Self-organizing Map-based Weight Design for Decomposition-based Many-objective Evolutionary Algorithm

Fangqing Gu and Yiu-ming Cheung

***Abstract*—Many-objective optimization problems, in which the number of objectives is greater than three, are undoubtedly more challenging compared with the bi- and tri-objective optimization problems. Currently, the decomposition-based evolutionary algorithms have shown promising performance in dealing with many-objective optimization problems. Nevertheless, these algorithms need to design the weight vectors, which has significant effects on the performance of the algorithms. In particular, when the Pareto front of problems is incomplete, these algorithms cannot obtain a set of uniformly distribution solutions by using the conventional weight design methods. In the literature, it is well-known that the self-organizing map can preserve the topological properties of the input data by using the neighborhood function, and its display is more uniform than the probability density of the input data. This phenomenon is advantageous to generate a set of uniformly distributed weight vectors based on the distribution of the individuals. Therefore, we will propose a novel weight design method based on self-organizing map, which can be integrated with most of the decomposition-based algorithms for solving many-objective optimization problems. In this paper, we choose the existing state-of-the-art decomposition-based algorithms as examples for such integration. This integrated algorithms are then compared with some state-of-the-art algorithms on eleven redundancy problems and eight non-redundancy problems, respectively. The experimental results show the effectiveness of the proposed approach.**

***Index Terms*—Many-objective Optimization, Evolutionary Algorithm, Self-organizing Map, Weight Design.**

I. Introduction

Without loss of generality, a multiobjective optimization problem (MOP) can be formulated as follows:

$$\min_{\mathbf{x}\in D} F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x}))^T \quad (1)$$

where $D \subset \mathbb{R}^n$ is the domain, $\mathbf{x} \in D$ is the decision variable, n is the dimension of the decision variable and $M \geq 2$ is the number of the objectives. For many-objective optimization problems (MaOPs), M is generally greater than three.

Fangqing Gu is with the Guangdong University of Technology, Guangdong, China. E-mail: fqgu@gdut.edu.cn

Yiu-ming Cheung is with the Department of Computer Science, Hong Kong Baptist University (HKBU), Hong Kong, SAR, China. Yiu-ming Cheung (corresponding author) is also with the Institute of Research and Continuing Education (IRACE) in HKBU and United International College, Beijing Normal University-HKBU, Zhuhai, China. E-mail: ymc@comp.hkbu.edu.hk

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants: 61672444 and 61272366, and the Faculty Research Grant of Hong Kong Baptist University (HKBU) under Projects: FRG2/16-17/051 and FRG2/15-16/049.

MaOPs are quite common in a variety of real applications [1], [2], [3]. In the literature, a number of multiobjective evolutionary algorithms (MOEAs), e.g. Pareto-based methods NSGA-II [4], SPEA2 [5], indicator-based approaches HypE [6], SMS-EMOA [7], and decomposition-based approaches like MOEA/D [8] and M2M [9], [10], [11], are applicable to solve MaOPs. Specifically, most of the solutions of MaOPs [12] in the population are nondominated to the Pareto-based methods. As a result, the Pareto-based methods may fail to converge to the Pareto front (PF) because the algorithms lose the selection pressure towards the PF. In view of this, some modified Pareto dominance algorithms [13], [14], [15], [16], [17] have been presented to increase the selection pressure towards the PF by decreasing the number of the nondominated solutions in the population. Such algorithms can solve the MaOPs to a certain extent. By contrast, it has been shown that the indicator-based approaches achieved competitive performance in terms of the average hypervolume for solving MaOPs [6]. However, the indicator-based algorithms suffer from the high computational cost. Actually, the complexity of the indicator-based algorithms grows exponentially with the number of the objectives [18], which seriously limits its applications from a practical perspective. Recently, the decomposition-based algorithms, in which each subproblem is associated with a search direction (i.e. weight vector) or target (i.e. reference point), have already shown its success in solving MOPs. Actually, such an algorithm also provides a promising way for solving MaOPs [19] as it can provide sufficient selection pressure towards the PF. Hereinafter, we will concentrate on studying the MaOPs within the framework of decomposition-based approaches.

As one of the most well-known decomposition-based algorithms, MOEA/D [12], [20] aggregates the objectives into a single objective by utilizing a weight vector and achieves good performance on most of MaOPs. Furthermore, Deb and Jain have presented a decomposition-based algorithm, namely NSGA-III [21], [22], which utilizes a set of uniformly distributed reference points to guide the search process. Empirical studies have shown that it outperforms MOEA/D when tackling unconstrained and constrained MaOPs. Among such algorithms, all need to design the weight vectors/reference points. In fact, MOEA/D [8], M2M [9] and MOEA/DD [23] all employ a set of predefined uniformly distributed weight vectors, while NSGA-III [21], [22] utilizes a set of predefined uniformly distributed reference points, analogous to MOEA/D. The experimental results in [24], [25] have demonstrated

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2695579, IEEE Transactions on Evolutionary Computation

that the performance of decomposition-based many-objective algorithms strongly depends on PF shapes. When the shape of the PF is close to the hyper-plane $\sum_{i=1}^{M} f_i = 1$ and complete, i.e. the projection of the PF on normal hyper-plane $\sum_{i=1}^{M} f_i = 1$ fills the whole region of the first octant of the hyper-plane $\sum_{i=1}^{M} f_i = 1$, these algorithms with uniformly weight vectors can obtain a set of uniformly distributed Pareto optimal solutions. However, this condition may not be satisfied in MaOPs, as well as MOPs. When the PF is incomplete, there are more than one scalar problem with different weight vectors leading to the same Pareto optimal solution as stated in Section III-A, which seriously deteriorates the algorithms' performance. Thus, how to design the weight vectors in a decomposition-based evolutionary algorithm is a key issue for solving an MaOP with incomplete PF.

A weight design method based on the normal-boundary insertion method [26] has been widely used to generate a restricted number of weight vectors. The number of the weight vectors is C_{H+M-1}^{M-1}, where H is the number of divisions along each objective [27]. The number of its weight vectors dramatically increases over the number of objectives. Undoubtedly, how to deal with the dilemma between limited computational resources and the dramatically increasing number of weight vectors is a challenging problem. To remedy this, some adaptive weight design approaches have been proposed [28]. Moreover, in [29], [30], an approximate PF is periodically estimated based on the distribution of current nondominated solutions. Then, the new weight vectors are uniformly created on the approximate PF. In [31], [28], the weights are periodically adjusted so that the solutions of the subproblems are far away each other. In addition, another weight adjustment method has been proposed by sampling the regression curve of objective vectors of the solutions [32]. To sum up, these weight design methods are essentially proposed for MOPs only. To the best of our knowledge, few studies have been conducted on weight design for MaOPs.

Self-organizing map (SOM) [33], [34] and its variants [35], [36], which use a neighborhood function to preserve the topological properties of the data, are one of the most popular neural networks and have been widely used in many applications [37], e.g. data clustering [38], [39], data visualization [40], image segmentation [41], and so forth. Thus far, some studies have been conducted for solving MOPs involving SOM. For example, Zhang et al. [42], [43] proposed self-organizing multiobjective evolutionary algorithm based on decomposition, in which the SOM approach is employed to discover the population distribution structure in decision space to guide the search. Also, a strategy based on SOM has been developed for the visualization of the Pareto optimal solutions in [44]. However, there are few works about employing the SOM to discover the distribution of the solution in the objective space for weight design of decomposition-based MOEAs. As shown in [45], the point density of the SOM model is usually proportional to the probability density function of the input data, but not linearly. It is flatter actually. This means that the display of SOM is more uniform than it would be if it represented the exact probability density of the input data. This phenomenon is advantageous to generate a set of uniformly distributed weight vectors based on the distribution of the individuals. Accordingly, this paper will develop a novel weight design method based on SOM for decomposition-based MOEAs. We periodically train an SOM network with N neurons by using the objective vectors of the recent individuals, where N is the population size. The weights of the neurons are employed as the weight vectors. The dimension of the weight of neurons is equal to the dimension of the objective vectors. Evidently, this weight design method can be applied to the optimization problems involving an arbitrary number of objectives. Thus, this weight design method can be applied into most of the decomposition-based MOEAs, e.g. MOEA/D, M2M, MOEA/DD, NSGA-III, and so forth. In this paper, we integrate the proposed weight design method into M2M and MOEA/D, named as MOEA/D-SOM and M2M-SOM, respectively. We compare them with the original M2M [9] and MOEA/D [8], and two latest algorithms, i.e. NSGA-III [21] and MOEA/DD [23], on benchmark DTLZ5 [46] and TOY problems [47] with redundancy objectives, and eight test problems proposed in [48] with non-redundancy objectives, respectively. Experiments have shown the promising results of the proposed algorithm.

The remainder of this paper is organized as follows: We give an overview of the Tchebycheff decomposition approach and the SOM in Section II. Section III presents the SOM-based weight design method and integrates it into M2M and MOEA/D, respectively, for solving MaOPs. Section IV provides the experimental results, as well as some discussions, to compare the proposed MOEA/D-SOM and M2M-SOM with the original MOEA/D and M2M, and two latest existing counterparts, i.e. NSGA-III, MOEA/DD, on eleven redundancy problems and eight non-redundancy problems, respectively. Finally, we draw a conclusion in Section V.

II. An Overview of Tchebycheff Decomposition Approach and Self-organizing Map

A. Tchebycheff Decomposition Approach

Many aggregation objectives methods have been developed to decompose a multiobjective problem into a number of scalar optimization problems [49]. Among these decomposition approaches, the Tchebycheff approach is probably the most important one. Under the Tchebycheff approach, problem in (1) can be solved by the set of scalar problems with different weight vectors:

$$\min_{\mathbf{x}\in D} g(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = \min_{\mathbf{x}\in D} \max_{1\le i\le M} \left\{ \frac{f_i(\mathbf{x}) - z_i^*}{w_i} \right\}^{1} \qquad (2)$$

where $\mathbf{w} = (w_1, w_2, \ldots, w_M)^T \in \mathbb{R}_M^+$ is a weight vector and $\mathbf{z}^* = (z_1^*, z_2^*, \ldots, z_M^*)^T$ is the ideal point, i.e. $z_i^* = \min_{\mathbf{x}\in D} f_i(\mathbf{x})$ for $i = 1, 2, \ldots, M$. It has been proved in [50] that, for each Pareto optimal solution $\mathbf{x}^*$ of (1), there exists at least one weight vector $\mathbf{w}$ such that $\mathbf{x}^*$ is the optimal solution of (2) and each optimal solution of (2) is a weak Pareto optimal solution of (1).

