An Improved Multiobjective Optimization Evolutionary Algorithm Based on Decomposition for Complex Pareto Fronts

Shouyong Jiang and Shengxiang Yang, *Senior Member, IEEE*

***Abstract*—The multiobjective evolutionary algorithm based on decomposition (MOEA/D) has been shown to be very efficient in solving multiobjective optimization problems (MOPs). In practice, the Pareto-optimal front (POF) of many MOPs has complex characteristics. For example, the POF may have a long tail and sharp peak and disconnected regions, which significantly degrades the performance of MOEA/D. This paper proposes an improved MOEA/D for handling such kind of complex problems. In the proposed algorithm, a two-phase strategy (TP) is employed to divide the whole optimization procedure into two phases. Based on the crowdedness of solutions found in the first phase, the algorithm decides whether or not to delicate computational resources to handle unsolved subproblems in the second phase. Besides, a new niche scheme is introduced into the improved MOEA/D to guide the selection of mating parents to avoid producing duplicate solutions, which is very helpful for maintaining the population diversity when the POF of the MOP being optimized is discontinuous. The performance of the proposed algorithm is investigated on some existing benchmark and newly designed MOPs with complex POF shapes in comparison with several MOEA/D variants and other approaches. The experimental results show that the proposed algorithm produces promising performance on these complex problems.**

***Index Terms*—Multiobjective evolutionary algorithm (MOEA), multiobjective evolutionary algorithm based on decomposition (MOEA/D), multiobjective optimization, niching, test problems.**

I. Introduction

IN THE evolutionary computation (EC) community, multiobjective evolutionary algorithms (MOEAs) have gained great popularity for solving practical multiobjective optimization problems (MOPs) [1], [2], [6], [42]. The advantages of MOEAs over traditional optimization methods mainly lie in that an MOEA works with a population of candidate solutions in parallel and thus is able to obtain an approximation of the Pareto-optimal front (POF) that consists of a set of tradeoff solutions, known as Pareto-optimal set (POS), for an MOP in a single run. In recent years, a number of well-known and powerful MOEAs based on Pareto optimality have been introduced [7], [21], [22], [25], [32], [37], [48], and their effectiveness for handling MOPs has been validated by empirical studies and various applications. Apart from those Pareto-dominance-based MOEAs, there has been increasing interest in decomposition-based approaches for evolutionary multiobjective optimization [3], [15]–[17], [20], [26], [44].

In this paper, we focus on the MOEA based on decomposition (MOEA/D) [44], which is a recent breakthrough in the design of MOEAs. In contrast to those MOEAs that use Pareto-dominance to guide the search, MOEA/D decomposes an MOP into a number of scalar optimization subproblems and simultaneously solves them in a collaborative manner. The performance of MOEA/D was compared with the improved nondominated sorting genetic algorithm (NSGA-II) in [7] and [26] for MOPs with simple and complicated Pareto sets, respectively, showing that MOEA/D was able to generate the best set of diverse nondominated solutions close to the POF in all tested cases. Furthermore, the efficiency of MOEA/D was confirmed by winning the unconstrained MOEA competition in the 2009 IEEE Congress on EC (IEEE CEC 2009) [46]. Since then, MOEA/D has attracted increasing research interest and various modified versions have been proposed in [19], [31], and [40]. Besides, the idea has also been integrated into hybrid algorithms [5], [27], [28], [36], [41].

The remarkable performance of MOEA/D is due to its diversity maintenance among subproblems and its information sharing or mating/updating strategy between individuals residing in the neighborhood. The diversity of subproblems guarantees a set of well-distributed solutions, which is achieved by decomposing an MOP by a uniform distribution of weight vectors, while information sharing helps a population converge toward the POF as fast as possible.

In the literature, there exists a number of works focusing on the uniformity of weight vectors to achieve the diversity of the resulting subproblems. While a set of evenly distributed weight vectors works well for simple MOPs, especially when the shape of the POF is close to the hyperplane $\sum_{i=1}^{M} f_i = 1$ (where M is the number of objectives and f_i is the ith objective) [30], it cannot give a good distribution for complex MOPs, such as the POF that has a long tail and sharp

Manuscript received August 30, 2014; revised November 20, 2014 and February 3, 2015; accepted February 5, 2015. This work was supported by the Engineering and Physical Sciences Research Council of U.K. under Grant EP/K001310/1. This paper was recommended by Associate Editor Y.S. Ong. *(Corresponding author: Shengxiang Yang.)*

The authors are with the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K. (e-mail: syang@dmu.ac.uk).

Digital Object Identifier 10.1109/TCYB.2015.2403131

peak [40]. This means MOEA/D will struggle to find points on the boundary or extreme regions of the POF in those cases. Besides, the mating mechanism used in MOEA/D is likely to produce duplicate solutions if the neighborhood contains similar solutions, especially when the POF of the MOP being optimized has disconnected regions.

In recent years, there has been an amount of research interest on these issues. As the uniformity of weight vectors plays a fundamental role in MOEA/D, some researchers [12], [40], [43] employ weight adjustment schemes to achieve a best approximation possible. However, Giagkiozis *et al.* [10] argued that an even or uniform distribution of weight vectors does not necessarily produce evenly distributed solutions on the POF. Another feasible way is to develop effective decomposition approaches [18], [19], [45]. For example, Zhang *et al.* [45] proposed a normal boundary intersection (NBI) style Tchebycheff approach to overcome the sensitivity to scales of the objectives, thereby expecting a good distribution of approximated POF. While it works well on bi-objective problems, this approach can not be easily extended to higher dimensional problems. Another interesting method for handing complex problems is to adopt an objective transform strategy [30] if the geometry of the POF is below the hyperplane $\sum_{i=1}^{M} f_i = 1$. The basic idea behind it is to identify the POF geometry, getting the transformed POF as close to this hyperplane as possible. The idea seems helpful. However, if there is noise in the approximated POF used in identifying the POF shape, this method will fail. Besides, the identification of the POF geometry or the transform parameter itself is an optimization problem, which requires extra computational resources.

To overcome these shortcomings, an improved MOEA/D with a two-phase strategy (TP) and a niche-guided scheme, denoted MOEA/D-TPN, is proposed in this paper. In MOEA/D-TPN, TP is conditional and will be activated when there are less boundary or extreme solutions than intermediate solutions in the population, which helps the algorithm to dedicate computational resources to find boundary solutions, and the niche-guided scheme is used to perform the mating operation with parents in the less crowded regions, thereby avoiding duplicated solutions in the offspring. MOEA/D-TPN is tested on some existing and newly designed complex MOPs, showing better performance than its predecessor. Additionally, the performance of MOEA/D-TPN is also compared with strength Pareto evolutionary algorithm 2 with shift-based density estimation (SPEA2+SDE) [24], NSGA-III [8], in our experimental study.

The rest of this paper is organized as follows. Section II presents a brief review of MOEA/D. The improved MOEA/D, i.e., MOEA/D-TPN, is described in Section III. Section IV presents some newly designed test problems with complex POFs and the experimental study based on them. A further discussion of the algorithm is provided in Section V. Section VI concludes this paper.

II. Review of MOEA/D

MOEA/D employs decomposition to convert an MOP into a number of scalar optimization subproblems and then optimizes them simultaneously. Three decomposition approaches have been used in [44]. For simplicity, we use the Tchebycheff approach in this paper, and the scalar optimization subproblems are in the following form:

$$\begin{aligned} \min \quad & g\left(x|\lambda, z^*\right) = \max_{1\le j\le M} \lambda_j \left(f_j(x) - z_j^*\right) \\ \text{s.t.} \quad & x \in \Omega_x \end{aligned} \tag{1}$$

where $\lambda = (\lambda_1, \ldots, \lambda_M)$ is a weight vector with $\sum_{j=1}^{M} \lambda_j = 1$, Ω_x is the solution space, $f_j(x)$ is the jth objective, and z^* is the reference point, i.e., $z_j^* = \min\{f_j(x)|x \in \Omega_x\}$ for each $j = 1, \ldots, M$.

Since its introduction, MOEA/D has received increasing research interest, and many studies have been published to further enhance its performance. Since the uniformity of weight vectors is of great importance to MOEA/D, researchers have proposed various weight vector design methods. Zhang and Li [44] adopted the simplex-lattice design to set a fixed number of weight vectors. But this method cannot guarantee a set of uniform solutions [40]. Based on the mapping relation between the weight vectors and optimal solutions, Qi *et al.* [40] proposed a new weight vector initialization method, called WS-transformation. Besides, they also designed an elite population-based adaptive weight vector adjustment (AWA) strategy, where an elite population is introduced to help add new subproblems into the real sparse regions of the objective space. This way, AWA is expected to solve complex MOPs with disconnected subregions in the POF more effectively. In [43], a uniform design method is applied to generate the weight vectors. This uniform design method has the advantages: it can generate a more uniform distribution of the weight vectors and the resulting population size does not increase nonlinearly with the number of objectives.

Apart from refining the weight vectors, there exists work on maintaining the population diversity. Li and Zhang [26] introduced the differential evolution (DE) operator [38] into MOEA/D, which leads to the MOEA/D-DE algorithm. Recently, an interesting mating selection method has been presented in [29]. To reach a balance between convergence and diversity of the search process, Li *et al.* [29] considered the selection of promising solutions for subproblems as a matching between subproblems and solutions, thus proposing a simple and effective stable matching model to coordinate the selection process. Following this line, an interrelationship-based mating selection has been also introduced in [23]. Other diversity maintenance mechanisms, such as the adaptive mating selection mechanism [4], ensemble of neighborhood sizes [47], variable neighborhood [33], and opposition-based learning strategy [34], have been successfully applied to MOEA/D.

Since MOEA/D uses decomposition approaches to decompose an MOP into a number of scalar subproblems, any attempt to improve decomposition approaches is not trivial. Some newly developed decomposition approaches include the NBI-style Tchebycheff approach [45], adaptive scalarizing functions [18], and simultaneous use of different scalarizing functions [19]. In [10], a novel generalized decomposition is introduced, providing a framework with which the decision maker can guide the underlying evolutionary algorithm toward

specific regions of interest of the entire POF with the desired distribution of POS.

There have been other approaches for enhancing MOEA/D to handle complicated MOPs. Liu *et al.* [30] suggested the use of objective transform for finding a set of uniformly distributed solutions if the POF shape is convex. Recently, a new version of MOEA/D, called MOEA/D-M2M, has been proposed in [31]. Interestingly, MOEA/D-M2M does not require any aggregation methods and a subpopulation is responsible for each decomposed subproblem.

III. Improved MOEA/D

In this section, the MOEA/D-TPN algorithm is presented. MOEA/D-TPN improves MOEA/D with two strategies: the TP strategy, which conditionally divides the whole optimization process into two phases and the niche-guided strategy, which helps increase the population diversity in a more reliable way. The two strategies are described below.

A. TP Optimization

As stated in [40], there are two main issues in MOEA/D. One is the nonuniformity of approximated solutions along the convex POF that has complex shapes with a sharp peak and long tail, where a small variation in one objective results in a large gap in another objective. In this case, MOEA/D offers dense solutions in the intermediate region of the POF and can hardly achieve well-distributed solutions in the extreme region of the POF although a set of uniform weight vectors is provided. Fortunately, MOEA/D is free from this drawback in concave MOPs [30]. Thus, it is natural to achieve uniform solutions for a convex MOP by solving the scalar optimization subproblems in a reversed form

$$\begin{aligned} \max \quad & g^r\left(x|\lambda, r^*\right) = \min_{1\le j\le M} \lambda_j\left(r_j^* - f_j(x)\right) \\ \text{s.t.} \quad & x \in \Omega_x \end{aligned} \tag{2}$$

where r^* is the nadir point constructed from the worst objective values for the entire POS, i.e., $r_j^* = \max\{f_j(x)|x \in \Omega_x\}$ for each $j = 1, \ldots, M$.

