Information Sciences 213 (2012) 14–38

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

A modification to MOEA/D-DE for multiobjective optimization problems with complicated Pareto sets

Yan-Yan Tan [a,*], Yong-Chang Jiao [a], Hong Li [b], Xin-Kuan Wang [a]

[a] School of Electronic Engineering, Xidian University, Xi'an, Shaanxi 710071, PR China
[b] School of Science, Xidian University, Xi'an, Shaanxi 710071, PR China

ARTICLE INFO

Article history:
Received 31 October 2010
Received in revised form 28 May 2012
Accepted 3 June 2012
Available online 9 June 2012

Keywords:
Multiobjective optimization
Decomposition
Differential evolution
Uniform design
Test problems

ABSTRACT

This paper presents an algorithm, named the uniform design multiobjective differential algorithm based on decomposition (UMODE/D), for optimizing multiobjective problems. The algorithm is a modification to the new version of MOEA/D based on differential evolution (DE), i.e., MOEA/D-DE proposed by Li and Zhang (2009) [20]. Its distinguishing features include: (1) The uniform design method is applied to generate the aggregation coefficient vectors so that the decomposed scalar optimization subproblems are uniformly distributed, and therefore the algorithm could explore uniformly the region of interest from the initial iteration; (2) The simplified quadratic approximation with three best points is employed to improve the local search ability and the accuracy of the minimum scalar aggregation function value. UMODE/D is compared with the original MOEA/D-DE and NSGA-II by solving a wide set of problems with complicated Pareto set shapes in this paper. Experimental results indicate that UMODE/D significantly outperforms MOEA/D-DE and NSGA-II on these test problems. Two sets of experiments are carried out to illustrate the efficiency of the uniform design method and the simplified quadratic approximation separately. In addition, UMODE/D is tested on CEC 2009 problems and combinatorial knapsack problems. Experimental results show that the proposed algorithm performs better than the further improved MOEA/D for almost all the CEC 2009 problems, and the results obtained are very competitive when comparing UMODE/D with some other algorithms on these multiobjective knapsack problems.

1. Introduction

A multiobjective optimization problem (MOP) can be stated as follows:

$$\begin{aligned} &\text{minimize } F(x) = (f_1(x), \ldots, f_m(x))^\top \\ &\text{subject to } x \in \Omega \end{aligned} \tag{1}$$

where $x = (x_1, \ldots, x_n)^\top$ is called decision vector, Ω is the decision space, $F : \Omega \to R^m$ consists of m real-valued objective functions and R^m is called the objective space. The attainable objective set is defined as set $\{F(x)|x \in \Omega\}$. The optimal solutions of (1) can be defined in terms of Pareto optimality [24].

Let $u = (u_1, \ldots, u_m)^\top, v = (v_1, \ldots, v_m)^\top \in R^m$ be two vectors. u is said to dominate v if and only if $u_i \leqslant v_i$ for every $i \in \{1, \ldots, m\}$ and $u_j < v_j$ for at least one index $j \in \{1, \ldots, m\}$. A point $x^* \in \Omega$ is Pareto optimal to (1) if there is no $x \in \Omega$ such

* Corresponding author.
E-mail addresses: yytan@mail.xidian.edu.cn, tyy_820812@yahoo.com.cn (Y.-Y. Tan), ychjiao@xidian.edu.cn (Y.-C. Jiao), lihong@mail.xidian.edu.cn (H. Li), wxkuan@126.com (X.-K. Wang).

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.06.007

that $F(x)$ dominates $F(x^*)$. The set of all the Pareto optimal points is called the Pareto set (PS), and the set of all the objective vectors corresponding to the PS is called the Pareto front (PF), where $PF = \{F(x^*) = (f_1(x^*), \ldots, f_m(x^*))^\top | x^* \in PS\}$ [24].

Multiobjective evolutionary algorithms (MOEAs) have been recognized to be well-suited for MOPs since early because they can process a set of solutions in parallel, thereby can obtain an approximation of the PF which consists of multiple Pareto optimal solutions in a single run. Up to present, various MOEAs have been proposed to solve MOPs effectively [1,4,5,9,15,19,23,25,26,30,31,33,34,37–40].

MOEA/D is a very recent one using decomposition [34]. MOEA/D explicitly decomposes the MOP (1) into a number of scalar optimization subproblems, which solves these subproblems simultaneously by evolving a population of solutions. At each generation, the population is composed of the best solutions found so far for each subproblem. It has been proved that MOEA/D has a lower complexity than NSGA-II [9], the most popular MOEA, at each iteration [20,34], and its further improved version has been ranked first among 13 entries in the unconstrained MOEA competition in CEC 2009 [35]. Therefore, its study is of great significance for the MOEAs.

In MOEA/D, each subproblem is solved by using information only from its neighboring subproblems. Neighborhood relations among subproblems are defined based on the distances between their aggregation coefficient vectors. The optimal solutions to two neighboring subproblems should be very similar. Thus, whether aggregation coefficient vectors of the subproblems can fill the entire space evenly is essential for the quality of Pareto optimality. In MOEA/D and its new version MOEA/D-DE, let $\lambda^1, \ldots, \lambda^N$ be the aggregation coefficient vectors, subproblem i corresponds to the coefficient vector $\lambda^i = (\lambda^i_1, \ldots, \lambda^i_m)^\top$, where $\lambda^i_j \geqslant 0, j = 1, \ldots, m, \sum_{j=1}^m \lambda^i_j = 1, i = 1, \ldots, N, m$ is the number of objectives, and N is the number of subproblems. Population size N (i.e., the number of subproblems) and aggregation coefficient vectors $\lambda^1, \ldots, \lambda^N$ in MOEA/D and MOEA/D-DE are controlled by m and an integer H. More precisely, $\lambda^1, \ldots, \lambda^N$ are all the weight vectors in which each individual weight takes a value from $\{0/H, 1/H, \ldots, H/H\}$. Therefore, the number of such vectors is $N = C^{m-1}_{H+m-1}$. The aforementioned method for generating weight vectors, named the simplex-lattice design method, was proposed by Scheffé in 1958 [28], which works well in MOEA/D and MOEA/D-DE. However, the distribution of the aggregation coefficient vectors is not very uniform, and the population size N will become very large when the number of the objectives m is large. For example, to keep the precise of the weight vectors, we set H as a constant, e.g., 20, then population size will be 21, 231, 1771, 10,626 respectively for problems with 2–5 objectives according to the simplex-lattice design. Meanwhile, this design method requires population size to satisfy the restriction $N = C^{m-1}_{H+m-1}$, N increases nonlinearly with m and its value cannot be set at will, which restricts MOEA/D's use to a certain extent in MOPs. Therefore, it is necessary to consider an advanced design method to generate the weight vectors [27,34]. The similar idea of considering a proper setting of the weight vectors was also addressed by Hughes in many-objective optimization [17].

Uniform design (UD) proposed by Fang [12] is a special method which represents a combination of number theory and numerical analysis. Like many mixed breeds, it has fascinations and attractions. The UD method has been successfully applied in science, engineering and industries [2,21,22]. Ford motor company has adopted uniform designs as a standard procedure in product and process designs. Investigations have shown that uniform design performs better at estimating nonlinear problems than other designs, and is robust against model assumptions [32]. These applications illustrate that the UD method is a power tool. The essence of the UD method is to find a set of points that are uniformly distributed over the design region of interest. Note that a "uniformly distributed" set of points as stated here means roughly that the set has a small discrepancy (described in Section 2), not a set of points which are uniformly distributed in the usual statistical sense.

In order to solve MOPs efficiently, we modify the MOEA/D-DE [20] and propose the uniform design multiobjective differential algorithm based on decomposition (UMODE/D) in this paper. Our modification is twofold: (1) In the global search part, the UD method is applied to generate the aggregation coefficient vectors of the subproblems, so that the decomposed scalar optimization subproblems are uniformly distributed, and the algorithm could explore uniformly the region of interest from the initial iteration; (2) In the local search part, a simplified quadratic approximation with three best points is employed to improve the local search ability and the accuracy of the minimum function value for each subproblem. The proposed algorithm is experimented on a large set of problems with complicated PS shapes taken from [20]. For all these problems, comparisons with the original algorithm and NSGA-II are made. Two sets of experiments on these test problems are carried out to illustrate the efficiency of the uniform design method and the simplified quadratic approximation separately. UMODE/D has also been tested on CEC 2009 problems and combinatorial knapsack problems. Simultaneously, comparisons with the further improved MOEA/D [35] and some other algorithms on these test problems have been given in this paper.

The remainder of this paper is organized as follows: Section 2 describes the UD method for generating the aggregation coefficient vectors. Section 3 presents the framework of UMODE/D. Numerical experiment results and comparisons are then given in Section 4. We finally conclude our paper in Section 5.

2. Uniform design method for generating the aggregation coefficient vectors

Experiments with mixtures are experiments in which the variants are proportions of ingredients in a mixture. An example is the experiment for determining the proportion of ingredients in a polymer mixture that will produce plastic products with the highest tensile strength. Similar experiments are very commonly encountered in industries. A mixture can be represented as:

$$\lambda = (\lambda_1, \ldots, \lambda_m)^{\top} \in \left\{(\lambda_1, \ldots, \lambda_m)^{\top} : \lambda_1 + \cdots + \lambda_m = 1; \lambda_1, \ldots, \lambda_m \geqslant 0\right\} = S^{m-1} \tag{2}$$

where $m \geqslant 2$ is the number of ingredients in the mixture. Set S^{m-1} is called the $(m-1)$-dimensional simplex. For details of design and modeling in experiments with mixtures, refer to the monograph by Cornell [7] and the survey article by Chan [3]. Among the designs for experiments with mixtures, simplex-lattice design [28] has the longest history, followed by simplex-centroid design [29] and axial design [8].

- Simplex-lattice design
  Suppose that the mixture has m components. Let H be a positive integer and suppose that each component takes $(H+1)$ equally spaced places from 0 to 1, i.e.

  $$\lambda_i = 0, 1/H, 2/H, \ldots, H/H, \quad i = 1, \ldots, m.$$

  For example, when $m = 3$, we have
  $H = 1$: 3 design points (1, 0, 0), (0, 1, 0) (0, 0, 1); $H = 2$: 6 design points (1, 0, 0), (0, 1, 0), (0, 0, 1), (1/2, 1/2, 0), (1/2, 0, 1/2), (0, 1/2, 1/2); $H = 3$: 10 design points (1, 0, 0), (0, 1, 0), (0, 0, 1), (1/3, 2/3, 0), (1/3, 0, 2/3), (0, 1/3, 2/3), (2/3, 1/3, 0), (2/3, 0, 1/3), (0, 2/3, 1/3), (1/3, 1/3, 1/3). $\{m,H\}$ -simplex-lattice can be used to represent this design which has C_{H+m-1}^{m-1} design points.
- Simplex-centroid design
  In an m -component simplex-centroid design, the design points are from m pure blends, C_m^2 binary mixtures, C_m^3 ternary mixtures and so on, with the finally overall centroid point $1/m, \ldots, 1/m$, the m-nary mixture. So the total number of design points is $2^m - 1$. For example, when $m = 3$, the design points are (1, 0, 0), (0, 1, 0), (0, 0, 1), (1/2, 1/2, 0), (1/2, 0, 1/2), (0, 1/2, 1/2), and (1/3, 1/3, 1/3).
- Axial design
  The line segment joining a vertex of simplex S^{m-1} with its centroid is called an axis. Let d be a positive number such that $0 < d < (m-1)/m$. The experimental points of the axial design are m points on m axes such that each point to the centroid has the same distance d.

MOEA/D and its new version MOEA/D-DE just adopt the simplex-lattice design to set the aggregation coefficient vectors. Fig. 1a–c shows the above three designs in $m = 3$. There are at least two problems can be found with these three designs [13]:

(1) The experimental points are not very uniformly distributed on the experimental domain.
(2) There are too many experimental points at the boundary of the experimental domain.

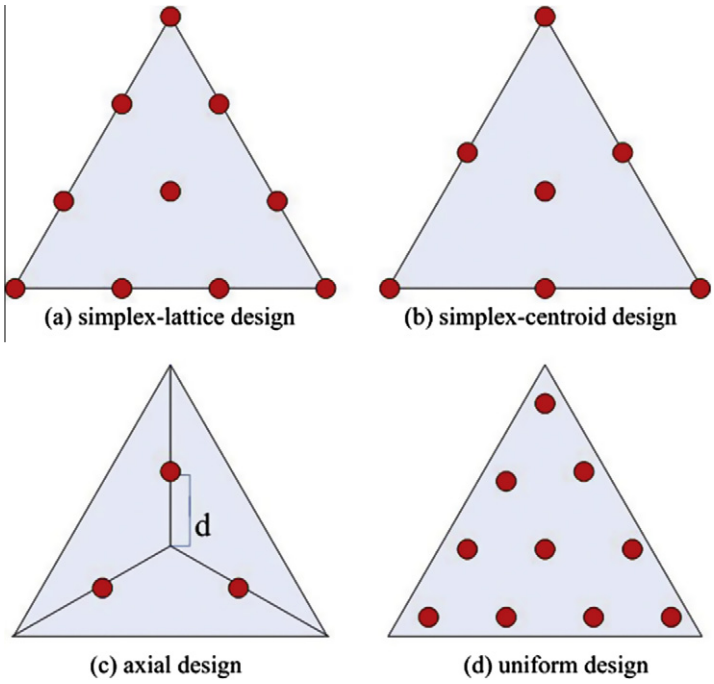

Fig. 1. Four designs of experiments with mixtures in $m = 3$.

MOEA/D and MOEA/D-DE decompose an MOP into a number of scalar optimization subproblems and solve them in parallel. The objective in each of these subproblems is an aggregation of all the objectives in the MOP under consideration. The uniformity of the aggregation coefficient vectors reflects the uniformity of the Pareto optimal solutions in a sense. MOEA/D and MOEA/D-DE use the simplex-lattice design method to set the aggregation coefficient vectors. The objective in a subproblem is an aggregation of only part not all the objectives in the MOP if a component of the scalar subproblem has zero weight value. Extremely, the objective in a subproblem will be only one objective if other objectives in the MOP all have zero weight value. So the boundary weight vectors are insignificant in that case. Therefore, the quality of the Pareto optimal solutions will be affected in MOEA/D and MOEA/D-DE if there have too many objectives with zero weight value in subproblems.

In addition, the numbers of weight vectors in these three designs are restricted. In fact, the number of weight vectors is C_{H+m-1}^{m-1} in the simplex-lattice design, 2^m-1 in the simplex-centroid design, and m in the axial design. However, in most experiments, the requirement of the number of weight vectors (i.e., the population size) is considerably flexible.

UD on S^{m-1}, proposed by Wang and Fang [13], overcomes these drawbacks and provides a more uniform coverage of the design region than the above designs. Fig. 1d shows a uniform design in $m = 3$. In this section, we will explain how UD on S^{m-1} can be constructed for generating the aggregation coefficient vectors used in the proposed algorithm. Let $\lambda^1,\ldots,\lambda^N$ be the aggregation coefficient vectors. Subproblem i corresponds to the coefficient vector $\lambda^i=(\lambda_1^i,\ldots,\lambda_m^i)^\top$, where $\lambda_j^i\geqslant 0, j=1,\ldots,m, \sum_{j=1}^m\lambda_j^i=1, i=1,\ldots,N, m$ is the number of objectives, and N is the population size (i.e., the number of aggregation coefficient vectors). The essence of the design is to find a set of points $\lambda^1,\ldots,\lambda^N$ which are uniformly distributed over the S^{m-1}. Such uniformity may be achieved by minimizing a discrepancy.