[1]Generally, it is formulated as $\min_{\mathbf{x}\in D} \max_{1\le i\le M} \{w_i(f_i(\mathbf{x}) - z_i^*)\}$. That is, we express the weight vector as the reciprocal of the convention weight wector in this paper.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2695579, IEEE Transactions on Evolutionary Computation

In general, as claimed in [29], [28], if the straight line $\frac{f_1'}{w_1} = \frac{f_2'}{w_2} = \ldots = \frac{f_M'}{w_M}$, taking $f_1', f_2', \ldots, f_M'$ as variables, intersects with the PF, the intersection point is the optimal solution of the scalar problem with the weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_M)^T$, where $f_i^{'} = f_i - z_i^*$ is the normalized function for $i = 1, \ldots, M$. In [29], we firstly presented the weight vectors as the coordinate of the points in R_M^+, i.e. $\mathbf{w} = (f_1', f_2', \ldots, f_M')^T \in R_M^+$, and then systematized them using the method in [28] for Tchebycheff approach. By this manner, if the straight line goes through the point $(f_1', f_2', \ldots, f_M')^T$ and ideal point $\mathbf{z}^*$ intersects with the PF, the intersection point is the optimal solution of the scalar problem with $\mathbf{w}$.

B. Self-organizing Map

SOM uses a neighborhood function to preserve the topological properties of the data. In the following, we will provide a brief description of SOM. Suppose a network contains N neurons and each neuron j is associated with a weight $\mathbf{v}_j$. Let $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$ be a set of the weight of the neurons, and $\mathbb{F}' = \{\mathbf{F}'_1, \mathbf{F}'_2, \ldots, \mathbf{F}'_{N'}\}$ be the input data set, where $N' > N$ is size of the input data. It is noteworthy that the dimension of the neurons' weight is equal to the dimension of the problem.

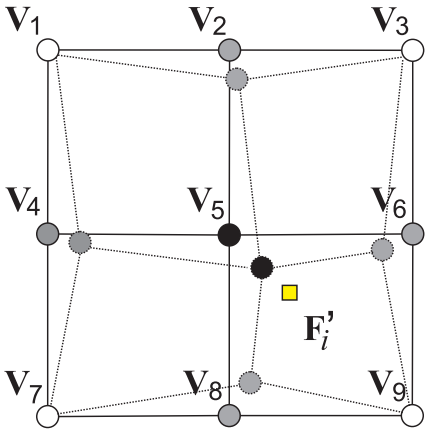


Fig. 1. An illustration of a 2-dimensional SOM.

Fig. 1 shows an illustration of a 2-dimensional SOM which consists of 9 neurons. At each iteration t, the SOM training consists of two stages, i.e. the competitive stage and the cooperative stage, as follows:

1) Competitive stage: For each $\mathbf{F}'_i \in \mathbb{F}'$, we determine the winner neuron which is the neuron with the smallest Euclidean distance to $\mathbf{F}'_i$, i.e.

$$c = arg\min_j \|\mathbf{F}'_i - \mathbf{v}_j(t)\| \tag{3}$$

where c is the index of the "winner" of $\mathbf{F}'_i$. As shown in Fig. 1, $\mathbf{v}_5$ with black is the closest to the input data $\mathbf{F}'_i$. This means that $\mathbf{v}_5$ is the winner of $\mathbf{F}'_i$.

2) Cooperative stage: In the training procedure, not only is the weight of the winning neuron updated, but also those of its neighboring neurons are. Then, the learning rule can be expressed as

$$\mathbf{v}_j(t+1) = \mathbf{v}_j(t) + \alpha_t h_{cj}(t) \left[\mathbf{F}'_i - \mathbf{v}_j(t)\right] \tag{4}$$

where t denotes the current learning iteration and $0 < \alpha_t < 1$ is the "learning rate". $h_{cj}(t)$ is the neighborhood kernel centered on the winner. In this paper, we use the "Gaussian" neighborhood kernel, i.e.

$$h_{cj}(t) = \begin{cases} exp(-d_{cj}^2/2\sigma_t^2) \text{ if } j \in U_c \\ 0 \quad otherwise \end{cases} \tag{5}$$

where U_c is the neighbors of the winner c, and the rectangular topology is used in this paper. Therefore, neurons $\mathbf{v}_2, \mathbf{v}_4, \mathbf{v}_6$ and $\mathbf{v}_8$ with grey are the neighbors of $\mathbf{v}_5$ as shown in Fig 1. d_{cj} is the distance between unit c and j, and σ_t denotes the width of the neighborhood radius. Both α_t and σ_t are some monotonically decreasing functions of time. As shown in Fig. 1, the winner and its neighbors move towards the input data, and the points with hashed outline are the updated winner and its neighbors.

III. The Proposed SOM-based Weight Design

A. A Principle of Weight Design

1) PF Representation: The weight vectors have to present the shape and distribution of the PF of the problems. Currently, the decomposition-based MOEAs usually utilize a set of predefined weight vectors $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N\}$, which are generated by a systematic approach developed in [26], and uniformly distributed on the hyper-plane $\sum_{i=1}^M f_i = 1$. According to the analysis of decomposition approach, the uniform weight vectors can lead to a set of uniformly distributed Pareto optimal solutions over the PF when the PF of the problems satisfies the following two conditions:

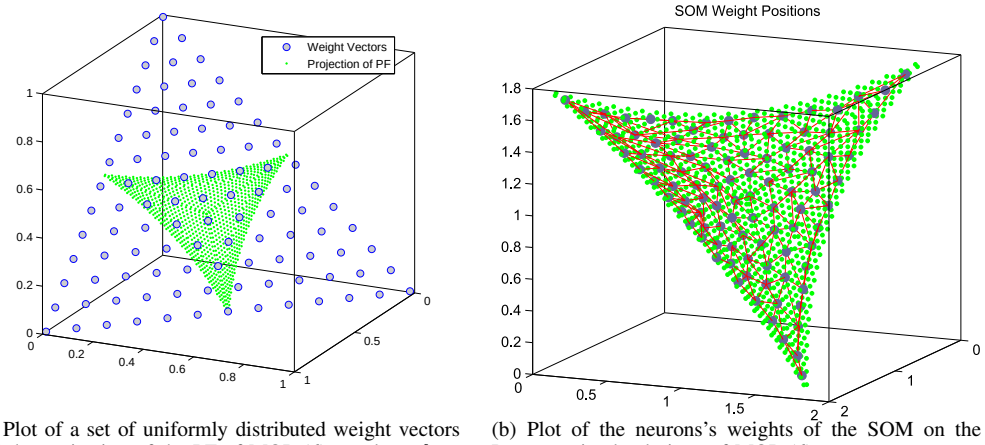
a) The shape of the PF is close to the hyper-plane $\sum_{i=1}^M f_i = 1$ as claimed in [29], [28];
b) The PF is complete, i.e. the projection of the PF on the hyper-plane $\sum_{i=1}^M f_i = 1$ fills the whole region of the first octant of the hyper-plane $\sum_{i=1}^M f_i = 1$.

However, the above conditions may not be satisfied in many MOPs, i.e. the PF of an MOP may be incomplete. In other words, there may exist some straight lines that do not intersect with the PF. This means that many scalar problems with different weight vectors may have the same Pareto optimal solution. For such problems, the algorithm with the predefined uniformly weight vectors would not obtain a set of uniformly distributed Pareto optimal solutions. As shown in Fig. 2(a), it plots the weight vectors which are uniformly distributed on the plane $f_1 + f_2 + f_3 = 1$ and the projection of the PF of an MOP as introduced in [51], [52], whose objectives are the distances to three points. That is,

$$\begin{cases} f_1(\mathbf{x}) = \|\mathbf{x} - \mathbf{p}_1\|_2 \\ f_2(\mathbf{x}) = \|\mathbf{x} - \mathbf{p}_2\|_2 \\ f_3(\mathbf{x}) = \|\mathbf{x} - \mathbf{p}_3\|_2 \end{cases} \tag{6}$$

where $\mathbf{p}_1 = (0, 1)^T$, $\mathbf{p}_2 = (-1/2, \sqrt{3}/2)^T$ and $\mathbf{p}_3 = (-1/2, -\sqrt{3}/2)^T$. It is clear that there are many lines joining the origin and the weight vectors (shown with open circles)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2695579, IEEE Transactions on Evolutionary Computation



(a) Plot of a set of uniformly distributed weight vectors and the projection of the PF of MOP (6) on plane $f_1 + f_2 + f_3 = 1$.

(b) Plot of the neurons's weights of the SOM on the Pareto optimal solutions of MOP (6).

Fig. 2. The uniform weight vectors and the weight vectors obtained by SOM algorithm.

do not intersect with the PF. The optimal solution of the scalar optimization problems with the weight vector, which does not intersect with the PF, locates the boundary of the PF. Therefore, it cannot obtain a set of uniformly distributed Pareto optimal solutions. In particular, this problem will become more prominent if there are some objectives that are correlated or redundant for an MaOP. Undoubtedly, an efficient weight design method will become more meaningful for MaOPs. Thus far, several adaptive weight design methods have been presented for bi- and tri-objective optimization problems to generate a set of representative weight vectors [28], [29], but few works on weight design have been done for MaOPs.

2) Uniformity: A set of uniformly distributed weight vectors generally lead to a set of uniform solutions by using the decomposition-based MOEAs. Thus, the weight vectors should be as uniform as possible. To this end, there are several machine learning methods, such as k-means and SOM, can be used to discover the distribution of the input data with an arbitrary dimension. Nevertheless, the k-means algorithm tends to approximate the density of the input data, while the display of SOM is more uniform than the probability density of the input data [45]. Hence, SOM is a promising tool for designing the weight vectors. Accordingly, we will present a novel weight design method based on SOM for decomposition-based many-objective evolutionary algorithms in the following sub-section.

B. The Proposed Weight Design Method Based on SOM

To avoid overfitting the input data, the size of the input data N' must be greater than N. Thus, an extra matrix $\mathbb{F} = [\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_{N'}]$ is introduced to save the objective vectors of some individuals for training the SOM network. Also, the normalized objective vectors $\mathbb{F}' = [\mathbf{F}'_1, \mathbf{F}'_2, \ldots, \mathbf{F}'_{N'}]$ are treated as the SOM's input data, where $\mathbf{F}'_i = \mathbf{F}_i - \mathbf{z}^*$ for $i = 1, 2, \ldots, N'$. Then, we can obtain N weight $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$ of the neurons by training the SOM network.

For clarity, Fig. 2(b) plots the weights of the neurons obtained by SOM algorithm for MOP (6), in which we uniformly sample 1000 points on the PF of problem (6) and then use them to train the SOM network. From this figure, we can see that the weights of the neurons can well preserve the distribution and shape of the PF. Furthermore, we can find that the region covered by the neurons' weights is slightly smaller than the PF. If we directly utilize the neurons' weights as the weight vectors in (2), it is unfavorable for obtaining the Pareto solution on the boundary of PF. Therefore, we set the weight vector as

$$\mathbf{w}_i = \mathbf{v}_i - \mathbf{v}^*, \tag{7}$$

for $i = 1, 2, \ldots, N$, where $\mathbf{v}^* = (v_1^*, v_2^*, \ldots, v_M^*)^T$ with $v_j^* = \min_{1 \le i \le N} v_{ij}$ and v_{ij} is the jth element of $\mathbf{v}_i$ for $j = 1, 2, \ldots, M$.

It is worth noting that such weight design method can be applied for the problems with an arbitrary number of objectives and without any adjustable parameters. Moreover, it is easy to be integrated into most of the decomposition-based MOEAs. In the following, we will integrate it into M2M and MOEA/D, respectively, as two examples for implementation.

