An Enhanced MOEA/D-DE and Its Application to Multiobjective Analog Cell Sizing

Bo Liu, Francisco V. Fernández, Qingfu Zhang, *Senior Member, IEEE*, Murat Pak, Suha Sipahi and Georges Gielen, *Fellow, IEEE*

Abstract—**Recently, a multiobjective evolutionary algorithm based on decomposition (MOEA/D) and its extended version by using differential evolution (DE) as the main search engine (MOEA/D-DE) were proposed, which outperform several widely used multiobjective evolutionary algorithms. MOEA/D decomposes a multiobjective problem into a number of scalar optimization sub-problems with a neighborhood structure and optimizes them simultaneously to approximate the Pareto-optimal set. In this paper, two mechanisms are investigated to enhance the performance of MOEA/D-DE. Firstly, a new replacement mechanism is proposed to call for a balance between the diversity of the population and the employment of good information from neighbors. Secondly, the scaling factor in DE is randomized to enhance the search ability. Comparisons are carried out with MOEA/D-DE on ten benchmark problems, showing that the proposed method exhibits significant improvements. Finally, the enhanced MOEA/D-DE is applied to a real world problem, the sizing of a folded-cascode amplifier with four performance objectives.**

I. INTRODUCTION

MANY real-world optimization applications involve several conflicting objectives [1]-[2]. According to different purposes and requirements in the decision-making process, multiobjective optimization techniques can be roughly classified into two categories [1]: (1) a prior methods: a decision maker specifies their preferences on these objectives and thus transform the multiobjective problem into a single objective one by using aggregation methods, and (2) a posteriori methods: they produce a number of well representative optimal trade-off candidate solutions for a decision-maker to check. Mathematically, a Pareto optimal solution is a candidate solution for achieving the best trade-off. There can be many, even infinitely many Pareto optimal solutions to a multiobjective optimization problem (MOP). The set of all the Pareto optimal solutions is called the Pareto set (PS) and its image in the objective space is the Pareto front (PF). Most multiobjective optimization evolutionary algorithms (MOEA) aim at finding a reasonable number of solutions to approximate the PF. In a sense, MOEAs belong to category 2.

Georges Gielen and Bo Liu are with ESAT-MICAS, Katholieke Universiteit Leuven, Leuven, Belgium (e-mail: {Georges.Gielen, Bo.Liu}@esat.kuleuven.be, liu_bo765@yahoo.com.cn). Francisco Fernández is with IMSE, CSIC and University of Sevilla, Sevilla, Spain. (e-mail: Francisco.Fernandez@imse-cnm.csic.es). Qingfu Zhang is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K. (e-mail: qzhang@essex.ac.uk). Murat Pak and Suha Sipahi are with Bogazici University, Istanbul, Turkey. (e-mail: {murat.pak, suha.sipahi}@boun.edu.tr).

Most MOEAs compare solutions based on dominance. However, domination cannot provide a full ranking among all the solutions. Therefore, these MOEAs need some other techniques for ranking solutions (e.g. crowding distances, fitness sharing, niching). Among these algorithms, non-dominated sorting genetic algorithm II (NSGA-II) [3] and strength Pareto evolutionary algorithm 2 (SPEA2) [4] have received much attention in real world applications. However, it is shown that these methods cannot always provide good results, especially when the MOP is complicated [5]-[6].

Recently, a new MOEA framework, multiobjective evolutionary algorithm based on decomposition (MOEA/D) [5], was proposed. It decomposes a MOP into a set of scalar optimization sub-problems with neighborhood relations. The neighborhood relations are defined by the distances between their aggregation coefficient vectors. In this way, the fitness assignment is the same as single objective optimization, and the diversity is maintained by the diverse search directions determined by the uniformly distributed weight vectors. The first version of MOEA/D uses simulated binary crossover (SBX) [7] and polynomial mutation [3] as the search engines. Later, a new version using the mutation (DE/best/1/bin [8]) in differential evolution (DE) as the main search engine was proposed and shown to outperform MOEA/D and NSGA-II, especially for complex problems.

There are several possibilities to enhance the performance of the MOEA/D-DE framework. The first one is the population replacement. The goal is to call for a balance between information sharing and diversity maintenance. In our method, when the number of parent solutions that can be replaced by a high quality child solution exceeds the maximum number, we rank the parent solutions and first replace those that are closer to the child solution. The second one is to enhance the search ability. We randomize the scaling factor in the DE mutation to achieve this.