To get aggregation coefficient vectors $\lambda^1,\ldots,\lambda^N$ which are uniformly distributed over the S^{m-1}, we need construct a collection of points $x_1,\ldots,x_N$ uniformly distributed on a unit hyper-cube first. Without loss of generality, let the design space be the s -dimensional unit cube $C^s=[0,1]^s$. We represent any point in C^s by $x_i=(x_{i1},\ldots,x_{is})^\top$, where $x_{i1},\ldots,x_{is}\in[0,1]$. For a given positive integer N, a uniform design with N points in C^s is a collection of points $P^*=\{x_1^*,\ldots,x_N^*\}\subset C^s$ such that $\mathbf{M}(P^*)=\min\mathbf{M}(P)$, where the minimization is carried out over all $P=\{x_1,\ldots,x_N\}\subset C^s$ with respect to some measure of uniformity, $\mathbf{M}$. There are more than one choices for $\mathbf{M}$, for example, the star discrepancy, the most commonly used centered L_2-discrepancy (CD) and the wrap-around L_2-discrepancy (WD). The centered L_2-discrepancy (CD), denoted by $\mathrm{CD}_2(P)$ [14]

$$\begin{aligned}\mathrm{CD}_2(P)=&\left(\frac{13}{12}\right)^s-\frac{2}{N}\sum_{k=1}^N\prod_{i=1}^s\left(1+\frac{1}{2}\left|x_{ki}-\frac{1}{2}\right|-\frac{1}{2}\left|x_{ki}-\frac{1}{2}\right|^2\right)\\&+\frac{1}{N^2}\sum_{k=1}^N\sum_{j=1}^N\prod_{i=1}^s\left(1+\frac{1}{2}\left|x_{ki}-\frac{1}{2}\right|+\frac{1}{2}\left|x_{ji}-\frac{1}{2}\right|-\frac{1}{2}\left|x_{ki}-x_{ji}\right|\right)\end{aligned}\tag{3}$$

is used in our experiments, for it is convenient to compute and is invariant under relabeling of coordinate axes. The CD is also invariant under reflection of points about any plane passing through the center and parallel to the faces of the unit cube C^s, that is, invariant when the ith coordinate x_i is replaced by $1-x_i$. Corresponding formulas for other discrepancies can be found in Fang et al. [14].

In order to construct a uniform design on the continuous $C^s=[0,1]^s$,we need to search for all possible sets of N points over C^s for a design with minimum discrepancy, which is an NP-hard problem for high-power computers even if N and s are not large. In practice, we often search over the discrete set of U-type design to construct approximate uniform design [12,14]. Computation shows that this approach produces good results. Tables of UDs in the website www.math.hkbu.edu.hk/UniformDesign are constructed from U-type designs.

Definition 2.1. [14]. A U-type design is an array of N rows and s columns with entries $1,\ldots,q_j$ in the jth column such that each entry in each column appears the same number of times ($j=1,\ldots,s$). The collection of all such designs is denoted by $U(N;q_1\times\cdots\times q_s)$, which is the design space. When all q_j's are the same, the design space will be denoted by $U(N;q^s)$. Design in $U(N;q_1\times\cdots\times q_s)$ (where the q_j's are distinct) are asymmetric, while designs in $U(N;q^s)$ are symmetric.

Definition 2.2. [14]. A design $U\in U(N;q_1\times\cdots\times q_s)$ is called a uniform design under the measure of discrepancy $\mathbf{M}$ if $\mathbf{M}(U)=\min_{V\in U(N;q_1\times\cdots\times q_s)}\mathbf{M}(V)$. The collection of all such designs is denoted by $U_N(q_1\times\cdots\times q_s)$. When $q_1=\cdots=q_s, U$ will be called a symmetric design, and $U_N(q_1\times\cdots\times q_s)$ will be denoted by $U_N(q^s)$.

If $U\in U(N;q_1\times\cdots\times q_s)$ is a U-type design consisting of N points $u_1,\ldots,u_N$, where $u_i=(u_{i1},\ldots,u_{is})^\top, i=1,\ldots,N$, let $x_{ij}=(u_{ij}-0.5)/q_j$, so that $P=\{x_1,\ldots,x_N\}\in C^s$. If $\mathbf{M}$ is a discrepancy on C^s, then $\mathbf{M}(U)=\mathbf{M}(P)$ [14]. Centered L_2- discrepancy used in this paper is just defined on C^s, therefore, our question now is how to construct $U_N(q^s)$.

When $q=N$ (N is the population size) and $s=m-1$ (m is the number of objectives), good lattice point (glp) method is adopted in our algorithm for generating $U_N(N^{m-1})$. Let $(N;h_1,\ldots,h_{m-1})$ be a vector with integral components satisfying $1\leqslant h_i<N,\ h_i\neq h_j(i\neq j),\ m-1<N$ and the greatest common divisors $(N,h_i)=1,\ i=1,\ldots,m-1$. Let

$$u_{ki}\equiv kh_i(\mathrm{mod}N),\ k=1,\ldots,N;\ i=1,\ldots,m-1.\tag{4}$$

Set $S_N = \{u_k = (u_{k1}, \ldots, u_{km-1})^\top, k = 1, \ldots, N\}$ is called a lattice point set of the generating vector $(N; h_1, \ldots, h_{m-1})$. If set S_N has the smallest discrepancy among all possible generating vectors, set S_N is the uniform design $U_N\left(N^{m-1}\right)$ that we need to construct.

Let $U = (u_{ki})_{k=1,\ldots,N;i=1,\ldots,m-1}$ be the uniform design $U_N\left(N^{m-1}\right)$ constructed above. Calculate

$$c_{ki} = (u_{ki} - 0.5)/N, \quad k = 1, \ldots, N, \quad i = 1, \ldots, m-1. \tag{5}$$

Let $C^* = \{c_i = (c_{i1}, \ldots, c_{i(m-1)})^\top, i = 1, \ldots, N\}$, then C^* is the uniform design on C^{m-1}.

In order to satisfy the restriction that $\sum_{j=1}^{m} \lambda_j^i = 1,\ i = 1, \ldots, N$, the following transformation on C^* is required

$$\begin{cases} \lambda_i^k = \left(1 - c_{ki}^{\frac{1}{m-i}}\right) \prod_{j=1}^{i-1} c_{kj}^{\frac{1}{m-j}} & i = 1, \ldots, m-1 \\ \lambda_m^k = \prod_{j=1}^{m-1} c_{kj}^{\frac{1}{m-j}} \end{cases} \tag{6}$$

for each $k = 1, \ldots, N$. Then $\{\lambda^k = (\lambda_1^k, \ldots, \lambda_m^k), k = 1, \ldots, N\}$ is a uniform design on S^{m-1}, thus the uniformly distributed weight vectors $\lambda^1, \ldots, \lambda^N$ are achieved.

3. Uniform design multiobjective differential algorithm based on decomposition (UMODE/D)

Under the framework of MOEA/D-DE [20], UMODE/D tries using the uniform design (UD) method to set the aggregation coefficient vectors. With the UD, the distribution of the weight vectors is more uniform over the design region, and the population size will not be controlled by H (a precision parameter) and m (the number of objectives). On the other hand, the simplified quadratic approximation with three best points, as a local search strategy, will be adopted in UMODE/D for improving the accuracy of the estimated minimum scalar aggregation function value. Three-point quadratic approximation works well in global optimization problems [18], which makes a better use of the values of the objective function already evaluated. Simulation results show that it is efficient in reducing the number of generations in UMODE/D. Overall, our modification to MOEA/D-DE is twofold: (1) In the global search part, the UD method is applied to generate the aggregation coefficient vectors of the subproblems so that the decomposed scalar optimization subproblems are uniformly distributed, and the algorithm could explore the region of interest uniformly from the initial iteration and (2) in the local search part, the simplified quadratic approximation with three best points is employed to improve the local search ability and the accuracy of the minimum function value for each subproblem.

There are several approaches for converting an MOP into a number of scalar optimization problems [24]. Tchebycheff approach is mainly employed in our experimental study. Let $\lambda^1, \ldots, \lambda^N$ be a set of evenly distributed weight vectors and z^* be the reference point. With the Tchebycheff approach, the objective function of the i-th subproblem is in the form [24]:

$$g^{te}(x|\lambda^i, z^*) = \max_{1 \leqslant j \leqslant m} \left\{ \lambda_j^i \left| f_j(x) - z_j^* \right| \right\}, \tag{7}$$

where $\lambda^i = \left(\lambda_1^i, \ldots, \lambda_m^i\right)^\top$. UMODE/D minimizes all these N objective functions simultaneously in a single run. At each generation, UMODE/D maintains the following:

- A population of N points $x^1, \ldots, x^N \in \Omega$, where x^i is the current solution to the ith subproblem;
- $FV^1, \ldots, FV^N$, where FV^i is the F-value of x^i, i.e., $FV^i = F(x^i)$ for each $i = 1, \ldots, N$;
- $z = (z_1, \ldots, z_m)^\top$, where z_i is the best value found so far for objective $f_i, i = 1, \ldots, N$.

Algorithm UMODE/D

Step 1: Initialization
- Step 1.1: Apply the UD method to generate the weight vectors $\lambda^1, \ldots, \lambda^N$.
- Step 1.2: Compute the Euclidean distances between any two weight vectors and then work out the T closest weight vectors to each weight vector. For each $i = 1, \ldots, N$, set $B(i) = \{i_1, \ldots, i_T\}$, where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the T closest weight vectors to λ^i.
- Step 1.3: Generate an initial population $x^1, \ldots, x^N$ by randomly sampling from Ω. Set $FV^i = F(x^i)$.
- Step 1.4: Initialize $z = (z_1, \ldots, z_m)^\top$ by setting $z_j = \min_{1 \leqslant i \leqslant N} f_j(x^i)$.

Step 2: Update: For $i = 1, \ldots, N$, do

Step 2.1: Selection of Mating/Update Range: Generate a number *rand* from [0, 1] randomly. Then set

$$P = \begin{cases} B(i) & r \text{ and } < \delta \\ \{1, \ldots, N\} & \text{otherwise} \end{cases}.$$

$\delta$is the probability that parent solutions are selected from the neighborhood.

Step 2.2: Reproduction: Set $r_b = i$ and randomly select two indexes r_1, r_2 from P, generate a solution $\bar{y}$ from x^{r_b} and x^{r_1}, x^{r_2} by a DE operator, and then perform the polynomial mutation operator on $\bar{y}$ with probability p_m to produce a new solution y.

Step 2.3: Repair: If an element of y is out of the boundary of Ω, its value is reset to be a randomly selected value inside the boundary.

Step 2.4: Update of z: For each $j = 1, \ldots, m$, if $z_j > f_j(y)$, then set $z_j = f_j(y)$.

Step 2.5: Update of Solutions: Set $c = 0$ and then do the following:

(1) If $c = \eta_r$ or P is empty, go to Step 2.6. Otherwise, randomly pick an index j from P.

Table 1
Comparison results for illustrate the efficiency of the uniform design method.

Problem	UMODE/D (case 1)			MOEA/D-DE		
	Mean	Min	Std.	Mean	Min	Std.
F1	0.0014	0.0014	0	0.0014	0.0014	0
F2	0.0130	0.0063	0.0080	0.0234	0.0079	0.0189
F3	0.0135	0.0046	0.0220	0.0239	0.0074	0.0230
F4	0.0162	0.0064	0.0071	0.0099	0.0047	0.0034
F5	0.0144	0.0085	0.0039	0.0225	0.0104	0.0161
F6	0.0290	0.0266	0.0026	0.0307	0.0287	0.0013
F7	0.0017	0.0015	0.0001	0.0017	0.0015	0.0002
F8	0.1665	0.1083	0.0449	0.1482	0.0741	0.0575
F9	0.0190	0.0089	0.0106	0.0314	0.0114	0.0181

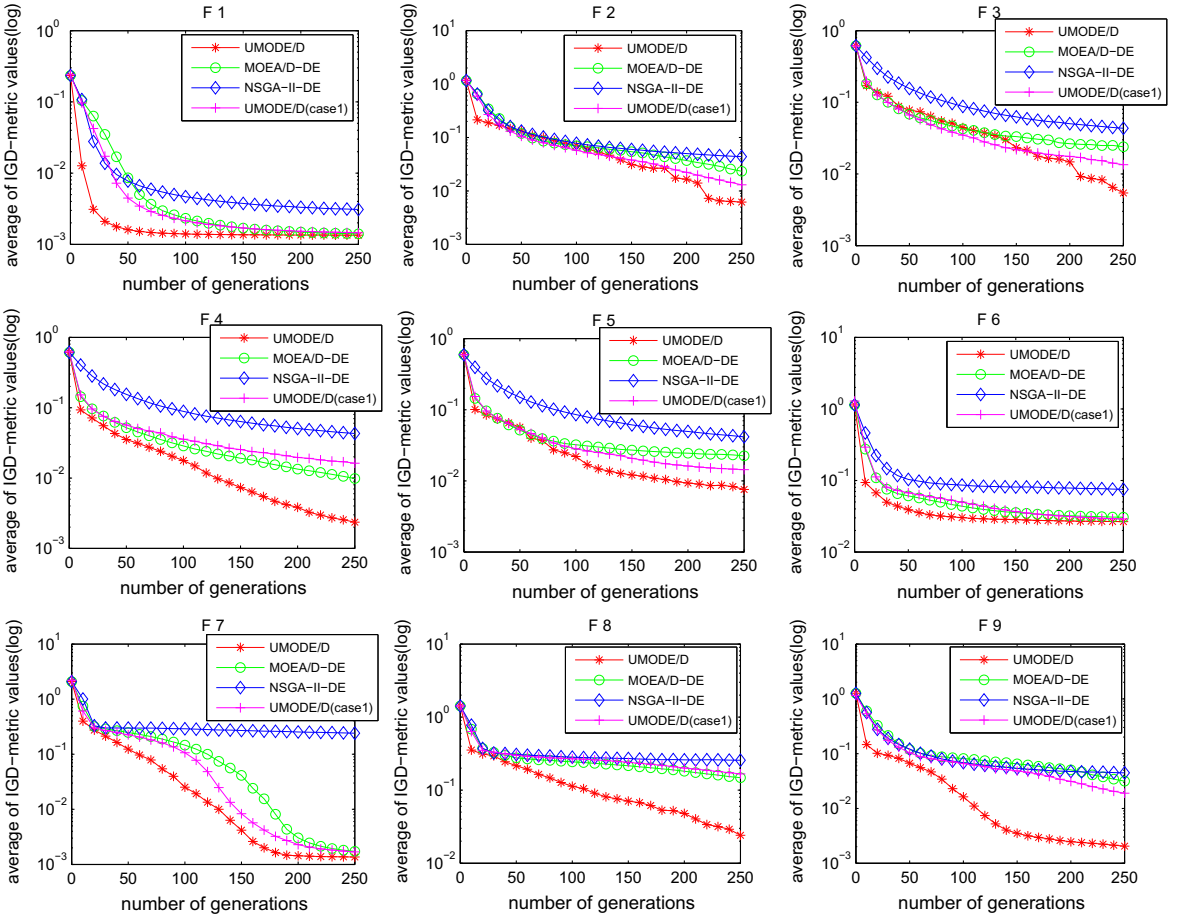

Fig. 2. Evolution of the mean of IGD-metric values obtained by UMODE/D, UMODE/D (case 1), MOEA/D-DE and NSGA-II-DE for F1–F9.

(2) If $g^{te}\left(y|\lambda^j,z\right) < g^{te}\left(x^j|\lambda^j,z\right)$, then set $x^j = y, FV^j = F(y)$ and $c = c + 1$.
(3) Remove j from P and go to (1).