C. M2M with SOM-based Weight Vectors

We have proposed a decomposition-based multiobjective evolutionary (M2M) [9], which decomposes problem (1) into a set of simple multiobjective optimization subproblems and then solves them simultaneously. It achieves a good performance for solving MOPs. In this paper, we will propose an improved M2M with SOM-based weight design, i.e. M2M-SOM, for solving MaOPs.

Suppose the objective space is divided into K subregions, and K direction vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K\}$ are randomly and distinctly selected from $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N\}$. $\mathbf{u}_k$ is served as the center of the kth subregion, where $k = 1, \ldots, K$. Then, $\mathbb{R}^+_M$ is divided into K disjoint subregions, denoted as $\Omega_1, \Omega_2, \ldots, \Omega_k, \ldots, \Omega_K$, with

$$\Omega_k = \{\mathbf{y} \in R^+_M | \langle \mathbf{y}, \mathbf{u}_k \rangle \ge \langle \mathbf{y}, \mathbf{u}_j \rangle \text{ for any } j = 1, \ldots, K\}, \tag{8}$$

where $\langle \mathbf{y}, \mathbf{u}_k \rangle = \frac{\mathbf{y}^T \mathbf{u}_k}{\|\mathbf{y}\| \|\mathbf{u}_k\|}$ is the cosine of the acute angle between $\mathbf{y}$ and $\mathbf{u}_k$. Based on this division, problem (1) can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2695579, IEEE Transactions on Evolutionary Computation

be decomposed into K subproblems, where subproblem k is expressed as

$$\begin{aligned} &\min_{\mathbf{x}\in D} F(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_M(\mathbf{x}))^T, \\ &\quad F(\mathbf{x}) \in \Omega_k. \end{aligned} \tag{9}$$

Algorithm 1: The procedure of the proposed M2M-SOM

input :
- A stopping criterion;
- N: the size of the population;
- K: the number of the subpopulations.

output: The nondominated solutions in $\mathbf{P}_t$.

Step 1 Initialization
$\mathbf{P}_0 \leftarrow$ Randomly initialize $5N$ initial individuals;
$\mathbb{F} \leftarrow \mathbf{P}_0.F$; /*$\mathbf{P}_0.F$: objective vectors of $\mathbf{P}_0$*/
$\mathbf{z}^* \leftarrow SetIdealPoint()$;
$\{\mathbf{w}_1, \ldots, \mathbf{w}_N\} \leftarrow SOM(\mathbb{F}, \mathbf{z}^*)$;
$\{\mathbf{u}_1, \ldots, \mathbf{u}_K\} \leftarrow SetCenter()$;
$I_k \leftarrow SetSubpopulation(\mathbf{P}_0, \mathbf{u}_k)$, $k = 1, \ldots, K$;
$\mathbf{P}_0 \leftarrow \cup_{k=1}^{K} I_k$; /*Reset population*/
$\mathbb{F} \leftarrow \mathbf{P}_0.F$;
$t \leftarrow 0$.
Step 2 Update:
$\mathbf{Q}_t \leftarrow CreateOffspringPopulation(\mathbf{P}_t)$;
$\mathbf{z}^* \leftarrow UpdateIdealPoint()$
/*Update the subpopulations*/
$\mathbf{R}_t \leftarrow \mathbf{P}_t \cup \mathbf{Q}_t$;
$I_k \leftarrow SetSubpopulation(\mathbf{R}_t, \mathbf{u}_k)$, $k = 1, \ldots, K$;
$\mathbf{P}_{t+1} \leftarrow \cup_{k=1}^{K} I_k$;
$\mathbb{F} \leftarrow \mathbb{F} \cup (\mathbf{P}_{t+1} \cap \mathbf{Q}_t).F$;
$t \leftarrow t + 1$.
Step 3 Weight vectors update
if $|\mathbb{F}| > 5N$ **then**
  $\{\mathbf{w}_1, \ldots, \mathbf{w}_N\} \leftarrow SOM(\mathbb{F}, \mathbf{z}^*)$;
  $\{\mathbf{u}_1, \ldots, \mathbf{u}_K\} \leftarrow SetCenter()$;
  $\mathbb{F} \leftarrow \mathbf{P}_t.F$;
end
Step 4 Stopping Criteria
If the stopping criteria is satisfied, then stop and find all the nondominated solutions in $\mathbf{P}_t$ and output them. Otherwise, goto **Step 2**.

Algorithm 2: The procedure of the proposed MOEA/D-SOM

input :
- A stopping criterion;
- N: the size of the population;
- T : the number of neighbors for each subproblem.

output: The nondominated solutions in $\mathbf{P}_t$.

Step 1 Initialization
$\mathbf{P}_0 \leftarrow$ Randomly initialize $5N$ initial individuals;
$\mathbb{F} \leftarrow \mathbf{P}_0.F$; /*$\mathbf{P}_0.F$: objective vectors of $\mathbf{P}_0$*/
$\mathbf{z}^* \leftarrow SetIdealPoint()$;
$\{\mathbf{w}_1, \ldots, \mathbf{w}_N\} \leftarrow SOM(\mathbb{F}, \mathbf{z}^*)$;
$B(i) \leftarrow SetNeighbor()$;
$\mathbf{P}_0 \leftarrow Selection()$ /*Reset population*/
$\mathbb{F} \leftarrow \mathbf{P}_0.F$;
$t \leftarrow 0$.
Step 2 Update:
$\mathbf{Q}_t \leftarrow \emptyset$
foreach $\mathbf{x}_i \in \mathbf{P}_t$ **do**
  $\mathbf{y} \leftarrow OffspringCreate(\mathbf{x}_i, B(i))$;
  $\mathbf{z}^* \leftarrow UpdateIdealPoint()$;
  /* Update of solutions*/
  foreach $j \in B(i)$ **do**
    if $g(\mathbf{y}|\mathbf{w}_j, \mathbf{z}^*) < g(\mathbf{x}_j|\mathbf{w}_j, \mathbf{z}^*)$ **then**
      $\mathbf{x}_j \leftarrow \mathbf{y}$
    end
  end
  $\mathbf{Q}_t \leftarrow \mathbf{Q}_t \cup \mathbf{y}$
end
$\mathbb{F} \leftarrow \mathbb{F} \cup (\mathbf{P}_{t+1} \cap \mathbf{Q}_t).F$;
$t \leftarrow t + 1$.
Step 3 Weight vectors update
if $|\mathbb{F}| > 5N$ **then**
  $\{\mathbf{w}_1, \ldots, \mathbf{w}_N\} \leftarrow SOM(\mathbb{F}, \mathbf{z}^*)$;
  $B(i) \leftarrow SetNeighbor()$;
  $\mathbb{F} \leftarrow \mathbf{P}_t.F$;
end
Step 4 Stopping Criteria
If the stopping criteria is satisfied, then stop and find all the nondominated solutions in $\mathbf{P}_t$ and output them. Otherwise, goto **Step 2**.

For each subproblem k, it owns a subpopulation I_k as described in [9], which is used to record the best individuals for subproblem k. The subpopulations are initialized as follows: Firstly, the algorithm evenly and randomly generates $5N$ individuals in the domain. N weight vectors $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N\}$ are learned by SOM algorithm according to the objective vectors of these individuals, and set the direction vectors $\{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K\}$. We assign these weight vectors to each subpopulation by Eq. (8), where $\mathbf{y} = \mathbf{w}$. The size n_k of subpopulation I_k is the number of the weight vectors assigned to subproblem k. Then, we initialize I_k as the individuals in the initial individuals whose objective values are in Ω_k. When the number of individuals in I_k is less than n_k, we randomly select $n_k - |I_k|$ from the initial individuals and add to I_k, where $|I_k|$ is the size of I_k. Otherwise, when the number of individuals in I_k is greater than n_k, for each $\mathbf{w}$ assigned in this subproblem, we select the best individual from I_k by Eq. (2). All the selected individuals constitute the new subpopulation I_k. In generation t, N new individuals are generated by the crossover and mutation operators. The new individuals and all the individuals in the subpopulations are used to update the subpopulations.

The objective vectors of the surviving offsprings are used to train the SOM. The detailed procedure of M2M-SOM is given in Algorithm 1. During the search, for each subpopulation k, $k = 1, \ldots, K$, the M2M-SOM maintains:

- I_k: the kth subpopulation;
- u^k: the direction vector;

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2695579, IEEE Transactions on Evolutionary Computation

- $\mathbb{F}$: the objective vectors for training SOM.

In Line 5 of Algorithm 1, the maximum iteration of SOM algorithm is set at 100 for the initiation of the weight vectors. Furthermore, for each individual in a subpopulation in Line 12, an individual is randomly selected from the same subpopulation to generate a new offspring, where $(\mathbf{P}_{t+1} \cap \mathbf{Q}_t).F$ is the objective vectors of the surviving offsprings. In Line 27, we only slightly tune the weight of neurons while updating the weight vectors. Therefore, we set the maximum iteration at 10 in this paper.

D. MOEA/D with SOM-based Weight Vectors

MOEA/D [8] is one of the most well-known EMO algorithms. It aggregates the objectives into a single objective by a weight vector. Then, it decomposes the MOP into a number of scalar problems and simultaneously optimizes them, in which, for each $\mathbf{w}_i$, a neighborhood is defined as a set of T closest weight vectors in $\{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N\}$, where T is the number of the neighbors. The neighborhood $B(i)$ of subproblem i is composed of the subproblems with the weight vectors from the neighborhood of $\mathbf{w}_i$. In this paper, we integrate the SOM-based weight design into MOEA/D, whereby obtaining MOEA/D-SOM, whose details are described in Algorithm 2. Please note that, in Line 7 of Algorithm 2, $Selection()$ returns a solution set which is selected from the initial individuals by Eq. (2) with each weight vector $\mathbf{w}_i$ for $i = 1, \ldots, N$. In Line 13, for each solution $\mathbf{x}_i$, an individual is randomly selected from its neighbors $B(i)$ to generate a new solution $\mathbf{y}$.

IV. Experiments

This section will conduct the experiments to investigate the performance of the proposed M2M-SOM and MOEA/D-SOM in comparison with the existing counterparts, namely, the original M2M and MOEA/D, MOEA/D-AWD [53], NSGA-III [21] and MOEA/DD [23]. In MOEA/D-AWD, it adaptively adjusts the weight vector according to the solution assigned to this weight vector. In the following subsections, we first provide the test problems used in our empirical studies. Then, we introduce the performance metrics, followed by the parameter settings used in the experiments. Finally, we show the experimental results with some discussions.

