' method [43] or chosen from the reference points initialised in the optimisation module. Afterwards, for each of these seed reference points, we find the nearest neighbour from the reference points initialised in the optimisation module. Then, the solutions associated with these selected reference points are used as the initial incumbent candidates. Otherwise, at the latter consultation sessions, we use the AVF learned from the last consultation session to score the current population. The μ solutions having the best AVF values are deemed as the ones that are satisfied by the DM most. Accordingly, these μ solutions are used as the incumbent candidates.

3.1.2 Learning

In principle, many off-the-shelf machine learning algorithms can be used to learn the AVF. In this paper, we treat it as a regression problem and use the Radius Basis Function network (RBFN) [62] to serve this purpose. In particular, RBFN, a single-layer feedforward neural network, is easy to train and its performance is relatively insensitive to the increase of the dimensionality. The idea of using RBFN as an approximation function was first proposed by Hardy [63] to fit irregular topological data.

Let $\mathcal{D} = \{(\mathbf{F}(\mathbf{x}^i), \psi(\mathbf{x}^i))\}_{i=1}^M$ denote the dataset for training the RBFN. The objective values of a solution $\mathbf{x}^i$ are the inputs and its corresponding value function $\psi(\mathbf{x}^i)$ scored by the DM is the output. In particular, we accumulate every μ solutions scored by the DM to form $\mathcal{D}$. An RBFN is a real-valued function $\Phi : \mathbb{R}^m \to \mathbb{R}$. Various RBFs can be used as the activation function of the RBFN, such as Gaussian, splines and multi-quadrics. In this paper, we consider the following Gaussian function:

$$\varphi = \exp(-\frac{\|\mathbf{F}(\mathbf{x}) - \mathbf{c}\|}{\sigma^2}), \tag{3}$$

where $\sigma > 0$ is the width of the Gaussian function. Accordingly, the AVF can be calculated as:

$$\Phi(\mathbf{x}) = \omega^0 + \sum_{i=1}^{\text{NR}} \omega^i \exp(-\frac{\|\mathbf{F}(\mathbf{x}) - \mathbf{c}^i\|}{\sigma^2}), \tag{4}$$

where NR is the number of RBFs, each of which is associated with a different center $\mathbf{c}^i$, $i \in \{1, \cdots, \text{NR}\}$. ω^i is the network coefficient, and ω^0 is a bias term, which can be set to the mean of the training data or 0 for simplicity. In our experiment, we use the RBFN program `newrb` provided by the Neural Network Toolbox from the MATLAB[2].

3.2 Preference Elicitation Module

As introduced in Section 2.2, the decomposition-based EMO algorithm is originally designed to use a set of evenly distributed reference points $W = \{\mathbf{w}^i\}_{i=1}^N$ to approximate the whole PF. When considering the DM's preference information, the ROI becomes a partial region of the PF. A natural idea, which translates the DM's preference information into the form that can be used in a decomposition-based EMO algorithm, is to adjust the distribution of reference points. Specifically, the preference elicitation module uses the following four-step process to achieve this purpose.

Step 1: Use $\Phi(\mathbf{x})$ learned in the consultation module to score each member of the current population P.

Step 2: Rank the population according to the scores assigned in Step 1, and find the top μ solutions. Reference points associated with these solutions are deemed as the promising ones, and store them in a temporary archive $W^U := \{\mathbf{w}^{Ui}\}_{i=1}^{\mu'}$.

Step 3: For $i = 1$ to μ' do

Step 3.1: If $\Phi(\mathbf{x}^{Ui}) < g(\mathbf{x}^{best}|\mathbf{w}^{best}, \mathbf{z}^*)$, then go to Step 3.2. Otherwise, move the remaining reference points toward $\mathbf{w}^{best}$ as follows:

$$w_j = w_j + \eta \times (w_j^{best} - w_j), \tag{5}$$

where $j \in \{1, \cdots, m\}$. Terminate the for-loop and go to Step 4.

Step 3.2: Find the $\lceil \frac{N-\mu'}{\mu'} \rceil$ closest reference points to $\mathbf{w}^{Ui}$ according to their Euclidean distances.

Step 3.3: Move each of these reference points toward $\mathbf{w}^{Ui}$ according to equation (5), where $\mathbf{w}^{best}$ is replaced with $\mathbf{w}^{Ui}$.

Step 3.4: Temporarily remove these reference points from W and go to Step 3.

---

[2]https://uk.mathworks.com/help/nnet/ug/radial-basis-neural-networks.html

Step 4: Output the adjusted reference points as the new W.

We would like to make some remarks on some important ingredients of the above process.

- In Step 1, the score of a solution, evaluated by the AVF learned in the consultation module, indicates its satisfaction with respect to the DM's preference information.

- In the decomposition-based EMO algorithm, each solution should be associated with a reference point. Therefore, in Step 2, the rank of a solution indicates the importance of its associated reference point with respect to the DM's preference information. The reference points stored in W^U are indexed according to the ranks of their associated solutions. In other words, $\mathbf{w}^{U1}$ represents the most important reference point, and so on.

- Furthermore, since a reference point might be associated with more than one solution (e.g., in NSGA-III), the number of promising reference points μ' might be smaller than μ, i.e., $1 \leq \mu' \leq \mu$.

- Step 3 is the main crux to adjust the distribution of reference points. The major purpose of this process is to move the other reference points toward those μ' promising ones. In particular, each of these promising reference points attracts around $\lceil \frac{N-\mu'}{\mu'} \rceil$ companions.

- In Step 3.1, $\mathbf{x}^{Ui}$ represents the solution associated with $\mathbf{w}^{Ui}$. $\mathbf{x}^{best}$ represents the best solution evaluated by the DM at the last consultation session, while $\mathbf{w}^{best}$ represents its associated reference point. The major purpose of Step 3.1 is to alleviate the risk of moving the reference points to a wrongly predicted promising one.

- In equation (5), the step size η controls the convergence rate toward the promising reference point.

- Step 3.2 is similar to a clustering process, while we give the reference point, which has a higher rank, a higher priority to attract its companions.

To have a better understanding of this preference elicitation process, Fig. 4 gives an example in a two-objective case. In particular, three promising reference points are highlighted by red circles. $\mathbf{w}^{U1}$ has the highest priority to attract its companions, and so on.
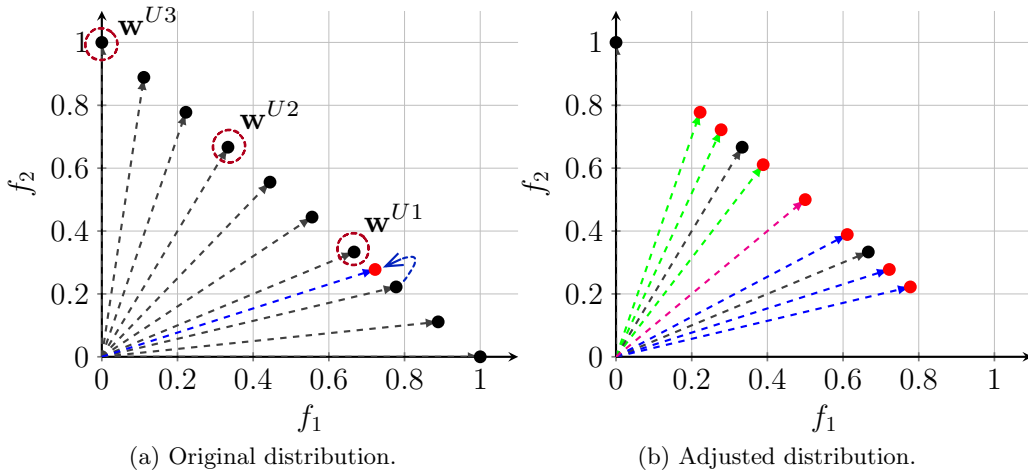

(a) Original distribution. (b) Adjusted distribution.

Figure 4: Illustration of the preference elicitation process.

3.3 Optimization Module

The optimisation module is the search engine that progressively finds the DM's preferred solutions. In principle, any decomposition-based EMO algorithm can be used to serve this purpose. For the

proof of principle purpose, this paper chooses MOEA/D and NSGA-III as the baseline algorithms, whose working mechanisms have been introduced in Section 2.2. Note that MOEA/D and NSGA-III can be used in a plug-in manner without any modification except the reference points. In particular, the reference points used in MOEA/D and NSGA-III need to be adjusted by the preference elicitation module after every consultation session. As for the offspring reproduction, we use the popular simulated binary crossover (SBX) [64] and polynomial mutation [65] for the proof of principle purpose.

4 Experimental Settings

To validate the effectiveness of our proposed interactive framework, we test the performance on benchmark problems with three to ten objectives. The interactive framework is embedded in MOEA/D and NSGA-III, and is respectively denoted as I-MOEA/D-PLVF and I-NSGA-III-PLVF. The widely used DTLZ [66] test problems are chosen to form the benchmark suite. Note that the DTLZ test problems are scalable to any number of objectives. Their formal definitions are described in Section I of the supplementary document.

