Algorithms **2015**, *8*, 157-176; doi:10.3390/a8020157

OPEN ACCESS

algorithms

ISSN 1999-4893

www.mdpi.com/journal/algorithms

Article

Multiobjective Cloud Particle Optimization Algorithm Based on Decomposition

Wei Li [1], Lei Wang [1,*], Qiaoyong Jiang [1], Xinhong Hei [2] and Bin Wang [1]

1. School of Computer Science and Engineering, Xi'an University of Technology, Xi'an 710048, China;
2. Shaanxi Key Laboratory for Network Computing and Security Technology, Xi'an 710048, China;

* Author to whom correspondence should be addressed; E-Mail: wangleeei@163.com; Tel.: +86-29-82312545; Fax: +86-29-82312545.

Academic Editor: George Karakostas

Received: 12 January 2015 / Accepted: 17 April 2015 / Published: 23 April 2015

Abstract: The multiobjective evolutionary algorithm based on decomposition (MOEA/D) has received attention from researchers in recent years. This paper presents a new multiobjective algorithm based on decomposition and the cloud model called multiobjective decomposition evolutionary algorithm based on Cloud Particle Differential Evolution (MOEA/D-CPDE). In the proposed method, the best solution found so far acts as a seed in each generation and evolves two individuals by cloud generator. A new individual is produced by updating the current individual with the position vector difference of these two individuals. The performance of the proposed algorithm is carried on 16 well-known multi-objective problems. The experimental results indicate that MOEA/D-CPDE is competitive.

Keywords: cloud particle differential evolution; MOEA/D; multiobjective optimization; evolutionary algorithm; cloud mutation

1. Introduction

Over the last two decades, multi-objective problems (MOPs) have received growing attention because of their wide applications [1–7]. Many multi-objective evolutionary algorithms (MOEAs) are designed to solve real-world applications. As is well known, however, objective functions of MOPs, often conflict with each other and improvement of one objective may lead to deterioration of another. Therefore, it is difficult to find an optimal solution that can satisfy all optimization objectives. To solve MOPs, researchers usually look for not only a single solution but a set of Pareto optimal solutions [8].

Many researchers apply Evolutionary Algorithms (EAs) for the MOPs because of the significant advantages of EAs [9–14]. The most popular MOEAs can be classified into three categories [15]: pareto dominance-based MOEAs, decomposition-based MOEAs, and indicator-based MOEAs. The distinctive features of methods based on pareto dominance, such as the strength Pareto evolutionary algorithm (SPEA)-II, Pareto Envelope-Based Selection Algorithm (PESA)-II and the Fast Non-Dominated Sorting Genetic Algorithm (NSGA-II), are the individual selection method, population diversity preservation mechanism based on fitness sharing, and elite reserved strategy. M. Tadahiko *et al.* [16] proposed a proportional weight specification method to solve multi-objective problems. Later, Zhang *et al.* proposed the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [17]. The algorithm decomposes a continuous MOP into several single-objective optimization problems. The population of each subproblem is composed of the best solution found so far. Each subproblem has several neighbors that are defined based on the distances between their corresponding weight vectors. MOEA/D aroused many researchers interest as soon as it was proposed. Several improvements on MOEA/D have been made recently [18–22]. At the same time, MOEA/D has been successfully applied to a number of application areas [15,23–25]. The classic methods of indicator-based are the multiobjective selection based on dominated hypervolume (SMS-EMOA) [26] and the algorithm for fast hypervolume-based many-objective optimization (HypE) [27].

Inspired by the cloud model, this paper presents a novel evolution approach. We employ cloud particles' evolution in the framework of MOEA based on decomposition (MOEA/D) called multiobjective decomposition evolutionary algorithm based on Cloud Particle Differential Evolution (MOEA/D-CPDE). At each generation, the individual that is the best solution found so far acts as a seed. The seed can be seen as the parent. Every seed evolves into two individuals by a cloud generator. Each of them not only has the characteristics of the seed but also has its own characteristics. Then, the algorithm updates the current individual with the position vector difference of these two individuals. The strategy can improve the diversity of the population and improve the performance of the algorithm.

The rest of the paper is organized as follows. Section 2 presents the related background including the framework of MOEA/D. Section 3 presents the cloud generator and MOEA/D-CPDE. Section 4 describes the benchmark problems, the parameter settings, performance metrics, and experimental results. Section 5 concludes the paper and points out some further studies.

2. Related Backgrounds

2.1. Problem Definition

A multiobjective optimization problem [28] (MOP) that is to optimize a set of functions synchronously can be described as follows:

$$\text{minimize} \quad \mathrm{F(x)} = (f_1(\mathrm{x}), f_2(\mathrm{x}), \cdots, f_m(\mathrm{x}))^T \tag{1}$$

subject to $\mathrm{x} \in \Omega$

where $\Omega \subset R^n$ is the feasible region of decision variable space, $\mathrm{x} = (x_1, x_2, \cdots, x_n)^T$ is the decision vector, and F: $\Omega \to R^m$ is a function that consists of *m* real-valued objective functions. R^m is called the objective space. If and only if $f_k(x_1) \le f_k(x_2)$, $1 \le k \le m$ and $\mathrm{F}(x_1) \ne \mathrm{F}(x_2)$, we say that x_1 dominates x_2. If there is no *x* which can dominate *x'*, we say that *x'* is a Pareto optimal solution. The set of all the Pareto optimal solutions is called the Pareto set (*PS*). The corresponding mapping of the Pareto-optimal set on the objective function space is called the Pareto optimal front *PF*.

2.2. Multiobjective Evolutionary Algorithm Based on Decomposition

MOEA/D [17] is a new framework for solving multi-objective optimization problems. It decomposes a multiobjective optimization problem into a number of scalar optimization subproblems and optimizes them simultaneously. In MOEA/D, the weighted sum approach, the Tchebycheff approach, and the boundary intersection approach are used to convert the problem of approximation of the *PF* into a number of scalar optimization problems. The description of MOEA/D is as follows.

Algorithm 1. The procedure of MOEA/D

Input: MOP, the multiobjective optimization problem; *NP*, population size; *GEN*, maximum generations; *FES*, the maximal number of function evaluations; *NB*, the number of weight vectors in the neighborhood of each weight vector;

Output: X, the non-dominated population; Approximation to the PF, $\{F(x^1), \ldots, F(x^N)\}$

```
Randomly initial population Xn
Compute the Euclidian distances between any two weight vectors and define neighborhood of each subproblem. For the ith subproblem, set B(i) = {i1,i2,…im}, where λ^i1, λ^i2, …, λ^im are the m closest weight vectors to λ^i
Initialize z=(z1,…, zm)^T by problem-specific method
while t < FES do
    for i = 1:NP do
        Produce y by DE-mutation strategies
        polynomial mutation
        Evaluation y
        for each j = 1 to m do
            if zj > fj(y) then zj = fj(y)
        endfor
        for each j∈B(i) do
            if g^te(y|λ^j,z) < g^te(x^j|λ^j,z)then
                x^j = y and F(x^j) = F(y)
            endif
        endfor
    endfor
endwhile
```

3. Frameworks of MOEA/D-CPDE

In MOEA/D, the new individual is produced by combining differential mutation with polynomial mutation, which is conducive to maintain the population diversity and improve the exploration ability. However, the convergence speed of MOEA/D may be slower. Inspired by the cloud model, a new evolution mechanism is proposed for solving the problem. Firstly, two individuals are produced by the parent with the cloud generator. Then, the search direction of the offspring is guided by the two individuals and differential operation. Finally, the new individual is updated by cloud particles mutation or polynomial mutation for improving the population diversity. In this way, the proposed algorithm can not only accelerate convergence speed but also improve the local exploitation.