Step 2.6 Local Search: Compute $g^{te}\left(x^{i_k}|\lambda^i,z\right)$ for each $k = 1,\ldots,T$, where $i_k \in B(i)$. Set $G = \left\{g^{te}\left(x^{i_k}\middle|\lambda^i,z\right), k = 1,\ldots,T\right\}$, and select three best points x^{i_b}, x^{i_a} and x^{i_c} corresponding to three smallest objective function values in G, where

$$g^{te}\left(x^{i_b}\middle|\lambda^i,z\right) < g^{te}\left(x^{i_a}\middle|\lambda^i,z\right) < g^{te}\left(x^{i_c}\middle|\lambda^i,z\right).$$

Table 2
Comparison results for illustrate the efficiency of the simplified quadratic approximation.

Problem	UMODE/D (case 2)			MOEA/D-DE		
	Mean	Min	Std.	Mean	Min	Std.
F1	0.0013	0.0013	0	0.0014	0.0014	0
F2	0.0111	0.0042	0.0063	0.0234	0.0079	0.0189
F3	0.0138	0.0039	0.0198	0.0239	0.0074	0.0230
F4	0.0090	0.0042	0.0035	0.0099	0.0047	0.0034
F5	0.0168	0.0082	0.0188	0.0225	0.0104	0.0161
F6	0.0315	0.0288	0.0028	0.0307	0.0287	0.0013
F7	0.0383	0.0015	0.0357	0.0017	0.0015	0.0002
F8	0.1216	0.0412	0.0898	0.1482	0.0741	0.0575
F9	0.0181	0.0069	0.0107	0.0314	0.0114	0.0181

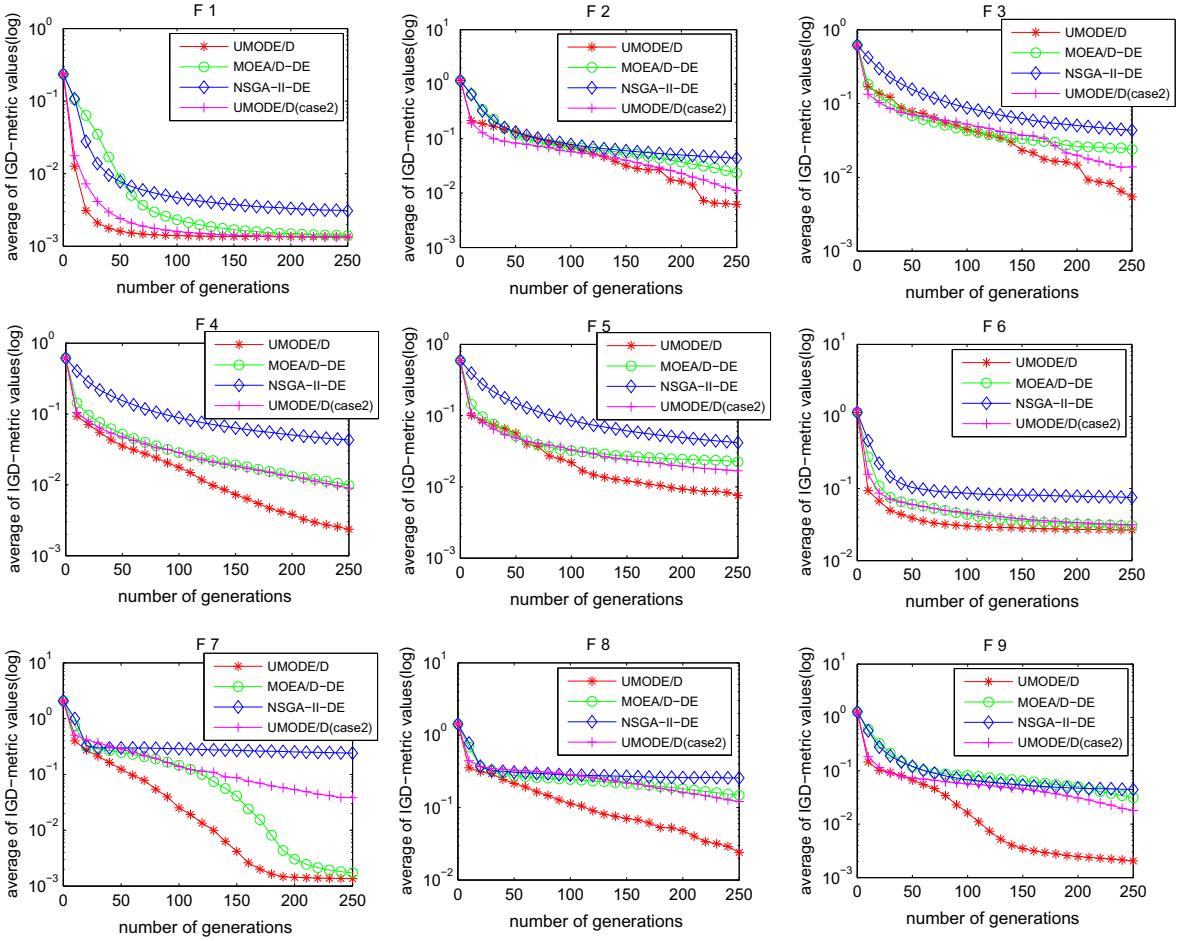

Fig. 3. Evolution of the mean of IGD-metric values obtained by UMODE/D, UMODE/D (case 2), MOEA/D-DE and NSGA-II-DE for F1–F9.

$g^{te}\left(x^{i_k}\middle|\lambda^i, z\right)$ abbreviated here as $g^{te}_{i_k}$. Let

$$x^{i_b} = \left(x^{i_b}_1, \ldots, x^{i_b}_n\right)^\top, \quad x^{i_a} = \left(x^{i_a}_1, \ldots, x^{i_a}_n\right)^\top \quad \text{and} \quad x^{i_c} = \left(x^{i_c}_1, \ldots, x^{i_c}_n\right)^\top,$$

n is the number of variables. Compute the trial vector $x^{i'} = \left(x^{i'}_1, x^{i'}_2, \ldots, x^{i'}_n\right)^\top$ in the following:
If $\left(x^{i_b}_j - x^{i_c}_j\right) g^{te}_{i_a} + \left(x^{i_c}_j - x^{i_a}_j\right) g^{te}_{i_b} + \left(x^{i_a}_j - x^{i_b}_j\right) g^{te}_{i_c} < \epsilon$, then set
$x^{i'}_j = x^i_j$,

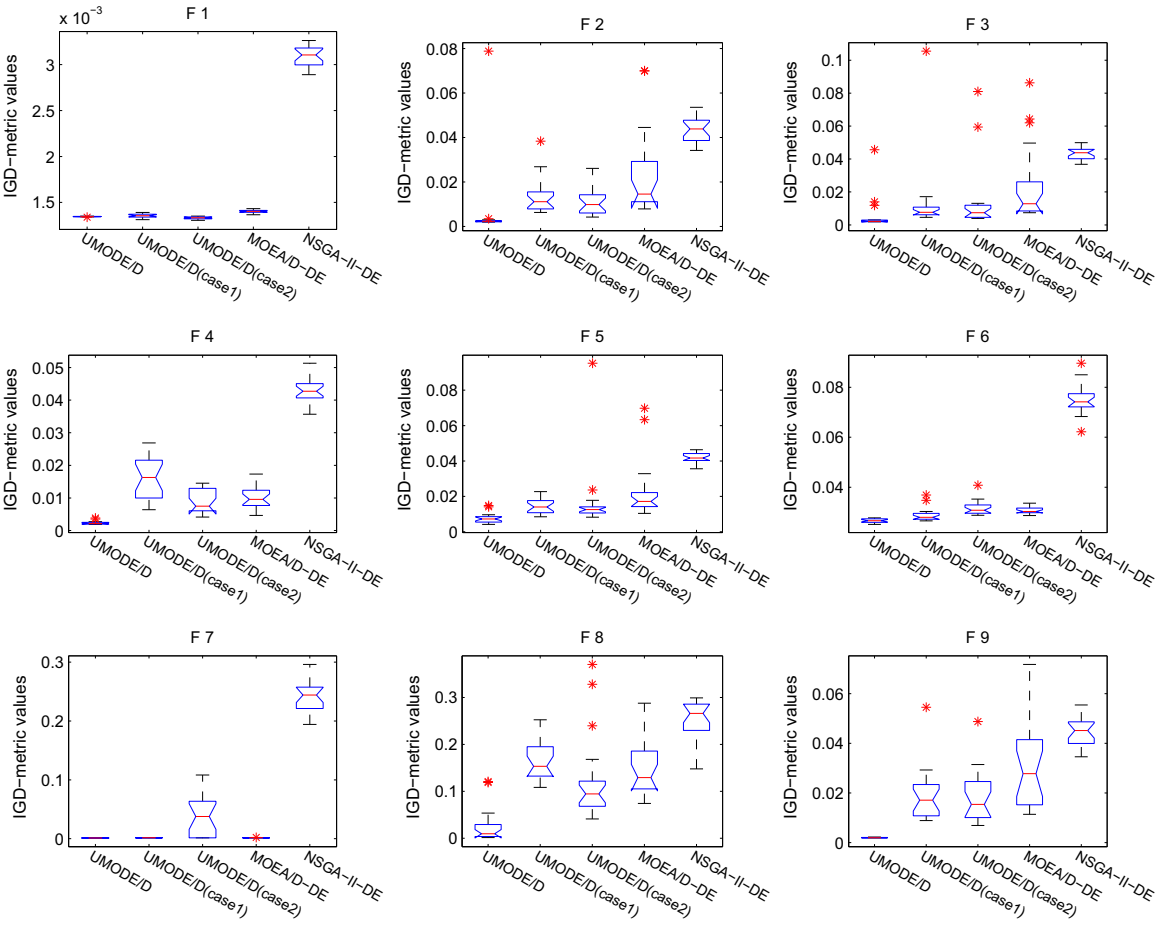

Fig. 4. Box plots of the IGD-metric values based on 20 independent runs for F1–F9.

Table 3
IGD-metric values of the nondominated solutions found by UMODE/D, MOEA/D-DE and NSGA-II-DE on F1–F9.

Problem	UMODE/D			MOEA/D-DE			NSGA-II-DE		
	Mean	Min	Std.	Mean	Min	Std.	Mean	Min	Std.
F1	0.0013	0.0013	0	0.0014	0.0014	0	0.0031	0.0029	0.0001
F2	0.0062	0.0020	0.0171	0.0234	0.0079	0.0189	0.0437	0.0341	0.0055
F3	0.0054	0.0018	0.0100	0.0239	0.0074	0.0230	0.0434	0.0368	0.0037
F4	0.0024	0.0019	0.0005	0.0099	0.0047	0.0034	0.0431	0.0357	0.0042
F5	0.0076	0.0042	0.0029	0.0225	0.0104	0.0161	0.0416	0.0356	0.0029
F6	0.0266	0.0251	0.0008	0.0307	0.0287	0.0013	0.0750	0.0622	0.0061
F7	0.0014	0.0013	0	0.0017	0.0015	0.0002	0.2407	0.1942	0.0258
F8	0.0241	0.0014	0.0354	0.1482	0.0741	0.0575	0.2569	0.1479	0.0397
F9	0.0020	0.0019	0.0001	0.0314	0.0114	0.0181	0.0446	0.0346	0.0060

else

$$x_j^{i'} = \frac{1}{2}\frac{\left[\left(x_j^{i_b}\right)^2 - \left(x_j^{i_c}\right)^2\right]g_{i_a}^{te} + \left[\left(x_j^{i_c}\right)^2 - \left(x_j^{i_a}\right)^2\right]g_{i_b}^{te} + \left[\left(x_j^{i_a}\right)^2 - \left(x_j^{i_b}\right)^2\right]g_{i_c}^{te}}{\left(x_j^{i_b} - x_j^{i_c}\right)g_{i_a}^{te} + \left(x_j^{i_c} - x_j^{i_a}\right)g_{i_b}^{te} + \left(x_j^{i_a} - x_j^{i_b}\right)g_{i_c}^{te}},$$

$j = 1, \ldots, n$, where ϵ is a suitable small positive value (e.g., 10^{-6}). If

$$g^{te}\left(x^{i'}\middle|\lambda^i, z\right) < g^{te}\left(x^i\middle|\lambda^i, z\right),$$

Table 4
UMODE/D (250 generations) versus MOEA/D-DE and NSGA-II-DE (500 generations).

Problem	UMODE/D			MOEA/D-DE			NSGA-II-DE		
	Mean	Min	Std.	Mean	Min	Std.	Mean	Min	Std.
F1	0.0013	0.0013	0	0.0015	0.0015	0	0.0044	0.0044	0
F2	0.0062	0.0020	0.0171	0.0028	0.0023	0.0004	0.0349	0.0203	0.0066
F3	0.0054	0.0018	0.0100	0.0068	0.0022	0.0099	0.0296	0.0228	0.0030
F4	0.0024	0.0019	0.0005	0.0040	0.0025	0.0014	0.0288	0.0251	0.0021
F5	0.0076	0.0042	0.0029	0.0127	0.0073	0.0069	0.0288	0.0244	0.0031
F6	0.0266	0.0251	0.0008	0.0289	0.0276	0.0014	0.0680	0.0522	0.0072
F7	0.0014	0.0013	0	0.0049	0.0015	0.0063	0.1171	0.0270	0.0716
F8	0.0241	0.0014	0.0354	0.0998	0.0487	0.0429	0.1981	0.1191	0.0494
F9	0.0020	0.0019	0.0001	0.0035	0.0025	0.0008	0.0395	0.0303	0.0061

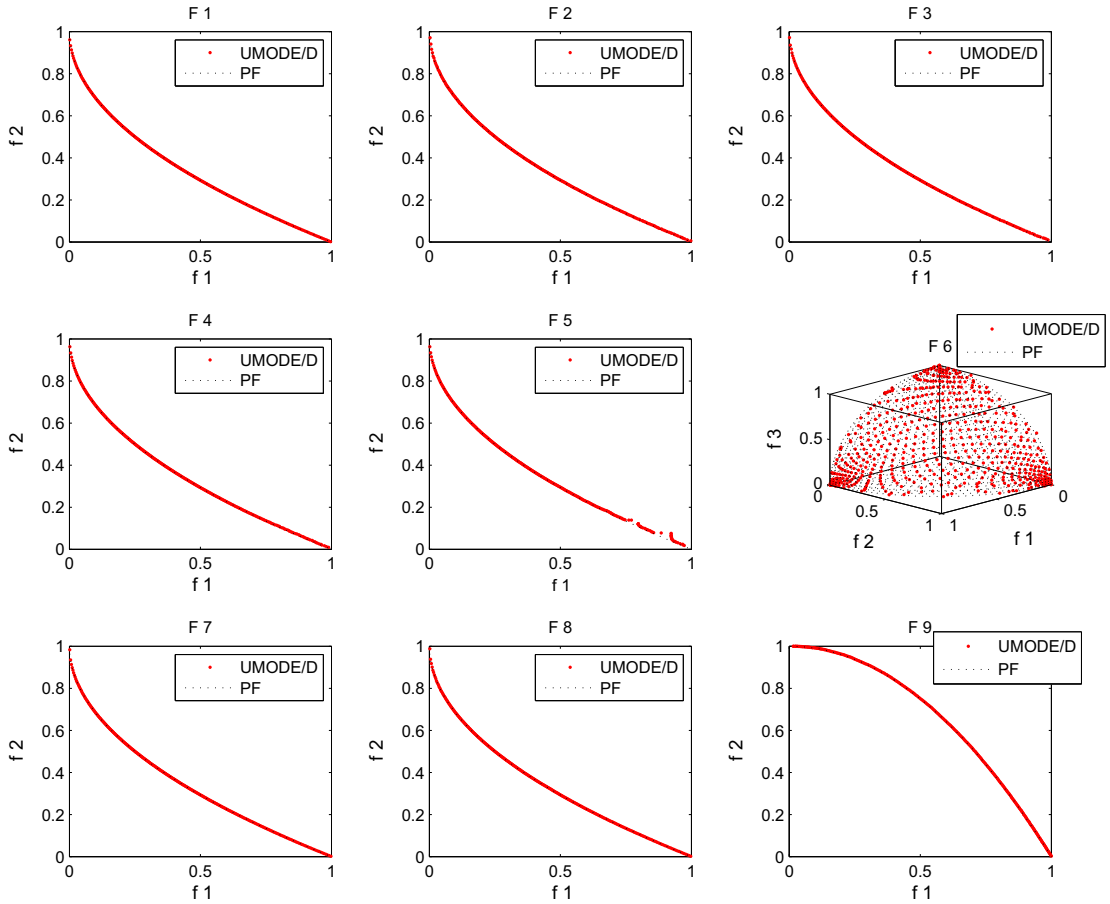

Fig. 5. Plots of the final populations with the lowest IGD-metric values found by UMODE/D in 20 runs in the objective space for F1–F9.

then set

$$x^i = x^{i'}.$$

Step 3 Stopping Criteria: If stopping criteria are satisfied, then stop and output $\{x^1,\ldots,x^N\}$ and $\{F(x^1),\ldots,F(x^N)\}$. Otherwise, go to Step 2.