The parameter settings of our proposed interactive framework are summarized as follows:

- number of incumbent candidates presented to the DM for scoring: $\mu = 2m + 1$ at the first consultation session and $\mu = 10$ afterwards;

- number of generations between two consecutive consultation sessions: $\tau = 25$;

- number of reference points and population size settings are given in Table 1 as suggested in [18];

- number of function evaluations (FEs) is given in Table 2 as suggested in [18].

- step size of the reference point update used in equation (5): $\eta = 0.5$;

- crossover probability and the distribution index for the SBX operator: $p_c = 1.0$ and $\eta_c = 30$;

- mutation probability and the distribution index for the polynomial mutation operator: $p_m = 0.9$ and $\eta_m = 20$;

Table 1: Number of reference points and population size.

m	♯ of reference points	I-NSGA-III-PLVF	I-MOEA/D-PLVF
3	91	92	91
5	210	212	210
8	156	156	156
10	275	276	275

Table 2: Number of FEs for DTLZ test problems.

Test instance	$m = 3$	$m = 5$	$m = 8$	$m = 10$
DTLZ1	400	600	750	1,000
DTLZ2	250	350	500	750
DTLZ3	1,000	1,000	1,000	1,500
DTLZ4	600	1,000	1,250	2,000

Each cell only gives the number of generations. The corresponding number of FEs is each tuple times the corresponding population size of I-NSGA-III-PLVF as shown in Table 1.

As mentioned in [59], the empirical comparison of interactive EMO methods is tricky since a model of the DM's behavior is required yet unfortunately sophisticated to represent. In this paper, we use a pre-specified golden value function, which is unknown to an interactive EMO algorithm, to play as an artificial DM. Specifically, the DM is assumed to minimize the following nonlinear Tchebycheff function:

$$\psi(\mathbf{x}) = \max_{1 \leq i \leq m} |f_i(\mathbf{x}) - \mathbf{z}^*|/w_i^*, \qquad (6)$$

where $\mathbf{z}^*$ is set to be the origin in our experiments, and $\mathbf{w}^*$ is the utopia weights that represents the DM's emphasis on different objectives. We consider two types of $\mathbf{w}^*$: one targets the preferred

solution on the middle region of the PF while the other targets the preferred solution on one side of the PF, i.e., biased toward a particular extreme. Since a m-objective problem has m extremes, there are m different choices for setting the biased $\mathbf{w}^*$. In our experiments, we randomly choose one for the proof-of-principle study. Since the Tchebycheff function is used as the value function and the analytical forms of the test problems are known, we can use the method suggested in [18] to find the corresponding Pareto-optimal solution (also known as the DM's "golden" point) with respect to the given $\mathbf{w}^*$. Detailed settings of $\mathbf{w}^*$ and the corresponding DM's golden point are given in Section II of the supplementary document.

To evaluate the performance of an interactive EMO algorithm for approximating the ROI, we consider using the approximation error of the obtained population P with respect to the DM's golden point $\mathbf{z}^r$ as the performance metric. Specifically, it is calculated as:

$$\mathbb{E}(P) = \min_{\mathbf{x} \in P} \mathsf{dist}(\mathbf{x}, \mathbf{z}^r) \tag{7}$$

where $\mathsf{dist}(\mathbf{x}, \mathbf{z}^r)$ is the Euclidean distance between $\mathbf{z}^r$ and a solution $\mathbf{x} \in P$ in the objective space.

To demonstrate the importance of using the DM's preference information, we also compare I-MOEA/D-PLVF and I-NSGA-III-PLVF with their corresponding baseline algorithms without considering the DM's preference information. In our experiments, we run each algorithm independently 21 times with different random seeds. In the corresponding table, we show the results in terms of the median and the interquartile range (IQR) of the approximation errors obtained by different algorithms. To have a statistical sound comparison, we use the Wilcoxon signed-rank test with a 95% confidence level to validate the significance of the better results.

5 Empirical Results

Our experiments are divided into three parts. First, we validate the effectiveness of our proposed interactive framework for finding the DM's preferred solution. Then, we empirically investigate the influence of the parameters associated with the interactive framework. At last, we investigate a scenario with random noises in the decision-making procedure.

5.1 Performance Comparisons on DTLZ Test Problems

Table 3: Performance comparisons of the approximation errors (median and the corresponding IQR) obtained by I-NSGA-III-PLVF and I-MOEA/D-PLVF versus their baseline MOEA/D and NSGA-III on DTLZ test problems.

		DTLZ1		DTLZ2		DTLZ3		DTLZ4	
m	ROI	I-MOEA/D-PLVF	MOEA/D	I-MOEA/D-PLVF	MOEA/D	I-MOEA/D-PLVF	MOEA/D	I-MOEA/D-PLVF	M
3	c	**0.00042(2.87E-3)**	0.03104(3.18E-3)	**0.01026(1.78E-2)**	0.10300(6.35E-3)	**0.00072(7.26E-3)**	0.10553(1.59E-3)	**0.01302(2.78E-2)**	0.10
	b	**0.00147(2.87E-3)**	0.03103(3.30E-3)	**0.00883(1.09E-2)**	0.09103(2.56E-3)	**0.00281(1.09E-2)**	0.08678(7.75E-3)	**0.00763(8.76E-3)**	0.09
5	c	**0.00417(1.73E-2)**	0.05262(1.90E-2)	**0.01721(2.86E-2)**	0.24170(1.90E-2)	**0.01128(8.77E-2)**	0.24420(4.62E-2)	**0.02762(5.74E-2)**	0.25
	b	**0.01082(2.09E-2)**	0.07648(1.65E-2)	**0.05082(4.73E-2)**	0.20491(1.45E-2)	**0.01792(1.53E-1)**	0.21623(2.35E-2)	**0.03716(6.28E-2)**	0.21
8	c	**0.00213(1.71E-2)**	0.01484(2.21E-3)	**0.01625(1.79E-1)**	0.26152(1.52E-2)	**0.06821(2.78E-1)**	0.42766(9.56E-3)	**0.06538(8.62E-2)**	0.72
	b	**0.01012(1.03E-1)**	0.05534(1.12E-2)	**0.04184(1.10E-1)**	0.12504(1.15E-2)	**0.08697(1.63E-1)**	0.15739(1.32E-2)	**0.12708(1.86E-1)**	0.21
10	c	**0.12690(2.71E-1)**	0.17885(1.10E-3)	**0.10871(1.62E-1)**	0.73855(8.54E-2)	**0.21682(5.71E-1)**	0.73645(2.81E-2)	**0.19273(2.63E-1)**	0.86
	b	**0.15428(1.77E-1)**	0.26343(5.05E-3)	**0.11829(2.08E-1)**	0.25957(2.88E-2)	**0.16287(2.55E-1)**	0.33443(6.99E-2)	**0.10175(3.28E-1)**	0.20
m	ROI	I-NSGA-III-PLVF	NSGA-III	I-NSGA-III-PLVF	NSGA-III	I-NSGA-III-PLVF	NSGA-III	I-NSGA-III-PLVF	N
3	c	**0.00027(7.10E-4)**	0.10382(3.56E-3)	**0.00033(4.12E-4)**	0.13951(5.20E-2)	**0.00273(2.32E-4)**	0.14149(2.58E-2)	**0.00518(2.19E-4)**	0.13
	b	**0.00077(3.10E-4)**	0.02822(2.42E-3)	**0.00067(1.68E-4)**	0.03979(2.18E-3)	**0.00077(9.23E-5)**	0.04594(1.02E-2)	**0.00748(6.69E-5)**	0.03
5	c	**0.03127(4.12E-2)**	0.21898(4.83E-3)	**0.00536(2.76E-2)**	0.24637(2.83E-2)	**0.03192(2.48E-2)**	0.27167(3.13E-2)	**0.01721(2.05E-2)**	0.26
	b	**0.06781(3.21E-2)**	0.20088(2.18E-2)	**0.00142(1.79E-2)**	0.21689(3.86E-2)	**0.07622(1.47E-2)**	0.20411(1.84E-2)	**0.03721(8.56E-3)**	0.21
8	c	**0.08332(4.29E-2)**	0.17453(5.29E-2)	**0.06562(3.11E-2)**	0.30752(1.27E-1)	**0.07570(4.34E-2)**	0.21631(1.74E-1)	**0.08773(2.88E-3)**	0.20
	b	**0.07793(1.02E-1)**	0.18973(2.62E-2)	**0.05571(1.53E-2)**	0.21554(1.82E-2)	**0.12148(7.82E-2)**	0.22883(3.18E-2)	**0.07242(7.62E-2)**	0.28
10	c	**0.11731(2.11E-1)**	0.22708(1.29E-2)	**0.07922(3.03E-2)**	0.87988(1.02E-2)	**0.18728(3.77E-1)**	0.88635(4.28E-3)	**0.13752(3.20E-1)**	0.87
	b	**0.12712(1.71E-1)**	0.16275(3.28E-2)	**0.11082(1.88E-1)**	0.23098(1.28E-3)	**0.21781(2.38E-1)**	0.24970(2.91E-2)	**0.11683(2.02E-1)**	0.22

The ROI column gives the type of the DM supplied utopia weights. c indicates the preference on the middle region of the PF while b indicates the preference on an extreme. All better results are with statistical significance according to Wilcoxon signed-rank test with a 95% confidence level, and are highlighted in bold face with a gray background.