3.1. Cloud Generator

The cloud model is proposed by Li [29] and has three parameters, including Expectation (E_x), Entropy (E_n) and Hyper-entropy (H_e), in which E_x is the expectation value of the distribution for all cloud particles in the domain, and E_n is the range of domain which can be accepted by linguistic values (qualitative concept). In other words, E_n is ambiguity and H_e is the dispersion degree of entropy (E_n). That is to say, H_e is the entropy of entropy. The cloud generator works according to (2) and (3), $\Phi(\mu, \delta)$ is the normal random variable that has an expectation μ and a variance δ; N is the number of cloud particle; C is the cloud particle generated by the cloud generator.

$$\mathrm{S} = \{s_i|\Phi(E_n, H_e), \ i=1,2,\cdots N\} \tag{2}$$

$$\mathrm{C} = \{c_i|\Phi(E_x, s_i), \ s_i \in \mathrm{S}, \ i=1,2,\cdots N\} \tag{3}$$

3.2. Cloud Particle Differential Evolution

First, the best solution found so far acts as a seed. Then, each seed produces two individuals (c_1 and c_2) by cloud generator. The indices r_1 and r_2 are mutually exclusive integers randomly chosen within [1, N]. F is a mutation factor that controls the balance between exploration and exploitation. $u_r \in [0,1]$ is a uniformly random number. Let y_i be the decision variable to be evolved. Then y_i is computed as follows:

$$y_i = \begin{cases} x_i + F(x_{r2} - x_{r1}) + \sqrt[3]{(0.5 - u_r)} * (c_1 - c_2), & if\ u_r \geq 0.5; \\ x_i + F(x_{r1} - x_{r2}) + \sqrt[3]{(0.5 - u_r)} * (c_1 - c_2), & otherwise. \end{cases} \tag{4}$$

3.3 Cloud Particle Mutation Operator

$$y_i^k = \begin{cases} y_i^k + (u_r - 0.5) * log_2(u_r) * (u_k - l_k), & if\ u_r \leq \frac{1}{D}; \\ y_i^k, & otherwise. \end{cases} \tag{5}$$

where u_k and l_k are the upper bound and lower bound of the kth decision vector respectively. D is the dimension of decision space. $u_r \in [0,1]$ is a uniformly random number.

Algorithm 2. The procedure of MOEA/D-CPDE.

Input: MOP, the multiobjective optimization problem; *NP*, population size; *GEN*, maximum generations; *FES*, the maximal number of function evaluations; *NB*, the number of weight vectors in the neighborhood of each weight vector;

Output: **X**, the non-dominated population; Approximation to the PF, $\{F(\mathbf{x}^1), \ldots, F(\mathbf{x}^N)\}$

1: Generate initial population $\mathbf{X}_n$ by cloud generator

2: Compute the Euclidian distances between any two weight vectors and define neighborhood of each subproblem. For the *i*th subproblem, set $B(i)=\{i_1,i_2,\ldots i_m\}$, where λ^{i1}, λ^{i2}, …, λ^{im} are the *m* closest weight vectors to λ^i.

3: Initialize $\mathbf{z} = (z_1,\ldots, z_m)^T$ by problem-specific method

4: while $t < FES$ do

5:   for $i = 1{:}NP$ do

6:     $Ex_i = x_i$

7:     $En_i = |Ex_i|/10$

8:     $He_i = En_i/1000$

9:     generate c_1 and c_2 by cloud generator

10:     Produce y_i by Cloud Particle Differential Evolution

11:     if $rand \geq 0.5$

12:       Cloud Particle Mutation Operator

13:     else

14:       polynomial mutation

15:     end

16:     Evaluation y

17:     for each j=1 to m do

18:       if $z_j > f_j(y)$ then $z_j = f_j(y)$

19:     endfor

20:     for each $j \in B(i)$ do

21:       if $g^{te}(y|\lambda^j,z) < g^{te}(x^j|\lambda^j,z)$ then

22:         $x^j = y$ and $F(x^j) = F(y)$

23:       endif

24:     endfor

25:   endfor

26: endwhile

4. Experimental and Discussions

In this section, we test MOEA/D, multiobjective memetic algorithm based on the decomposition approach and the particle swarm optimization (PSO) algorithm (MOEA/D-DE+PSO)and multiobjective decomposition evolutionary algorithm based on Cloud Particle Differential Evolution (MOEA/D-CPDE) on DEB, ZDT problems [30] and IEEE Congress on Evolutionary Computation 2009 (CEC'09) test instances [31], in which DEB, ZDT, unconstrained problem 1 to unconstrained problem 7 (UF1–UF7) are two objective problems and UF8-UF10 are three objective problems. All of them are unconstraint problems.

4.1. Parameter Settings

For all experiments, 30 independent runs are carried out on the same machine with Matlab R2009b, and conducted with the maximum number of function evaluations as the termination criterion. The goal is to ensure a fair comparison and reduce the statistical error.

In the numerical experiments, the dimension of all test problems is set as 30 for ZDT problems and UF1–UF10, and 2 for DEB. The population size is set to 100, and the maximum number of function evaluation is set as 100,000 for DEB and ZDT problems, and 300,000 for UF1-UF10 for each algorithm.

For MOEA/D-CPDE, there are four other control parameters: $CR = 1$, $F = 0.5$, $E_n = |E_x|/10$, $H_e = E_n/1000$.

The number of neighbors for each subproblem is set as $0.1 \times Np$, where Np is the population size.

4.2. Performance Metrics

In this paper, we use the Inverted Generational Distance (*IGD*) metric, which is a comprehensive index of convergence and diversity [32] to assess the performance of the proposed algorithm. The smaller the value of the *IGD* metric, the better is the obtained set of Pareto optimal solutions.

Let P^* be a set of uniformly distributed points in the objective space along the Pareto-optimal front (PF). Let P be an approximate solution to the *PF*, then the average distance from P^* to P can be defined as follows [33].

$$\mathrm{IGD}(\mathrm{P}, P^*) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \tag{6}$$

where $d(v, P)$ is the Euclidean distance of v to P. The lower value of *IGD* means that P is much closer to the *PF*.

4.3. Experimental Results

This section presents the numerical results obtained by MOEA/D, MOEA/D-DE+PSO [34] and MOEA/D-CPDE on DEB, ZDT problems and the CEC'09 test instances (UF1-UF10).

Table 1 shows the results obtained by MOEA/D, MOEA/D-DE+PSO and MOEA/D-CPDE. The results of MOEA/D-DE+PSO are taken from literature [34]. It records the IGD-metric values after 100,000 function evaluations in 30 runs. There are five columns in Table 1. The columns refer to best (*i.e.*, minimum), mean, standard deviation (std) and worst (*i.e.*, maximum) of the *IGD* values obtained by the algorithms for DEB and ZDT problems. The experimental results indicate that MOEA/D-CPDE performs better than MOEA/D and MOEA/D-DE+PSO on ZDT1, ZDT2, ZDT4 and ZDT6. The minimum of MOEA/D is the smallest on ZDT3. Table 1 concludes that the cloud differential evolution strategy plays an important role in guiding the population evolution and accelerating the convergence rate of MOEA/D-CPDE for DEB and ZDT problems.

Table 2 shows the *IGD* in terms of minimum, mean, standard deviation and maximum values obtained by MOEA/D, MOEA/D-DE+PSO and MOEA/D-CPDE for 30 independent times. The results of MOEA/D-DE+PSO are taken from literature [34]. Table 2 shows that, in terms of IGD-metric, MOEA/D-CPDE performs better on UF1, UF4-UF7. MOEA/D-DE+PSO performs better on UF2, UF3, UF8 and UF10. MOEA/D-CPDE performs better in terms of the standard deviation and maximum on UF9. MOEA/D-DE+PSO performs better in terms of the minimum and mean best solution on UF9. It may be noted that MOEA/D-CPDE has not performed well on a few functions. UF8–UF10 are three objective functions with complicated *PS* shapes. The population diversity mechanism in MOEA/D-CPDE is worse for the complicated functions. Therefore, the complicated functions may be very challenging for MOEA/D-CPDE. Because of the poor convergence speed, MOEA/D performs worse for some complicated problems with the given number of function evaluations as the termination criterion. The dynamic use of DE and PSO enable MOEA/D-DE+PSO to get better approximate solutions for some problems. However, the individual produced by DE and PSO comes from two randomly divided subsets. Therefore, MOEA/D-DE+PSO performs worse for some problems.

Table 1. The Inverted Generational Distance (*IGD*) values for DEB, ZDT1~ZDT4 and ZDT6 approximated by multiobjective evolutionary algorithm based on decomposition (MOEA/D), MOEA/D-DE+PSO and MOEA/D-CPDE in 30 independent runs.