TABLE I
The parameters of MAOP1-MAOP8

Instances	M	ϕ
MAOP1	4	$[0, \frac{1}{2}\pi, \pi, \frac{3}{2}\pi]$
MAOP2	4	$[0, \frac{5}{6}\pi, \pi, \frac{7}{6}\pi]$
MAOP3	5	$[0, \frac{2}{5}\pi, \frac{4}{5}\pi, \frac{6}{5}\pi, \frac{8}{5}\pi]$
MAOP4	5	$[0, \frac{2}{3}\pi, \frac{5}{6}\pi, \frac{7}{6}\pi, \frac{4}{3}\pi]$
MAOP5	6	$[0, \frac{1}{3}\pi, \frac{2}{3}\pi, \pi, \frac{4}{3}\pi, \frac{5}{3}\pi]$
MAOP6	6	$[0, \frac{1}{2}\pi, \frac{5}{6}\pi, \pi, \frac{7}{6}\pi, \frac{3}{2}\pi]$
MAOP7	7	$[0, \frac{2}{7}\pi, \frac{4}{7}\pi, \frac{6}{7}\pi, \frac{8}{7}\pi, \frac{10}{7}\pi, \frac{12}{7}\pi]$
MAOP8	7	$[0, \frac{1}{2}\pi, \frac{3}{4}\pi, \frac{9}{10}\pi, \frac{11}{10}\pi, \frac{5}{4}\pi, \frac{3}{2}\pi]$

TABLE II
The population size and the value of H ($N1(H)$) in M2M, MOEA/D, MOEA/DD, NSGA-III and MOEA/D-AWD, ($N2$) in M2M-SOM and MOEA/D-SOM, and the maximum number of function evaluation (FES) for different numbers of the objective.

M	$N1(H)$	$N2$	FES
4	286(10)	324	150,000
5	330(7)	361	175,000
6	462(6)	400	200,000
7	462(5)	484	250,000
8	330(4)	529	175,000
9	495(4)	576	200,000
10	715(4)	625	300,000

A. Test Problems

Two classes of test problems with incomplete PF, i.e. redundancy problems and non-redundancy problems, are considered in this paper.

1) Redundancy problem with incomplete PF: Benchmark DTLZ5(m, M) [46] chosen as the redundancy problem with incomplete PF is formulated as follows:

- DTLZ5(m, M)

$$\begin{cases} f_1(\mathbf{x}) & =(1+g)\Pi_{i=1}^{M-1} cos(\theta_i) \\ f_{l=2:M-1}(\mathbf{x}) & =(1+g)\Pi_{i=1}^{M-l} cos(\theta_i) sin(\theta_{M-l+1}) \\ f_M(\mathbf{x}) & =(1+g) sin(\theta_1) \end{cases} \quad (10)$$

with $g = \sum_{i=M}^{M+k-1}(x_i - 0.5)^2$ and

$$\theta_i = \begin{cases} \frac{\pi}{2} x_i & i = 1, \ldots, m-1, \\ \frac{\pi}{4(1+g)}(1+2gx_i) & i = m, \ldots, M-1, \end{cases}$$

subject to $\sum_{j=0}^{m-2} f_{M-j}^2 + 2^{p_i} f_i^2 \geq 1$ for $i = 1, \ldots, M-m+1$, where

$$p_i = \begin{cases} M-m & i = 1, \\ (M-m+2)-i & i = 2 : M-m+1, \end{cases}$$

and $0 \leq x_i \leq 1$ for $i = 1, \ldots, n$, $k = n - M + 1$ is the number of variables which are used to design the function g. As suggested in [54], $k = 10$ is used in the experiments. The PF occurs for the minimum of g function, i.e. at $x_i = 0.5$ for $i = M, \ldots, n$ and satisfies $\sum_{i=1}^{M} f_i^2 = 1$. There are m essential objectives $\{f_i, f_{M-m+2}, \ldots f_M\}$, $i \in \{1, \ldots, M-m+1\}$ and $M-m$ redundancy objectives. Obviously, the PF of DTLZ5(m, M) is incomplete as $m < M$. In this paper, four test instances with different numbers of essential objectives and objectives, i.e. DTLZ5(2,5), DTLZ5(3,5), DTLZ5(2,10) and DTLZ5(3,10) are considered.

Toy problem [55] is also chosen as the other redundancy problem. It is formulated as follows:

- TOY problem

$$\begin{cases} f_1(\mathbf{x}) = |A - x_1| \\ f_2(\mathbf{x}) = |B - x_1| \\ f_i(\mathbf{x}) = |C - x_{i-1}|, i = 3, \ldots, M, \end{cases} \quad (11)$$

where $A = 4, B = 6$ and $C = 5$, and $0 \leq x_i \leq 10$ for $i = 1, \ldots, M-1$, and the dimension of the decision variable

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2695579, IEEE Transactions on Evolutionary Computation

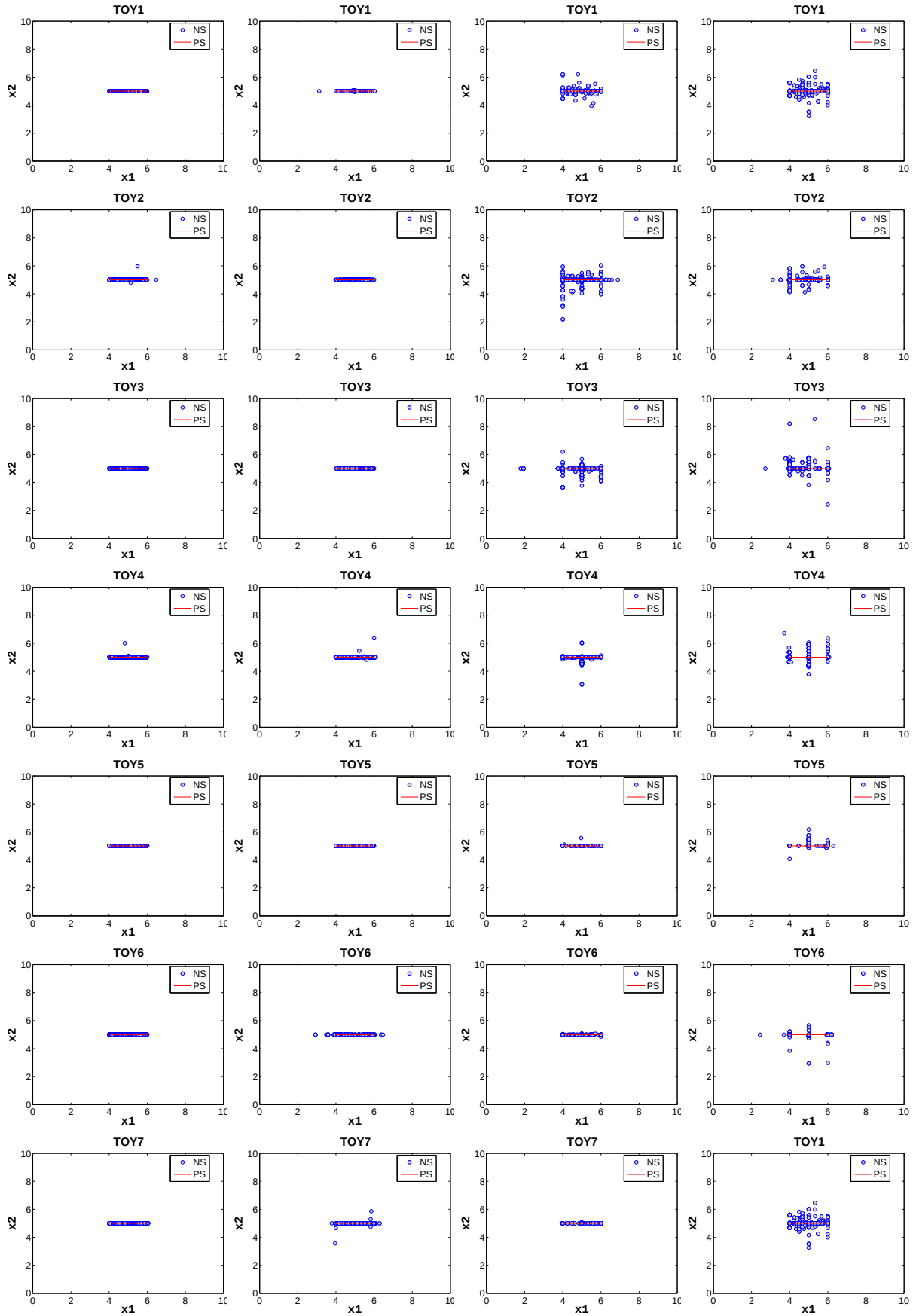


Fig. 3. A plot of the final nondominated solutions (NS) in the subspace x1 vs x2 of the decision space with the minimum IGD-metric value found by M2M-SOM (the first column), MOEA/D-SOM (the second column), MOEA/D-AWD (the third column), and MOEA/D (the fourth column) for TOY1-TOY7.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2695579, IEEE Transactions on Evolutionary Computation

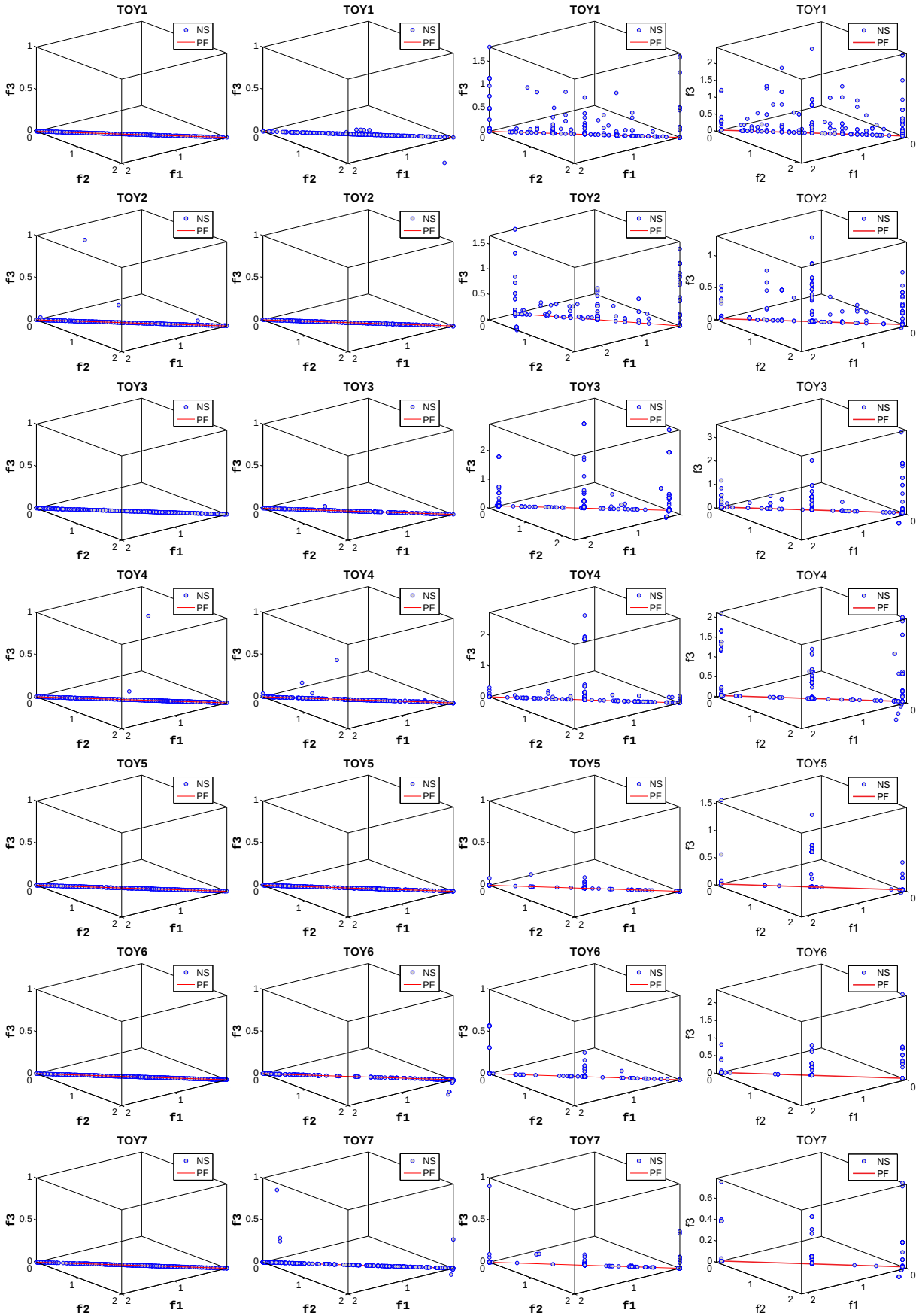


Fig. 4. A plot of the final nondominated solutions (NS) in the three-dimensional subspace f_1, f_2 and f_3 with the minimum IGD-metric value found by M2M-SOM (the first column), MOEA/D-SOM (the second column), MOEA/D-AWD (the third column), and MOEA/D (the fourth column) for TOY1-TOY7.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2695579, IEEE Transactions on Evolutionary Computation

