Journal of Network Intelligence ©2016 ISSN 2414-8105(Online)
Taiwan Ubiquitous Information Volume **2**, Number **1**, February **2017**

A New Evolutionary Algorithm Based on Uniform and Contraction for Many-objective Optimization

Dai Cai

College of Computer Science
Shaanxi Normal University
Xi'an, 710062, China
daicai8403@hotmail.com

Xiujuan Lei

College of Computer Science
Shaanxi Normal University
Xi'an, 710062, China

Received September 2016; revised February 2017

Abstract. *For many-objective optimization problems, how to get a set of solutions with good convergence and diversity is a difficult and challenging work. To achieve the goal, a new evolutionary algorithm based on uniform and contraction is proposed. The uniform design method is utilized to generate the weight vectors. The weight vectors uniformly distribute over the design space and the size of the weight vectors neither nonlinearly with the number of objectives nor considers a formulaic setting. Moreover, a sub-population strategy is used to enhance the local search ability to improve the convergence. The proposed algorithm adopts a ranking method to contract the non-dominance area to determine the best solution of each sub-population. The comparison with the several existing well-known algorithms: NSGAII, MOEA/D and HypE, on six benchmark functions with 5 to 25 objectives, indicate that the proposed algorithm is able to obtain more accurate Pareto front with better diversity.*

Keywords: Many-objective optimization problems, Decomposition, Uniform design, Contraction.

1. **Introduction.** Because there are many problems with several optimization objectives or criteria in the real world, research on multi-objective optimization problems (MOP) becomes one of the hottest topics of intelligent computation. Among various multi-objective optimization algorithms, multi-objective evolutionary algorithms (MOEAs), which make use of the strategy of the population evolution, are an effective method for solving MOPs. Many algorithms have been proposed in the last twenty years [1-6]. However, most of these algorithms are mainly designed for two or three objectives and are based on Pareto dominance. For many-objective optimization problems (the number of objectives is more than 4), the experimental studies have shown that the existing MOEAs based on Pareto dominance have poor ability to solve the many-objective problems [7]. One of the main reasons is the deterioration of the search ability of these MOEAs. The deterioration occurs due to the increase of the number of objectives such that in the early stages of evolution there will be a lot of non-dominance solutions. Consequently, the Pareto dominance-based fitness evaluation cannot generate any selection pressure toward the Pareto front (PF). Moreover, when the number of objectives increases, the dimensions of PF and proportion of non-dominated solutions in the population are also increasing, which can lead to difficult to obtain a set of solutions which have good diversity and converge to the PF for a MOEA. Thus, how to effectively solve many-objective optimization problems has caused the attention of many scholars and has become the research hot topic.

Currently, there are mainly there kinds of categories to solve the many-objective problems. The first category utilizes an indicator function, such as the hypervolum [8-10], as the fitness function. This kind of algorithms is also referred to as IBEAs (indicator-based evolutionary algorithms), and their high search ability has been shown in the literature [11]. Bader and Zitzler [12] propose a fast hypervolume-based many-objective optimization algorithm (HypE) which uses Monte Carlo simulation to speed up approximate the exact hypervolume values. However, one of their main drawbacks is the computation time for the hypervolume calculation which exponentially increases with the number of objectives [13], and even if the hypervolume values are calculated by Monte Carlo approximations, its running time is more than 10 hours after 50000 objective function evaluations for seven-objective problems [14]. This limits the application of hypervolume indicator-based evolutionary algorithms to many-objective optimization problems.

A FTTX RoF-WDM wired/wireless transponder system is a configuration that provides both wirelThe second category takes advantages of solution ranking methods. Specifically, solution ranking methods are used to discriminate among solutions in order to enhance the selection pressure toward the PF, which make sure the solutions are able to converge to the PF. At present, numerous approaches have been proposed for ranking solutions when dealing with many-objective problems. Bentley and Wakefield [15] propose ranking composition methods which extract the separated fitness of every solution into a list of fitness values for each objective. Kokolo and Hajime [16] propose a relaxed form of dominance (RFD) to deal with what they called dominant resistant solutions, i.e., these solutions that are extremely inferior to others in at least one objective, but hardly-dominated. Farina and Amato [17] propose a dominance relation which takes into consideration the number of objectives where a solution is better, equal and worse than another solution. Sato et al. [18] propose a method to strength or weaken the selection process by expanding or contracting the solutions' dominance area (CDAS). When the dominance of solutions is contracted, the CDAS can guides the search process to converge to the true PF, and the CDAS has been used in some MOEAs to improve the convergence (e.g. [19-21]). However, when the CDAS improve the convergence, it is hard to maintain the diversity.

The third category utilizes the scalarizing functions to deal with the many-objective problem. According to the literatures [22-24], scalarizing function-based algorithms could better deal with many-objective problems than Pareto dominance-based algorithms. The main advantage of scalarizing function-based algorithms is the simplicity of their fitness evaluation which can be easily calculated for even the quantity of objectives is huge. The representative MOEA in this category is MOEA/D [25] (multi-objective evolutionary algorithm based on decomposition), which works well on a wide range of multi-objective problems with many objectives, discrete decision variables and complicated Pareto sets [26-28]. However, for many-objective problems, the decomposition-based evolutionary algorithms will face the following two problems: 1) how to generate a set of uniformly distributed weight vectors to uniformly divide the objective space; 2) how to make each sub-problem to converge to the Pareto optimal solutions. In MOEA/D [25], the uniformity of the used weighted vectors determines the uniformity of the obtained non-dominated optimal solutions, however, the used weighted vectors in MOEA/D are not very uniform and the size N of these weighted vectors should satisfy the restriction $N = C_{H+m-1}^{m}$. Thus N cannot be freely assigned and it will increase nonlinearly with the increase of m (where m is the number of objectives and H is an integer.), which restricts the application of MOEA/D to a certain extent in multi-objective optimization problems. Therefore, for many-objective problems, how to set weight vectors is a very difficult but critical task, and it is necessary to consider an efficient and simple method to produce the weight vectors [29]. Hughes [30] also considers a similar idea to set the weight vectors. In addition, the angles between the weight vectors will increase with the number of objectives (too many weight vector can increase the amount of calculation), which may lead to weak the correlation between the sub-problems and may cause the convergence performance of decomposition-based algorithms to degrade.

The ranking methods can improve the convergence and the decomposition can maintain the diversity. Though these two methods are commonly used approaches in many-objective evolutionary algorithms, but, to our knowledge, they have mostly been used separately. In this paper, an evolutionary algorithm based on decomposition and a ranking method (EA/UC) is proposed to solve the many-objective optimization problems. Firstly, EA/UC adopts a uniform design method to generate a set of points which are uniformly distributed on a unit sphere, and these points are the weight vectors. The number of weight vectors only needs to larger than the number of objectives (In general, the number of the weight vectors is greater than the number of objectives.). Secondly, a sup-population strategy is used to enhance the local search ability of the proposed algorithm. We make each sub-problem have a sub-population, each sub-problem use the information provided by its corresponding sub-population to improve the convergence performance. Then, a selection strategy based on decomposition and the sup-population strategy is design to carry out the global search and local search. Thirdly, we have designed a new ranking method based on contraction of the non-dominance area. The ranking method is used in EA/UC to compare solutions, which will provide stronger selection pressure toward the PF than Pareto dominance. Fourthly, an update strategy based on decomposition is proposed to maintain the diversity. We firstly decompose the objective space into a set of sub-regions based on a set of weight vectors and then make each sub-region have more than one solution to the maximum extent, finally any solution of each sub-region is replace by a solution which must locates in the sub-region. Moreover, the experiments demonstrate that EA/UC can significantly outperform MOEA/D, NSGAII-CDAS (NSGAII based on CDAS) and HypE on a set of test instances.

The rest of this paper is organized as follows: Section 2 introduces the main concepts of the multi-objective optimization; Section 3 presents a new many-objective evolutionary algorithm; while Section 4 shows the experiment results of the proposed algorithm and the related analysis; finally, Section 5 draws the conclusions and proposes the future work.