When handling an MOP, however, its convexity–concavity is not always known beforehand. This gives rise to the problem on how to properly choose the subproblem form. Besides, the nadir point is also not available before the evolution. For these reasons, we propose the TP method that divides the whole evolution procedure into two phases. The first $M_r\%$ of the entire computing resources is called the first phase, where MOEA/D uses the scalar subproblem form of Eq. (1) and concentrates on convergence and diversity. At the end of the first phase, a crowding-based method is used to evaluate whether MOEA/D has obtained a set of uniform solutions or not. The crowding-based method assesses the density of solutions in both the intermediate region and the extreme region of the POF. If the solutions in the intermediate region is denser than those in the extreme region, it implies that the MOP is probably convex, and the use of the reversed scalar subproblem form will be more suitable in the remaining optimization phase. Note that the nadir point used in Eq. (2) for the second phase can be constructed by the obtained Pareto-optimal solutions from the first phase. The following illustrates how the crowding-based method works.

Any weight vector λ used in MOEA/D should satisfy the restriction $\lambda_1 + \cdots + \lambda_M = 1$, where each ingredient λ_j is nonnegative. According to the arithmetic mean–geometric mean inequality theorem, $\prod_{j=1}^{M} \lambda_j \le (\sum_{j=1}^{M} \lambda_j/M)^M$ or simply $\prod_{j=1}^{M} \lambda_j \le (1/M)^M$ always holds, and the equality condition of this inequality is that if and only if all ingredients of λ are equal, which means λ is located in the center of the hyperplane $\lambda_1 + \cdots + \lambda_M = 1$. In other words, the larger $\prod_{j=1}^{M} \lambda_j$ is, the closer the weight vector is to the central weight vector. Therefore, we can categorize the set of N weight vectors employed in MOEA/D into two subsets, the intermediate subset W_m and the extreme subset W_e, as follows:

$$\begin{aligned} W_m &= \left\{ \lambda^i \in \Omega_\lambda | \prod_{j=1}^{M} \lambda_j^i \ge 0.5\left(\frac{1}{M}\right)^M \right\} \\ W_e &= \Omega_\lambda / W_m \end{aligned} \tag{3}$$

where Ω_λ is the set of N weight vectors generated by the simplex-lattice design used in the original MOEA/D. A value of 0.5 is used in Eq. (3) as it can generate a good classification between W_m and W_e. A too large or too small value of this parameter will cause size imbalance of the classified subsets. Fig. 1 shows examples in 2-D and 3-D cases.

Then, according to the weight vector subsets W_m and W_e, the population can be divided into two sub-populations, called the intermediate sub-population P_m and the extreme sub-population P_e, respectively. The crowdedness of the two sub-populations are estimated by

$$D_{\text{mid}} = \frac{1}{|P_m|} \sum_{i \in P_m} \gamma(i) \tag{4}$$

$$D_{\text{ext}} = \frac{1}{|P_e|} \sum_{i \in P_e} \gamma(i) \tag{5}$$

where D_{mid} and D_{ext} are the crowdedness values for the intermediate and extreme sub-populations, respectively, $\gamma(i)$ quantifies the crowding level of the ith solution in a sub-population, which is defined as follows:

$$\gamma(i) = \frac{1}{T} \sum_{j \in B(i)} d(i, j) \tag{6}$$

where d_{ij} is the distance between the solutions i and j and $B(i)$ and T are the neighboring members of solution i and the neighborhood size in MOEA/D, respectively. The definition of $\gamma(i)$ is not trivial because it measures the closeness of the T neighboring solutions to the ith solution.

It should be noted that the second phase is an alternative, which means if there is no much difference between D_{mid} and D_{ext}, the second phase can be eliminated from the evolution procedure, and the first phase will last for the whole evolution.

B. Niche-Guided Mating/Update Selection

Another issue regarding MOEA/D is that it will produce many similar solutions due to the neighborhood mating and updating strategy. The neighborhood mating plays

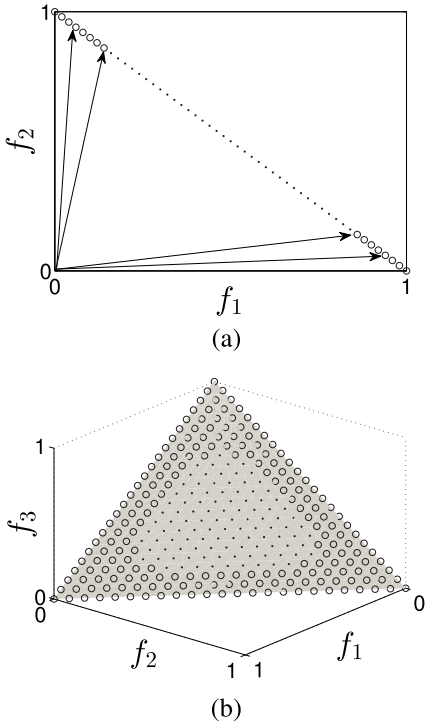

Fig. 1. Distribution of the extreme weight vectors (circles) and intermediate weight vectors (black dots). (a) Two-objective case. (b) Three-objective case.

an important role in MOEA/D. In the original version of MOEA/D [44], the authors used the simulated binary crossover (SBX) operator to produce new solutions. Later in MOEA/D-DE [26], the same authors employed the DE operator [38], instead of SBX, to perform the mating operation and increased the population diversity by selecting three parent solutions from the whole population with a small probability $1-\delta$. This modification, however, can not guarantee that the selected solutions are distinct and may result in producing similar solutions in each generation. When an MOP has disconnected sub-regions on the POF, the effect is more severe and many scalar subproblems may obtain similar solutions on breakpoints, leading to a decrease in the population diversity. Thus, the selection of mating range is also of great importance to the performance of MOEA/D.

In the following, we propose a niche-guided scheme for the selection of the mating/update range. The scheme computes the niche count of each individual over its T neighboring individuals instead of all members of the population. The niche count for each individual i, denoted $nc(i)$, is calculated by summing a sharing function over its T neighboring individuals as

$$nc(i) = \sum_{j=1}^{T} \mathrm{sh}\big(d_{ij}\big) \tag{7}$$

where d_{ij} is the distance between individuals i and j and $\mathrm{sh}(d_{ij})$ is the sharing function that measures the similarity level between individuals i and j, which is defined as

$$\mathrm{sh}\big(d_{ij}\big) = \begin{cases} 1-\left(\frac{d_{ij}}{\sigma_{\mathrm{share}}}\right)^{\alpha}, & \text{if } d_{ij} \le \sigma_{\mathrm{share}} \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where σ_{share} is a predefined niche radius and α is a constant, called the sharing level [11].

If the niche count of an individual is over a given threshold, it means that the individual is similar to its T neighboring individuals and it is desirable to choose individuals outside the neighborhood as the mating parents. Hence, the mating/update range can be defined as follows:

$$P'' = \begin{cases} P, & \text{if } nc(i) < \beta \\ P', & \text{otherwise} \end{cases} \tag{9}$$

where β is the threshold that is closely related to the niche radius σ_{share}, i.e., β is mainly determined by σ_{share}. As the maximum value of $nc(i)$ is T, we set β to be $T/2$ in this paper. P and P' are calculated by

$$P = \begin{cases} B(i), & \text{if } rand_1 < \delta \\ \{1,\ldots,N\}, & \text{otherwise} \end{cases} \tag{10}$$

$$P' = \begin{cases} P, & \text{if } rand_2 < 0.5 \\ \{1,\ldots,N\}/B(i), & \text{otherwise} \end{cases} \tag{11}$$

where $rand_1$ and $rand_2$ generate independently random numbers in the range $[0, 1]$, P remains the same as in MOEA/D-DE [26]. The definition of P' means that, if the niche count of individual i reaches the threshold β, it has a 50% chance to select individuals outside its neighborhood as the mating parents. This makes sense when MOEA/D pursues a high level of diversity but the neighboring solutions are very similar.

C. Framework of MOEA/D-TPN

For simplicity, we apply the DE operator [38] and polynomial mutation operator [6] to produce offspring in the proposed algorithm, which is the case with MOEA/D-DE [26]. The DE operator generates a candidate solution $\hat{y}$ by

$$\hat{y_k} = \begin{cases} x_k^{r_1} + F \times \left(x_k^{r_2} - x_k^{r_3}\right), & \text{with probability } CR \\ x_k^{r_1}, & \text{with probability } 1-CR \end{cases} \tag{12}$$

where $\hat{y_k}$ is the kth component of $\hat{y}$ and $x_k^{r_1}$, $x_k^{r_2}$, and $x_k^{r_3}$ are three distinct individuals randomly chosen from the population. CR and F are two control parameters.

The polynomial mutation produces a solution $y = (y_1,\ldots,y_n)$ from $\hat{y}$ as follows:

$$y_k = \begin{cases} \hat{y_k} + \sigma_k \times (b_k - a_k), & \text{with probability } p_m \\ \hat{y_k}, & \text{with probability } 1-p_m \end{cases} \tag{13}$$

with

$$\sigma_k = \begin{cases} (2 \times rand)^{\frac{1}{\eta+1}}, & \text{if } rand < 0.5 \\ 1-(2-2\times rand)^{\frac{1}{\eta+1}}, & \text{otherwise} \end{cases}$$

where $rand$ is a uniform random number from [0, 1]. The distribution index η and the mutation rate p_m are two control parameters. b_k and a_k are the lower and upper bounds of the kth decision variable, respectively.

The framework of MOEA/D-TPN is given in Algorithm 1. We would like to make the following remarks on the algorithm.

1) Since the distribution of the N weight vectors greatly affects the performance of MOEA/D, we use the WS-transformation scheme in [40] to generate a uniform spread of N weight vectors. Besides, as shown in line 12

Algorithm 1 MOEA/D-TPN

1: **Input**:
- *MaxIteration*: the stopping criterion
- M_r: the proportion of computational resources allocated for the first evolution phase
- N: the number of subproblems considered in MOEA/D
- T: the neighborhood size
- δ: the probability that parents are selected from the neighborhood
- n_r: the maximum number of solutions to be replaced by child solutions
- σ_{share}: the niche radius

2: **Output**: An approximated POF
3: **Initialization:** Generate a uniform spread of N weight vectors: $\lambda^1, \ldots, \lambda^N$ and then compute the T closest weight vectors to each weight vector by the Euclidean distance. For each $i = 1, \ldots, N$, set $B(i) = \{i_1, \ldots, i_T\}$ where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the T closest weight vectors to λ^i
4: Generate an initial population $x^1, \ldots, x^N$ by uniformly randomly sampling from the decision space
5: Set $FV^i := F(x^i)$, $EP_1 := \emptyset$, and $EP_2 := \emptyset$
6: **for** $gen := 1$ to *MaxIteration* **do**
7: **if** $gen == M_r * MaxIteration$ **then**
8: Calculate D_{mid} and D_{ext} by Eqs. (4) and (5), respectively.
9: **if** $D_{mid} < 0.9 D_{ext}$ **then**
10: Save the population in EP_1
11: Set the nadir point $r = \{r_1, \ldots, r_M\}$ where $r_j = max_{1 \le i \le N} f_j(x^i)$
12: Re-initialize the N weight vectors for the second evolution phase and re-calculate the T closest weight vectors to each weight vector
13: **end if**
14: **end if**
15: **for** $i := 1$ to N **do**
16: Calculate the niche count of individual i and set the mating/update range as P'' by Eq. (9)
17: Set $r_1 = i$ and randomly select two indexes r_2 and r_3 from P''
18: Apply the DE operator on individuals r_1, r_2 and r_3 by Eq. (12) to generate a solution $\bar{y}$, and perform the polynomial mutation operator on $\bar{y}$ by Eq. (13) to produce a new solution y
19: Update the reference point z if the evolution is in the first phase
20: Check if y is better than any individual x^j in P'' ($g(y|\lambda^j, z) \le g(x^j|\lambda^j, z)$ for the first phase and $g(y|\lambda^j, r) \ge g(x^j|\lambda^j, r)$ for the second phase). If y is better and no more than n_r individuals in P'' have been replaced, x^j is replaced by y
21: **end for**
22: **end for**
23: Save the final population in EP_2
24: Output the non-dominated solutions from $EP_1 \cup EP_2$

of Algorithm 1, at the beginning of the second phase, the N weight vectors should be reinitialized since the scalar subproblem form (i.e., the search direction) has changed. To obtain a uniform distribution of weight vectors for the reversed scalar subproblem form, we firstly use the simplex-lattice design to generate N points $\{w^1, \ldots, w^N\}$ on the hyperplane $W_1 + \cdots + W_M = 1$, then calculate the N search directions $\hat{w}^i (i = 1, \ldots, N)$ by $\hat{w}^i = (1 - w_1^i, \ldots, 1 - w_M^i)$. After that, the WS-transformation is applied on $\hat{w}^i (i = 1, \ldots, N)$ to generate the working weight vectors. Due to the variation of the weight vectors, we have to recalculate the T closest weight vectors to each weight vector (line 12).