Function	Min	Mean	Std	Max	Algorithms
DEB	0.005354	0.005355	4.7946e-7	0.005355	MOEA/D
	-	-	-	-	MOEA/D-DE+PSO
	0.005354	**0.005354**	**5.0741e-7**	0.005355	MOEA/D-CPDE
ZDT1	0.003988	0.004054	5.1587e-5	0.004165	MOEA/D
	0.004012	0.004142	7.1000e-5	0.004359	MOEA/D-DE+PSO
	0.003966	**0.004036**	**4.7909e-5**	**0.004088**	MOEA/D-CPDE
ZDT2	0.003799	0.003819	1.5762e-5	0.003864	MOEA/D
	0.003811	0.003874	4.2400e-5	0.003992	MOEA/D-DE+PSO
	0.003784	**0.003808**	**1.1640e-5**	**0.003831**	MOEA/D-CPDE
ZDT3	**0.007041**	0.007084	3.4618e-5	0.007200	MOEA/D
	0.008400	0.009025	5.9532e-4	0.012406	MOEA/D-DE+PSO
	0.007049	**0.007081**	**2.8241e-5**	**0.007171**	MOEA/D-CPDE
ZDT4	0.125709	0.195898	6.7636e-2	0.256064	MOEA/D
	0.004141	0.007550	1.9710e-3	0.011584	MOEA/D-DE+PSO
	0.003953	**0.003954**	**8.0301e-7**	**0.003956**	MOEA/D-CPDE
ZDT6	0.008892	0.013408	3.6559e-3	0.017854	MOEA/D
	0.006977	0.014575	4.0370e-3	0.022697	MOEA/D-DE+PSO
	0.005087	**0.005970**	**6.5858e-4**	**0.006620**	MOEA/D-CPDE

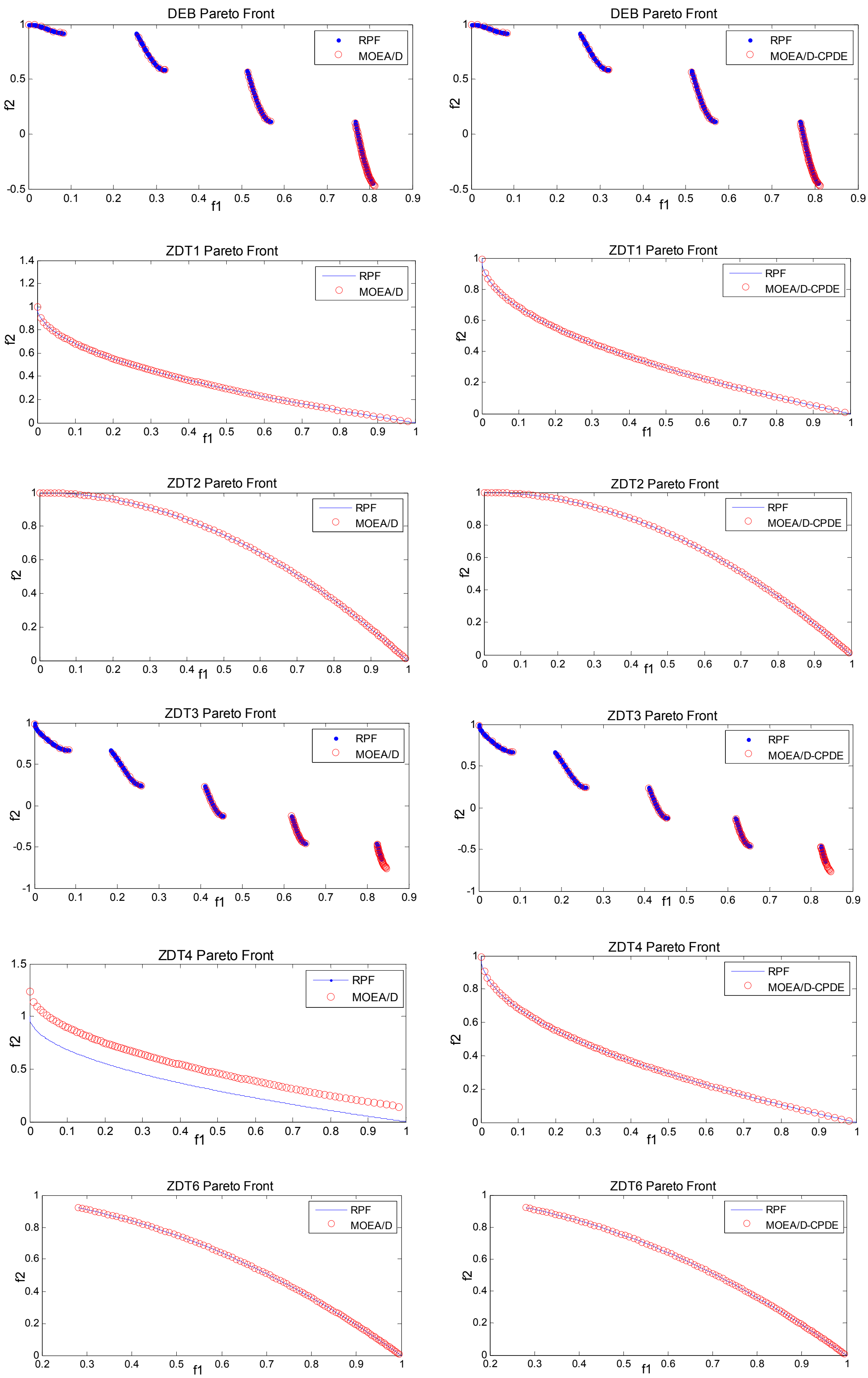

Figure 1. Plots of the final non-dominated solutions in the objective space of DEB, ZDT1~ZDT4, and ZDT6.

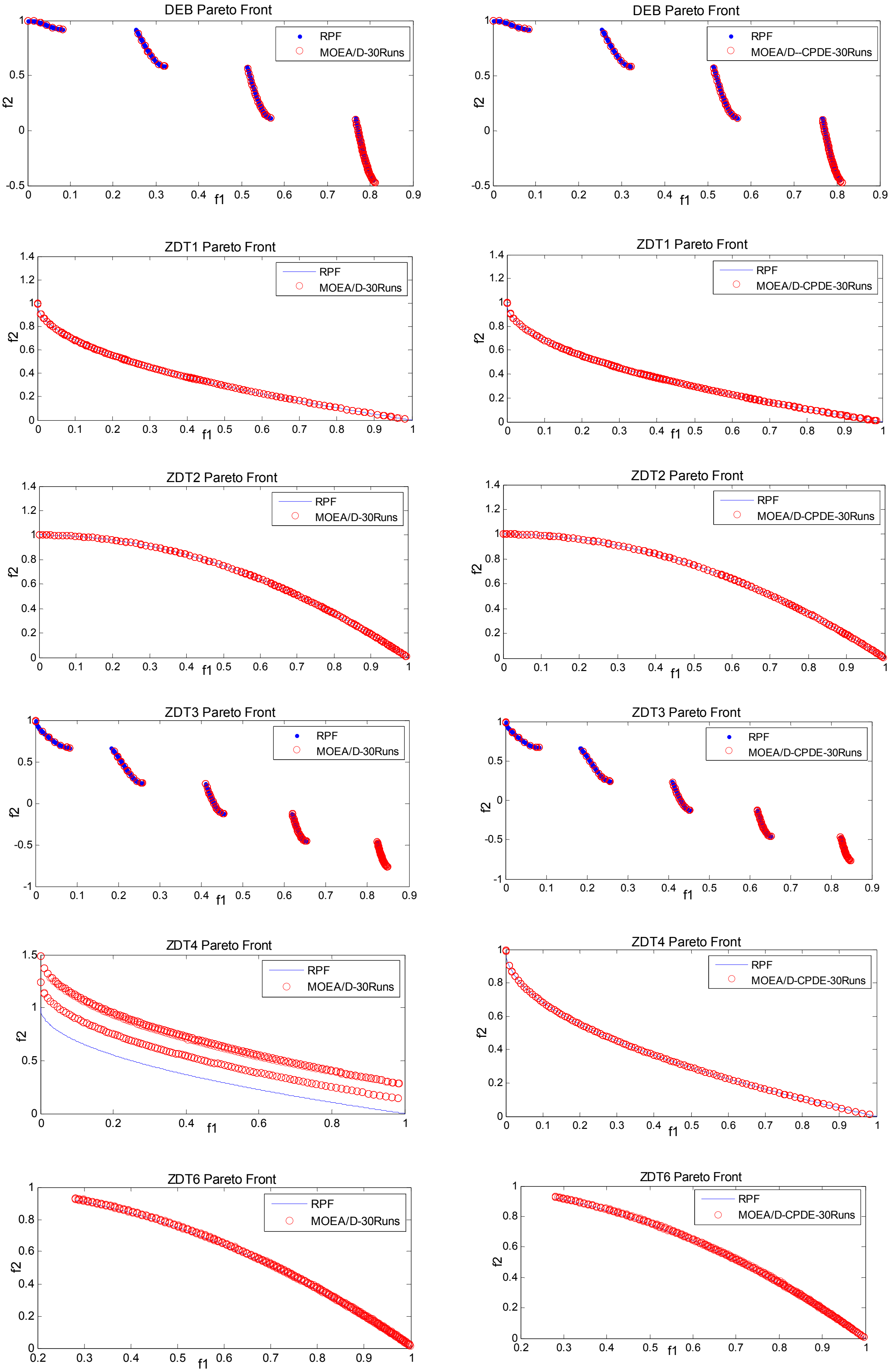

Figure 2. Plots of the all 30 PFs for DEB, ZDT1~ZDT4, and ZDT6.