2. Multi-objective Optimization.

A multi-objective optimization problem (MOP) can be described as follows [31]:

$$\begin{cases} \min F(x) = (f_1(x), f_2(x), ..., f_m(x)), & \text{s.t. } g_i(x) \leq 0, i = 1, 2, ..., \beta \\ h_j(x) = 0, & j = 1, 2, ..., p \end{cases} \tag{1}$$

where $x = (x_1, ..., x_n) \in X \subset R^n$ is called decision variable and X is n-dimensional decision space. $f_i(x)(i = 1, ..., m)$ is the i^{th} objective to be minimized, $g_i(x)(i = 1, 2, ..., \beta)$ defines the i^{th} inequality constraint and $h_j(x)(j = 1, 2, ..., p)$ defines the j^{th} equality constraint. Furthermore, all the constraints determine the set of feasible solutions which are denoted by Ω. To be specific, we try to find a feasible solution $x \in \Omega$ minimizing each objective function $f_i(x)(i = 1, ..., m)$ in F. In the following, four important definitions [32] for multi-objective problems are given.

Definition 2.1. *(Pareto dominance): Pareto dominance between solutions $x, z \in \Omega$ is defined as follow.*

If

$$\begin{array}{l} \forall i \in 1, 2, ..., m, \; f_i(x) \leq f_i(z) \\ \wedge \exists i \in 1, 2, ..., m, \; f_i(x) < f_i(z) \end{array} \tag{2}$$

are satisfied, x dominates (Pareto dominate)z (denoted $x \succ z$).

Definition 2.2. *(Pareto optimal): A solution vector x is said to be Pareto optimal with respect to Ω, if $\nexists z \in \Omega : z \succ x$.*

Definition 2.3. *(Pareto optimal set (PS)): The set of Pareto optimal solutions (PS) is defined as:*

$$PS = \{x \in \Omega \mid \nexists z \in \Omega : z \succ x\} \tag{3}$$

Definition 2.4. *(Pareto front): The Pareto optimal front (PF) is defined as:*

$$PF = \{F(x) \mid x \in PS\} \tag{4}$$

3. A New Evolution Algorithm Based on Uniform and Contraction.

The proposed algorithm (EA/UC) is specifically designed to obtain a set of solutions with good diversity and convergence. EA/UC consists four parts, i.e. space and population decomposition, crossover operations, selection strategy and update strategy. Space and population decomposition and update strategy are used to maintain the diversity of the obtained solutions. The crossover operations are used to search the decision space. The selection strategy can help the obtained solutions to converge to the true PF. In this section, these four parts are introduced in the following subsection.

3.1. Uniform Design.

Uniform design which is proposed by Fang and Wang [33] represents a combination of number theory and numerical analysis. The uniform design method has been successfully implemented in science, engineering and industries [34-39]. The literature [40] has shown that the uniform design performs better at estimating nonlinear problems than other designs. The foremost goal of the uniform design method is to find a set of points that are uniformly distributed over the design space, and the set has a small discrepancy. The uniform design method has been used in MOEAs to generate the weight vectors, for example, Leung and Wang [38] use the uniform design method to generate multiple weight vectors which are uniformly scattered points on a unit hypercube and each point on the unit hypercube yields a weight vector; the literature [28] uses the uniform design method to yield weight vectors and design a uniform design multi-objective evolutionary algorithm based on decomposition for many-objective optimization problems, but the algorithm only tests five-objective problems.

In this section, two uniform design methods are briefly introduced. The main goal of a uniform design is to sample a small set of points from a given closed and bounded set $G \subset R^M$ such that the sampled points are uniformly scattered on G. In the following, we consider only two specific cases of G and describe the main features of uniform design. For more details, we refer to [33].

We first explain the meaning of uniformly scattered points on the set G, which will be done with G being an m-dimensional unit hypercube,

$$C = \{(\theta_1, \theta_2, ..., \theta_M) \mid 0 \le \theta_i \le 1, i = 1, ..., M\} \tag{5}$$

For any given point in C, say $r = (r_1, r_2, ..., r_M)$, a hyper-rectangle between 0 and r, denoted by $C(r)$, is defined by

$$C(r) = \{(\theta_1, \theta_2, ..., \theta_M) \mid 0 \le \theta_i \le r_i, i = 1, ..., M\} \tag{6}$$

For a given set of q points in C, suppose that $q(e)$ of these points fall in the hyper-rectangle $C(e)$, where $e = (e_1, e_2, ..., e_M)$. Then the fraction of the points falling in the hyper-rectangle $C(e)$ is given by $q(e)/q$. Since the volume of the unit hypercube is 1, hence, the fraction of volume of this hyper-rectangle is given by $e_1 * e_2 * ... * e_M$. The uniform design on C is then defined as determining q points in C such that the following discrepancy is minimized:

$$\underset{e \in C}{sup} \left| \frac{q(e)}{q} - e_1 e_2 ... e_M \right| \tag{7}$$

Table 1. The corresponding values of parameter δ for different values of q and M

q	M	δ
5	2-4	2
7	2-6	3
11	2-10	3
13	2	5
	3	4
	4-12	6
17	2-16	10
19	2-3	8
	4-18	14
23	2,13-14,20-22	7
	8-12	15
	3-7,15-19	17
29	2	12
	3	9
	4-7	16
	8-12,16-24	8
	13-15	14
	25-28	18
31	2,5-12,20-30	12
	3-4,13-19	22
200	5	163
	10	143
	15	51
	20,25	101

3.1.1. *Uniform Design on* $[l, u]$ *and* C. Let $l = (l_1, ..., l_M)$ and $u = (u_1, ..., u_M)$ be two points of R^M, where $l_i \le u_i, i = 1, ..., M$.
Denote

$$[l, u] = \{(\theta_1, \theta_2, ..., \theta_M) \mid l_i \leq u_i, i = 1, ..., M\} \tag{8}$$

Finding a set of exactly uniformly scattered points on $[l, u]$ and C is, in general, very difficult, but there are some efficient methods to look for a set of well approximately uniformly scattered points on $[l, u]$ and C. One of the simple and efficient methods is the Good-Lattice-Point method (GLP) [33], which generates a set of q uniformly scattered points on C, denoted by $C(q, M)$, and a set of q uniformly scattered points on $[l, u]$, denoted by $O(q, M)$, respectively, as follows:

Given q and M, determine a number δ (Table I lists the vales of δ for different values of q and M). Generate a $q \times M$ integer matrix called uniform array denoted by

$$G(q, M) = [G_{ij}]_{q\times M}, \text{ where } G_{ij} = (i\delta^{\ (j-1)} mod q) + 1,\ i = 1\ q,\ j = 1\ M \tag{9}$$

Each row of matrix $G(q, M)$ defines a point $C_i = (c_{i1}, c_{i2}, ..., c_{iM})$ of $C(q, M)$ and a point $O_i = (o_{i1}, o_{i2}, ..., o_{iM})$ of $O(q, M)$ by

$$c_{ij} = \frac{2G_{ij} - 1}{2q},\ o_{ij} = l_j + c_{ij}(u_j - l_j),\ i = 1\ q,\ j = 1\ M \tag{10}$$

$C(q, M)$ and $O(q, M)$ are given by

$$C(q, M) = \{C_i \mid i = 1\ q\} \text{ and } O(q, M) = \{O_i \mid i = 1\ q\} \tag{11}$$

3.1.2. *Uniform Design on a Unite Sphere $U(m)$ in R^m.* Let

$$U(m) = \{(f_1, ..., f_m) \mid f_1^2 + ... + f_m^2 = 1,\ f_i \geq 0,\ i = 1\ m\} \tag{12}$$

denote the nonnegative hyper-quadrant of the unit sphere in the objective space. The main idea of uniform design on $U(m)$ is mapping q points approximately uniformly distributed on an $(m-1)$-dimensional unit hyper-rectangle C to q points approximately uniformly distributed on $U(m)$, or mapping q points approximately uniformly distributed on a subset of C to q points uniformly distributed on a subset of $U(m)$. A set of q points uniformly distributed on $U(m)$, denoted by $D(q, m)$, can be generated according to the following steps:

Generate by the formula (11) a set of q uniformly distributed points on $C = [0, 1]^{(m-1)}$ or on a subset of C and denote this set of points by $C(q, m-1) = \{C_i = (c_{i1}, c_{i2}, ..., c_{(im-1)}), i = 1\ q\}$. Each row of matrix $C(q, m-1)$ defines a point $D_i = (d_{i1}, d_{i2}, ..., d_{im})$ of $D(q, m)$ by:

$$d_{ij} = \left\{ \begin{array}{ll} \sin(0.5c_{i1}\pi), & \text{if } j = m \\ \prod_{s=1}^{m-1} \cos(0.5c_{is}\pi), & \text{if } j = 1 \\ \sin(0.5c_{im-j+1}\pi) \prod_{s=1}^{m-j} \cos(0.5c_{is}\pi), & \text{if } 2 \leq j \leq m-1 \end{array} \right. \tag{13}$$

Then $D(q, m) = \{D_i = (d_{i1}, d_{i2}, ..., d_{im}), i = 1\ q\}$ is a set of q points uniformly distributed on $U(m)$.