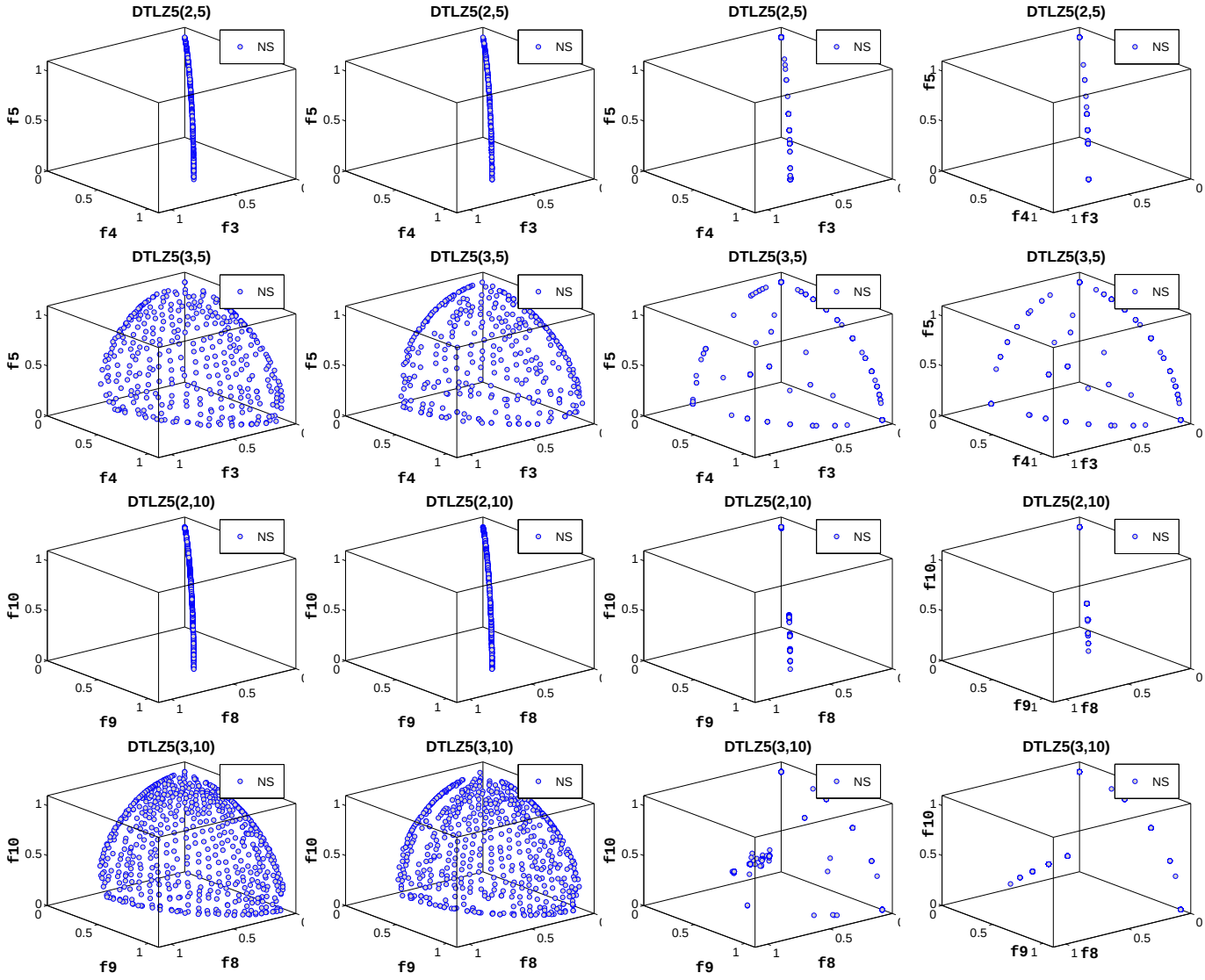


Fig. 5. A plot of the final nondominated solutions (NS) in a three-dimensional subspace of the objective space with the minimum IGD-metric value found by M2M-SOM (the first column), MOEA/D-SOM (the second column), MOEA/D-AWD (the third column), and MOEA/D (the fourth column) for DTLZ5(2,5), DTLZ5(3,5), DTLZ5(2,10) and DTLZ5(3,10).

$n = M - 1$. Its Pareto-optimal set (PS) is $x_1 \in [A, B]$ and $x_i = C$ for all $i = 2, \ldots, n$. The corresponding PF can be described as:

$$\begin{aligned} & f_1 \in |A - B| \\ & f_2 \in |A - B| \\ & f_1 + f_2 = |A - B| \\ & f_j = 0 \quad \forall j = 3, \ldots, M, \end{aligned}$$

where the objectives $f_j, j = 3, \ldots, M$ are redundant. To solve the problem, one way is to optimize these objectives simultaneously, which, however, makes the problem more difficult because the amount of dominating solutions in the neighborhood of the current population gets smaller [55]. We therefore optimize them in a separate way instead. Seven test instances TOY1-TOY7 with the number of objectives from four to ten are used in our empirical studies.

2) Non-redundancy problem with incomplete PF: A framework for MaOPs has been presented in [48], which is a variant of the P^* problems introduced in [52] tailored to our needs. It is formulated as follows:

• MAOP

$$\begin{cases} f_1(\mathbf{x}) = \|\mathbf{x}_I - \mathbf{p}_1\|_2 + \beta(\mathbf{x}_{II} - \gamma(\mathbf{x}_I)) \\ f_2(\mathbf{x}) = \|\mathbf{x}_I - \mathbf{p}_2\|_2 + \beta(\mathbf{x}_{II} - \gamma(\mathbf{x}_I)) \\ \quad \cdots\cdots \\ f_M(\mathbf{x}) = \|\mathbf{x}_I - \mathbf{p}_M\|_2 + \beta(\mathbf{x}_{II} - \gamma(\mathbf{x}_I)) \end{cases} \tag{12}$$

where

- $\mathbf{x}_I = (x_1, x_2)$ and $\mathbf{x}_{II} = (x_3, \ldots, x_n)$ are two subvectors of $\mathbf{x} = (\mathbf{x}_I, \mathbf{x}_{II})$, $-2 \le x_j \le 2$ for $j = 1, 2, \ldots, n$;
- $\mathbf{p}_i = (\sin\phi_i, \cos\phi_i)$ for $i = 1, \ldots, M$ and $\phi_i \in [0, 2\pi]$;
- $\|\mathbf{x}_I - \mathbf{p}_i\|_2$ is defined as the Euclidean distance from $\mathbf{x}_I$ to $\mathbf{p}_i$, $i = 1, 2, \ldots, M$;
- $\beta(\mathbf{x}_{II} - \gamma(\mathbf{x}_I)) = \sum_{j=3}^{n} x_j - 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right)$.

As shown in [48], the PF occurs at $x_j = 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right)$ for $j = 3, \ldots, n$ and $\mathbf{x}_I$ in the convex hull spanned with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2695579, IEEE Transactions on Evolutionary Computation

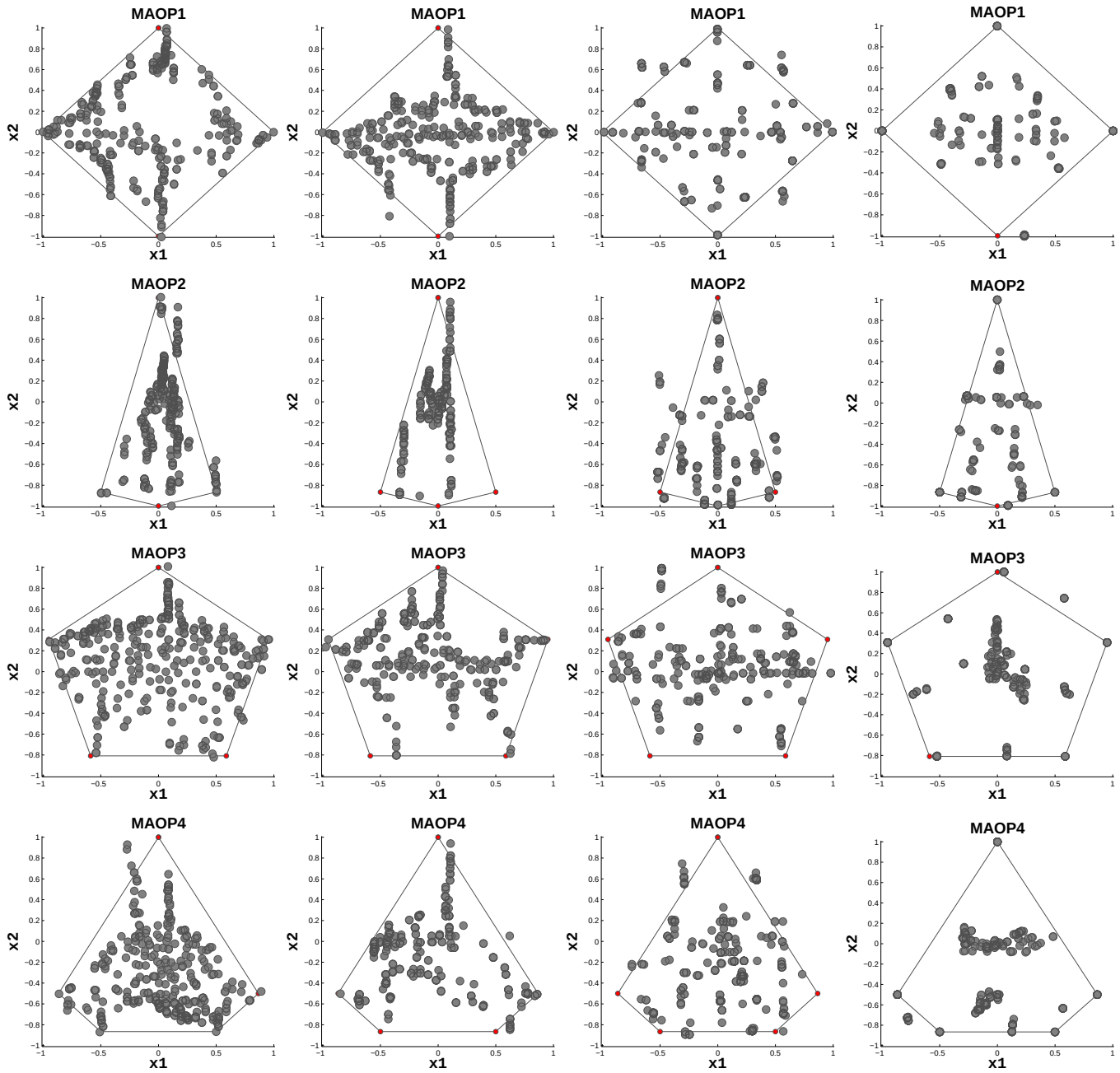


Fig. 6. A plot of the final population in the subspace of x_1 versus x_2 with the median IGD-metric value found by M2M-SOM (the first column), MOEA/D-SOM (the second column), NSGA-III (the third column), and MOEA/D-AWD (the fourth column) on MAOP1-MAOP4.

