Neurocomputing 146 (2014) 48–64

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

MOEA/D with opposition-based learning for multiobjective optimization problem

Xiaoliang Ma [a,b], Fang Liu [a,b,*], Yutao Qi [a,b], Maoguo Gong [b], Minglei Yin [a,b], Lingling Li [b], Licheng Jiao [b], Jianshe Wu [b]

[a] School of Computer Science and Technology, Xidian University, Xi'an 710071, China
[b] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 23 October 2013
Received in revised form
14 February 2014
Accepted 3 April 2014
Available online 11 July 2014

Keywords:
Multi-objective optimization
Evolutionary algorithm
Decomposition
Opposition-based learning

ABSTRACT

Multiobjective evolutionary algorithm based on decomposition (MOEA/D) has attracted a great deal of attention and has obtained enormous success in the field of evolutionary multiobjective optimization. It converts a multiobjective optimization problem (MOP) into a set of scalar optimization subproblems and then uses the evolutionary algorithm (EA) to optimize these subproblems simultaneously. However, there is a great deal of randomness in MOEA/D. Researchers in the field of evolutionary algorithm, reinforcement learning and neural network have reported that the simultaneous consideration of randomness and opposition has an advantage over pure randomness. A new scheme, called opposition-based learning (OBL), has been proposed in the machine learning field. In this paper, OBL has been integrated into the framework of MOEA/D to accelerate its convergence speed. Hence, our proposed approach is called opposition-based learning MOEA/D (MOEA/D-OBL). Compared with MOEA/D, MOEA/D-OBL uses an opposition-based initial population and opposition-based learning strategy to generate offspring during the evolutionary process. It is compared with its parent algorithm MOEA/D on four representative kinds of MOPs and many-objective optimization problems. Experimental results indicate that MOEA/D-OBL outperforms or performs similar to MOEA/D. Moreover, the parameter sensitivity of generalization opposite point and the probable to use OBL is experimentally investigated.

1. Introduction

Multiobjective optimization problem (MOP) [1] is to optimize a set of functions synchronously. The problem can be stated as follows:

$$\begin{cases} \min \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T \\ \text{subject to } \mathbf{x} \in \Omega \end{cases} \quad (1)$$

where $\Omega \subset \mathbb{R}^n$ is the feasible region of decision variable space, $\mathbf{x} \in \mathbb{R}^n$ is the decision vector, and $\mathbf{F}: \Omega \to \mathbb{R}^m$ is made up of m objective functions that are conflicting with each other.

We can say that $\mathbf{x}_A \in \Omega$ dominates the vector $\mathbf{x}_B \in \Omega$ (denoted as $\mathbf{x}_A \prec \mathbf{x}_B$) if and only if $f_i(\mathbf{x}_A) \le f_i(\mathbf{x}_B)$, $\forall i \in \{1, \ldots, m\}$ and $\mathbf{F}(\mathbf{x}_A) \ne \mathbf{F}(\mathbf{x}_B)$. If there is no vector $\mathbf{x} \in \Omega$ such that $\mathbf{x} \prec \mathbf{x}^*$, we say that $\mathbf{x}^*$ is a Pareto optimal solution. Then the Pareto optimal set is marked as $PS = \{\mathbf{x}^* | \neg \exists \mathbf{x} \in \Omega, \mathbf{x} \prec \mathbf{x}^*\}$. The corresponding mapping of the Pareto-optimal set on the objective function space is called the Pareto optimal front $PF = \{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in PS\}$. In the absence of the decision maker's preference information, the aim of a multiobjective evolutionary algorithm (MOEA) is to find a set of respective and uniform solutions over the PF (in the objective space) or the PS (in the decision space).

Under some conditions, each Pareto optimal solution to a MOP (1) could be an optimal solution of a scalar optimization problem whose objective is an aggregation of all the objective functions $\{f_i(\mathbf{x}) | i = 1, \ldots, m\}$ [2,3]. Therefore, approximation of the whole Pareto-optimal front can be decomposed into a number of scalar objective optimization subproblems. Zhang et al. [3–5] combined the decomposition approaches with MOEA and proposed a simple and generic multiobjective evolutionary algorithm MOEA/D. MOEA/D decomposes a MOP into a set of scalar optimization problems and then uses the EA to optimize these subproblems simultaneously. In recent years, MOEA/D has attracted increasing research interest and many follow-up studies can be categorized into four aspects. The first one is to combine MOEA/D with other nature inspired meta-heuristics [6–9]. The second is to integrate new decomposition strategies into the framework of MOEA/D [10–13]. The third is to generate newly initial weight vectors [14,15] and adaptively adjust the distribution of subproblems in the process of evolution [7,16–18]. The last is to change the offspring reproducing mechanisms in MOEA/D [19–21].

* Corresponding author at: School of Computer Science and Technology, Xidian University, Xi'an 710071, China.
E-mail address: f63liu@163.com (F. Liu).

http://dx.doi.org/10.1016/j.neucom.2014.04.068
0925-2312/

However, there is a great deal of randomness in the evolutionary algorithms. To reduce the randomness, Tizhoosh [22,23] proposed the concept of opposition-based learning (OBL). The main idea behind OBL is to consider an estimate and its corresponding opposite estimate synchronously in order to obtain a better approximation for the optimal solution. Simply random resampling or selection of solutions from the current population has the opportunity of visiting or even revisiting unproductive regions of the search space [24]. A mathematical proof shows that opposite points are more likely to approach the optimal solution than simply random points [25,26]. In recent years, OBL has attracted increasing research interest. According to a review of OBL-based algorithms [27], follow-up studies can be divided into five areas: reinforcement learning algorithms [23,28,29], neural networks [22,30,31], fuzzy logic [32,33], single-objective optimization algorithms [24,34–38] and multiobjective optimization algorithms [39–44].

In this paper, OBL is incorporated into in the framework of MOEA/D. Hence, our proposed approach is called opposition-based learning MOEA/D (MOEA/D-OBL). Compared with its parent algorithm MOEA/D, MOEA/D-OBL has two main differences: population initialization and reproduction. We will enhance these two steps using the OBL scheme, including newly initial population based on opposition learning and new offspring produced by opposition-based learning strategy. The newly initial population in MOEA/D-OBL mainly uses the center sampling-based opposite points [45]. Opposition-based learning strategy combines the evolution operator [3,4] and opposition-based local search. The aim of introducing opposition-based local search is to accelerate the convergence speed of MOEA/D.

The rest of this paper is organized as follows. Section 2 introduces the related background including the adopted decomposition approach and opposition-based learning. Section 3 presents the proposed MOEA/D-OBL algorithm. Section 4 shows the comparison results between the newly developed algorithm and its parent algorithm MOEA/D. Some further studies on the effectiveness of MOEA/D-OBL are also made in this part. Section 5 concludes the paper.

2. Related backgrounds

2.1. Tchebycheff decomposition for MOPs

Quite a few approaches have been developed to decompose a MOP (1) into a number of scalar optimization problems [2,3,10]. Among these decomposition methods, the Tchebycheff decomposition [2] is widely used due to its ability of handling MOPs with non-convex PFs.

$$\min_{\mathbf{x}\in\Omega} g^{tch}(\mathbf{x}|\mathbf{W},\mathbf{z}^*)=\min_{\mathbf{x}\in\Omega}\max_{1\le i\le m}\{w_i*(f_i(\mathbf{x})-z_i^*)\}$$

where $\mathbf{w}$ is a weight vector and $\mathbf{z}^*=(z_1^*,\ldots,z_m^*)$ is the ideal point (that is to say $z_i^*=\min\{f_i(\mathbf{x})|\mathbf{x}\in\Omega\}$).

As the analysis in Qi et al. [18], they prove that if the straight line $w_1*(f_1-z_1)=\cdots=w_m*(f_m-z_m)$ has an intersection with PF, then the intersection is the optimal solution to the subproblem with weight vector $\mathbf{w}$. Fig. 1 (left) shows the geometry relationship between the weight vector $\mathbf{w}$ of subproblem and its optimal solution for Tchebycheff decomposition.

Obviously, the geometry relationship between the weight vector of a subproblem and the direction of its optimal solution is nonlinear for Tchebycheff approach [46,18]. Deb and Jain [13] proposed an improvement, called modified Tchebycheff decomposition approach, to Tchebycheff decomposition approach. The subproblem is defined as follows:

$$\min_{\mathbf{x}\in\Omega} g^{mtch}(\mathbf{x}|\mathbf{W},\mathbf{z}^*)=\min_{\mathbf{x}\in\Omega}\max_{1\le i\le m}\left\{\frac{f_i(\mathbf{x})-z_i^*}{w_i}\right\}$$

For the difficulty of denominator $w_i=0$, we replace it by a small number (for example, 10^{-8}).

Following the analysis in Qi et al. [18], we have the following result for the modified Tchebycheff decomposition. If the straight line $(f_1-z_1^*)/w_1=(f_2-z_2^*)/w_2=\cdots=(f_m-z_m^*)/w_m$ has an intersection with PF, then the intersection is the optimal solution to the subproblem with weight vector $\mathbf{w}$. The direction of the optimal solution is consisted with the weight vector for the modified Tchebycheff decomposition. Because the straight line $(f_1-z_1^*)/w_1=(f_2-z_2^*)/w_2=\cdots=(f_m-z_m^*)/w_m$ is across the ideal point $\mathbf{z}^*$ in the direction vector $\mathbf{w}$. Fig. 1 (right) illustrates the contour lines of a scalar subproblem and the geometric relationship between the weight vector of a subproblem and its optimal solution for modified Tchebycheff decomposition. Therefore, with the uniformly distributed weight vectors, the modified Tchebycheff decomposition approach is expected to obtain more uniform optimal solutions over the PF than the Tchebycheff decomposition approach. The reason is that the relationship between the weight vector of a subproblem and its optimal solution is consistent for the modified Tchebycheff decomposition while nonlinear for the Tchebycheff decomposition.

2.2. Opposition-based learning (OBL)

Generally speaking, MOEAs start with an initial population and try to improve them toward the Pareto-optimal solutions in the process of evolution. Optimization stops when some predefined criteria are met. Without a prior knowledge about the Pareto-optimal solutions, we usually start with a random population.

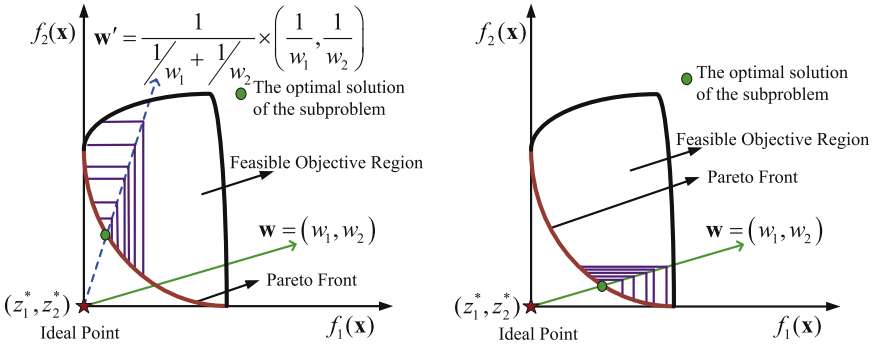


Fig. 1. Plot of the contour lines of a scalar subproblem and the geometric relationship between the weight vector of a subproblem and its optimal solution for Tchebycheff decomposition on the left and modified Tchebycheff decomposition on the right.

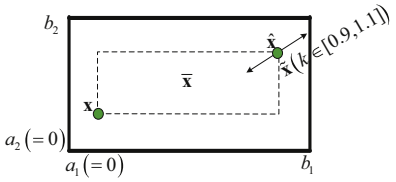


Fig. 2. Opposite points defined in 2-dimension space $\Omega=[a_1,b_1]\times[a_2,b_2]$.

The computation time is related to the distance of the initial population from the Pareto-optimal solutions in the decision space. We can improve our selection of starting with a solution by simultaneously checking its opposite solution. According to the analysis of Rahnamayan et al. [25,26], 50% of the time for a random guess is further from the solution than its opposite guess. Therefore, starting with solution and its opposite solution has the potential to accelerate convergence [24]. The same approach can be used not only to the initial population but also to the current population in the process of evolution. The concept of opposite points is defined as follows.

Definition of Opposite Number [22]. Let $x\in[a,b]$ be a real number. The opposite number $\hat{x}$ is defined as follows:

$$\hat{x}=a+b-x \tag{2}$$

The opposite point in a multidimensional case can be defined based on the opposite number.

Definition of Opposite Point [22]. Let $\mathbf{x}=(x_1,x_2,\ldots,x_n)\in\mathbb{R}^n$ and $x_i\in[a_i,b_i]$, $\forall i=1,2,\ldots,n$. The opposite point is defined as $\hat{\mathbf{x}}=(\hat{x}_1,\hat{x}_2,\ldots,\hat{x}_n)$ where

$$\hat{x}_i=a_i+b_i-x_i,\quad i=1,\ldots,n \tag{3}$$

Fig. 2 takes 2-dimension for example to show the location of opposite point.