Since the computation time and complexity for the discrepancy of a set of weight vectors (Eq. (7)) exponentially increase with the number of objectives, which lead to the optimal value of δ is difficult to determine a large number of objectives (i.e., $m \geq 20$). In this paper, our goal is to find a set of weight vectors uniformly distributed on a nonnegative hyper-quadrant of a unit sphere, so the optimal value of δ should be determined by the discrepancy of $D(q, m)$ not $G(q, M)$. Moreover, note that the set of weight vectors are on the nonnegative hyper-quadrant of the unit sphere, so the discrepancy of $D(q, m)$ can be replaced by inverted generational distance (IGD [41]) of $D(q, m)$. Let P^* be a set of uniformly distributed points on $U(m)$, the inverted generational distance from P^* to $D(q, m)$ is defined as:

$$IGD\big(P^*, D(q, m)\big) = \frac{\sum_{v \in P^*} d(v, D(q, m))}{|P^*|} \tag{14}$$

where $d(v, D(q, m))$ is the minimum Euclidean distance between v and the points in $D(q, m)$. If $|P^*|$ is large enough to represent the $U(m)$ very well, $IGD(P^*, D(q, m))$ could measure both the diversity and convergence of $D(q, m)$ in a sense. Because the points of $D(q, m)$ are on the $U(m)$, so the smaller value of $IGD(P^*, D(q, m))$ is, the better the diversity of $D(q, m)$ will be. Thus, the IGD can simply and well measure the diversity of $D(q, m)$ such that the optimal value of δ can be readily determined here.

Note that the optimal value of δ can be readily determined in the uniform design method such that a set of uniformly distributed weight vectors can be easily generated. Moreover, the number of weight vectors only needs to larger than the number of objectives. In general, the number of the weight vectors is larger than the number of objectives. Hence, the uniform design method can easily generate a set of uniformly distributed weight vectors and its size can get any number greater than the number of objectives.

3.2. **Space and Population Decomposition.** The objective space Ω of a MOP is decomposed into a set of sub-regions $\Omega_1, \Omega_2, ..., \Omega_N$ based on a set of weight vectors and a sub-population strategy is used, so obtained solutions are classified to ensure each sub-region owns K solutions (K is the size of each sub-population, and there are N sub-populations in this work.). Specifically, for a given set of weight vectors $\gamma^1, \gamma^2, ..., \gamma^N$ (where N is the number of the weight vectors) which are generated by the uniform method of Subsection 3.1.2, the current obtained solutions POP and the objective space are classified by the following equations:

$$\begin{aligned} &P^i = \{x \mid x \in POP, \Delta(F(x), \gamma^i) = \max_{1 \leq j \leq N}\{\Delta(F(x), \gamma^j)\}\} \\ &\Delta(F(x), \gamma^i) = \frac{\gamma^i * (F(x)-Z)^T}{\|\gamma^i\| * \|F(x)-Z\|},\ i = 1.,,,.m \end{aligned} \tag{15}$$

$$\Omega_i = \{F(x) \mid x \in \Omega, \Delta(F(x), \gamma^i) = \max_{1 \leq j \leq N}\{\Delta(F(x), \gamma^i)\}\} \tag{16}$$

where $Z = (Z_1, ..., Z_m)$ is a reference point with $Z_i = min\{f_i(x) \mid x \in \Omega\}$; $\Delta(F(x), \gamma^i)\}$ is the cosine of the angle between γ^i and $F(x) - Z$. The solution set POP is divided into N classes $P^1, P^2, ?, P^N$ by Eq. (15) and the objective space Ω is divided into N sub-regions $\Omega_1, ..., \Omega_N$ by Eq. (16). If the number of $P^i (1 \leq i \leq N)$ is smaller than K, some solutions are randomly selected from POP and put into P^i, which makes each sub-region have more than K solutions so that the diversity of obtained solutions is improved and useful information is provided to optimize each sub-region.

Note that above decomposition method has the following properties:
1) The PFs of all these sub-regions constitute the PF of problem (1)
2) Since the PS of each sub-region is just a small part of the PS of problem (1), it could be close to linear even when the whole PS of problem (1) has a nonlinear geometric shape. Therefore, in terms of PS shapes, Eq. (15) and (16) make problem (1) simpler than before.
3) This classification (decomposition) method does not require any aggregation methods, and it just requires the user to choose a set of weight vectors. Therefore, to some extent, it requires little computation.

3.3. **New Method Based on Contraction of the Non-dominance Area.** We have proposed a ranking method based on the contraction of the non-dominance area [50] to enhance the selection pressure toward the PF for many-objective optimization problems. The proposed method controls the levels of contraction or expansion of the dominance area of solutions by modifying the fitness value of a solution for each objective function. After modifying the fitness values of each solution, the dominance relation changes that originally are non-dominated become dominated by others. The modification of the fitness value of a solution x is defined by:

$$f_i'(x) = r(max(\sin(\omega_i), \cos(\omega_i)))^H (i = 1, ..., m) \tag{17}$$

where $r = \|F(x)\|$, $\cos(\omega_i) = (f_i(x))/\|F(x)\|$, $H > 0$. If $(f_1'(x), ..., f_m'(x))$ dominates $(f_1'(z), ..., f_m'(z))$, the solution x is better than the solution z; if $(f_1'(x), ..., f_m'(x))$ and $(f_1'(z), ..., f_m'(z))$ do not dominate each other, the solutions x and z also do not dominate each other.

H is a user-defined parameter which allows to control the dominance area of the solution x along the i^{th} dimension. The size of the non-dominance area of the solution x will becomes large (small) when H becomes small (large). When the non-dominance area of a solution is contracted, the number of PFs will decrease and ranking of solutions by non-dominance will become coarser, which can enhance the selection pressure.

3.4. **Update Strategy based Decomposition for POP.** In this section, a update strategy based decomposition and a sub-population strategy is proposed to maintain the diversity of obtained solutions and improve the convergence. After POP and the objective space Ω are classified, each weight vector has a set of solutions which is the sub-population of the corresponding sub-objective space. Then the ranking method in Section 3 is used to rank the solutions of each sub-population. And how to determine the best solution of each sub-population is firstly introduced. For each sub-region Ω_i of POP (each sub-region corresponding to a sub-population), the details of determining the best solution of each sub-population are as follows
1) If the sub-region Ω_i is empty, the solution whose objective vector has the smallest angle to the weight vector γ^i among the P^i is defined as the current best solution of the sub-population.
2) If the sub-region is not empty, there are two situations. If there is only one solution in this sub-region,

the solution is the current best solution of the P^i. If there are two or more solutions in this sub-region, these solutions are firstly ranked by the ranking method. Then the solution of the first front whose objective vector has the smallest angle to the weight vector γ^i corresponding to this sub-region is defined as the current best solution of the sub-population.

Note that, if a sub-population does not have K solutions, some solutions are randomly selected from POP to put the sub-populations such that each sub-population has K solutions, and the best solution is stored as the K^{th} solution. Thus, all $N \times K$ solutions can be denoted as $(x_1, ..., x_K, ..., x_{(N-1)\times K+1}, ..., x_{N\times K})$, where $(x_{(i-1)\times K+1}, ..., x_{i\times K})$ are solutions of i^{th} sub-population and $x_{i\times K}$ is the best solution of the sub-population. The update strategy for the $N \times K$ solutions is introduced as follows. For a new solution y which belong to the sub-region Ω_i, there are two cases:

1) If some solutions of the sub-region Ω_i do not belong to the sub-region, a solution which does not belong to the sub-region Ω_i among the i^{th} sub-population is deleted from the sub-population and y is put into the sub-population, and newly determine the best solution of the sub-population.

