Computers & Operations Research 40 (2013) 1648–1660

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

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

MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives

Yan-yan Tan [a,*], Yong-chang Jiao [a], Hong Li [b], Xin-kuan Wang [a]

[a] National Key Laboratory of Antennas and Microwave Technology, Xidian University, Xi'an, Shaanxi 710071, PR China
[b] School of Science, Xidian University, Xi'an, Shaanxi 710071, PR China

ARTICLE INFO

Available online 12 January 2012

Keywords:
Multiobjective optimization
Decomposition
Uniform design
Weight vector
Many-objective problems

ABSTRACT

To extend multiobjective evolutionary algorithm based on decomposition (MOEA/D) in higher dimensional objective spaces, this paper proposes a new version of MOEA/D with uniform design, named the uniform design multiobjective evolutionary algorithm based on decomposition (UMOEA/D), and compares the proposed algorithm with MOEA/D and NSGA-II on some scalable test problems with three to five objectives. UMOEA/D adopts the uniform design method to set the aggregation coefficient vectors of the subproblems. Compared with MOEA/D, distribution of the coefficient vectors is more uniform over the design space, and the population size neither increases nonlinearly with the number of objectives nor considers a formulaic setting. The experimental results indicate that UMOEA/D outperforms MOEA/D and NSGA-II on almost all these many-objective test instances, especially on problems with higher dimensional objectives and complicated Pareto set shapes. Experimental results also show that UMOEA/D runs faster than NSGA-II for the problems used in this paper. In additional, the results obtained are very competitive when comparing UMOEA/D with some other algorithm on the multiobjective knapsack problems.

1. Introduction

Multiobjective evolutionary algorithms (MOEAs) have been recognized to be well-suited for multiobjective optimization problems (MOPs) since early because they can process a set of solutions in parallel, thereby can obtain an approximation of the Pareto front which consists of multiple Pareto optimal solutions in a single run. Up to present, various MOEAs have been proposed to solve MOPs effectively [1–9]. Multiobjective evolutionary algorithm based on decomposition (MOEA/D) is a very recent one using decomposition [6]. MOEA/D explicitly decomposes an MOP into a number of scalar optimization subproblems. It solves these subproblems simultaneously by evolving a population of solutions. At each generation, the population is composed of the best solution found so far for each subproblem. It has been proved that MOEA/D has a lower complexity than NSGA-II [2], the most popular MOEA at each iteration [6,10], and the algorithm has been ranked first among 13 entries in the unconstrained MOEA competition in CEC 2009 [11]. Therefore, its study is of great significance for the MOEAs.

In the field of multiobjective evolutionary optimization, a lot of test problems and applications with two or three objectives have been studied. Problems with more than three objectives, which have been termed many-objective problems by Farina and Amato [12], have been tackled only rarely. Many techniques that work well for only a few objectives are anticipated to have difficulties in high-dimensional objective spaces. Thus, many-objective optimization is significantly more challenging than scenarios usually being analyzed.

MOEA/D decomposes an MOP into a number of scalar optimization subproblems and solves them simultaneously. The objective in each of these subproblems is an aggregation of all the objectives in the MOP under consideration. 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. In a sense, the uniformity of the aggregation coefficient vectors reflects the uniformity of the Pareto optimal solutions for MOEAs based on decomposition. Thus, whether aggregation coefficient vectors of the subproblems can fill the entire space evenly is essential for the quality of Pareto optimality. A variety of methods to obtain an evenly distributed subset of weights in a simplex for scalarization functions are available in the specialized literature (see for example [13,14]). MOEA/D used the simplex-lattice design method [15]. It works well in MOEA/D, however, the distribution of the

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

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2012.01.001

aggregation coefficient vectors is not very uniform, and the design method requires population size N to satisfy the restriction $N = C_{H+m-1}^{m-1}$, N increases nonlinearly with m (m is the number of objectives) 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 [6,16]. 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 [18] 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 [19–21]. Ford motor company has adopted uniform designs as a standard procedure in product design and process design. Investigations have shown that uniform design performs better at estimating nonlinear problems than other designs, and is robust against model assumptions [22]. These applications illustrate that the UD method is a power tool. The essence of the UD method is to find a set of vectors that are uniformly distributed over the design space. Note that a "uniformly distributed" set of vectors as stated here means roughly that the set has a small discrepancy (described in Section 3), not a set of vectors which are uniformly distributed in the usual statistical sense.

Uniform design multiobjective evolutionary algorithm based on decomposition (UMOEA/D) is proposed in this paper, which uses the uniform design method to generate the aggregation coefficient vectors of the subproblems. Compared with MOEA/D (using the simplex-lattice design method), population size neither becomes very large as the number of objectives increases nor considers a formulaic setting, and the quality of the Pareto optimality is improved for many-objective problems. Additionally, UMOEA/D's superiority over NSGA-II on diversity, convergence and running time is also illustrated in our experimental study. However, our motivation is not to modify MOEA/D but to extend its application in many-objective optimization.

The remainder of this paper is organized as follows. Section 2 presents the multiobjective optimization problem. Section 3 describes the UD method for generating the aggregation coefficient vectors. Section 4 introduces the framework of UMOEA/D. Section 5 compares UMOEA/D with MOEA/D and some other algorithm, experimental results and analysis are then given in this section. Section 6 concludes this paper.

2. Statement of the problem

A bound-constrained multiobjective optimization problem can be stated as follows:

$$\begin{aligned} &\text{minimize} \quad F(x) = (f_1(x), \ldots, f_m(x))^\top \\ &\text{subject to} \quad l_i \le x_i \le u_i, \end{aligned} \tag{1}$$

where $x = (x_1, \ldots, x_n)^\top$ is called decision vector, $l = (l_1, \ldots, l_n)^\top$ and $u = (u_1, \ldots, u_n)^\top$ are the lower bound and upper bound of the decision vector respectively. F is a vector of objectives which are to be optimized. The optimal solutions of (1) can be defined in terms of Pareto optimality [23].

Let $u = (u_1, \ldots, u_m)^\top, v = (v_1, \ldots, v_m)^\top$ be two vectors, u is said to dominate v if and only if $u_i \le v_i$ for every $i \in \{1, \ldots, m\}$ and $u_j < v_j$ for at least one index $j \in \{1, \ldots, m\}$. A point $x^* \in [l,u]$ is Pareto optimal to (1) if there is no $x \in [l,u]$ such 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 Pareto objective vectors is the Pareto front (PF): $\mathrm{PF} = \{F(x^*) = (f_1(x^*), \ldots f_m(x^*))^\top \mid x^* \in \mathrm{PS}\}$ [23].

Instead of searching for a single or just a few optimal solutions as in solving single-objective problems, the goal of handling multiobjective problems is to find the Pareto front as well as the Pareto set of the problem. Problems with more than three objectives are termed many-objective problems. Given limited computational resource, including time and storage, how to provide good solutions in terms of both quality and spread is the key and challenging task for many-objective optimization.

MOEA/D is a very recent MOEA, its superiority over other state-of-the-art MOEAs has been proved in [6,10,11]. It is a hybrid framework making use of decomposition methods in mathematics and the optimization paradigm in evolutionary computation. MOEA/D used the simplex-lattice design method [15] to set the aggregation coefficient vectors. Let N be the number of decomposed subproblems (i.e., the population size) and $\lambda^1, \ldots, \lambda^N$ be the aggregation coefficient vectors, subproblem i corresponds to the coefficient vector $\lambda^i = (\lambda_1^i, \ldots, \lambda_m^i)^\top$, i.e., $\lambda_j^i \ge 0, j = 1, \ldots, m$ and $\sum_{j=1}^m \lambda_j^i = 1$ for all $i = 1, \ldots, N$ (m is the number of objectives). Because of the simplex-lattice design method, population size N (i.e., the number of subproblems) and aggregation coefficient vectors $\lambda^1, \ldots, \lambda^N$ are controlled by m and an integer H in MOEA/D. 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 $N = C_{H+m-1}^{m-1}$. It works well in MOEA/D, however, the distribution of the aggregation coefficient vectors is not very uniform, and the population size N will become very large when m, the number of the objectives is large. For example, to keep the precise of the weight vectors, we set H a constant (e.g., 20), then population size will be 21, 231, 1771, 10626 respectively for problems with 2–5 objectives according to the simplex-lattice design. Meanwhile, this design method requires population size to satisfy the restriction $N = C_{H+m-1}^{m-1}$, 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. Uniform design is a kind of space-filling design, it is a special method which represents a combination of number theory and numerical analysis. In the next section, we shall explain how uniform design method can be constructed for generating the aggregation coefficient vectors in the proposed UMOEA/D.

3. 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 an experiment for determining the proportion of ingredients in a polymer mixture that will produce plastics products with the highest tensile strength. Similar experiments are very commonly encountered in industries. Designs for deciding how to mix the ingredients are called experimental designs with mixtures. A design of N runs for mixtures of m ingredients is a set of N points in the domain:

$$T_m = \{(\lambda_1, \ldots, \lambda_m) : \lambda_j \ge 0, \quad j = 1, \ldots, m, \lambda_1 + \cdots + \lambda_m = 1\}. \tag{2}$$

In the last six decades, a lot of work which appeared in the statistical literature proposed many kinds of designs. Scheffé introduced the simplex-lattice designs and the corresponding polynomial models [15]. Later he introduced an alternative design, the simplex-centroid design [24], to the general simplex-lattice. Cornell gave a suggestion of axial design [25] and he gave a comprehensive review of nearly all the statistical articles on designs of experiments with mixtures and data analysis [26].