Definition of Center Sampling-Based Opposite Point [45]. Let $\mathbf{x}=(x_1,x_2,\ldots,x_n)\in\mathbb{R}^n$ and $x_i\in[a_i,b_i]$, $\forall i=1,2,\ldots,n$. $\hat{\mathbf{x}}=(\hat{x}_1,\hat{x}_2,\ldots,\hat{x}_n)$ is its opposite point. The center sampling-based opposite point $\bar{\mathbf{x}}=(\bar{x}_1,\bar{x}_2,\ldots,\bar{x}_n)$ is defined as a random point between $\mathbf{x}$ and its opposite point $\hat{\mathbf{x}}$:

$$\bar{x}_i=x_i+rand_i\times(\hat{x}_i-x_i)=x_i+rand_i\times(a_i+b_i-2x_i),\ i=1,\ldots,n \tag{4}$$

where $rand_i$ is a uniform random number in [0, 1]. The area in the dotted box of Fig. 2 is an example to illustrate the locations of center sampling-based opposite points.

Definition of Generalization Opposite Point [47]. Let $\mathbf{x}=(x_1,x_2,\ldots,x_n)\in\mathbb{R}^n$ and $x_i\in[a_i,b_i]$, $\forall i=1,2,\ldots,n$. The generalization opposite point is defined as $\tilde{\mathbf{x}}=(\tilde{x}_1,\tilde{x}_2,\ldots,\tilde{x}_n)$, where

$$\tilde{x}_i=k\times(a_i+b_i)-x_i,\quad i=1,\ldots,n \tag{5}$$

where k is a real number. When k is set as 0, the generalization opposite point $\tilde{\mathbf{x}}=-\mathbf{x}$. When $k=1$, the generalization opposite point is same as the opposite point $\tilde{\mathbf{x}}=\hat{\mathbf{x}}$. We can obtain different generalization opposite points (in a line) by altering the parameter k as shown in Fig. 2.

3. The proposed MOEA/D-OBL

As mentioned in the section of introduction, there are two main differences between the proposed MOEA/D-OBL and its parent MOEA/D: population initialization and producing new generations. We will enhance these two steps using the OBL scheme. The original MOEA/D is chosen as a parent algorithm and the proposed opposition-based ideas are integrated into the framework of MOEA/D to accelerate its convergence speed.

3.1. Opposition-based population initialization

The newly initial population includes two kinds of point(s). One is the center sampling-based opposite points. The second is the center point of the search space Ω. Algorithm 1 gives the detail of opposition-based population initialization.

Random number generation, in the absence of a prior knowledge, is the common choice to create an initial population for machine learning [48,49] and MOEAs [50,51,3]. However, simply random resampling of solutions in the search space has the opportunity of visiting or even revisiting unproductive regions [24]. Recently, Researchers [25,26] in the field of evolutionary algorithms, reinforcement learning and neural networks have reported that the simultaneous consideration of randomness and opposition has an advantage over pure randomness.

Algorithm 1. Population initialization based on opposition learning.

Require: • N : the size of evolution population.
• n : the number of variables in MOP (1).
• $\Omega=[a_1,b_1]\times[a_2,b_2]\times\ldots\times[a_n,b_n]$: the search space in MOP (1).

Ensure: an initial evolution population EP.

Step 1 : Generate a random population RP with the size of N.

Step 2 : Calculate the opposite population based on center sampling (CDEd).

$CDEd_j^i=RP_j^i+rand_j\times(a_j+b_j-2\times RP_j^i), i=1,\ldots,N, j=1,\ldots,n.$ Where RP_j^i and $CDEd_j^i$ denote j-th variable of the i-th vector of the random population (RP) and the opposite population based on center sampling (CDEd), respectively. For $j=1,\ldots,n$, $rand_j$ is a uniform random number in [0, 1].

Step 3: Use the center point $(\frac{a_1+b_1}{2},\ldots,\frac{a_n+b_n}{2})$ of search space Ω to replace one member of population CDEd as the initial evolution population EP.

3.1.1. Opposition versus randomness for initialization population

Compared with a randomly initial population, our population initialization uses a center sampling-based opposite population. Therefore, the reason of using the opposite solutions to substitute for random solutions will be analyzed. In order to evaluate the performance of different initial populations, we give two assumptions about the performance measurement of (initial) population.

Assumption 1. For a black-box problem (in the absence of a prior information about the Pareto-optimal solutions), it is rational to suppose that all solutions in the search space have the same opportunity to be the Pareto optimal solution (this assumption is similar to Rahnamayan and Wang [52]).

Assumption 2. For a MOEA, its computation time for finding a set of respective and uniform solutions over PS is positive correlation to the distance of the initial population from the PS (this assumption is similar to Rahnamayan et al. [24]).

According to the above two assumptions, the Pareto optimal solutions are supposed to obey uniform distribution. For the Pareto-optimal solutions with uniform distribution, a mathematical proof had been proposed to show that opposite point is more likely to be closer to the optimal solution than purely random one in general [25,26]. Researchers had also experimentally shown

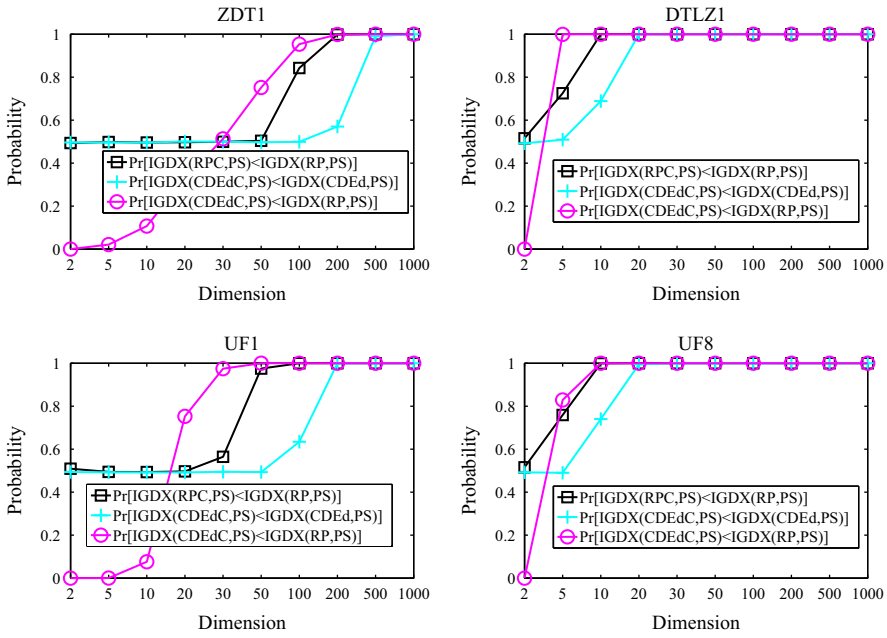


Fig. 3. Plot of the superior probabilities on different dimensions by introducing center point to initialize population. 'RP' and 'RPC', respectively, represent a random population and a random population with center point to replace its one member. Pr[IGDX(RPC, PS) < IGDX(RP, PS)] represents that the probability of population 'RPC' is closer to the PS than random population (RP). Other legends can be understood similarly.

that solutions generated in the center-based region have an advantage over the randomly initial population [52,45].

3.1.2. Center point versus random point for population initialization

Compared with a randomly initial population, our population initialization uses the center point of search space. Therefore, the reason for using the center point to replace random point will be discussed in two aspects. One is based on the experimental and theoretical view for Pareto optimal solutions with uniform distribution. The second is based on the experiments for the widely used MOPs.

According to Assumption 1, without a prior knowledge about the Pareto optimal solutions, we assume that Pareto optimal solutions obey uniform distribution. From the experimental view, the center point has the maximum chance to be closer to the optimal solution (Pareto optimal solutions) and at the same time the minimum average distance from the optimal solution [52]. Rahnamayan and Wang [52] also suggested that the center point is a valuable point. Moreover, from the theoretical view, we prove that the center point has the minimal expected distance (L_1-norm) to the Pareto optimal solutions with uniform distribution.

Theorem 3.1. *Let the Pareto optimal solution (optimal solution) $\mathbf{x} \in \Omega = [a_1, b_1] \times \ldots \times [a_n, b_n]$ be random vector with uniform distribution, then the center point $((a_1+b_1)/2, \ldots, (a_n+b_n)/2)$ of search space Ω has the minimal expected distance (L_1-norm) to $\mathbf{x}$.*

The proof can be found in Appendix A. If we just scatter one point into the search space, the center point of search space rather than a random point is the best selection based on Theorem 3.1. Moreover, by using center point of search space to replace a random point, the initial population in MOEA/D-OBL is expected to have an advantage over that of MOEA/D. Thirdly, to some degree, Theorem 3.1 offers an explanation of why the center point of search space is a valuable point.

One may wonder that under some conditions, the PS of a continuous MOP is a piecewise continuous $(m-1)$-dimensional manifold rather than uniform distribution, where m is the number of the objective functions. From the experimental view, we also study the performance by introducing center point for the widely used MOPs with representative shape of PS, such as ZDT1, DTLZ1, UF1 and UF8. Fig. 3 shows the superior probabilities on different dimensions by introducing center point to initialize population. 'PRC' represents a random population with center point to replace its one member. 'CDEdC', which is constructed by Algorithm 1, indicates the opposition population based on center sampling (CDEd) with center point to replace its one member. The implement detail for Fig. 3 can be found in Appendix B. As shown in Fig. 3, introducing center point can improve the performance of initialization population for the four selected MOPs. Especially for large scale multiobjective problems (dimension > 100), the superior probability by introducing the center point to initialize population is close to 1.

3.2. Opposition-based learning strategy

Opposition-based learning strategy combines the genetic operator used in MOEA/D [3,4] with opposition-based local search operator. Unlike opposition-based initialization, opposition-based local search operator calculates the generalization opposite points dynamically. Instead of using variables' predefined interval boundaries ($[a_i, b_i], i = 1, \ldots, n$), it uses the minimum ($\mathbf{Min}^i$) and maximum ($\mathbf{Max}^i$) values of the neighbor solutions to i-th subproblem as the boundary. Due to the new offspring generated by this opposite-based learning locating in the hyper-rectangle formed by $\mathbf{Min}^i$ and $\mathbf{Max}^i$, we call this strategy as opposition-based local search operator. Algorithm 2 gives the detail of reproduction based on learning strategy.

Fig. 4 shows the distribution of probable offspring for the opposition-based learning strategy. We select two representative distributions of solutions in neighbor subproblems, including approximate linear and quadratic distributions, to show the effect of the proposed opposition-based learning strategy.

Taking Fig. 4 for example, the red circle represents the neighbor solutions $\mathbf{x}^l, l \in B\{i\}$. $B\{i\}$ is the indexes of neighbor subproblems to i-th subproblem and $\{\mathbf{x}^l | l \in B\{i\}\}$ are the neighbor solutions of i-th subproblem in the decision space. The blue-green dot dash line is a hyper-rectangle. The hyper-rectangle is formed by the minimum

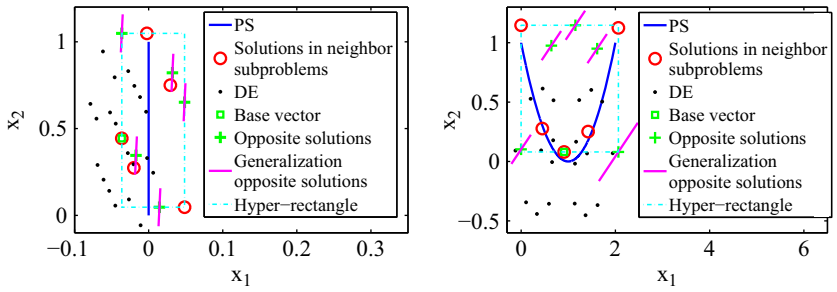


Fig. 4. Plot of the distribution of offspring for DE operator and OBL operator. The OBL operator is based on generalization opposite points. Where the parameter F is set as 0.5 and CR is set as 1 for DE operator. The parameter k is a random number in [0.8, 1.2] for constructing generalization opposite solutions. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

($\mathbf{Min}^i$) and maximum ($\mathbf{Max}^i$) values of all the solutions of neighbor subproblems. That is to say, $\mathbf{Min}_j^i = \min_{l\in B\{i\}}\{x_j^l\}$ and $\mathbf{Max}_j^i = \max_{l\in B\{i\}}\{x_j^l\}$. The green '+' indicates the opposite point ($\mathbf{Min}^i + \mathbf{Max}^i - \mathbf{x}^l$) of the neighbor solution $\mathbf{x}^l, l\in B\{i\}$. The red segment line presents the generalization opposite solutions ($k\times(\mathbf{Min}^i+\mathbf{Max}^i)-\mathbf{x}^l$) of the neighbor solution $\mathbf{x}^l, l\in B\{i\}$ with random number k varying in [0.8, 1.2]. The definition of generalization opposite point can be found in Section 2.2.

For the validity of OBL method which generates generalization opposite points, Fig. 4 gives a visual show. From Fig. 4, we can see that the offspring generated by differential evolution (DE) operator [4,5] operator mainly distributes in an ellipse (in 2-dimension decision space) with the solution of i-th subproblem as its center. On the other hand, generalization opposite points are located in a line segment near the opposite point. An important character is that the generating solutions of DE and OBL are mainly located in the hyper-rectangle which just contains the solutions of neighbor subproblems. The difference between DE and OBL is that their generating solutions distribute in the different regions. Therefore, DE operator and OBL operator have a strongly complementary. That is the reason why we combine DE with OBL to reproduce offspring.

Algorithm 2. Reproduction based on learning strategy.