2) If all solutions of the sub-region Ω_i belong to the sub-region and the new solution y is better than the solution $x_{K\times i}$ under the strategy which is used to determine the best solution of each sub-population, then a solution is randomly selected from $\{x_{(i-1)\times K+1}, ..., x_{(i-1)\times K+K-1}\}$ and replaced by $x_{K\times i}$, and $x_{K\times i}$ is replaced by y.

It can be seen that the update strategy has the following properties: 1) It makes each sub-region have K solutions. This can improve the diversity of the solutions of POP in objective space. 2). If both the non-dominated and dominated solutions exist in a sub-region, a non-dominated solution is chosen and kept. This can help solutions to converge to PF fast. 3) The objective vector of the best solution has the smallest angle to the weight vector of the corresponding sub-region. This can make the solutions in POP distribute relatively evenly in objective space. 4) The new solution only compares with the best solution of the corresponding sub-population, which reduce the consumption of computation. 5) After the best solution of a sub-population is updated, some a solution of the sub-population is also updated, which improve the quality of solutions in the sub-population to effectively optimize the best solution of the sub-population.

3.5. **Crossover Operations.**

In this algorithm, a differential evolution (DE) [42] operator and a simplified quadratic approximation (SQA) are used as the crossover operations. For three given solutions x^{r1}, x^{r2}, x^{r3}, they generate a new solution by the following formula:

$$x_i^{new} = \begin{cases} x_i^{r1} + F(x_i^{r2} - x_i^{r3}), & ifrand(0,1) < CR \\ x_i^{r1}, & otherwise \end{cases} \tag{18}$$

where $F \in [0, 2]$ is a scale factor which controls the length of the exploration vector $(x^{r2} - x^{r3})$; CR is a constant value namely crossover rate; $i = 1, ..., n$ and x_i^{r1} indicates the i^{th} component of x^{r1}. The main goal of DE is to perform the global search in this algorithm.

SQA [43-44] is a simple and effective search method and does not require the differential coefficient of the objective function. SQA is used in [43] to perform the global search, while [44] uses SQA as the local search operator to improve the performance of Price algorithm. In this algorithm, the main goal of SQA is to carry out the local search. To implement SQA, a MOP is transformed into a single objective problem by a weight vector and an aggregate function. In this work, the aggregate function is described as following:

$$\underset{x\in\Omega}{minimize}\; g^{TE}(x \mid \lambda, Z^*) = \max_{1\le j\le m} \{|f_j(x) - z_j^*|/\lambda_j\} \tag{19}$$

where $\lambda = (\lambda_1, ..., \lambda_m)$ is a weight vector and Z^* be the reference point, i.e., $Z^* = (z_1^*, ..., z_m^*)$, $z_1^* = min\{f_i(x) \mid x \in \Omega\}$ for each $i = 1, ..., m$. The Eq. (19) is a variant of Tchebycheff approach. If the optimal solution x_i^* of (19) is the Pareto optimal solution of (1), then $(f_1(x_i^*) - z_1^*) : ... : (f_m(x_i^*) - z_m^*) = \lambda_1 : ... : \lambda_m$. EA/UC uses (21) as the aggregation function to convert a MOP into a number of scalar optimization problems.

For three given solutions x^1, x^2, x^3, weight vector $\lambda = (\lambda_1, ..., \lambda_m)$ and the reference point Z^*, and assume $g^{TE}(x^1 \mid \lambda, Z^*) = g^1 < g^{TE}(x^2 \mid \lambda, Z^*) = g^2 < g^{TE}(x^3 \mid \lambda, Z^*) = g^3$, the new solution $x^{new} = (x_1^{new}, ..., x_n^{new})$ is generated by the following formula:

$$x_i^{new} = \frac{0.5\{[(x_i^1)^2 - (x_i^3)^2]g^2 + [(x_i^3)^2 - (x_i^2)^2]g^1 + [(x_x^2)^2 - x(i^1)^2]g^3\}}{(x_i^1 - x_i^3)g^2 + (x_i^3 - x_i^2)g^1 + (x_i^2 - x_i^1)g^3}, i = 1, ..., n \tag{20}$$

3.6. **Selection Strategy.** The selection strategy has a great impact on the performance of local search and global search, thus an appropriate selection strategy can improve the performance of an algorithm. In this work, the selection strategy based on the decomposition and sub-population strategy is designed to achieve the goal. Firstly, compute the Euclidean distance between any two weight vectors and then work out the T closet weight vectors to each weight vector [27]. For each $i = 1, ..., N$, set $B(i) = K * i_1, ..., K * i_T$ where $\lambda^{i_1}, ..., \lambda^{i_T}$ are the T closet weight vectors to λ^i and K is sub-population size. Then set

$$P = \begin{cases} \{(i-1) * K + 1, ..., i * K\}, & \text{if } rand1 < p1 \\ \begin{cases} B(i), & \text{if } rand2 < J \\ \{K * 1, ..., K * N\}, & \text{otherwise} \end{cases}, & otherwise \end{cases} \tag{21}$$

where $rand1$ and $rand2$ are two random number and their scope is $[0, 1]$, $p1$ and J are two parameters. J is set to 0.9 as the same in [27]. When P is set, randomly select two indexes r_2 and r_3 from P. If two indexes r_2 and r_3 are selected from $\{(i-1) * K + 1, ..., i * K\}$, then generate a solution from x_{K*i}, x_{r_2} and x_{r_3} by Eq. (20), where the weight vector of the aggregate function (Eq. (19)) is λ^i; otherwise, generate a solution from x_{K*i}, x_{r_2} and x_{r_3} by Eq. (18).

If P is set to $\begin{cases} B(i), & \text{if } rand2 < J \\ \{K * 1, ..., K * N\}, & \text{otherwise} \end{cases}$, the local search and global search are carried out, however, for many-objective optimization problems, the performance of this local search may be poor, which is because the correlation between the sub-problems becomes weaken and the information provided by the neighbor of a sub-problems does not effectively optimize the sub-problem. If P is set to $\{(i-1) * K + 1, ..., i * K\}$, the local search is implemented to enhance the performance of local search and improve the convergence.

In this work, we adapt the following formula to set the value of $p1$:

$$p1 = \frac{0.9 * fZ}{\max fz} \tag{22}$$

where fz is the current total number of function evolutions and $maxfz$ is the maximal number of function evaluations. In the early stages of algorithm, the global search is mainly carried out, and more local search is performed with the increase of number of function evaluations.

3.7. **Steps of the Proposed Algorithm.** Based on all above, a new evolutionary algorithm based on decomposition and the ranking method (EA/UC) is proposed and the steps of algorithm EA/UC are as follows: Input:

N: the number of weight vectors (the sub-problems);
K: the size of the sub-population;
T: the number of weight vectors in the neighborhood of each weight vector, $0 < T < N$;
$\lambda^1, \lambda^2, ..., \lambda^N$: a set of N uniformly distributed weight vectors;

Output: Approximation to the PF: $\{F(x^K), F(x^{2*K}), ..., F(x^{N*K})\}$

Step 1. Initialization:
Step 1.1. Generate an initial population $x^1, x^2, ..., x^{N*K}$ randomly or by a problem-specific method.
Step 1.2. Initialize $Z = (z_1, ..., z_m)$ by a problem-specific method.
Step 1.3 Divide the initial population into N classes by Eq. (15) and each class has K solutions, and then determine the best solution of each class (sub-population) by modifying fitness function value of each objective via the ranking method and the strategy of Subsection 3.4.
Step 1.4. Compute the Euclidean distances between any two weight vectors and the work out the T closet weight vectors to each weight vector. For each $i = 1, ..., N$, set $B(i) = \{i_1, ..., i_T\}$, where $\lambda^{i_1}, ..., \lambda^{i_T}$ are the T closest weight vectors to λ^i.

