International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **5(B)**, May **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 3677–3688

A MULTIOBJECTIVE EVOLUTIONARY ALGORITHM USING DYNAMIC WEIGHT DESIGN METHOD

Fangqing Gu[1], Hai-lin Liu[1,*] and Kay Chen Tan[2]

[1]Faculty of Applied Mathematics
Guangdong University of Technology
No. 100, Waihuan Xi Road, Panyu District, Guangzhou, P. R. China
gufangqing84@sina.com; *Corresponding author: hlliu@gdut.edu.cn

[2]Faculty of Engineering
National University of Singapore
21 Lower Kent Ridge Road, Singapore
eletankc@nus.edu.sg

Received November 2010; revised March 2011

Abstract. *In most multiobjective evolutionary algorithms (MOEA) based on aggregating objectives, the weight vectors are user-supplied or generated randomly, and they are static in the algorithms. If the Pareto front (PF) shape is not complex, the algorithms can find a set of uniformly distributed Pareto optimal solutions along the PF; otherwise, they might fail. A dynamic weight design method based on the projection of the current nondominant solutions and equidistant interpolation is proposed in this paper. Even if the PF is complex, we can find evenly distributed Pareto optimal solutions by this method. Some test instances are constructed to compare the performance of the MOEA/D using dynamic weight design method with that of MOEA/D. The results indicate that the dynamic weight design method can dramatically improve the performance of the algorithms.*

Keywords: Equidistant interpolation, Multiobjective optimization, Evolutionary algorithm, Uniformly distribution

1. Introduction.

Multiobjective optimization problem (MOP) has been widely applied in science, engineering, management, military and other fields [1, 2]. Generally, it is hard to find its exact solutions. Evolutionary algorithms play an important role in solving problems with multiple conflicting objectives. They can obtain a set of nondominant solutions which are close to the PF in a single run. The model of MOP is described as follows:

$$\min_{x \in D}(f_1(x), \cdots, f_m(x)) \tag{1}$$

where D is the decision space and m indicates the number of the objectives.

A variety of MOEAs [1-17] have been developed in the past decade. They can be divided into two types: one is domination-based fitness assignment such as strength Pareto evolutionary algorithm II (SPEA-II) [5], Pareto archived evolutionary strategy (PAES) [6] and nondominant sorting genetic algorithm II (NSGA-II) [10], and the other is objectives-based fitness assignment such as the weighted sum method and min-max strategy [11-16]. MOEA/D [11] performs the MOP via decomposing it into several single objective optimization subproblems. It approximates the PF by using a combination of the subproblems' solutions. It features low computational complexity and fast convergence.

In many real-life applications of multiobjective optimization, the decision maker may not be interested in having an unduly large number of Pareto optimal solutions. Therefore, it is a non-trivial task to find a manageable number of Pareto optimal solutions which are evenly distributed along the PF. The existing MOEAs have adopted various strategies to obtain such solutions. For example, the algorithms in [12, 13, 15] adopt the proper weighted vectors, NSGA-II employs the crowding distance technique and the MOEA/D utilizes the advanced decomposition technique.

The MOEA/D is one of the most successful objectives-based fitness assignment MOEAs. There are two weight design methods for MOEA/D. These weight vectors are uniformly spread on the first quadrant of hyperplane $f_1 + f_2 + \cdots + f_m = 1$ or of hypersphere $f_1^2 + f_2^2 + \cdots + f_m^2 = 1$. It works well if the PF is closed to the hyperplane $f_1 + f_2 + \cdots + f_m = 1$ in the former case or to the hypersphere $f_1^2 + f_2^2 + \cdots + f_m^2 = 1$ in the later case. However, for the other PF shapes, it may not find the evenly distributed solutions. As $m = 2$, as shown in Figure 2, let (w_1^1, w_2^1), (w_1^2, w_2^2), (w_1^3, w_2^3) be three weight vectors uniformly distributed on the first quadrant with unit circle. The solutions $(f_1(x_A), f_2(x_A))$, $(f_1(x_B), f_2(x_B))$, $(f_1(x_C), f_2(x_C))$ will be found by using MOEA/D with these weight vectors. Obviously, the distance from A to B is greater than the distance from B to C. Therefore, we cannot obtain the uniformly distributed Pareto optimal solutions by using this weight design method. Additionally, for the Pareto optimal point x_D, there are a number of weight vectors such that x_D is the Pareto optimal solution. As a result, if the PF is discontinuous, there are many subproblems obtaining the same Pareto optimal solutions. It is bad to maintain the population diversity and improve the performance of the algorithm.

In order to overcome the above drawbacks, we propose a dynamic weight design method for the MOEAs based on aggregating objectives. In this method, equidistant interpolation is used to approximate the PF after several generations over time according to the projection of the current nondominant solutions. We design the weight vectors according to the mean of several neighboring interpolation points on the estimative PF. Since the interpolation points are relatively evenly distributed along the estimative PF, the points used to design weights are relatively evenly distributed along it. Some dynamic weight aggregation are presented in [18, 21, 22]. They work well in bi-objective evolutionary algorithms. However, The weight design method proposed in this paper can be applied to almost all of the objectives-based fitness assignment MOEAs. In our experimental study, we apply the dynamic weight design method to MOEA/D. The improved MOEA/D is denoted as DMOEA/D. A set of weight vectors are designed according to the geometric significance of min-max strategy. In this paper, we construct six test instances with different PF shapes to compare the performance of DMOEA/D with that of MOEA/D. The experiments have verified the feasibility and efficiency of the weight design method.