2) In line 7, we set the proportion of the computational resources allocated for the first evolution phase to be M_r. Intuitively, a large M_r gives the algorithm a large proportion of computational resources to converge toward the POF and makes the estimated nadir point (line 11) more reliable. If M_r is too large, it may lead to the algorithm not having enough computational resources to execute the second-phase evolution. Besides, for easy-to-converge problems, $M_r = 0.5$ may be enough for estimating the crowdedness of the population as well as the nadir point. Accordingly, M_r is suggested to be in [0.5, 0.8]. In this paper, M_r is set to 0.7 based on some preliminary experiments.
3) In some cases, the algorithm has already achieved a set of well-distributed solutions along the target POF at the end of the first phase, but D_{mid} may be slightly smaller than D_{ext} (line 8), which gives an illusion that there are more solutions in the intermediate region than in the extreme region. To reduce the risk of being misled, D_{mid} is compared with 90% of the D_{ext} value when comparing the crowdedness of the two regions (line 9).
4) Similar to MOEA/D-DE [26], the proposed algorithm adopts the DE operator to produce new solutions (line 17). But it selects parent individuals in a reliable way that guarantees a wide exploration when the neighboring solutions are over-crowded. In other words, the niche-guided scheme helps maintain the population diversity.
5) In some sense, the second phase can be regarded as a stage of refining the already obtained solutions in the first phase since the second phase is expected to find a set of well-distributed solutions. Thus, in the second phase, instead of reinitializing the population, we use the final population of the first phase as the initial population. Besides, this will not interrupt the evolution process, which is quite important to the convergence performance of any MOEAs.
6) The algorithm uses two external population archives, EP_1 and EP_2, to store the approximated solutions in the two phases. Then, at the end of the whole evolution, nondominated solutions are identified from EP_1 and EP_2 and served as the final optimization results. Intuitively, EP_1 tends to keep intermediate solutions, while EP_2 is more likely to store boundary solutions.

D. Computational Cost of One Generation of MOEA/D-TPN

MOEA/D-TPN has the same framework as MOEA/D-DE [26], thus the increased computation cost is

TABLE I
TEST INSTANCES

Instance	Description	Domain	Number of Variables	Notes
F1	$f_1(x) = (1+g(x))x_1$ $f_2(x) = (1+g(x))(1-\sqrt{x_1})^5$ $g(x) = 2\sin(0.5\pi x_1)(n-1+\sum_{i=2}^{n}(y_i^2 - \cos(2\pi y_i)))$ where $y_{i=2:n} = x_i - \sin(0.5\pi x_i)$ POF: $f_2 = (1-\sqrt{f_1})^5$ POS: $x_i = \sin(0.5\pi x_i), i = 2,\ldots,n$	$[0,1]^n$	30	Uni-modal Convex Separable
F2	$f_1(x) = (1+g(x))(1-x_1)$ $f_2(x) = \frac{1}{2}(1+g(x))(x_1 + \sqrt{x_1}\cos^2(4\pi x_1))$ $g(x) = 2\sin(0.5\pi x_1)(n-1+\sum_{i=2}^{n}(y_i^2 - \cos(2\pi y_i)))$ where $y_{i=2:n} = x_i - \sin(0.5\pi x_i)$ POF: $f_2 = \frac{1}{2}(1-f_1+\sqrt{1-f_1}\cos^2(4\pi(1-f_1)))$ POS: $x_i = \sin(0.5\pi x_i), i = 2,\ldots,n$	$[0,1]^n$	30	Multi-modal Disconnected Separable
F3	$f_1(x) = (1+g(x))x_1$ $f_2(x) = \frac{1}{2}(1+g(x))(1-x_1^{0.1} + (1-\sqrt{x_1})^2\cos^2(3\pi x_1))$ $g(x) = 2\sin(0.5\pi x_1)(n-1+\sum_{i=2}^{n}(y_i^2 - \cos(2\pi y_i)))$ where $y_{i=2:n} = x_i - \sin(0.5\pi x_i)$ POF: $f_2 = \frac{1}{2}(1-f_1^{0.1} + (1-\sqrt{f_1})^2\cos^2(3\pi f_1))$ POS: $x_i = \sin(0.5\pi x_i), \forall x_i \in \mathbf{x_{II}}$	$[0,1]^n$	30	Multi-modal Disconnected Separable
F4	$f_1(x) = (1+g(x))(\frac{x_1}{\sqrt{x_2 x_3}})$ $f_2(x) = (1+g(x))(\frac{x_2}{\sqrt{x_1 x_3}})$ $f_3(x) = (1+g(x))(\frac{x_3}{\sqrt{x_1 x_2}})$ $g(x) = \sum_{i=4}^{n}(x_i - 2)^2$ POF: $f_1 f_2 f_3 = 1$ POS: $x_i = 2, i = 3,\ldots,n$	$[1,4]^n$	30	Uni-modal Convex Separable
UF4	$f_1(x) = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1} h(y_j)$ $f_2(x) = 1 - x_1^2 + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2} h(y_j)$ $y_j = x_j - \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2,\ldots,n$ $h(y_j) = \frac{\lvert y_j\rvert}{1+e^{2\lvert y_j\rvert}}, j = 2,\ldots,n$ where $J_1 = \{j \mid j \text{ is odd and } 2 \le j \le n\}, J_2 = \{j \mid j \text{ is even and } 2 \le j \le n\}$ POF: $f_1^2 + f_2 = 1$ POS: $x_j = \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2,\ldots,n$	$[0,1] \times [-2,2]^{n-1}$	10	Multi-modal Concave Non-separable
Convex DTLZ2	$f_1(x) = ((1+g(x))\cos(0.5\pi x_1)\cos(0.5\pi x_2))^4$ $f_2(x) = ((1+g(x))\cos(0.5\pi x_1)\sin(0.5\pi x_2))^4$ $f_3(x) = ((1+g(x))\sin(0.5\pi x_1))^2$ $g(x) = \sum_{i=3}^{n}(x_i - 0.5)^2$ POF: $\sqrt{f_1} + \sqrt{f_2} + f_3 = 1$ POS: $x_i = 0.5, i = 3,\ldots,n$	$[0,1]^n$	10	Uni-modal Convex Separable

attributed to its detection step for the TP scheme and calculation of the niche count for niche-guided mating selection. The detection step (line 7 in Algorithm 1) requires $O(MNT)$ computations. Reinitialization (line 11 in Algorithm 1) would also requires $O(MNT)$. Calculation of the niche count for each individual (line 15 in Algorithm 1) requires $O(MT)$ computations. Other operations have smaller complexity. Therefore, the computational cost of MOEA/D-TPN in each generation is $O(MNT)$.

IV. EXPERIMENTAL STUDY

A. Test Problems

To investigate the performance of MOEA/D-TPN on problems with complex POF shapes, we develop some new test instances (F1–F4), each of which has a nonconcave POF shape. Besides, two more commonly used test functions, UF4 [46] and convex DTLZ2 [8], are also included in the experiments. UF4 is a two-objective problem with a complicated POS shape, whilst the convex DTLZ2 is adapted from the original DTLZ2 problem [9], whose Pareto-optimal surface is almost flat at the edges, but changes sharply in the intermediate region. Deb and Jain [8] has shown in their experimental study that, on the convex DTLZ2, some MOEA/D variants fail to find points on the boundary of the POF. The details of these test instances are presented in Table I.

B. Parameter Settings

Four MOEA/D variants, i.e., MOEA/D, MOEA/D-N (i.e., MOEA/D with the niche scheme only), MOEA/D-TP (i.e., MOEA/D with the TP scheme only), and MOEA/D-TPN, are tested on the test instances. Note that, the term "MOEA/D" here refers to the MOEA/D-DE optimizer except that the WS-transformation method [40] is used for weight vector initialization. The control parameters for the compared algorithms were set the same as in [26]. In the last three variants, the sharing radius σ_{share} was set to 0.005, and the TP control parameter M_r was set to 0.7.

The population size in each algorithm was set to 200 for the two-objective instances and 300 for the three-objective instances. Each algorithm was executed 30 runs independently for each test instance on a computer with a configuration of Intel Core2 Duo CPU 2.4 GHz processor and 4 GB memory. The maximum number of generations was set to 500 for all the test instances.

C. Performance Metric

In our experimental studies, we adopt the following two widely used performance metrics.

1) *Inverted Generational Distance [49]:* Inverted generational distance can provide reliable information on both the diversity and convergence of obtained solutions. Let PF be a set of solutions uniformly sampled from the true POF, and PF* be the approximated solutions in the objective space, the metric measures the gap between PF* and PF, which is calculated as follows:

$$IGD\big(\mathrm{PF}^*, \mathrm{PF}\big) = \frac{\sum_{p\in \mathrm{PF}} d(p, \mathrm{PF}^*)}{|\mathrm{PF}|} \tag{14}$$

where $d(p, \mathrm{PF}^*)$ is the distance between the member p of PF and the nearest member of PF*.

2) *Hypervolume [48]:* Hypervolume metric measures the size of the objective space dominated by the approximated solutions S and bounded by a reference point $R = (R_1, \ldots, R_M)^T$ that is dominated by all points on the POF, and is computed by

$$\mathrm{HV}(S) = \mathrm{Leb}\left(\bigcup_{x\in S} [f_1(x), R_1] \times \cdots \times [f_M(x), R_M]\right) \tag{15}$$

where $Leb(A)$ is the Lebesgue measure of a set A. In our experiments, R is set to $(2.0, 2.0)^T$ for bi-objective test instances.

D. Comparison Among MOEA/D Variants

Figs. 2 and 3 plot the approximated POF achieved by each MOEA/D variant with the best IGD value on each test instance. F1 and F3 are problems with complex POF shapes of a sharp tail and a long tail. So, the use of TP evidently provides a good distribution in the extreme regions. F2 has several disconnected regions on the POF, and the original MOEA/D only finds a small part of solutions on the POF. Nevertheless, the use of the niche-guided scheme helps MOEA/D to solve this difficulty. It is also interesting that MOEA/D-TP also achieves a good approximation on this problem. One possible reason for this is that in the second phase, the weight vectors need to be reinitialized, which may cause the variation of search directions, and thus increase the population diversity. In Fig. 3, again, TP is very helpful in identifying boundary points for F4 and convex DTLZ2. UF4 is a concave problem, and the proposed TP scheme will not be activated in this case, which helps us focus analysing on the influence of the niche-guided scheme. Besides, UF4 also has strong dependencies between variables. Thus, to obtain a good approximation of this problem, an algorithm must be capable of maintaining population diversity. As shown in the middle column of Fig. 3, MOEA/D variants with the use of the niche-guided scheme have improved population diversity, thus achieving a better distribution of approximated solutions than those without this strategy. It can be also observed that TP is not activated in this case since MOEA/D-TPN and MOEA/D-N achieve similar results in Fig. 3 on UF4.

The plots of MOEA/D (which uses only WS-transformation) on F1, F3, and convex DTLZ2 in Figs. 2 and 3 also clearly indicate that a uniform distribution of weight vectors will not necessarily lead to a set of evenly distributed solutions on the POF. This may motivate researchers to seek other possible improvements like decomposition approaches, instead of pursuing a better distribution of weight vectors, to enhance MOEA/D.