Step 2. Update:
For $i = 1, ..., N$, do
Step 2.1. Reproduction: Randomly select two indexes r_2 and r_3 from P which is determined by the strategy of Subsection 3.6, then generate a set of new solution y from x_{K*i}, x_{r_2} and x_{r_3} by using the crossover operators.
Step 2.2. Mutation: Apply a mutation operator on y to produce y'. Step 2.3. Update of Z: For $k = 1, ..., m$, if $z_k < f_k(y')$, then set $z_k = f_k(y')$.
Step 2.4. The sub-region which the solution y' belongs to is determined by the Eq. (15). The corresponding sub-population is updated by the update strategy of Subsection 3.4.
End for.

Step 3. Stopping Criteria: If stopping criteria is satisfied, then stop and output $F(x^K), F(x^{2*K}), ..., F(x^{N*K})$; otherwise, go to Step 2.

4. **Experimental Study.**

In this section, EA/UC will be compared with three well known algorithms: multi-objective evolutionary algorithm based on decomposition (MOEA/D [27]), non-dominated sorting genetic algorithm II based on control the dominance area of solutions (NSGAII-CDAS) and fast hypervolume-based many-objective optimization algorithm (HypE) through experiments. The experiments are conducted on six widely used and challenging enough many-objective benchmark problems.

4.1. **Test Problems.**

Among six benchmark problems, four scalable test problems DTLZ1, DTLZ2, DTLZ3 and DTLZ4 of the DTLZ family [45] which is set of test problems often used in the analysis of MOEAs. They are selected for this study because they have the following important advantages: 1) they are easy to construct; 2) convergence and diversity are difficult to be controlled; 3) they can be easy scaled to any number of decision variables and objectives; 4) the global PF is known analytically and they have many local PFs. DTLZ1 has $(11^{n-m+1} - 1)$ local PFs and DTLZ3 has $(3^{n-m+1} - 1)$ local PFs [45]. The DTLZ1 and DTLZ3 problems can be used to investigate the global search ability of the algorithms and the ability to solve large numbers of objectives problems. All local PFs are parallel to the global PF. F1 and F2 with variable complicated PS shapes are used by Tan [28]. This class of functions with complicated PS shapes is proposed by Zhang [27] and is used as the test instances of the unconstrained MOEA competition in CEC 2009 [46]. It has been shown that these complicated PS shapes, as well as the geometrical shapes of the PF, can cause difficulty for MOEAs [47]. The F1 and F2 problems are used in order to investigate the ability of EA/UC to search the decision space.

4.2. **Parameter Settings.**

The experiments are carried out on a personal computer (Intel Xeon CPU 2.53GHz, 3.98G RAM). The solutions are all coded as real vectors. Polynomial mutation [48] operators and differential evolution (DE) [42] are applied directly to real vectors in three algorithms, i.e. EA/UC, MOEA/D and NSGAII-CDAS. The crossover rate and scaling factor in DE operator are set to 1.0 and 0.5, respectively. The aggregate function of MOEA/D is the Tchebycheff approach [27], and weight vectors are generated by using the uniform design method for MOEA/D and EA/UC. The number of the weight vectors in the neighborhood in MOEA/D and EA/UC is set to 20 for all test problems. The parameter of the ranking method H is set to 0.75. The parameter of CDAS is set to 0.25 for NSGAII-CDAS. The parameters of HypE are the same as [12]. For each algorithm, 20 independent runs are performed with population size of 200 for all these instances. The maximal number of function evaluations is set to 400000 for all test problems. The values of default parameters are the same as in the corresponding papers.

4.3. **Experimental Measures.**

In this paper, the following three performance metrics are used to compare the performance of the different algorithm quantitatively: generational distance (GD) [41], inverted generational distance (IGD) [41] and Wilcoxon Rank-Sum Test [49]. GD measures how far the known Pareto front is away from the true Pareto front. If GD is equal to 0, all points of the known PF belong to the true PF. GD allows us to observe whether the algorithm can converges to some region in the true PF. IGD measures how far the true PF is away from the known PF. If IGD is equal to 0, the known PF contains every point of the true PF. IGD shows whether points of the known PF are evenly distributed throughout the true PF. Here, GD and IGD indicators are used simultaneously to observe whether the solutions are distributed over the entire PF. In experiments, we randomly generated 100000 uniformly distributed optimal solutions which compose of the true PF for all problems. Wilcoxon Rank-Sum test [49] is used in the sense of statistics to compare the mean IGD and GD of the compared algorithms. It test whether the performance of EA/UC on each test problem is better ("+"), same ("="), or worse ("-") than/as that of the compared algorithms at a significance level of 0.05 by a two-tailed test.

4.4. **Comparisons of EA/UC with MOEA/D, NSGAII-CDAS and HypE.**

In this section, some simulation results are presented in Tables II and III on six test problems with 5-25 objectives respectively, and the comparisons are made which demonstrate the performance of EA/UC. The comparisons mainly focus on two aspects: 1) the convergence of the obtained solutions to the true Pareto optimal front; 2) the diversity of the non-dominated solutions.

Table II and Table III show the mean and standard deviation of the IGD metric and GD metric obtained by these four algorithms for these six test problems with 5-25 objectives respectively, and Wilcoxon test for pair wise comparisons of algorithms on each test instance are also shown in Tables II and III. DTLZ1-k represents that the number of objectives adopted in DTLZ1 is k.

It can be seen from Table II that, for the Wilcoxon test of the IGD metric, EA/UC outperforms MOEA/D, NSGAII-CDAS and HypE on these all test problems. From the Table II, we also can see that the mean values of IGD obtained by MOEA/D are much smaller than those obtained by NSGAII-CDAS and HypE for these all test problems, which indicates that for these test problems, MOEA/D can obtain solutions with better diversity than NSGAII-CDAS and HypE. However, the mean values of IGD obtained by EA/UC are much smaller than those obtained by MOEA/D, especially, e.g., for problem DTLZ3 with 10-25 objectives, the mean value of IGD obtained by EA/UC is smaller at least 70% than that obtained by MOEA/D. These indicate that for these test problems, the solutions obtained by EA/UC have better coverage than those obtained by other three algorithms and have good convergence. These also imply that the proposed update strategy based on decomposition and the sub-population is good at maintaining the diversity and the proposed algorithm has a good convergence.

We can see from Table III that, for the Wilcoxon test of the IGD metric, EA/UC outperforms MOEA/D on twenty-nine problems, outperforms NSGAII-CDAS on fifteen problems, and outperforms HypE on twenty-two problems, performs worse than NSGAII-CDAS on twelve problems and HypE on seven problems (a total of thirty problems). These results indicate EA/UC can obtain better convergence than MOEA/D, NSGAII-CDAS and HypE on most problems. From the table, we can also see that the mean values of GD obtained by EA/UC are smaller than those obtained by MOEA/D for these all problems, which indicate that EA/UC has better convergence performance than MOEA/D on these problems. For NSGAII-CDAS and HypE, though the mean values of GD obtained by them are much smaller than those obtained by EA/UC for some problems, but the mean values of IGD obtained by they are much larger than those obtained by EA/UC, which imply that solutions obtained by them may concentrate in some regions of the PF.

In summary, the comparisons of the simulation results of these four algorithms show that EA/UC is able to obtain a set of solutions with better diversity and good convergence, and EA/UC has a better performance than MOEA/D for all these problems.

where "+" means that EA/UC outperforms its competitor algorithm, "-" means that EA/UC is outperformed by its competitor algorithm, and "=" means that the competitor algorithm has the same performance as EA/UC.

4.5. **The Sensitivity Analysis to Parameters.** The parameter of the ranking method H, the sub-population size K and the number of neighbors T are three main control parameters of the proposed algorithm EA/UC. To study the sensitivity of the parameters in the proposed algorithm EA/UC, we have to test different settings of the parameter in the implementation of EA/UC.

"+" means that EA/UC outperforms its competitor algorithm, "-" means that EA/UC is outperformed by its competitor algorithm, and "=" means that the competitor algorithm has the same performance as EA/UC.

In the experiments of the sensitivities of the parameters (i.e., H, K and T), we have tested different settings of the parameters in the implementation of EA/UC on two typical problems DTLZ1-10 and DTLZ3-10 (they have many local PFs). In the experiments, we use different values of H from 0.2 to 15, use different values of K from 3 to 40, and use different the numbers of neighbors from 4 to 100, other parameters are same as those in Subsection 6.2, respectively.