Require: • i : the index of subproblem.
• $B(i)$: the indexes of neighbor subproblems to i-th subproblem.
• $\{\mathbf{x}^l | l\in B(i)\}$: the neighbor solutions in the decision space to i-th subproblem.
• $Pobl$: The probable to use OBL local search operator.
Ensure: an new offspring $\hat{\mathbf{y}}$.
If $rand < Pobl$, **then** generate new offspring using opposition-based learning local search by
$\hat{\mathbf{y}}_j = k\times(\mathbf{Min}_j^i+\mathbf{Max}_j^i)-\mathbf{x}_j^l, j=1,\ldots,n, l\in B(i)$
• where k is a uniform random number in [0.8, 1.2], $\mathbf{Min}_j^i = min_{l\in B(i)}\{\mathbf{x}_j^l\}$, $\mathbf{Max}_j^i = max_{l\in B(i)}\{\mathbf{x}_j^l\}$ and $\mathbf{x}^l$ is a randomly neighbor solution of i-th subproblem.
Else use genetic operators to construct offspring
• Set $r_1 = i$ and randomly select two indexes r_2 and r_3 from $B(i)$, construct a solution $\overline{\mathbf{y}}$ based on $\mathbf{x}^{r_1}$, $\mathbf{x}^{r_2}$ and $\mathbf{x}^{r_3}$ by using the genetic operator, and apply the polynomial mutation operator [53] on $\overline{\mathbf{y}}$ to generate a new solution $\hat{\mathbf{y}}$.

3.3. The framework of MOEA/D-OBL

MOEA/D-OBL has the same framework as MOEA/D. It optimizes each subproblem by using the information of its neighboring subproblems. For each subproblem, its neighborhood subproblems consist of a number of the nearest subproblems. The main differences between MOEA/D-OBL and MOEA/D are the population initialization method and the opposition-based learning strategy to generate new offspring. During the search procedure, MOEA/D-OBL maintains the following items:

- a number of subproblems $\{subp^1,\ldots,subp^N\}$: $subp^i = (\mathbf{x}^i, \mathbf{F}^i, \mathbf{w}^i)$, $i=1,\ldots,N$, where $\mathbf{x}^i$, $\mathbf{F}^i$ and $\mathbf{w}^i$ are the current optimal solution, corresponding objective vector $\mathbf{F}^i = \mathbf{F}(\mathbf{x}^i)$ and weight vector to the i-th subproblem, respectively.
- $\mathbf{z} = (z_1,\ldots,z_m)^T$, where z_i is the current best value obtained for the i-th objective f_i.

Algorithm 3describes the proposed MOEA/D-OBL.

Algorithm 3. MOEA/D-OBL.

Require: • N : the number of the subproblems applied in MOEA/D-OBL.
• m : the number of objective functions in MOP (1).
• T : the neighborhood size of each subproblem.
• $\{\mathbf{w}^1,\ldots,\mathbf{w}^N\}$: a set of uniformly distributed weight vectors.
Ensure: $\mathbf{x}^1,\ldots,\mathbf{x}^N$ and $\mathbf{F}^1,\ldots,\mathbf{F}^N$.
Step 1 Initialization
1.1 **Initialization population**: Use the Algorithm 1 to initialize the evolution population $EP = \{\mathbf{x}^1,\ldots,\mathbf{x}^N\}$ and set $\mathbf{F}^i = \mathbf{F}(\mathbf{x}^i)$; Set the reference by $z_i = \min\{f_i(\mathbf{x}^1),\ldots,f_i(\mathbf{x}^N)\}, i=1,\ldots,m$.
1.2 **Initialize the subproblems**: Generate the uniformly distributed weight vectors $\mathbf{w}^1,\ldots,\mathbf{w}^N$ and calculate T closest weight vectors to each weight vector [3,4]. For each $i=1,\ldots,N$, let $B(i) = \{i_1,\ldots,i_T\}$. Where $\mathbf{w}^{i_1},\ldots,\mathbf{w}^{i_T}$ are the T nearest weight vectors to $\mathbf{w}^i$.
Step 2: Evolution
For each $i=1,\ldots,N$, do
2.1 **Use Algorithm 2 to reproduce new offspring $\hat{\mathbf{y}}$ for i-th subproblem**.
2.2 **Repair and evaluate**: Check each weight of $\hat{\mathbf{y}}$, if a weight of $\hat{\mathbf{y}}$ is out of the boundary of Ω, its value will be reset by a random value inside of Ω. Assume $\mathbf{y}$ be the repaired solution, evaluate the new offspring $\mathbf{y}$.
2.3 **Update the reference point**: For each $j=1,\ldots,m$, if $f_j(\mathbf{y}) < z_j$, then set $z_j = f_j(\mathbf{y})$.
2.4 **Update the solutions of neighbor subproblems based on the modified Tchebycheff approach**: For each $l\in B(i)$, if $g^{mtch}(\mathbf{y}|\mathbf{w}^l,\mathbf{z}) \le g^{mtch}(\mathbf{x}^l|\mathbf{w}^l,\mathbf{z})$, then let $\mathbf{x}^l = \mathbf{y}$ and $\mathbf{F}^l = \mathbf{F}(\mathbf{y})$.

Step 3: Stopping Criteria: If the stopping criterion is met, then cease and output $\mathbf{x}^1, \ldots, \mathbf{x}^N$ and $\mathbf{F}^1, \ldots, \mathbf{F}^N$. Otherwise, go to Step 2.

4. Experimental study

In this section, we firstly compare MOEA/D-OBL with MOEA/D [3,4] on four representative kinds of test problems including bi-objective ZDT1–ZDT4, ZDT6 problems [54], tri-objective DTLZ1–DTLZ4 and DTLZ7 problems [50], UF1–UF10 problems [55], tri-objective WFG1–WFG9 problems [56]. Secondly, we observe the ability of MOEA/D-OBL on many-objective optimization problems. Finally, the sensibility analysis of parameter k and $Pobl$ in constructing generalization opposite point is also studied. The simulating codes of the compared methods run on a workstation with Inter Core6 2.8 GHz CPU and 32 GB RAM.

4.1. Experimental setting

In our experimental study, the code of MOEA/D can be found at http://cswww.essex.ac.uk/staff/zhang/. The simulated binary crossover (SBX) operator and the polynomial mutation [53] are applied for ZDT problems, DTLZ problems and WFG problems. And the differential evolution (DE) and polynomial mutation [5] are used for UF problems.

The parameter k is set as a random number in $[0.8, 1.2]$. The parameter $Pobl$ is set as 0.1. The dimension of decision variable is set as 500 for ZDT1–ZDT4 and ZDT6, 100 for DTLZ1–DTLZ4 and DTLZ7, 30 for UF1–UF10 and 24 for WFG4–WFG9. The population size is set to 100 for the five bi-objective ZDT problems, 120 for the tri-objective DTLZ and WFG problems, 600 for bi-objective UF1–UF7, 1000 for tri-objective UF8–UF10, 220 for 4-objective DTLZ1–DTLZ4 problems and 792 for 6-objective DTLZ1–DTLZ4 problems. The maximum number of function evaluation is set as 400,000 for ZDT problems, 300,000 for tri-objective DTLZ problems, 300,000 for UF and tri-objective WFG problems, and 500,000 for the 4-objective DTLZ1–DTLZ4 problems and 1,000,000 for the 6-objective DTLZ1–DTLZ4 problems. All the compared algorithms stop when the function evaluation costs reach the maximum value.

The modified Tchebycheff approach with objective normalization [3] is used for both MOEA/D and MOEA/D-OBL.

4.2. Metrics

In the experimental part of this work, we use the Inverted Generational Distance (IGD) metric and Hypervolume (HV) metric which are comprehensive indexes of convergence and diversity [57].

Assume $\mathbf{P}^*$ be a set of uniformly distributed solutions over the Pareto-optimal front (PF). Suppose $\mathbf{P}$ be an approximate solutions to the PF, the average distance from $\mathbf{P}^*$ to $\mathbf{P}$ can be defined by

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

where $d(\mathbf{v}, \mathbf{P})$ is the minimum Euclidean distance from $\mathbf{v}$ to the approximate solutions $\mathbf{P}$. Therefore, the smaller the IGD-metric value is, the better the quality of the approximation solutions $\mathbf{P}$ is. For all the compared algorithms, we use the evolutionary population as the $\mathbf{P}$ to calculate the IGD-metric. The number of the solutions in $\mathbf{P}^*$ used to calculate the value of IGD is 1000 for 2-objective problems, 2500 for 3-objective problems, 5985 for 4-objective problems and 11,628 for 6-objective problems.

Hypervolume metric measures the volume enclosed by the approximation points $\mathbf{P}$ with a reference point $\mathbf{r}$. Let $\mathbf{P} = \{\mathbf{p}^1, \ldots, \mathbf{p}^N\}$ be the approximate non-dominated points to the PF and $\mathbf{r}$ be the reference point meeting $\mathbf{p}^i \prec \mathbf{r}, \forall i = 1, \ldots, N$. Hypervolume metric can be defined by

$$HV(\mathbf{P}, \mathbf{r}) = volume\left(\bigcup_{i=1}^{N} HyperRectangle(\mathbf{p}^i, \mathbf{r})\right)$$

where $HyperRectangle(\mathbf{p}^i, \mathbf{r}) = \{\mathbf{v} | p_j^i \le v_j \le r_j, j = 1, \ldots, m\}$ indicates a hyper-rectangle formed by the reference point $\mathbf{r}$ and the i-th non-dominated point $\mathbf{p}^i$.

In this work, the reference point $\mathbf{r}$ is made up of the maximum values of non-dominated solutions found by the compared algorithms. We independently run each compared algorithm 30 times to calculate the statistical values of Hypervolume-metric and IGD-metric.

4.3. Experimental results

In this part, four kinds of test problems are used to test the performance of MOEA/D-OBL. They are bi-objective ZDT problems, tri-objective DTLZ problems, UF problems and tri-objective WFG problems.

4.3.1. Experimental results on MOPs with simple PSs

In this part of experiments, the classical ZDT and the DTLZ problems are bi-objective and tri-objective problems, respectively. The mathematical definition of the ZDT and DTLZ problems can be found in [50].

Table 1
Mean and standard deviation of Hypervolume metric and IGD metric values obtained by MOEA/D-OBL and MOEA/D on ZDT problems with 500 variable and tri-objective DTLZ instances with 100 variables.

Problem	MOEA/D-OBL				MOEA/D			
	HV		IGD		HV		IGD	
	Mean	std	Mean	std	Mean	std	Mean	std
ZDT1(500)	**7.0107e−1**	2.8234e−3	**9.3156e−3**	9.3030e−4	6.8127e−1	2.7954e−3	2.2211e−2	1.5914e−3
ZDT2(500)	**3.3322e−1**	3.5880e−3	**7.2667e−3**	1.3319e−3	3.2304e−1	2.1012e−3	1.3325e−2	7.8743e−4
ZDT3(500)	7.7911e−1	3.0555e−3	8.6395e−3	1.8840e−4	**7.8055e−1**	2.8643e−3	**8.5729e−3**	1.2732e−4
ZDT4(500)	**1.4957e+3**	1.0723e−1	**4.6014e−3**	1.6512e−5	9.4741e+2	9.4700e+1	4.1263e+2	2.5890e+1
ZDT6(500)	**5.7026e−1**	2.1034e−3	**3.4415e−3**	1.1477e−5	2.0674e−1	7.6216e−3	3.7845e−1	9.2481e−3
DTLZ1(3,100)	**3.6692e+8**	1.1964e1	**1.7911e−2**	4.8496e−5	3.5992e8	2.7232e6	1.7693e+2	2.1048e+1
DTLZ2(3,100)	**4.6463e−1**	2.9541e−3	**5.0308e−2**	4.0448e−4	4.6182e−1	2.8293e−3	5.0583e−2	3.2388e−4
DTLZ3(3,100)	**1.4451e+10**	7.3620e+2	**4.8467e−2**	1.5520e−4	1.4422e+10	1.2466e+7	3.3336e+2	4.9538e+1
DTLZ4(3,100)	**4.4019e−1**	3.5270e−3	**4.9407e−2**	4.1619e−4	4.3227e−1	2.9771e−3	5.0592e−2	4.8791e−4
DTLZ7(3,100)	**7.6962e−1**	5.9337e−3	**7.7965e−2**	8.1441e−4	7.6761e−1	5.6256e−3	7.8052e−2	8.2932e−4

Table 1 shows the mean and standard deviation of the Hypervolume-metric and IGD-metric values by each algorithm for solving ZDT problems with 500 variables and tri-objective DTLZ problems with 100 variables. It indicates that, as far as Hypervolume-metric and IGD-metric are considered, MOEA/D-OBL performs better than MOEA/D on nine out of 10 problems, except for the ZDT3 problem. Especially for difficult ZDT4, DTLZ1 and DTLZ3 problems, MOEA/D-OBL does much better than its parent algorithm MOEA/D.

Fig. 5 shows the evolution of the average IGD metric values of the found solutions in the current populations among 30 independent runs for ZDT and tri-objective DTLZ problems. This figure presents that MOEA/D-OBL converges much faster and maintains better uniformity than MOEA/D on eight out of 10 problems, except for ZDT3 and DTLZ7 problems. In the early evolutionary, MOEA/D-OBL has an obvious advantage on ZDT4, DTLZ1–DTLZ4 problems in generating the initial population. The reason is that the center point, which is generated in the proposed opposition-based population initialization, of search space is just the Pareto-optimal solution for these five problems. For ZDT1, ZDT2 and ZDT6 problems, the suggested opposition-based learning strategy plays an important role in accelerating the convergence rate of MOEA/D-OBL.