This paper is organized as follows. Section 2 gives the geometric significance of weighted sum method and the min-max strategy and proposes the weight design method for these algorithms. Section 3 approximates the PF and discusses the uniformity. Section 4 presents the DMOEA/D algorithm in detail. Section 5, we construct some test instances and use them to evaluate the performance of the weight design method. Finally, we draw a conclusion in Section 6.

2. The Dynamic Weight Design Method.

2.1. The geometric significance and the weight design method for weighted sum method.

The weighted sum method is the simplest approach and is probably the most widely used approach. This method aggregates a set of objectives into a single objective with appropriate weight vectors and then optimizes the following scalar optimization problem:

$$\min_{x \in D} \sum_{i=1}^{m} \omega_i f_i(x) \tag{2}$$

where $\omega = (\omega_1, \omega_2, \cdots, \omega_m)$ is a weight vector. Obviously, each optimal solution of Equation (2) is a Pareto optimal solution of Equation (1) [3].

For $m = 2$, the PF and the objective space of Equation (1) can be depicted in Figure 1. When the PF is convex, the normal vector of PF is $\vec{n} = (n_1, n_2)$ at point A. The straight line with normal vector $\vec{n}$ can be described as follows: $Z = \sum_{i=1}^{2} n_i f_i(x)$. It is tangent to PF at point A. Obviously, $\min\limits_{x \in D} \sum_{i=1}^{2} n_i f_i(x) = \sum_{i=1}^{2} n_i f_i(x_A)$. The x_A is one Pareto optimal solution of Equation (1).

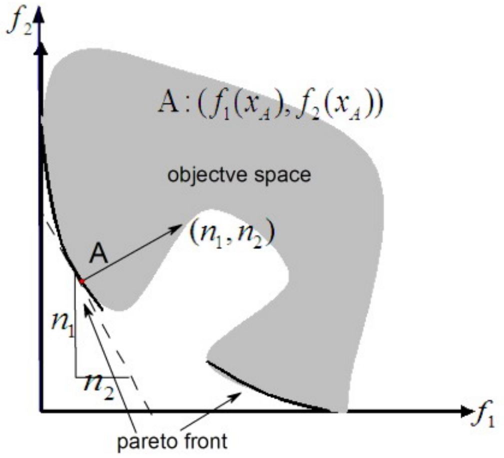


Figure 1. The geometric significance of weighted sum method

For $m = 3$, as mentioned above, the normal vector of the PF is $\vec{n} = (n_1, n_2, n_3)$ at point A. The plane with normal vector $\vec{n}$ is tangent to PF at point A. Then, x_A is a Pareto optimal solution of $\min\limits_{x \in D} \sum_{i=1}^{3} n_i f_i(x)$. Generally, in m dimension objective space, $\vec{n} = (n_1, n_2, \cdots, n_m)$ is the normal vector of the PF at point A. Since the PF is convex, the hyperplane with normal vector $\vec{n}$ will be tangent to the PF at point A. x_A is one Pareto optimal solution of $\min\limits_{x \in D} \sum_{i=1}^{m} n_i f_i(x)$.

From the above analysis of the geometric significance of the weighted sum method, it can be concluded as follows. If we obtain the normal vectors of the PF at the points evenly distributed along the PF, a set of evenly distributed solutions will be obtained by using the weighted sum method based on these normal vectors. Therefore, the main task is to find a set of evenly distributed points over the PF. The normal vector of point A can be approximated by the normal vector of the hyperplane which is determined by point A and the latest $m - 1$ points to A.

2.2. The geometric significance and the weight design method for min-max strategy.

The min-max strategy focuses on the following problem:

$$\min_{x \in D} \max_{1 \le i \le m} \{w_i f_i(x)\} \tag{3}$$

For each Pareto optimal point x^*, there is at least one weight vector w such that x^* is the optimal solution of (3) and each optimal solution of (3) is a weak Pareto optimal solution of (1). Therefore, one is able to obtain different Pareto optimal solutions by solving a set of single objective optimization problems defined by the min-max strategy with different weight vectors.

For $m = 2$, the PF and the objective space can be depicted in Figure 2. The equation of the straight line across the origin point with the direction vector $\left\{\frac{1}{w_1^1}, \frac{1}{w_2^1}\right\}$ can be described as $w_1^1 f_1 = w_2^1 f_2$. As shown in Figure 2, it is intersected with PF at point $(f_1(x_A), f_2(x_A))$. Obviously

$$\min_{x \in D} \max_{i=1,2} \left\{w_i^1 f_i(x)\right\} = \max_{i=1,2} \left\{w_i^1 f_i(x_A)\right\}$$

x_A is one Pareto optimal solution of (1) [3].

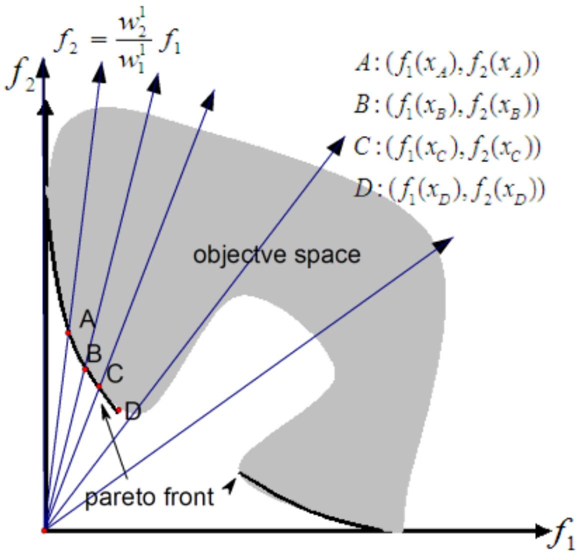


Figure 2. The geometric significance of min-max strategy

Through the above analysis of the straight line $w_1^* f_1 = w_2^* f_2$ in the objective space, we can obtain the following conclusions:

a. If it is intersected with PF at point $(f_1(x^*), f_2(x^*))$, then x^* is the solution of the $\min\limits_{x \in D} \max\limits_{i=1,2} \{w_i^1 f_i(x)\}$.

b. If it is non-intersect with PF, as shown in Figure 2, the point D is the latest from the PF to the straight line, and then x_D must be the solution of the $\min\limits_{x \in D} \max\limits_{i=1,2} \{w_i^1 f_i(x)\}$.