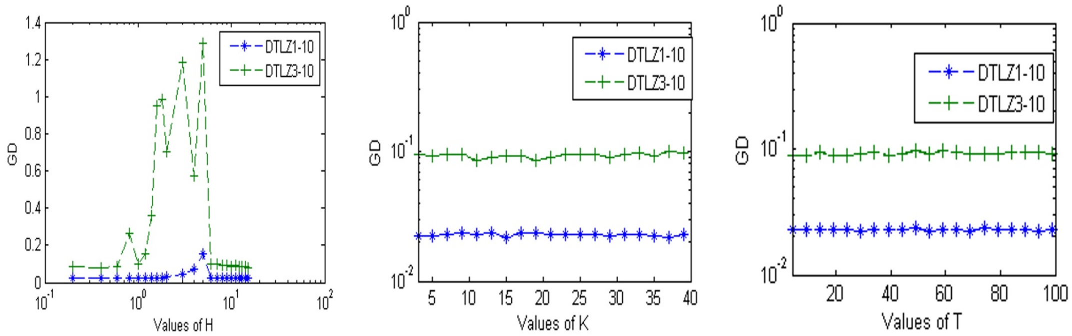
Fig. 1 and Fig. 2 respectively show the variations of the values of GD and IGD with the different values of H, K and T on two typical problems DTLZ1-10 and DTLZ3-10.

It can be seen from Fig. 1 and Fig. 2 that, for different values of H, the values of GD and IGD only have a little bit variation on DTLZ1-10, but have a big variation on DTLZ3-10 when the value of H is more than 1, which indicate that EA/UC with a small value of H (smaller than 1) will has a good performance, thus the value of H should be set to smaller than 1; for different values of K and T, the values of GD and IGD only have a little bit variation, which indicates that the proposed algorithm EA/UC is not sensitive to the two parameters. Moreover, the space complexity of EA/UC will be increases with the K and T, thus the values of K should be smaller than 10 and the values of T should be smaller than 20 to reduce the space complexity and keep the performance of the proposed algorithm.

5. **Conclusion.** In this paper, a new uniform evolutionary algorithm based on uniform and contraction is proposed to obtain a set of solutions with good convergence and diversity for many-objective optimization problems. In order to maintain the diversity of obtained solutions, an update strategy based on decomposition is designed. The objective space of a MOP is firstly divided into a set of sub-regions by a set of weight vectors which is generated by a uniform design. The uniform design method can generate a set of uniformly distributed weight vectors with the minimum value of IGD, and the number

Table 2. IGD obtained by EA/UC, MOEA/D, NSGAII-CDAS and HypE

Instance	EA/UC		MOEA/D		NSGAII-CDAS		HypE	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
DTLZ1-5	0.0417	0.0009	0.0590(+)	0.0045	0.1359(+)	0.0190	0.1741(+)	0.0453
DTLZ1-10	0.0409	0.0005	0.0830(+)	0.0060	0.1209(+)	0.0221	0.5271(+)	0.0358
DTLZ1-15	0.0607	0.0008	0.1044(+)	0.0116	0.1379(+)	0.0332	0.3113(+)	0.0258
DTLZ1-20	0.0632	0.0028	0.0981(+)	0.0075	0.1652(+)	0.0374	0.5570(+)	0.0368
DTLZ1-25	0.0617	0.0053	0.0975(+)	0.0066	0.1610(+)	0.0234	0.3892(+)	0.0248
DTLZ2-5	0.1458	0.0007	0.1678(+)	0.0120	1.1097(+)	0.2259	0.4192(+)	0.0616
DTLZ2-10	0.2008	0.0019	0.2880(+)	0.0257	0.9251(+)	0.2462	0.6492(+)	0.0414
DTLZ2-15	0.2321	0.0039	0.3292(+)	0.0214	1.0240(+)	0.2840	0.6593(+)	0.1015
DTLZ2-20	0.2517	0.0046	0.3300(+)	0.0461	1.1762(+)	0.1467	0.6827(+)	0.1164
DTLZ2-25	0.2519	0.0059	0.3458(+)	0.0449	1.2154(+)	0.1792	0.7398(+)	0.1731
DTLZ3-5	0.1331	0.0011	0.1569(+)	0.0028	1.0058(+)	0.2559	0.5007(+)	0.0564
DTLZ3-10	0.2028	0.0020	0.3561(+)	0.0372	0.9322(+)	0.2828	0.7398(+)	0.0365
DTLZ3-15	0.2276	0.0032	0.4614(+)	0.0701	0.9794(+)	0.3951	0.7525(+)	0.1753
DTLZ3-20	0.2466	0.0043	0.7186(+)	0.0302	0.5750(+)	0.1143	0.7506(+)	0.0984
DTLZ3-25	0.2542	0.0102	0.7471(+)	0.0046	0.7672(+)	0.2920	0.7459(+)	0.1230
DTLZ4-5	0.1456	0.0009	0.1659(+)	0.0080	1.1737(+)	0.2395	0.4807(+)	0.0271
DTLZ4-10	0.2017	0.0007	0.2929(+)	0.0256	1.1984(+)	0.2579	0.6494(+)	0.0605
DTLZ4-15	0.2308	0.0042	0.3518(+)	0.0297	1.2936(+)	0.1002	0.6548(+)	0.0893
DTLZ4-20	0.2546	0.0053	0.3279(+)	0.0181	1.0223(+)	0.3849	0.6987(+)	0.1149
DTLZ4-25	0.2502	0.0049	0.3397(+)	0.0324	0.9548(+)	0.2005	0.7120(+)	0.1708
F1-5	0.1445	0.0010	0.1634(+)	0.0109	1.1512(+)	0.1068	0.6995(+)	0.0049
F1-10	0.1994	0.0012	0.3357(+)	0.0308	1.3108(+)	0.1522	0.7069(+)	0.0139
F1-15	0.4455	0.0018	0.5640(+)	0.0785	0.7438(+)	0.0027	0.7532(+)	0.0273
F1-20	0.4633	0.0034	0.6691(+)	0.0240	0.7332(+)	0.0075	0.7800(+)	0.0479
F1-25	0.4616	0.0047	0.6926(+)	0.0174	0.7392(+)	0.0019	0.8923(+)	0.0482
F2-5	0.0734	0.0008	0.1080(+)	0.0027	0.4860(+)	0.1008	0.6324(+)	0.0097
F2-10	0.0860	0.0018	0.1808(+)	0.0227	0.5325(+)	0.0446	0.6450(+)	0.0034
F2-15	0.1156	0.0025	0.2902(+)	0.0452	0.4887(+)	0.0588	0.6512(+)	0.0283
F2-20	0.1259	0.0071	0.3658(+)	0.0207	0.4597(+)	0.0367	0.7360(+)	0.0695
F2-25	0.1480	0.0362	0.3756(+)	0.0203	0.5099(+)	0.0166	0.8719(+)	0.0318

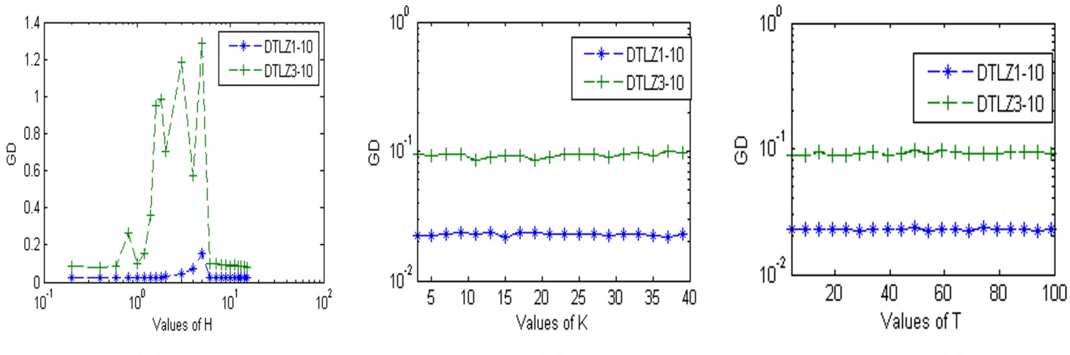


(a) (b) (c)

Figure 1. (a) Sensitivity of H on DTLZ1-10 and DTLZ3-10; (b) Sensitivity of K on DTLZ1-10 and DTLZ3-10; (c) Sensitivity of T on DTLZ1-10 and DTLZ3-10.

