Progressive Preference Learning: Proof-of-Principle Results in MOEA/D

Ke Li (✉)

Department of Computer Science, University of Exeter, Exeter EX4 4QF, UK
k.li@exeter.ac.uk

Abstract. Most existing studies on evolutionary multi-objective optimisation (EMO) focus on approximating the whole Pareto-optimal front. Nevertheless, rather than the whole front, which demands for too many points (especially when having many objectives), a decision maker (DM) might only be interested in a partial region, called the region of interest (ROI). Solutions outside this ROI can be noisy to the decision making procedure. Even worse, there is no guarantee that we can find DM preferred solutions when tackling problems with complicated properties or a large number of objectives. In this paper, we use the state-of-the-art MOEA/D as the baseline and develop its interactive version that is able to find solutions preferred by the DM in a progressive manner. Specifically, after every several generations, the DM is asked to score a limited number of candidates. Then, an approximated value function, which models the DM's preference information, is learned from the scoring results. Thereafter, the learned preference information is used to obtain a set of weight vectors biased towards the ROI. Note that these weight vectors are thus used in the baseline MOEA/D to search for DM preferred solutions. Proof-of-principle results on 3- to 10-objective test problems demonstrate the effectiveness of our proposed method.

Keywords: Interactive multi-objective optimisation · Preference learning · MOEA/D

1 Introduction

The multi-objective optimisation problem (MOP) considered in this paper is formulated as:

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

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. $\boldsymbol{\Omega}$ is the feasible set in the decision space $\mathbb{R}^n$ and $\mathbf{F} : \boldsymbol{\Omega} \rightarrow \mathbb{R}^m$ is the corresponding attainable set in the objective space

Supported by Royal Society under grant IEC/NSFC/170243.

K. Deb et al. (Eds.): EMO 2019, LNCS 11411, pp. 631–643, 2019.
https://doi.org/10.1007/978-3-030-12598-1_50

$\mathbb{R}^m$. Without considering the DM's preference information, given two solutions $\mathbf{x}^1, \mathbf{x}^2 \in \mathbf{\Omega}$, $\mathbf{x}^1$ 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 $\mathbf{x} \in \mathbf{\Omega}$ is said to be Pareto-optimal if and only if there is no solution $\mathbf{x}' \in \mathbf{\Omega}$ that dominates it. The set of all Pareto-optimal solutions is called the Pareto-optimal set (PS) and their corresponding objective vectors form the Pareto-optimal front (PF). Accordingly, the ideal point is defined as $\mathbf{z}^* = (z_1^*, \cdots, z_m^*)^T$, where $z_i^* = \min_{\mathbf{x} \in PS} f_i(\mathbf{x})$.

Evolutionary algorithms, which work with a population of solutions and can approximate a set of trade-off solutions simultaneously, have been widely accepted as a major tool for solving MOPs. Over the past two decades and beyond, many efforts have been devoted to developing EMO algorithms, e.g. NSGA-II [7], IBEA [23] and MOEA/D [22]. The ultimate goal of multi-objective optimisation is to help the DM find solutions that meet at most her/his preference. Supplying a DM with a large amount of trade-off points not only increases her/his workload, but also provides many irrelevant or even noisy information to the decision making process. Moreover, due to the curse of dimensionality, approximating the whole high-dimensional PF not only becomes computationally inefficient (or even infeasible), but also causes a severe cognitive obstacle for the DM to comprehend the high-dimensional data. To facilitate the decision making process, it is more practical to incorporate the DM's preference information into the search process. By doing so, it allows the computational efforts to be concentrated on the ROI and thus has a better approximation therein. Generally speaking, preference information can be incorporated *a priori*, *posteriori* or *interactively*. Note that the traditional EMO just goes along the posteriori way whose disadvantages have been described before. When the preference information is elicited a priori, it is used to guide the solutions towards the ROI. However, it is non-trivial to faithfully model the preference information before solving the MOP at hand. In practice, articulating the preference information in an interactive manner, which has been studied in the multi-criterion decision making (MCDM) field for over four decades, seems to be interesting. This enables DMs to progressively learn and understand the characteristics of the MOP at hand and adjust their preference information. As a consequence, the solutions are effectively driven towards the ROI.