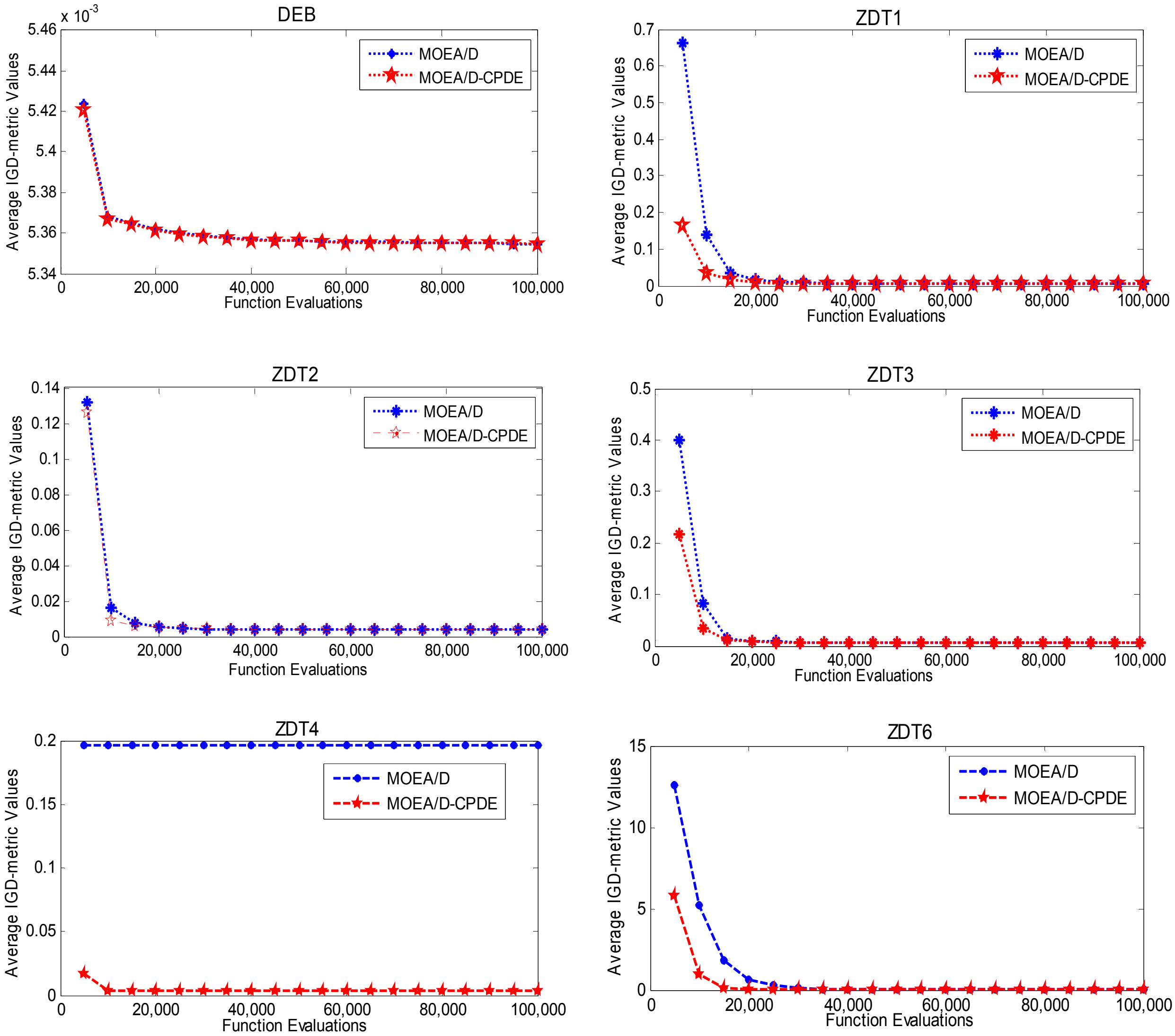

Figure 3. Average IGD values versus number of function evaluations over DEB, ZDT1~ZDT4, and ZDT6

Table 2. The IGD values statistics of the non-dominated solutions found by MOEA/D, MOEA/D-DE+PSO and MOEA/D-CPDE on UF1-UF10 in 30 independent runs.

Function	Min	Mean	Std	Max	Algorithms
	0.004711	0.023042	0.053302	0.259563	MOEA/D
UF1	**0.004466**	0.028572	0.023609	0.071182	MOEA/D-DE+PSO
	0.004594	**0.005203**	**0.000248**	**0.006144**	MOEA/D-CPDE
	0.009082	0.014285	0.007173	0.049397	MOEA/D
UF2	0.010218	**0.011737**	**0.000756**	**0.013261**	MOEA/D-DE+PSO
	0.008998	0.011877	0.001714	0.015334	MOEA/D-CPDE
	0.016796	0.062967	0.028005	0.142894	MOEA/D
UF3	**0.003940**	**0.006039**	**0.001826**	**0.010494**	MOEA/D-DE+PSO
	0.008435	0.044706	0.029862	0.114730	MOEA/D-CPDE

Table 2. *Cont.*

UF4	0.061278	0.069968	0.006585	0.089038	MOEA/D
	0.047843	0.054604	0.004053	0.071241	MOEA/D-DE+PSO
	0.043632	**0.045344**	**0.001319**	**0.049677**	MOEA/D-CPDE
UF5	0.208215	0.422297	0.128282	0.729846	MOEA/D
	0.282111	0.490882	0.118561	0.708999	MOEA/D-DE+PSO
	0.115859	**0.203445**	**0.045309**	**0.258254**	MOEA/D-CPDE
UF6	0.162838	0.495739	0.199592	0.824257	MOEA/D
	0.186140	0.243517	0.237355	0.674953	MOEA/D-DE+PSO
	0.073819	**0.091573**	**0.030068**	**0.210588**	MOEA/D-CPDE
UF7	**0.005013**	0.147765	0.219858	0.598471	MOEA/D
	0.006726	0.243517	0.237355	0.674953	MOEA/D-DE+PSO
	0.005112	**0.006067**	**0.000640**	**0.007461**	MOEA/D-CPDE
UF8	0.090851	0.145518	0.031061	0.229571	MOEA/D
	0.057600	**0.078455**	0.006532	**0.101902**	MOEA/D-DE+PSO
	0.112641	0.123347	**0.006231**	0.139016	MOEA/D-CPDE
UF9	0.087260	0.166323	0.033758	0.194023	MOEA/D
	0.035499	**0.071131**	0.035008	0.149478	MOEA/D-DE+PSO
	0.074763	0.080028	**0.002555**	**0.083058**	MOEA/D-CPDE
UF10	0.231245	0.431299	0.084250	0.637907	MOEA/D
	0.184050	**0.187158**	**0.001552**	**0.190097**	MOEA/D-DE+PSO
	0.369133	0.499921	0.078278	0.688156	MOEA/D-CPDE