Table II shows the results regarding the IGD and HV metrics. Specially, reference points $(5, 5, 5)^T$ and $(2, 2, 2)^T$ are selected for the computation of the HV metric for the F4 and convex DTLZ2 test problems, respectively. Best values are marked in boldface. For each instance, the two-sided Wilcoxon signed rank test at the 95% confidence level is performed between MOEA/D-TPN and each of the other compared algorithms. "+," "−," or "*" denotes that the performance of the compared algorithm is significantly better than, worse than, or equivalent to that of MOEA/D-TPN, and they are marked on the median values. It is clear that MOEA/D-TPN performs significantly better than MOEA/D and MOEA/D-N on all the test instances, and performs similarly to MOEA/D-TP on F3, F4, UF4, and convex DTLZ2. This further validates the efficiency of the proposed algorithm in solving complex problems tested in this paper.

E. Comparison With Peer Algorithms

To have a fair comparison, peer algorithms, such as MOEA/D-NBI [45] and MOEA/D with objective transform (TMOEA/D) [30], that are specially designed for solving MOPs with complex POF geometries should be tested in our experiments. Since MOEA/D-NBI is only applicable to bi-objective cases, we choose F1–F3, and UF4 as the test instances in this part. The parameter settings of each algorithm are derived from the referenced paper.

The statistical results of the three algorithms are presented in Table III. It can be clearly seen from the table that MOEA/D-TPN significantly outperforms the other two algorithms on the four tested problems in terms of IGD and HV. For TMOEA/D, the poor performance can be attributed to that the shapes of the test instances are extremely irregular, and TMOEA/D may not be able to accurately identify them, thus it cannot achieve a good approximation. For MOEA/D-NBI, apart from the extremely irregular POF geometries of the test problems, the lack of effective diversity maintenance strategy may be another reason for its poor performance. Nevertheless, MOEA/D-TPN considers the distribution of solutions and uses TP to adjust search directions. Accordingly, it achieves a better approximation for each instance. To have a better understanding of differences among the three different kinds of algorithms, we also plot the approximations obtained by TMOEA/D and MOEA/D-NBI for F1, F2, and UF4 in Fig. 4.

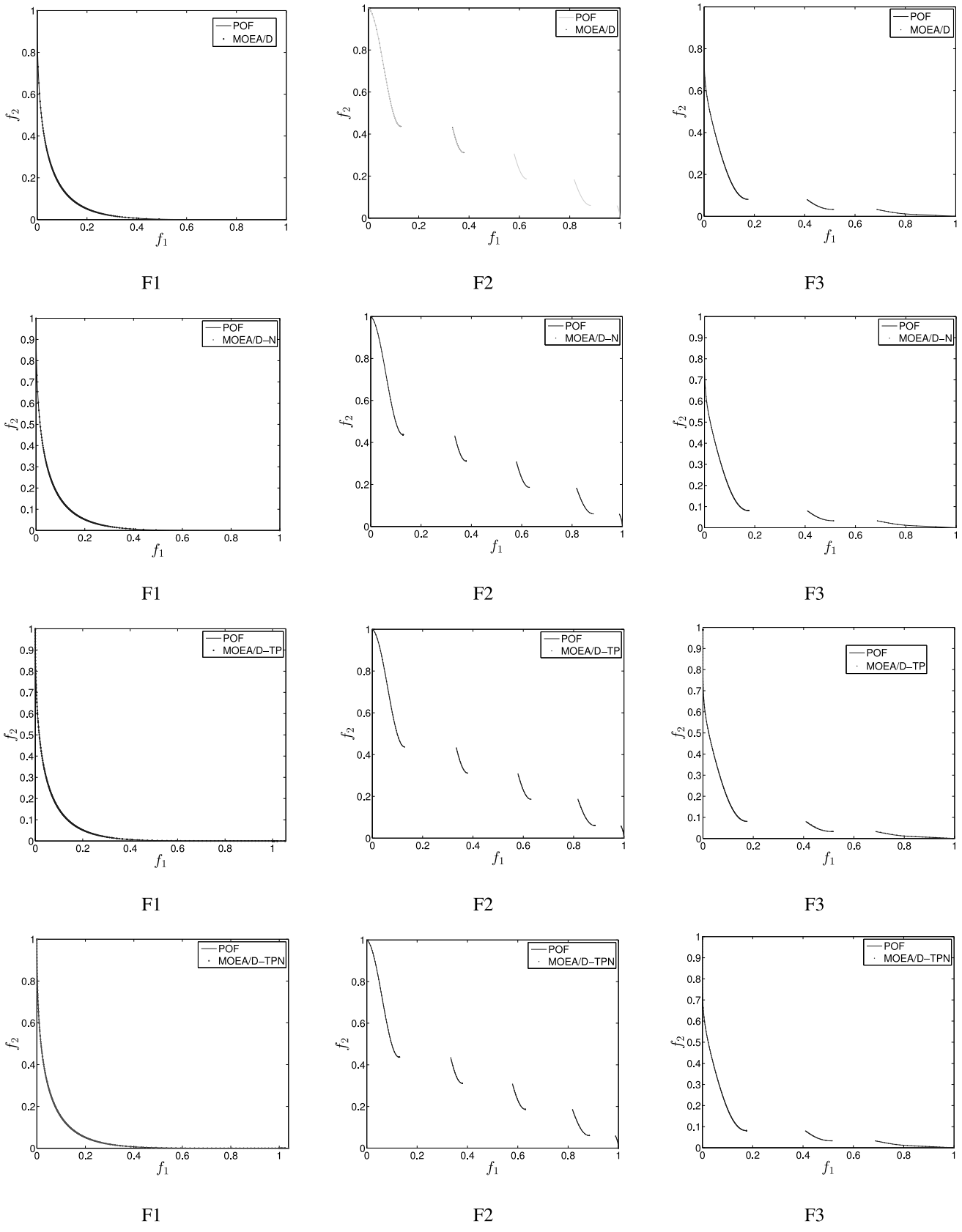

Fig. 2. Approximated POFs with the lowest IGD values among 30 runs on F1–F3.

F. Comparison With Other Algorithms

In this section, SPEA2+SDE [24] and NSGA-III [8] are used for comparison. The algorithms compared are newly-developed techniques and have shown to be very efficient for solving MOPs. To increase difficulties for these algorithms, four extra test problems with complex POF shapes

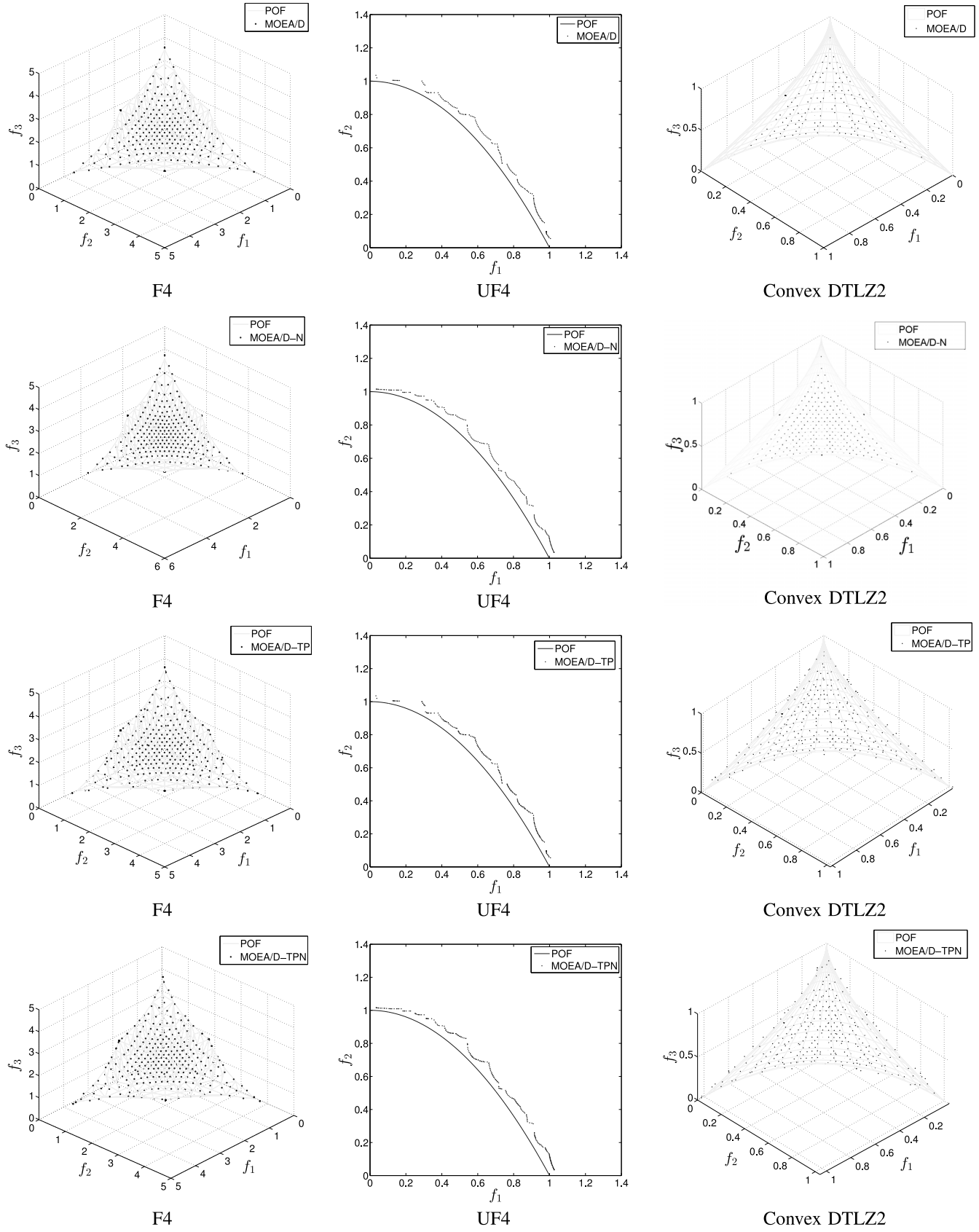

Fig. 3. Approximated POFs with the lowest IGD values among 30 runs on F4, UF4, and convex DTLZ2.

are used in the experiment, which are presented in Table IV. POL is derived from Poloni's study [39], whose POF is discontinuous and has a long tail. mF4 is a modified version of F4, and is more difficult for MOEAs to find boundary solutions on the POF since both the search space and the objective space have been scaled. The modification leads to