In the past decade, the development for hybrid EMO-MCDM schemes, where the DM's preference information is integrated into EMO either a priori or interactively, have become increasingly popular. Generally speaking, their ideas can be briefly summarised as the following five categories.

1. The first one employs weight information, e.g. relative importance order [12], to model DM's preference information. However, it is difficult to control the guidance of the search towards the ROI and there is no obvious motivation to utilise weights in an interactive manner.
2. The second sort modifies the trade-off information by either classifying objectives into different levels and priorities or expressing DM's preference information via fuzzy linguistic terms according to different aspiration levels, e.g. [19]. This is method is interesting yet complicated, especially when the number of

objectives becomes large [20]. In addition, using such approach interactively increases the DM's burden.
3. The third category tries to bias the density of solutions towards the ROI by considering DM's preference information, e.g. [4]. However, density/diversity management itself in EMO is difficult, especially in a high-dimensional space.
4. The fourth class, as a recent trend, combines DM's preference information with performance indicators in algorithm design, e.g. [21]. Nevertheless, the computational cost of certain popular performance indicator, e.g. hypervolume [1] increases exponentially with the number of objectives.
5. The last one uses aspiration level vector, which represents the DM's desired values of each objective, to assist the search process, e.g. [10,13]. As reported in [3], aspiration level vector have been recognised as one of the most popular ways to elicit DM's preference information. Without a demanding effort from the DM, she/he is able to guide the search towards the ROI even when encountering a large number of objectives.

Take MOEA/D, a state-of-the-art EMO algorithm, as the baseline, this paper develops a simple yet effective progressive preference learning paradigm. It progressively learns an approximated value function (AVF) from the DM's behaviour in an interactive manner. The learned preference information is thus used to guide the population towards the ROI. Generally speaking, the progressive preference learning paradigm consists of the following three modules.

- *Optimisation module*: it uses the preference information elicited from the preference elicitation module to find the preferred solutions. In principle, any EMO algorithm can be used as the search engine while this paper takes MOEA/D for proof-of-principle purpose.
- *Consultation module*: it is the interface by which the DM interacts with the optimisation module. It supplies the DM with a few incumbent candidates to score. Thereafter, the scored candidates found so far are used to form the training data, based on which a machine learning algorithm is applied to find an AVF that models the DM's preference information.
- *Preference elicitation module*: it aims at translating the preference information learned from the consultation module in the form that can be used in MOEA/D. In particular, the learned preference information is used to obtain a set of weight vectors biased towards the ROI.

In the remaining paragraphs, the technical detail of the progressive preference learning for MOEA/D will be described step by step in Sect. 2. Proof-of-principle experiments, shown in Sects. 3 and 4, demonstrate the effectiveness of our proposed algorithm for finding DM preferred Pareto-optimal solutions on benchmark problems with 3 to 10 objectives. At the end, Sect. 5 concludes this paper and provides some future directions.

2 Proposed Method

Generally speaking, the method proposed in this paper is a generic framework for progressive preference learning. It consists of three interdependent modules, i.e. consultation, preference elicitation and optimisation. For proof-of-principle purpose, this paper uses the state-of-the-art MOEA/D as the search engine in the optimisation module. It uses the preference information provided by the preference elicitation module to approximate DM's preferred solutions. In addition, it periodically supplies the consultation module with a few incumbent candidates to score. Since no modification has been done upon MOEA/D, we do not intend to delineate its working mechanism here while interested readers are suggested to refer to [22] for details. The following paragraphs will focus on describing the consultation and preference elicitation modules.

2.1 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 [2], 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.

There are two major questions to address when we want to approximate the DM's preference information: (1) which solutions can be used for scoring? and (2) how to learn an appropriate AVF?