- 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.,

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

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 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 the simplex T_m 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 just adopts the simplex-lattice design to set the aggregation coefficient vectors. Fig. 1(a–c) shows the above three designs in $m=3$. There are at least two problems can be found with these three designs [27]:

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

MOEA/D decomposes an MOP into a number of scalar optimization subproblems and solves 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 for MOEA/D. MOEA/D uses 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 if there have too many objectives with zero weight value in subproblems.

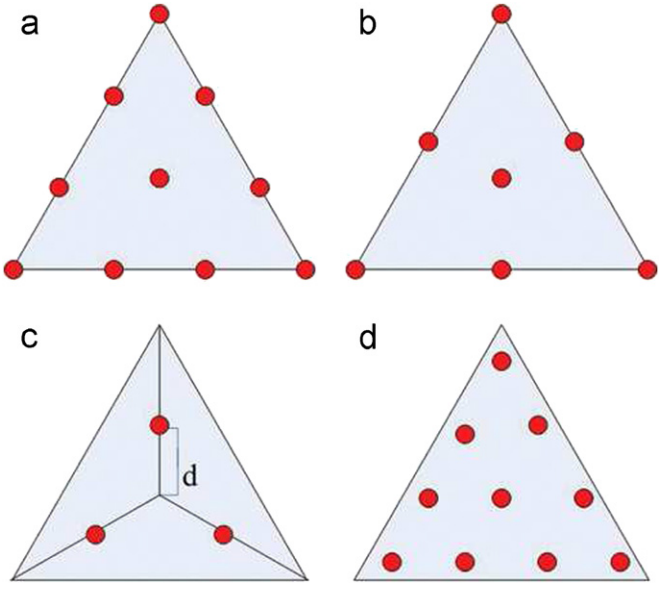

Fig. 1. Four designs of experiments with mixtures in $m=3$. (a) Simplex-lattice design, (b) simplex-centroid design, (c) axial design and (d) uniform design.

In addition, the numbers of weight vectors in these three designs are restricted, i.e., the number of weight vectors is of the type C_{H+m-1}^{m-1} in the simplex-lattice design, of the form 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. Uniform design for experiments with mixtures (UDEM), proposed by Wang and Fang [28], overcomes these drawbacks [27,28]. Fig. 1(d) shows the UDEM in $m=3$. UDEM is the UD on T_m.

In what follows, we shall explain how UDEM can be constructed for generating the aggregation coefficient vectors in the proposed UMOEA/D. UMOEA/D decomposes an MOP into a number of scalar optimization subproblems. Let N be the number of decomposed subproblems (i.e., the population size), UMOEA/D evolves these N subproblems simultaneously in a single run. Let $\lambda^1, \ldots, \lambda^N$ be the aggregation coefficient vectors, subproblem j corresponds to the coefficient vector $\lambda^j = (\lambda_1^j, \ldots, \lambda_m^j)^\top$, $\lambda_i^j \geq 0, i = 1, \ldots, m$ and $\sum_{i=1}^m \lambda_i^j = 1$ for all $j=1,\ldots,N$, m is the number of objectives. The essence of the UDEM is to find a set of coefficient vectors $P = \{\lambda^1, \ldots, \lambda^N\}$ that distribute uniformly on T_m. Let $M(P)$ be a measure of non-uniformity of P, that is, the smaller the M, the higher the uniformity. If we can find a set of N coefficient vectors P^* with minimum M, then P^* is called a UDEM under M. There is more than one definition of discrepancy for M to measure the non-uniformity of P, 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 $\mathrm{CD}_2(P)$ [29]:

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

is used in our experiments, for it is convenient to compute and invariant under relabeling of coordinate axes. Corresponding formulas for other discrepancies can be found in Fang et al. [29].

Now the question is how to determine the set of coefficient vectors with the smallest discrepancy. The problem is solved through two steps. Let C^m be the super unit cube: $C^m = [0,1] \times \cdots \times [0,1] = [0,1]^m$. First, generate a set of vectors $U^* = \{u_i^* = (u_{i1}^*, \ldots, u_{i(m-1)}^*)^\top, i = 1, \ldots, N\}$ on C^{m-1} with the smallest discrepancy. Then, obtain the coefficient vectors $\lambda^1, \ldots, \lambda^N$ with the uniform design on T_m by transforming U^*.

There are several methods for generating a set of vectors U^*, good lattice point (glp) is adopted in this paper. Find the candidate set of positive integers:

$$H_N = \{h : h < N, \text{the greatest common divisor of } N \text{ and } h \text{ is one}\}. \tag{4}$$

For any $m-1$ distinct elements of H_N, $h_1, h_2, \ldots, h_{m-1}$, generate a $N \times (m-1)$ matrix $U = (u_{ij})$ where $u_{ij} = ih_j (\mathrm{mod}\, N)$ and the multiplication operation modulo N is modified as $1 \leq u_{ij} \leq N$. Denote U by $U(N,h)$, where $h = (h_1, \ldots, h_{m-1})$ is called generating vector

of the U. Denote by $G_{N,m-1}$ the set of all such matrices $U(N,h)$. Find a generating vector h^* with $U(N,h^*) \in G_{N,m-1}$ such that it has the smallest discrepancy over the set $G_{N,m-1}$. Let $U(N,h^*) = (u_{ki})_{k=1,\ldots,N;i=1,\ldots,m-1}$, calculate

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

Then, $U^* = \{u^*_i = (u^*_{i1},\ldots,u^*_{i(m-1)})^\top, i = 1,\ldots,N\}$ is the uniform design on C^{m-1} that we need to construct.

For satisfying the restriction that $\sum_{i=1}^{m} \lambda^j_i = 1$ for all $j = 1,\ldots,N$, the following transformation on U^* is required:

$$\begin{cases} \lambda^k_i = (1 - u^*_{ki} 1/(m-i)) \prod_{j=1}^{i-1} u^*_{kj} 1/(m-j) \quad i = 1,\ldots,m-1, \\ \lambda^k_m = \prod_{j=1}^{m-1} u^*_{kj} 1/(m-j) \end{cases} \tag{6}$$

for each $k = 1,\ldots,N$. Then $\{\lambda^k = (\lambda^k_1,\ldots,\lambda^k_m), k = 1,\ldots,N\}$ is a uniform design on T_m, i.e., the uniformly distributed weight vectors $\lambda^1,\ldots,\lambda^N$ are achieved.

Note that the cardinality of H_N in (4) can be determined by the Euler function $\phi(N)$ [27]. Let $N = p_1^{r_1} \cdots p_t^{r_t}$ be prime decomposition of N, where $p_1,\ldots,p_t$ are different primes and $r_1,\ldots,r_t$ are positive integers. Then $\phi(N) = N(1-1/p_1)\cdots(1-1/p_t)$. For example, $\phi(N) = N-1$ if N is a prime, and $\phi(N) < N/2$ if N is even. The maximum number of components is less than $\phi(N)/2+1$ if the uniform design is generated by the glp method [27]. Then, when the number of objectives exceeds the limit, other methods need to be considered. For the detailed description of other methods, refer to [27]. Fortunately, the number of objectives m is far smaller than $\phi(N)/2+1$ (N is the population size) in most cases when we adopt the uniform design method to generate the aggregation coefficient vectors in experiments.

4. Uniform design multiobjective evolutionary algorithm based on decomposition (UMOEA/D)

MOEA/D decomposes an MOP into a number of scalar optimization subproblems and solves 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 for MOEA/D. UMOEA/D under the framework of MOEA/D [6], tries using the UDEM method to set the aggregation coefficient vectors. With UDEM, the distribution of the weight vectors has the minimum discrepancy, their distribution is uniform under numerical analysis [27]. Meanwhile, population size of the UMOEA/D will not be controlled by the number of objectives with this design method, thus, it is suitable for optimizing high-dimensional objective problems.

There are several approaches for converting an MOP into a number of scalar optimization problems [23]. In our experimental tests, Tchebycheff approach is mainly employed. Let $\lambda^1,\ldots,\lambda^N$ be a set of uniformly distributed weight vectors and z^* be the reference point, i.e., $z^* = (z^*_1,\ldots,z^*_m)^\top, z^*_i = \min\{f_i(x) | x \in [l,u]\}$ for each $i = 1,\ldots,m$. With the Tchebycheff approach, the objective function of the i-th subproblem is in the form [23]:

$$g^{te}(x|\lambda^i, z^*) = \max_{1 \le j \le m} \{\lambda^i_j | f_j(x) - z^*_j |\}, \tag{7}$$

where $\lambda^i = (\lambda^i_1,\ldots,\lambda^i_m)^\top$. UMOEA/D minimizes all these N objective functions simultaneously in a single run.