TABLE II
BEST, MEDIAN, AND WORST IGD AND HV VALUES OF THE FOUR ALGORITHMS ON THE TEST PROBLEMS

	IGD				HV			
Prob.	MOEA/D	MOEA/D-N	MOEA/D-TP	MOEA/D-TPN	MOEA/D	MOEA/D-N	MOEA/D-TP	MOEA/D-TPN
F1	4.4032E-02	4.4105E-02	2.8330E-03	**2.8260E-03**	3.9568	3.9564	3.9568	**3.9592**
	(4.4287E-02)$^-$	(4.4278E-02)$^-$	(2.8465E-03)$^-$	**2.8455E-03**	(3.9508)$^-$	(3.9500)$^-$	(3.9496)$^-$	**3.9536**
	4.4504E-02	4.4482E-02	2.8670E-03	**2.8610E-03**	3.9432	3.9400	3.9424	**3.9436**
F2	2.2880E-03	2.2830E-03	1.6630E-03	**1.6480E-03**	3.6968	3.6964	3.6968	**3.6986**
	(2.2930E-03)$^-$	(2.2890E-03)$^-$	(1.6995E-03)$^-$	**1.6780E-03**	(3.6744)$^-$	(3.6734)$^-$	(3.6742)$^-$	**3.6774**
	1.5622E-01	2.3370E-03	1.5570E-01	**1.8000E-03**	3.5120	3.6588	3.5184	**3.6608**
F3	1.1038E-02	1.0936E-02	2.2620E-03	**2.2344E-03**	3.9224	3.9272	3.9252	**3.9280**
	(1.1181E-02)$^-$	(1.1245E-02)$^-$	(**2.3545E-03**)*	2.3835E-03	(3.9146)$^-$	(3.9176)$^-$	(3.9195)*	**3.9198**
	1.1739E-02	1.3015E-01	**2.4950E-03**	2.6090E-03	3.9012	3.8724	**3.9032**	3.9024
F4	8.3978E-02	8.4353E-02	7.5811E-02	**7.1559E-02**	99.8375	100.1625	**100.9213**	100.9125
	(8.4398E-02)$^-$	(8.4962E-02)$^-$	(**7.6760E-02**)*	7.6872E-02	(98.9938)$^-$	(98.6813)$^-$	(**99.9688**)*	99.8337
	8.5074E-02	8.5784E-02	7.8385E-02	**7.8119E-02**	98.2250	98.1250	99.0000	**99.0250**
UF4	3.6898E-02	3.5806E-02	3.6853E-02	**3.5704E-02**	3.2328	3.2272	3.2312	**3.2416**
	(4.2410E-02)$^-$	(4.2400E-02)$^-$	(**4.2323E-02**)*	4.2361E-02	(3.1866)$^-$	(3.1932)$^-$	(3.1934)*	**3.1946**
	4.8510E-02	4.8282E-02	4.8409E-02	**4.8264E-02**	3.1524	3.1536	3.1512	**3.1604**
Convex DTLZ2	9.1408E-02	8.9932E-02	**2.8870E-02**	2.9198E-02	7.9096	7.9232	7.9688	**7.9696**
	(9.3433E-02)$^-$	(9.3486E-02)$^-$	(3.3324E-02)*	**3.2598E-02**	(7.8932)$^-$	(7.9020)$^-$	(7.9544)*	**7.9572**
	9.6259E-02	9.5988E-02	3.5477E-02	**3.5458E-02**	7.8696	7.8856	7.9400	**7.9472**

TABLE III
BEST, MEDIAN, AND WORST IGD AND HV VALUES OF THE PEER ALGORITHMS ON FOUR TEST PROBLEMS

	IGD			HV		
Prob.	TMOEA/D	MOEA/D-NBI	MOEA/D-TPN	TMOEA/D	MOEA/D-NBI	MOEA/D-TPN
F1	3.4720E-03	2.8690E-03	**2.8260E-03**	3.9592	3.9563	**3.9592**
	(3.8615E-03)$^-$	(2.8720E-03)$^-$	**2.8455E-03**	(3.9514)$^-$	(3.9496)$^-$	**3.9536**
	4.5490E-03	2.8800E-03	**2.8610E-03**	3.9432	3.9428	**3.9436**
F2	2.9960E-03	**1.6410E-03**	1.6480E-03	3.6960	3.7008	**3.6986**
	(4.2245E-03)$^-$	(1.9650E-03)$^-$	**1.6780E-03**	(3.6720)$^-$	(3.6744)$^-$	**3.6774**
	1.8772E-02	1.5566E-01	**1.8000E-03**	3.6532	3.5316	**3.6608**
F3	3.4610E-03	2.4550E-03	**2.2344E-03**	3.9280	3.9272	**3.9280**
	(4.3285E-03)$^-$	(2.6590E-03)$^-$	**2.3835E-03**	(3.9174)$^-$	(3.9174)$^-$	**3.9198**
	1.3451E-02	2.9640E-03	**2.6090E-03**	3.8996	3.9016	**3.9024**
UF4	4.9577E-02	3.8695E-02	**3.5704E-02**	3.2136	3.2402	**3.2416**
	(6.4952E-02)$^-$	(4.2808E-02)$^-$	**4.2361E-02**	(3.1356)$^-$	(3.1894)$^-$	**3.1946**
	8.7681E-02	4.9567E-02	**4.8264E-02**	3.0544	3.0812	**3.1604**

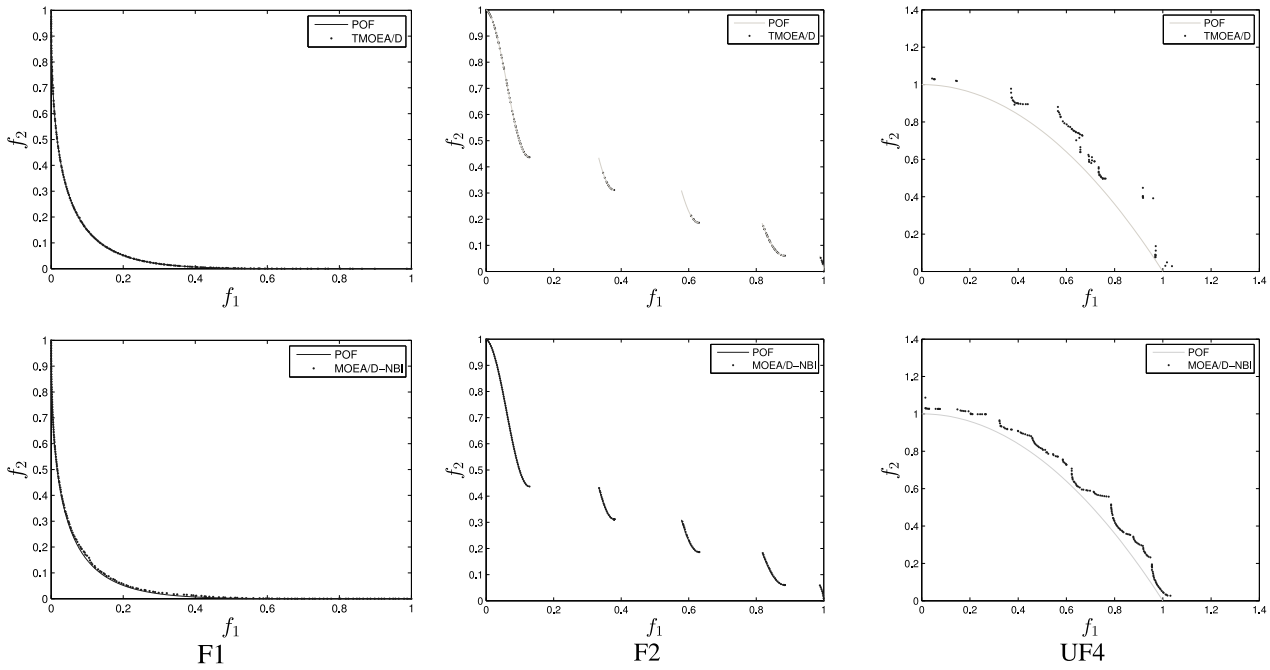

Fig. 4. Approximated POFs with the lowest IGD values among 30 runs on F1, F2, and UF4.

the variation of the objective scale, i.e., $f_i \in [0.1, 10], i = 1, 2, \ldots, M$. F5 has an irregular POF shape and poses difficulties for MOEAs to approximate extreme regions. F6 is a multimodal convex problem and has deceptive property. It challenges algorithms in finding a global POF as well as extreme regions. In other words, the four test functions from

TABLE IV
EXTRA TEST INSTANCES

Instance	Description	Domain	Number of Variables	Notes
POL	$f_1(x) = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2$ $f_2(x) = (x_1 + 3)^2 + (x_2 + 1)^2$ $A_1 = 0.5\sin(1) - 2\cos(1) + \sin(2) - 1.5\cos(2)$ $A_2 = 1.5\sin(1) - \cos(1) + 2\sin(2) - 0.5\cos(2)$ $B_1 = 0.5\sin(x_1) - 2\cos(x_1) + \sin(x_2) - 1.5\cos(x_2)$ $B_2 = 1.5\sin(x_1) - \cos(x_1) + 2\sin(x_2) - 0.5\cos(x_2)$	$[-\pi, \pi]^n$	2	Uni-modal Non-convex Non-separable
mF4	$f_1(x) = (1 + g(x))(\frac{x_1}{\sqrt{x_2 x_3}})$ $f_2(x) = (1 + g(x))(\frac{x_2}{\sqrt{x_1 x_3}})$ $f_3(x) = (1 + g(x))(\frac{x_3}{\sqrt{x_1 x_2}})$ $g(x) = \sum_{i=4}^{n} (x_i - 5)^2$ POF: $f_1 f_2 f_3 = 1$ POS: $x_i = 5, i = 3, \ldots, n$	$[1, 10]^n$	30	Uni-modal Convex Separable
F5	$f_1(x) = (1 + g(x))\,((1 - x_1)x_2)$ $f_2(x) = (1 + g(x))\,(x_1(1 - x_2))$ $f_3(x) = (1 + g(x))\,(1 - x_1 - x_2 + 2x_1 x_2)^6$ $g(x) = \sum_{i=3}^{n} (x_i - 0.5)^2$ POF: $f_3 = (1 - f_1 - f_2)^6$ POS: $x_i = 0.5, i = 3, \ldots, n$	$[0, 1]^n$	30	Uni-modal Convex Non-separable
F6	$f_1(x) = \cos^4(0.5\pi x_1)\cos^4(0.5\pi x_2)$ $f_2(x) = \cos^4(0.5\pi x_1)\sin^4(0.5\pi x_2)$ $f_3(x) = \left(\frac{1+g(x)}{1+\cos^2(0.5\pi x_1)}\right)^{\frac{1}{1+g(x)}}$ $g(x) = \frac{1}{10}\sum_{i=3}^{n}(1 + x_i^2 - \cos(2\pi x_i)))$ POF: $f_3(1 + \sqrt{f_1} + \sqrt{f_2}) = 1$ POS: $x_i = 0, i = 3, \ldots, n$	$[0, 1]^n$	30	Multi-modal Convex Separable

TABLE V
BEST, MEDIAN, AND WORST IGD, HV, AND T (SECONDS) VALUES OF THE THREE ALGORITHMS ON THE EXTRA TEST PROBLEMS

	IGD			HV			T		
Prob.	SPEA2+SDE	NSGA-III	MOEA/D-TPN	SPEA2+SDE	NSGA-III	MOEA/D-TPN	SPEA2+SDE	NSGA-III	MOEA/D-TPN
POL	7.9373E-02	1.4470E-01	**5.6305E-02**	5.3820E+02	5.3535E+02	**5.3832E+02**	2.0201E+02	3.9143E+02	**6.9348E+01**
	(1.0496E-01)$^-$	(2.0711E-01)$^-$	**5.9582E-02**	(5.3589E+02)$^-$	(5.3535E+02)$^-$	**5.3646E+02**	(2.0550E+02)$^-$	(4.2283E+02)$^-$	**7.3065E+01**
	1.7027E-01	2.2150E-01	**6.1822E-02**	5.3220E+02	5.3106E+02	**5.3256E+02**	2.2123E+02	6.2033E+02	**8.1042E+01**
mF4	1.4703E+00	1.4805E-01	**1.2396E-01**	1.4285E+03	1.6610E+03	**1.6665E+03**	3.5728E+02	4.5907E+02	**7.5071E+01**
	(1.4738E+00)$^-$	(1.5435E-01)$^-$	**1.2535E-01**	(1.4160E+03)$^-$	(1.6533E+03)$^-$	**1.6599E+03**	(3.6065E+02)$^-$	(5.7852E+02)$^-$	**8.6718E+01**
	1.4783E+00	1.6239E-01	**1.2754E-01**	1.4030E+03	1.6440E+03	**1.6520E+03**	4.2019E+02	6.4615E+02	**9.4585E+01**
F5	1.8505E-01	2.5032E-02	**1.3646E-02**	7.7864E+00	7.9840E+00	**7.9832E+00**	4.5921E+02	6.0510E+02	**7.2329E+01**
	(1.8536E-01)$^-$	(2.7036E-02)$^-$	**1.4040E-02**	(7.7364E+00)$^-$	(7.9760E+00)$^-$	**7.9770E+00**	(4.7347E+02)$^-$	(6.1632E+02)$^-$	**8.0427E+01**
	1.8613E-01	2.9269E-02	**1.4545E-02**	7.6664E+00	7.9704E+00	**7.9714E+00**	5.1030E+02	7.9555E+02	**9.1976E+01**
F6	3.3742E-02	4.4938E-01	**1.6892E-02**	6.0536E+00	3.8400E+00	**6.0656E+00**	4.3947E+02	4.8475E+02	**7.7635E+01**
	(4.8208E-02)$^-$	(5.1625E-01)$^-$	**1.9734E-02**	(5.9720E+00)$^-$	(3.7672E+00)$^-$	**6.0012E+00**	(4.5637E+02)$^-$	(5.0677E+02)$^-$	**7.8930E+01**
	6.6152E-02	5.4934E-01	**2.4442E-02**	5.8832E+00	3.7000E+00	**5.9440E+00**	4.7875E+02	6.7858E+02	**9.9403E+01**

Table IV are more challenging than those used in our previous experiment, imposing various effects on the performance of MOEAs. Specially, reference points $(20, 30)^T$, $(12, 12, 12)^T$, and $(2, 2, 2)^T$ are selected for the HV calculation of POL, mF4, and another three-objective problems, respectively. The parameter settings in each algorithm are derived from the referenced paper.