From the results shown in Table 3, we observe the overwhelming superiority of I-MOEA/D-PLVF and I-NSGA-III-PLVF, over the baseline MOEA/D and NSGA-III, for approximating the DM preferred solution. In particular, they obtain statistically significantly better metric values (i.e., smaller approximation error) on all test problems. In the following paragraphs, we discuss the results from three aspects.

- Fig. 5 to Fig. 12 plot the populations (with respect to the best approximation error) obtained by different algorithms. Note that since the observations on DTLZ3 and DTLZ4 test problems are similar to those on DTLZ2 test problem, we only show the plots on DTLZ1 and DTLZ2 test problems in this paper while the complete results are put in Section III of the supplementary document. From these plots, we can observe that both I-MOEA/D-PLVF and I-NSGA-III-PLVF are always able to find solutions that well approximate the unknown DM's golden point in a decent accuracy as shown in Table 3. In contrast, since the baseline MOEA/D and NSGA-III are designed to approximate the whole PF, it is not surprised to see that most of their solutions are away from the DM's golden point. Although some of the solutions obtained by the baseline MOEA/D and NSGA-III can by chance pass the ROI, i.e., the vicinity of the DM's golden point, they still have a observable distance from the DM's golden point. Moreover, the other solutions away from the ROI will unarguably result in the cognitive noise to *posteriori* decision-making procedure, especially for problems that have many objectives.
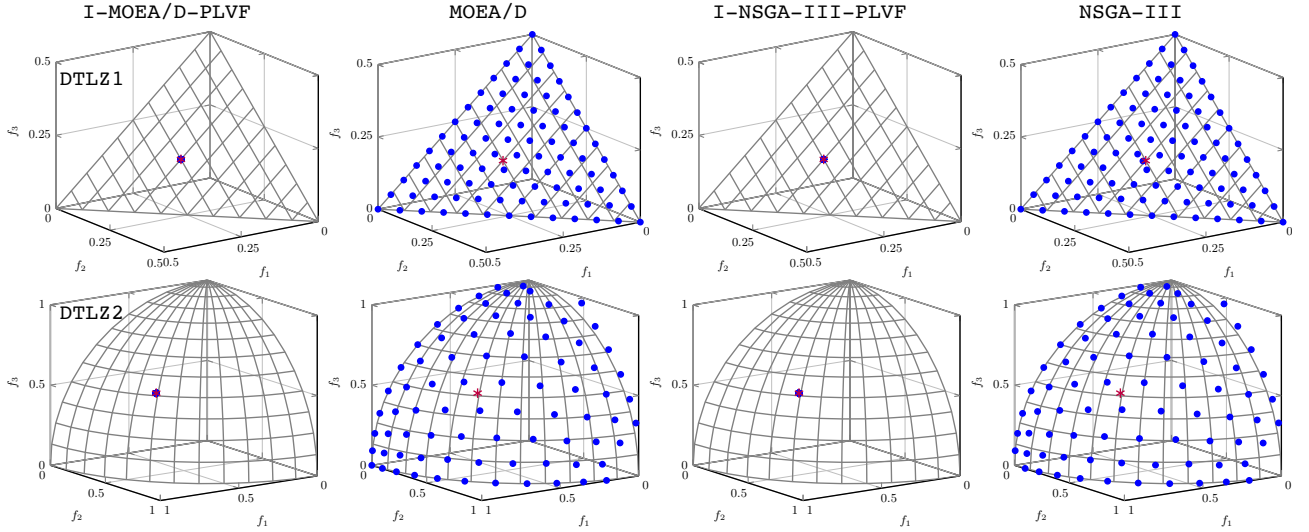
Figure 5: Solutions obtained on 3-objective DTLZ1 and DTLZ2 test problems where $\mathbf{z}^r$, which prefers the middle region of the PF, is represented as the red asterisk.
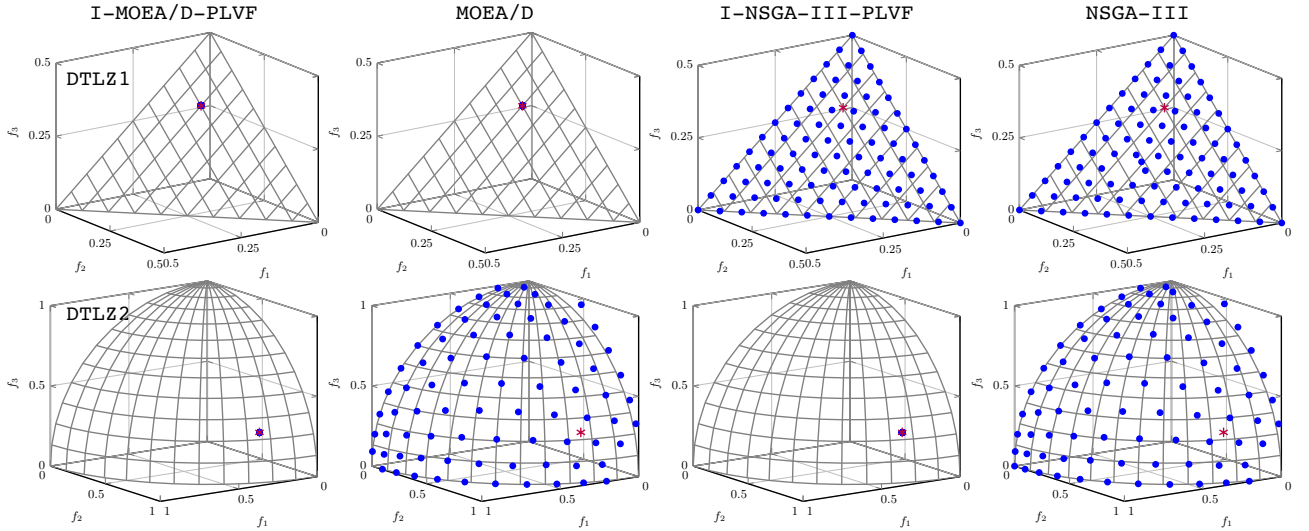

Figure 6: Solutions obtained on 3-objective DTLZ1 and DTLZ2 test problems where $\mathbf{z}^r$, which prefers one side of the PF, is represented as the red asterisk.