Table 3. GD obtained by EA/UC, MOEA/D, NSGAII-CEAS and HypE

Instance	EA/UC		MOEA/D		NSGAII-CDAS		HypE	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
DTLZ1-5	0.033	0.0006	0.0396(=)	0.0013	0.0291(−)	0.0025	0.0356(=)	0.0089
DTLZ1-10	0.0232	0.0004	0.0994(+)	0.0047	0.0201(=)	0.0069	0.0263(+)	0.0097
DTLZ1-15	0.0403	0.0005	0.1773(+)	0.0058	0.0180(−)	0.0052	0.0455(+)	0.0078
DTLZ1-20	0.0443	0.0009	0.1849(+)	0.0212	0.0167(−)	0.0058	0.0592(+)	0.0134
DTLZ1-25	0.0653	0.0025	0.1734(+)	0.0108	0.0190(−)	0.0056	0.0273(−)	0.0248
DTLZ2-5	0.029	0.0009	0.0444(+)	0.0014	0.0493(+)	0.0481	0.0651(+)	0.0444
DTLZ2-10	0.0879	0.0027	0.2932(+)	0.0092	0.0137(−)	0.0188	0.2134(+)	0.069
DTLZ2-15	0.1184	0.0028	0.5304(+)	0.0545	0.1070(=)	0.2271	0.6325(+)	0.0807
DTLZ2-20	0.1732	0.0065	0.5483(+)	0.0662	0.2360(+)	0.4678	0.7014(+)	0.0715
DTLZ2-25	0.1749	0.0039	0.5609(+)	0.0939	0.4258(+)	0.6205	0.8418(+)	0.0926
DTLZ3-5	0.0466	0.0012	0.0656(+)	0.0017	0.0000(−)	0	0.0970(+)	0.0456
DTLZ3-10	0.0922	0.0036	0.3127(+)	0.0175	0.2368(+)	0..3386	0.3644(+)	0.0714
DTLZ3-15	0.1334	0.0077	0.5794(+)	0.0398	0.3618(+)	0.4955	0.9975(+)	0.0954
DTLZ3-20	0.2005	0.0237	0.9208(+)	0.0661	0.0004(−)	0.0002	0.9948(+)	0.0943
DTLZ3-25	0.1797	0.0045	1.1308(+)	0.0351	0.0216(−)	0.047	0.9324(+)	0.1026
DTLZ4-5	0.0294	0.0006	0.0446(+)	0.0028	0.0880(+)	0.0552	0.0679(+)	0.0541
DTLZ4-10	0.0837	0.0041	0.2990(+)	0.0084	0.2199(+)	0.3327	0.2829(+)	0.0704
DTLZ4-15	0.1179	0.0071	0.5544(+)	0.025	0.4684(+)	0.5673	0.6642(+)	0.0986
DTLZ4-20	0.1688	0.0066	0.5432(+)	0.0453	0.3580(+)	0.5138	0.7964(+)	0.0871
DTLZ4-25	0.1765	0.0055	0.5622(+)	0.0564	0.0114(−)	0.0214	0.8452(+)	0.1077
F1-5	0.0516	0.0016	0.0804(+)	0.0022	0.0903(+)	0.0767	0.0282(−)	0.0341
F1-10	0.0982	0.0051	0.3120(+)	0.0091	0.3670(+)	0.2803	0.0387(−)	0.0815
F1-15	0.508	0.0115	0.6411(+)	0.0465	0.0014(−)	0.0006	0.0531(−)	0.0906
F1-20	0.609	0.0547	0.9512(+)	0.031	0.0050(−)	0.002	0.2194(−)	0.0127
F1-25	0.5569	0.0503	1.1630(+)	0.0195	0.0041(−)	0.0016	0.4526(−)	0.0913
F2-5	0.0266	0.0019	0.0410(+)	0.0028	0.0259(=)	0.0034	0.0375(−)	0.0425
F2-10	0.0488	0.0018	0.1868(+)	0.009	0.1492(+)	0.0137	0.3200(+)	0.0467
F2-15	0.1087	0.0019	0.4262(+)	0.0199	0.2144(+)	0.0131	0.3409(+)	0.0339
F2-20	0.1389	0.0038	0.5802(+)	0.0164	0.2278(+)	0.0207	0.5529(+)	0.0379
F2-25	0.1551	0.0236	0.6198(+)	0.0293	0.2288(+)	0.0091	0.6657(+)	0.0372



(a) (b) (c)

Figure 2. (a) Sensitivity of H on DTLZ1-10 and DTLZ3-10; (b) Sensitivity of K on DTLZ1-10 and DTLZ3-10; (c) Sensitivity of T on DTLZ1-10 and DTLZ3-10.

of weight vectors will not be controlled by the number of objectives with this design method. Thus, the method is suitable for large objectives problems. Secondly, this update strategy makes each sub-region have more than one solution to the maximum extent and makes a solution of a sub-region be replaced by fresh solution which must belong to the sub-region. In addition, in order to obtain a set of solutions with good convergence, the ranking method is used to enhance the selection pressure toward the true PF; thirdly, a sub-population strategy is used to increase the local search ability, the information provided by the sub-population can effectively optimize the corresponding sub-problem, then a simplified quadratic approximation is used to carry out the local search to improve the convergence. The experimental results show that, with these schemes, the proposed algorithm is able to search well in continuous domain, and achieve accurate Pareto sets and wide Pareto fronts efficiently. Moreover, compared with three well known algorithms MOEA/D, NSGAII-CDAS and HypE, simulation results showed that the proposed algorithm is able to find much better spread of solutions and these solutions have better convergence to the true PF. In the future, how to improve the efficiency of the algorithm is still to be studied.

6. **Acknowledgments.** This work was supported by National Natural Science Foundation of China (no. 61502290), China Postdoctoral Science Foundation (no. 2015M582606), Industrial Research Project of Science and Technology in Shaanxi Province (no.2015GY016), Fundamental Research Funds for the Central Universities (no. GK201603094) and Natural Science Basic Research Plan in Shaanxi Province of China (no. 2016JQ6045).

REFERENCES

[1] C.A. Coello Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*, Springer-Verlag/New York, Inc., Secaucus, NJ, USA, 2006.

[2] W. F. Gao, G. G. Yen, and S. Y. Liu, A Cluster-Based Differential Evolution With Self-Adaptive Strategy for Multimodal Optimization, em IEEE Transactions on Cybernetics, vol. 44(8), pp. 1214–1327, 2014.

[3] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, em IEEE Transactions on Evolutionary Computation, vol. 6(2), pp. 182–97, 2002.

[4] A. Nebro, J. Durillo, J. Garcia-Nieto, C.A.C. Coello, F. Luna, and E. Alba, SMPSO: a new PSO-based metaheuristic for multi-objective optimization, em Proc. of IEEE symposium on Computational Intelligence in Multi-criteria Decision-making, MCDM '09, 2009: pp. 66–73, 2009.

[5] N. Beume, B. Naujoks, and M. Emmerich, SMS-EMOA: multiobjective selection based on dominated hypervolume, em European Journal of Operational Research, vol. 180(3), pp. 1653–1669, 2007.

[6] A. M. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. F. Zhang, Mu;tiobjective evolutionary algorithms: A survey of the state of the art, em Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 32–49, 2011.

[7] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, Evolutionary many-objective optimization: a short review, em IEEE Congress on Evolutionary Computation, pp. 2419–2426, 2008.

[8] R. C. Purshouse and P. J. Fleming, On the evolutionary optimization of many conflicting objectives, em IEEE Trans. On volutionary Computation, vol. 11, no. 6, pp. 770–784, 2007.

[9] T. Friedrich, C. Horoba, and F. Neumann, Multiplicative approximations and the hypervolume indicator, em Proc. of 2009 Genetic and Evolutionary Computation Conference, pp. 571–578, 2009.