Scoring. To relieve the DM's cognitive load and her/his fatigue, we only ask the DM to score a limited number (say $1 \leq \mu \ll N$) of incumbent candidates chosen from the current population. Specifically, we use the AVF learned from the most recent consultation session to score the current population. The μ solutions having the best AVF values are used as the incumbent candidates, i.e. deemed as the ones that are satisfied by the DM most. However, if it is at the first consultation session, no AVF is available for scoring. In this case, we first initialise another μ 'seed' weight vectors, which can either be generated by the Das and Dennis' method [6] or chosen from the weight vectors initialised in the optimisation module. Afterwards, for each of these 'seed' weight vectors, we find the nearest neighbour from the weight vectors initialised in the optimisation module. Then, the solutions associated with these selected weight vectors are used as the initial incumbent candidates.

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) [5] 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.

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 $\Phi : \mathbb{R}^m \rightarrow \mathbb{R}$. Various RBFs can be used as the activation function of the RBFN, such as Gaussian, splines and multiquadrics. In this paper, we consider the following Gaussian function:

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

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

$$\Phi(\mathbf{x}) = \omega^0 + \sum_{i=1}^{\mathrm{NR}} \omega^i \exp(-\frac{\|\mathbf{F}(\mathbf{x}) - \mathbf{c}^i\|}{\sigma^2}), \tag{3}$$

where NR is the number of RBFs, each of which is associated with a different centre $\mathbf{c}^i$, $i \in \{1, \cdots, \mathrm{NR}\}$. 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[1].

2.2 Preference Elicitation Module

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, this paper uses the Tchebycheff function [16–18] to form a subproblem as follows:

$$\begin{aligned} &\text{minimize } g(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) = \max_{1 \le i \le m} |f_i(\mathbf{x}) - z_i^*|/w_i \\ &\text{subject to} \quad \mathbf{x} \in \Omega \end{aligned} \tag{4}$$

where $\mathbf{z}^*$ is the ideal point and $\mathbf{w}$ is the weight vector associated with this subproblem. Since the optimal solution of each subproblem is a Pareto-optimal solution of the original MOP, MOEA/D can in principle approximate the whole PF with a necessary diversity by using a set of evenly distributed weight vectors $W = \{\mathbf{w}^i\}_{i=1}^N$, where N is the population size. When considering the DM's preference information, the ROI becomes a partial region of the PF. A natural

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

idea, which translates the DM's preference information into the form that can be used in MOEA/D, is to adjust the distribution of weight vectors. 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. weight vectors 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: Find the $\lceil\frac{N-\mu}{\mu}\rceil$ closest weight vectors to $\mathbf{w}^{Ui}$ according to their Euclidean distances.

Step 3.2: Move each of these weight vectors towards $\mathbf{w}^{Ui}$ according to

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

where $j \in \{1, \cdots, m\}$.

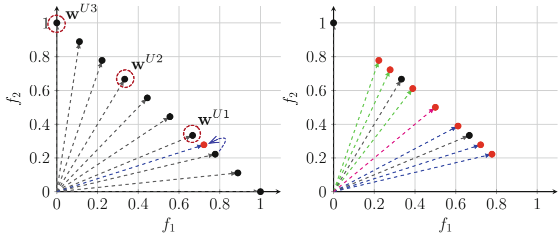
Step 3.3: Temporarily remove these weight vectors from W and go to Step 3.

Step 4: Output the adjusted weight vectors as the new W.

In the following paragraphs, we would like to make some remarks on some important ingredients of the above process.

- In MOEA/D, each solution should be associated with a weight vector. Therefore, in Step 2, the rank of a solution also indicates the importance of its associated weight vector with respect to the DM's preference information. The weight vectors stored in W^U are indexed according to the ranks of their associated solutions. In other words, $\mathbf{w}^{U1}$ represents the most important weight vector, and so on.
- Step 3 implements the adjustment of the distribution of weight vectors according to their satisfaction to the DM's preference information. Specifically, each of those μ promising weight vectors is used as a pivot, towards which its closest $\lceil\frac{N-\mu}{\mu}\rceil$ neighbours are moved according to Eq. 5.
- η in Eq. 5 controls the convergence rate towards the promising weight vector. For proof-of-principle purpose, we set $\eta = 0.5$ in this paper.
- Step 3 is similar to a clustering process, while we give the weight vector, which has a higher rank, a higher priority to attract its companions.