In the DE operator used in Step 2.2, the variant DE/best/1 is used:

$$\text{DE/best/1}: v_k = x_k^{r_b} + F \times \left(x_k^{r_1} - x_k^{r_2}\right). \tag{8}$$

Each element $\bar{y}_k$ in $\bar{y} = (\bar{y}_1,\ldots,\bar{y}_n)$ is generated as follows:

$$\bar{y}_k = \begin{cases} v_k & \text{with probability } CR \\ x_k^{r_b} & \text{with probability } 1 - CR \end{cases}, \tag{9}$$

where CR and F are control parameters.

UMODE/D maintains two extra measures of MOEA/D-DE for improving the population diversity: (1) Allowing two parent solutions to be selected from the whole population with a low probability $1-\delta$. In such a way, a very wide range of offspring solutions could be generated due to the dissimilarity among these parent solutions. (2) The maximal number of solutions replaced by a offspring solution is bounded by η_r, which should be set to be much smaller than T in the implementation. Therefore, there is little chance that a solution has many copies in the population.

4. Experimental study

In this section, UMODE/D is experimented on a wide set of test problems with complicated PS shapes given in [20], on which MOEA/D-DE and NSGA-II-DE have been tested [20]. Its comparisons with MOEA/D-DE and NSGA-II-DE for solving

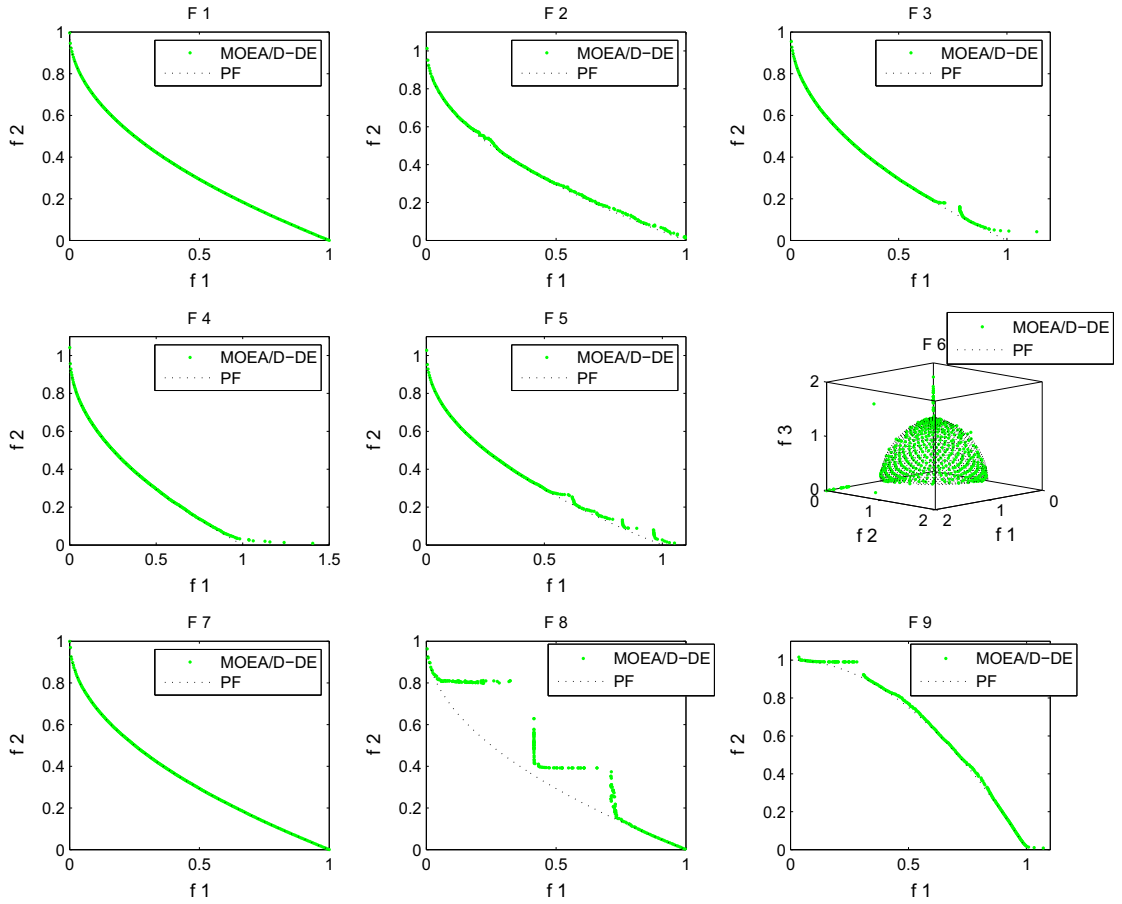

Fig. 6. Plots of the final populations with the lowest IGD-metric values found by MOEA/D-DE in 20 runs in the objective space for F1–F9.

these problems are then given. Two sets of experiments on these test problems are carried out to illustrate the efficiency of the uniform design method and the simplified quadratic approximation separately. NSGA-II-DE used in this paper is the same as NSGA-II-SBX in [9] except that it replaces the SBX operator in NSGA-II-SBX by the DE operator. For the detailed descriptions of MOEA/D-DE and NSGA-II-DE, refer to [20].

UMODE/D is also tested on CEC 2009 problems in this section for comparing with a further improved MOEA/D [35] in the CEC 2009 competition. In addition, UMODE/D is applied to slove three instances of the knapsack problem. Comparisons with NSGA-II [9], SPEA2 [40] and PESA [6] on these knapsack problems are also given.

4.1. Test instances

Nine test problems (i.e., F1–F9 [20]) are used in our experimental studies. Detailed descriptions of them are provided in the Appendix at the end of the paper. Constructed by Zhang and Li, F1–F9 are a series of continuous functions with complicated PS shapes. It has been proved that these complicated PSs, as well as the geometrical shapes of the PF, could affect the performance of MOEAs [16,10].

F1–F5 are 2-objective instances. They have the same convex PF shape but their PS shapes are various nonlinear curves in the decision spaces. F6 has three objectives and its PS is a nonlinear 2-D surface. F7 and F8 have identical PF shape with F1. Their PS shapes are similar to that of F1. However, there are many local Pareto solutions in F7 and F8, thus they are difficult to be optimized during evolution. Different from F1–F8 with convex PFs, F9 has a concave PF and its PS is identical with that of F2. All these test instances are minimization of the objectives.

Thirteen unconstrained CEC 2009 problems UF1–UF13 are also tested in this experimental study. Among these 13 problems, the first 7 are 2-objective problems and the next 3 are 3-objective problems while the last 3 are 5-objective problems. The numbers of decision variables are 30 for UF1–UF13. All these 13 instances are minimization of the objectives. For the detailed description of these test problems, refer to [36].

In addition, as combinatorial optimization problems, three instances of the knapsack problem taken from [41] are tested in the experiments, each with 750 items and 2, 3, and 4 objectives, respectively. For the random choice of the profit and weight values as well as the constraint handling technique, refer to the original study.

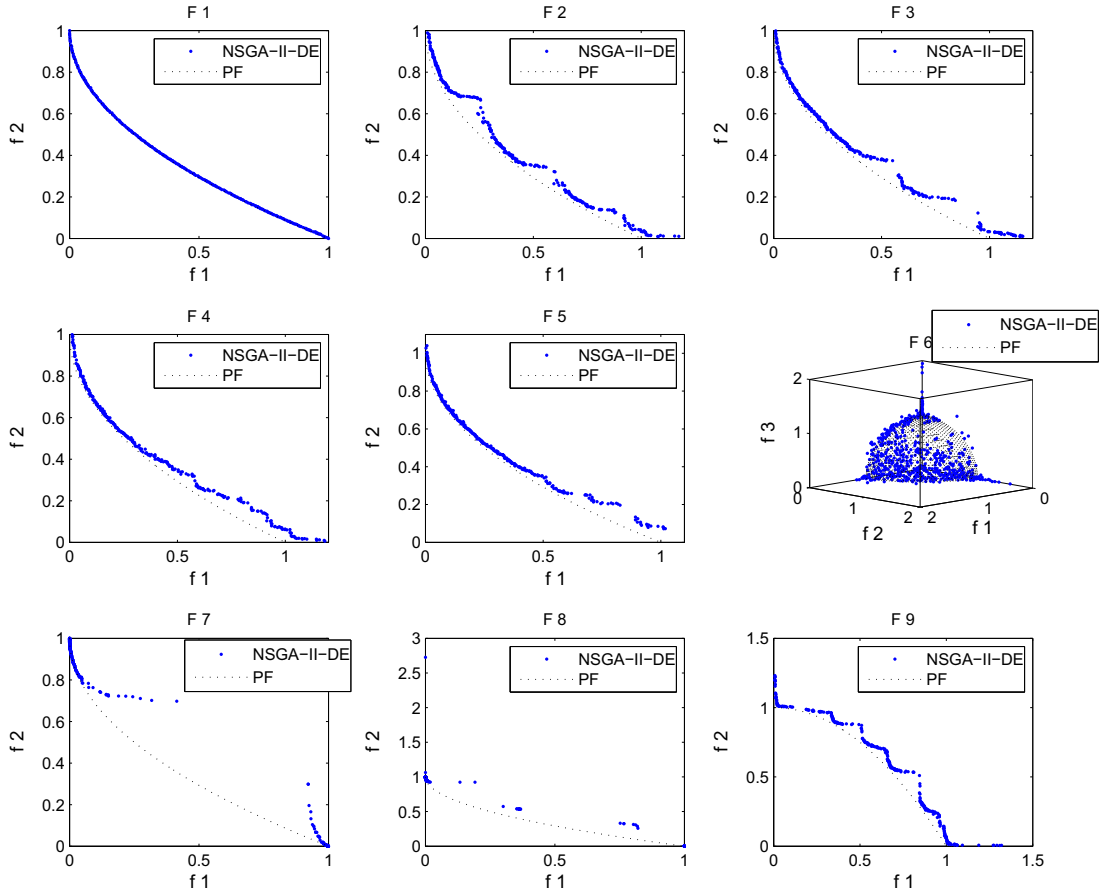

Fig. 7. Plots of the final populations with the lowest IGD-metric values found by NSGA-II-DE in 20 runs in the objective space for F1–F9.

4.2. Parameter setting

Compared algorithms are of the same reproduction operators for each test problem in the experiments for fair comparisons. For all continuous problems, the individuals are coded as real vectors. DE and polynomial mutation operators are applied directly to real parameter values in compared algorithms for continuous problems. For combinatorial multiobjective knapsack problems, the individuals are represented as bit strings, where each bit corresponds to one decision variable. Single-point crossover and bitwise mutation are adopted in UMODE/D, NSGA-II, SPEA2 and PESA for knapsack problems. The parameter settings in this paper are as follows:

(1) Control parameters in genetic operators:
- $CR = 1.0$ and $F = 0.5$ in the DE operator;
- $\eta_m = 20$ and $p_m = 1/n$ in the mutation operator.

(2) Population size and weight vectors in the algorithms:
- For F1–F9, population size N and weight vectors in MOEA/D-DE are controlled by an integer H. More precisely, $\lambda^1, \ldots, \lambda^N$ are all the weight vectors in which each individual weight takes a value from $\{0/H, 1/H, \ldots, H/H\}$. Therefore, the population size (i.e., the number of weight vectors) is $N = C_{H+m-1}^{m-1}$, where m is the number of objectives. H is set to be 299 and 33 for the 2-objective and 3-objective test instances respectively. Consequently, the population size N is 300 for the 2-objective instances and 595 for the 3-objective instances in all three algorithms for fair comparisons. However, different from MOEA/D-DE, the values of the weight vectors in UMODE/D are obtained by running the UD program;
- For fair comparisons with the further improved MOEA/D [35], about population size setting on UF1–UF13, refer to [35]: 600 for 2-objective, 1000 for 3-objective, and 1500 for 5-objective problems. Different from the further improved MOEA/D, the values of the weight vectors in UMODE/D are obtained by running the UD program;
- For the knapsack problems, population size N is set to 250 for $m = 2$, 300 for $m = 3$, and 350 for $m = 4$ in all four algorithms (UMODE/D, NSGA-II, SPEA2 and PESA).

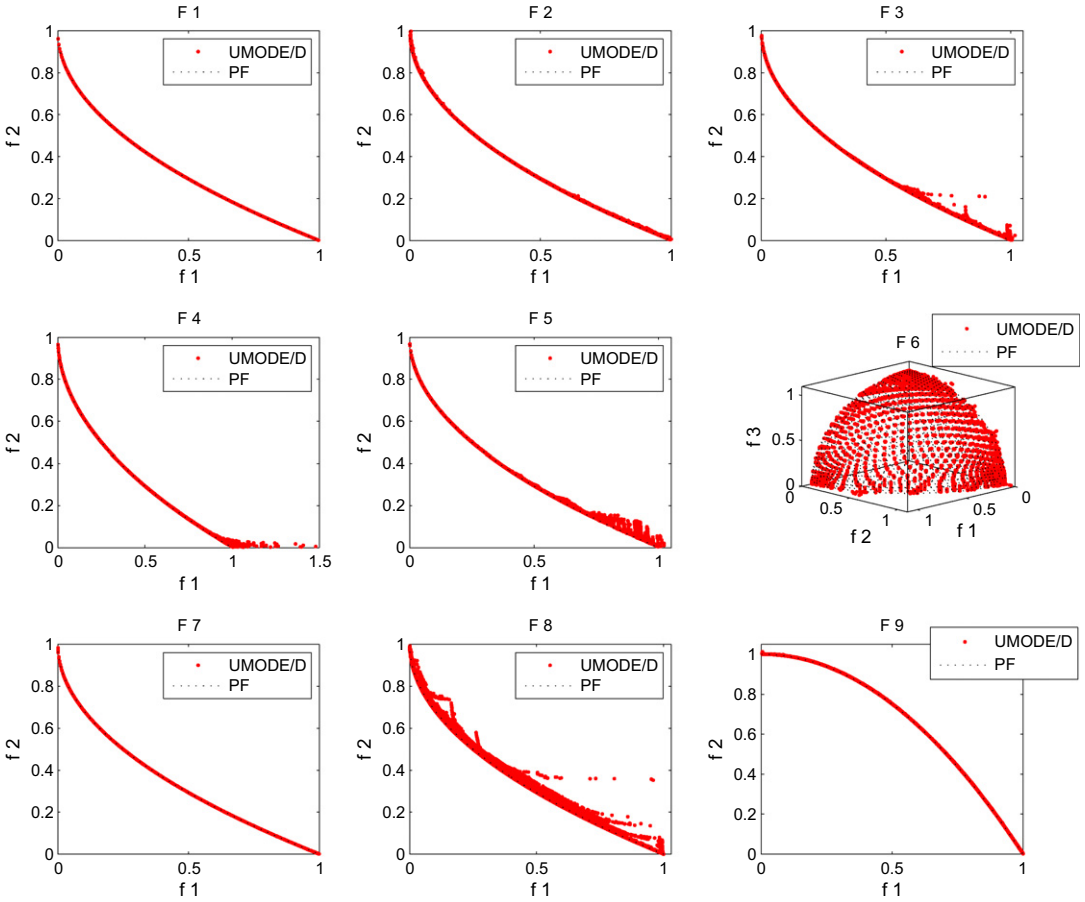

Fig. 8. Plots of all the 20 final populations produced by UMODE/D in the objective space for F1–F9.