$\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M\}$. As $M = 3$ and $\phi_i = 0, \frac{2\pi}{3}, \frac{4\pi}{3}$, the PF of the problem is shown as in Fig. 2(b). The PF of such problem is non-degenerate, i.e. there is no redundant objectives. However, the PF is incomplete as shown in Fig. 2(b). When ϕ_i is equally spaced over $[0, 2\pi]$, the objectives of problem (12) have the same importance and value ranges; Otherwise, its objectives have different importance. Eight test instances MAOP1-MAOP8 with different numbers of objectives are used in our empirical studies. Table I lists the number of objectives and ϕ_i for the test instances. The objectives of the odd-numbered test instances have the same importance, while the even-numbered test instances have different importance.

B. Performance Metrics

The IGD-metric [56] and H-metric [47] are used to evaluate the performance of the compared algorithms. In the following, we give a brief introduction of these two metrics.

1) IGD-metric: Let Q^* be a set of points which are uniformly distributed along the PF in the objective space, and Q be an estimate of Q^*. The distance between Q^* and Q is defined as:

$$IGD(Q|Q^*) = \frac{\sum_{v \in Q^*} d(v, Q)}{|Q^*|},$$

where $d(v, Q)$ is the minimum Euclidean distance from the point v to Q. Obviously, the smaller value of IGD is, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2695579, IEEE Transactions on Evolutionary Computation

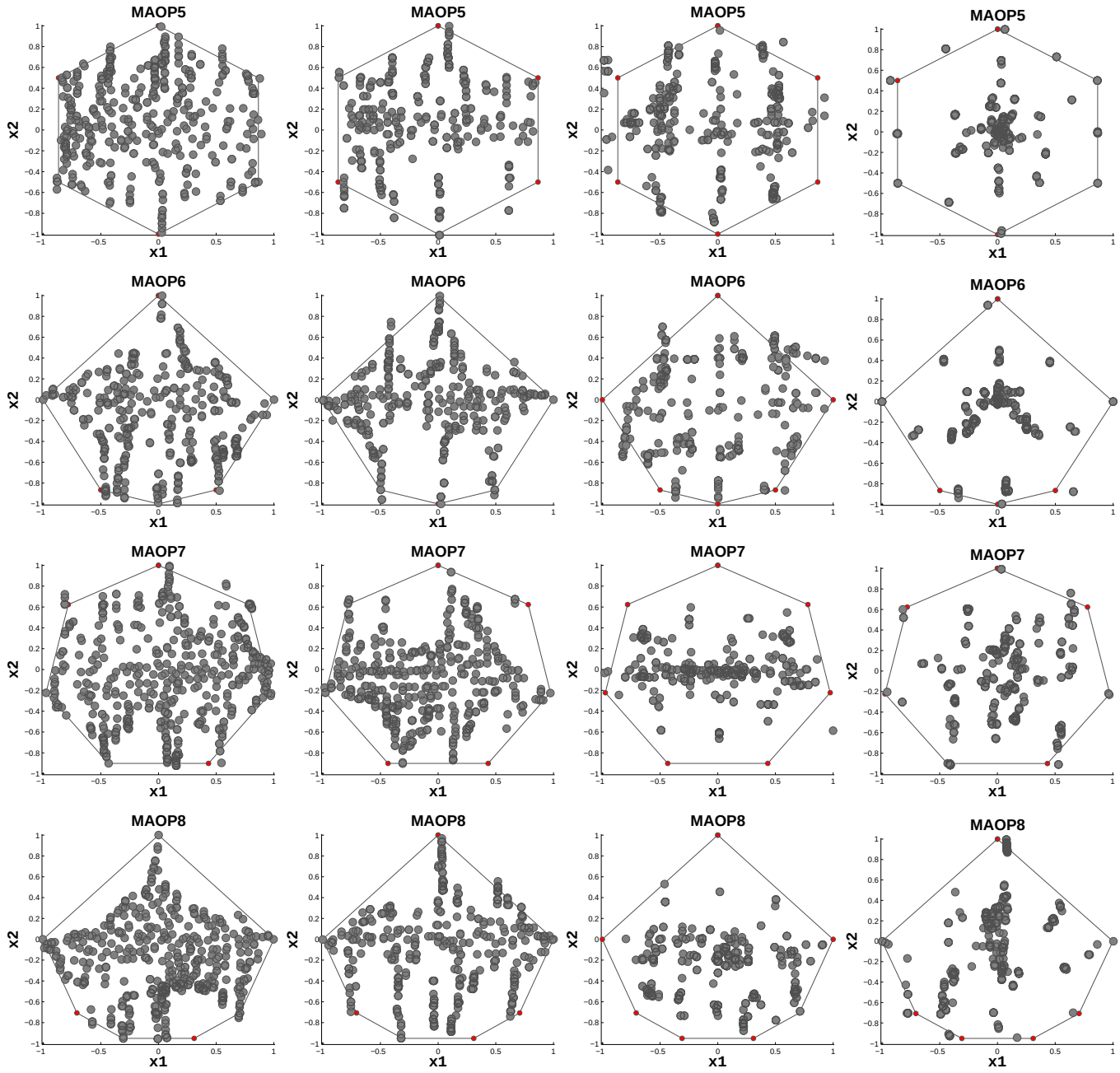


Fig. 7. A plot of the final population in the subspace of x_1 versus x_2 with the median IGD-metric value found by M2M-SOM (the first column), MOEA/D-SOM (the second column), NSGA-III (the third column), and MOEA/D-AWD (the fourth column) on MAOP5-MAOP8.

better the algorithm performs. The PFs of DTLZ5 and TOY problem are precisely known and normative. We generate 1000 points for DTLZ5(2,5), DTLZ5(2,10) and TOY1-TOY7, and 1891 points for DTLZ5(3,5), DTLZ5(3,10) to constitute Q^* by the method introduced in [23]. For the test instances MAOP1-MAOP8, we utilize the objective values of 1000 points which are uniformly sampled in the convex hull spanned with $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M\}$ as described in [48] to form Q^*.

2) H-metric: Let $\mathbf{z}^r = (z_1^r, \ldots, z_M^r)$ be a point in the objective space which is dominated by any Pareto optimal objective vectors. Let Q be the obtained approximation to the PF in the objective space. Then, $H(Q|\mathbf{z}^r)$ is the value of Q (with regard to $\mathbf{z}^r$) is the volume of the region which is dominated by Q and dominates $\mathbf{z}^r$. That is,

$$H(Q|\mathbf{z}^r) = Vol\left(\bigcup_{\mathbf{x}\in Q}[f_1(\mathbf{x}), z_1^r] \times \ldots \times [f_M(\mathbf{x}), z_M^r]\right),$$

where $Vol(.)$ indicates the Lebesgue measure. The larger the H-metric is, the better the approximation is. The algorithm proposed in [6] is adopted to calculate the value of H-metric. In our experiments, the point $\mathbf{z}^r$ is set at the maximum value of each objective of the points used in IGD-metric. Since the PFs of DTLZ5 and TOY problem are degenerated, the value of H-metric of Q^* is a small number for DTLZ5 and 0 for TOY problem. Thus, we only use it to evaluate the performance

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2695579, IEEE Transactions on Evolutionary Computation

TABLE III
THE MINIMUM(BEST) AND MEAN OF IGD-METRIC VALUES OF THE FINAL SOLUTIONS OBTAINED BY SEVEN ALGORITHMS OVER 20 INDEPENDENT RUNS ON TOY TEST PROBLEMS AND DTLZ5

IGD-metric	M2M-SOM		M2M		MOEA/D-SOM		MOEA/D		MOEA/D-AWD		NSGA-III		MOEA/DD	
Instance	best	mean	best	mean	best	mean	best	mean	best	mean	best	mean	best	mean
TOY1	0.00430(1)	0.00599(1)	0.04853(5)	0.06681(5)	0.00760(2)	0.01727(2)	0.04488(4)	0.05273(4)	0.02948(3)	0.03929(3)	0.06642(6)	0.07100(7)	0.07049(7)	0.07060(6)
TOY2	0.00431(1)	0.00602(1)	0.09260(5)	0.12182(7)	0.00724(2)	0.01683(2)	0.05985(4)	0.10127(5)	0.03920(3)	0.06849(3)	0.09948(6)	0.10365(6)	0.10065(7)	0.10085(4)
TOY3	0.00356(1)	0.00531(1)	0.08905(4)	0.17437(7)	0.00643(2)	0.02740(2)	0.10648(5)	0.13779(6)	0.05315(3)	0.09724(3)	0.11102(6)	0.12507(5)	0.11734(7)	0.11757(4)
TOY4	0.00305(1)	0.00547(1)	0.09839(5)	0.23930(7)	0.00664(2)	0.04486(2)	0.08682(4)	0.19236(6)	0.02907(3)	0.05724(3)	0.14045(6)	0.14213(5)	0.14062(7)	0.14108(4)
TOY5	0.00326(1)	0.00561(1)	0.19468(7)	0.33146(7)	0.00564(2)	0.05644(2)	0.09825(4)	0.24649(6)	0.05024(3)	0.08325(3)	0.17533(6)	0.17640(5)	0.17532(5)	0.17616(4)
TOY6	0.00264(1)	0.00919(1)	0.13583(4)	0.34552(7)	0.01730(2)	0.06548(2)	0.14592(5)	0.23853(6)	0.04264(3)	0.07177(3)	0.17499(7)	0.18612(5)	0.17412(6)	0.17659(4)
TOY7	0.00383(1)	0.00818(1)	0.17313(5)	0.38318(7)	0.01035(2)	0.06494(2)	0.14086(4)	0.29130(6)	0.03843(3)	0.06507(3)	0.17790(7)	0.18772(5)	0.17447(6)	0.17653(4)
DTLZ5(2,5)	0.00208(1)	0.00248(1)	0.07758(6)	0.09479(6)	0.00307(2)	0.00449(2)	0.05625(5)	0.06684(5)	0.04199(4)	0.05213(4)	0.02418(3)	0.03742(3)	—	—
DTLZ5(3,5)	0.03317(1)	0.03813(1)	0.11304(6)	0.12039(6)	0.03751(2)	0.04140(2)	0.11137(5)	0.11347(4)	0.10502(4)	0.11372(5)	0.06527(3)	0.06938(3)	—	—
DTLZ5(2,10)	0.00121(1)	0.00137(1)	0.11609(4)	0.17495(6)	0.00194(2)	0.00278(2)	0.13687(5)	0.14487(4)	0.13995(6)	0.15475(5)	0.07355(3)	0.12431(3)	—	—
DTLZ5(3,10)	0.02557(1)	0.02756(1)	0.19683(5)	0.23140(5)	0.02799(2)	0.03090(2)	0.22447(6)	0.24151(6)	0.18988(4)	0.22484(4)	0.15259(3)	0.17543(3)	—	—
Total	11	11	56	70	22	22	51	58	39	39	56	50	—	—
Final Rank	1	1	5	6	2	2	4	5	3	3	5	4	—	—