- From the results shown in Table 3, we find that it seems to be more difficult for the baseline MOEA/D and NSGA-III to find the DM's preferred solution on the middle region of the PF than
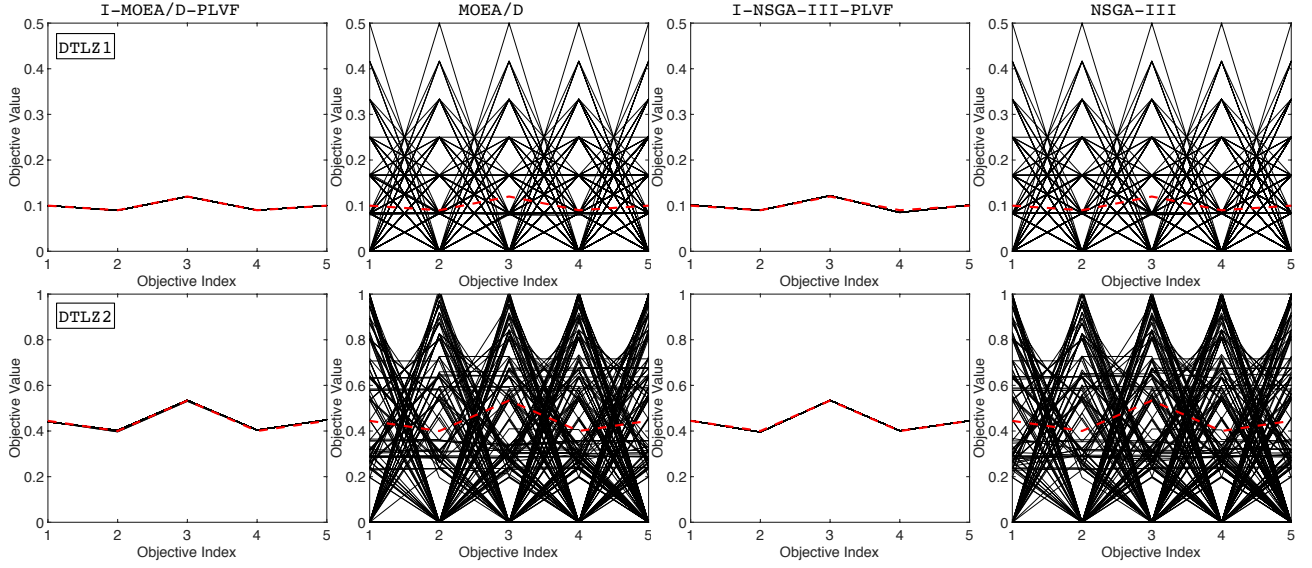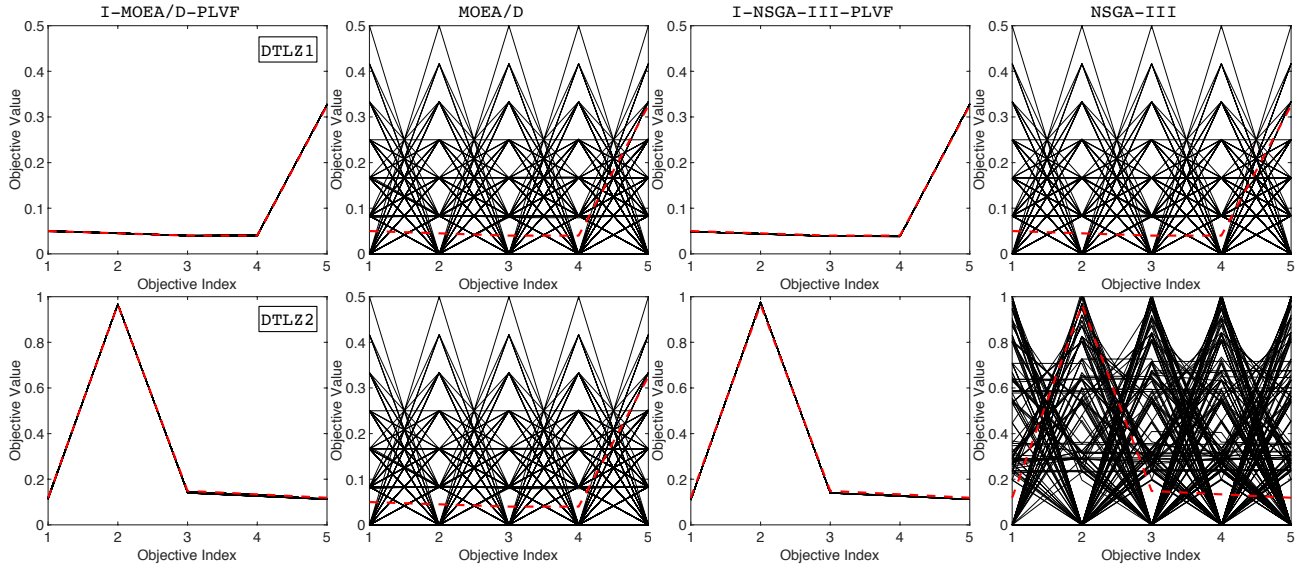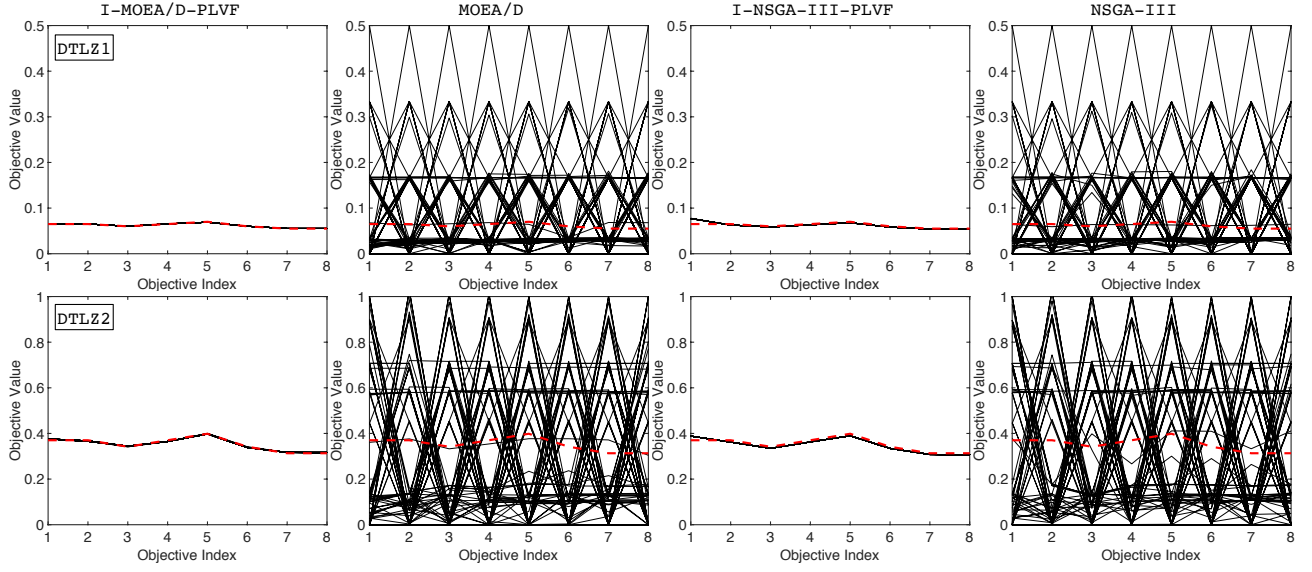
Figure 7: Solutions obtained on 5-objective DTLZ1 and DTLZ2 test problems where $\mathbf{z}^r$, which prefers the middle region of the PF, is represented as the red dotted line.
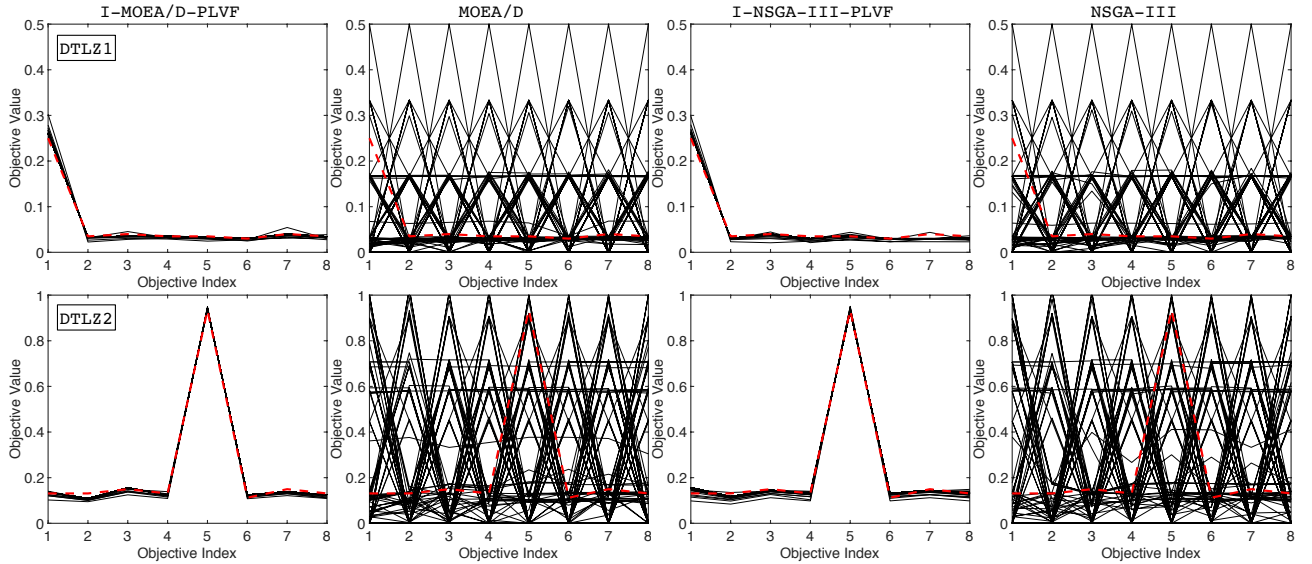