(3) Number of runs and stopping condition: Each compared algorithm is run 20 times independently for F1–F9, and 30 times independently for UF1–UF13 and the knapsack problems. All these algorithms stop after a given number of function evaluations. The maximal number of function evaluations is set to $250 \times N$ for F1–F9, $500 \times N$ for the knapsack problems, and 300,000 for UF1–UF13.

(4) Other control parameters in UMODE/D and MOEA/D-DE:

- $T = 20$ for F1–F9, $T = 0.1N$ for UF1–UF13 and $T = 10$ for the knapsack problems;
- $\delta = 0.7$;
- $\eta_r = 2$ for F1–F9 and the knapsack problems, and $\eta_r = 0.01N$ for UF1–UF13;
- $\epsilon = 10^{-6}$.

4.3. Performance metric

In order to compare the performance of the different algorithms quantitatively, performance metrics are needed [11,41,42]. The inverted generational distance (IGD) [42] is adopted in MOEA/D, MOEA/D-DE and the further improved MOEA/D for assessing the performance of the algorithms [20,34,35]. For fair comparisons, we also apply the IGD-metric in our experimental studies. In addition, two other metrics, C-metric [41] and hypervolume metric [41], for the knapsack problems are included in the experiments. The performance metrics used in this paper are described as follows:

- **Inverted generational distance (IGD)** [42]: Let P^* be a set of uniformly distributed points in the objective space along the PF, and P be an approximation to the PF. The inverted generational distance from P^* to P is defined as:

$$\mathrm{IGD}(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \tag{10}$$

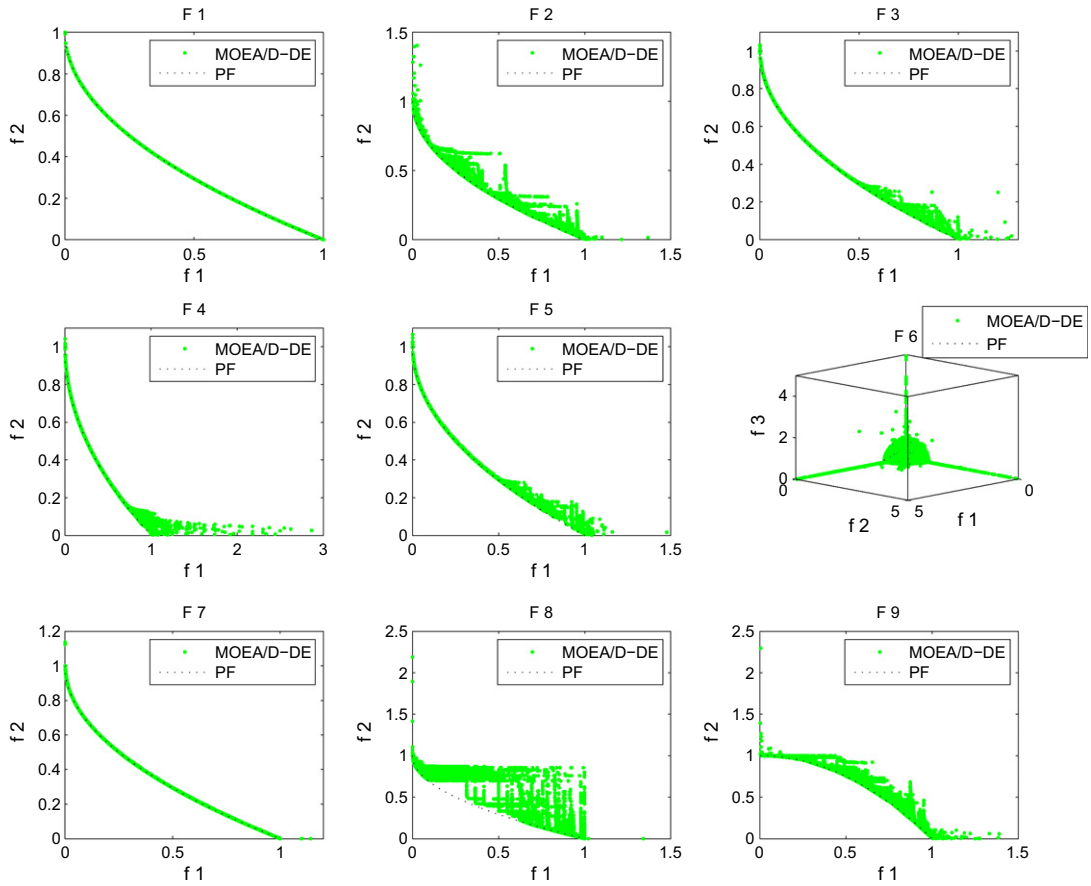

Fig. 9. Plots of all the 20 final populations produced by MOEA/D-DE in the objective space for F1–F9.

where $d(v,P)$ is the minimum Euclidean distance between v and the points in P. If $|P^*|$ is large enough to represent the PF very well, IGD(P^*,P) could measure both the diversity and the convergence of P in a sense. A lower value of IGD(P^*,P) implies that P must be very close to the PF and cannot miss any part of the whole PF.

- **Coverage of two sets (C):** The C-metric [41] measures the 'degree' of dominance of a Pareto front over another Pareto front. Let A and B be two approximations to the Pareto front of an MOP. C (A,B) is defined as the percentage of the solutions in B that are dominated by at least one solution in A, i.e.,

$$C(A,B) = \frac{|\{u \in B | \exists v \in A : v \text{ dominates } u\}|}{|B|} \tag{11}$$

C (A,B) is not necessarily equal to 1-C (B,A). C $(A,B) = 1$ means that all solutions in B are dominated by some solutions in A, while C $(A,B) = 0$ implies that no solution in B is dominated by a solution in A.

- **Hypervolume (HV)** [41]: The hypervolume metric measures the size of the region, which is dominated by the obtained Pareto front. Therefore, the higher value of the HV-metric is preferred. In lower dimensions with 2 and 3 objectives, the hypervolume $vol(v)$ is known as area and volume respectively. Mathematically, the HV-metric is described as:

$$HV(P) = \left\{ \bigcup vol(v) | v \in P \right\} \tag{12}$$

4.4. Experimental results and analysis

4.4.1. Comparison of UMODE/D with MOEA/D-DE and NSGA-II-DE for F1–F9

As is shown in Section 3, the proposed algorithm has been derived from two modifications to MOEA/D-DE: (1) Constructing the aggregation coefficient vectors by the UD method and (2) employing a simplified quadratic approximation with three best points as a local search operator for each subproblem. Here, two sets of experiments are carried out to illustrate the

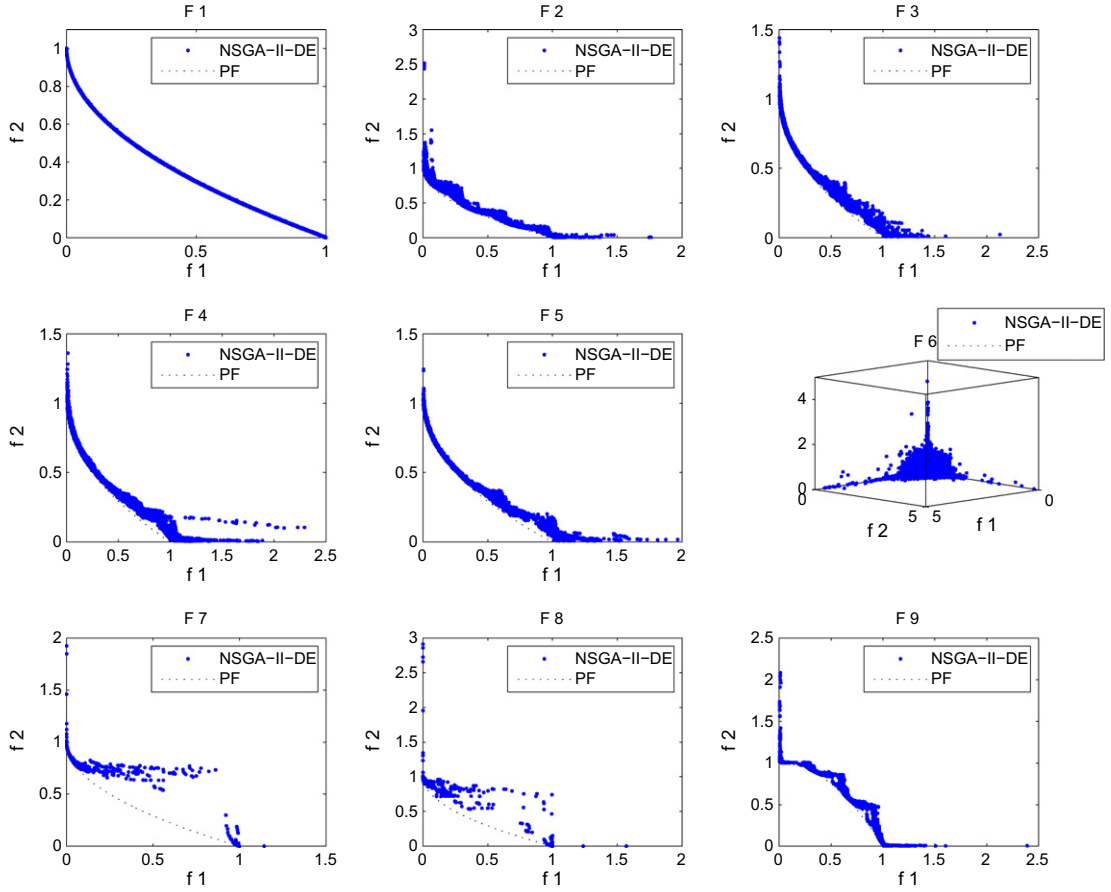

Fig. 10. Plots of all the 20 final populations produced by NSGA-II-DE in the objective space for F1–F9.

efficiency of these two modifications separately. For convenience, UMODE/D (case 1) stands for MOEA/D-DE only with the UD method, and UMODE/D (case 2) stands for MOEA/D-DE only with the local search.

In order to illustrate the efficiency of the UD method, we execute UMODE/D (case 1) and compare the results with those obtained by MOEA/D-DE for F1–F9. The comparison results are shown in Table 1. Fig. 2 presents the evolution of the average IGD-metric values of the population with the number of generations in four algorithms (UMODE/D, UMODE/D (case 1), MOEA/D-DE and NSGA-II-DE). From Table 1, we can see that in terms of IGD-metric UMODE/D (case 1) performs better than MOEA/D-DE in general. In 5 cases the mean IGD-metric value obtained by UMODE/D (case 1) is smaller than that obtained by MOEA/D-DE, and in 2 cases the mean IGD-metric value obtained by UMODE/D (case 1) is equal to that obtained by MOEA/D-DE. However, in 2 cases the mean IGD-metric value obtained by UMODE/D (case 1) is a little bit larger than that obtained by MOEA/D-DE. Fig. 2 visually shows that, UMODE/D (case 1) converges, in terms of the generations, similarly or faster than MOEA/D-DE in minimizing the IGD-metric values for all the problems except F4 and F8. These results indicate that in most cases the UD method is favorable for reducing the IGD-metric value.

In order to illustrate the efficiency of the simplified quadratic approximation, we execute UMODE/D (case 2) and compare the results with those obtained by MOEA/D-DE for F1–F9. The comparison results are shown in Table 2. Fig. 3 presents the evolution of the average IGD-metric values of the population with the number of generations in four algorithms (UMODE/D, UMODE/D (case 2), MOEA/D-DE and NSGA-II-DE). The box plots of the IGD-metric values based on 20 independent runs among five algorithms (UMODE/D, UMODE/D (case 1), UMODE/D (case 2), MOEA/D-DE and NSGA-II-DE) are visualized in Fig. 4. It can be observed from Table 2 and Fig. 4 that in terms of IGD-metric UMODE/D (case 2) performs better than MOEA/D-DE on all the test problems except F6 and F7. Fig. 3 illustrates that UMODE/D (case 2) converges, in terms of the generations, similarly or faster than MOEA/D-DE in minimizing the IGD-metric values for most of the test problems. These results indicate that in most cases the simplified quadratic approximation is effective in achieving better accuracy of the minimum scalar aggregation function value.

For all the test problems presented in the Appendix, comparisons of UMODE/D with MOEA/D-DE and NSGA-II-DE in Tables 3,4 and Figs. 2–16 are made. Table 3 presents the minimum, mean, and standard deviation of the IGD-metric values of the 20 final populations. Literature [20] gave the minimum, mean, and standard deviation of the IGD-metric values of MOEA/D-DE and NSGA-II-DE based on 20 independent runs under 500 generations. For comparison, we present them and the results of UMODE/D under 250 generations together in Table 4. Since no single metric is always able to rank different MOEAs appropriately, Figs. 5–7 and 11–13 plot, in both the objective space and decision space, the distribution of the final

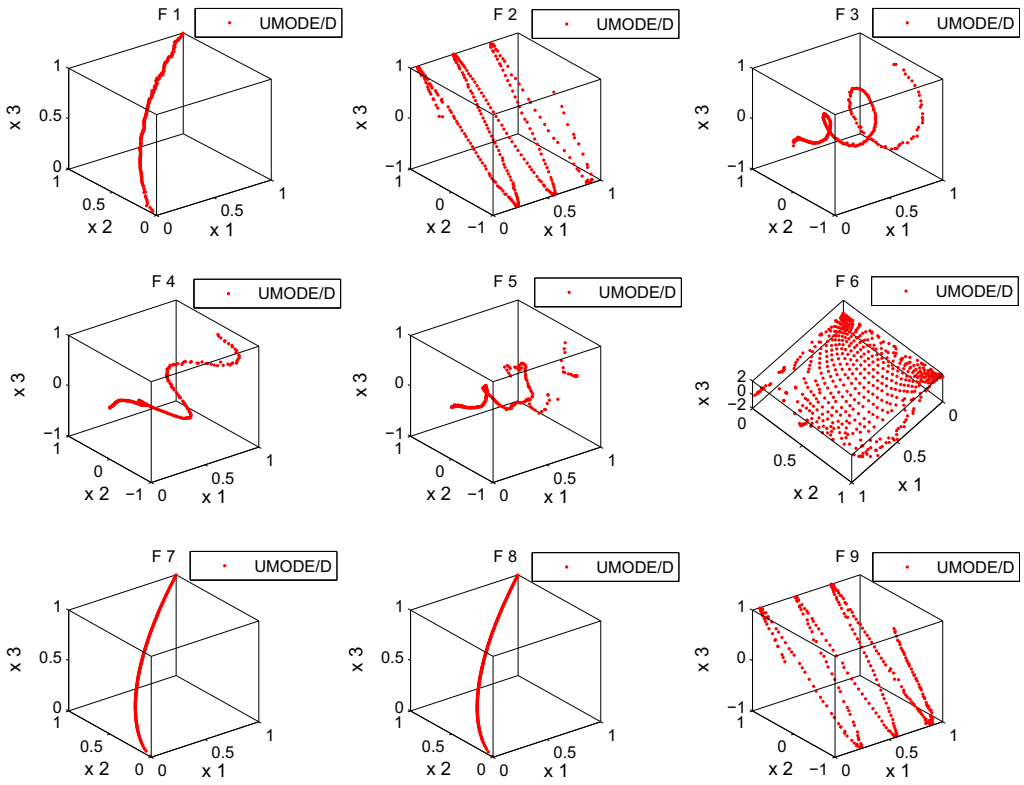

Fig. 11. Plots of the final populations with the lowest IGD-metric values found by UMODE/D in 20 runs in $x1$–$x2$–$x3$ space for F1–F9.