Algorithm 1. UMOEA/D.

input :N: the number of the subproblems;
T: the number of the weight vectors in the neighborhood of each weight vector, $0 < T < N$;
$\lambda^1,\ldots,\lambda^N$: a set of N uniformly distributed weight vectors;
output: Approximation to the PF: $\{F(x^1),\ldots,F(x^N)\}$

```
1  initialization:
2    foreach i = 1 to N do B(i) = {a,b,...,t};   /*λ^a,λ^b,...,λ^t are the T closest weight vectors to λ^i */
3    initial population x^1,...,x^N by randomly sampling from [l,u], FV^i = F(x^i) ;
4    reference point z = (z_1,...,z_m)^T ;/*z_j = min_{1≤i≤N} f_j(x^i) */;
5  repeat
6    for i = 1 to N do
7      ȳ ← recombination(x^i,x^k,x^l);   /*k and l are randomly selected from B(i) */;
8      y ← mutation(ȳ);
9      if y∉[l,u] then y ← repair(y);
10     foreach j = 1 to m do
11       if z_j > f_j(y) then z_j = f_j(y);
12     end
13     foreach j ∈ B(i) do
14       if g^te(y|λ^j,z) < g^te(x^j|λ^j,z) then
15         x^j = y;
16         FV^j = F(y);
17       end
18     end
19   end
20 until stop criteria are met ;
```

5. Experimental study

In this section, MOEA/D [6] and NSGA-II [2] are compared with UMOEA/D on six continuous scalable test problems with 3–5 objectives. In addition, UMOEA/D is applied on three instances of the knapsack problem (KP), comparisons with NSGA-II [2], SPEA2 [1] and PESA [9] are also given.

5.1. Test instances

Scalable test problems or problems with more than three objectives are limited in the existing MOEAs research. Tables 1 and 2 describe the test problems used in this paper. First, four scalable test problems DTLZ1, DTLZ2, DTLZ3 and DTLZ4 of the DTLZ test function family [30] are invoked in our study. They are chosen on the basis of the following properties: (a) they are easy to construct (bottom-up approach [30]), (b) they can be scaled to any number of decision variables and objectives, (c) exact shape and location of the resulting PF for these problems are known and (d) difficulties in both converging to the true PF and maintaining a widely distributed set of solutions can be controlled. For these four problems, the PS corresponds to $x_m,\ldots,x_n = 0.5$ with arbitrary values for $x_1,\ldots,x_{m-1}$, and the objective function values corresponding to the PF lie on $\sum_{i=1}^{m} f_i = 0.5$ for DTLZ1 and on $\sum_{i=1}^{m} f_i^2 = 1$ for DTLZ2, DTLZ3 and DTLZ4. The g function in DTLZ3 introduces $(3^m - 1)$ local PFs[1] and one

[1] Local Pareto Optimal Set: Given some Pareto optimal set P, if $\forall x \in P$, $\neg\exists y$ satisfying $\|y-x\|_\infty \le \epsilon$, where ϵ is a small positive number (in principle, y is obtained by perturbing x in a small neighborhood), and for which $F(y) \preceq F(x)$, then the solutions in P constitute a local Pareto optimal set. The set of all the local Pareto optimal objective vectors is the local Pareto front (PF).

Table 1
Test problems used in this study.

Problem	Objective functions	Domain
DTLZ1	$f_1(x)=0.5x_1x_2\cdots x_{m-1}(1+g(x_m))$ $f_2(x)=0.5x_1x_2\cdots(1-x_{m-1})(1+g(x_m))$ $\vdots$ $f_{m-1}(x)=0.5x_1(1-x_2)(1+g(x_m))$ $f_m(x)=0.5(1-x_1)(1+g(x_m))$ where $g(x_m)=100[\lvert x_m\rvert+\sum_{x_i\in x_m}((x_i-0.5)^2-\cos(20\pi(x_i-0.5)))]$	$[0,1]^n$
DTLZ2	$f_1(x)=\cos(x_1\pi/2)\cos(x_2\pi/2)\cdots\cos(x_{m-2}\pi/2)\cos(x_{m-1}\pi/2)(1+g(x_m))$ $f_2(x)=\cos(x_1\pi/2)\cos(x_2\pi/2)\cdots\cos(x_{m-2}\pi/2)\sin(x_{m-1}\pi/2)(1+g(x_m))$ $\vdots$ $f_{m-1}(x)=\cos(x_1\pi/2)\sin(x_2\pi/2)(1+g(x_m))$ $f_m(x)=\sin(x_1\pi/2)(1+g(x_m))$ where $g(x_m)=\sum_{x_i\in x_m}(x_i-0.5)^2$	$[0,1]^n$
DTLZ3	$f_1(x)=\cos(x_1\pi/2)\cos(x_2\pi/2)\cdots\cos(x_{m-2}\pi/2)\cos(x_{m-1}\pi/2)(1+g(x_m))$ $f_2(x)=\cos(x_1\pi/2)\cos(x_2\pi/2)\cdots\cos(x_{m-2}\pi/2)\sin(x_{m-1}\pi/2)(1+g(x_m))$ $\vdots$ $f_{m-1}(x)=\cos(x_1\pi/2)\sin(x_2\pi/2)(1+g(x_m))$ $f_m(x)=\sin(x_1\pi/2)(1+g(x_m))$ where $g(x_m)=100[\lvert x_m\rvert+\sum_{x_i\in x_m}((x_i-0.5)^2-\cos(20\pi(x_i-0.5)))]$	$[0,1]^n$
DTLZ4	$f_1(x)=\cos(x_1^\alpha\pi/2)\cos(x_2^\alpha\pi/2)\cdots\cos(x_{m-2}^\alpha\pi/2)\cos(x_{m-1}^\alpha\pi/2)(1+g(x_m))$ $f_2(x)=\cos(x_1^\alpha\pi/2)\cos(x_2^\alpha\pi/2)\cdots\cos(x_{m-2}^\alpha\pi/2)\sin(x_{m-1}^\alpha\pi/2)(1+g(x_m))$ $\vdots$ $f_{m-1}(x)=\cos(x_1^\alpha\pi/2)\sin(x_2^\alpha\pi/2)(1+g(x_m))$ $f_m(x)=\sin(x_1^\alpha\pi/2)(1+g(x_m))$ where $g(x_m)=\sum_{x_i\in x_m}(x_i-0.5)^2,\alpha=100$	$[0,1]^n$
F1	$f_1(x)=x_1x_2\cdots x_{m-1}+\frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}\left(x_j-x_2\cdots x_{m-1}\sin\left(6\pi x_1+\frac{j\pi}{n}\right)\right)^2$ $f_2(x)=x_1x_2\cdots(1-x_{m-1})+\frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}\left(x_j-x_2\cdots x_{m-1}\sin\left(6\pi x_1+\frac{j\pi}{n}\right)\right)^2$ $\vdots$ $f_{m-1}(x)=x_1(1-x_2)+\frac{2}{\lvert J_{m-1}\rvert}\sum_{j\in J_{m-1}}\left(x_j-x_2\cdots x_{m-1}\sin\left(6\pi x_1+\frac{j\pi}{n}\right)\right)^2$ $f_m(x)=(1-x_1)+\frac{2}{\lvert J_m\rvert}\sum_{j\in J_m}\left(x_j-x_2\cdots x_{m-1}\sin\left(6\pi x_1+\frac{j\pi}{n}\right)\right)^2$ where $J_1=\{j\mid m\le j\le n,$and j is a multiplication of $m\}$, $J_2=\{j\mid m\le j\le n,$and $j-1$ is a multiplication of $m\}$, $\vdots$ $J_m=\{j\mid m\le j\le n,$and $j-m+1$ is a multiplication of $m\}$ Its PS is $x_j=x_2\cdots x_{m-1}\sin\left(6\pi x_1+\frac{j\pi}{n}\right), j=m,\ldots,n.$	$[0,1]^{m-1}\times[-1,1]^{n-m+1}$

global PF[2] [30]. All local PFs are parallel to the global PF and an MOEA can get stuck to any of these local PFs, before converging to the global PF (at $g(x_m)=0$). DTLZ4 modifies problem DTLZ2 with a different meta-variable mapping: $x_i\to x_i^\alpha$. The parameter $\alpha=100$ is suggested here [30]. The problem can test a MOEA's ability to maintain a good distribution of solutions.

Inspired by the strategies for constructing test problems in [10], we have proposed two other scalable continuous test instances (F1 and F2) with variable linkages/complicated PS shapes as test functions in our study. This class of functions with complicated PS shapes, proposed by Zhang and Li [10,11] were ever used as the test instances of the unconstrained MOEA competition in CEC 2009 [11]. It has been proved that these complicated PS shapes, as well as the geometrical shapes of the PF, can cause difficulties for MOEAs [31,32].

In addition, as combinatorial optimization problems three instances of the knapsack problem taken from [36] are tested in 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 we refer to the original study.

5.2. Parameter settings

Algorithms are implemented on personal computer (AMD Athlon (tm) 64×2 Dual Core Processor 3800+ 2.0 GHz, 512 MB RAM). Compared algorithms are of the same reproduction operators for each test problem in experiments for comparisons fair. For all continuous problems, the individuals are coded as real

[2] Global PF is what is termed PF, this text's terminology is easily extended to denote a local PF.

Table 2
Test problems used in this study(continued).