Fig. 6 illustrates in the objective space, the distribution of the final found solutions with the median IGD values obtained by MOEA/D-OBL and MOEA/D on ZDT problems with 500 variables and tri-objective DTLZ problems with 100 variables. It is visually evident that, in terms of the convergence of final solutions, MOEA/D-OBL is much better than its parent algorithm MOEA/D for ZDT1, ZDT2, ZDT4, ZDT6, DTLZ1 and DTLZ3 problems. Especially for difficult ZDT4 with 500 variables and tri-objective DTLZ1 and DTLZ3 problems with 100 variables, MOEA/D-OBL can work well under the given conditions, while MOEA/D failed to approximate the whole Pareto-optimal front. As shown in Fig. 6, the solutions found by MOEA/D are far away the true PF on difficult ZDT4, DTLZ1 and DTLZ3 problems.

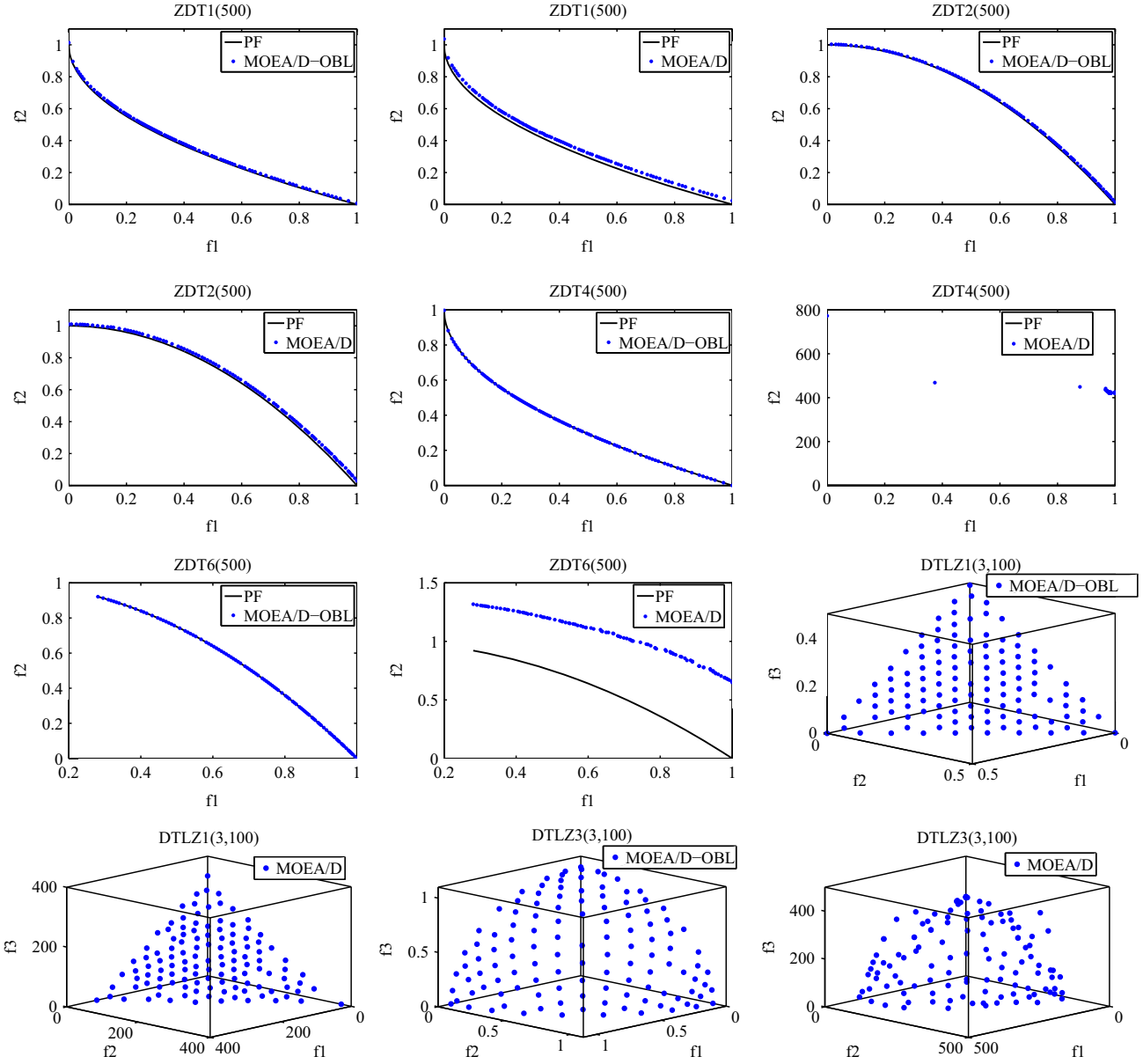


Fig. 5. The distribution of the final obtained solutions with the median IGD values found by MOEA/D-OBL and MOEA/D on bi-objective ZDT problems with 500 variables and tri-objective DTLZ problems with 100 variables.

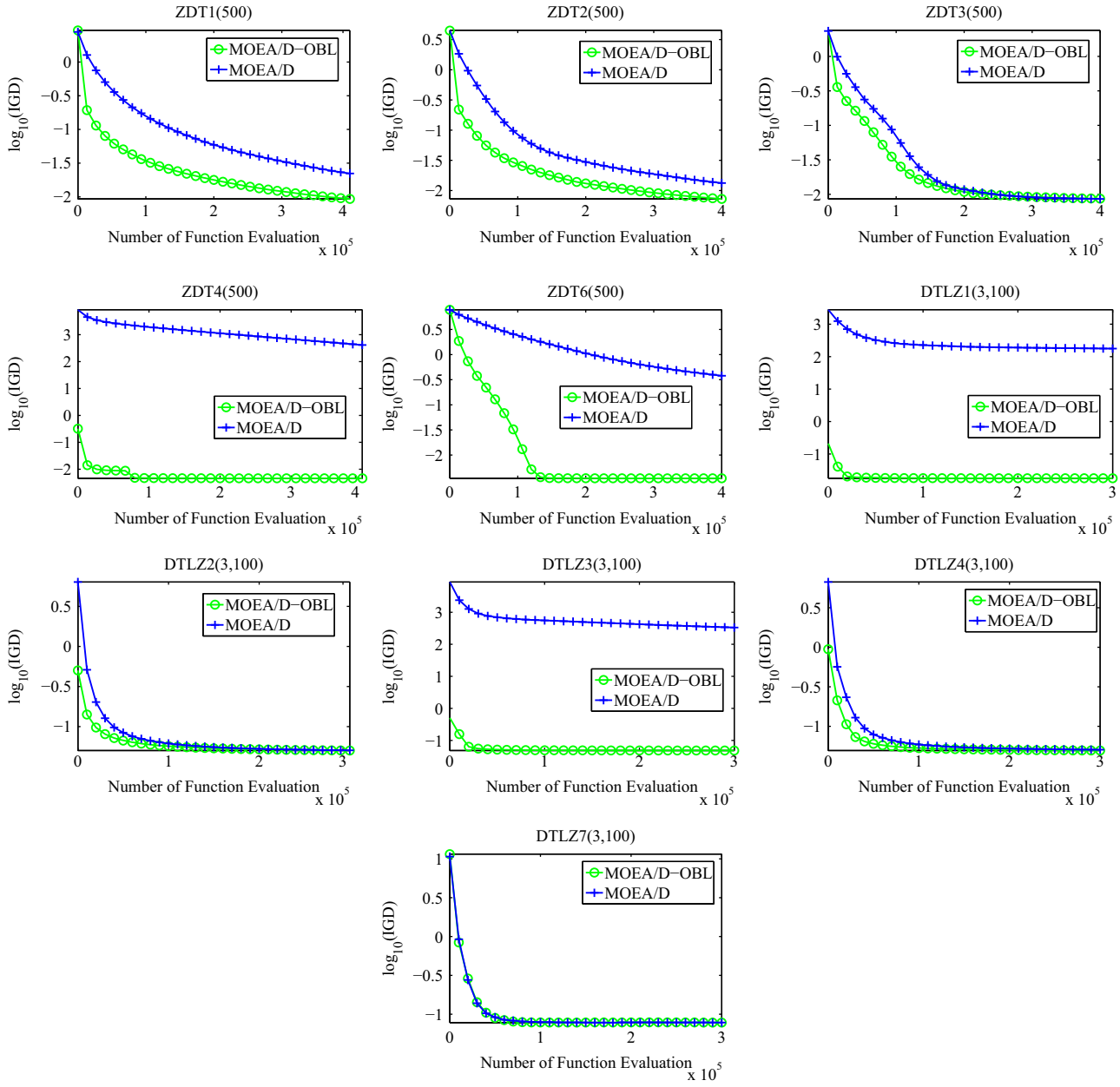


Fig. 6. The evolution of the average IGD metric values of the obtained solutions in the current populations by two algorithms for five bi-objective ZDT problems with 500 variables and five tri-objective DTLZ problems with 100 variables.

4.3.2. Experimental results on MOPs with complicated PSs

In practice, many problems may have a variable linkage on the Pareto optimal solutions. To study the effectiveness of MOEA/D-OBL on this kind of problem, we use the UF problems as the test instances. The mathematical definition of the UF problems can be found in [55].

Table 2 gives the mean and standard deviation of the Hypervolume-metric and IGD-metric values of the final solutions obtained by each algorithm for solving UF problems with 30 variables. This table presents that, in terms of Hypervolume-metric and IGD-metric, MOEA/D-OBL performs better than MOEA/D on UF1, UF3–UF5, UF7–UF8, UF10, while MOEA/D is better than MOEA/D-OBL on UF2 and UF9. Especially for difficult UF4, UF5, UF6, UF8 and UF10 problems, MOEA/D-OBL is the best based on the IGD-metric.

Fig. 7 plots the evolution of the average IGD metric values of the obtained solutions in the current populations on UF problems. From this figure we can find out that MOEA/D-OBL converges much faster and maintains better uniformity than MOEA/D in solving UF1, UF3–UF5 and UF7. For the UF2 problem with complex shape of Pareto-optimal solutions, MOEA/D converges faster and performs better than MOEA/D-OBL. As shown in Fig. 7, for UF5 problem, MOEA/D-OBL has an obvious advantage over MOEA/D in the early evolutionary. The reason is that the center point, which is used in the proposed opposition-based population initialization, of search space is near the Pareto-optimal solution for UF5.

Fig. 8 shows, in the objective space, the distribution of the found solutions with the median IGD values obtained by the compared algorithms on UF problems with 30 variables. This figure illustrates that in terms of the uniformity of final solutions, MOEA/D-OBL is better than its parent algorithm MOEA/D for UF3, UF5, UF7 and UF8 problems. For UF2, MOEA/D performs better than MOEA/D-OBL as far as the uniformity of final solutions is considered.

Table 2
Mean and standard deviation of Hypervolume metric and IGD metric values found by MOEA/D-OBL and MOEA/D on UF instances with 30 variables.

Problem	MOEA/D-OBL				MOEA/D			
	HV		IGD		HV		IGD	
	Mean	std	Mean	std	Mean	std	Mean	std
UF1(30)	**1.0422e+1**	7.7010e−3	**1.4065e−3**	6.9640e−5	1.0392e+1	4.2618e−2	2.8662e−3	1.8155e−3
UF2(30)	2.1908	8.3194e−2	1.3531e−2	1.2407e−2	**2.2214**	6.3499e−2	**8.8753e−3**	6.4820e−3
UF3(30	**6.0942**	6.7012e−3	**3.2117e−3**	2.3242e−3	6.0507	5.1356e−2	6.8461e−3	4.6435e−3
UF4(30)	**6.1902e−1**	6.1758e−3	**5.2742e−2**	3.0695e−3	6.0281e−1	6.3847e−3	6.1775e−2	3.1860e−3
UF5(30)	**1.0407e+1**	5.0586e−1	**1.1800e−1**	4.5233e−2	1.0366e+1	2.5112e−1	1.3686e−1	3.5474e−2
UF6(30)	7.9560	5.7581e−1	**1.4515e−1**	8.0357e−2	**7.9755**	8.1260e−1	1.5084e−1	1.4392e−1
UF7(30)	**5.4671**	9.5841e−3	**1.6406e−3**	4.0604e−4	5.4393	3.7811e−2	4.4690e−3	1.2659e−3
UF8(30)	**1.5631e+1**	1.3138e−2	**2.7440e−2**	3.2188e−3	1.5629e+1	1.9585e−2	2.9591e−2	1.1297e−2
UF9(30)	8.8995	4.7196e−1	9.3068e−2	7.9748e−2	**8.9195**	4.6460e−1	**9.1408e−2**	7.9392e−2
UF10(30)	**1.9459e+3**	5.5284e+1	**4.1235e−1**	4.2752e−2	1.9373e+3	5.4043e+1	4.3223e−1	5.9775e−2