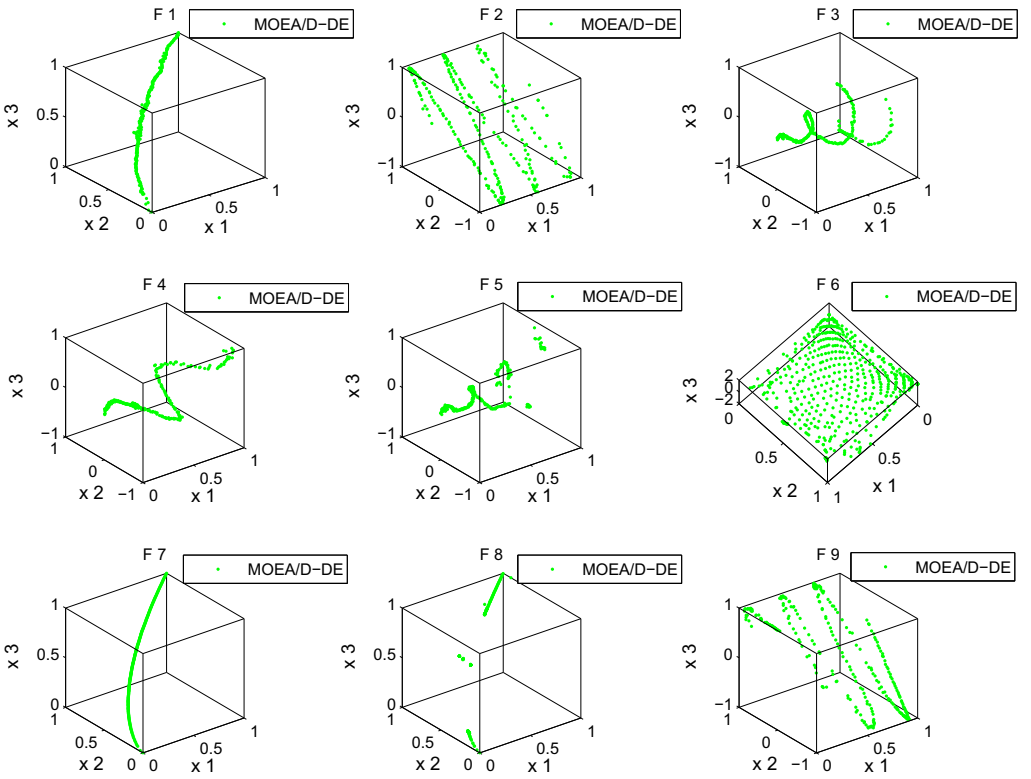

Fig. 12. Plots of the final populations with the lowest IGD-metric values found by MOEA/D-DE in 20 runs in $x1$–$x2$–$x3$ space for F1–F9.

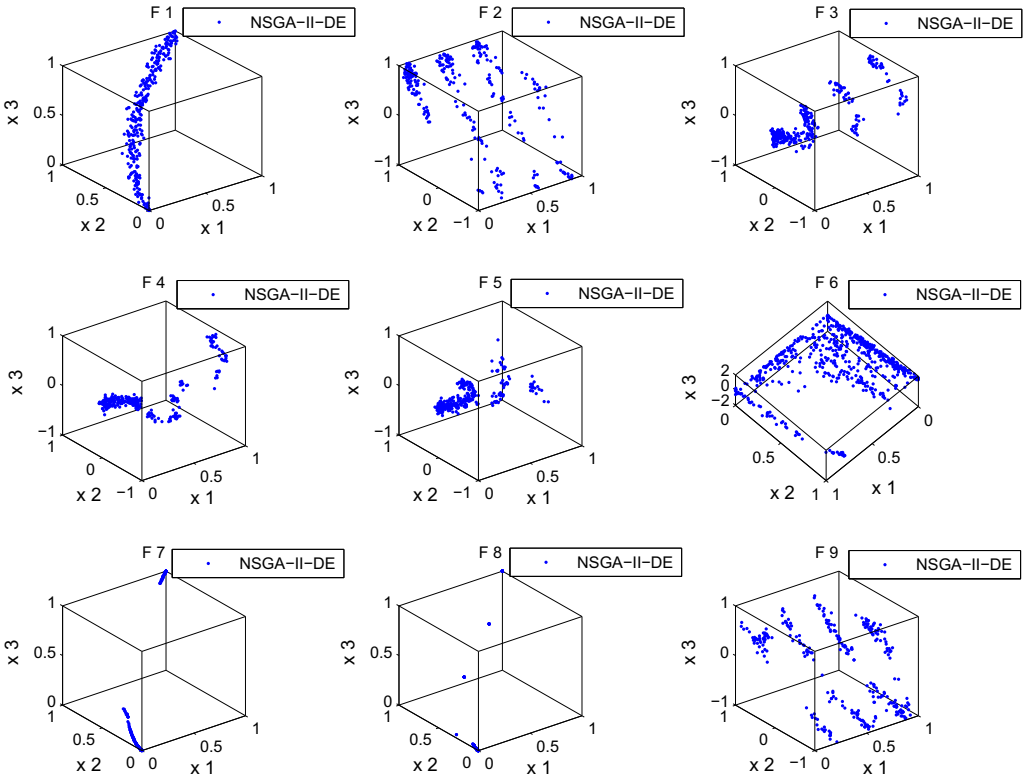

Fig. 13. Plots of the final populations with the lowest IGD-metric values found by NSGA-II-DE in 20 runs in $x1$–$x2$–$x3$ space for F1–F9.

populations with the lowest IGD-metric values obtained in 20 runs of each algorithm on each test instance. To show their distribution ranges, all the 20 final populations are also plotted together in Figs. 8–10 and 14–16.

Under the same number of generations, it is clear from Table 3 and Figs. 2–4 that, UMODE/D are much more effective and efficient than MOEA/D-DE and NSGA-II-DE in reducing the IGD-metric values for all the test problems. Figs. 5–16 visually show that the final populations in UMODE/D are significantly better than those in MOEA/D-DE and NSGA-II-DE in approximating both the PFs and PSs. NSGA-II-DE fails, within the given number of generations, in satisfactorily approximating the PFs and PSs on all the test instances except F1, which is relatively simple. MOEA/D-DE performs better than NSGA-II-DE. Generally speaking, the distributions of the final populations in MOEA/D-DE are not satisfactory. In contrast, UMODE/D is able to find a good approximation on all the test instances except F5.

When the number of generations in MOEA/D-DE and NSGA-II-DE is twice as many as that in UMODE/D, Table 4 reveals that UMODE/D still performs the best except for F2 on which MOEA/D-DE gets the smallest mean IGD-metric value, and UMODE/D can obtain the minimum value of IGD-metric for each test instance in 20 independent runs. Therefore, UMODE/D requires a smaller number of evolutional generations and yields relatively accurate solutions. These results indicate that in most cases the simplified quadratic approximation is effective in reducing the number of generations and in improving the accuracy of the minimum scalar aggregation function value.

In terms of IGD-metric, it can be observed from Table 3 and Figs. 2–4 that UMODE/D converges, in terms of the generations, much faster than MOEA/D-DE and NSGA-II-DE in minimizing the IGD-metric values for all the problems, especially for F4, F5, F8 and F9. MOEA/D-DE performs similarly to NSGA-II-DE for F2 and F9, and converges faster than NSGA-II-DE for the other problems. In addition, the box plots of the IGD-metric values based on 20 independent runs in Fig. 4 indicate that, UMODE/D behaves stably compared with MOEA/D-DE and NSGA-II-DE.

In terms of PFs and PSs of the final populations with the lowest IGD-metric values, Figs. 5–7 and 11–13 reveal that the final solutions obtained by UMODE/D have better spread and convergence than those obtained by MOEA/D-DE and NSGA-II-DE, especially for F7, F8 with many local Pareto solutions and F9 which has a concave PF. MOEA/D-DE cannot obtain representative Pareto solutions for F5, F6, F8, and F9 within the given number of generations. NSGA-II-DE performs the worst. 20 final populations are plotted together, Figs. 8–10 and 14–16 strengthen the conclusion that UMODE/D performs better than MOEA/D-DE and NSGA-II-DE on both the convergence and diversity for all the test instances. These results imply that the UD method and the simplified quadratic approximation are favorable in solving the MOPs.

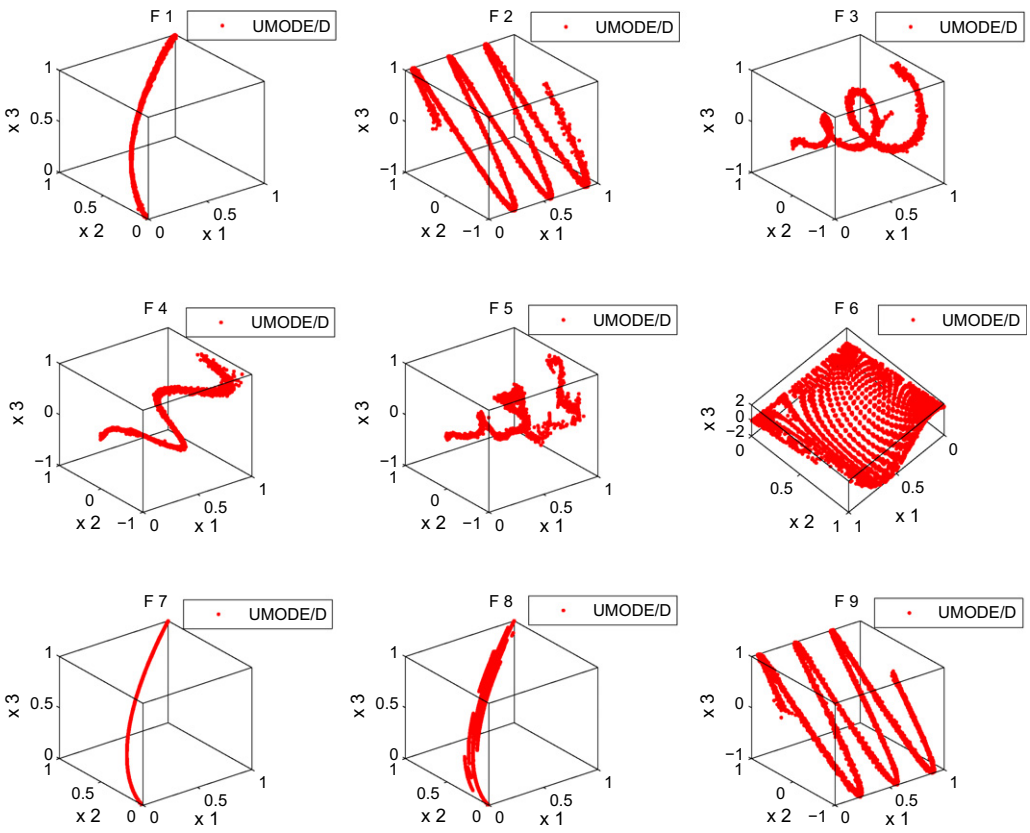

Fig. 14. Plots of all the 20 final populations found by UMODE/D in $x1$–$x2$–$x3$ space for F1–F9.

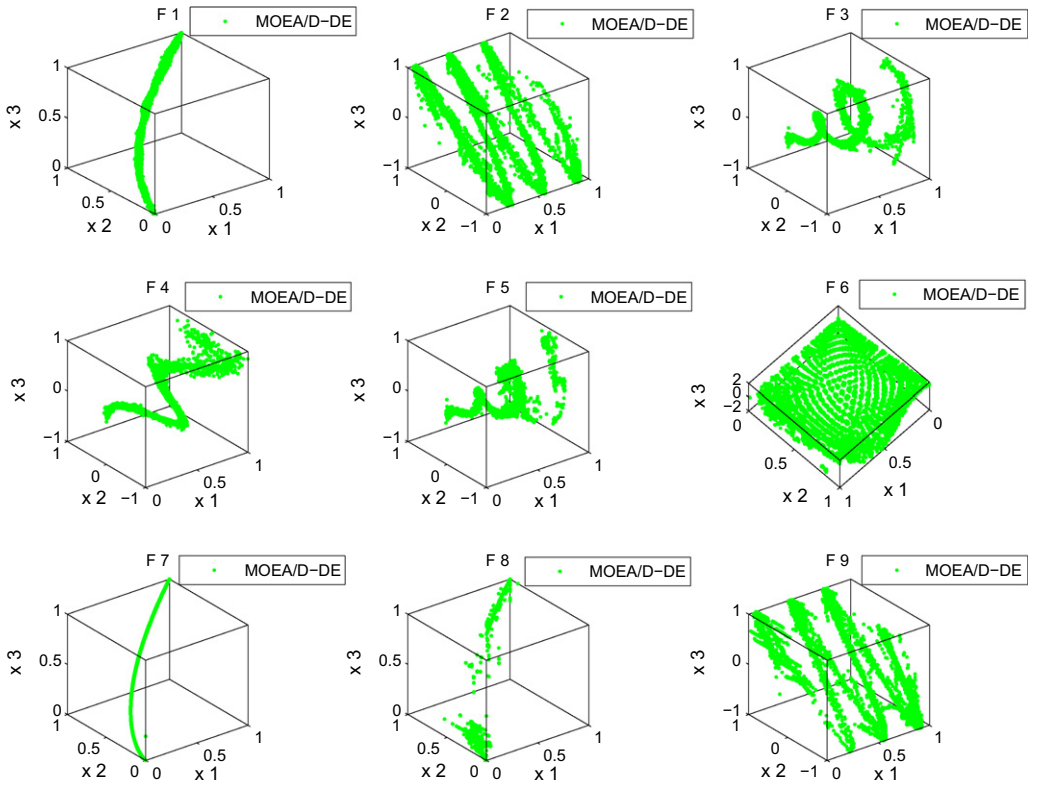

Fig. 15. Plots of all the 20 final populations found by MOEA/D-DE in $x1$–$x2$–$x3$ space for F1–F9.

F7 and F8 have identical PF shape with F1, and their PS shapes are similar to F1, but there are many local Pareto solutions in F7 and F8. It is clear from Figs. 5–16 that, NSGA-II-DE cannot approximate the PFs and PSs on F7 and F8 within the given number of generations, and MOEA/D-DE also performs unsatisfactorily on F8. The experimental results of MOEA/D-DE and NSGA-II-DE on F7 and F8, in conjunction with the results on F1, suggest that the presence of many local Pareto optimal solutions would deteriorate the performance of the algorithms. However, UMODE/D outperforms MOEA/D-DE and NSGA-II-DE on problems with nonlinear PSs and many local Pareto optimal solutions.

F9 has a concave PF and its PS is identical with that of F2. The different experimental results on F2 and F9 presented in Figs. 5–16 show that the performances of MOEA/D-DE and NSGA-II-DE have become slightly worse in the case of concave PF. In contrast, UMODE/D is less sensitive to the PF shape. These results demonstrate that a concave PF does hinder the performance of the MOEAs very much if the PS is nonlinear as in F9. On the other hand, these results imply that UMODE/D is robust for MOPs.

4.4.2. Comparison of UMODE/D with a further improved MOEA/D for UF1–UF13

The further improved MOEA/D [35] has been ranked first among 13 entries in the unconstrained MOEA competition in CEC 2009. To see how the proposed UMODE/D prevails its superiority over the further improved MOEA/D [35], a comprehensive numerical comparison is carried out on the CEC 2009 problems UF1–UF13. Under the same stopping condition, UMODE/D is compared with the further improved MOEA/D in terms of quality of the IGD-metric values.