To better understand this preference elicitation process, Fig. 1 gives an intuitive example in a two-objective case. In particular, three promising weight vectors are highlighted by red circles. $\mathbf{w}^{U1}$ has the highest priority to attract its companions, and so on. We can observe that the weight vectors are biased towards those promising ones after the preference elicitation process.



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

Fig. 1. Illustration of the preference elicitation process.

3 Experimental Settings

To validate the effectiveness of our proposed algorithm, dubbed as I-MOEA/D-PLVF, for approximating the DM preferred solutions, the widely used DTLZ [11] test problems are chosen to form the benchmark suite. Note that the DTLZ problems are scalable to any number of objectives. The parameter settings of our proposed progressive preference learning paradigm are summarised 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 weight vectors, population size settings and number function evaluations (FEs) are set as suggested in [15]. Due to the page limit, they can be found in the supplementary document[2] of this paper.
- the simulated binary crossover [8] is used as the crossover operator while its probability and distribution index are set as: $p_c = 1.0$ and $\eta_c = 30$;
- the polynomial mutation [9] is used as the mutation operator while its probability and distribution index are set as: $p_m = \frac{1}{n}$ and $\eta_m = 20$;

As discussed in [14], 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 minimise the following nonlinear function:

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

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 towards a particular extreme. Since a m-objective problem has m extremes, there are

[2] https://coda-group.github.io/emo19-supp.pdf.

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 [15] 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 can be found in the supplementary document of this paper.

To evaluate the performance of I-MOEA/D-PLVF 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 with its 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.

4 Empirical Results

From the results shown in Table 1, as we expected, I-MOEA/D-PLVF shows overwhelming superiority over the baseline MOEA/D for approximating the DM's

Table 1. Performance comparisons of the approximation errors (median and the corresponding IQR) obtained by I-MOEA/D-PLVF versus the baseline MOEA/D on DTLZ1 to DTLZ4 test problems.

		DTLZ1		DTLZ2	
m	ROI	I-MOEA/D-PLVF	MOEA/D	I-MOEA/D-PLVF	MOEA/D
3	c	**4.213E-4(2.87E-3)**	3.104E-2(3.18E-3)	**1.026E-2(1.78E-2)**	1.030E-1(6.35E-3)
	b	**1.471E-3(2.87E-3)**	3.103E-2(3.30E-3)	**8.832E-3(1.09E-2)**	9.103E-2(2.56E-3)
5	c	**4.173E-3(1.73E-2)**	5.262E-2(1.90E-2)	**1.721E-2(2.86E-2)**	2.417E-1(1.90E-2)
	b	**1.082E-2(2.09E-2)**	7.648E-2(1.65E-2)	**5.082E-2(4.73E-2)**	2.049E-1(1.45E-2)
8	c	**2.130E-3(1.71E-2)**	1.484E-2(2.21E-3)	**1.625E-2(1.79E-1)**	2.615E-1(1.52E-2)
	b	**1.012E-2(1.03E-1)**	5.534E-2(1.12E-2)	**4.185E-2(1.10E-1)**	1.250E-1(1.05E-2)
10	c	**1.269E-1(2.71E-1)**	1.789E-1(1.10E-3)	**1.087E-1(1.62E-1)**	7.386E-1(8.54E-2)
	b	**1.543E-1(1.77E-1)**	2.634E-1(5.05E-3)	**1.183E-1(2.08E-1)**	2.596E-1(2.88E-2)
		DTLZ3		DTLZ4	
m	ROI	I-MOEA/D-PLVF	MOEA/D	I-MOEA/D-PLVF	MOEA/D
3	c	**7.214E-4(7.26E-3)**	1.055E-1(1.59E-3)	**1.303E-2(2.78E-2)**	1.042E-1(1.89E-3)
	b	**2.811E-3(1.09E-2)**	8.678E-2(7.75E-3)	**7.634E-3(8.76E-3)**	9.469E-2(8.07E-3)
5	c	**1.128E-2(8.77E-2)**	2.442E-1(4.62E-2)	**2.762E-2(5.74E-2)**	2.569E-1(2.37E-3)
	b	**1.792E-2(1.53E-1)**	2.162E-1(2.35E-2)	**3.717E-2(6.28E-2)**	2.121E-1(6.66E-3)
8	c	**6.821E-2(2.78E-1)**	4.277E-1(9.56E-3)	**6.538E-2(8.62E-2)**	7.236E-1(1.07E-2)
	b	**8.697E-2(1.63E-1)**	1.574E-1(1.32E-2)	**1.271E-1(1.86E-1)**	2.164E-1(1.69E-2)
10	c	**2.168E-1(5.71E-1)**	7.365E-1(2.81E-2)	**1.927E-1(2.63E-1)**	8.676E-1(1.07E-1)
	b	**1.629E-1(2.55E-1)**	3.344E-1(6.99E-2)	**1.018E-1(3.28E-1)**	2.055E-1(4.21E-2)