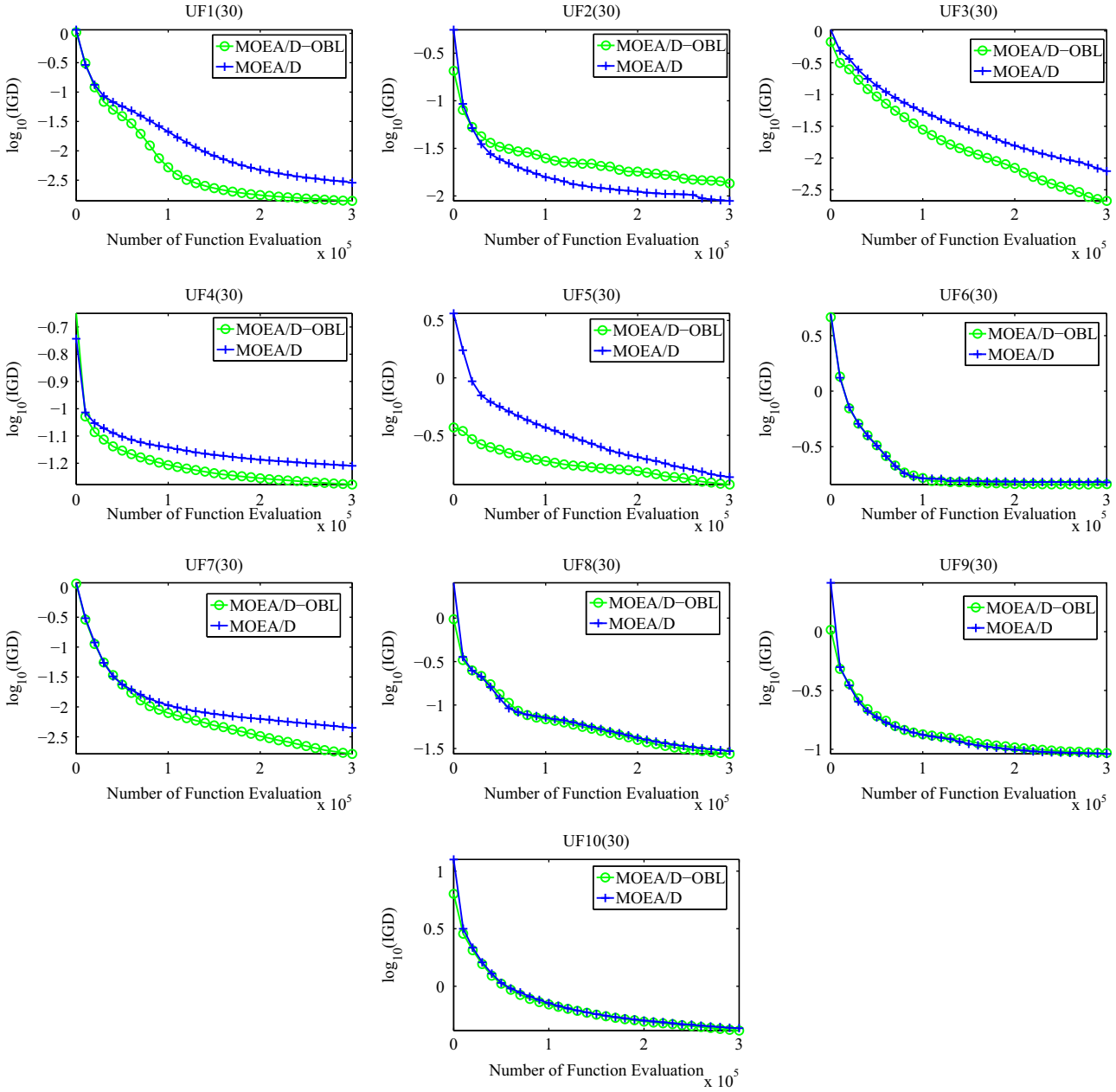


Fig. 7. The evolution of the average IGD metric values of the obtained solutions in the current populations by two algorithms for 10 UF problems with 30 variables.

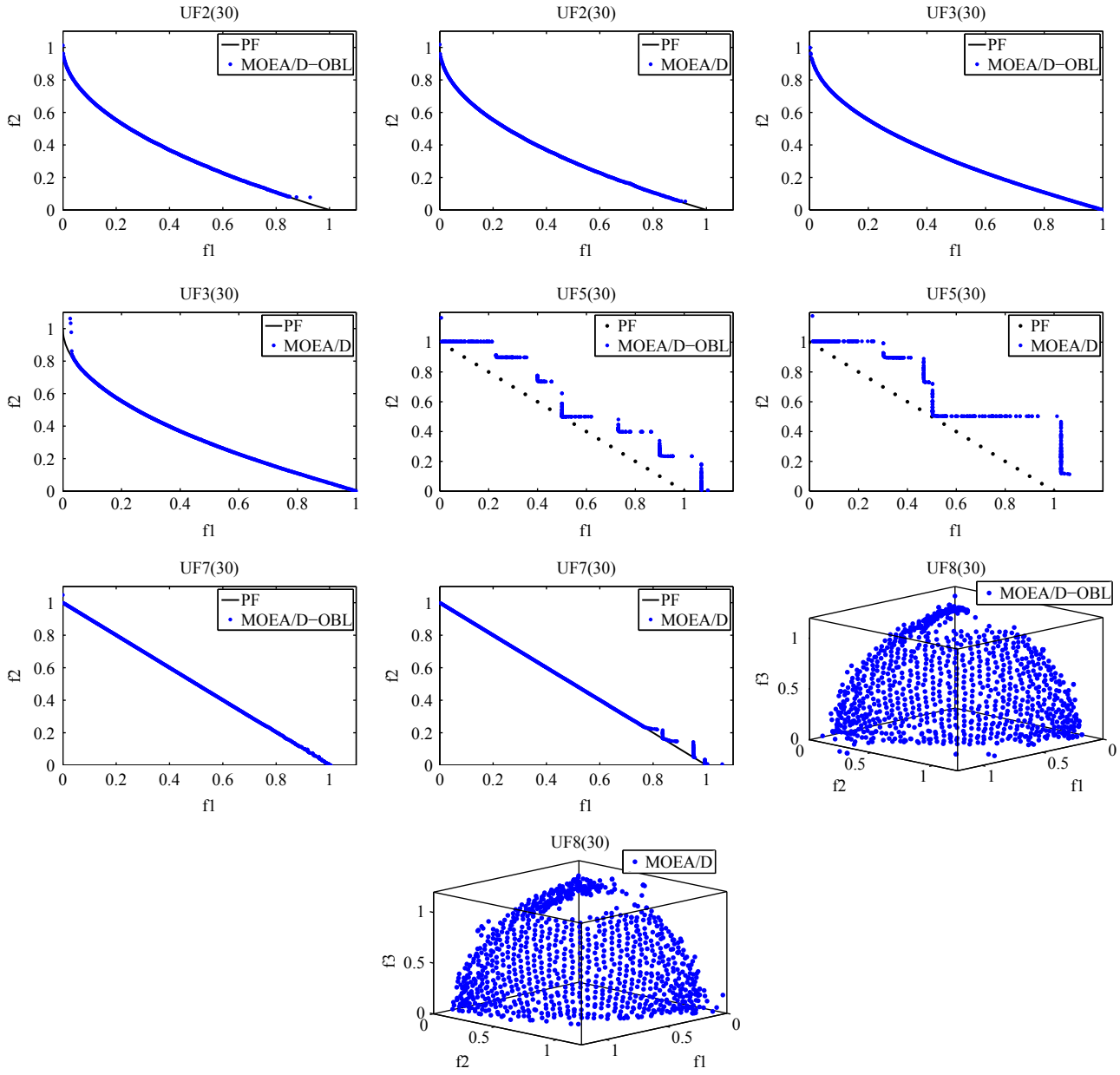


Fig. 8. The distribution of the final obtained solutions with the median IGD values found by MOEA/D-OBL and MOEA/D on UF problems with 30 variables.

4.3.3. Experimental results on WFG problems

To observe the algorithm's performance on problems having differently scaled objective values, we select WFG problems as the test problems. The mathematical definition of WFG problems can be found in [56].

Table 3 presents the mean and standard deviation of the Hypervolume-metric and IGD-metric values of the final solutions obtained by the compared algorithms in solving tri-objective WFG1–WFG9 problems with 24 variables. It shows that in terms of both Hypervolume-metric and IGD-metric, MOEA/D-OBL is better than MOEA/D on WFG2–WFG4 and WFG6–WFG9 problems, while MOEA/D performs better than MOEA/D in solving WFG1 and WFG5 problems.

Fig. 9 illustrates the evolution of the average IGD metric values of the found solutions in the current populations for WFG problems. MOEA/D-OBL converges faster and performs better than MOEA/D on WFG2 and WFG9 problems. For the other problems, the proposed opposition-based population initialization and opposition-based learning strategy have no effect on accelerating the convergence rate of MOEA/D. The reason may be that WFG problems are not difficult problems. As shown in Fig. 9, MOEA/D-OBL and MOEA/D can converge fast in the early evolutionary. There is no enough time for the proposed opposition-based learning strategy to work.

Fig. 10 illustrates, in the objective space, the distribution of the final found solutions with the median IGD values obtained by each algorithm on WFG2 problem with 24 variables. MOEA/D-OBL can obtain more uniform solution than MOEA/D on the parts with dished.

4.4. Further investigations of MOEA/D-OBL

In this part, we want to study the ability of MOEA/D-OBL for solving many-objective problems and the parameter sensibility of k in constructing generalization opposite point.

4.4.1. Experimental results on many-objective problems

In this section, 4-objective and 6-objective DTLZ1–DTLZ4 problems are selected to study the ability of MOEA/D-OBL on many objective problems.

Table 3
Mean and standard deviation of Hypervolume metric and IGD metric values obtained by MOEA/D-OBL and MOEA/D on tri-objective WFG instances with 24 variables.

Problem	MOEA/D-OBL				MOEA/D			
	HV		IGD		HV		IGD	
	Mean	std	Mean	std	Mean	std	Mean	std
WFG1(3,24)	4.4388e+1	4.6103e−2	1.6622e−1	2.8290e−3	**4.4374e+1**	6.3176e−2	**1.6564e−1**	2.4415e−3
WFG2(3,24)	**4.3896e+1**	4.7261e−2	**2.1607e−1**	3.5864e−3	4.2673e+1	2.7007	2.5045e−1	7.8965e−2
WFG3(3,24)	**1.6449e+1**	6.3177e−2	**4.9485e−2**	2.7493e−3	1.6432e+1	8.3448e−2	5.0537e−2	2.5774e−3
WFG4(3,24)	**2.0362e+1**	1.3577e−1	**1.9660e−1**	1.2926e−3	2.0331e+1	2.1919e−1	1.9675e−1	1.3989e−3
WFG5(3,24)	1.9597e+1	9.1845e−2	2.1133e−1	7.7585e−4	**1.9645e+1**	1.3930e−1	**2.1119e−1**	7.0936e−4
WFG6(3,24)	**2.0842e+1**	2.4563e−1	**1.9885e−1**	1.6945e−3	2.0660e+1	4.0047e−1	2.0073e−1	3.2200e−3
WFG7(3,24)	**2.0003e+1**	9.6238e−2	**1.9452e−1**	4.0748e−4	1.9962e+1	1.3412e−1	1.9465e−1	7.8375e−4
WFG8(3,24)	**2.2653e+1**	1.6722	**2.9951e−1**	2.3996e−2	2.1848e+1	1.7324e−1	3.1125e−1	1.6794e−3
WFG9(3,24)	**2.1783e+1**	1.1501	**2.1256e−1**	1.7454e−2	2.0555e+1	1.6450	2.3194e−1	2.5224e−2

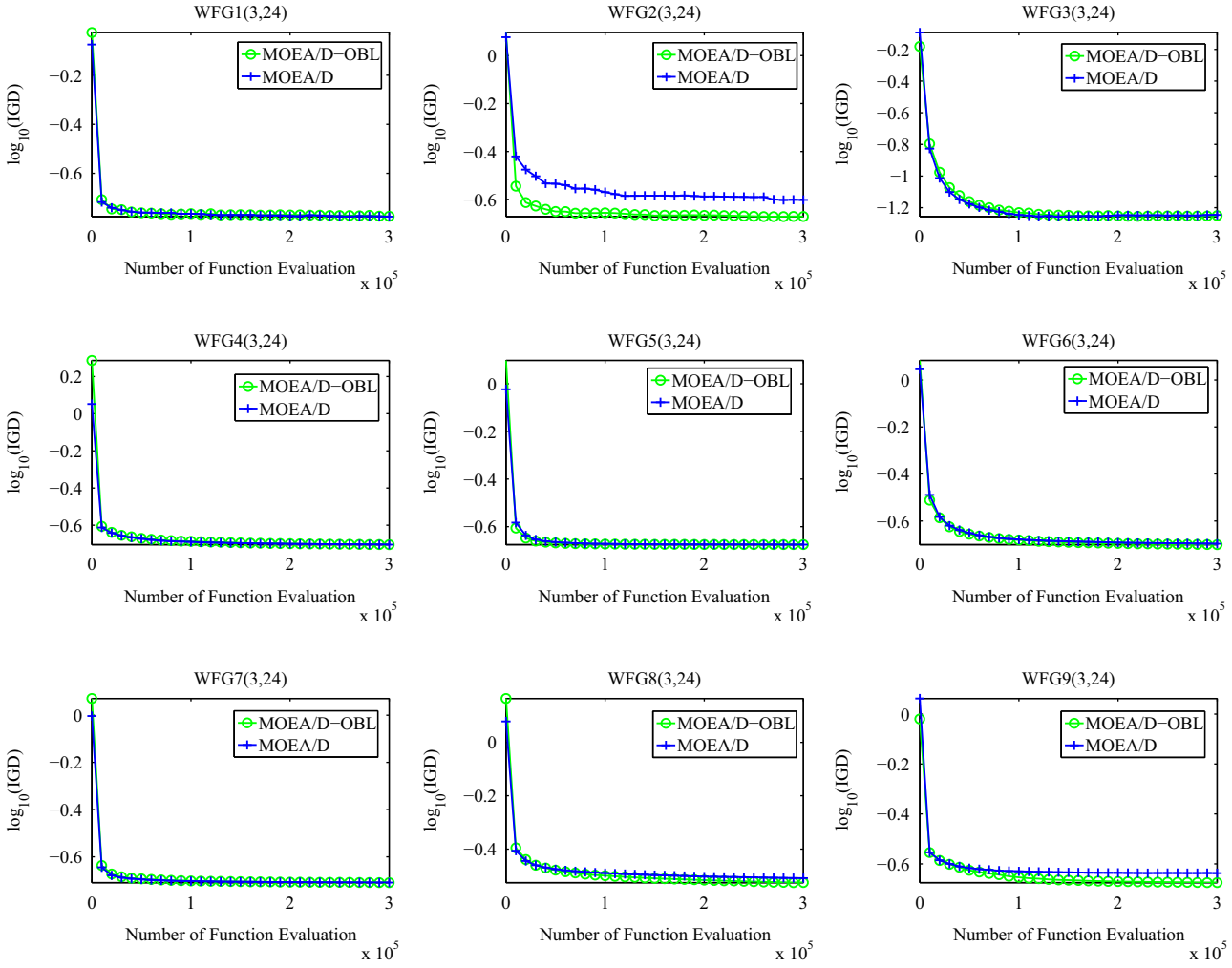


Fig. 9. The evolution of the average IGD metric values of the obtained solutions in the current populations among 30 independent runs with 300,000 function evaluations in two algorithms for nine tri-objective WFG problems with 24 variables.