All the parameter settings are the same as those mentioned in Section 4.2. The statistical results of the further improved MOEA/D and UMODE/D are summarized in Table 5, which include the best, mean, worst, and standard deviation (std) of the performance metric values for each benchmark problem based on 30 independent runs. The results of the further improved MOEA/D presented in the table are cited from [35]. As described in Table 5, for almost all the test problems, the mean, best, and worst IGD-metric values obtained by UMODE/D are smaller respectively than those obtained by the further improved MOEA/D, which indicates that UMODE/D outperforms MOEA/D in approximating the PF in terms of both the diversity and the convergence. From Table 5, it can be noted that the standard deviations of the performance metric found by UMODE/D are small except for UF12, and the results obtained by UMODE/D are not greater than MOEA/D for most of the test instances. These results imply that UMODE/D is robust and stable in solving this kind of MOPs.

Fig. 17 plots in the objective space, the distribution of the final population with the lowest IGD-metric value obtained in 30 runs on UF1–UF10. It is clear from the figure that, UMODE/D is able to find good approximations to UF1–UF3 and UF6–UF8. Its approximations to F5 and F9 are acceptable. However, it fails, within the given number of function evaluations, to

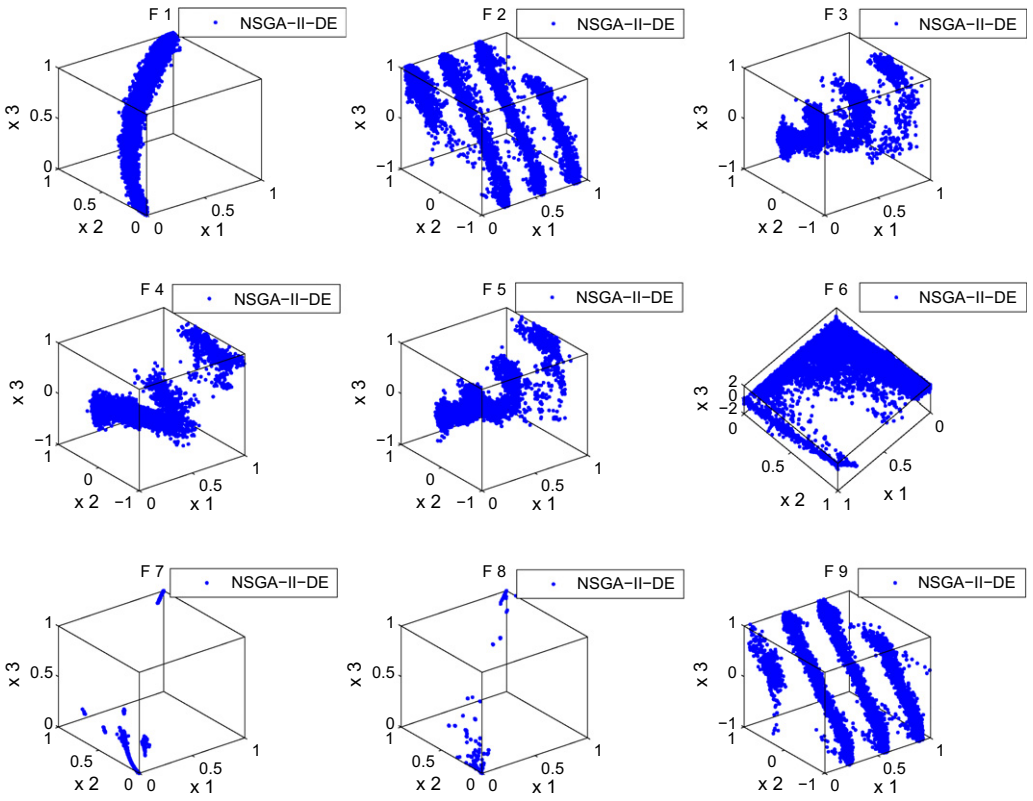

Fig. 16. Plots of all the 20 final populations found by NSGA-II-DE in $x1$–$x2$–$x3$ space for F1–F9.

Table 5
IGD statistics of UMODE/D and the further improved MOEA/D based on 30 independent runs.

Problem	UMODE/D				MOEA/D [35]			
	Mean	Std.	Best	Worst	Mean	Std.	Best	Worst
UF1	0.00421	0.00005	0.00411	0.00434	0.00435	0.00029	0.00399	0.00519
UF2	0.00475	0.00050	0.00428	0.00654	0.00679	0.00182	0.00481	0.01087
UF3	0.00420	0.00008	0.00409	0.00444	0.00742	0.00589	0.00394	0.02433
UF4	0.04266	0.00207	0.03898	0.04742	0.06385	0.00534	0.05687	0.08135
UF5	0.07254	0.09053	0.01077	0.46717	0.18071	0.06811	0.08028	0.30621
UF6	0.00258	0.00044	0.00185	0.00350	0.00587	0.00171	0.00342	0.01005
UF7	0.00408	0.00005	0.00401	0.00421	0.00444	0.00117	0.00405	0.01058
UF8	0.05476	0.00994	0.04568	0.10181	0.05840	0.00321	0.05071	0.06556
UF9	0.07263	0.05531	0.03025	0.14587	0.07896	0.05316	0.03504	0.14985
UF10	0.28450	0.15598	0.08231	0.74538	0.47415	0.07360	0.36405	0.64948
UF11	0.10429	0.01453	0.09843	0.18075	0.11032	0.00233	0.10692	0.11519
UF12	112.7791	28.2400	60.2714	170.2450	146.7813	41.8281	66.1690	214.2261
UF13	1.84875	0.00344	1.84186	1.85636	1.8489	0.0198	1.8346	1.8993

approximate satisfactorily the PFs of the problems UF4 and UF10. Nevertheless, as evidenced from Table 5, in general the UD and the simplified quadratic approximation can enhance the performance of MOEA/D.

4.4.3. Comparison of UMODE/D with some other algorithms for the Knapsack problems

In order to verify the performance of the proposed UMODE/D on combinatorial optimization problems, three instances of the knapsack problem taken from [41] are tested in the experiments, each with 750 items and 2, 3, and 4 objectives, respectively. For the random choice of the profit and weight values as well as the constraint handling technique, refer to the original study. Comparisons with NSGA-II, SPEA2 and PESA on these three knapsack problems are given in this subsection.

Table 6 shows the mean and standard deviation of the C-metric values of the final approximations obtained by UMODE/D and other three algorithms. Table 7 presents the mean of the HV-metric values obtained by four algorithms (UMODE/D, NSGA-II, SPEA2, PESA) for the knapsack problems. The box plots of the HV-metric values based on 30 independent runs

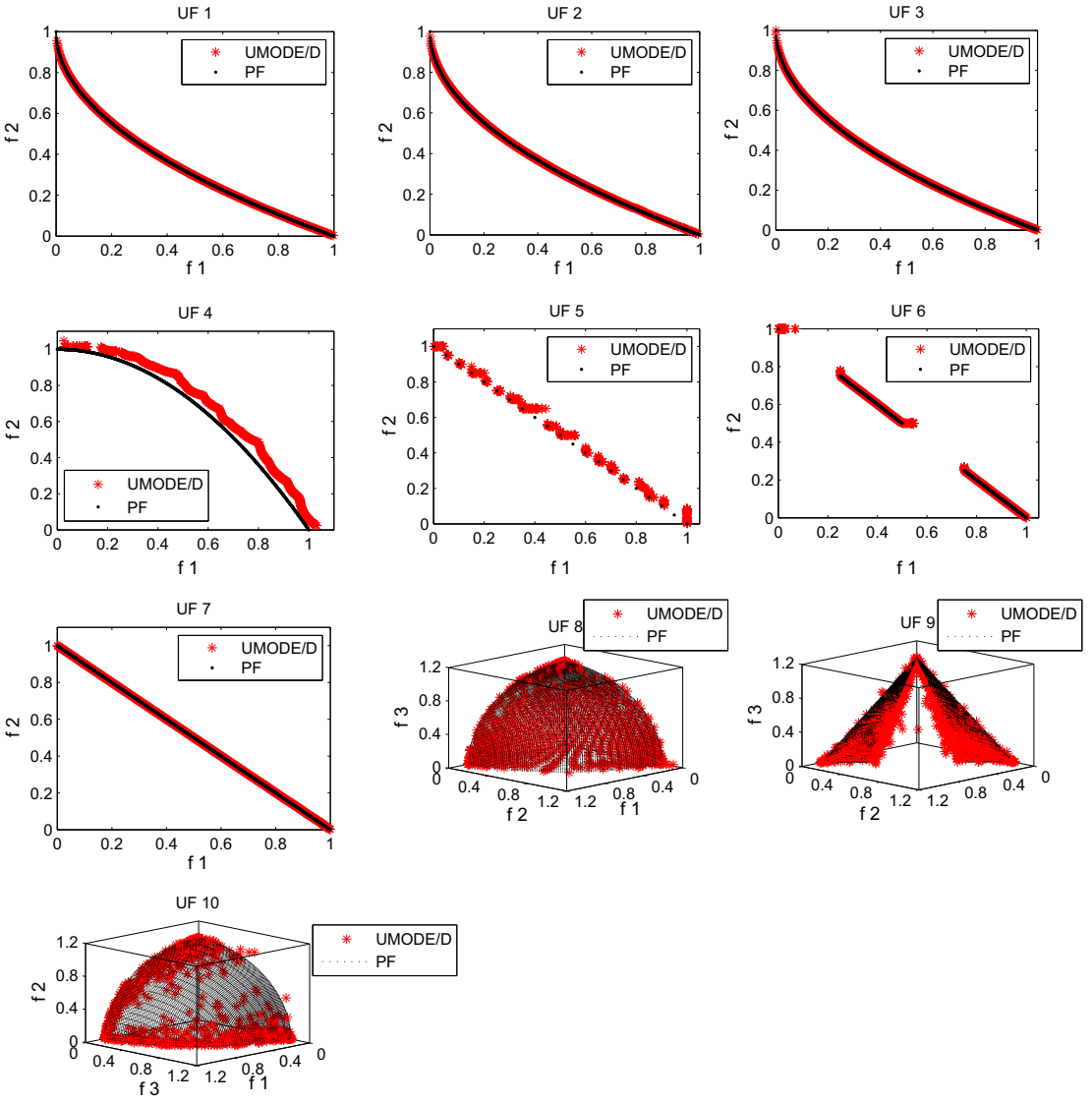

Fig. 17. The best approximation to UF1–UF10 obtained by UMODE/D.

Table 6
Average values (standard deviation) of C-metric based on 30 independent runs for the knapsack problems.

Algorithm		C (A,B)		
A	B	750-2	750-3	750-4
UMODE/D	NSGA-II	0.6507(0.2177)	0.9834(0.0349)	1(0)
	SPEA2	0.8577(0.1645)	0.9587(0.0534)	0.9984(0.0028)
	PESA	0.2756(0.2181)	0.6829(0.2687)	0.9422(0.0733)
NSGA-II	UMODE/D	0.0909(0.0632)	0.0002(0.0007)	0(0)
SPEA2		0.0524(0.0610)	0.0006(0.0010)	0(0)
PESA		0.1584(0.0637)	0.0079(0.0088)	0.0001(0.0002)

among these four algorithms are visualized in Fig. 18. Fig. 19 plots the distributions of the final approximation with the largest HV-metric value found by each algorithm for the 2-objective knapsack problem.

Table 6 shows that the final approximation obtained by UMODE/D is better than that obtained by NSGA-II, SPEA2 and PESA in terms of C-metric for all the knapsack problems. Taking instance 750-3 as an example, on average, 98.34%,

Table 7
Average values of HV-metric based on 30 independent runs for the knapsack problems.

Problem	UMODE/D	NSGA-II	SPEA2	PESA
750-2	8.83748×10^{8}	8.10993×10^{8}	8.19415×10^{8}	8.07935×10^{8}
750-3	2.64595×10^{13}	2.19519×10^{13}	2.20511×10^{13}	2.16483×10^{13}
750-4	6.82113×10^{17}	4.91646×10^{17}	4.97285×10^{17}	4.82413×10^{17}

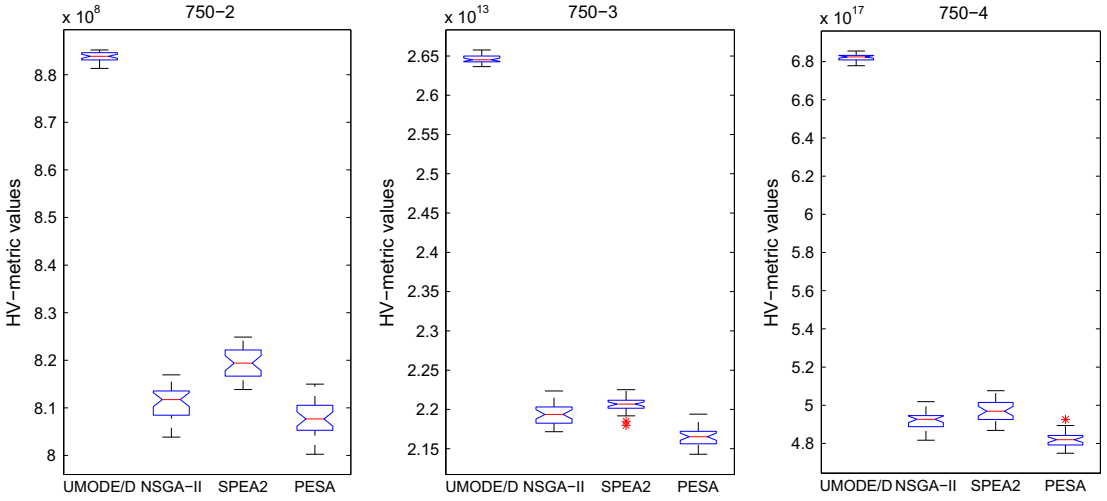

Fig. 18. Box plots of the HV-metric values based on 30 independent runs for the knapsack problems.

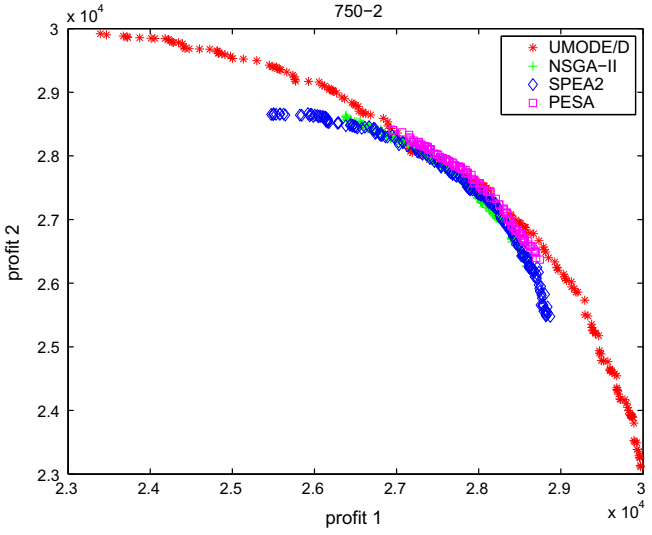

Fig. 19. Plots of the nondominated solutions with the largest HV-metric in 30 runs for the 2-objective knapsack problem.

95.87%, and 68.29% of the final solutions generated by NSGA-II, SPEA2, and PESA respectively are dominated by those generated by UMODE/D, and only 0.02%, 0.06%, and 0.79% vice versa respectively.

It is evident from Table 7 and Fig. 18 that the HV-metric values of UMODE/D are significantly larger than that of other three algorithms for each knapsack problem. These results indicate that UMODE/D dominates more search space than NSGA-II, SPEA2 and PESA. Fig. 19 visually shows that UMODE/D can find higher quality solutions than other three compared algorithms. Overall, we can claim that UMODE/D outperforms NSGA-II, SPEA2 and PESA on these knapsack problems.