Figure 8: Solutions obtained on 5-objective DTLZ1 and DTLZ2 test problems where $\mathbf{z}^r$, which prefers one side of the PF, is represented as the red dotted line.
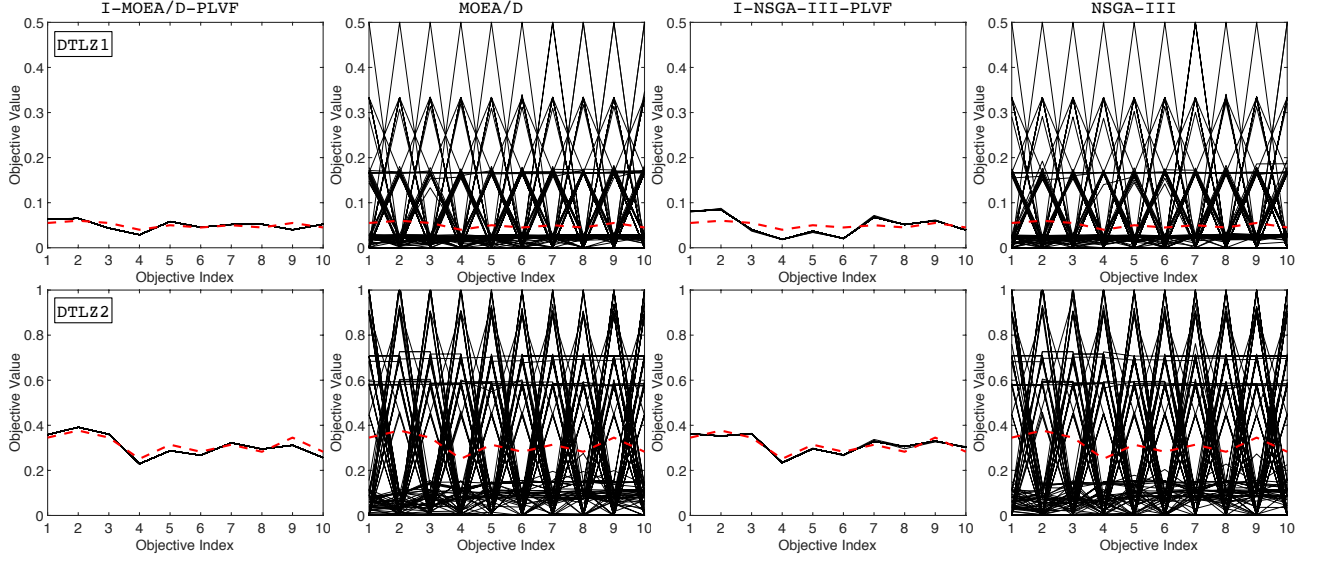
Figure 9: Solutions obtained on 8-objective DTLZ1 and DTLZ2 test problems where $\mathbf{z}^r$, which prefers the middle region of the PF, is represented as the red dotted line.

Figure 10: Solutions obtained on 8-objective DTLZ1 and DTLZ2 test problems where $\mathbf{z}^r$, which prefers one side of the PF, is represented as the red dotted line.

Figure 11: Solutions obtained on 10-objective DTLZ1 and DTLZ2 test problems where $\mathbf{z}^r$, which prefers the middle region of the PF, is represented as the red dotted line.
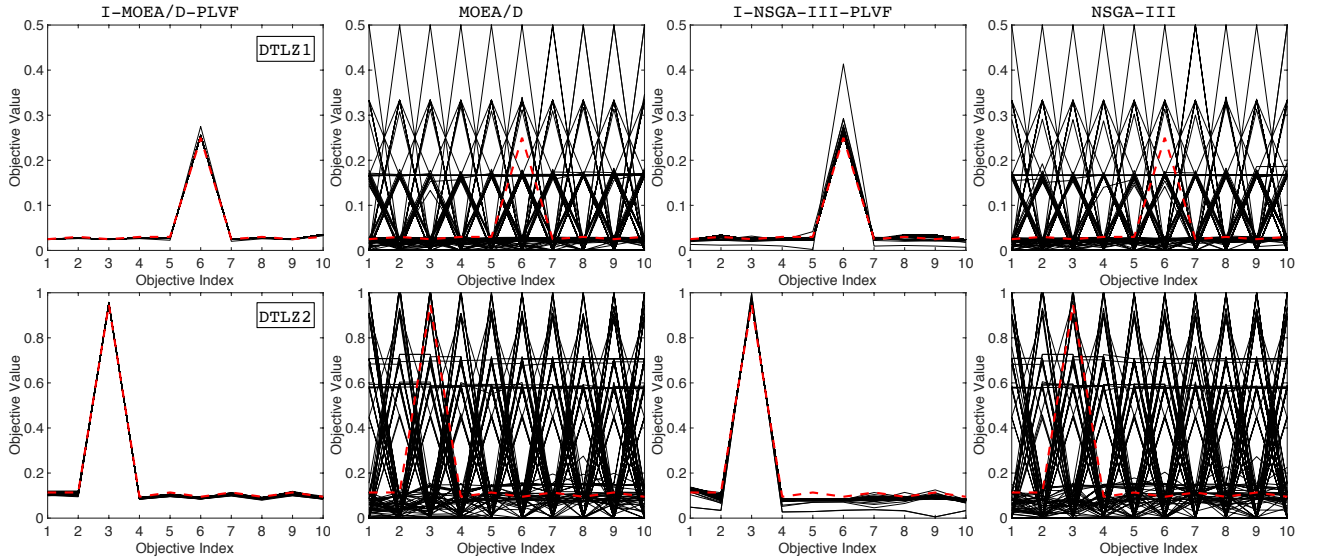

Figure 12: Solutions obtained on 10-objective DTLZ1 and DTLZ2 test problems where $\mathbf{z}^r$, which prefers one side of the PF, is represented as the red dotted line.

those biased toward a particular extreme of the PF. This is because if the ROI is on one side of the PF, it is more or less close to the boundary. The baseline MOEA/D and NSGA-III, which were originally designed to approximate the whole PF, can always find solutions on the boundary, whereas it becomes increasingly difficult to find solutions on the middle region of the PF with the increase of the number of objectives. Therefore, the approximation error to a DM's golden point on one side of the PF seems to be better than those on the middle region of the PF. In contrast, since our proposed interactive framework can progressively learn the DM's preference information and adjust the search direction, I-MOEA/D-PLVF and I-NSGA-III-PLVF can well approximate the ROI in any part of the PF.

- Furthermore, we find that the performance of I-MOEA/D-PLVF and I-NSGA-III-PLVF do not depend on the shape of the PF (in particular DTLZ1 test problem has a linear PF while DTLZ2 to DTLZ4 test problems have a concave PF). But the performance of the proposed interactive framework can be influenced by the difficulty of the search space. In particular, if the search space contains many local PFs, like DTLZ1 and DTLZ3, the evolving population may need a long time to jump over these local PFs. Even worse, some region of the PF will be more difficult to approximate than the others. If this region happens to be the ROI, the DM will wrongly assign a higher score to the solutions outside the ROI. In Fig. 13 and Fig. 14, we plot the variations of the approximation error versus the number of generations on the 3-objective case while more comprehensive results can be found in Section IV of the supplementary document. From these plots, we can see that the approximation error on the relatively simple DTLZ2 test problem quickly drops down at the early stage of the evolution. But for problems with many local PFs, i.e., DTLZ1 and DTLZ3 test problems, the trajectories of approximation error struggle longer time before dropping down. Although DTLZ4 test problem does not have local PFs, its search space has strong bias toward certain objective coordinates. Accordingly, we observe the the fluctuation of the trajectories over generations. This might be caused by the biased evolving population which mislead the DM in decision-making.
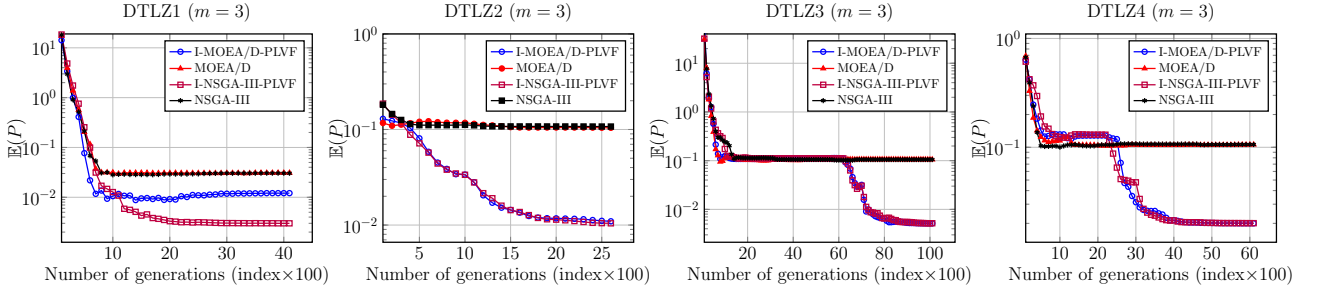

Figure 13: Trajectories of the approximation error versus the number of generations on 3-objective DTLZ1 to DTLZ4 test problems. The DM's 'golden' point prefers the middle region of the PF.
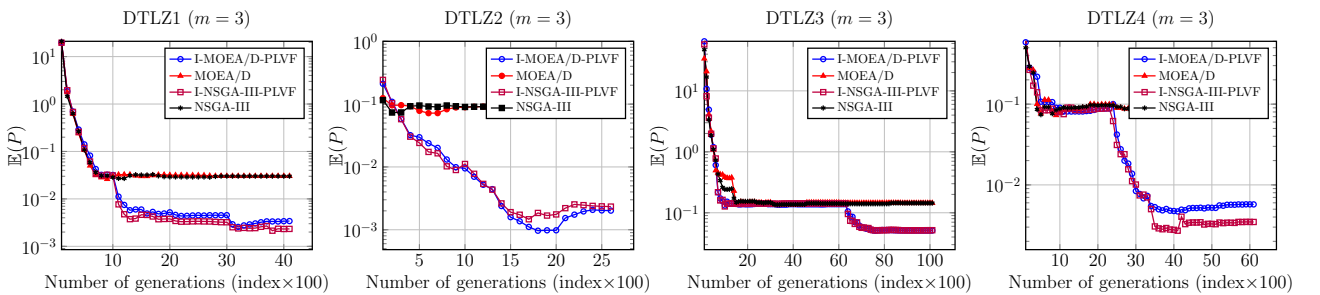

Figure 14: Trajectories of the approximation error versus the number of generations on 3-objective DTLZ1 to DTLZ4 test problems. The DM's 'golden' point prefers one side of the PF.