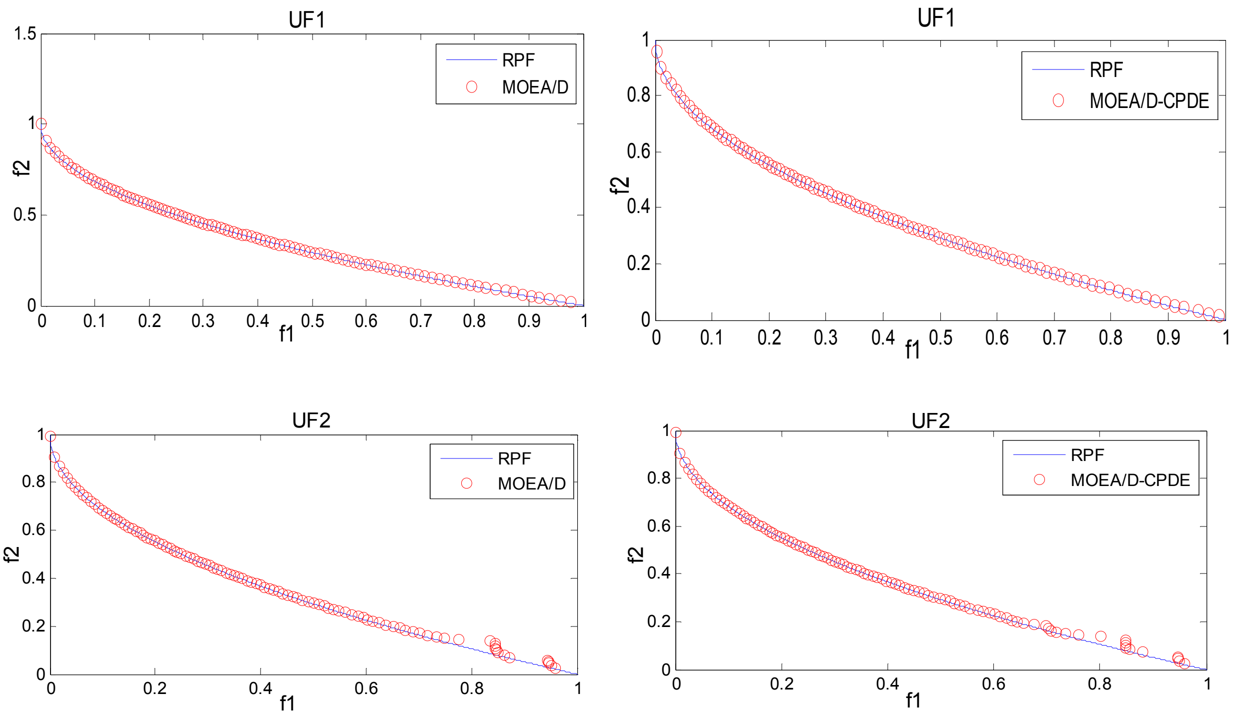

Figure 4. *Cont.*

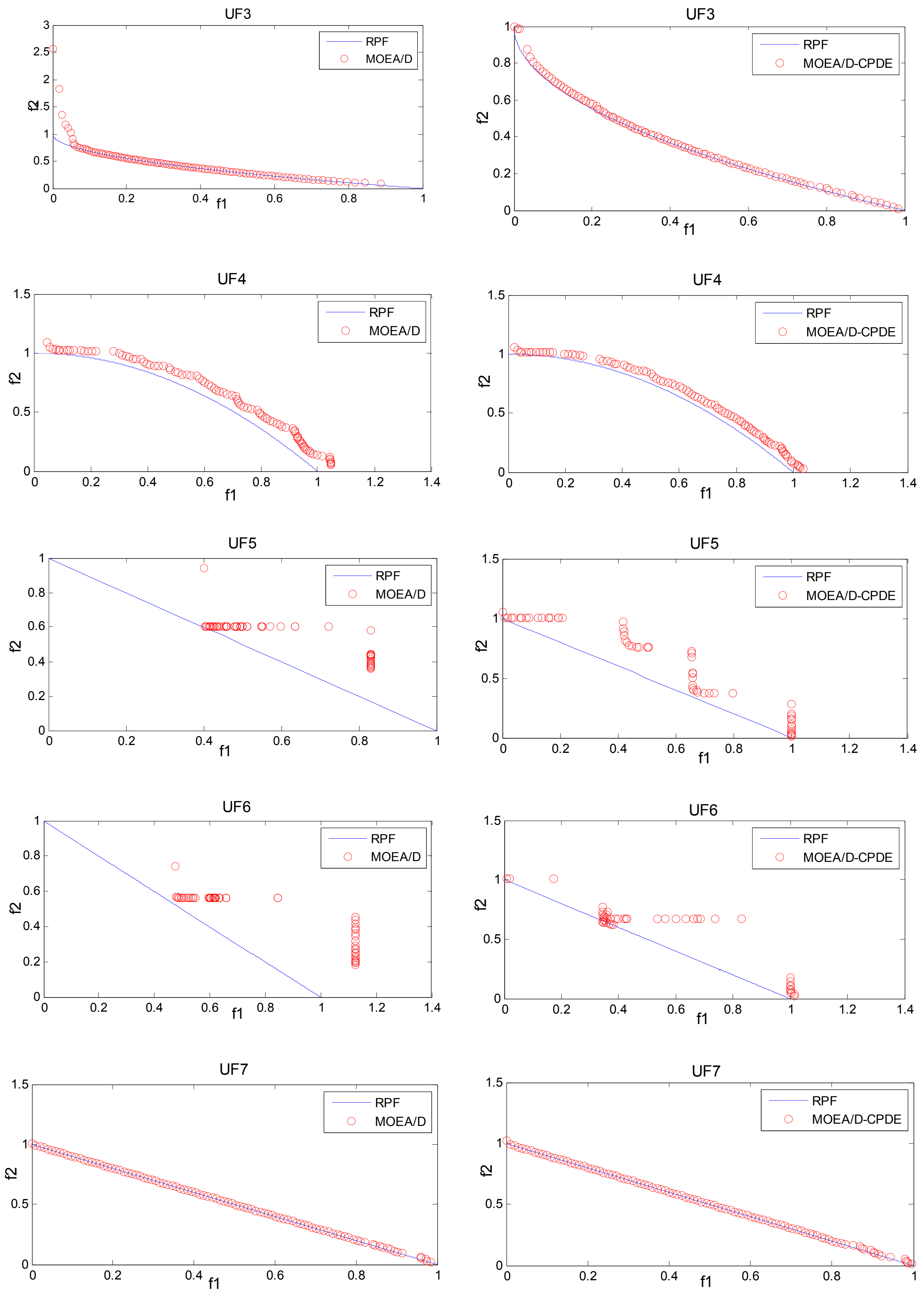

Figure 4. *Cont.*

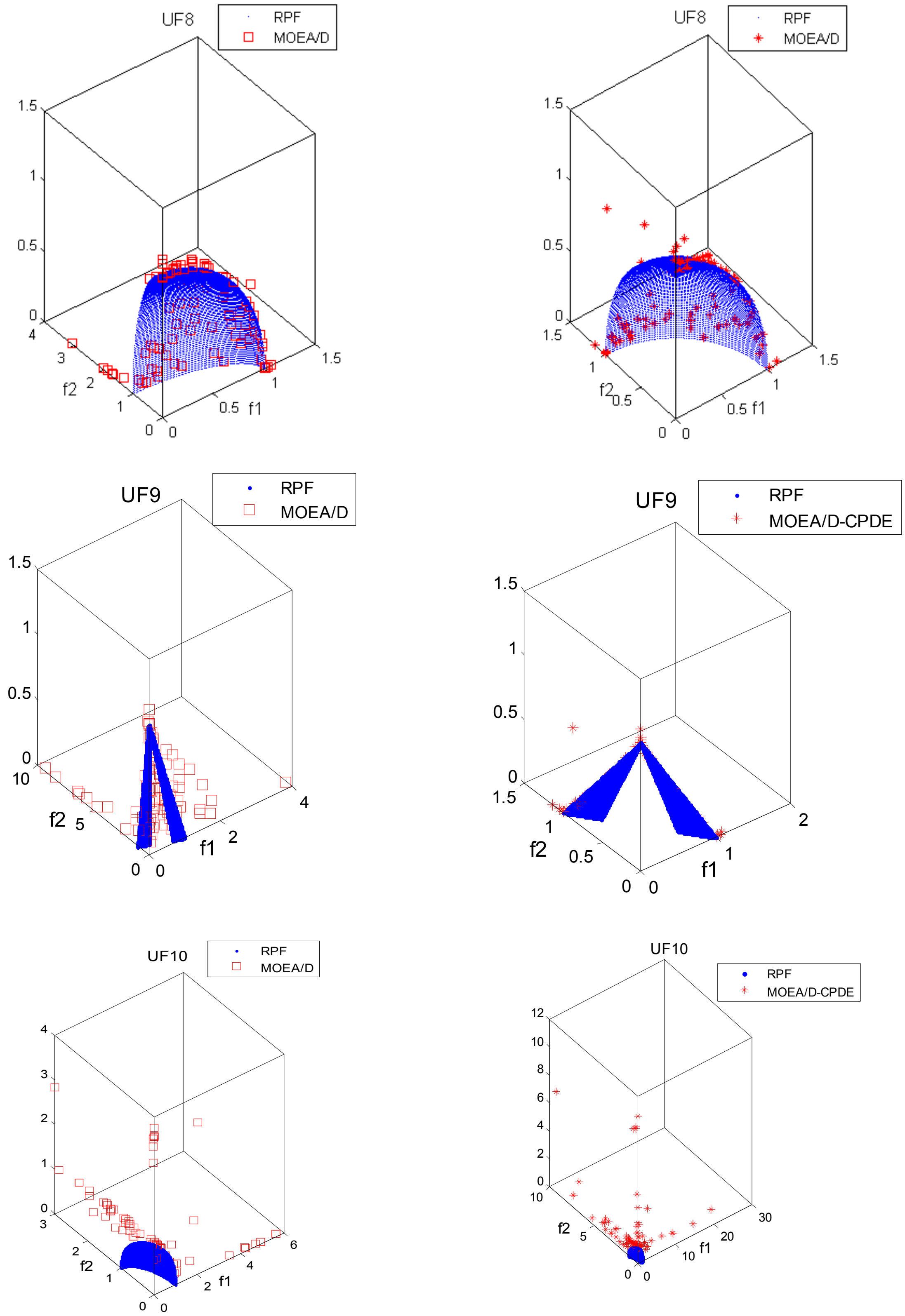

Figure 4. Plots of the best PF in the objective space of the UF1-UF10 in 30 independent runs.