As $m = 3$, the equation of the straight line across the origin point with the direction vector $\left\{\frac{1}{w_1^*}, \frac{1}{w_2^*}, \frac{1}{w_3^*}\right\}$ can be described as follows:

$$w_1^* f_1 = w_2^* f_2 = w_3^* f_3$$

If the line is intersected with the PF defined in Equation (1), then the intersection point is $(f_1(x^*), f_2(x^*), f_3(x^*))$. The value x^* is the solution of $\min\limits_{x \in D} \max\limits_{1 \leq i \leq 3} \{w_i^1 f_i(x)\}$.

All in all, in m dimension objective space, the equation of the straight line across the origin point with the direction vector $\left\{\frac{1}{w_1^*}, \frac{1}{w_2^*}, \cdots, \frac{1}{w_m^*}\right\}$ can be described as follows:

$$w_1^* f_1 = w_2^* f_2 = \cdots = w_m^* f_m$$

If it is intersected with the PF defined in Equation (1), then the intersection point is $(f_1(x^*), f_2(x^*), \cdots, f_m(x^*))$. The value x^* is the solution of the $\min\limits_{x\in D}\max\limits_{1\le i\le m}\{w_i^1 f_i(x)\}$.

We can conclude from the geometric significance of the min-max strategy. If $(f_1(x_A), f_2(x_A), \cdots, f_m(x_A))$ is a point on the PF, we can obtain the Pareto optimal solution x_A according to Equation (3) with weight $\left(\frac{1}{f_1(x_A)}, \frac{1}{f_2(x_A)}, \cdots, \frac{1}{f_m(x_A)}\right)$. Let $P_1, P_2, \cdots, P_N$ be a set of points which are evenly distributed along the PF. The coordinates of these points in the objective space are $(f_1^1, f_2^1, \cdots, f_m^1), (f_1^2, f_2^2, \cdots, f_m^2), \cdots, (f_1^N, f_2^N, \cdots, f_m^N)$. We can obtain a set of evenly distributed Pareto optimal solutions by using the min-max strategy with weights $\left(\frac{1}{f_1^1}, \frac{1}{f_2^1}, \cdots, \frac{1}{f_m^1}\right), \left(\frac{1}{f_1^2}, \frac{1}{f_2^2}, \cdots, \frac{1}{f_m^2}\right), \cdots, \left(\frac{1}{f_1^N}, \frac{1}{f_2^N}, \cdots, \frac{1}{f_m^N}\right)$.

3. **Approximate the PF and Discuss the Uniformity.** From the analysis above, if there are a set of evenly distributed points along the PF, we can obtain a set of evenly distributed solutions by using the above weight design approach. However, the PF is unknown in advance. We can approximately estimate the PF by using the current nondominant solutions, and then uniformly generate some points on the estimated PF. The detailed description of the weight design approach is stated as follows.

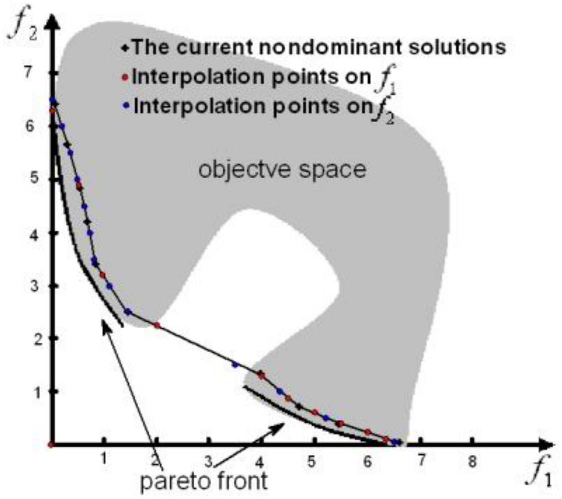


FIGURE 3. Illustration of weight design method for two objectives

For $m = 2$, the current nondominant solutions, represented by rhombic black points in Figure 3, are the nodes of interpolation P. Let the number of the interpolation points be $10N$, where N is the size of the population. We compute the number of the interpolation points to ith objective function values $N_i = \frac{10N \times D_i}{\sum_{j=1}^{2} D_j}$, where D_j is the maximum value of the projection of the current nondominant solutions on the ith sub objective. We fit a curve to the data in P by using piecewise linear interpolation (marked with solid line). N_1 interpolation points, of which the projections on the first sub objective are uniform, are generated. And then we delete the points of which the distance between them to the nearest nodes of interpolation is bigger than $\frac{D_1}{10}$. We can obtain some points which are represented by red solid circles in Figure 3. The other N_2 interpolation points are generated by the same method, of which the projections on the second sub objective are uniform. M points generated by these two interpolations is served as an approximate estimate of the PF. These points are ordered by f_1. And then $\left[\frac{M}{N}\right]$ or $\left[\frac{M}{N}\right]+1$ points which are adjacent in f_1 are divided into a group, where $[x]$ identify the largest integer less than or equal to x. These M points are divided into N groups. The mean of the points in the same group is considered as a point to design weight.