5.2 Parametric Studies

As introduced in Section 4, besides the intrinsic parameters associated with an EMO algorithm, e.g., population size, crossover and mutation probabilities, in our proposed interactive framework, there are some additional parameters that may affect the performance for approximating the ROI. They are: the number of incumbent candidates presented to the DM for scoring (μ), the number of generations between two consecutive consultation sessions (τ), and the step size of the reference point update (η). In this subsection, we study the effects of these parameters, while keeping the other parameters of I-MOEA/D-PLVF and I-NSGA-III-PLVF the same as introduced in Section 4. In particular, we use DTLZ1 and DTLZ2 as the test problems, given the observations on DTLZ3 and DTLZ4 test problems can be generalised from those in DTLZ2. Each algorithm is run 21 times with different random seeds.

5.2.1 Effect of μ

As introduced in Section 3.1, μ determines the number of labeled data (scored by the DM) that can be used to train the AVF model. It makes sense that the more data you can provide, the more accurate AVF model you can expect. However, presenting the DM too many alternatives for scoring will definitely increase her/his workload, thus lead to the fatigue. On the other hand, the model accuracy will be impaired if the data is not sufficient. To study the effect of μ, we consider three different settings, i.e., $\mu \in \{5, 10, 20\}$. Furthermore, to validate the importance of an accurate AVF model for helping the interactive framework, we also investigate an "utopia" scenario where I-MOEA/D-PLVF and I-NSGA-III-PLVF directly use the DM's golden value function in the preference elicitation module. In Fig. 15, we show the variations of the median approximation error with respect to different μ settings and the utopia scenario. Note that we only show the three-objective case since the other observations are similar (more comprehensive results can be found in Section V of the supplementary document). As expected, I-MOEA/D-PLVF and I-NSGA-III-PLVF always perform best when directly using the DM's golden value function. This observation supports the importance of an accurate model. Moreover, I-MOEA/D-PLVF and I-NSGA-III-PLVF can have a better performance when using a large μ.
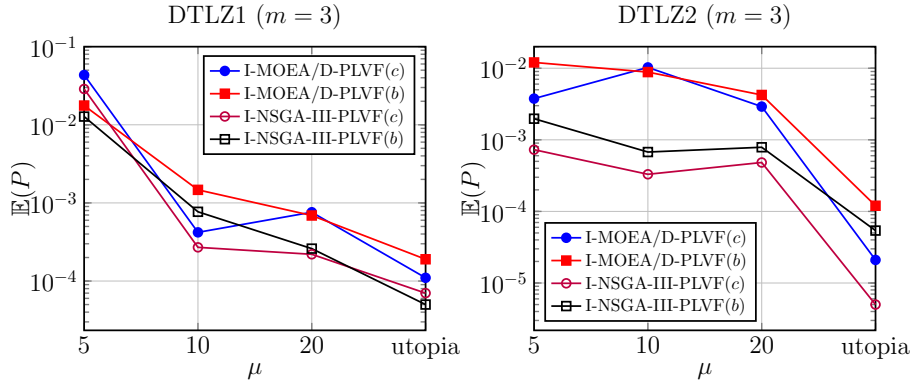

Figure 15: Variations of the approximation errors with different μ settings. (c) indicates the preference on the middle region of the PF, while (b) indicates the preference on an extreme.

5.2.2 Effect of τ

Here we study the effect of τ by considering three different settings, i.e, $\tau \in \{10, 25, 50\}$. In Fig. 16, we plot the variations of the median approximation error with respect to different τ settings in the three-objective scenario while more comprehensive results can be found in Section V of the supplementary document. Specifically, a small τ means that we need to frequently ask the DM for scoring the candidate solutions and then update the AVF model accordingly. To a certain extent, this operation can improve the model accuracy for approximating the DM's preference information. However, similar to the overfitting phenomenon in machine learning, too frequent DM calls also have the risk of premature convergence on some local optima. As shown in Fig. 16, the performance of I-MOEA/D-PLVF and

17

I-NSGA-III-PLVF is not promising when setting $\tau = 10$ on DTLZ1 test problem which has more than 11^4 local PFs. On the other hand, if the DM is rarely been consulted by using a large τ, the consultation module can hardly get enough information from the DM. Thus, we can hardly expect that the AVF model can provide useful information that truly represent the DM's preference information to the optimisation module. As expected, the performance of I-MOEA/D-PLVF and I-NSGA-III-PLVF is always not satisfactory when setting $\tau = 50$.
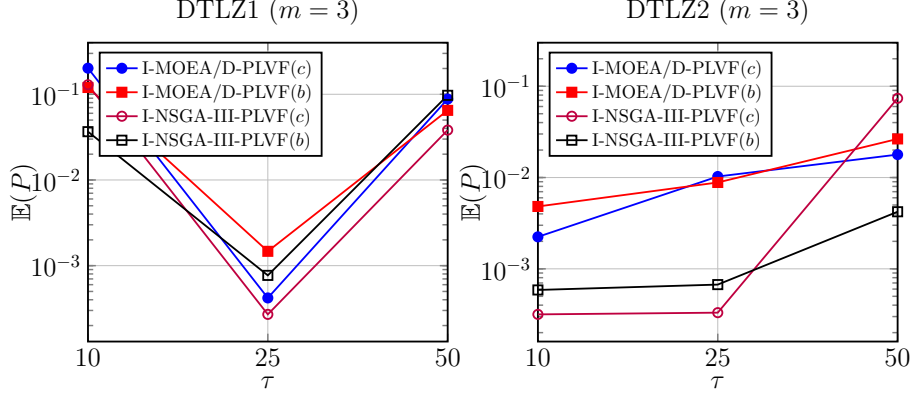

Figure 16: Variations of the approximation errors with different τ settings. (c) indicates the preference on the middle region of the PF, while (b) indicates the preference on an extreme.

5.2.3 Effect of η

As introduced in Section 3.2, η controls the convergence rate of the reference points toward the promising ones identified by the AVF model learned from the consultation module. A large η will lead to a fast convergence, thus it may have a risk of pre-mature convergence toward an undesired region. On the contrary, a small η may slow down the convergence toward the ROI within the limited number of FEs. To study the effect of μ, we consider different μ settings as $\eta \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. From the results shown in Fig. 17 and more comprehensive results shown in Section V of the supplementary document, we find that the best setting of η is problem dependent. But neither too large nor too small η can offer a satisfactory result.
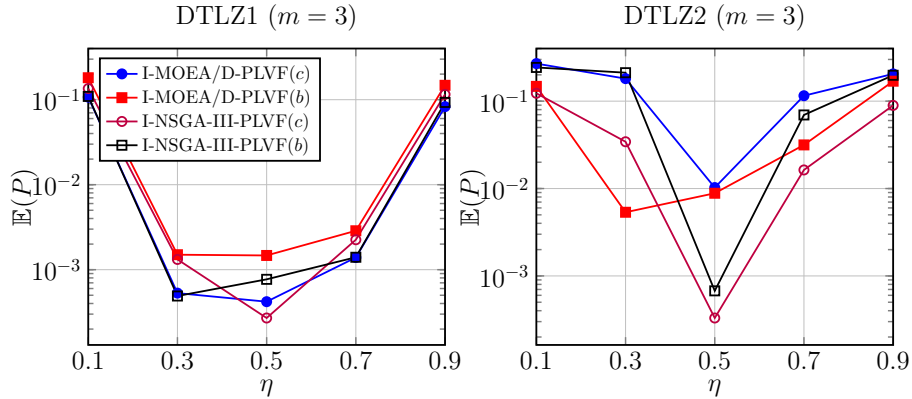

Figure 17: Variations of the approximation errors with different η settings. (c) indicates the preference on the middle region of the PF, while (b) indicates the preference on an extreme.