Problem	Objective functions	Domain
F2	$f_1(x)=\cos(x_1\pi/2)\cdots\cos(x_{m-2}\pi/2)\cos(x_{m-1}\pi/2)+\frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}\left(x_j-2x_2\cdots x_{m-1}\sin\left(2\pi x_1+\frac{j\pi}{n}\right)\right)^2$ $f_2(x)=\cos(x_1\pi/2)\cdots\cos(x_{m-2}\pi/2)\sin(x_{m-1}\pi/2)+\frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}\left(x_j-2x_2\cdots x_{m-1}\sin\left(2\pi x_1+\frac{j\pi}{n}\right)\right)^2$ $\vdots$ $f_{m-1}(x)=\cos(x_1\pi/2)\sin(x_2\pi/2)+\frac{2}{\lvert J_{m-1}\rvert}\sum_{j\in J_{m-1}}\left(x_j-2x_2\cdots x_{m-1}\sin\left(2\pi x_1+\frac{j\pi}{n}\right)\right)^2$ $f_m(x)=\sin(x_1\pi/2)+\frac{2}{\lvert J_m\rvert}\sum_{j\in J_m}\left(x_j-2x_2\cdots x_{m-1}\sin\left(2\pi x_1+\frac{j\pi}{n}\right)\right)^2$ where $J_1J_2,\ldots J_m$ are the same as those of F1. Its PS is $x_j=2x_2\cdots x_{m-1}\sin\left(2\pi x_1+\frac{j\pi}{n}\right), j=m,\ldots,n.$	$[0,1]^{m-1}\times[-2,2]^{n-m+1}$
KP-750-m	$f_i(x)=\sum_{j=1}^{n}p_{ij}x_j$ s.t. $\sum_{j=1}^{n}w_{ij}x_j\le c_i 1\le i\le m, m=2,3,4.$ where p_{ij} is the profit of item j in knapsack i,w_{ij} is the weight of item j in knapsack i,and c_i is the capacity of knapsack i.	$\{0,1\}^n$

vectors. Our pilot experiments and references [6,10] have indicated that simulated binary crossover (SBX) [33] is more suitable for DTLZ test function family, as well as differential evolution (DE) [7] for problems with complicated PS shapes. Then, crossover (SBX for DTLZ1-DTLZ4, as well as DE for F1 and F2) and mutation (polynomial mutation [33]) operators are applied directly to real parameter values in UMOEA/D, MOEA/D and NSGA-II 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 UMOEA/D, NSGA-II, SPEA2 and PESA for knapsack problems. The parameter settings in this paper are as follows.

(1) Control parameters in reproduction operators:
   (a) Distribution index $\eta_c=20$ and crossover probability $p_c=1$ in the SBX operator;
   (b) Crossover rate $CR=1.0$ and scaling factor $F=0.5$ in the DE operator;
   (c) Distribution index $\eta_m=20$ and mutation probability $p_m=1/n$ in mutation operator.
(2) Population size and weight vectors in the algorithms: Population size N and weight vectors in MOEA/D 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, on the other hand, considering the number of non-dominated solutions increases with the number of objectives in NSGA-II [34], we set H to be 33, 17 and 13 for continuous test instances with objectives 3–5 respectively in MOEA/D. Consequently, the population size N is 595, 1140 and 2380 for the 3-objective, 4-objective and 5-objective continuous test problems respectively in all three algorithms (UMOEA/D, MOEA/D and NSGA-II) for pair comparisons. Different from MOEA/D, weight vectors $\lambda^1,\ldots,\lambda^N$ are obtained by uniform design in UMOEA/D. The population size N is set to 250 for $m=2$, 300 for $m=3$, and 350 for $m=4$ in all four algorithms (UMOEA/D, NSGA-II, SPEA2 and PESA) for the knapsack problems.
(3) Number of the weight vectors in the neighborhood in UMOEA/D and MOEA/D: $T=20$ for the continuous problems and 10 for the knapsack problems.
(4) Number of runs and stopping condition: Each algorithm is run 20 times independently for each test instance. All these algorithms stop after a given number of function evaluations. The maximal number of function evaluations is set to $250\times N$ for all the continuous test problems, and $500\times N$ for the knapsack problems.

5.3. Performance assessment

In order to compare the performance of the different algorithms quantitatively, performance metrics are needed. Performance metrics used in this paper are described below:

- *Inverted generational distance* (*IGD*) [35]: Let P^* be a set of uniformly distributed points in the objective space along the PF. Let P be an approximation to the PF, the inverted generational distance from P^* to P is defined as

$$\mathrm{IGD}(P^*,P)=\frac{\sum_{v\in P^*}d(v,P)}{\lvert P^*\rvert} \tag{8}$$

  where $d(v,P)$ is the minimum Euclidean distance between v and the points in P. If $\lvert P^*\rvert$ is large enough to represent the PF very well, IGD(P^*,P) could measure both the diversity and convergence of P in a sense. To have a low value of IGD(P^*,P), P must be very close to the PF and cannot miss any part of the whole PF.
  In experiments, we select 500, 1000 and 2000 uniformly distributed points in PF and let these points be P^* for the 3-objective, 4-objective and 5-objective continuous problems respectively.
- *Coverage of two sets* (C): The C-metric [36] 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.,

$$C(A,B) = \frac{|\{u \in B | \exists v \in A : v \text{ dominates } u\}|}{|B|} \quad (9)$$

C(*A*,*B*) is not necessarily equal to 1-C(*A*,*B*). 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*) [36]: 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 low dimension, 2- and 3-objective spaces, it is known as area and volume respectively. Mathematically, the HV-metric is described as

$$HV(P) = \left\{ \bigcup vol(v) | v \in P \right\} \quad (10)$$

The fourth metric used in this study is simply the running time of an algorithm (in seconds) for the particular settings. It has been included in the performance metric set to evaluate how an MOEA scales in terms of time with increase in number of objectives.

5.4. Experimental results and analysis

The number of function evaluations matters when the objective functions are very costly to evaluate, especially when the number of the objectives increases. Figs. 2, 4, and 6 show the evolution of the average IGD-metric values of the current

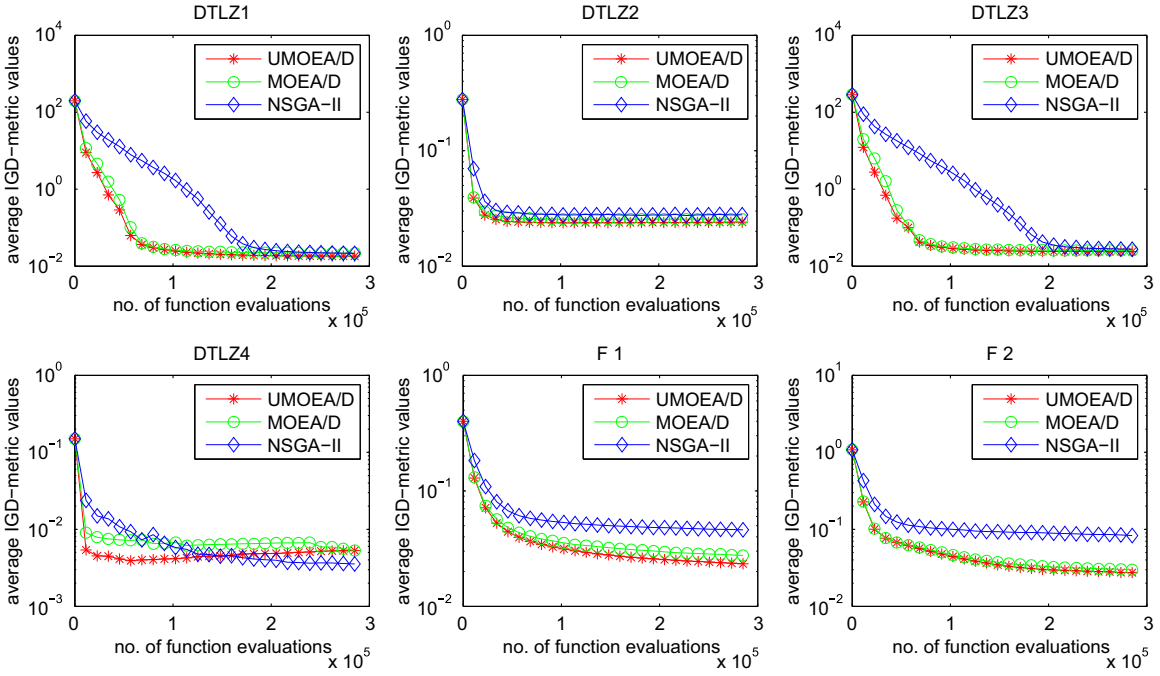

Fig. 2. Evolution of the mean of IGD-metric values for 3-objective continuous test problems.

Table 3
IGD statistics of the UMOEA/D, MOEA/D and NSGA-II based on 20 independent runs for continuous test problems.