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 grey background.

'golden' 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 the following aspects.

– Due to the page limit, we only plot some results on 3- and 10-objective scenarios in Figs. 2, 3, 4 and 5, while more comprehensive results can be found in the supplementary document. From these plots, we can observe that I-MOEA/D-PLVF is always able to find solutions that well approximate the unknown DM's 'golden' point with a decent accuracy as shown in Table 1. In contrast, since the baseline MOEA/D is 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 can by chance pass the ROI, i.e. the vicinity of the DM's 'golden'

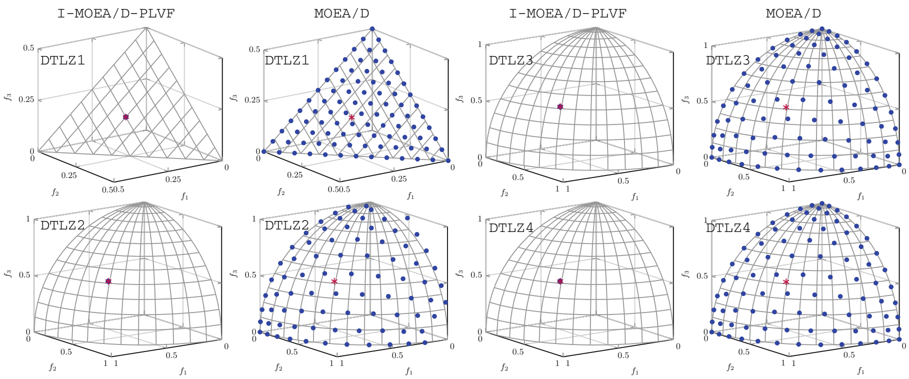


Fig. 2. Solutions obtained on 3-objective DTLZ1 to DTLZ4 problems where $\mathbf{z}^r$, which prefers the middle region of the PF, is represented as the red dotted line. (Color figure online)

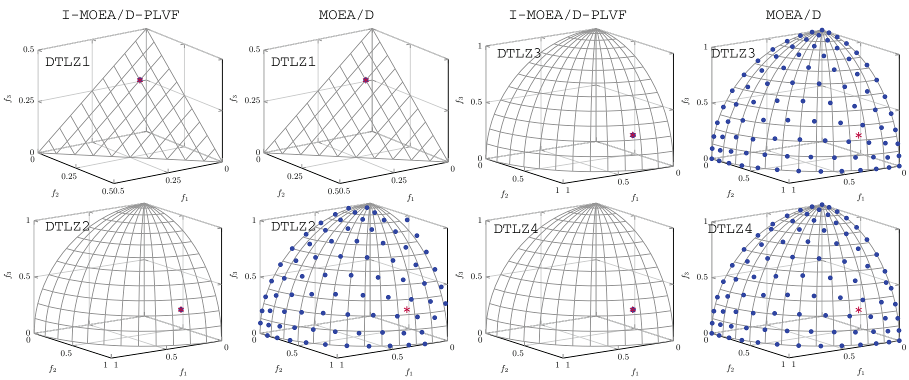


Fig. 3. Solutions obtained on 10-objective DTLZ1 to DTLZ4 problems where $\mathbf{z}^r$, which prefers one side of the PF, is represented as the red dotted line. (Color figure online)