The rest of the paper is organized as follows. Section II describes the basic concepts of MOPs. Section III introduces the basic MOEA/D framework and the two mechanisms to extend MOEA/D-DE. The tests and comparisons based on benchmark problems and a multiobjective analog cell sizing problem are shown in section IV. The concluding remarks are offered in Section V.

II. Multiobjective Optimization

A multiobjective optimization problem can be stated as follows:

$$\min\{f_1(x),\dots f_m(x)\}\ ,\ x \in \Omega \tag{1}$$

where $x=(x_1,\dots x_n)$ is the decision variable vector and $f_i(x)$ are the objective functions. Ω is the decision space. A solution x is said to dominate solution y if and only if $f_i(x) \le f_i(y)$ for every $i \in \{1,\cdots,m\}$ and $f_j(x) < f_j(y)$ for at least one index $j \in \{1,\cdots,m\}$. A point $x^* \in \Omega$ is Pareto optimal to (1) if there is no point $x \in \Omega$ such that $f(x)$ dominates $f(x^*)$. $f(x^*)$ is Pareto-optimal objective vector. The set of all the Pareto-optimal points is called the Pareto Set (PS). The set of all the Pareto-optimal objective vectors is called the Pareto Front (PF).

III. The Enhanced MOEA/D-DE Algorithm

A. Algorithm Structure

A general framework of MOEA/D is proposed in [5]. The first step is converting the approximation of the PF into *N* scalar optimization sub-problems by decomposition. Three decomposition methods are considered in [5]. In this paper, we use the Tchebycheff approach [9]. The scalar function is as follows:

$$\text{minimize } g(x \mid \lambda, z^*) = \max_{1\le i\le m}\{\lambda_i \mid f_i(x)-z_i^* \mid\} \tag{2}$$

s.b. $x \in \Omega$

where $\lambda=(\lambda_1,\cdots,\lambda_m)$ is a weight vector and $\Sigma_{i=1}^{m}\lambda_i=1$. Ω is the solution space and $z^*=(z_1^*,\cdots,z_m^*)$ is the reference point. If *N* is reasonably large and $\lambda^1,\cdots,\lambda^N$ are properly selected, the optimal solutions to those scalar functions will provide a good approximation to the PS/PF. The major components in MOEA/D are its neighborhood concept, and its population replacement mechanism. The enhanced MOEA/D-DE, proposed in this paper, works as follows:

Input:
(1) an MOP
(2) a stopping criterion
(3) *N*: the number of sub-problems
(4) *T*: the neighborhood size
(5) δ : the probability that parent solutions are selected from the neighborhood
(6) n_r : the maximum number of solutions replaced by a child solution
(7) CR: crossover rate in DE
(8) μ,σ : the mean and variance of the scaling factor $\hat{F}$ in the DE mutation
(9) p_m : the probability to perform polynomial mutation
(10) λ : weight vector (the generation method is in [5])
Output:
(1) Approximation to the PF
(2) Approximation to the PS
Procedure:
Step 1: Initialization

Step 1.1: Compute the Euclidean distances between the weight vectors and work out the *T* closest weight vectors to each weight vector. For $i=1,\cdots N$, set $B(i)=\{i_1,\cdots,i_T\}$. $\lambda^{i_1},\cdots,\lambda^{i_T}$ are the *T* closest vectors to λ^i.

Step 1.2: Randomly generate an initial population $x_1,\cdots,x_n$. Calculate the fitness values of the population.

Step 1.3: Initialize $z=\{z_1,\cdots,z_m\}$, where $z_j=\min_{1\le i\le N} f_j(x^i)$.

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

Step 2.1: Selection of the mating pool:
Generate a random number which is uniformly distributed in [0,1]. Set

$$P=\begin{cases} B(i) & \text{if rand}<\delta \\ \{1,\cdots,N\} & \text{otherwise}\end{cases} \tag{3}$$

Step 2.2: Reproduction:
Set $r_1=i$ and randomly select two indexes r_2 and r_3 from *P*, and generate a new solution $\overline{y}$ by a new DE mutation (see Section III (C)). Then, perform a polynomial mutation [3] on $\overline{y}$ with probability p_m to produce a new solution *y*.

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

Step 2.4: Update of the reference point:
For $j=1,\cdots,m$, if $z_j > f_j(y)$, set $z_j = f_j(y)$.