5.3 Random Error in DM's Preference Information

In the previous experiments, the DM's preference information is modelled as a deterministic value function. However, as a human being, the DM can show certain level of inconsistencies when providing her/his preference information in practice. To study the effect brought by the inconsistencies in the

preference elicitation, we consider adding some stochastic noises to the DM's golden value function. Specifically, the stochastic DM's golden value function is defined as:

$$\overline{\psi}(\mathbf{x}) = \psi(\mathbf{x}) \times \mathsf{noise}(1, \delta_t^2) \tag{8}$$

where $\psi(\mathbf{x})$ is calculated according to equation (6). $\mathsf{noise}(*, *)$ is a Gaussian noise, $\delta_t = \kappa \times (1 - \frac{T}{T_{\max}})$, κ determines the strength of the noise level. T is the current generation counter and $T_{\max}$ is the pre-defined maximum number of generations. In this case, we can expect that the DM's inconsistencies decrease with the evolution. This setting tries to simulate a realistic DM who is likely to make errors at the beginning due to her/his lack of knowledge of the problem. With the progress of evolution, the DM becomes more consistent in preference elicitation and decision-making. To study the influence of DM's inconsistencies on the performance of our proposed interactive framework, we set $\kappa \in \{0.0, 0.1, 0.5\}$ separately. In particular, it becomes the deterministic value function if $\kappa = 0.0$. Here we choose DTLZ1 and DTLZ2 as the test problems, and keep the parameters of I-MOEA/D-PLVF and I-NSGA-III-PLVF the same as introduced in Section 4. Fig. 18 shows the variations of the approximation error with respect to different κ settings (more comprehensive results are shown in Section VI of the supplementary document). As expected the performance of I-MOEA/D-PLVF and I-NSGA-III-PLVF degenerates with the increase of κ.
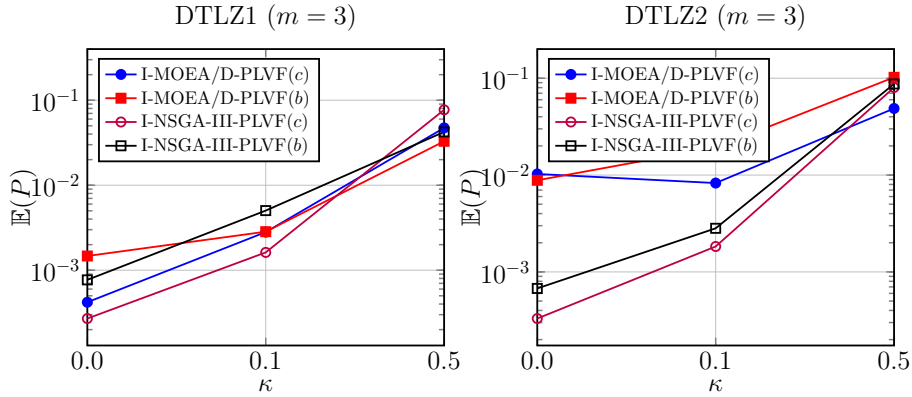

Figure 18: Variations of the approximation errors with different κ settings. (c) indicates the preference on the middle region of the PF, while (b) indicates the preference on an extreme.

6 Conclusions and Future Directions

This paper has proposed a simple yet elegant interactive framework which can be embedded in any decomposition-based EMO algorithm so that the ultimate goal is not a complete PF, but a preferred solution of the DM's choice. This interactive framework consists of three modules, i.e, consultation, preference elicitation and optimisation. Specifically, the consultation module aims to progressively learn an AVF to model the DM's preference information. In particular, during the consultation session, the DM is presented with a few incumbent candidates for scoring according her/his preference. Once the AVF is learned, the preference elicitation module translates it into the form that can be used in a decomposition-based EMO algorithm, i.e., a set of reference points that are biased toward the ROI. Extensive experiments on three to ten-objective test problems fully demonstrate the effectiveness of our proposed interactive framework for helping two widely used decomposition-based EMO algorithms, i.e., MOEA/D and NSGA-III, for finding the DM's preferred solution.

The proposed interactive framework can be extended in a number of different ways.

- First, this paper assume that the DM's preference information is represented as a monotonic value function. However, in practice, it is not uncommon that the DM judges some of the alternatives to be incomparable. How to discriminate the order information from incomparable comparisons? Moreover, instead of assigning a scalar score to a solution, it is interesting to study how to derive the preference information through holistic comparisons among incumbent solutions.

- This paper only considers the unconstrained MOPs, in which any preference information elicited by the DM is feasible. In the presence of constraints or the PF by itself is discontinuous, some region of the attainable objective space is infeasible but is also unknown to the DM. It is interesting to study how to handle the infeasible information and to inform the DM to modify her/his preference information in due course.

- Although this paper has restricted the value function to be of certain form (6), other more value function structures can also be considered. Furthermore, it is interesting to further investigate the robustness consideration in deriving the AVF. More studies are required to investigate the side effects brought by the inconsistencies in decision-making and the ways to mitigate that.

- In this paper, the termination criterion is the same as the standard EMO algorithm, i.e., the fixed number of FEs. However, as shown in the plots of the trajectories of the approximation errors, we can see that algorithms can be terminated earlier before the pre-defined number of FEs. It is interesting to investigate some optimal stopping criterion that is able to automatically terminates an algorithm when it is necessary.

As discussed in [67], it seems to be an inspiring direction for future research of MO to combine the ideas from the EMO algorithms and MCDM. More efforts are required to develop pragmatic algorithms for MO and decision-making.

Acknowledgment

References

[1] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Trans. Evolutionary Computation*, vol. 17, no. 3, pp. 321–344, 2013.

[2] M. Antonelli, P. Ducange, and F. Marcelloni, "A fast and efficient multi-objective evolutionary learning scheme for fuzzy rule-based classifiers," *Information Sciences*, vol. 283, pp. 36–54, 2014.

[3] S. N. Qasem, S. M. Shamsuddin, S. Z. M. Hashim, M. Darus, and E. Al-Shammari, "Memetic multiobjective particle swarm optimization-based radial basis function network for classification problems," *Information Sciences*, vol. 239, no. 1, pp. 165–190, 2013.

[4] S. Rostami, D. O'Reilly, A. Shenfield, and N. Bowring, "A novel preference articulation operator for the evolutionary multi-objective optimisation of classifiers in concealed weapons detection," *Information Sciences*, vol. 295, pp. 494–520, 2014.

[5] A. A. Montaño, C. A. C. Coello, and E. Mezura-Montes, "Multiobjective evolutionary algorithms in aeronautical and aerospace engineering," *IEEE Trans. Evolutionary Computation*, vol. 16, no. 5, pp. 662–694, 2012.

[6] H. R. Cheshmehgaz, M. Desa, and A. Wibowo, "A flexible three-level logistic network design considering cost and time criteria with a multi-objective evolutionary algorithm," *Journal of Intelligent Manufacturing*, pp. 1–17, 2011.

[7] C. Zhou, J. Zheng, K. Li, and H. Lv, "Objective reduction based on the least square method for large-dimensional multi-objective optimization problem," in *ICNC'09: Proc. of 15th International Conference on Natural Computation*, 2009, pp. 350–354.

[8] K. Li, J. Zheng, M. Li, C. Zhou, and H. Lv, "A novel algorithm for non-dominated hypervolume-based multiobjective optimization," in *SMC'09: Proc. of the 2009 IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 5220–5226.

[9] K. Li, S. Kwong, R. Wang, J. Cao, and I. J. Rudas, "Multi-objective differential evolution with self-navigation," in *SMC'12: Proc. of the 2012 IEEE International Conference on Systems, Man, and Cybernetics*, 2012, pp. 508–513.

[10] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, and R. Shen, "Achieving balance between proximity and diversity in multi-objective evolutionary algorithm," *Inf. Sci.*, vol. 182, no. 1, pp. 220–242, 2012.

[11] K. Li, S. Kwong, R. Wang, K. Tang, and K. Man, "Learning paradigm based on jumping genes: A general framework for enhancing exploration in evolutionary multiobjective optimization," *Inf. Sci.*, vol. 226, pp. 1–22, 2013.

[12] K. Li, R. Wang, S. Kwong, and J. Cao, "Evolving extreme learning machine paradigm with adaptive operator selection and parameter control," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 21, pp. 143–154, 2013.

[13] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Trans. Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2014.

[14] K. Li, Á. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evolutionary Computation*, vol. 18, no. 1, pp. 114–130, 2014.

[15] K. Li and S. Kwong, "A general framework for evolutionary multiobjective optimization via manifold learning," *Neurocomputing*, vol. 146, pp. 65–74, 2014.

[16] M. Wu, S. Kwong, Q. Zhang, K. Li, R. Wang, and B. Liu, "Two-level stable matching-based selection in MOEA/D," in *SMC'15: Proc. of the 2015 IEEE International Conference on Systems, Man, and Cybernetics*, 2015, pp. 1720–1725.