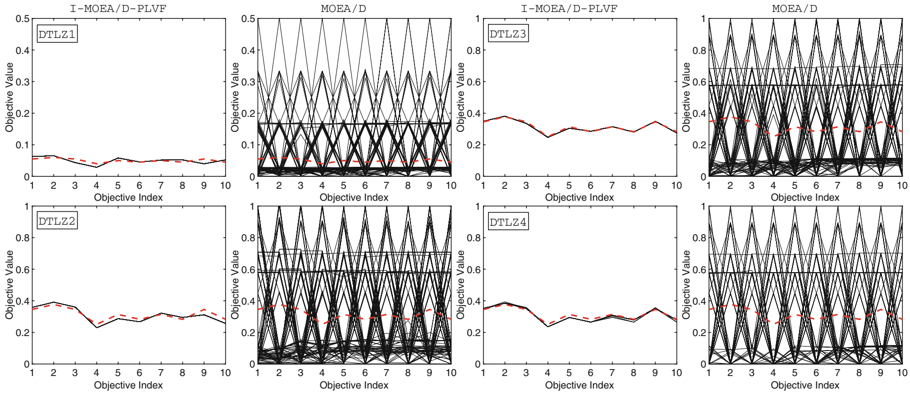


Fig. 4. Solutions obtained on 10-objective DTLZ1 to DTLZ4 problems where $\mathbf{z}^r$, which prefers the middle of the PF, is represented as the red dotted line. (Color figure online)

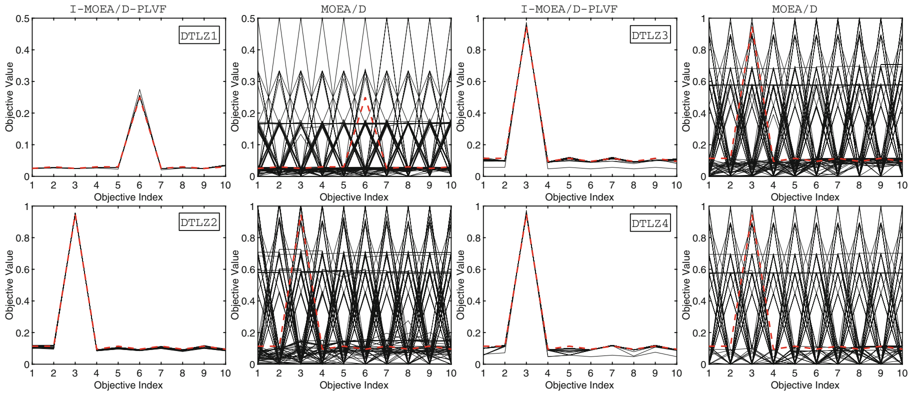


Fig. 5. Solutions obtained on 10-objective DTLZ1 to DTLZ4 problems where $\mathbf{z}^r$, which prefers one side of the PF, is represented as the red dotted line. (Color figure online)

point, they still have an 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, e.g. as shown in Figs. 4 and 5.

– From the results shown in Table 1, we find that it seems to be more difficult for the baseline MOEA/D to find the DM's preferred solution on the middle region of the PF than 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, 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 I-MOEA/D-PLVF can progressively learn the DM's preference information and adjust the search direction, it well approximates the ROI in any part of the PF.

5 Conclusions

This paper has proposed a simple yet effective paradigm for progressively learning the DM's preference information in an interactive manner. It consists of three modules, i.e. optimisation, consultation and preference elicitation. For proof-of-principle purpose, this paper uses the state-of-the-art MOEA/D as the baseline algorithm in the optimisation module. The consultation module aims to progressively learn an AVF that models 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 the optimisation module, i.e. a set of weight vectors that are biased towards the ROI. Proof-of-principle results on 3- to 10-objective test problems demonstrate the effectiveness of our proposed I-MOEA/D-PLVF for approximating the DM's preferred solution(s).

In principle, the progressive preference learning paradigm proposed in this paper is a generic framework which can be used to help any EMO algorithm to approximate DM preferred solution(s) in an interactive manner. For proof-of-principle purpose, we use MOEA/D as the search engine in the optimisation module. Therefore, the learned preference information is translated as a set of biased weight vectors in the preference elicitation module. One of the future directions is to adapt this to other formats according to the characteristics of the baseline algorithm. In addition, this paper assumes 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 candidates. Although this paper has restricted the value function to be the form as Eq. 6, other more value function formulations 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. Last but not the least, there are a couple of parameters associated with the proposed progressive preference learning paradigm, i.e. those listed in Sect. 3. It is important to investigate the effects of these parameters as a part of future work.