Objectives	Problem	UMOEA/D			MOEA/D			SS(1)	NSGA-II			SS(2)
		Mean	Min	Std	Mean	Min	Std		Mean	Min	Std	
3	DTLZ1	0.0182	0.0179	0.0001	0.0219	0.0216	0.0001	+	0.0216	0.0205	0.0007	+
	DTLZ2	0.0240	0.0237	0.0001	0.0256	0.0254	0.0001	+	0.0279	0.0263	0.0007	+
	DTLZ3	0.0240	0.0236	0.0002	0.0256	0.0253	0.0002	+	0.0277	0.0270	0.0006	+
	DTLZ4	0.0053	0.0023	0.0076	0.0053	0.0016	0.0046	·	0.0036	0.0031	0.0003	·
	F1	0.0234	0.0222	0.0007	0.0276	0.0256	0.0010	+	0.0460	0.0423	0.0018	+
	F2	0.0274	0.0253	0.0027	0.0301	0.0281	0.0017	+	0.0832	0.0684	0.0077	+
4	DTLZ1	0.0352	0.0348	0.0002	0.0862	0.0853	0.0004	+	0.0478	0.0445	0.0019	+
	DTLZ2	0.0534	0.0530	0.0002	0.0879	0.0868	0.0005	+	0.0622	0.0602	0.0015	+
	DTLZ3	0.0536	0.0529	0.0004	0.0864	0.0844	0.0008	+	0.0632	0.0607	0.0015	+
	DTLZ4	0.0045	0.0039	0.0003	0.0166	0.0129	0.0033	+	0.0113	0.0076	0.0031	+
	F1	0.0407	0.0398	0.0006	0.0827	0.0802	0.0011	+	0.0975	0.0901	0.0042	+
	F2	0.0640	0.0600	0.0023	0.0871	0.0821	0.0020	+	0.1086	0.0979	0.0075	+
5	DTLZ1	0.0438	0.0432	0.0003	0.1645	0.1618	0.0009	+	0.1059	0.0878	0.0194	+
	DTLZ2	0.0758	0.0749	0.0005	0.1750	0.1711	0.0016	+	0.0987	0.0946	0.0028	+
	DTLZ3	0.0768	0.0756	0.0006	0.1710	0.1643	0.0027	+	0.1205	0.1055	0.0180	+
	DTLZ4	0.0057	0.0053	0.0002	0.0256	0.0255	0	+	0.0232	0.0167	0.0060	+
	F1	0.0467	0.0461	0.0004	0.1598	0.1575	0.0010	+	0.2524	0.2158	0.0203	+
	F2	0.0818	0.0809	0.0005	0.1669	0.1552	0.0057	+	0.5043	0.3987	0.0542	+

population to P^* with the number of function evaluations for continuous problems with 3–5 objectives respectively. These results indicate that UMOEA/D converges, in terms of the number of the function evaluations, much faster than MOEA/D and NSGA-II in minimizing the IGD-metric value for almost all test instances, especially for problems with higher dimensional objectives. In other words, UMOEA/D needs fewer function evaluations than MOEA/D and NSGA-II for minimizing the IGD-metric value for almost all these continuous test problems, which suggests that UMOEA/D is more efficient and effective than MOEA/D and NSGA-II for these continuous problems. Meanwhile, for 3-objective and 5-objective problems, MOEA/D converges, in terms of the number of the function evaluations, faster than NSGA-II for DTLZ1, DTLZ3, F1 and F2, and at about the same speed as or a bit slower than NSGA-II for DTLZ2 and DTLZ4, while for 4-objective problems, MOEA/D converges, in terms of the number of the function evaluations, faster than NSGA-II for DTLZ1, DTLZ3, and performs similar to NSGA-II for other problems.

Table 3 presents the minimum, mean, and standard deviation of the IGD-metric values of the 20 final populations. In order to determine the statistical significance of the advantage of UMOEA/D over MOEA/D and NSGA-II, *t*-test is applied on the IGD-metric and the results are shown in the ninth and last columns of Table 3 respectively. MOEA/D and NSGA-II are compared with UMOEA/D on the continuous problems. Note that here '+' indicates UMOEA/D is superior to the compared algorithm at the 0.05 significance level and '·' otherwise. Box plots of the IGD-metric values based on 20 independent runs among three algorithms for the continuous problems with 3–5 objectives are visualized in Figs. 3, 5, and 7 respectively. Figs. 3, 5, and 7, together with Table 3 reveal that

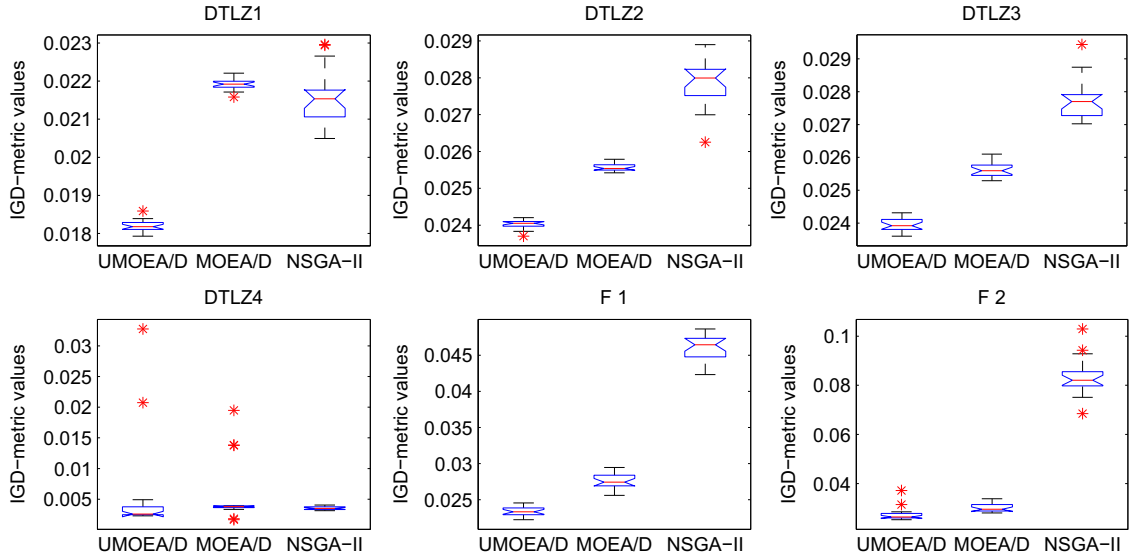

Fig. 3. Box plots of the IGD-metric values based on 20 independent runs obtained by UMOEA/D, MOEA/D and NSGA-II for 3-objective continuous test problems.

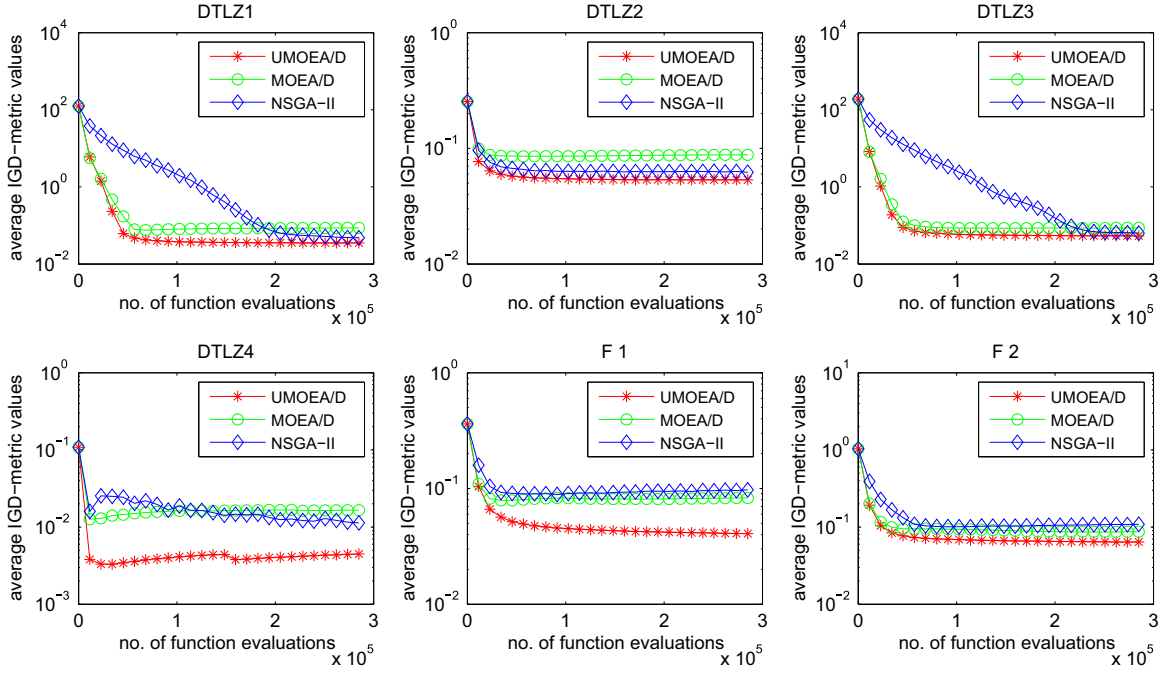

Fig. 4. Evolution of the mean of IGD-metric values for 4-objective continuous test problems.