5. Conclusion

MOEA/D proposed by Zhang and Li has been proved to be an effective and efficient algorithm. MOEA/D uses the simplex-lattice design method to set the weight values of the aggregation coefficient vectors. However, the distribution of the weight vectors is not very uniform and the population size increases nonlinearly with the number of objectives, which restricts MOEA/D's use to a certain extent in solving MOPs. To overcome these drawbacks and improve its performance, we propose UMODE/D. The algorithm is a modification to the new version of MOEA/D based on DE, i.e., MOEA/D-DE proposed by Hui Li and Qingfu Zhang for solving multiobjective problems with complicated PS shapes. As shown in Section 3, the proposed algorithm has been derived from two modifications to the MOEA/D-DE: (1) Constructing the aggregation coefficient vectors by the UD method and (2) employing a simplified quadratic approximation with three best points as a local search operator for each subproblem. In order to verify the performance of UMODE/D, we have compared it with the original one MOEA/D-DE and NSGA-II-DE for solving a wide set of MOPs with complicated PS shapes.

Two sets of experiments on these test problems have been carried out to illustrate the efficiency of the uniform design method and the simplified quadratic approximation separately. Comparison results show that UMODE/D significantly outperforms MOEA/D-DE and NSGA-II-DE on all these test problems, even the number of generations in MOEA/D-DE and NSGA-II-DE is twice as many as that in UMODE/D. In other words, UMODE/D requires a smaller number of generations and yields a more accurate Pareto set in many cases. The final solutions obtained by UMODE/D have better spread and convergence than those obtained by MOEA/D-DE and NSGA-II-DE, especially on problems with many local Pareto solutions or with concave PFs.

In addition, UMODE/D has been tested on CEC 2009 problems and combinatorial knapsack problems. Experimental results show that the proposed algorithm performs better than the further improved MOEA/D for almost all the CEC 2009 problems, and the results obtained are very competitive when comparing UMODE/D with some other algorithms on these multiobjective knapsack problems.

With the UD method, UMODE/D could explore the search space uniformly, enhance the diversity of the population, and reduce the chance of being trapped in local minima from the initial iteration. On the other hand, the simplified quadratic approximation with three best points could improve the accuracy of the minimum scalar aggregation function value and enhance the local search ability of the algorithm. The close integration of the global search part of the algorithm with the simplified quadratic approximation makes the proposed algorithm converge fast and adapt itself to the problem being solved.

In the future, we plan to study the ability of UMODE/D for dealing with many-objective problems. How to improve the efficiency of the algorithm remains to be studied further.

Acknowledgment

The authors gratefully thank Professor Qingfu Zhang in the University of Essex for his constructive suggestions on this work.

Appendix A. Test problems

Problem F 1

$$f_1 = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}\left(x_j - x_1^{0.5\left(1.0+\frac{3(j-2)}{n-2}\right)}\right)^2$$

$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}\left(x_j - x_1^{0.5\left(1.0+\frac{3(j-2)}{n-2}\right)}\right)^2$$

where

$$J_1 = \{j | j \text{ is odd and } 2 \leqslant j \leqslant n\},$$
$$J_2 = \{j | j \text{ is even and } 2 \leqslant j \leqslant n\}.$$

$x = (x_1, \cdots, x_n)^\top \in [0,1]^n$. Its PS is

$$x_j = x_1^{0.5\left(1.0+\frac{3(j-2)}{n-2}\right)}, \quad j = 2, \ldots, n.$$

We choose $n = 30$ in our experiments.

Problem F 2

$$f_1 = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}\left(x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2$$
$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}\left(x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2$$

where J_1 and J_2 are the same as those of F1, and $x = (x_1, \ldots, x_n)^\top \in [0,1] \times [-1,1]^{n-1}$. Its PS is

$$x_j = \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), \quad j = 2, \ldots, n.$$

We choose $n = 30$ in our experiments.

Problem F 3

$$f_1 = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}\left(x_j - 0.8x_1\cos\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2$$
$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}\left(x_j - 0.8x_1\sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2$$

where J_1 and J_2 are the same as those of F1, and $x = (x_1, \ldots, x_n)^\top \in [0,1] \times [-1,1]^{n-1}$. Its PS is

$$x_j = \begin{cases} 0.8x_1\cos\left(6\pi x_1 + \frac{j\pi}{n}\right) & j \in J_1 \\ 0.8x_1\sin\left(6\pi x_1 + \frac{j\pi}{n}\right) & j \in J_2 \end{cases}.$$

We choose $n = 30$ in our experiments.

Problem F 4

$$f_1 = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}\left(x_j - 0.8x_1\cos\left(\frac{6\pi x_1 + \frac{j\pi}{n}}{3}\right)\right)^2$$
$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}\left(x_j - 0.8x_1\sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2$$

where J_1 and J_2 are the same as those of F1, and $x = (x_1, \ldots, x_n)^\top \in [0,1] \times [-1,1]^{n-1}$. Its PS is

$$x_j = \begin{cases} 0.8x_1\cos\left(\frac{6\pi x_1 + \frac{j\pi}{n}}{3}\right) & j \in J_1 \\ 0.8x_1\sin\left(6\pi x_1 + \frac{j\pi}{n}\right) & j \in J_2 \end{cases}.$$

We choose $n = 30$ in our experiments.

Problem F 5

$$f_1 = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}\left\{x_j - \left[0.3x_1^2\cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right]\cos\left(6\pi x_1 + \frac{j\pi}{n}\right)\right\}^2$$
$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}\left\{x_j - \left[0.3x_1^2\cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right]\sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right\}^2$$

where J_1 and J_2 are the same as those of F1, and $x = (x_1, \ldots, x_n)^\top \in [0,1] \times [-1,1]^{n-1}$. Its PS is

$$x_j = \begin{cases} \left[0.3x_1^2\cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right]\cos\left(6\pi x_1 + \frac{j\pi}{n}\right) & j \in J_1 \\ \left[0.3x_1^2\cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right]\sin\left(6\pi x_1 + \frac{j\pi}{n}\right) & j \in J_2 \end{cases}.$$

We choose $n = 30$ in our experiments.

Problem F 6

$$f_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{|J_1|}\sum_{j\in J_1}\left(x_j - 2x_2\sin\left(2\pi x_1 + \frac{j\pi}{n}\right)\right)^2$$
$$f_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{|J_2|}\sum_{j\in J_2}\left(x_j - 2x_2\sin\left(2\pi x_1 + \frac{j\pi}{n}\right)\right)^2$$
$$f_3 = \sin(0.5x_1\pi) + \frac{2}{|J_3|}\sum_{j\in J_3}\left(x_j - 2x_2\sin\left(2\pi x_1 + \frac{j\pi}{n}\right)\right)^2$$

where

$$J_1 = \{j|_3 \leqslant j \leqslant n, \text{ and } j-1 \text{ is a multiplication of } 3\},$$
$$J_2 = \{j|_3 \leqslant j \leqslant n, \text{ and } j-2 \text{ is a multiplication of } 3\},$$
$$J_3 = \{j|_3 \leqslant j \leqslant n, \text{ and } j \text{ is a multiplication of } 3\}.$$

$x = (x_1,\ldots,x_n)^\top \in [0,1]^2 \times [-2,2]^{n-2}$. Its PS is

$$x_j = 2x_2\sin\left(2\pi x_1 + \frac{j\pi}{n}\right), \quad j = 3,\ldots,n.$$

We choose $n = 10$ in our experiments.

Problem F 7

$$f_1 = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}\left(4y_j^2 - \cos(8y_j\pi) + 1.0\right)$$
$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}\left(4y_j^2 - \cos(8y_j\pi) + 1.0\right)$$

where J_1 and J_2 are the same as those of F1 and $y_j = x_j - x_1^{0.5\left(1.0+\frac{3(j-2)}{n-2}\right)}, j = 2,\ldots,n$. $x = (x_1,\ldots,x_n)^\top \in [0,1]^n$. It has many local Pareto solutions, and its PS is

$$x_j = x_1^{0.5\left(1.0+\frac{3(j-2)}{n-2}\right)}, \quad j = 2,\ldots,n.$$

We choose $n = 10$ in our experiments.

Problem F 8

$$f_1 = x_1 + \frac{2}{|J_1|}\left(4\sum_{j\in J_1}y_j^2 - 2\prod_{j\in J_1}\cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$$
$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\left(4\sum_{j\in J_2}y_j^2 - 2\prod_{j\in J_2}\cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$$

where J_1 and J_2 are the same as those of F1 and $y_j = x_j - x_1^{0.5\left(1.0+\frac{3(j-2)}{n-2}\right)}, j = 2,\ldots,n$. $x = (x_1,\ldots,x_n)^\top \in [0,1]^n$. It has many local Pareto solutions, and its PS is

$$x_j = x_1^{0.5\left(1.0+\frac{3(j-2)}{n-2}\right)}, \quad j = 2,\ldots,n.$$

We choose $n = 10$ in our experiments.

Problem F 9

$$f_1 = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}\left(x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2$$
$$f_2 = 1 - x_1^2 + \frac{2}{|J_2|}\sum_{j\in J_2}\left(x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right)^2$$

where J_1 and J_2 are the same as those of F1, and $x = (x_1,\ldots,x_n)^\top \in [0,1]\times[-1,1]^{n-1}$. Its PS is

$$x_j = \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), \quad j = 2,\ldots,n.$$

We choose $n = 30$ in our experiments.

References

[1] P. Chakraborty, S. Das, G.G. Roy, A. Abraham, On convergence of the multiobjective particle swarm optimizers, Information Sciences 181 (8) (2011) 1411–1425.
[2] L.Y. Chan, M.L. Lo, Quality improvement in the manufacture of liquid crystal displays using uniform design, International Journal of Materials and Product Technology 20 (2004) 127–142.
[3] L.Y. Chan, Optimal designs for experiments with mixtures: a survey, Communications in Statistics: Theory and Methods 29 (2000) 2281–2312.
[4] Y. Chen, X. Zou, W. Xie, Convergence of multi-objective evolutionary algorithms to a uniformly distributed representation of the Pareto front, Information Sciences 181 (16) (2011) 3336–3355.
[5] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handing multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 256–279.
[6] D.W. Corne, J.D. Knowles, M.J. Oates, The Pareto-envelope based selection algorithm for multi-objective optimization, in: Parallel Problem Solving from Nature, PPSN VI, LNCS, Springer-Verlag, Berlin, 2000, pp. 869–878.
[7] J.A. Cornell, Experiments with Mixtures, Designs, Models and the Analysis of Mixture Data, Wiley, New York, 2002.
[8] J.A. Cornell, Experiments with mixtures: a review, Technometrics 15 (1973) 437–455.
[9] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
[10] K. Deb, A. Sinha, S. Kukkonen, Multi-objective test problems, linkages, and evolutionary methodologies, in: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation-GECCO'06, Seattle, WA, 2006, pp. 1141–1148.
[11] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (1) (2011) 3–18.
[12] K.T. Fang, The uniform design: application of number-theoretic methods in experimental design, Acta Math Appl Sinica 3 (1980) 363–372.
[13] K.T. Fang, Y. Wang, Number-Theoretic Methods in Statistics, Chapman and Hall, London, 1994.
[14] K.T. Fang, D.K.J. Lin, Uniform designs and their application in industry, Handbook of Statistics 22 (2003) 131–170.
[15] M.G. Gong, L.C. Jiao, H.F. Du, L.F. Bo, Multiobjective immune algorithm with nondominated neighbor-based selection, Evolutionary Computation 16 (2) (2008) 225–255.
[16] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Transactions on Evolutionary Computation 10 (5) (2006) 477–506.
[17] E.J. Hughes, Evolutionary many-objective optimisation: many once or one many?, in: IEEE Congress on Evolutionary Computation (CEC'05), Edinburgh, UK, 2–5 September 2005, pp. 222–227.
[18] Y.C. Jiao, C.Y. Dang, Y. Leung, Y. Hao, A modification to the new version of the Price's algorithm for continuous global optimization problems, Journal of Global Optimization 36 (4) (2006) 609–626.
[19] D. Kundu, K. Suresh, S. Ghosh, S. Das, B. Panigrahi, S. Das, Multiobjective optimization with artificial weed colonies, Information Sciences 181 (12) (2011) 2441–2454.
[20] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Transactions on Evolutionary Computation 12 (2) (2009) 284–302.
[21] R. Li, D.K.J. Lin, Y. Chen, Uniform design: design, analysis and applications, International Journal of Materials and Product Technology 20 (2004) 101–114.
[22] Y.Z. Liang, K.T. Fang, Q.S. Xu, Uniform design and its applications in chemistry and chemical engineering, Chemometrics and Intelligent Laboratory Systems 58 (1) (2001) 43–57.
[23] Efrén Mezura-Montes, Mariana Edith Miranda-Varela, Rubí del Carmen Gómez-Ramón, Differential evolution in constrained numerical optimization: An empirical study, Information Sciences 180 (22) (2010) 4223–4262.
[24] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer, Norwell, MA, 1999.
[25] B.Y. Qu, P.N. Suganthan, Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection, Information Sciences 180 (17) (2010) 3170–3181.
[26] K.S.N. Ripon, S. Kwong, K.F. Man, A real-coding jumping gene genetic algorithm (RJGGA) for multiobjective optimization, Information Sciences 177 (2) (2007) 632–654.
[27] N.J.W.T.J. Santers, W.I. Notz, The Design of Analysis of Computer Experiments, Springer-Verlag, Berlin, Germany, 2003.
[28] H. Scheffé, Experiments with mixtures, Journal of Royal Statistical Society Series B 20 (1958) 344–360.
[29] H. Scheffé, Simplex-centroid designs for experiments with mixtures, Journal of Royal Statistical Society Series B 25 (1963) 235–263.
[30] C. Shi, Z. Yan, K. Lü, Z. Shi, B. Wang, A dominance tree and its application in evolutionary multi-objective optimization, Information Sciences 179 (20) (2009) 3540–3560.
[31] H.K. Singh, T. Ray, W. Smith, C-PSA: constrained Pareto simulated annealing for constrained multi-objective optimization, Information Sciences 180 (13) (2010) 2499–2513.
[32] Q.S. Xu, Y.Z. Liang, K.T. Fang, The effects of different experimental designs on parameter estimation in the kinetics of a reversible chemical reaction, Chemometrics and Intelligent Laboratory Systems 52 (2000) 155–166.
[33] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm, IEEE Transactions on Evolutionary Computation 21 (1) (2008) 41–63.
[34] Q. Zhang, H. Li, MOEA/D: a multi-objective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.
[35] Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, Working Report CES-491, School of CS and EE, University of Essex, Colchester, UK, February 2009.
[36] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Mmultiobjective optimization test instances for the CEC 2009 sepcial session and competition, Technical Report CES-487, the School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK, 2008.
[37] S.Z. Zhao, P.N. Suganthan, Two-lbests based multi-objective particle swarm optimizer, Engineering Optimization 43 (1) (2011) 1–17.
[38] S.Z. Zhao, P.N. Suganthan, Q. Zhang, Decomposition Based Multiobjective Evolutionary Algorithm with an Ensemble of Neighborhood Sizes, IEEE Transactions on Evolutionary Computation, in press, http://dx.doi.org/10.1109/TEVC.2011.2166159.
[39] A. Zhou, B.Y. Qu, H. Li, S.Z. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state-of-the-art, Swarm and Evolutionary Computation 1 (1) (2011) 32–49.
[40] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization, in: K.C. Giannakoglou, D.T. Tsahalis, J. Périaux, K.D. Papailiou, T. Fogarty (Eds.), Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems (EUROGEN 2001), Athens, Greece, 2001, pp. 95–100.
[41] E. Zitzler, L. Thiele, Multiple objective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Transactions on Evolutionary Computation 3 (4) (1999) 257–271.
[42] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation 7 (2) (2003) 117–132.