Generally, in m dimension objective space, D_i is the maximin value of the projection of the current nondominant solutions on the ith sub objective, for $i = 1, 2, \cdots, m$. Let the number of the interpolation points be $10^{m-1}N$. $N_i = \frac{10^{m-1}N\prod_{j=1,j\neq i}^{m} D_j}{\sum_{i=1}^{m}\prod_{j=1,j\neq i}^{m} D_j}$ is the number of interpolation points to ith coordinate plane $\prod_{j=1,j\neq i}^{m} D_j$, for $i = 1, \cdots, m$. We fit a hypersurface of the form $f_i = g_i(f_1, \cdots, f_{i-1}, \cdots, f_{i+1}, \cdots, f_m)$ to the data in the current nondominant solutions by using piecewise linear interpolation. N_i interpolation points are generated by using linear griddata method, of which the projections on ith coordinate plane are uniform. All the points generated by these m interpolations are served as an approximate estimate of the PF. The mean of some adjacent points is considered as a point to design weights.

The uniformity of the points used to estimate the PF is discussed in detail below. It confirms that the points for designing weights are evenly distributed along the estimated PF.

Theorem 3.1. *Let γ be a hyperplane in m dimension space, of which the area is S, $\tilde{S}$ is the total area of the projections of γ on to each coordinate plane.*
1) $\tilde{S}$ gets the maximum value $\sqrt{m}S$ if and only if the angle contained by γ and every coordinate plane is the same.
2) $\tilde{S}$ gets the minimum value S if and only if γ is parallel to one coordinate plane.

Proof: Let the unit normal vector of γ be $\vec{n} = (n_1, n_2, \cdots, n_m)$, $\sum_{i=1}^{m} n_i^2 = 1$. The normal vector of each coordinate plane is $\vec{e_i} = (0, \cdots, 1_i, \cdots, 0)$ for $i = 1, 2, \cdots, m$. $\vec{e_i}$ represents a vector that ith component is 1 and the other components are 0. The area of the projection of γ on to ith coordinate plane is $S_i = \vec{n}.\vec{e_i}S = n_iS$ for $i = 1, 2, \cdots, m$.

1. Proof of Conclusion 1

Since the angles contained by γ and each coordinate plane is the same, then $\vec{e_1}.\vec{n} = \vec{e_2}.\vec{n} = \cdots = \vec{e_m}.\vec{n}$. That is, $n_i = \frac{1}{\sqrt{m}}$ for $i = 1, 2, \cdots, m$. Thus, $\tilde{S} = \sum_{i=1}^{m} S_i = \sum_{i=1}^{m} n_iS = \sqrt{m}S$. Using the Cauchy inequality, we then have $\sum_{i=1}^{m} n_i \leq \sqrt{m\left(\sum_{i=1}^{m} n_i^2\right)} = \sqrt{m}$. If and only if $n_i = \frac{1}{\sqrt{m}}$, the equality is satisfied. That means the maximum value of $\tilde{S}$ is $\sqrt{m}S$.

2. Proof of Conclusion 2

Since γ is parallel to one coordinate plane, let γ be parallel to ith coordinate plane. That is, $n_i = 1$, $n_j = 0$, $j \neq i$ for $j = 1, 2, \cdots, m$. Since $0 \leq n_i \leq 1$, then $n_i^2 \leq n_i$. If and only if $n_i = 1$ or $n_i = 0$, the equality is satisfied. That means the minimum value of $\tilde{S}$ is $\tilde{S} = \sum_{i=1}^{m} S_i = \sum_{i=1}^{m} n_iS \geq \sum_{i=1}^{m} n_i^2S = S$.

The PF is a hypersurface in m dimension space. Suppose the area of a small element which is cut out of the hypersurface is S. When the small element is smaller, it could be considered as a hyperplane. Since the uniform grid interpolation is used in the projections of the small element on to each coordinate plane, the number of the interpolation points N is proportional to the total area of the projections. It is given as

$$N = \frac{\tilde{S}}{d^{m-1}} \tag{4}$$

where d is the distance of two adjacent grid. From Theorem 3.1, it can be seen that the maximum value of N is $N_{\max} = \frac{\sqrt{m}S}{d^{m-1}}$ and the minimum value of N is $N_{\min} = \frac{S}{d^{m-1}}$. Since the hypersurface is fitted to the data in the current nondominant solutions, any small element cannot be parallel to a coordinate plane. Then the number of the interpolation points in a small element is bigger than $N_{\min}$. Therefore, if the area of both small elements is the same, the ratio between the number of projection points in them is λ. Then

$$\lambda > \frac{N_{\min}}{N_{\max}} = \frac{\sqrt{m}}{m} \tag{5}$$

It is obvious that the points for designing weights are relatively even.

For $m = 3$, the projection of a small element of which area is S on to the coordinate planes is depicted in Figure 4. The small element like a plane with normal vector $\vec{n} = (n_1, n_2, n_3)$. S_1, S_2, S_3 are the projection of the small element on to the coordinate planes. We can conclude that the number of the projection points is $N = \frac{\sum_{i=1}^{3} n_i}{d^2} S$ by Equation (4). From the analysis above, It can be concluded as follows:

a. If the PF is a plane, the normal vector at any point of the PF is the same, then we can obtain uniformly distributed interpolation points along the PF.

b. If the PF is in other shapes, The ratio between the number of projection points in any both small elements with the same area is bigger than $\frac{\sqrt{3}}{3}$. Thus, the distribution of the interpolation points is relatively even.

c. The number of the interpolation points is not related to the position of the small element in the objective space. Even if the different objectives take different scaling, it can also obtain a set of uniformly distributed interpolation points along the PF.