Table V gives the statistical results of the IGD, HV, and T values on these test functions, where "T" represents the run-time of an algorithm, recorded in seconds. It can be observed that, for the tested problems, MOEA/D-TPN achieves significantly better results than the other two algorithms in terms of IGD and HV. Specifically, judging from the IGD values, SPEA2+SDE struggles to converge toward the POF on mF4 and F5, while NSGA-III performs poorly on POL and F6. As for the run-time performance, MOEA/D-TPN clearly exceeds the other two algorithms by a significant margin, SPEA2+SDE ranks the second, and NSGA-III ranks the last.

The results in Fig. 5 shows the approximated POF of the lowest IGD value over 30 runs obtained by each algorithm. It is clear to see that MOEA/D-TPN is able to converge to the POFs and cover the entire POFs of the first three test functions, and finds most of solutions for F6. On F6, SPEA2+SDE loses the boundary areas, while MOEA/D-TPN keeps most of them. However, MOEA/D-TPN loses extreme areas, while SPEA2+SDE keeps well. While SPEA2+SDE performs well on POL and F6, it struggles to converge to the POFs of mF4 and F5. NSGA-III achieves fair results on POL, mF4, and F5, but it fails to cover the whole extreme regions of these problems. NSGA-III performs poorly and obtains many points on a local POF surface on F6 since this a deceptive

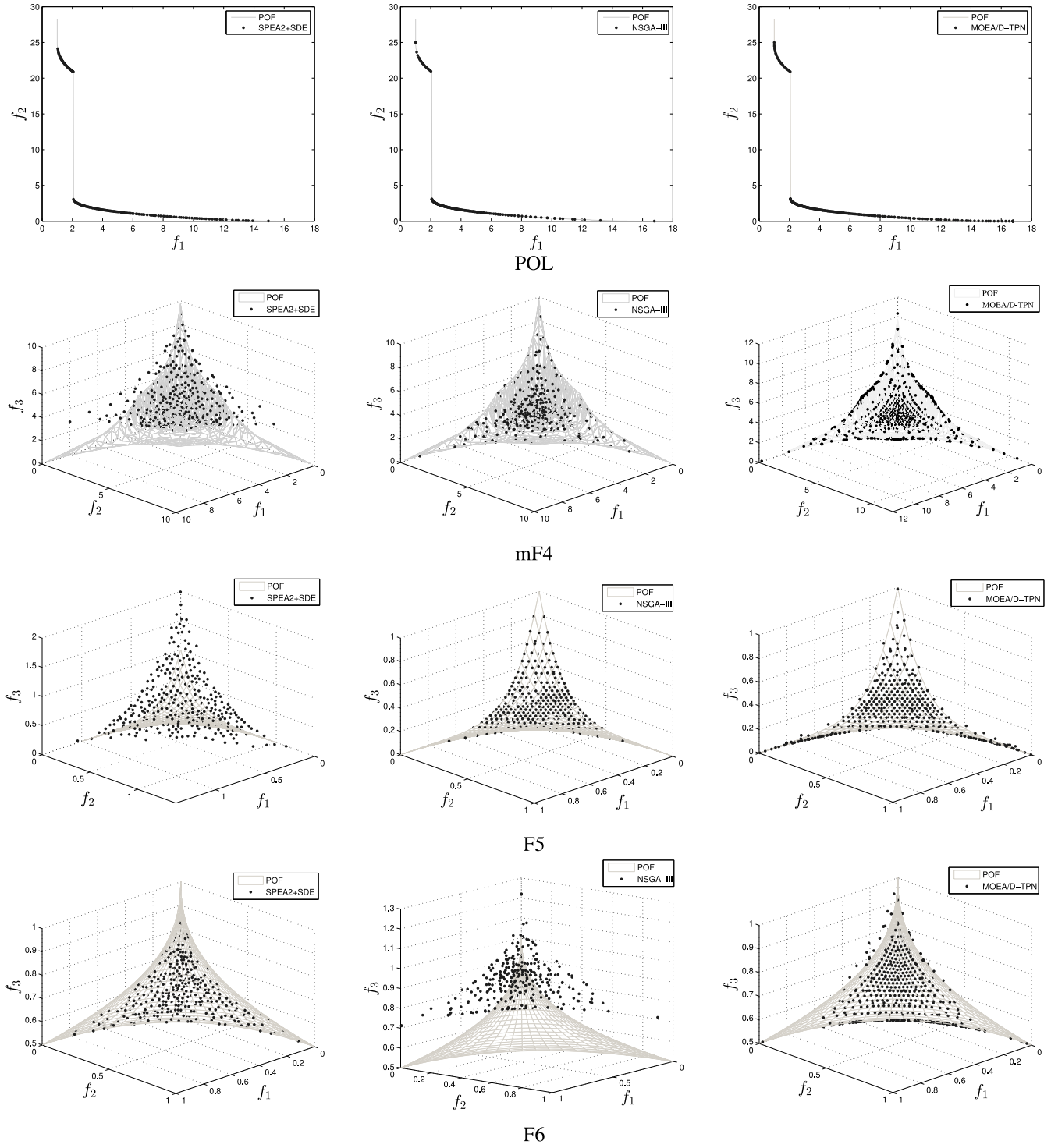

Fig. 5. Approximated POFs with lowest IGD values among 30 runs on POL, mF4, F5, and F6.

problem with a complex POF, but the other algorithms have successfully found many global solutions on this problem.

Fig. 6 shows the evolution curves of the average IGD values for the test problems against generation. Clearly, all the algorithms have a fast convergence performance on the test functions. Meanwhile, a notable improvement of the IGD values can be observed in the improved MOEA/D when TP is activated (at the 350th generation), leading to MOEA/D-TPN achieving better late-stage performance than SPEA2+SDE and NSGA-III on this problems.

V. Discussion, Limitation, and Future Research

A. Further Investigation of TP

As has been shown in the previous experimental study, the TP scheme evidently improves MOEA/D by providing a better set of evenly-distributed solutions for the tested problems.

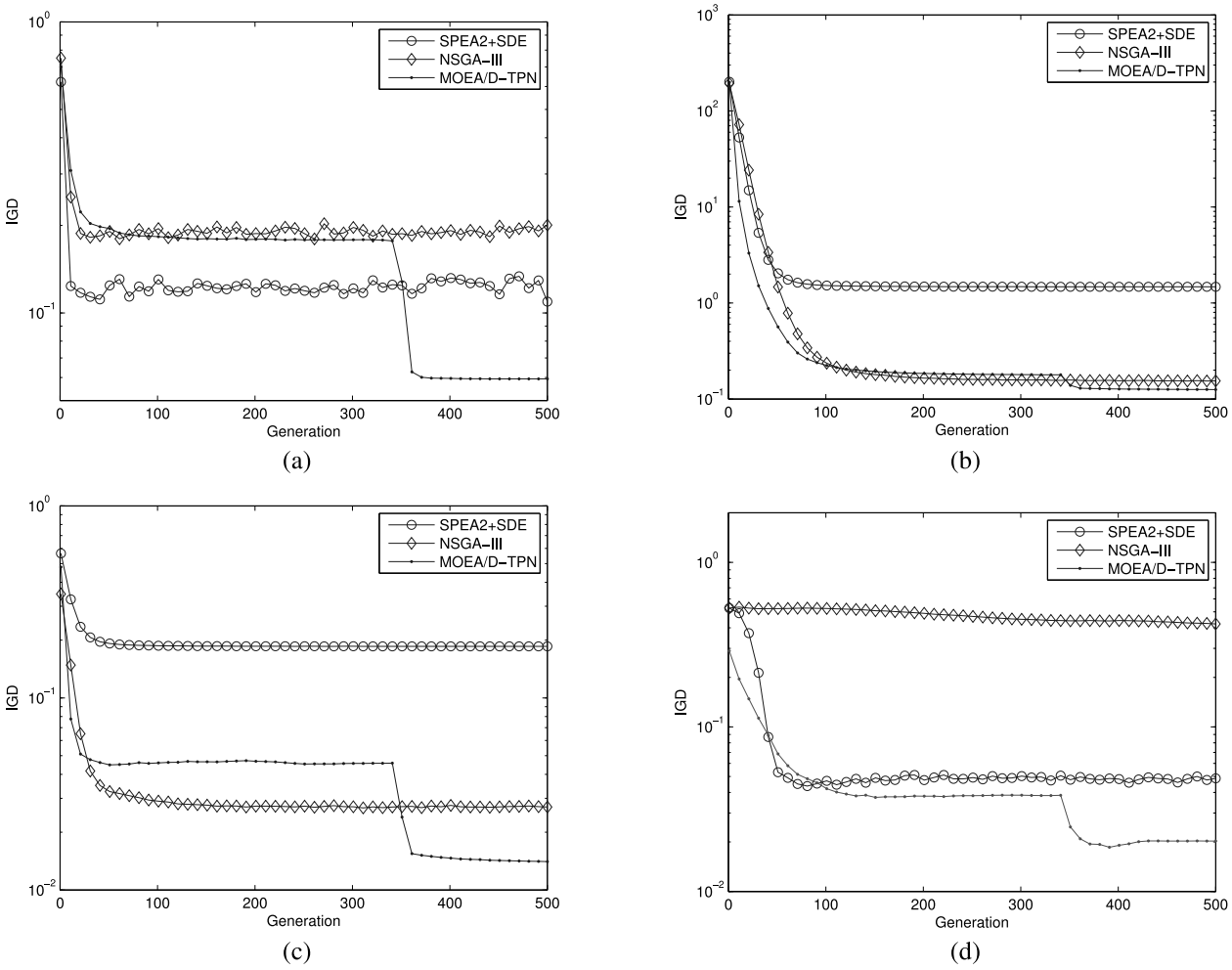

Fig. 6. Evolution of the mean IGD values of the test problems. (a) POL. (b) mF4. (c) F5. (d) F6.

There are two basic issues involved in TP, however, which may influence the performance of the proposed algorithm. One is the definitions of the intermediate and extreme regions, since the definitions are based on the continuous POF, we would wonder what will happen to them for disconnected problems. The other one is whether TP has an influence on population diversity. Therefore, we carried out the following experiments.

1) Influence of Different Types of POF on D_{mid} and D_{ext}: To have a better understanding of the definition and function of D_{mid} and D_{ext}, we use F1–F3 as the test problems, which represent two different types of problems. All the parameter settings of MOEA/D-TPN remain unchanged in the experiment. The crowdedness values of D_{mid} and D_{ext} against generation are plotted in Fig. 7, where V_s denotes a stable value obtained by D_{ext} in the first phase (350 generations). The figure clearly shows that, for both continuous and discontinuous problems, D_{mid} is smaller than D_{ext} when they reach their stable level, and the introduction of the second phase evolution helps find more extreme solutions, thus narrowing the gap between their crowdedness values. This indicates, for disconnected problems, the definitions of intermediate and extreme regions are also very meaningful and helpful. In the early stage (roughly 30 generations), especially shown for F2 in Fig. 7, D_{mid} seems to be higher than D_{ext}, and this is probably because the level of extreme solutions is too low, and many extreme individuals of W_e share the same solutions, leading to a small value of D_{ext}. It is worth noting that, the figure also confirms that parameter 0.9 is able to determine whether there is a significant difference between D_{mid} and D_{ext}.