TABLE IV
THE MINIMUM (BEST) AND MEAN OF IGD-METRIC VALUES OF THE FINAL SOLUTIONS OBTAINED BY THE SEVEN ALGORITHMS OVER 20 INDEPENDENT RUNS ON TEST PROBLEMS MAOP1-MAOP8

IGD-metric	M2M-SOM		M2M		MOEA/D-SOM		MOEA/D		MOEA/D-AWD		NSGA-III		MOEA/DD	
Instance	best	mean	best	mean	best	mean	best	mean	best	mean	best	mean	best	mean
MAOP1	0.10868(2)	0.14268(1)	0.19525(6)	0.22860(6)	0.10537(1)	0.15638(3)	0.26958(7)	0.32890(7)	0.19220(5)	0.20670(5)	0.14045(3)	0.14374(2)	0.18220(4)	0.19691(4)
MAOP2	0.07813(1)	0.10648(1)	0.16419(6)	0.19826(5)	0.11290(3)	0.17957(4)	0.23885(7)	0.32870(7)	0.12757(4)	0.16321(3)	0.10454(2)	0.10743(2)	0.15283(5)	0.19989(6)
MAOP3	0.09984(1)	0.14295(1)	0.25966(4)	0.29643(4)	0.14817(2)	0.19640(3)	0.31866(7)	0.35255(7)	0.27735(5)	0.30200(5)	0.16572(3)	0.17941(2)	0.28589(6)	0.30767(6)
MAOP4	0.09543(1)	0.14283(1)	0.24591(4)	0.28797(4)	0.14006(2)	0.18891(3)	0.29704(7)	0.37826(7)	0.27634(6)	0.30366(5)	0.16343(3)	0.17312(2)	0.26869(5)	0.30507(6)
MAOP5	0.09521(1)	0.13322(1)	0.29150(5)	0.32899(5)	0.14482(2)	0.18942(2)	0.34116(6)	0.39840(7)	0.24065(4)	0.26897(4)	0.19007(3)	0.20481(3)	0.34355(7)	0.38695(6)
MAOP6	0.10865(1)	0.14069(1)	0.28098(5)	0.33091(5)	0.12996(2)	0.17898(2)	0.34195(7)	0.42921(7)	0.25116(4)	0.29470(4)	0.17324(3)	0.18355(3)	0.32395(6)	0.36561(6)
MAOP7	0.10556(1)	0.13738(1)	0.39144(5)	0.46736(5)	0.14556(2)	0.19753(2)	0.59160(7)	0.65765(7)	0.21337(3)	0.23648(3)	0.25837(4)	0.27128(4)	0.45550(6)	0.50927(6)
MAOP8	0.11893(1)	0.14972(1)	0.37675(5)	0.47229(6)	0.13748(2)	0.17224(2)	0.50676(7)	0.63951(7)	0.21015(3)	0.24029(4)	0.21423(4)	0.22926(3)	0.42352(6)	0.44972(5)
Total	9	8	40	40	16	21	55	56	34	34	25	21	45	45
Final Rank	1	1	5	5	2	3	7	7	4	4	3	2	6	6

of the compared algorithms on the test instances: MAOP1-MAOP8.

C. Parameter Settings

The simulated binary crossover (SBX) and polynomial mutation are used in all the algorithms for generating offspring. The control parameters in these two operators are the same in all algorithms. The crossover rate is set at 1 and the mutation rate is $1/n$, where n is the dimension of the variables. The distribution index of crossover and mutation operators are $\eta_c = 20$ and $\eta_m = 20$, respectively.

The uniform weight vectors $\{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N\}$ used in M2M, MOEA/D, NSGA-III and MOEA/DD and the initial weight vectors of MOEA/D-AWD are generated by a systematic approach proposed in [26]. Each weight takes a value from

$$\left\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\right\}.$$

Therefore, the number of the weight vectors is C_{H+M-1}^{M-1}. Table II lists the population size and the value of the parameter H, denoted as $N1(H)$, in M2M, MOEA/D, MOEA/DD, NSGA-III and MOEA/D-AWD for different numbers of objectives. Moreover, Table II also lists the population size in M2M-SOM and MOEA/D-SOM ($N2$). Each algorithm is run 20 times independently for each test instance. The compared algorithms stop when the number of function evaluations reaches the maximum number. Table II lists the maximum number of function evaluations (FES) for different numbers

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2017.2695579, IEEE Transactions on Evolutionary Computation

TABLE V
THE MAXIMUM (BEST) AND MEAN OF H-METRIC VALUES OF THE FINAL SOLUTIONS OBTAINED BY THE SEVEN ALGORITHMS OVER 20 INDEPENDENT RUNS ON TEST PROBLEMS MAOP1-MAOP8

H-metric	M2M-SOM		M2M		MOEA/D-SOM		MOEA/D		MOEA/D-AWD		NSGA-III		MOEA/DD	
Instance	best	mean	best	mean	best	mean	best	mean	best	mean	best	mean	best	mean
MAOP1	2.65369(1)	2.53443(1)	2.19051(6)	2.03409(6)	2.63839(2)	2.45610(2)	2.00513(7)	1.85121(7)	2.26040(4)	2.22597(4)	2.39411(3)	2.35449(3)	2.21851(5)	2.16362(5)
MAOP2	2.99040(1)	2.82821(1)	2.54543(6)	2.37555(6)	2.74164(2)	2.42944(4)	2.20048(7)	1.70908(7)	2.72696(4)	2.59440(2)	2.73831(3)	2.40082(5)	2.58429(5)	2.44003(3)
MAOP3	1.90749(1)	1.82462(1)	1.31801(6)	1.15376(7)	1.77569(2)	1.61851(2)	1.22309(7)	1.15984(6)	1.47525(4)	1.28648(4)	1.61303(3)	1.55768(3)	1.33496(5)	1.26602(5)
MAOP4	1.92570(1)	1.78537(1)	1.29685(5)	1.14531(5)	1.73357(2)	1.46292(3)	1.24190(6)	1.00462(7)	1.36261(4)	1.23281(4)	1.59952(3)	1.55147(2)	1.17150(7)	1.11020(6)
MAOP5	3.39609(1)	3.28848(1)	2.48401(5)	2.28522(5)	3.17850(2)	2.93271(2)	2.24953(6)	2.07037(6)	2.84996(3)	2.54068(4)	2.84645(4)	2.72083(3)	2.08116(7)	1.87196(7)
MAOP6	3.77280(1)	3.56345(1)	2.75079(5)	2.46757(5)	3.33732(2)	3.07635(2)	2.24915(6)	2.06469(6)	3.10507(3)	2.63655(4)	3.08839(4)	2.96823(3)	2.19861(7)	1.93118(7)
MAOP7	2.71882(1)	2.64019(1)	1.89340(5)	1.66972(5)	2.64557(2)	2.29121(3)	1.66585(6)	1.54398(6)	2.48527(3)	2.35094(2)	2.21276(4)	2.15049(4)	1.53138(7)	1.41248(7)
MAOP8	3.77467(1)	3.49714(1)	2.41088(5)	2.09923(5)	3.36334(2)	3.02563(2)	1.90821(6)	1.73486(6)	3.20144(3)	2.90736(4)	3.02777(4)	2.94252(3)	1.88200(7)	1.65058(7)
Total	8	8	42	43	16	20	51	51	28	28	28	26	50	47
Final Rank	1	1	5	5	2	2	7	7	3	4	3	3	6	6

of objectives.

The specific parameter setting in each algorithm is given as follows: In MOEA/D, MOEA/D-SOM and MOEA/D-AWD, the number of the neighbors of each subproblem is set at $T = \lfloor 0.1 * N \rfloor$ as suggested in [8]. The number of subpopulations is set at $K = \lfloor \sqrt{N} \rfloor$ in M2M and M2M-SOM as suggested in [9].

D. Experimental Results and Analysis

Tables III and IV list the minimum (best) and mean values of IGD-metric of the final solutions obtained by the compared algorithms in the 20 independent runs for all the test instances. The values of IGD-metric for each test instance are sorted in an ascending order. The numbers in the brackets of these tables are their ranks. The total rank and the final rank are listed in the last two row in the tables. Since MOEA/DD is designed for unconstrained optimization problems, and DTLZ5 test problems are constrained optimization problems, we didn't provide the results of MOEA/DD for DTLZ5 test problems. Table IV lists the maximum (best) and mean values of H-metric of the final solutions obtained by the seven algorithms for MAOP1-MAOP8. The values of H-metric each test instance are sorted in a descending order. The numbers in the brackets of the table are their ranks. Also the total rank and final rank are listed in the table.

Additionally, due to the limitation of space, we only plotted the results obtained by the first four best algorithms in terms of final rank on IGD-metric. Figures 3 and 4 show the non-dominated solutions obtained by the first four algorithms, i.e., M2M-SOM, MOEA/D-SOM, MOEA/D-AWD and MOEA/D with the minimum IGD-metric in the 20 independent runs for test instances TOY1-TOY7 in the decision subspace, i.e. x_1 versus x_2, and in the objective subspace constituted by f_1, f_2 and f_3. Figure 5 plots the nondominated solutions obtained by the first four best algorithms, i.e., M2M-SOM, MOEA/D-SOM, NSGA-III and MOEA/D-A with the minimum IGD-metric in the 20 independent runs for DTLZ5(2,5), DTLZ5(3,5), DTLZ5(2,10) and DTLZ5(3,10). Figures 6 and 7 plot the final population obtained by the first four best algorithms on MaOP test problems, i.e., M2M-SOM, MOEA/D-SOM, NSGA-III and MOEA/D-AWD with the minimum IGD-metric in the 20 independent runs in the subspace of x_1 versus x_2 for MAOP1-MAOP8. As discussed in [48], the Pareto solutions must be within the convex hull spanned with points $\{\mathbf{p}_1, \ldots, \mathbf{p}_M\}$.