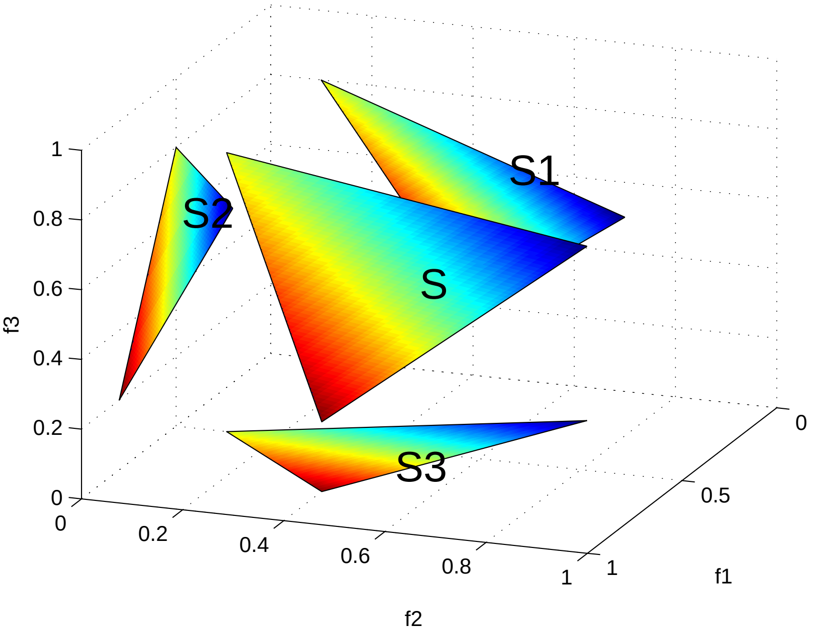


FIGURE 4. The projection of a small element on to the coordinate planes

4. **Dynamic Weight Design Method Applied to MOEA/D.** The proposed dynamic weight design method based on the current nondominant solutions can be applied to almost all of the objectives-based fitness assignment MOEAs. Since the objective is to study the feasibility and efficiency of the dynamic weight design method, we only apply the weight design method to MOEA/D.

We denote the MOEA/D with the dynamic weight design method as DMOEA/D. The steps of the algorithm DMOEA/D are as follows:

```
Algorithm of DMOEA/D
Step 1 Initialization:
    Step 1.1 give the current generation $gen = 1$, the maximum number of evolution generation $Max\_gen$ and the weight vectors updated frequency $F$;
    Step 1.2 set the number of the subproblems $N$, $N$ initialized weight vectors $\lambda^1, \cdots, \lambda^N$, the number of the weight vectors in the neighborhood of each weight vector $T$;
    Step 1.3 compute the Euclidean distances between any two weight vectors and then work out the $T$ closest weight vectors. For each $i = 1, \cdots, N$, set $B(i) = \{i_1, \cdots, i_T\}$, where $\lambda^{i_1}, \cdots, \lambda^{i_T}$ are the $T$ closest weight vectors to $\lambda^i$;
    Step 1.4 generate an initial population $x^1, \cdots, x^N$ randomly and initialize the reference point $z = (z_1, \cdots, z_m)^T$.
Step 2 Update:
    For $i = 1, \cdots, N$, do
    Step 2.1 randomly select two indexes $k, l$ from $B(i)$, and then generate a new solution $y$ from $x^k$ and $x^l$ by using genetic operator;
    Step 2.2 update of $z$: For each $j = 1, \cdots, m$, if $z_j < f_j(y)$, then set $z_j = f_j(y)$;
    Step 2.3 update of neighboring solutions: For each index $j \in B(i)$, if $y$ dominant $x^j$, then set $x^j = y$.
Step 3 $gen = gen + 1$. If $\mathrm{mod}(gen, F) = 0$, update the weight vectors $\lambda^1, \cdots, \lambda^N$ and $B(i)$.
Step 4 If $gen < Max\_gen$, go to Step 2. Otherwise, go to Step 5.
Step 5 Stop and output the Pareto optimal solutions.
```

5. Experiments.

The DTLZ suite of benchmark problems [19] are easy to scalable to any number of objectives and control the PF shapes. However, the PS shapes of DTLZ test suite are often strikingly simple. Zhang devised a way of controlling the PS shapes in [14]. We constructed four 2-objective test instances F1-F4 and two 3-objective test instances F5 and F6 by combination of these methods. Its objectives to be minimized take the following form:

$$\begin{cases} f_1(x) = (1 + g(x_{II} - \beta(x_I)))\alpha_1(x_I) \\ \quad\vdots \\ f_m(x) = (1 + g(x_{II} - \beta(x_I)))\alpha_m(x_I) \end{cases}$$

where $x = (x_1, \cdots, x_n) \in \prod_{i=1}^{n}[l_i, u_i]$, l_i is the low boundary of x_i and u_i is the upper boundary of x_i. $x_I = (x_1, \cdots, x_{m-1})$ and $x_{II} = (x_m, \cdots, x_n)$ are two subvectors of x. g is a function from R^{n-m+1} to R^+. $g(z) = 0$ if and only if $z = 0$. β is a function from $\prod_{i=1}^{m-1}[l_i, u_i]$ to $\prod_{i=m}^{n}[l_i, u_i]$. α_i $(i = 1, \cdots, m)$ are functions from $\prod_{i=1}^{m-1}[l_i, u_i]$ to R. g is used to change the complexity of the test instances. α_i are to control the PF shapes and β is to control the PS shapes. In this paper, $g = \sum_{i=2}^{n}(x_i - \sin(2\pi x_1 + \frac{i}{n}\pi))^2$, $x \in [0, 1] \times [-1, 1]^{n-1}$, $n = 10$ for the 2-objective test instances and $g = \sum_{i=3}^{n}(x_i - \sin(2\pi x_1 + \frac{i}{n}\pi))^2$, $x \in [0, 1]^2 \times [-1, 1]^{n-2}$, $n = 10$ for the 3-objective test instances. The setting of α_i and the PF shapes are shown in Table 1.