2) Influence of TP on Population Diversity: As discussed in Section IV-D, TP not only helps in finding extreme solutions, but seems to have the potential to increase population diversity for F2. To further investigate the diversity performance of TP, we tested MOEA/D + TP on F2. Fig. 8 shows the population at the 350th, 360th, 370th, and 380th generations. We can observe that, at the end of the first phase (350th generation), the algorithm only finds solutions on the top left of the POF. When TP is triggered, the population gradually covers the disconnected subregions on the bottom right of the POF, which means TP helps diversify the population across the whole POF. One possible reason for this is that search directions have been changed in the second phase, which in return increases the chance for the population to explore new regions, thus providing a better distribution and coverage of solutions.

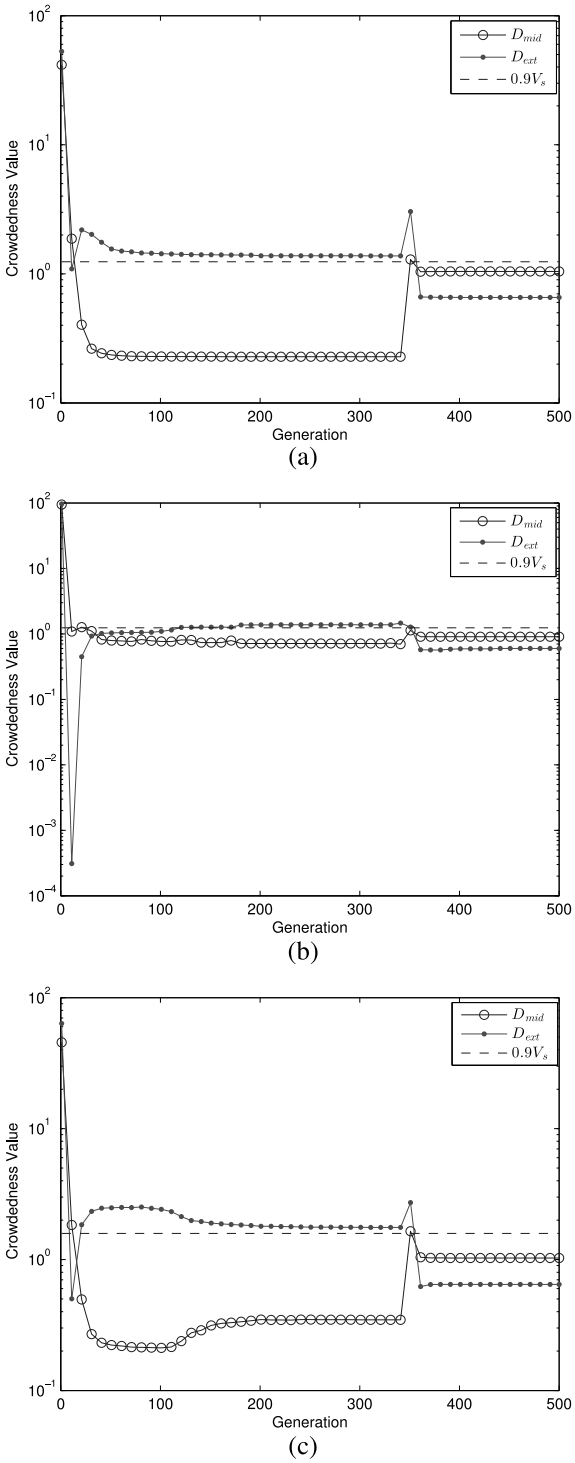

Fig. 7. Evolution curves of D_{mid} and D_{ext} on (a) F1, (b) F2, and (c) F3.

B. Effect of the Parameter M_r

The previous experimental results have shown that the TP scheme can improve the performance of MOEA/D. To further investigate the effect of the TP control parameter M_r on the performance of MOEA/D-TPN, six values of M_r (i.e., 0.5, 0.6, 0.7, 0.8, 0.9, 1) are tested on the instances F1 and F3. Note that, $M_r = 1.0$ means that the algorithm does not use the TP scheme. All the other parameters remain the same as in Section IV-B.

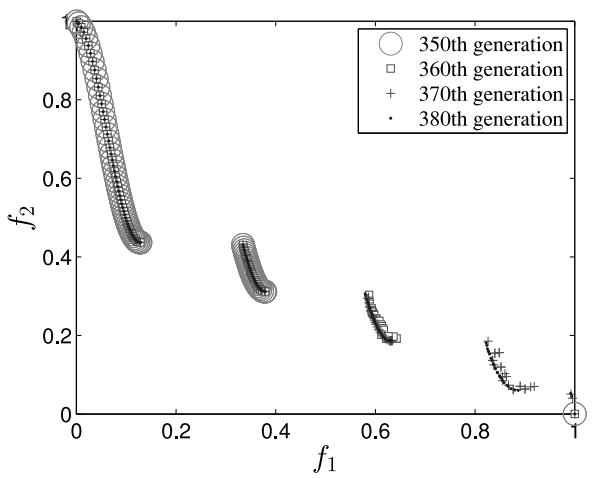

Fig. 8. Influence of TP on population diversity for F2.

The IGD values are shown in Table VI. It can be clearly seen that $M_r = 0.7$ produces better results than the other settings for F1 and F3. Besides, the IGD values further demonstrate that the use of TP provides a notable improvement (different IGD values of $M_r = 1$ and $M_r < 1$) in approximating the POFs of F1 and F3.

C. Effect of the Sharing Radius σ_{share}

In the proposed algorithm, the sharing radius σ_{share} in the sharing function is an important factor. A large σ_{share} will reduce the impact of the niche-guided scheme, while a small one may lead to too much exploration in the search space and may not ensure a fast convergence of the population. In the following, σ_{share} is set to 0.001, 0.005, 0.01, 0.05, 0.2, and 0.5, and MOEA/D-N is tested on F2 and UF4, both of which have shown to be sensitive to σ_{share} in the previous study. Note that, we do not use MOEA/D-TPN here because TP helps to solve the problem, which may shadow the importance of the niche-guided scheme. So, to avoid this situation, it is better to investigate the effect of the sharing radius separately.

Table VII gives the effect of varying the sharing radius σ_{share}. Clearly, high values of σ_{share} deteriorate IGD values because the algorithm is given more chance to produce a new individual by selecting neighboring parent individuals, while a low value will make the algorithm concentrate on global exploration and cannot provide sufficient convergence performance. Notably, setting the value of σ_{share} to 0.005 offers better results on the tested instances F2 and UF4.

D. Effect of the Population Size

The population size (N) is another key factor that greatly affects the performance of algorithms. To investigate its influence, we test three levels of population sizes for the algorithms SPEA2+SDE, NSGA-III, and MOEA/D-TPN on the problems POL and mF4. Table VIII shows the population size settings for these algorithms. Since both NSGA-III and MOEA/D-TPN use the simplex-lattice design to set the number of reference points or decomposed sub-problems, their population sizes should satisfy the requirement $N = C_{M+H-1}^{H}$, where H is the

TABLE VI
IGD VALUES OBTAINED BY MOEA/D-TPN WITH DIFFERENT M_r SETTINGS FOR THE INSTANCES F1 AND F3

	F1			F3		
M_r	Best	Median	Worst	Best	Median	Worst
0.5	2.8338E-03	2.8487E-03	2.8652E-03	2.3177E-03	2.5467E-03	2.7673E-03
0.6	2.8328E-03	2.8476E-03	2.8664E-03	2.2376E-03	2.5282E-03	2.7168E-03
0.7	2.8256E-03	2.8455E-03	2.8614E-03	2.2342E-03	2.3835E-03	2.6090E-03
0.8	2.8272E-03	2.8506E-03	2.8662E-03	2.2667E-03	2.3880E-03	2.5654E-03
0.9	2.8367E-03	2.8564E-03	2.8911E-03	2.2795E-03	2.4364E-03	2.6435E-03
1	4.4105E-02	4.4278E-02	4.4482E-02	1.0936E-02	1.1245E-02	1.3015E-01

TABLE VII
IGD VALUES OBTAINED BY MOEA/D-N WITH DIFFERENT σ_{share} SETTINGS FOR THE INSTANCES F2 AND UF4

	F2			UF4		
σ_{share}	Best	Median	Worst	Best	Median	Worst
0.001	2.2851E-03	2.2935E-03	2.3380E-03	4.0583E-02	4.4081E-02	4.8602E-02
0.005	2.2832E-03	2.2891E-03	2.3370E-03	3.5806E-02	4.2000E-02	4.8282E-02
0.01	2.2857E-03	2.2931E-03	2.3401E-03	3.9485E-02	4.2244E-02	4.8336E-02
0.05	2.3556E-03	2.5442E-03	2.8170E-03	3.9932E-02	4.3017E-02	4.8391E-02
0.2	3.2096E-03	4.0473E-03	6.5844E-03	4.1503E-02	4.3274E-02	4.8961E-02
0.5	4.3035E-03	4.9096E-03	1.3758E-02	4.1904E-02	4.3827E-02	5.0084E-02

TABLE VIII
POPULATION SIZE SETTINGS FOR THREE ALGORITHMS

	POL			mF4		
Level	SPEA2+SDE	NSGA-III	MOEA/D-TPN	SPEA2+SDE	NSGA-III	MOEA/D-TPN
1	52	52	50	48	48	45
2	100	100	100	92	92	91
3	200	200	200	192	192	190

TABLE IX
BEST, MEDIAN, AND WORST IGD VALUES OF THREE ALGORITHMS ON POL AND MF4

	POL			mF4		
Level	SPEA2+SDE	NSGA-III	MOEA/D+TPN	SPEA2+SDE	NSGA-III	MOEA/D+TPN
1	1.8971E-01	3.1802E-01	**1.8071E-01**	1.5419E+00	**3.5911E-01**	4.2461E-01
	2.6678E-01	5.3476E-01	**2.0400E-01**	1.5695E+00	**3.7089E-01**	4.4307E-01
	3.6169E-01	6.0683E-01	**2.6377E-01**	1.6279E+00	**3.8630E-01**	4.7349E-01
2	1.4821E-01	2.4915E-01	**9.4054E-02**	1.4931E+00	2.6242E-01	**2.5146E-01**
	1.8373E-01	3.2066E-01	**1.0687E-01**	1.5119E+00	2.7067E-01	**2.6229E-01**
	2.4422E-01	3.6438E-01	**1.2287E-01**	1.5232E+00	2.7626E-01	**2.6902E-01**
3	7.9373E-02	1.4470E-01	**5.6305E-02**	1.4775E+00	1.7747E-01	**1.6328E-01**
	1.0496E-01	2.0711E-01	**5.9582E-02**	1.4795E+00	1.8393E-01	**1.6476E-01**
	1.7027E-01	2.2150E-01	**6.1822E-02**	1.4842E+00	1.8544E-01	**1.6714E-01**

number of divisions on each objective and M is the number of objectives. The population size for NSGA-III is set as the smallest multiple of four higher than N, which is referenced to the setting in [8]. SPEA2+SDE has the same population size setting as NSGA-III.

Table IX gives the IGD results of the three algorithms at different population size levels. It can be observed that, for POL, MOEA/D-TPN outperforms the rest on each level of the population size by a clear margin. When the population size is too small, NSGA-III obtains better IGD values for mF4 than MOEA/D-TPN and SPEA2+SDE, but it is gradually exceeded by MOEA/D-TPN with an increase in the population size. This implies MOEA/D-TPN work well if a normal size of population is available for MOEA/D-TPN.