Table 4 demonstrates the mean and standard deviations of the Hypervolume-metric and IGD-metric for the compared algorithms on 4-objective and 6-objective DTLZ1–DTLZ4 problems with 100 variables. As far as Hypervolume-metric and IGD-metric are considered, the proposed MOEA/D-OBL performs better than MOEA/D on seven out of eight problems, except for 4-objective DTLZ2 problem. It indicates that the proposed MOEA/D-OBL can work for many-objective problems.

Fig. 11 illustrates the evolution of the average IGD metric values of the obtained solutions in the current populations for 4-objective and 6-objective DTLZ1–DTLZ4 problems. On the early evolutionary, MOEA/D-OBL has an obvious advantage over MOEA/D. The success of MOEA/D-OBL can be attributed to the proposed opposition-based population initialization. The center point of search space for DTLZ1–DTLZ4 is just the Pareto-optimal solution.

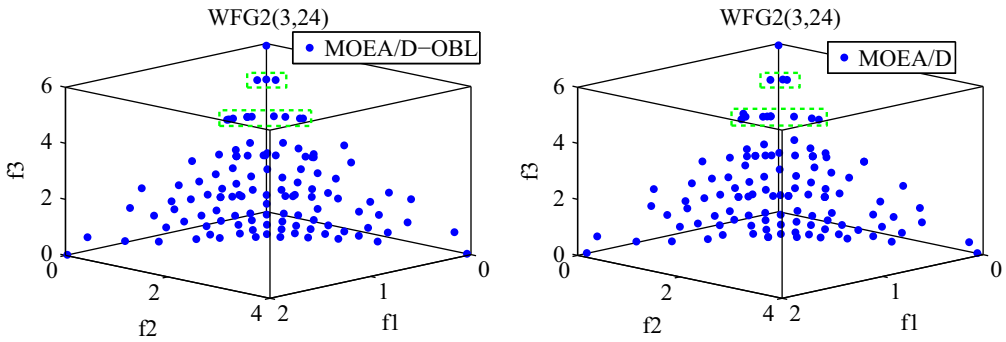


Fig. 10. The distribution of the final obtained solutions with the median IGD values found by MOEA/D-OBL and MOEA/D on tri-objective WFG2 problem with 24 variables.

Table 4
Mean and standard deviation of Hypervolume metric and IGD metric values obtained by MOEA/D-OBL and MOEA/D on 4-objective and 6-objective DTLZ1-DTLZ4 problems with 100 variables.

Problem	MOEA/D-OBL				MOEA/D			
	HV		IGD		HV		IGD	
	Mean	std	Mean	std	Mean	std	Mean	std
DTLZ1(4,100)	**3.5171e+12**	1.3212e+3	**3.2722e−2**	1.5328e−3	3.5167e+12	5.0443e+8	5.3694e+1	5.5470e+1
DTLZ2(4,100)	6.5083e−1	2.9372e−3	9.5169e−2	5.6386e−4	**6.5172e−1**	3.6130e−3	**9.5128e−2**	8.1867e−4
DTLZ3(4,100)	**5.0515e+13**	1.2669e+5	**1.1795e−1**	1.0483e−1	5.0512e+13	3.8666e+9	7.2484e+1	6.4217e+1
DTLZ4(4,100)	**6.3406e−1**	2.2413e−3	**9.4527e−2**	9.1757e−4	6.2375e−1	2.5764e−3	9.7059e−2	2.0453e−3
DTLZ1(6,100)	**4.1348+19**	1.6937e+12	**1.6174e−1**	1.0341e−1	4.1345e+19	1.0931e+14	3.3481e+1	3.4288e+1
DTLZ2(6,100)	**1.4405**	1.1537e−2	**1.9800e−1**	7.1534e−3	1.4354	1.6047e−2	2.0277e−1	1.0214e−2
DTLZ3(6,100)	**4.2112+21**	2.6134e+13	**4.5522e−1**	2.5291e−1	4.2111e+21	1.9211e+16	5.2408e+1	4.4566e+1
DTLZ4(6,100)	**1.1103**	2.6754e−3	**1.8401e−1**	1.7518e−3	1.0912	9.5484e−3	1.9550e−1	6.7997e−3

4.4.2. Sensibility analysis of the parameter k in generating generalization opposite point

Due to the parameter k in generalization opposite point affecting the generated offspring, it is interesting for us to study how the performance of MOEA/D-OBL is affected by k.

To test how the performance of MOEA/D-OBL is affected for various intervals of random number k, 30 independent trials were performed for the intervals [0.5,1.5], [0.6,1.4], [0.7,1.3], [0.8,1.2], [0.9,1.1], [0.95,1.05], [1,1] on the bi-objective UF3–UF5 and tri-objective UF8–UF10 problems. The experimental setting is the same as Section 4.1. In Fig. 12, the mean performance of the proposed algorithm for different intervals of the parameter k is illustrated, where the parameter 0.5 on the abscissa means the random interval [0.5, 1.5] is used. It is evident that different intervals of random number k have various performance of the proposed algorithm. It is difficult to select a fix value. For difficult UF3, UF4, UF5, UF10 problems, the big random interval is preferred, while it may be not established for UF8–UF9 problems. In the experimental study, the parameter k is set as a random number in the interval [0.8, 1.2].

4.4.3. Sensibility analysis of the parameter Pobl in reproduction

Due to the parameter *Pobl* in reproduction affecting the generated offspring, it is interesting to study how the performance of MOEA/D-OBL is affected by the value of *Pobl*. To evaluate the effect of parameter *Pobl* on the performance of MOEA/D-OBL, 30 independent trials were performed on 20 MOPs. These test MOPs include five ZDT problems with 500 variables, five tri-objective DTLZ problems with 100 variables and 10 UF problems with 30 variables. The experimental setting is the same as Section 4.1.

Fig. 13 illustrates the average IGD metric of the proposed algorithm with parameter *Pobl* from 0.1 to 1 on 20 MOPs. In general, the performance of MOEA/D-OBL is dropped with an increase of the values of *Pobl* and different types of MOPs have different features. For the difficult ZDT4, DTLZ1 and DTLZ3 problems with PSs locating in center region of search space, MOEA/D-OBL with various values of *Pobl* has obvious advantage over MOEA/D. The reason is that the center point, which is generated in the initialization population of MOEA/D-OBL, of search space is just the Pareto-optimal solution for the three problems. For ZDT1, ZDT2 and ZDT6 problems whose PSs are located in the boundary of search space, the performance of MOEA/D-OBL has no obvious drop with an increase of the values of *Pobl* except for 1. The reason is that MOEA/D-OBL uses the opposition-based learning local search which has more exploration ability than DE. Therefore, with suitable boundary repair operator, MOEA/D-OBL can have nice performance. For the simple ZDT3, DTLZ2, DTLZ4 and DTLZ7 problems, the performance of MOEA/D-OBL has no obvious drop with an increase of *Pobl* for $Pobl \leq 0.5$. Moreover, with an increase of the values of *Pobl*, the performance of MOEA/D-OBL on most UF problems has more obvious drop than that on ZDT and DTLZ problems.

As clearly shown in Fig. 13, MOEA/D-OBL performs well with *Pobl* from 0.1 to 0.4 on most used MOPs. Thus, we can claim that MOEA/D-OBL is not very sensitive to small values of *Pobl*.

5. Conclusions

In this paper, we have integrated opposition-based learning into the framework of MOEA/D. The proposed MOEA/D-OBL is an improved MOEA/D with two major modifications. One is an opposition-based initial population. The other is an opposition-based learning strategy. We compare MOEA/D-OBL with its parent algorithm MOEA/D. Experimental studies have been preformed on

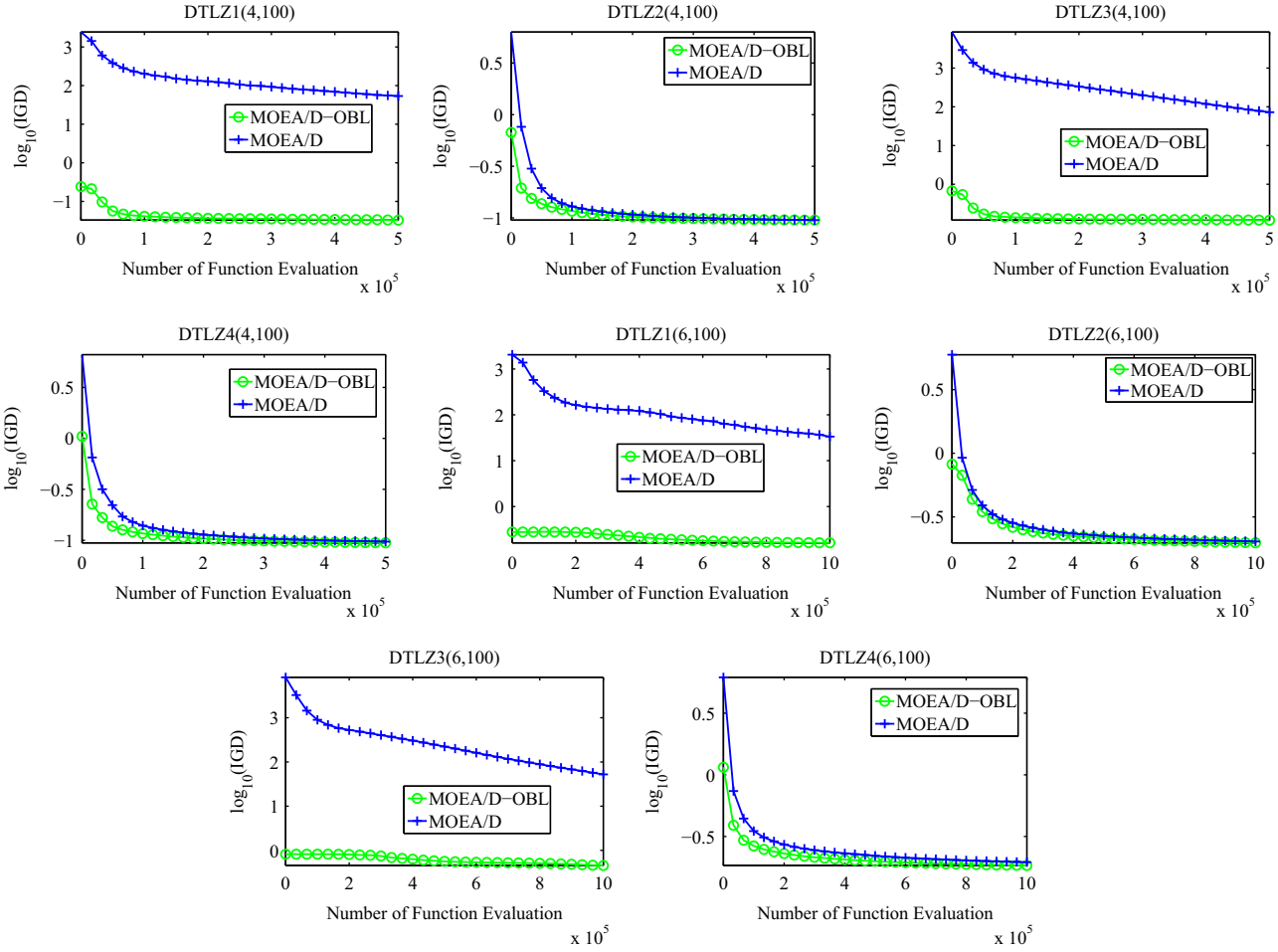


Fig. 11. The evolution of the average IGD metric values of the obtained solutions in the current populations by two algorithms for 4-objective and 6-objective DTLZ1-DTLZ4 problems with 100 variables.