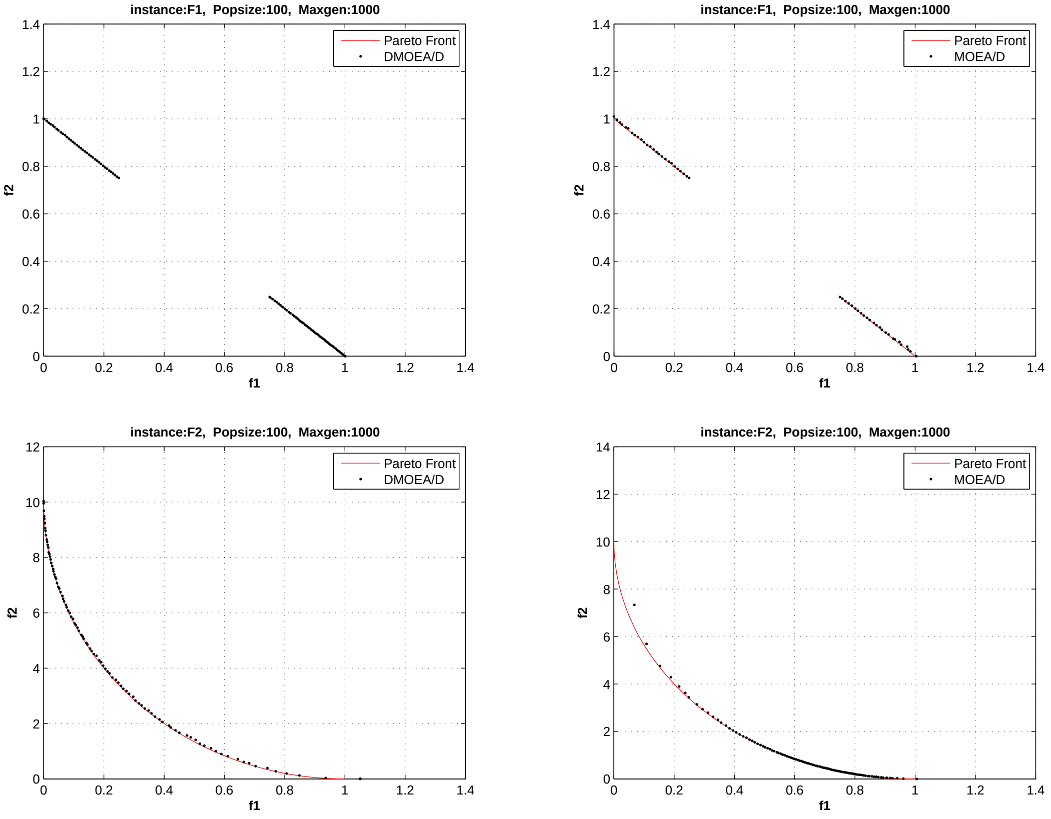
In order to evaluate the performance of algorithm in a quantitative way, the IGD-metric [14, 15, 20] is used in this paper. Suppose Q^* is the set of points uniformly distributed along the PF and Q is an approximation to the PF. The distance between the Q^* and Q can be defined as:

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

TABLE 1. The setting of α_i and the PF shapes for all test instances

Instance	The setting of α_i	The PF shape
F1	$\alpha_1 = x_1$ $\alpha_2 = 2 - x_1 - sign(\cos(2\pi x_1))$	Linear, Disconnected
F2	$\alpha_1 = 1 - \cos\left(\frac{x_1}{2}\pi\right)$ $\alpha_2 = 10 - 10\sin\left(\frac{x_1}{2}\pi\right)$	Convex, Different value range
F3	$\alpha_1 = x_1$ $\alpha_2 = \begin{cases} 1 - 19x_1 & \text{if } x_1 \leq 0.05 \\ \frac{1}{19} - \frac{x_1}{19} & \text{else} \end{cases}$	Convex
F4	$\alpha_1 = x_1$ $\alpha_2 = 2 - 2x_1^{0.5}\cos(3x_1^2\pi)^2$	Convex, Disconnected, Different value range
F5	$\alpha_1 = \left(1 - \cos\left(\frac{x_1}{2}\pi\right)\right)\left(1 - \cos\left(\frac{x_2}{2}\pi\right)\right)$ $\alpha_2 = \left(1 - \cos\left(\frac{x_1}{2}\pi\right)\right)\left(1 - \sin\left(\frac{x_2}{2}\pi\right)\right)$ $\alpha_3 = 1 - \sin\left(\frac{x_1}{2}\pi\right)$	Convex
F6	$\alpha_1 = \left(1 - \cos\left(\frac{x_1}{2}\pi\right)\right)\left(1 - \cos\left(\frac{x_2}{2}\pi\right)\right)$ $\alpha_2 = \left(1 - \cos\left(\frac{x_1}{2}\pi\right)\right)\left(1 - \sin\left(\frac{x_2}{2}\pi\right)\right)$ $\alpha_3 = 2 - \sin\left(\frac{x_1}{2}\pi\right) - sign(\cos(4x_1\pi))$	Convex, Disconnected

where $d(v, Q)$ is the minimum Euclidean distance from the point v to Q. Obviously, the smaller value of IGD, the better performance of the algorithm. We uniformly selected 1000 points to construct the set Q^* along the PF.