References

1. Auger, A., Bader, J., Brockhoff, D., Zitzler, E.: Hypervolume-based multiobjective optimization: theoretical foundations and practical implications. Theor. Comput. Sci. **425**, 75–103 (2012)
2. Battiti, R., Passerini, A.: Brain-computer evolutionary multiobjective optimization: a genetic algorithm adapting to the decision maker. IEEE Trans. Evol. Comput. **14**(5), 671–687 (2010)
3. Bechikh, S., Kessentini, M., Said, L.B., Ghédira, K.: Chapter four - preference incorporation in evolutionary multiobjective optimization: a survey of the state-of-the-art. Adv. Comput. **98**, 141–207 (2015)
4. Branke, J., Deb, K.: Integrating user preferences into evolutionary multi-objective optimization. In: Jin, Y. (ed.) Knowledge Incorporation in Evolutionary Computation, vol. 167, pp. 461–477. Springer, Berlin (2005). https://doi.org/10.1007/978-3-540-44511-1_21
5. Buhmann, M.D.: Radial Basis Functions. Cambridge University Press, Cambridge (2003)
6. Das, I., Dennis, J.E.: Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems. SIAM J. Optim. **8**, 631–657 (1998)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
8. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Syst. **9**, 1–34 (1994)
9. Deb, K., Goyal, M.: A combined genetic adaptive search (GeneAS) for engineering design. Comput. Sci. Inform. **26**, 30–45 (1996)
10. Deb, K., Sundar, J., Bhaskara, U., Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithms. Int. J. Comput. Intell. Res. **2**(3), 273–286 (2006)
11. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization. Advanced Information and Knowledge Processing, pp. 105–145. Springer, London (2005). https://doi.org/10.1007/1-84628-137-7_6
12. Jin, Y., Okabe, T., Sendho, B.: Adapting weighted aggregation for multiobjective evolution strategies. In: Zitzler, E., Thiele, L., Deb, K., Coello Coello, C.A., Corne, D. (eds.) EMO 2001. LNCS, vol. 1993, pp. 96–110. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44719-9_7
13. Li, K., Chen, R., Min, G., Yao, X.: Integration of preferences in decomposition multiobjective optimization. IEEE Trans. Cybern. **48**(12), 3359–3370 (2018)
14. Li, K., Deb, K., Yao, X.: R-metric: evaluating the performance of preference-based evolutionary multi-objective optimization using reference points. IEEE Trans. Evol. Comput. (2017). accepted for publication
15. Li, K., Deb, K., Zhang, Q., Kwong, S.: An evolutionary many-objective optimization algorithm based on dominance and decomposition. IEEE Trans. Evol. Comput. **19**(5), 694–716 (2015)
16. Li, K., Fialho, Á., Kwong, S., Zhang, Q.: Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **18**(1), 114–130 (2014)

17. Li, K., Kwong, S., Zhang, Q., Deb, K.: Interrelationship-based selection for decomposition multiobjective optimization. IEEE Trans. Cybern. **45**(10), 2076–2088 (2015)
18. Li, K., Zhang, Q., Kwong, S., Li, M., Wang, R.: Stable matching based selection in evolutionary multiobjective optimization. IEEE Trans. Evol. Comput. **18**(6), 909–923 (2014)
19. Parmee, I.C., Cvetkovic, D.: Preferences and their application in evolutionary multiobjective optimization. IEEE Trans. Evol. Comput. **6**(1), 42–57 (2002)
20. Said, L.B., Bechikh, S., Ghédira, K.: The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making. IEEE Trans. Evolut. Comput. **14**(5), 801–818 (2010)
21. Wagner, T., Trautmann, H., Brockhoff, D.: Preference articulation by means of the $R2$ indicator. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) EMO 2013. LNCS, vol. 7811, pp. 81–95. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37140-0_10
22. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**, 712–731 (2007)
23. Zitzler, E., Künzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., et al. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 832–842. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30217-9_84