E. Limitations and Future Research

The study has two main limitations. The first is that there are some parameters involved in MOEA/D-TPN, and it is not easy for users to set in advance without any knowledge about a problem. For example, the setting of the control parameter M_r used in the TP scheme may vary with the difficulty of the problem considered. On one hand, for easy problems, $M_r = 0.5$ may be enough to drive most of individuals toward the true POF, but for difficult problems, $M_r = 0.5$ may not be enough to obtain an accurate estimation of population crowdedness and the nadir point. On the other hand, a large M_r is good for estimating population crowdedness, but may cause the algorithm to have few computational resources in the second phase, thus unable to have a good approximation at the end of the evolution. For these reasons, M_r is suggested to be in [0.5, 0.8]. Note that, the TP scheme is really helpful for complex problems in question and a value of 0.7 for M_r generates good results for them in this paper. Another key parameter is σ_{share}, and its setting relates to the number of objectives and the population size. $\sigma_{\text{share}} = 0.005$ may not be the best choice for all the problems, however, the experimental

study has seen an improvement in MOEA/D with the use of niche. The second limitation is this paper mainly focuses on bi- and three-objective complex problems, which may induce some bias. Although MOEA/D-TPN has achieved encouraging performance on the problems in question, that does not imply MOEA/D-TPN is always better than the other compared algorithms. Further investigations are required to have a better understanding of its performance.

As a first step, the analysis has been focusing on low-dimensional test problems. Future work will extend the improved MOEA/D to many-objective optimization and dynamic multiobjective optimization. We will also consider to develop a self-adaptive version of MOEA/D-TPN that is less sensitive to parameter settings and even free from key parameters. Another interesting research line will be to design a smart version of MOEA/D-TPN, where the solutions on the POF are smart distributed [13], [14], [35]. In other words, we will emphasize the regions of the POF that entail significant tradeoff, and de-emphasizes the regions corresponding to little tradeoff. Besides, the issue on how to set a arbitrary population size of MOEA/D in high-dimensional problems will be addressed in our further work.

VI. Conclusion

An extensive review of the performance of MOEA/D has shown that the MOEA/D algorithms tend to find solutions located in the intermediate region of an irregular POF and have difficulty in solving problems with complex POF shapes, i.e., the POF with a sharp peak and long tail, and disconnected regions. This paper has extended the MOEA/D framework to overcome this drawback by introducing a novel TP scheme and a niche-guided scheme. While TP alternatively divides the whole optimization procedure, according to the crowdedness of solutions on the POF, into two phases, and helps the population in finding points on the boundary of the POF in the second phase; the niche-guided scheme is used to maintain the population diversity in cases where the POF has several disconnected regions that may result in MOEA/D producing many duplicate solutions. In this way, the improved MOEA/D is expected to be capable of handling difficult landscapes and achieving desirable optimization results.

The proposed algorithm has been tested on some complex problems, including existing and newly designed test functions. Experimental results have shown that the improved MOEA/D achieves better performance than several compared MOEA/D variants, and outperforms another two algorithms, i.e., SPEA2+SDE and NSGA-III, on the test problems. We would like to summarize some key findings regarding the proposed MOEA/D-TPN as follows.

1) The niche-guided scheme provides a high level of population diversity for complex test problems (e.g., F2, UF4) during the evolution, especially when the problem being optimized is discontinuous.
2) When solving problems whose POF has a sharp peak or a long tail, MOEA/D with TP shows a significant better performance than that without TP. The improved MOEA/D has been tested on the convex version of DTLZ2, mF4, and F5, proving that it is very efficient for finding points on the boundary of the POF. Besides, TP can adjust search directions and diversify individuals (e.g., F2), which may be helpful disconnected problems.
3) The experimental results have also revealed that, although a good distribution of weight vectors (i.e., WS-transformation) for MOEA/D is available, it cannot necessarily guarantee the evenness of the resulting solutions on complex POF shapes. This may motivate researchers to seek other possible improvements like decomposition approaches, to enhance MOEA/D instead of pursuing only a better distribution of weight vectors.

Despite that, MOEA/D-TPN has achieved encouraging performance on the tested problems, there still exists areas for continued improvement. A number of future directions have been identified from this paper, and MOEA/D-TPN will be extended to handling more complicated problems and meeting a series of requirements.

References

[1] W. A. Albukhanajer, J. A. Briffa, and Y. Jin, "Evolutionary multiobjective image feature extraction in the presence of noise," *IEEE Trans. Cybern.*, [Online]. Available: http://dx.doi.org/10.1109/TCYB.2014.2360074

[2] C. A. C. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA, USA: Kluwer, 2002.

[3] N. Chen *et al.*, "An evolutionary algorithm with double-level archives for multiobjective optimization," *IEEE Trans. Cybern.*, [Online]. Available: http://dx.doi.org/10.1109/TCYB.2014.2360923

[4] T. Chiang and Y. Lai, "MOEA/D-AWS: Improving MOEA/D by an adaptive mating selection mechanism," in *Proc. IEEE Congr. Evol. Comput.*, New Orleans, LA, USA, 2011, pp. 1473–1480.

[5] C. Dai and Y. Wang, "A new multiobjective evolutionary algorithm based on decomposition of the objective space for multiobjective optimization," *J. Appl. Math.*, vol. 2014, 2014, Art. ID 906147. [Online]. Available: http://dx.doi.org/10.1155/2014/906147

[6] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[7] K. Deb, S. Agrawwal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[8] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Sep. 2013.

[9] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scable test problems for evolutionary multi-objective optimization," Kanpur Genet. Algorithms Lab. (KanGAL), Indian Inst. Technol., Kanpur, Kanpur, India, Tech. Rep. 2001001, 2001.

[10] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Inf. Sci.*, vol. 282, pp. 363–387, Oct. 2014.

[11] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. 2nd Int. Conf. Genet. Algorithms*, Hillsdale, NJ, USA, 1987, pp. 41–49.

[12] F. Gu and H. Liu, "A novel weight design in multi-objective evolutionary algorithm," in *Proc. Int. Conf. Comput. Intell. Secur.*, Nanning, China, 2010, pp. 137–141.

[13] B. J. Hancock and C. B. Mattson, "The smart normal constraint method for directly generating a smart Pareto set," *Struct. Multidiscipl. Optim.*, vol. 48, no. 4, pp. 763–775, 2013.

[14] J. M. Herrero, G. Reynoso-Meza, M. Martnez, X. Blasco, and J. Sanchis, "A smart-distributed Pareto front using ev-MOGA evolutionary algorithm," *Int. J. Artif. Intell. Tools*, vol. 23, no. 2, 2014, Art. ID 1450002.

[15] E. J. Hughes, "Multiple single objective Pareto sampling," in *Proc. IEEE Congr. Evol. Comput.*, vol. 4. Canberra, ACT, Australia, 2003, pp. 2678–2684.

[16] H. Ishibuchi, T. Doi, and Y. Nojima, "Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization algorithms," in *Proc. 9th Int. Conf. Parallel Probl. Solv. Nat.*, Reykjavik, Iceland, 2006, pp. 493–502.

[17] H. Ishibuchi and T. Murata, "Multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 28, no. 3, pp. 392–403, Aug. 1988.

[18] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm," in *Proc. 5th Int. Conf. Evol. Multi-Criterion Optim.*, Nantes, France, 2009, pp. 438–452.

[19] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proc. Genet. Evol. Comput. Conf.*, Portland, OR, USA, 2010, pp. 519–526.

[20] S. Jiang, Z. Cai, J. Zhang, and Y. S. Ong, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *Proc. Int. Conf. Nat. Comput.*, vol. 3. Shanghai, China, 2011, pp. 1260–1264.

[21] S. Jiang, J. Zhang, Y. S. Ong, N. S. Zhang, and P. S. Tan, "A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, [Online]. Available: http://dx.doi.org/10.1109/TCYB.2014.2367526

[22] J. D. Knowles and D. W. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Washington, DC, USA, 1999, pp. 98–105.

[23] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Inter-relationship based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.*, to be published. [Online]. Available: http://dx.doi.org/10.1109/TCYB.2014.2365354

[24] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 348–365, Jun. 2014.

[25] M. Li, S. Yang, and X. Liu, "Evolutionary algorithms with segment-based search for multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1295–1313, Aug. 2014.

[26] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[27] K. Li, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and ant colony," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859, Dec. 2013.

[28] K. Li, Q. Zhang, and R. Battiti, "Hybridization of decomposition and local search for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1808–1820, Oct. 2014.

[29] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching based selection in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 909–923, Dec. 2013.

[30] H. Liu, F. Gu, and Y. Cheung, "T-MOEA/D: MOEA/D with objective transform in multiobjective problems," in *Proc. Int. Conf. Inf. Sci. Manage. Eng.*, Xi'an, China, 2010, pp. 282–285.

[31] H. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.

[32] H. Lu and G. G. Yen, "Rank-density-based multiobjective genetic algorithm and benchmark test function study," *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 325–343, Aug. 2003.

[33] H. Lu, Z. Zhu, X. Wang, and L. Jin, "A variable neighborhood MOEA/D for multiobjective test task scheduling problem," *Math. Probl. Eng.*, vol. 2014, Art. ID 423621, p. 6, 2014 [Online]. Available: http://dx.doi.org/10.1155/2014/423621

[34] X. Ma *et al.*, "MOEA/D with opposition-based learning for multiobjective optimization problem," *Neurocomputing*, vol. 146, pp. 48–64, Dec. 2014.

[35] C. A. Mattson, A. A. Mullur, and A. Messac, "Smart Pareto filter: Obtaining a minimal representation of multiobjective design space," *Eng. Optim.*, vol. 36, no. 6, pp. 721–740, 2004.

[36] Y. Mei, K. Tang, and X. Yao, "Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem," *IEEE Trans. Evol. Comput.*, vol. 15, no. 5, pp. 151–165, Apr. 2011.

[37] K. Nag, T. Pal, and N. R. Pal, "ASMiGA: An archive-based steady-state micro genetic algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 1, pp. 40–52, Jan. 2015.

[38] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization* (Natural Computing). Secaucus, NJ, USA: Springer, 2005.

[39] C. Poloni, "Hybrid GA for multiobjective aerodynamic shape optimization," in *Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Periaux, M. Galan, and P. Cuesta, Eds. New York, NY, USA: Wiley, 1997, pp. 397–414.

[40] T. Qi *et al.*, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.

[41] J. K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 495–511, Aug. 2013.

[42] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications* (Studies in Fuzziness and Soft Computing), vol. 166. New York, NY, USA: Springer, 2005, pp. 313–352.

[43] Y. Tan, Y. Jiao, H. Li, and X. Wang, "MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1648–1660, 2013.

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

[45] Q. Zhang, H. Li, D. Maringer, and E. Tsang, "MOEA/D with NBI-style Tchebycheff approach for portfolio management," in *Proc. IEEE Congr. Evol. Comput.*, Barcelona, Spain, 2010, pp. 1–8.

[46] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," School Comput. Sci. Electr. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. CES-491, 2009.

[47] S. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 442–446, Jun. 2012.

[48] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[49] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

Shouyong Jiang received the B.Sc. degree in information and computation science and the M.Sc. degree in control theory and control engineering both from Northeastern University, Shenyang, China, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree with the School of Computer Science and Informatics, De Montfort University, Leicester, U.K.

His current research interests include evolutionary computation, multiobjective optimization, and dynamic optimization.

Shengxiang Yang (M'00–SM'14) received the B.Sc. and M.Sc. degrees in automatic control and the Ph.D. degree in systems engineering from Northeastern University, Shenyang, China in 1993, 1996, and 1999 respectively.

He is currently a Professor of Computational Intelligence and the Director of the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. His current research interests include evolutionary and genetic algorithms, swarm intelligence, computational intelligence in dynamic and uncertain environments, artificial neural networks for scheduling, and relevant real-world applications. He has over 190 publications.

Prof. Yang is the Chair of the Task Force on Evolutionary Computation in Dynamic and Uncertain Environments, under the Evolutionary Computation Technical Committee of the IEEE Computational Intelligence Society and the Founding Chair of the Task Force on Intelligent Network Systems, under the Intelligent Systems Applications Technical Committee of the IEEE Computational Intelligence Society.