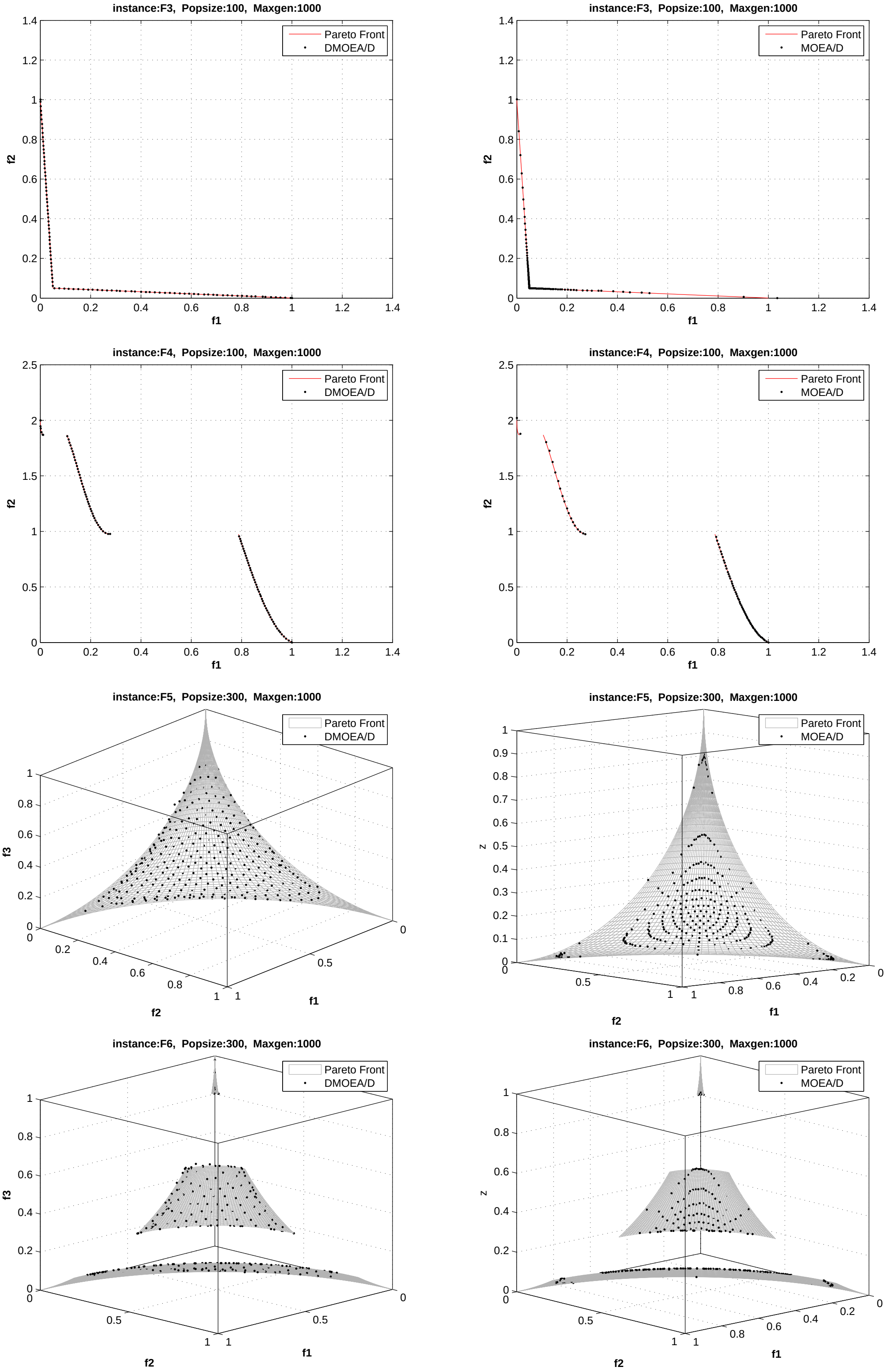


FIGURE 5. The left and right plots are respectively nondominated solutions with the smallest IGD-value in 30 runs of DMOEA/D and MOEA/D for all test instances F1-F6

In the experiment, we compare DMOEA/D with MOEA/D. The weight vectors of MOEA/D uniformly spread on the first quadrant with line $f_1 + f_2 = 1$ for 2-objective tests and on the first part with plane $f_1+f_2+f_3=1$ for 3-objective tests. Furthermore, the common parameters are set at the same values as those used in the algorithm MOEA/D [11]. The population size N is set at 100 for all the 2-objective test instances and 300 for the other two 3-objective test instances respectively. The maximum number of generation is $Max\_gen = 1000$ and the weight vectors updated frequency is $F = Max\_gen/20$. DMOEA/D and MOEA/D are run 30 times independently for each test instance. Figure 5 shows the nondominated solutions with the smallest IGD-value in 30 runs of MOEA/D and DMOEA/D. Table 2 shows the comparison result of IGD-value between these two algorithms.