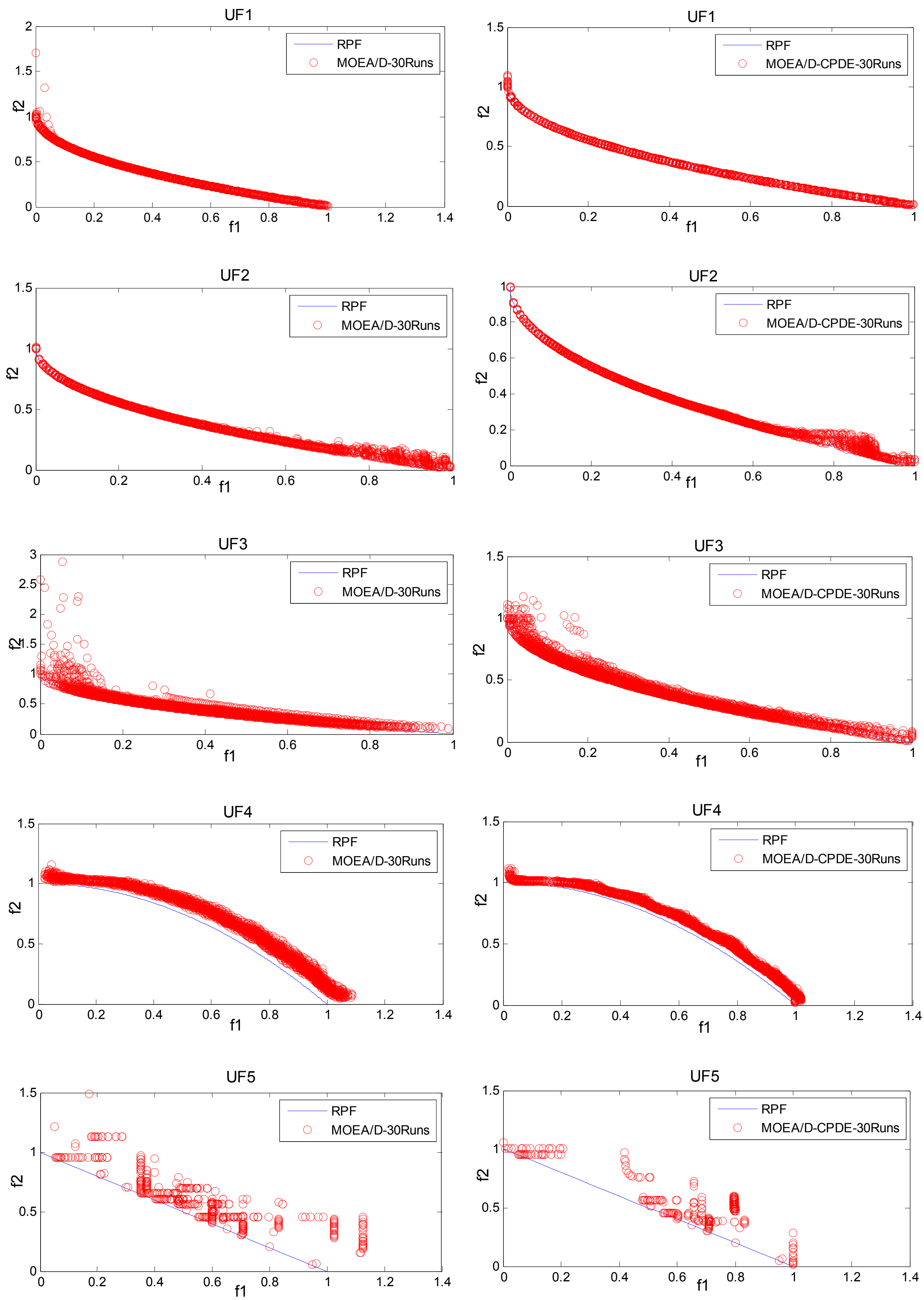

Figure 5. *Cont.*

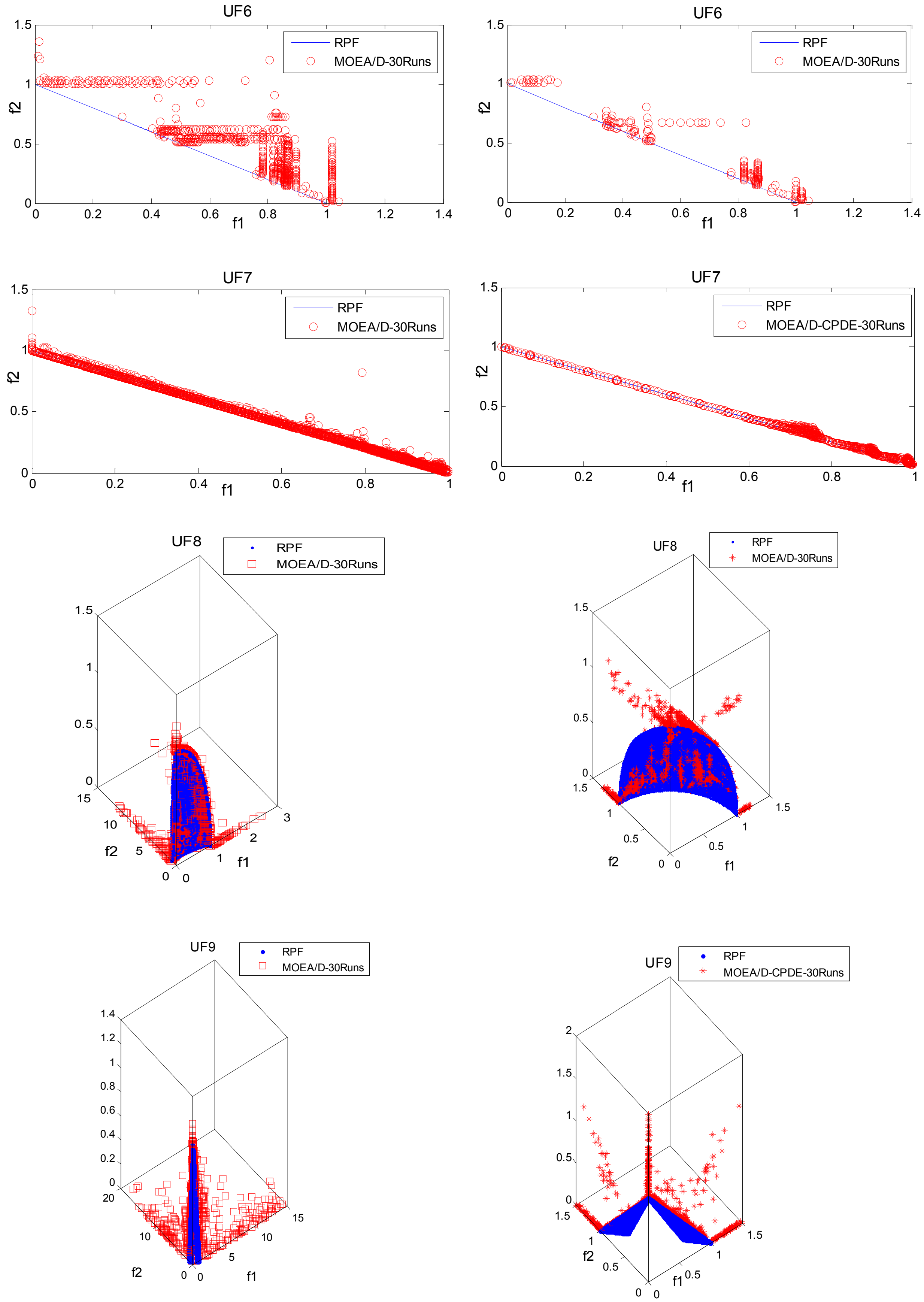

Figure 5. *Cont.*

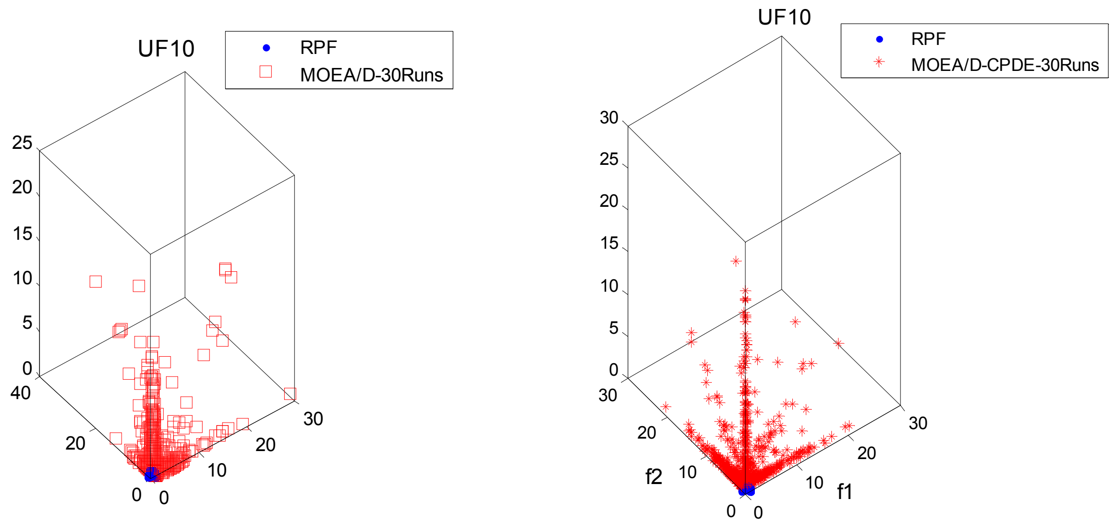

Figure 5. Plots of the all 30 PFs in the objective space of the UF1-UF10 in 30 independent runs