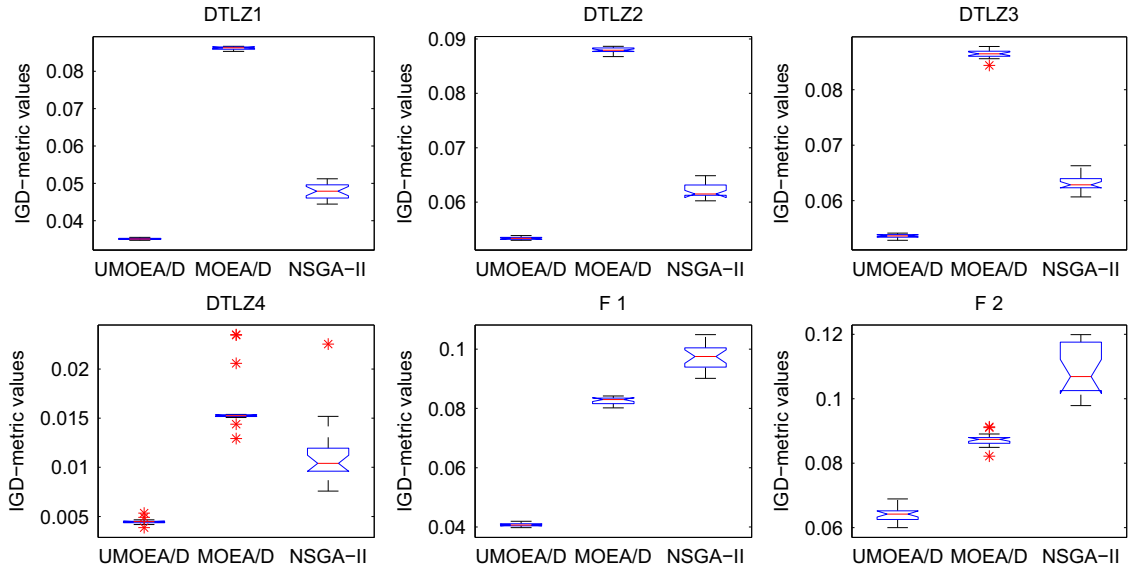

Fig. 5. Box plots of the IGD-metric values based on 20 independent runs obtained by UMOEA/D, MOEA/D and NSGA-II for 4-objective continuous test problems.

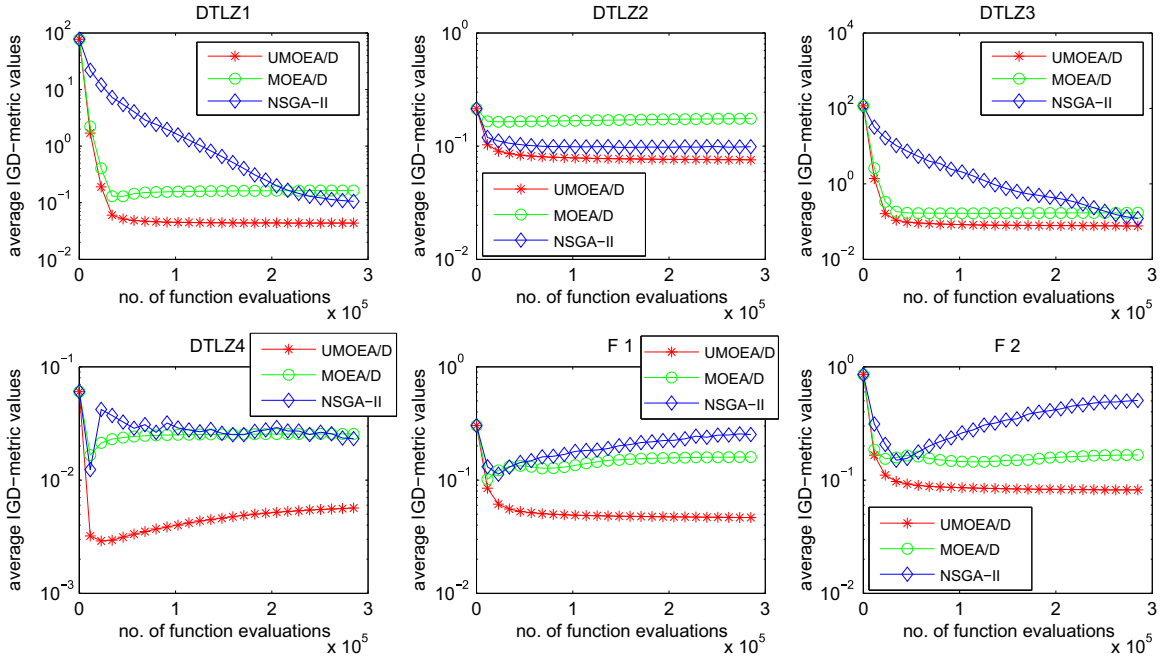

Fig. 6. Evolution of the mean of IGD-metric values for 5-objective continuous test problems.

UMOEA/D behaves stably, and in terms of IGD-metric, the final solutions obtained by UMOEA/D are better than MOEA/D and NSGA-II for all the continuous test problems except 3-objective DTLZ4. Roughly speaking, NSGA-II outperforms or performs similar to MOEA/D on the DTLZ test function family, but performs worse on F1 and F2 with complicated PS shapes.

Table 4 presents the mean, minimum, maximum and standard deviation of the CPU time (in seconds) used by three algorithms (UMOEA/D, MOEA/D and NSGA-II) over 3–5 objectives based on 20 independent runs. Roughly speaking, the CPU time used by UMOEA/D for each problem has no prominent difference with what used by MOEA/D. For DTLZ test function family, UMOEA/D runs about 16 times as fast as NSGA-II with the same number of function evaluations for the 3-objective test instances, more than 20 times for the 4-objective test instances, and about 25 times for the 5-objective test problems. For F1 and F2 with 3–5 objectives, UMOEA/D runs, on average about 10 times as fast as NSGA-II with the same number of function evaluations. Therefore, we can claim that UMOEA/D runs faster than NSGA-II.

It is clear from Table 3 and Figs. 2–7 that, the increase in the number of objectives can deteriorate the performance of all these three algorithms. Fig. 6 shows that the evolution of the IGD-metric with the number of function evaluations get worse for problems F1 and F2 with complicated PS shapes in MOEA/D and NSGA-II when the number of the objectives is up to 5 especially in NSGA-II, may for the reason that the number of non-dominated solutions could increase with the number of objectives in algorithms, as well as the problems with

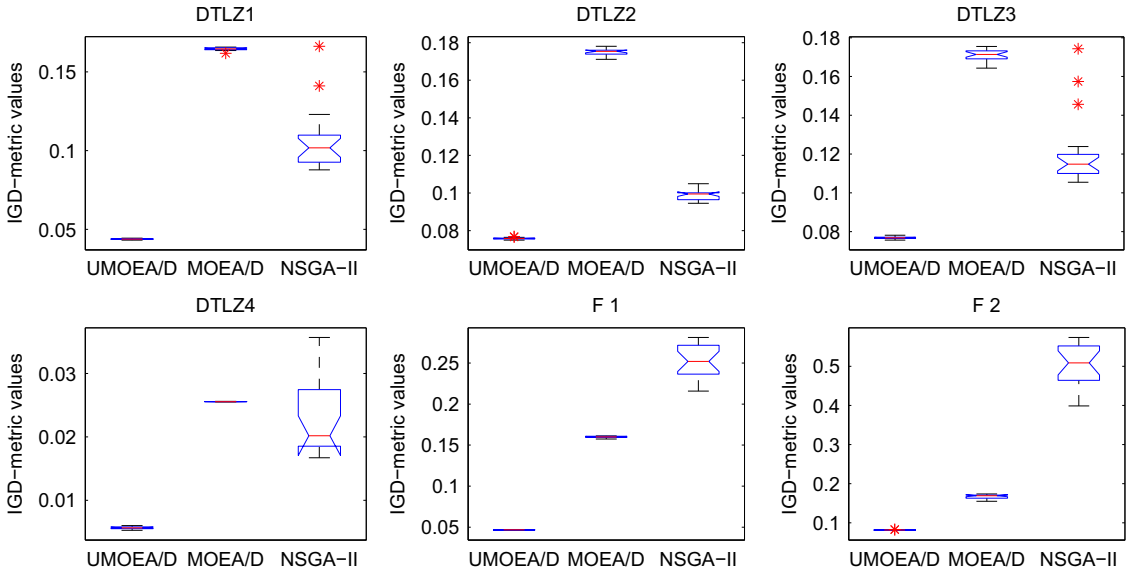

Fig. 7. Box plots of the IGD-metric values based on 20 independent runs obtained by UMOEA/D, MOEA/D and NSGA-II for 5-objective continuous test problems.

Table 4
Running time statistics of the UMOEA/D, MOEA/D and NSGA-II based on 20 independent runs for continuous test problems.