Table 2. The IGD-value of DMOEA/D and MOEA/D in 30 independent runs for all test instances

IGD-value	DMOEA/D			MOEA/D		
Example	best	mean	std	best	mean	std
F1	0.002507	0.002562	2.8912e-8	0.004468	0.004575	5.1324e-8
F2	0.027132	0.027883	4.6723e-8	0.402172	0.459411	2.5671e-5
F3	0.005123	0.005559	5.3912e-8	0.028102	0.030804	8.3089e-6
F4	0.006021	0.006712	2.2371e-8	0.012371	0.014072	4.2314e-5
F5	0.022132	0.023100	9.5612e-8	0.030781	0.035962	9.7307e-6
F6	0.018923	0.019817	2.1329e-7	0.028913	0.031779	6.1645e-5

Table 2 presents the best, the mean and standard deviation of IGD-value of the final solutions obtained by both algorithms for all test instances. This table reveals that in terms of IGD-value, the final solutions obtained by DMOEA/D are better than MOEA/D for all test instances. Figure 5 shows in the objective space, the distribution of the final solutions obtained in the run with the lowest IGD-value of both algorithms for all test instances. It is evident that as to the uniformness of final solutions, DMOEA/D is also better than MOEA/D for all instances. We can conclude from the above results that the dynamic weight design method can improve the uniformness of final solutions for various PF shapes. Although the shape of the PF is simple in F1, the final solutions obtained by DMOEA/D are still better than MOEA/D because there are many subproblems obtaining the same Pareto optimal solution in MOEA/D. When the different objectives take different scaling in instance F2, MOEA/D should work well if the objective normalization is considered. But DMOEA/D still obtain a set of uniformly distributed Pareto optimal solutions along the PF without normalizing objectives.

6. **Conclusion.** In this paper, we have proposed a dynamic weight design method based on the projection of the current nondominant solutions and equidistant interpolation. It can be applied to almost all of the MOEAs which are based on aggregating objectives. The numerical simulations on seven difficult problems have been shown that MOEA/D with the dynamic weight design method is able to generate a very uniform distribution of Pareto optimal solutions on the PF.

Acknowledgment. The authors are grateful to Q. Zhang for his help on this work. This work was jointly supported by the Natural Science Foundation of China (60974077), the Natural Science Foundation of Guangdong Province (S2011030002886, 10251009001000002).

REFERENCES

[1] C. A. C. Coello, D. A. V. Veldhuizen and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer, Norwell, MA, 2002.

[2] F. Neri and E. Mininno, Mimetic compact differential evolution for cartesian robot control, *IEEE Computational Intelligence Magazine*, vol.5, no.2, pp.54-65, 2010.

[3] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, New York, 2001.

[4] 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, 1999.

[5] E. Zitzler, M. Laumanns and L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, *Proc. of Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems Conference*, Athens, Greece, pp.95-100, 2001.

[6] J. D. Knowles and D. W. Corne, The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization, *Proc. of Congress on Evolutionary Computation*, Washington D.C., USA, pp.98-105, 1999.

[7] Q. Zhang, Reactive search and intelligent optimization, *IEEE Computational Intelligence Magazine*, vol.5, no.1, pp.59-60, 2010.

[8] V. Vegh, G. K. Pierens and Q. M. Tieng, A variant of differential evolution for discrete optimization problems requiring mutually distinct variables, *International Journal of Innovative Computing, Information and Control*, vol.7, no.2, pp.897-915, 2011.

[9] S. M. Fard, A. Hamzeh and K. Ziarati, A new cooperative co-evolutionary multiobjective algorithm for function optimization, *International Journal of Innovative Computing, Information and Control*, vol.7, no.5(A), pp.2529-2542, 2011.

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

[11] 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, 2007.

[12] H.-L. Liu, Y. Wang and Y.-M. Cheung, A multiobjective evolutionary algorithm using min-max strategy and sphere coordinate transformation, *Intelligent Automation and Soft Computing*, vol.15, pp.361-384, 2009.

[13] Y. Leung and Y. Wang, Multiobjective programming using uniform design and genetic algorithm, *IEEE Transactions on SMC, Part C*, vol.30, no.3, pp.293-304, 2000.

[14] H. Li and Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, *IEEE Trans. Evol. Comput.*, vol.12, no.2, pp.284-302, 2009.

[15] H.-L. Liu and X. Li, The multiobjective evolutionary algorithm based on determined weight and sub-regional search, *Proc. of Congress on Evolutionary Computation*, Norway, pp.1928-1934, 2009.

[16] H.-L. Liu, F. Gu and Y. Cheung, T-MOEA/D: MOEA/D with objective transform in multi-objective problems, *Proc. of International Conf. on Information Science and Management Engineering*, Xi'an, China, pp.282-285, 2010.

[17] C. A. C. Coello, G. T. Pulido and M. S. Lechuga, Handling multiple objectives with particle swarm optimization, *IEEE Trans. Evol. Comput.*, vol.8, no.3, pp.256-279, 2004.

[18] F. Gu and H.-L. Liu, A novel weight design in multiobjective evolutionary algorithm, *Proc. of International Conf. on Computational Intelligence and Security*, Nanning, China, pp.137-141, 2010.

[19] S. Huband, P. Hingston, L. Barone and L. While, A review of multiobjective test problems and a scalable test problem toolkit, *IEEE Trans. Evol. Comput.*, vol.10, no.5, pp.477-506, 2006.

[20] 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, 2003.

[21] D. Lim, Y. Jin, Y. S. Ong and B. Sendhoff, Generalizing surrogate-assisted evolutionary computation, *IEEE Trans. Evol. Comput.*, vol.14, no.3, pp.329-355, 2010.

[22] Y. Jin, M. Olhofer and B. Sendhoff, Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how? *Proc. of Genetic and Evolutionary Computation Conference*, San Francisco, CA, USA, pp.1042-1049, 2001.