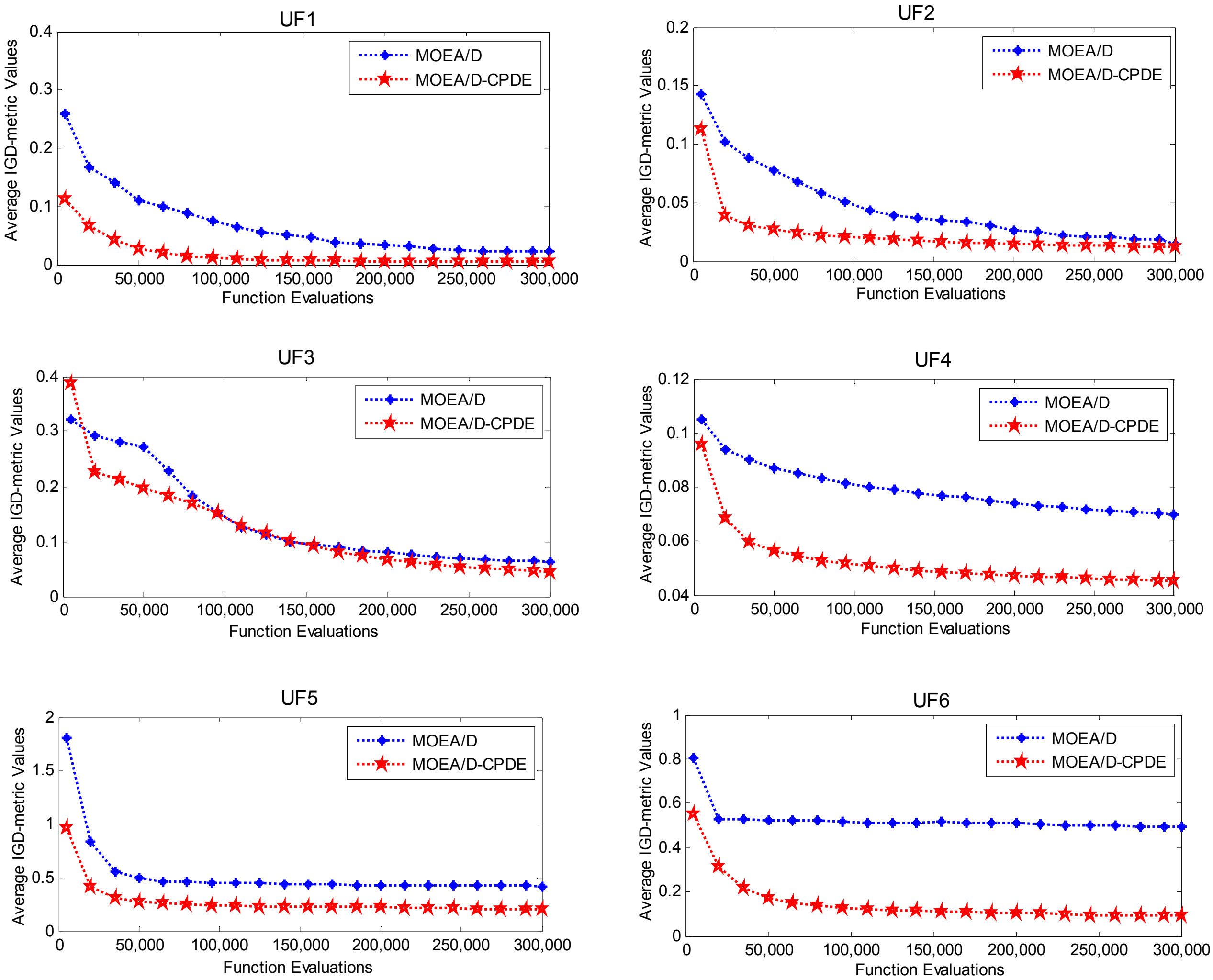

Figure 6. *Cont.*

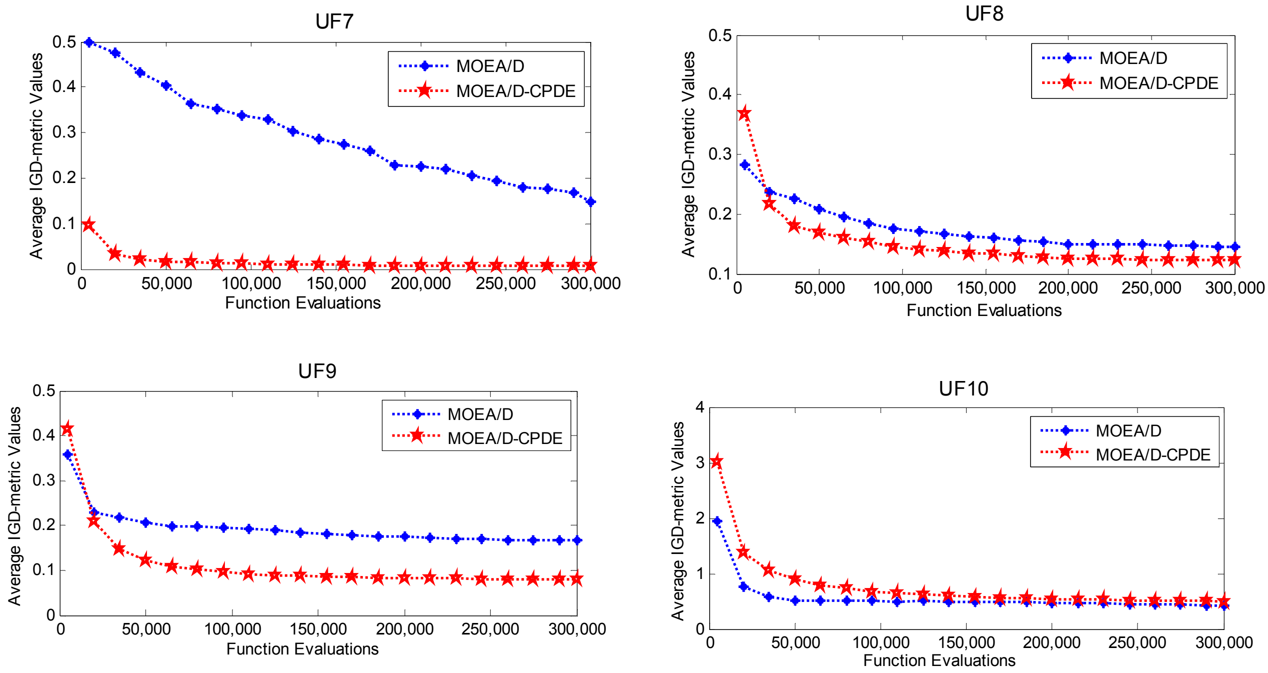

Figure 6. Evolution of the average IGD-metric values versus function evaluations for UF1-UF10.