[10] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, Theory of the hypervolume indicator: Optimal g-distributions and the choice of the reference point, em Proc. of Foundations of Genetic Algorithm X, pp. 87–102, 2009.

[11] E. Zitzler and S. K?nzli, Indicator-based selection in multiobjective search, em Lecture Notes in Computer Science 3242: Parallel Problem Solving from Nature, PPSN VIII, Springer, Berlin, pp. 832–842, 2004.

[12] T. Wagner, N. Beume, and B. Naujoks. Pareto, aggregation and indicator-based methods in many-objective optimization, em Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007, Springer, Berlin, pp. 742–756, 2007.

[13] J. Bader and E. Zitzler. HypE, An algorithm for fast hypervolume- based many-objective optimization, em Evol. Comput., vol. 19, pp. 45–76, 2011.

[14] N. Beume, C. Fonseca, M. Lopez-Ibane, L. Paquete, and J. Vahrenhold, On the complexity of computing the hypervolume indicator, em IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, pp. 1075–1082, 2009.

[15] karl Bringmann, Tobias Friedrich, Christian Igel and Thomas Vob, Speeding up many-objective optimization by Monte Carlo approximations, em Artificial Intelligence, vol. 204, pp. 22–29, 2013.

[16] P. J. Bentley and J. P. Wakefield, Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms, em Proc. of World Conference on Soft Computing in Design and Manufacturing, pp. 231–240, 1997.

[17] I. Kokolo, K. Hajime, and K. Shigenobu, Failure of Pareto-based MOEAs: Does Nondominated Really Mean Near to Optimal?, em Proceedings of the Congress on Evolutionary Computation, vol. 2, pp. 957–962, 2001.

[18] M. Farina and P. Amato, A fuzzy definition of ”optimality” for many-criteria optimization problems, em IEEE Transactions on Systems, Man, and Cybernetics Part A-Systems and Humans, vol. 34, no. 3, pp. 315–326, 2004.

[19] H. Sato, H. E. Aguirre, and K. Tanaka, Controlling dominance area of solutions and its impact on the performance of mOEAs, em Proc. of EMO 2007. LNCS, vol. 4403, pp. 5–20, 2007.

[20] Andre B. de Carvalho and Aurora Pozo, Measuring the convergence and diversity of CDAS Multiobjective Particle Swarm Optimization Algorithms: A study of many-objective problems, em Neurocomputing, vol. 75, pp. 43–51, 2012.

[21] Andre B. de Carvalho and Aurora Pozo, The control of dominance area in particle swarm optimization algorithms for many-objective problems, em Proc. of 2010 Eleventh Brazilian Symposium on Neural Networks, pp. 140–145, 2010.

[22] Sato H, Aguirre H E and Tanaka K, Improved S-CDAS using crossover controlling the number of crossed genes for many-objective optimization, em Proc. of the 13th Annual Genetic and Evolutionary Computation Conference, pp. 753–760, 2011.

[23] E. J. Hughes, Evolutionary many-objective optimization: Many once or one many?, em Proc. of 2005 IEEE Congress on Evolutionary Computation, pp. 222–227, 2005.

[24] E. J. Hughes, MSOPS-II: A general-purpose many-objective optimizer, em Proc. of 2007 IEEE Congress on Evolutionary Computation, pp. 3944–3951, 2007.

[25] H. Ishibuchi and Y. Nojima, Optimization of scalarizing functions through evolutionary multiobjective optimization, em Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007, pp. 51–65, 2007.

[26] Q. Zhang and H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, em IEEE Trans. on Evolutionary Computation, vol. 11, no. 6, pp. 712–731, 2007.

[27] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations, em Proc. of 2009 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1820–1825, 2009.

[28] H. Li, and Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, em IEEE Trans. on Evolutionary Computation, vol. 13, no. 2, pp. 284–302, 2009.

[29] Y. Tan, Y. Jiao, H.g Li, and X. Wang, MOEA/D +uniform design: A new version of MOEA/D for optimization problems with many objectives, em Computers & Operations Research, vol. 6, no. 40, pp. 1648–1660, 2013.

[30] J. S. Thomas, J. W. Brian, and N. William, The design of analysis of computer experiments, em Berlin: Springer-Verlag, 2003.

[31] E. J. Hughes, Evolutionary many-objective optimisation: many once or one many, em Proc. of IEEE congress on evolutionary computation (CEC’05), vol. 1, pp. 222–7, Edinburgh, UK, 2005.

[32] D. A. Van Veldhuizen, Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations, em Ph.D. dissertation, Dept. Electr. Comput. Eng. Graduate School Eng., Air Force Instit. Technol., Wright- Patterson AFB, OH, May 1999.

[33] E. Zitzler, K. Deb, and L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, em Evol. Comput., vol. 8, no. 2, pp. 173–195, 2000.

[34] K. T. Fang and Y. Wang, Number-Theoretic Method in Statistics, em Chapman and Hall, London, 1994.

[35] Y. Z. Liang, K. T. Fang, and Q. S. Xu, Uniform design and its applications in chemistry and chemical engineering, em Chemometrics and Intelligent Laboratory Systems, vol. 58, pp. 43–57, 2001.

[36] L. Y. Chan and M. L. Lo, Quality improvement in the manufacture of liquid crystal displays using uniform design, em International Journal of Materials and Product Technology, vol. 20, pp. 127–42, 2004.

[37] Y. W. Leung and Y. P. Wang, Multiobjective programming using uniform design and genetic algorithm, em IEEE Trans. Systems, Man, and Cybernetics-Part C: Applications and Reviews, vol.30, no. 3, 2000.

[38] R. Li, D. Lin, and Y. Chen, Uniform design: design analysis and applications, em International Journal of Materials and Product Technology, vol. 20, pp. 101–114, 2004.

[39] G. Xu, J. Zhang, and Y. Tang, Level permutation method for constructing uniform designs under the wrap-around L-2-discrepancy, em Journal of Complexity, vol. 30, no. 1, pp. 46–53, 2014.

[40] R. Chen, Y. Hsu, Y. Hung, and W. Wang, Discrete particle swarm optimization for constructing uniform design on irregular regions, em Computational Statistics & Data Analysis, vol. 72, pp. 282–297, 2014.

[41] Q. S. Xu, Y. Z. Liang, and K. T. Fang, The effects of different experimental designs on parameter estimation in the kinetics of a reversible chemical reaction, em Chemometrics and Intelligent Laboratory Systems, vol. 52, pp. 155–166, 2000.

[42] C. A. Coello Coello and Nareli C. Cortes, Solving multiobjective optimization problems using an artificial immune system, em Genetic Programming and Evolvable Machines., vol. 6, no. 2, pp. 163–190, 2005.

[43] K. Price, R. M. Storn, and J. A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, em ser. Natural Computing. Secaucus, NJ, USA: Springer-Verlag, 2005.

[44] M. M. Ali, A. Torn, and S. Viitanen, A numerical comparison of some modified controlled random search algorithms, em Journal of Global Optimization, vol. 11, no. 4), pp. 377–385, 1997.

[45] Y. C. Jiao, C. Dang, Y. Leung, et al., A modification to the new version of the Price's algorithm for continuous global optimization problems, em Journal of Global Optimization, vol. 36, no. 4, pp.: 609–626, 2006.

[46] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, Scalable multi-objective optimization test problems, em Proc. of Congress on Evolutionary Computation (CEC 2002), pp. 825–830, 2002.

[47] Q. F. Zhang, W. Li, and H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, em Working Report CES-491. School of CS and EE, University of Essex, Colchester, UK; 2009.

[48] S. Huband, P. Hingston, L. Barone, and L. While, A review of multiobjective test problems and a scalable test problem toolkit, em IEEE Transactions on Evolutionary Computation, vol. 10, no. 5, pp. 477–506, 2006.

[49] K. Deb, em Multiobjective Optimization Using Evolutionary Algorithms, New York: Wiley, 2001.

[50] S. Robert, J. Torrie, and D. Dickey, Principles and Procedures of Statistics: A Biometrical Approach, em McGraw-Hill, New York, NY, 1997.

[51] C. Dai, and Y. Wang, A New Evolutionary Algorithm Based on Contraction Method for Many-objective Optimization Problems, em Applied Mathematics and Computation, vol. 245, pp. 192–205, 2014.