[17] K. Li, K. Deb, and Q. Zhang, "Evolutionary multiobjective optimization with hybrid selection principles," in *CEC'15: Proc. of 2015 IEEE Congress on Evolutionary Computation*, 2015, pp. 900–907.

[18] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[19] K. Li, S. Kwong, and K. Deb, "A dual-population paradigm for evolutionary multiobjective optimization," *Inf. Sci.*, vol. 309, pp. 50–72, 2015.

[20] K. Li, M. N. Omidvar, K. Deb, and X. Yao, "Variable interaction in multi-objective optimization problems," in *PPSN'16: Proc. of 14th International Conference on Parallel Problem Solving from Nature*, 2016, pp. 399–409.

[21] K. Li and K. Deb, "Performance assessment for preference-based evolutionary multi-objective optimization using reference points," COIN Report, Tech. Rep., 2016.

[22] M. Wu, S. Kwong, Y. Jia, K. Li, and Q. Zhang, "Adaptive weights generation for decomposition-based multi-objective optimization using gaussian process regression," in *GECCO'17: Proc. of the 2017 Genetic and Evolutionary Computation Conference*, 2017, pp. 641–648.

[23] K. Li, K. Deb, O. T. Altinöz, and X. Yao, "Empirical investigations of reference point based methods when facing a massively large number of objectives: First results," in *EMO'17: Proc. of the 9th International Conference on Evolutionary Multi-Criterion Optimization*, 2017, pp. 390–405.

[24] M. Wu, K. Li, S. Kwong, Y. Zhou, and Q. Zhang, "Matching-based selection with incomplete lists for decomposition multiobjective optimization," *IEEE Trans. Evolutionary Computation*, vol. 21, no. 4, pp. 554–568, 2017.

[25] K. Li, K. Deb, Q. Zhang, and Q. Zhang, "Efficient nondomination level update method for steady-state evolutionary multiobjective optimization," *IEEE Trans. Cybernetics*, vol. 47, no. 9, pp. 2838–2849, 2017.

[26] R. Chen, K. Li, and X. Yao, "Dynamic multiobjectives optimization with a changing number of objectives," *IEEE Trans. Evolutionary Computation*, vol. 22, no. 1, pp. 157–171, 2018.

[27] K. Li, K. Deb, and X. Yao, "R-metric: Evaluating the performance of preference-based evolutionary multi-objective optimization using reference points," *IEEE Trans. Evol. Comput.*, 2018, accepted for publication.

[28] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multi-objective optimization," *IEEE Trans. Evol. Comput.*, 2018, accepted for publication.

[29] M. Wu, K. Li, S. Kwong, Q. Zhang, and J. Zhang, "Learning to decompose: A paradigm for decomposition-based multi-objective optimization," *IEEE Trans. Evol. Comput.*, 2018, accepted for publication.

[30] K. Li, R. Chen, G. Min, and X. Yao, "Integration of preferences in decomposition-based evolutionary multi-objective optimization," *IEEE Trans. Cybernetics*, 2018, accepted for publication.

[31] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[32] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *EUROGEN'01: Proc. of 2001 Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems*, 2002, pp. 95–100.

[33] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[34] J. Branke, T. Kauler, and H. Schmeck, "Guidance in evolutionary multi-objective optimization," *Advances in Engineering Software*, vol. 32, no. 6, pp. 499–507, 2001.

[35] K. Deb, "Multi-objective evolutionary algorithms: Introducing bias among pareto-optimal solutions," in *Advances in Evolutionary Computing*, A. Ghosh and S. Tsutsui, Eds. Springer Berlin Heidelberg, 2003, pp. 263–292.

[36] J. Branke and K. Deb, "Integrating user preferences into evolutionary multi-objective optimization," in *Knowledge Incorporation in Evolutionary Computation*, Y. Jin, Ed. Springer Berlin Heidelberg, 2005, vol. 167, pp. 461–477.

[37] K. Deb, J. Sundar, U. Bhaskara, and S. Chaudhuri, "Reference point based multiobjective optimization using evolutionary algorithms," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 273–286, 2006.

[38] K. Deb and A. Kumar, "Interactive evolutionary multi-objective optimization and decision-making using reference direction method," in *Genetic and Evolutionary Computation Conference, GECCO 2007, Proceedings, London, England, UK, July 7-11, 2007*, 2007, pp. 781–788.

[39] ——, "Light beam search based multi-objective optimization using evolutionary algorithms," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2007, 25-28 September 2007, Singapore*, 2007, pp. 2125–2132.

[40] G. W. Greenwood, X. Hu, and J. G. D'Ambrosio, "Fitness functions for multiple objective optimization problems: Combining preferences with pareto rankings," in *Proceedings of the 4th Workshop on Foundations of Genetic Algorithms. San Diego, CA, USA, August 5 1996*, 1996, pp. 437–455.

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

[42] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE Trans. Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[43] I. Das and J. E. Dennis, "Normal-Boundary Intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, pp. 631–657, 1998.

[44] S. Jiang, Z. Cai, J. Zhang, and Y. Ong, "Multiobjective optimization by decomposition with pareto-adaptive weight vectors," in *ICNC'11: Proc. of the 7th International Conference on Natural Computation*, 2011, pp. 1260–1264.

[45] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.

[46] A. Mohammadi, M. N. Omidvar, X. Li, and K. Deb, "Integrating user preferences and decomposition methods for many-objective optimization," in *CEC'14: Proc. of the 2014 IEEE Congress on Evolutionary Computation*, 2014, pp. 421–428.

[47] X. Ma, F. Liu, Y. Qi, L. Li, L. Jiao, X. Deng, X. Wang, B. Dong, Z. Hou, Y. Zhang, and J. Wu, "MOEA/D with biased weight adjustment inspired by user preference and its application on multi-objective reservoir flood control problem," *Soft Comput.*, vol. 20, no. 12, pp. 4999–5023, 2016.

[48] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Trans. Cybernetics*, vol. 45, no. 10, pp. 2076–2088, 2015.

[49] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[50] A. Jaszkiewicz and J. Branke, "Interactive multiobjective evolutionary algorithms," in *Multiobjective Optimization–Interactive and Evolutionary Approaches*. Springer, 2008, pp. 179–193.

[51] 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*, 2015, vol. 98, pp. 141–207.

[52] H. Wang, M. Olhofer, and Y. Jin, "A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges," *Complex & Intelligent Systems*, vol. 3, no. 4, pp. 233–245, 2017.

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

[54] A. Jaszkiewicz, "Interactive multi-objective optimization with pareto memetic algorithm," *Foundations of Computing and Decision Sciences*, vol. 32, no. 1, pp. 15–32, 2007.

[55] K. Deb, A. Sinha, P. J. Korhonen, and J. Wallenius, "An interactive evolutionary multiobjective optimization method based on progressively approximated value functions," *IEEE Trans. Evolutionary Computation*, vol. 14, no. 5, pp. 723–739, 2010.

[56] R. Battiti and A. Passerini, "Brain-computer evolutionary multiobjective optimization: A genetic algorithm adapting to the decision maker," *IEEE Trans. Evolutionary Computation*, vol. 14, no. 5, pp. 671–687, 2010.

[57] T. Joachims, "Training linear svms in linear time," in *KDD'06: Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 217–226.

[58] J. Branke, S. Greco, R. Slowinski, and P. Zielniewicz, "Interactive evolutionary multiobjective optimization using robust ordinal regression," in *EMO'09: Proc. of the 5th International Conference on Evolutionary Multi-Criterion Optimization*, 2009, pp. 554–568.

[59] ——, "Learning value functions in interactive evolutionary multiobjective optimization," *IEEE Trans. Evolutionary Computation*, vol. 19, no. 1, pp. 88–102, 2015.

[60] P. Korhonen, H. Moskowitz, and J. Wallenius, "A progressive algorithm for modeling and solving multiple-criteria decision problems," *Operational Research*, vol. 34, no. 5, pp. 726–731, 1986.

[61] P. Korhonen, H. Moskowitz, P. Salminen, and J. Wallenius, "Further developments and tests of a progressive algorithm for multiple criteria decision making," *Operational Research*, vol. 41, no. 6, pp. 1033–1045, 1993.

[62] M. D. Buhmann, *Radial Basis Functions*. Cambridge University Press, 2003.

[63] R. L. Hardy, "Multiquadric equations of topography and other irregular surfaces," *Journal of Geophysical Research*, vol. 76, no. 8, pp. 1905–1915, 1971.

[64] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, pp. 1–34, 1994.

[65] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.

[66] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," in *Evolutionary Multiobjective Optimization*, ser. Advanced Information and Knowledge Processing, A. Abraham, L. Jain, and R. Goldberg, Eds. Springer London, 2005, pp. 105–145.

[67] R. C. Purshouse, K. Deb, M. M. Mansor, S. Mostaghim, and R. Wang, "A review of hybrid evolutionary multiple criteria decision making methods," in *CEC'14: Proc. of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*, 2014, pp. 1147–1154.