5. Conclusion

In this paper, we proposed an MOEA/D-CPDE algorithm based on MOEA/D. In MOEA/D-CPDE, two individuals are generated by the cloud generator guidance mechanism. Furthermore, the cloud particles mutation operator is introduced to maintain the population diversity. The performance of MOEA/D, MOEA/D-DE+PSO and MOEA/D-CPDE are compared on several continuous multiobjective optimization problems. The experiments conclude that the MOEA/D-CPDE generally offers better performance than those of other algorithms in the paper for some functions. However, the performance of MOEA/D-CPDE is worse for some functions with complicated *PS* shapes. The reason is that the misleading effect of two individuals generated by the cloud generator may cause the premature convergence. How to effectively guide the population to evolve will be considered in the future.

Acknowledgments

The authors express their sincere thanks to the reviewers for their significant contributions to the improvement of the final paper. This research is partly supported by the support of the National Natural Science Foundation of China (No. 61272283, U1334211), Shaanxi Science and Technology Project (No. 2014KW02-01), Xi'an Science and technology plan project CXY1437(5), Xi'an BeiLin Science Research Plan(No.GX1306).

Author Contributions

Joint work.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Coello, Coello, C.A.; Pulido, G.T.; Montes, E.M. Current and future research trends in evolutionary multiobjective optimization. From Industrial Applications to Academic Speculations, In *Book Information Processing with Evolutionary Algorithms*; Wu, X., Jain, L., Graña, M., Duro, R.J., d'Anjou, A., Wang, P.P., Eds.; Springer: London, UK, 2005; pp. 213–231. DOI:10.1007/1-84628-117-2_15
2. Andersson, J. A survey of multiobjective optimization in engineering design, Tech. Rep. LiTH-IKP-R-1097, Department of Mechanical Engineering, Linköping University, Linköpings, Sweden, 2000.
3. Marler, R.T.; Arora, J.S. Survey of multi-objective optimization methods for engineering. *Struct. Multidiscip. Optim.* **2004**, *26*, 369–395.
4. Lei, D. Multi-objective production scheduling: a survey. *Int. J. Adv. Manuf. Technol*. **2009**, *43*, 926–938.
5. Tapia, M.G.C.; Coello, Coello, C.A. Applications of multi-objective evolutionary algorithms in economics and finance: a survey. *Evol. Comput.* **2007**, 532–539.
6. Luna, F.; Durillo, J.J; Nebro, A.J.; Alba, E. Evolutionary algorithms for solving the automatic cell planning problem: S survey. *Eng. Optim*. **2010**, *42*, 671–690.
7. Lust, T.; Teghem, J. The multiobjective traveling salesman problem: a survey and a new approach. In: *Advances in Multi-Objective Nature Inspired Computing*; Coello Coello, C.A., Dhaenens, C., Eds.; Springer: Heidelberg, Berlin, Germany, 2010; 272, pp. 119–141.
8. Coello Coello, C.A.; Van Veldhuizen, D.A.; Lamont, G.B. *Evolutionary Algorithms for Solving Multiobjective Problems*; Springer US: New York, NY, USA, 2002.
9. Deb, K.; Agrawal, S.; Pratap, A.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE *Tran. Evol. Comput*. **2002**, *6*, 182–197.
10. Knowles, J.; Corne, D. Memetic algorithms for multiobjective optimiza-tion: issues, methods and prospects, In: *Recent Advances in Memetic Algorithms*; Springer: Berlin Heidelberg, Germany, 2004; pp. 313–352.
11. Suman, B.; Kumar, P. A survey of simulated annealing as a tool for single and multiobjective optimization, *J. Oper. Res. Soc.* **2006**, *57*, 1143–1160.
12. Reyes-Sierra, M.; Coello Coello, C.A. Multi-objective particle swarm optimiz-ers: a survey of the state-of-the-art, *Int. J. Comput. Intell. Res*. **2006**, *2*, 287–308.
13. Gong, M.G; Jiao, L.C; Du, H.F.; Bo, L.F. Multiobjective immune algorithm with nondominated neighbor-based selection. *Evol. Comput*. **2008**, *16*, 225–255.
14. Zhang, Q.F.; Zhou, A.M.; Jin, Y. RM-MEDA: a regularity model-based multi-objective estimation of distribution algorithm. *IEEE Tran. Evol. Comput*. **2008**, *21*, 41–63.
15. Largo, Á.R.; Zhang, Q.F. A. Miguel V. Rodríguez. A multiobjective evolutionary algorithm based on decomposition with normal boundary intersection for traffic grooming in optical networks, *Inform. Sci*. **2014**, *289*, 91–116.

16. Tadahiko, M.; Hisao, I.; Mitsuo, G. Specification of genetic search directions in cellular multi-objective genetic algorithms, Specification of Genetic Search Directions in Cellular Multi-objective Genetic Algorithms, In *Evolutionary Multi-Criterion Optimization*; Zitzler, E., Thiele, L., Deb, K., Coello Coello, C.A., Corne, D., Eds.; SpringerI Berlin Heidelberg, Germany, EMO 2001, *1993*; pp. 82–95.
17. Zhang, Q.; Li, H. MOEA/D: a multiobjective evolutionary algorithm based on decomposition, *IEEE Tran. Evol. Comput*. **2007**, *11*, 712–731.
18. Palmers, P.; McConaghy, T.; Steyaert, M.; Gielen, G.G.E. Massively multi-topology sizing of analog integrated circuits. Proceedings of Conference on Design, Automation and Test in Europe, Leuven, Belgium, 2009; pp. 706–711.
19. Nebro, A.J.; Durillo, J.J. A study of the parallelization of the multi-objective metaheuristic MOEA/D. Proceedings of 4th International Conference on Learning and Intelligent Optimization, LION 4, Venice, Italy, 18–22 January 2010; pp. 303–317.
20. Mashwani, W.K.; Salhi, A. A decomposition-based hybrid multiobjective evolutionary algorithm with dynamic resource allocation, *App. Soft Comput.* **2012**, *12*, 2765–2780.
21. Tan, Y.Y.; Jiao, Y.C.; Li, H.; Wang, X.K. MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives, *Comput. Oper. Res.* **2013**, *40*, 1648–1660.
22. Ma, X.L.; Liu, F.; Qi, Y.T.; Gong, M.G.; Yin, M.L.; Li, L.L.; Jiao, L.C.; Wu, J.S. MOEA/D with opposition-based learning for multiobjective optimization problem. *Neurocomputing* **2014**, *146*, 48–64.
23. Chang, P.C.; Chen, S.H.; Zhang, Q.; Lin, J.L. MOEA/D for flowshop scheduling problems. *Evol. Comput.* **2008**, *2008*, 1433–1438.
24. Yuen, T.J.; Raml, R. Comparison of computational efficiency of MOEA/D and NSGA-II for passive vehicle suspension optimization. Proceedings of 24th European Conference on Modelling and Simulation, Kuala Lumpur, Malaysia, ECMS 2010; pp. 2010, 219–225.
25. Konstantinidis, A.; Charalambous, C.; Zhou, A.; Zhang, Q. Multi-objective mobile agent-based sensor network routing using MOEA/D. *Evol Comput.* **2010**, 1–8.
26. Nicola, B.; Naujoks, B.; Emmerich, M. SMS-EMOA:Multiobjective selection based on dominated hypervolume, *Eur. J. Oper. Res*. **2007**, *181*, 1653–1669.
27. Johannes, B.; Eckart, Z. HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evol. Comput.* **2010**, *19*, 45–76.
28. Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*; Wiley: New York, NY, USA, 2001; p. *16*.
29. Li, D.Y. Uncertainty in Knowledge Representation. *Eng. Sci.* **2000**, *2*, 73–79(in Chinese).
30. Zitzler, E.; Deb, K.; Thiele, L. Comparsion of multiobjective evolutionary algorithms: emperical results. *Evol. Comput.* **2000**, 8, 173–195.
31. Zhang, Q.; Liu, W.; Li, H. The Performance of a new version of MOEA/D on CEC'09 unconstrained MOP Test Instances, IEEE Congress On Evolutionary Computation (IEEE CEC 2009), Trondheim, Norway, May 18-21, 2009; pp. 203–208.
32. Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C.; Fonseca, V. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.* **2003**, *7*, 117–132.

33. Zhang, Q.; Zhou, A.; Zhao, S.; Suganthany, P.N.; Liu, W.; Tiwariz, S. Multiobjective optimization test instances for the CEC 2009 special session and competition, Technical. Report CES-487, 2009. Available online: http://dces.essex.ac.uk/staff/zhang/MOEAcompetition/cec09testproblem0904.pdf (accessd on 22 April 2015)
34. Wali Khan, M.; Abdellah. S. Multiobjective memetic algorithm based on decomposition. *App. Soft Comput.* **2014**, *21*, 221–243.