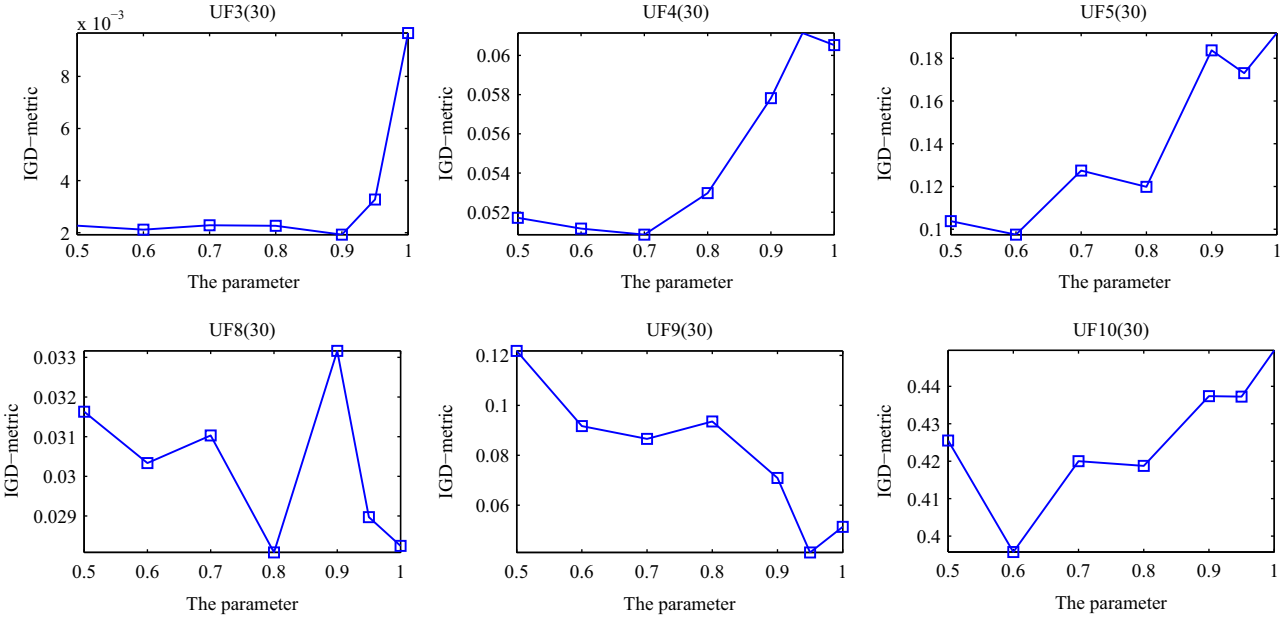


Fig. 12. Mean IGD-metric performance of MOEA/D-OBL over the sampled random intervals for bi-objective UF3-UF5 and tri-objective UF8-UF10 problems.

four representative kinds of test problems including ZDT problems, DTLZ problems, UF problems and WFG problems. The proposed MOEA/D-OBL has also been used for eight many-objective optimization problems.

Our experimental results show that in terms of solution quality, MOEA/D-OBL has outperformed or performed similar to MOEA/D on most test instances, especially for difficult problems. We have also demonstrated that for many-objective 4-objective and

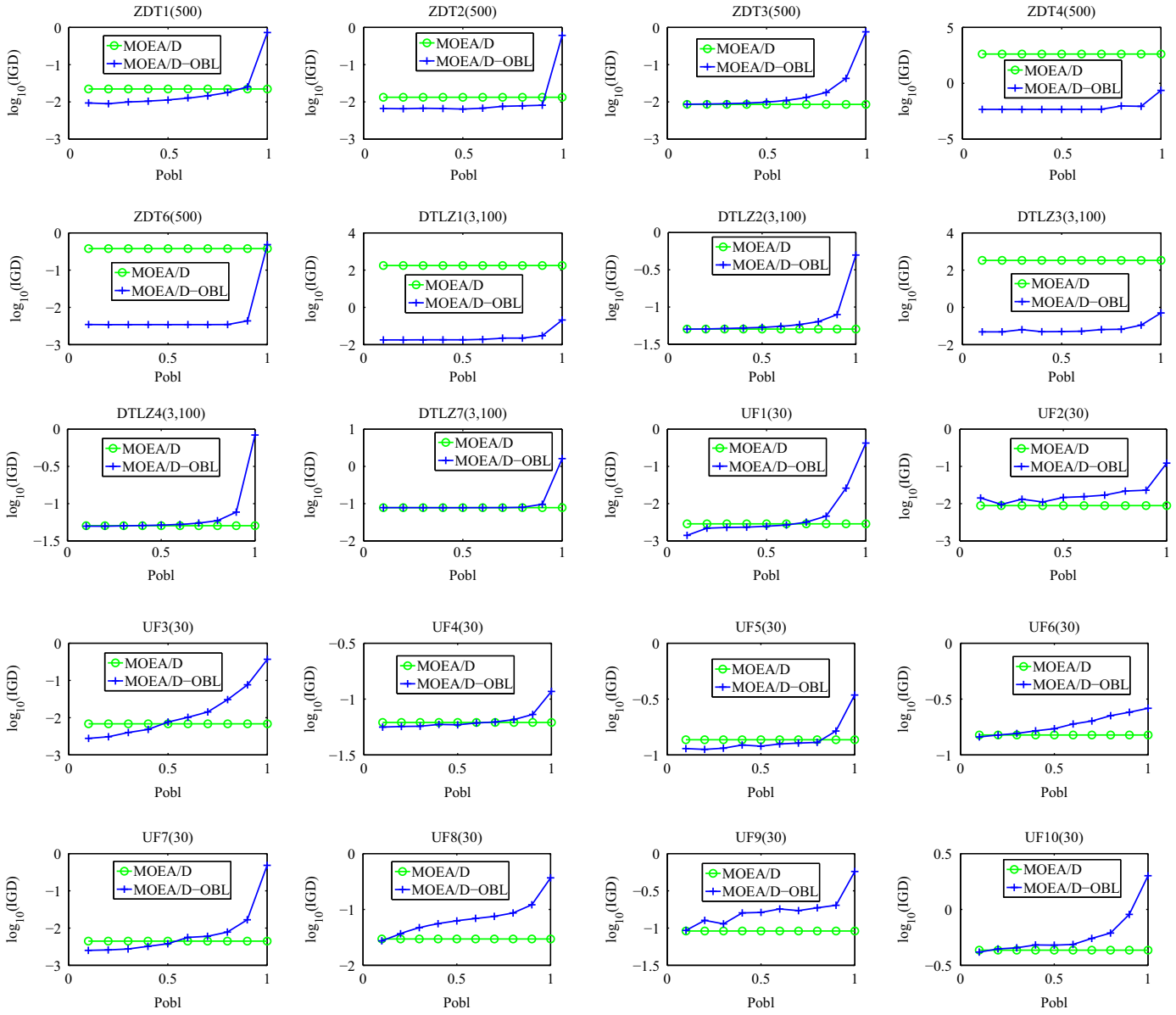


Fig. 13. Plot of mean IGD-metric performance of MOEA/D-OBL with *Pobl* from 0.1 to 1 on five ZDT problems with 500 variables, five tri-objective DTLZ problems with 100 variables and 10 UF problems with 30 variables.

6-objective DTLZ1–DTLZ6 problems, MOEA/D-OBL also outperforms or performed similar to MOEA/D in terms of solution quality. The sensitivity of the parameter k in MOEA/D-OBL has been experimentally investigated. Different random intervals of parameter k have various performance of the proposed algorithm. The sensitivity of the parameter *Pobl* in MOEA/D-OBL has also been experimentally studied. MOEA/D-OBL is not very sensitive to small values of *Pobl*.

The C++ source code of MOEA/D-OBL is available from the authors at request.

Acknowledgments

This work have supported by the National Basic Research Program (973 Program) of China No. 2013CB329402, the National Natural Science Foundation of China under Grant nos. 61173090, 61273317, 61271301, 61272279, 61001202, 61072106, 61072139, 61203303 and 61003199, the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) No. B07048, the National Research Foundation for the Doctoral Program of Higher Education of China Nos. 20100203120008, 20110203110006, the Program for Cheung Kong Scholars and Innovative Research Team in University No. IRT1170, the Fundamental Research Funds for the Central Universities under Grant nos. K5051203007, K5051203002.

Appendix A. The Proof of Theorem 3.1

Proof. Let $\mathbf{y} \in \Omega = [a_1, b_1] \times \ldots \times [a_n, b_n]$, its expected distance (L_1-norm) to the Pareto-optimal solution $\mathbf{x}$ can be calculated as follows:

$$
\begin{aligned}
E(\|\mathbf{y}-\mathbf{x}\|_1) &= \frac{\int_{\Omega}\|\mathbf{y}-\mathbf{x}\|_1\, d\mathbf{x}}{\prod_{l=1}^{n}(b_l-a_l)} \\
&= \frac{\int_{\Omega}\left(\sum_{i=1}^{m}|y_i-x_i|\right) d\mathbf{x}}{\prod_{l=1}^{n}(b_l-a_l)} = \frac{\sum_{i=1}^{m}\int_{\Omega}(|y_i-x_i|)\, d\mathbf{x}}{\prod_{l=1}^{n}(b_l-a_l)} \\
&= \frac{\sum_{i=1}^{m}\int_{\Omega_i}(dx_1 \ldots dx_{i-1}\, dx_{i+1} \ldots dx_n) \times \int_{[a_i,b_i]}(|y_i-x_i|)\, dx_i}{\prod_{l=1}^{n}(b_l-a_l)}
\end{aligned}
$$

$$= \sum_{i=1}^{m} \left[\frac{\int_{a_i}^{y_i} (y_i - x_i)\, dx_i + \int_{y_i}^{b_i} (x_i - y_i)\, dx_i}{b_i - a_i} \right]$$

$$= \sum_{i=1}^{m} \left[\frac{\left(y_i - \frac{a_i + b_i}{2}\right)^2 + \left(\frac{b_i - a_i}{2}\right)^2}{b_i - a_i} \right]$$

where $\Omega_i = [a_1, b_1] \times \ldots \times [a_{i-1}, b_{i-1}] \times [a_{i+1}, b_{i+1}] \times \ldots \times [a_n, b_n]$.

By $\partial[E(\|\mathbf{y} - \mathbf{x}\|_1)]/\partial \mathbf{y} = [2(y_1 - ((a_1 + b_1)/2))/(b_1 - a_1), \ldots, 2(y_n - ((a_n + b_n)/2))/(b_n - a_n)]^T = \mathbf{0} \Longrightarrow y_i = (a_i + b_i)/2, i = 1, \ldots, n$.

Therefore, the center point of search space Ω has the minimal expected distance (L_1-norm) to $\mathbf{x}$. □

Appendix B. The pseudo-code of center point versus random point for initializing population

Algorithm 4. Calculating the performance of various population initialization methods.

Require: $NSample \leftarrow 100,000$; $TestCase \leftarrow$ '*DTLZ*1'; $N \leftarrow 100$;
$Dim \leftarrow$ [2 5 10 20 30 50 100 200 500 1000];
$Ndim \leftarrow size(Dim, 2)$;
$DistRp \leftarrow zeros(Ndim, 1)$; $DistRpc \leftarrow zeros(Ndim, 1)$;
$DistCded \leftarrow zeros(Ndim, 1)$; $DistCdedc \leftarrow zeros(Ndim, 1)$;
for $j = 1$ to $Ndim$ **do**
- $n \leftarrow Dim(j)$; According to the dimension of variables n, obtain the *lower*, *upper* and Pareto-optimal set (PS) of the given problem *TestCase* in decision space.
- Calculate the center point $cp \leftarrow [\frac{lower_1 + upper_1}{2}, \ldots, \frac{lower_n + upper_n}{2}]^T$.
- $RpcBettRp = 0$; $CdedcBettCded = 0$; $CdedcBettRp = 0$;
- **for** $i = 1$ to $NumSample$ **do**
  - Generate four independent populations: random population $RP_{n \times N}$, newly independent random population $\overline{RP}_{n \times N}$, an opposition population based on center sampling $CDEd_{n \times N}$ and an newly independent opposition population based on center sampling $\overline{CDEd}_{n \times N}$.
  $\overline{RP}(:, 1) \leftarrow cp$; $RPC \leftarrow \overline{RP}$; $\overline{CDEd}(:, 1) \leftarrow cp$; $CDEdC \leftarrow \overline{CDEd}$;
  - Calculate the IGDX distance of *RP*, *RPC*, *CDEd* and *CDEdC* from the Pareto-optimal set (PS).
    * $drp =$ CalIGDX(*RP*,*P* S); $DiRp(j) = DiRp(j) + drp/NSample$;
    * $drpc =$ CalIGDX(*RPC*,*PS*);
    $DiOp(j) = DiRpc(j) + drpc/NSample$;
    * $dcded =$ CalIGDX(*CDEd*,*PS*);
    $DiCded(j) = DiCded(j) + dcded/NSample$;
    * $dcdedc =$ CalIGDX(*CDEdC*,*PS*);
    $DiCdedc(j) = DiCdedc(j) + dcdedc/NSample$;
  - **if** $drpc < drp$ **then** $RpcBettRp \leftarrow RpcBettRp + 1$;
  - **if** $dcdedc < dcded$ **then**
  $CdedcBettCded \leftarrow CdedcBettCded + 1$;
  - **if** $dcdedc < drp$ **then** $CdedcBettRp \leftarrow CdedcBettRp + 1$;
- **end for**
- $PrRpcBetterRp(j) \leftarrow RpcBettRp/NSample$;
- $PrCdedcBetterCded(j) \leftarrow CdedcBettCded/NSample$;
- $PrCdedcBetterRp(j) \leftarrow CdedcBettRp/NSample$;

end for

Appendix C. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.neucom.2014.04.068.