Objectives	Problem	UMOEA/D				MOEA/D				NSGA-II			
		Mean	Min	Max	Std	Mean	Min	Max	Std	Mean	Min	Max	Std
3	DTLZ1	20.7	20	21	0.470	20.75	20	21	0.444	361.85	352	390	9.996
	DTLZ2	20.5	20	21	0.513	20.6	20	21	0.503	318.55	317	322	1.504
	DTLZ3	20.7	20	21	0.470	21.25	20	30	2.100	357.15	351	360	2.231
	DTLZ4	20.55	20	21	0.510	20.8	20	22	0.696	318.7	317	321	1.031
	F1	39.45	39	45	0.510	39.35	39	40	0.489	359.95	358	367	2.438
	F2	39.8	39	40	1.322	39.5	39	40	0.513	350.7	343	368	6.105
4	DTLZ1	62.85	62	64	0.489	62.65	62	63	0.489	1460.6	1312	2047	234.364
	DTLZ2	62.6	62	65	0.754	62.4	62	63	0.503	1251.9	1229	1440	50.2624
	DTLZ3	63	62	64	0.324	63.1	62	65	0.788	1357.7	1341	1370	9.3261
	DTLZ4	62.9	62	64	0.553	63.4	62	73	2.303	1240.8	1230	1249	5.0291
	F1	149.85	149	151	0.745	150.15	149	154	0.933	1305.3	1288	1486	24.1536
	F2	149.55	148	151	0.826	149.95	149	152	1.099	1312.3	1261	1405	44.9443
5	DTLZ1	246.95	244	252	2.164	248.25	244	253	2.789	6463.9	6243	6819	161.178
	DTLZ2	245.35	245	246	0.489	245	244	247	0.858	6202.1	6039	6350	125.479
	DTLZ3	246.35	245	250	1.04	244.15	243	245	0.745	6566.4	6357	7075	225.861
	DTLZ4	246.25	246	247	0.444	245.35	245	246	0.489	6155.9	6020	6593	146.270
	F1	755.9	735	788	42.12	732.15	723	748	15.16	5973.1	5836	6299	133.324
	F2	745.1	734	925	11.87	733.95	727	794	5.226	6039.1	5920	6370	121.775

Table 5
Average values (standard deviation) of *C*-metric based on 20 independent runs for the knapsack problems.

Algorithm		*C*(*A*,*B*)		
A	*B*	750-2	750-3	750-4
UMOEA/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	UMOEA/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)

complicated PS shapes could be very challenging for an MOEA. However, UMOEA/D outperforms MOEA/D and NSGA-II for this class of higher dimensional problems with complicated PS shapes.

Table 5 shows the mean and standard deviation of the *C*-metric values of the final approximations obtained by UMOEA/D and other three algorithms. Table 6 presents the mean of the HV-metric values in four algorithms (UMOEA/D, NSGA-II,

Table 6
Average values of HV-metric based on 20 independent runs for the knapsack problems.

Problem	UMOEA/D	NSGA-II	SS(1)	SPEA2	SS(2)	PESA	SS(3)
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}	+

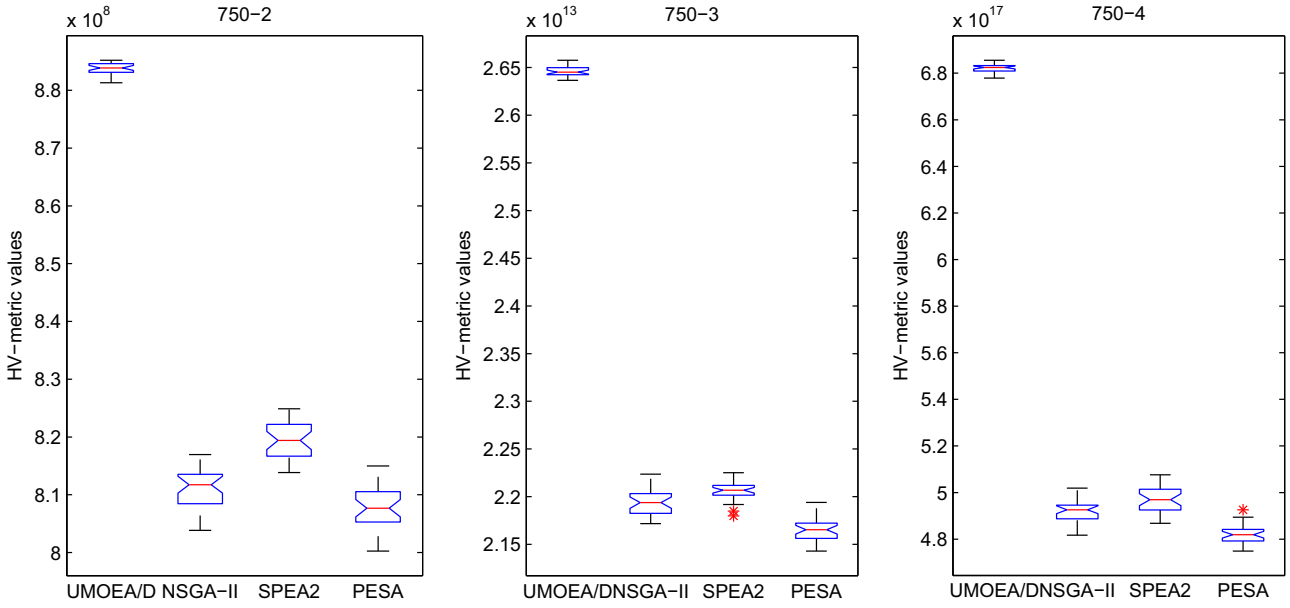

Fig. 8. Box plots of the HV-metric values based on 20 independent runs for the knapsack problems.

SPEA2, PESA) for the knapsack problems. In the fourth, sixth and eighth columns of Table 6, we report the statistical significance (SS) of the advantage of UMOEA/D over MOEA/D, NSGA-II, SPEA2 and PESA respectively in the mean HV-metric value. The box plot of the HV-metric values based on 20 independent runs among these four algorithms is visualized in Fig. 8. Fig. 9 plots the distribution of the final approximation with the largest HV-metric value found by each algorithm for the 2-objective knapsack problem.

Table 5 shows that the final approximation obtained by UMOEA/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%, 95.87% and 68.29% of the final solutions generated by NSGA-II, SPEA2 and PESA respectively are dominated by those by UMOEA/D, and only 0.02%, 0.06% and 0.79% vice versa respectively.

It is evident from Table 6 and Fig. 8 that the HV-metric value of UMOEA/D is significantly larger than that of other three algorithms for each knapsack problem. These results indicate that UMOEA/D dominates more search space than NSGA-II, SPEA2 and PESA. Fig. 9 visually shows that UMOEA/D can find higher quality solutions than other three compared algorithms. Overall, we can claim that UMOEA/D outperforms NSGA-II, SPEA2 and PESA on these knapsack problems.

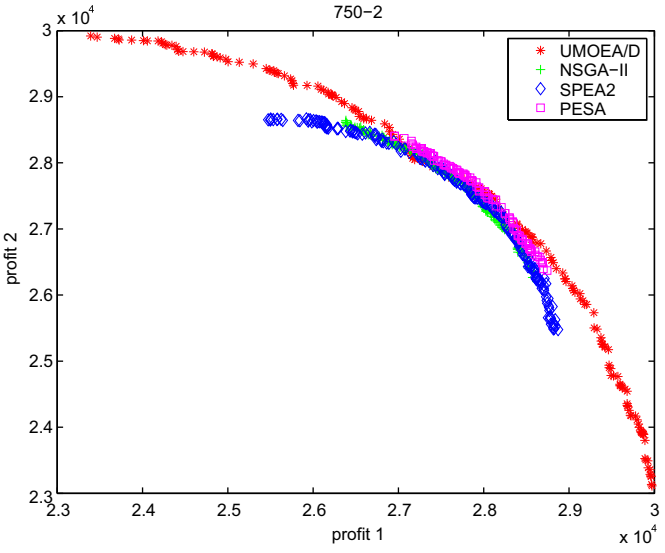

Fig. 9. Plots of the nondominated solutions with the largest HV-metric in 20 runs for the 2-objective knapsack problem.

5.5. Impacts of parameter settings

(1) Sensitivity of T in UMOEA/D: T is a major control parameter in UMOEA/D. To study the sensitivity of the performance to T in UMOEA/D, we have tested different settings of T in the implementation of UMOEA/D for 3-objective DTLZ2. All the other parameter settings are the same as in Section 5.2 except the setting of T. As clearly shown in Fig. 10(a), UMOEA/D performs very well for all the values of T except very small ones on DTLZ2. Thus, we can claim that UMOEA/D is not very sensitive to the setting of T, at least for MOPs that are somehow similar to these test instances. Meanwhile, the reason that UMOEA/D dose not work well when T is very

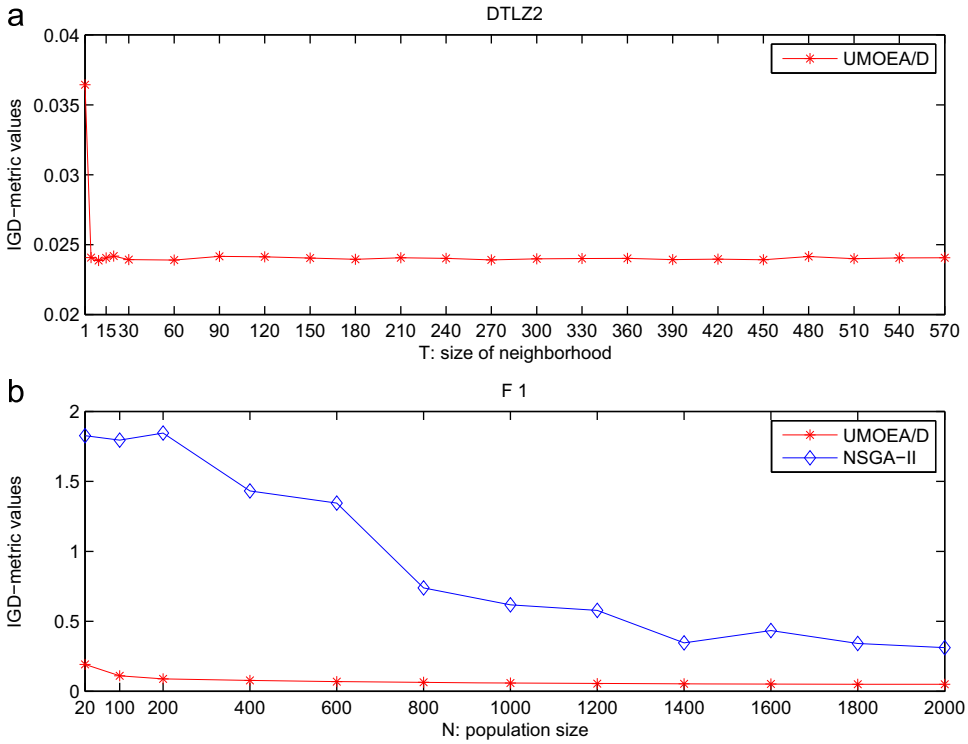

Fig. 10. Sensitivity of T and N in UMOEA/D.

small could be because UMOEA/D with too small T is poor at exploration.