TOY problem: The results on TOY problem in Table III, and Figures 3 and 4 indicate that the algorithms with the SOM-based weight vectors outperform the other algorithms. Specifically, the proposed M2M-SOM is the best optimizer, which is able to achieve a good approximation of the PF. The reason that M2M-SOM is slightly better than MOEA/D-SOM is because the newly generated solution, which may be far away from its neighbors especially in MaOPs, updates its neighbors only in MOEA/D-SOM. In the literature, an improvement strategy, namely MOEA/D-GR, has been proposed in [57] to deal with such an issue. MOEA/D-GR firstly assigns the newly generated solution to the closest weight vector and then updates its neighbor. Furthermore, it can be seen that MOEA/D is slightly better than M2M. A plausible reason is that there are no Pareto-optimal solutions to some subpopulation for M2M. The performance of the algorithms with the proposed weight vector design method is better than that of MOEA/D-AWD because the weight vectors in MOEA/D-AWD are adjusted according to only one solution which is assigned to this weight vector. It may make the weight vectors over-fit the current solutions. Moreover, a special note goes to the performance of NSGA-III, since it sorts the solutions according to the domination relationship of the solutions and the amount of dominating solutions in the neighborhood of the current population is very smaller for TOY problem, it is difficult to find a dominating solution by using Pareto-dominance. This is consistent with the results found in [55].

DTLZ5: From Table III, we can see that the results obtained by the algorithms with the proposed SOM-based weight are

much superior than that of its counterparts in term of IGD-metric. Figure 5 reveals that M2M-SOM and MOEA/D-SOM can achieve a good approximation of the PF. Specifically, the performance of M2M and MOEA/D are not very promising because the PF of the test instances is incomplete and there are numerous scalar problems obtaining the same Pareto optimal solutions. NSGA-III can obtain a promising result as the non-dominated sorting can provide a sufficient selection pressure towards to the PF. It is because it permits a reference point mapped to multiple solutions and no solution is mapped to some reference points. It is good at maintaining the diversity of the population.

MaOP: In Table IV, the performance of the algorithms with the proposed SOM-based weight is better than that of its counterparts in term of IGD-metric for MAOP1-MAOP8. From Table V, it can be seen that M2M-SOM and MOEA/D-SOM are the best optimizer, which achieves the greater H-metric values in almost all of MaOP test instances. Further, Figures 6 and 7 both indicate that almost all solutions obtained by M2M-SOM and MOEA/D-SOM are within the convex hull, which reveals the quality and distribution of the nondominated solutions obtained by an algorithm. This implies that the solutions from M2M-SOM and MOEA/D-SOM converge well to the PF. In contrast, there are few solutions obtained by NSGA-III and MOEA/D-AWD converge to the PF.

V. Conclusion

In this paper, we have proposed a weight design method based on SOM, which can be integrated with most of the decomposition-based evolutionary algorithms for solving MaOP. In implementation, we have integrated this weight design method into M2M and MOEA/D, respectively, i.e. M2M-SOM and MOEA/D-SOM, as two examples. In the proposed algorithms, we update the weight vectors by the weight of the neurons of SOM network which is trained by the objective vectors of the surviving offsprings. Such weight can be adjusted based on the distribution of the individuals. We have investigated the performance of the proposed algorithms on eleven redundancy test problems and eight non-redundancy test problems, respectively. The empirical results have demonstrated the superiority of M2M-SOM and MOEA/D-SOM for solving MaOPs with incomplete PF in comparison with the existing three counterparts.

References

[1] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. C. Coello, "A survey of multiobjective evolutionary algorithms for data mining: Part I," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 4–19, 2014.

[2] M. Asafuddoula, H. K. Singh, and T. Ray, "Six-sigma robust design optimization using a many-objective decomposition-based evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 490–507, 2015.

[3] J. Herrero, A. Berlanga, and J. Lopez, "Effective evolutionary algorithms for many-specifications attainment: application to air traffic control tracking filters," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 151–168, 2009.

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

[5] Z. Eckart, L. Marco, and T. Lothar, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, 2001, pp. 95–100.

[6] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[7] B. Nicola, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

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

[9] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

[10] H.-L. Liu, F. Gu, Y.-M. Cheung, S. Xie, and J. Zhang, "On solving wcdma network planning using iterative power control scheme and evolutionary multiobjective algorithm," *IEEE Computational Intelligence Magazine*, vol. 9, no. 1, pp. 44–52, 2014.

[11] F. Gu, H.-L. Liu, Y.-M. Cheung, and S. Xie, "Optimizing wcdma network planning by multiobjective evolutionary algorithm with problem-specific genetic operation," *Knowledge and Information Systems*, vol. 45, no. 3, pp. 679–703, 2015.

[12] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 264–283, 2014.

[13] X. Zou, Y. Chen, M. Liu, and L. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 5, pp. 1402–1412, 2008.

[14] H. Wang and X. Yao, "Corner sort for pareto-based many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 1, pp. 92–102, 2014.

[15] L. Batista, F. Campelo, F. Guimaraes, and J. Ramirez, "A comparison of dominance criteria in many-objective optimization problems," in *Proc. IEEE Congress on Evolutionary Computation*, 2011, pp. 2359–2366.

[16] D. Hadka and P. Reed, "Borg: An auto-adaptive many-objective evolutionary computing framework," *Evolutionary Computation*, vol. 21, no. 2, pp. 231–259, 2013.

[17] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[18] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–35, 2015.

[19] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 445–460, 2015.

[20] O. Schütze, A. Lara, and C. A. C. Coello, "On the influence of the number of objectives on the hardness of a multiobjective optimization problem," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 444–455, 2011.

[21] 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 Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[22] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

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

[24] H. Ishibuchi, H. Masuda, and Y. Nojima, "Pareto fronts of many-objective degenerate test problems," *IEEE Transactions on Evolutionary Computation*, 2015.

[25] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Transactions on Evolutionary Computation*, Doi: 10.1109/TEVC.2016.2587749, 2016.

[26] 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, no. 3, pp. 631–657, 1998.
[27] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition," in *Proc. Evolutionary Multi-Criterion Optimization*, 2013, pp. 428–442.
[28] 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.
[29] F. Gu, H.-L. Liu, and K. C. Tan, "A multiobjective evolutionary algorithm using dynamic weight design method," *International Journal of Innovative Computing Information and Control*, vol. 8, no. 5B, pp. 3677–3688, 2012.
[30] F. Gu and H.-L. Liu, "A novel weight design in multi-objective evolutionary algorithm," in *Proc. 2010 International Conference on Computational Intelligence and Security*, 2010, pp. 137–141.
[31] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evolutionary Computation*, vol. 19, no. 4, pp. 561–595, 2011.
[32] S. Jiang, Z. Cai, J. Zhang, and Y.-S. Ong, "Multiobjective optimization by decomposition with pareto-adaptive weight vectors," in *Proc. 2011 Seventh International Conference on Natural Computation*, 2011, pp. 1260–1264.
[33] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.
[34] ——, "The self-organizing map," *Neurocomputing*, vol. 21, no. 1, pp. 1–6, 1998.
[35] Y.-M. Cheung and L. Law, "Rival-model penalized self-organizing map," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 289–295, 2007.
[36] M. Tucci and M. Raugi, "A filter based neuron model for adaptive incremental learning of self-organizing maps," *Neurocomputing*, vol. 74, no. 11, pp. 1815–1822, 2011.
[37] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proceedings of the IEEE*, vol. 84, no. 10, pp. 1358–1384, 1996.
[38] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 586–600, 2000.
[39] A. Flexer, "On the use of self-organizing maps for clustering and visualization," *Intelligent Data Analysis*, vol. 5, no. 5, pp. 373–384, 2001.
[40] J. Vesanto, "Som-based data visualization methods," *Intelligent data analysis*, vol. 3, no. 2, pp. 111–126, 1999.
[41] N. Li and Y. Li, "Feature encoding for unsupervised segmentation of color images," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 33, no. 3, pp. 438–447, 2003.
[42] H. Zhang, A. Zhou, S. Song, Q. Zhang, X. Z. Gao, and J. Zhang, "A self-organizing multiobjective evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 792–806, 2016.
[43] H. Zhang, X. Zhang, X.-Z. Gao, and S. Song, "Self-organizing multiobjective optimization based on decomposition with neighborhood ensemble," *Neurocomputing*, vol. 173, pp. 1868–1884, 2016.
[44] S. Obayashi and D. Sasaki, "Visualization and data mining of pareto solutions using self-organizing map," in *Proc. Evolutionary multi-criterion optimization*, 2003, pp. 796–809.
[45] D. R. Dersch and P. Tavan, "Asymptotic level density in topological feature maps," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 230–236, 1995.
[46] H. K. Singh, A. Isaacs, and T. Ray, "A pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 539–556, 2011.
[47] M. Emmerich, N. Beume, and B. Naujoks, "An EMO algorithm using the hypervolume measure as selection criterion," in *Proc. Evolutionary multi-criterion optimization*, 2005, pp. 62–76.
[48] Y.-M. Cheung, F. Gu, and H.-L. Liu, "Objective extraction for many-objective optimization problems: algorithm and test problems," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 755–772, 2016.
[49] K. Deb, *Multiobjective Optimization using Evolutionary Algorithms*. New York: Wiley, 2001.
[50] K. Miettinen, *Nonlinear multiobjective optimization*. Springer Science & Business Media, 1999.
[51] O. Schutze, A. Lara, and C. A. Coello Coello, "On the influence of the number of objectives on the hardness of a multiobjective optimization problem," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 444–455, 2011.
[52] M. Köppen and K. Yoshida, "Many-objective particle swarm optimization by gradual leader selection," in *Adaptive and Natural Computing Algorithms*, 2007, pp. 323–331.
[53] X. Guo, X. Wang, and Z. Wei, "Moea/d with adaptive weight vector design," in *Proc. 11th International Conference on Computational Intelligence and Security*, 2015, pp. 291–294.
[54] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.
[55] H. Zille and S. Mostaghim, "Using ϵ-dominance for hidden and degenerated pareto-fronts," in *Proc. IEEE Symposium Series on Computational Intelligence*, 2015, pp. 845–852.
[56] P. A. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.
[57] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive replacement strategies for MOEA/D," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 474–486, 2016.

Fangqing Gu received the B.S. degree from Changchun University, Jilin, China, in 2007 and the M.S. degree from Guangdong University of Technology, Guangzhou, China, in 2011. He received the Ph.D. degree from the Department of Computer Science at Hong Kong Baptist University in 2016. He joined the School of Applied Mathematics, Guangdong University of Technology, Guangdong, as a Lecturer. His research interests include data mining, machine learning and evolutionary computation.

Yiu-ming Cheung received the Ph.D. degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, in 2000. Currently, he is a Full Professor with the Department of Computer Science, Hong Kong Baptist University. His research interests include machine learning, information security, image and video processing, and pattern recognition. He is the Founding Chair of the Computational Intelligence Chapter of IEEE Hong Kong Section and the Vice Chair of Technical Committee on Intelligent Informatics (TCII) of IEEE Computer Society. He is an IET/IEE Fellow and BCS Fellow.