References

[1] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, New York, America, 2001.
[2] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, Norwell, America, 1999.
[3] Q. Zhang, H. Li, MOEA/D: a multi-objective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[4] H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 12 (2) (2009) 284–302.
[5] Q. Zhang, W. Liu, H. Li, The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances, Technical Report, School of CS EE, University of Essex, 2009.
[6] L. Ke, Q. Zhang, R. Battiti, Multiobjective Combinatorial Optimization by Using Decomposition and Ant Colony, Technical Report, University of Essex, 2010.
[7] H. Li, D. Landa-Silva, An adaptive evolutionary multi-objective approach based on simulated annealing, Evol. Comput. 19 (4) (2011) 561–595.
[8] N. Moubayed, A. Petrovski, J. McCall, A novel multi-objective particle swarm optimisation based on decomposition, in: International Conference on Parallel Problem Solving from Nature, 2010, pp. 1–10.
[9] Z. Martinez, C. Coello Coello, A multi-objective particle swarm optimizer based on decomposition, in: Genetic and Evolutionary Computation Conference, 2011.
[10] Q. Zhang, H. Li, D. Maringer, E. Tsang, MOEA/D with NBI-style Tchebycheff approach for portfolio management, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–8.
[11] H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm, in: International Conference on Evolutionary Multi-Criterion Optimization, 2009, pp. 438–452.
[12] H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Simultaneous use of different scalarizing functions in MOEA/D, in: Genetic and Evolutionary Computation Conference, 2010, pp. 519–526.
[13] K. Deb, H. Jain, An Improved NSGA-II Procedure for Many-Objective Optimization, Part I: Solving Problems with Box Constraints, Technical Report 2012009, KanGAL, 2012.
[14] I. Giagkiozis, R. Purshouse, P. Fleming, Generalized Decomposition and Cross Entropy Methods for Many-Objective Optimization, Technical Report 1029, ACSE RESEARCH, 2012.
[15] Y. Tan, Y. Jiao, H. Li, X. Wang, MOEA/D + uniform design: a new version of MOEA/D for optimization problems with many objectives, Comput. Oper. Res. 40 (6) (2013) 1648–1660.
[16] F. Gu, H. Liu, A novel weight design in multi-objective evolutionary algorithm, in: International Conference on Computational Intelligence and Security, 2010, pp. 137–141.
[17] S. Jiang, Z. Cai, J. Zhang, Y.S. Ong, Multiobjective optimization by decomposition with Pareto-adaptive weight vectors, in: International Conference on Natural Computation, 2011, pp. 1260–1264.
[18] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, J. Wu, MOEA/D with adaptive weight adjustment, Evol. Comput. 22 (2) (2014) 231–264.
[19] C. Chen, C. Chen, Q. Zhang, Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization, in: IEEE Congress on Evolutionary Computation, 2009, pp. 209–216.
[20] W. Huang, H. Li, On the differential evolution schemes in MOEA/D, in: International Conference on Natural Computation, 2010, pp. 2788–2792.
[21] K. Sindhya, S. Ruuska, T. Haanpaa, K. Miettinen, A new hybrid mutation operator for multiobjective optimization with differential evolution, Soft Comput. Fusion Found. Methodol. Appl. (2011) 1–15.
[22] H. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, in: International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA 2005 and International Conference on Intelligent Agents, Web Technologies, vol. 1, 2005, pp. 695–701.
[23] H. Tizhoosh, Opposition-based reinforcement learning, J. Adv. Comput. Intell. Intell. Inform. 10 (4) (2006) 578–585.
[24] S. Rahnamayan, H. Tizhoosh, M. Salama, Opposition-based differential evolution, IEEE Trans. Evol. Comput. 12 (1) (2008) 64–79.
[25] S. Rahnamayan, H. Tizhoosh, M. Salama, Opposition versus randomness in soft computing techniques, Appl. Soft Comput. J. 8 (2) (2008) 906–918.
[26] S. Rahnamayan, G. Wang, M. Ventresca, An intuitive distance-based explanation of opposition-based sampling, Appl. Soft Comput. 12 (2012) 2828–2839.
[27] F. Al-Qunaieer, H. Tizhoosh, S. Rahnamayan, Opposition based computing—a survey, in: IEEE Congress on Evolutionary Computation, 2010.
[28] M. Shokri, H. Tizhoosh, M. Kamel, Opposition-based Q(λ) algorithm, in: IEEE International Conference on Neural Networks, 2006, pp. 254–261.
[29] M. Mahootchi, Storage system management using reinforcement learning techniques and nonlinear models (Ph.D. thesis), University of Waterloo, 2009.
[30] M. Ventresca, H. Tizhoosh, Opposite transfer functions and backpropagation through time, in: IEEE Symposium on Foundations of Computational Intelligence, 2007, pp. 570–577.
[31] M. Ventresca, H. Tizhoosh, Improving gradient-based learning algorithms for large scale feedforward networks, in: International Joint Conference on Neural Networks, 2009, pp. 3212–3219.
[32] H. Tizhoosh, Opposite fuzzy sets with applications in image processing, in: International Fuzzy Systems Association, 2009, pp. 36–41.

[33] H. Tizhoosh, F. Sahba, Quasi-global oppositional fuzzy thresholding, in: International Conference on Fuzzy Systems, 2009, pp. 1346–1351.
[34] S. Rahnamayan, G. Wang, Solving large scale optimization problems by opposition-based differential evolution (ODE), WSEAS Trans. Comput. 7 (10) (2008) 1792–1804.
[35] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, M. Ventresca, Enhancing particle swarm optimization using generalized opposition-based learning, Inf. Sci. 181 (2011) 4699–4714.
[36] Q. Xu, L. Wand, B. HE, N. Wang, Modified opposition-based differential evolution for function optimization, J. Comput. Inf. Syst. 7 (5) (2011) 1582–1591.
[37] M. Ahandani, H. Alavi-Rad, Opposition-based learning in the shuffled differential evolution algorithm, Soft Comput. 16 (2012) 1303–1337.
[38] H. Wang, Opposition-based barebones particle swarm for constrained nonlinear optimization problems, Math. Probl. Eng. 2012 (2012), 12 pages, http://dx.doi.org/10.1155/2012/761708 (Article ID 761708).
[39] R. Thangaraj, M. Pant, Differential evolution algorithm for solving multi-objective optimization problems, Recent Adv. Math. 23 (5) (2011) 40–45.
[40] R. Patel, M. Raghuwanshi, L. Malik, Decomposition based multi-objective genetic algorithm (DMOGA) with opposition based learning, in: Conference on Computational Intelligence and Communication Networks, 2012.
[41] S. Leung, X. Zhang, S. Yuen, Multiobjective differential evolution algorithm with opposition-based parameter control, in: IEEE World Congress on Computational Intelligence, 2012, pp. 10–15.
[42] F. Bourennani, S. Rahnamayan, G. Naterer, Multi-objective genetic local search algorithm and its application to flowshop scheduling, IEEE Trans. Syst. Man Cybern. 16 (3) (1996).
[43] W. Huang, S. Oh, Identification of fuzzy inference systems by means of a multiobjective opposition-based space search algorithm, Math. Probl. Eng. 2013 (2013), 13 pages, http://dx.doi.org/10.1155/2013/725017 (Article ID 725017).
[44] M. Ma, Y. He, Y. Xie, X. Zhao, Multi-objective particle swarm optimizer with opposition-based learning, J. Parallel Distrib. Comput. 9 (18) (2013) 7165–7172.
[45] A. Esmailzadeh, S. Rahnamayan, Enhanced differential evolution using center-based sampling, in: IEEE Congress on Evolutionary Computation, 2011, pp. 2640–2648.
[46] H. Liu, Y. Wang, Y. Cheung, A multiobjective evolutionary algorithm using min-max strategy and sphere coordinate transformation, Intell. Autom. Soft Comput. 15 (2009) 361–384.
[47] H. Wang, Z. Wu, S. Rahnamayan, L. Kang, A scalability test for accelerated DE using generalized opposition-based learning, in: Conference on Intelligent Systems Design and Applications, 2009, pp. 1090–1095.
[48] M. Anthony, Neural Network Learning: Theoretical Foundations, Cambridge University Press, Cambridge, England, 1999.
[49] R. Sutton, A. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, Massachusetts, America, 2003.
[50] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[51] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: multiobjective selection based on dominated hypervolume. European journal of operational research, IEEE Trans. Evol. Comput. 181 (2007) 1653–1669.
[52] S. Rahnamayan, G. Wang, Center-based sampling for population-based algorithms, in: IEEE Congress on Evolutionary Computation, 2009, pp. 2640–2648.
[53] K. Deb, H. Beyer, Self-adaptive genetic algorithms with simulated binary crossover, Evol. Comput. 9 (2) (2001) 197–221.
[54] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195.
[55] Q. Zhang, A. Zhou, S. Zhao, P. Suganthan, W. Liu, S. Tiwari, Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition, Technical Report CES-887, 2008.
[56] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Trans. Evol. Comput. 10 (5) (2006) 477–506.
[57] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, V. Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Trans. Evol. Comput. 7 (2) (2003) 117–132.

Xiaoliang Ma was born in Zhejiang, China, in 1984. He received the B.S. degree in computing computer science and technology from Zhejiang Normal University, Jinhua China, in 2006. Now he is currently working towards the Ph.D. degree at the school of computing of Xidian University.

His research interests include evolutionary computation, multi-agent systems.

Fang Liu was born in Beijing, China, in 1963. She received the B.S. degree in computer science and technology from the Xi'an Jiaotong University, in 1984 and the M.S. degree in computer science and technology from the Xidian University, in 1995.

Currently, She is a professor at Xidian University. Her research interests include signal and image processing, nonlinear circuit and systems theory, learning theory and algorithms, optimization problems, wavelet theory, and data mining.

Yutao Qi was born in Henan, China, in 1981. He received the B.S. degree in software engineering from the software school of Xidian University, Xi'an China, in 2003, and the M.S. degree in computer science and technology from the Institute of the Information Processing, Xidian University, in 2006.

Currently, he is an associate professor at Xidian University. His research interests include evolutionary computation, multi-agent systems, artificial immune systems, parallel computing and data mining.

Maoguo Gong received the B.S. degree in electronic engineering (with first class honors) and Ph.D. degree in electronic science and technology from Xidian University, Xi'an, China, in 2003 and 2009, respectively. Since 2006, he has been a Teacher with Xidian University. In 2008 and 2010, he was promoted as an Associate Professor and as a Full Professor, respectively, both with exceptive admission. He is currently a Full Professor with the Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, Xidian University. His research interests include computational intelligence with applications.

He is a member of the IEEE Computational Intelligence Society, an Executive Committee Member of the Natural Computation Society of Chinese Association for Artificial Intelligence, and a Senior Member of the Chinese Computer Federation. He was the recipient of the New Century Excellent Talent in University of the Ministry of Education of China, the Eighth Young Scientist Award of Shaanxi, the New Scientific and Technological Star of Shaanxi Province, and the Science and Technology Award of Shaanxi Province (First Level, 2008 and 2010), etc.

Minglei Yin was born in Shandong, China, in 1990. He received the B.S. degree in computing computer science and technology from Dalian University, Dalian China, in 2013. Now he is currently working towards the M.Sc. degree at the school of computing of Xidian University.

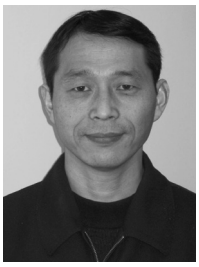
His research interest majors in evolutionary computation.

Lingling Li received the B.S. degree at the School of Electronic Engineering, Xidian University , Xi'an, China, in 2011. Now she is currently working towards the Ph.D. degree at the School of Electronic Engineering of Xidian University.

Her current research interests include community detection in networks, multiobjective optimization.

Licheng Jiao was born in Shaanxi, China, in 1959. He received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, in 1982, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1984 and 1990, respectively. From 1984 to 1986, he was engaged as an Assistant Professor in the Civil Aviation Institute of China, Tianjing, China. During 1990 and 1991, he was a Postdoctoral Fellow at the National Key Lab for Radar Signal Processing, Xidian University, Xi'an. Currently he is the Dean of the Electronic Engineering School and the Director of the Institute of Intelligent Information Processing of Xidian University.

His current research interests include signal and image processing, nonlinear circuit and systems theory, learning theory and algorithms, optimization problems, wavelet theory, and data mining.



Jianshe Wu received his B.Sc. in Physics from the Sichuan University of China, in 2004, he received his M.Sc. in Circuits and Systems and Ph.D. in Pattern Recognition and Intelligent Systems from the Xidian University, in 2001 and 2007, respectively. Then, he continued research in complex networks, nonlinear dynamics, and machine learning at Xidian University.

His current research is related to the application of machine learning algorithms in complex networks and dynamics computing.