(2) Setting of population size N: Population size N is an important control parameter in MOEA, especially for problems with many objectives. To investigate the impact of population size on the performances of UMOEA/D and NSGA-II, We have tested different settings of N in both algorithms on F1 with 5 objectives. All the other parameters remain the same as in Section 5.2. Fig. 10(b) reveals the IGD-metric values in both algorithms with 12 different settings of N. It is evident that UMOEA/D is less sensitive to the setting of population size N under the ranges considered than NSGA-II. It also suggests that the tuning of population size can not make NSGA-II beat UMOEA/D on this instance, and thus the advantage of UMOEA/D over NSGA-II may come from its decomposition strategy.

6. Conclusion

MOEA/D proposed by Zhang and Li has been proved to be an effective and efficient algorithm. It uses the simplex-lattice design method to set the values of 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 restrict MOEA/D's use in optimizing problems with many objectives. To overcome these drawbacks and extend the insight in the behavior of MOEA/D in higher dimensional objective spaces, we propose UMOEA/D and compare it with other algorithms on some scalable continuous test problems and the knapsack problems.

UMOEA/D under the framework of MOEA/D, adopts the uniform design method to generate the coefficient vectors. With the uniform design, the distribution of the weight vectors has the minimum discrepancy, their distribution is uniform over the design space. Meanwhile, population size of the UMOEA/D will not be controlled by the number of objectives with this design method, thus, it is suitable for optimizing high-dimensional objective problems.

Experimental results have shown that UMOEA/D outperforms MOEA/D and NSGA-II on most of the continuous test instances, especially on problems with higher dimensional objectives and complicated PS shapes. Experimental results have also indicated that UMOEA/D runs faster than NSGA-II for each continuous problem with three to five objectives, i.e., UMOEA/D could be good at finding a small number of Pareto solutions in less time. In addition, as combinatorial optimization problems three instances of the multiobjective knapsack problem have been tested on UMOEA/D and other three algorithms. Comparisons show that UMOEA/D significantly outperforms NSGA-II, SPEA2 and PESA on these knapsack problems. In the end of this paper, we have experimentally investigated the sensitivity to the neighborhood size T and population size N of UMOEA/D. Experimental results show that UMOEA/D is not very sensitive to the setting of T, and it is less sensitive to the setting of population size N than NSGA-II.

In the future, we plan to study the ability of UMOEA/D for MOPs in uncertain and noisy environments, and how to improve the efficiency of the algorithm remains to be studied further.

Acknowledgment

The authors would like to thank professor Qingfu Zhang for his constructive suggestions on this work. They also gratefully thank anonymous referees for their invaluable comments and remarks.

References

[1] Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization. In: Proceedings of the evolutionary methods for design optimization and control with applications to industrial problems, Athens, Greece; 2001. p. 95–100.

[2] Deb K, Agrawal S, Pratap A, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 2002;6(2):182–97.
[3] Coello Coello CA, Pulido GT, Lechuga MS. Handling multiple objectives with particle swarm optimization. IEEE Transactions on Evolutionary Computation 2004;8(3):256–79.
[4] Gong MG, Jiao LC, Du HF, Bo LF. Multiobjective immune algorithm with nondominated neighbor-based selection. Evolutionary Computation 2008; 16(2):225–55.
[5] Zhang QF, Zhou AM, Jin Y. RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm. IEEE Transactions on Evolutionary Computation 2008;21(1):41–63.
[6] Zhang QF, Li H. MOEA/D: a multi-objective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 2007;11(6): 712–31.
[7] Price K, Storn RM, Lampinen JA. Differential evolution: a practical approach to global optimization. Natural computing series. NJ, USA: Springer; 2005;
Yano H. Interactive decision making for multiobjective programming problems with fuzzy domination structures. International Journal of Innovative Computing, Information and Control 2009;5(12(B)):4867–75.
[8] Hughes EJ. MSOPS-II: A general-purpose many-objective optimiser. in: IEEE congress on evolutionary computation (CEC'07), Singapore; 2007. p. 3944–51.
[9] Corne DW, Knowles JD, Oates MJ. The Pareto-envelope based selection algorithm for multi-objective optimization. in: Schoenauer M, Deb K, Rudolph G, Yao X, Lutton E, Merelo JJ, editors. Parallel problem solving from nature, PPSN VI. Lecture notes in computer science. Berlin: Springer-Verlag; 2000. p. 869–78.
[10] Li H, Zhang QF. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation 2009;12(2):284–302.
[11] Zhang QF, Liu W, Li H. 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; 2009.
[12] Farina M, Amato P. On the optimal solution definition for many-criteria optimization problems. in: Proceedings of the NAFIPS-FLINT international conference 2002 Piscataway NJ. IEEE Service Center; 2002. p. 233–8.
[13] Steuer RE. Multiple criteria optimization: theory, computation and application. NY: Wiley; 1985.
[14] Jaszkiewicz A, Slowinski R. The LBS package – a microcomputer implementation of the light beam search method for multiple-objective non-linear mathematical programming. Working Paper WP-94-07, International Institute for Applied Systems Analysis, Laxenburg, Austria; 1994.
[15] Scheffé H. Experiments with mixtures. Journal of Royal Statistical Society Series B 1958;20:344–60.
[16] Thomas JS, Brian JW, William N. The design of analysis of computer experiments. Berlin: Springer-Verlag; 2003.
[17] Hughes EJ. Evolutionary many-objective optimisation: many once or one many. In: IEEE congress on evolutionary computation (CEC'05), Edinburgh, UK, vol. 1, 2005. p. 222–7.
[18] Fang KT. The uniform design: application of number-theoretic methods in experimental design. Acta Mathematicae Applicatae Sinica 1980;3:363–72.
[19] Liang YZ, Fang KT, Xu QS. Uniform design and its applications in chemistry and chemical engineering. Chemometrics and Intelligent Laboratory Systems 2001;58:43–57.
[20] Chan LY, Lo ML. Quality improvement in the manufacture of liquid crystal displays using uniform design. International Journal of Materials and Product Technology 2004;20:127–42.
[21] Li R, Lin DKJ, Chen Y. Uniform design: design analysis and applications. International Journal of Materials and Product Technology 2004;20:101–14.
[22] Xu QS, Liang YZ, Fang KT. The effects of different experimental designs on parameter estimation in the kinetics of a reversible chemical reaction. Chemometrics and Intelligent Laboratory Systems 2000;52:155–66.
[23] Miettinen K. Nonlinear multiobjective optimization. Norwell, MA: Kluwer; 1999.
[24] Scheffé H. Simplex-centroid designs for experiments with mixtures. Journal of Royal Statistical Society Series B 1963;25:235–63.
[25] Cornell JA. Experiments with mixtures: a review. Technometrics 1973; 15:437–55.
[26] Cornell JA. Experiments with mixtures, designs, models, and the analysis of mixture data. New York: Wiley; 1990.
[27] Fang KT, Wang Y. Number-theoretic methods in statistics. London: Chapman and Hall; 1994.
[28] Wang Y, Fang KT. Number theoretic method in applied statistics (II). Chinese Annals of Mathematics, Series B 1990;11:384–94.
[29] Fang KT, Lin DKJ. Uniform designs and their application in industry. Handbook of statistics, vol. 22; 2003. p. 131–70.
[30] Deb K, Thiele L, Laumanns M, Zitzler E. Scalable multiobjective optimization test problems. In: IEEE congress on evolutionary computation (CEC'02), Piscataway, NJ, vol. 1, 2002. p. 825–30.
[31] Huband S, Hingston P, Barone L, While L. A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation 2006;10(5):477–506.
[32] Deb K, Sinha A, Kukkonen S. 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, p. 1141–8.
[33] Deb K. Multi-objective optimization using evolutionary algorithms. New York: Wiley; 2001.
[34] Sato H, Aguirre HE, Tanaka K. Controlling dominance area of solutions and its impact on the performance of MOEAs. in: Proceedings of the 4th international conference on evolutionary multi-criterion optimization (EMO 2007). Lecture notes in computer science, vol. 4403. Berlin: Springer; 2007. p. 5–20.
[35] Zitzler E, Thiele L, Laumanns M, Fonseca CM, Fonseca VG. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation 2003;7(2):117–32.
[36] Zitzler E, Thiele L. Multiple objective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation 1999;3(4):257–71.