Step 2.5: Replacement of solutions:
(1) For each j in P, calculate $g(y \mid \lambda^j, z)$ and $g(x^j \mid \lambda^j, z)$.
(2) Set *c*=0. If $g(y \mid \lambda^j, z) \le g(x^j \mid \lambda^j, z)$, c=c+1
(3.1) If $c \le n_r$, for each *j* with $g(y \mid \lambda^j, z) \le g(x^j \mid \lambda^j, z)$, set $x^j = y$.
(3.2) If $c > n_r$, for each *j* with $g(y \mid \lambda^j, z) \le g(x^j \mid \lambda^j, z)$, calculate the Euclidean distances between $f(y)$ and $f(x^j)$ and then rank them. Choose n_r solutions with the smallest distances. Set $x^j = y$.

Step 3: Stopping Criterion:
If the stopping criterion is satisfied, then stop the algorithm and output $\{x^1,\cdots,x^N\}$ and $\{f(x^1),\cdots,f(x^N)\}$. Otherwise, go to **Step 2**.

It can be seen that the main revisions to MOEA/D-DE [6] are the replacement of solutions (step 2.5) and the reproduction process (step 2.2), which will be described in the following sections.

B. *Replacement of Solutions*

In MOEAs, the replacement mechanism is intended to improve the quality (in terms of domination) of the population and maintain the diversity. Although in decomposition-based methods, search in different directions according to different weight vectors can "naturally" help the diversity, diversity maintenance is also affected by the replacement mechanism. A high quality child solution may replace most of the current solutions to its neighboring sub-problems. Consequently, diversity decreases significantly. In MOEA/D [5], the maximum number of solutions that can be replaced by a child solution is the size of the neighborhood, *T*, whose disadvantage is shown in [6]. MOEA/D-DE improves the replacement mechanism by adding a bound n_r, which is much smaller than *T*. A high quality child solution can replace n_r current solutions at most, which helps the diversity maintenance.

However, setting the value of n_r is not a trivial problem. n_r controls the balance of information sharing and diversity maintenance. If n_r is large, the information of a good solution can be shared by more current solutions, but the risk of diversity reduction is higher. In contrast, if n_r is small, the information can be shared by less solutions, but the diversity is maintained.

An empirical rule is proposed in [10] by setting $T = 0.1N$, $n_r = 0.01N$, and the n_r current solutions which will be replaced by a high quality child solution are randomly chosen if the bound is exceeded. Generally, this rule is reasonable. Nevertheless, both conditions, c (the number of current solutions with $g(y \mid \lambda^j, z) \leq g(x^j \mid \lambda^j, z)$) much smaller than $0.01N$ and c much larger than $0.01N$ may appear in the evolution process. When c is much larger than $0.01N$, randomly selecting $0.01N$ individuals to be replaced may not be a good solution.

It can be seen that n_r is approximately 10% of *T*, that is, for one segment with 10 points that can be replaced by a high quality child solution, only one of them can be updated. Such n_r is small to share the good information. On the other hand, n_r cannot be larger to keep the diversity. Hence, selecting which points should be replaced in order to make the sharing more effectively is a significant problem. We argue that in the objective space, points that are near to the newly generated high quality child solution can be benefit more compared with the ones that have longer distance from it if the replacement is performed. The reason is that for neighbors which have similar fitness landscapes, their optimal solutions should be close to each other in the decision space. This is the basic principle of MOEA/D. Our argument can be seen as "neighbor's neighbor". In Fig. 1, the bottom point with a coordinate (1,1) is the high quality child solution, and can replace all of the 6 points with '*' symbol. If only one can be selected, then the two points in the circle will benefit more than the other 4 points if the schema of the bottom point is used.

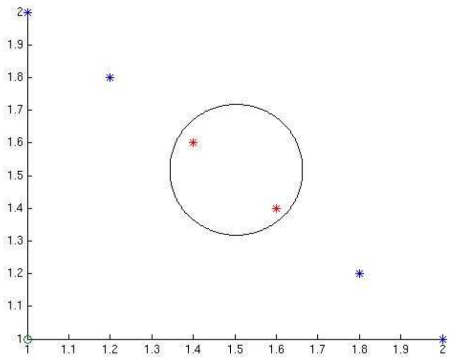


Fig. 1. Illustration of the replacement mechanism

Therefore, our mechanism is that when a high quality child solution, which has the ability to replace most of the current solutions in *T*, appears, instead of randomly choosing n_r current solutions, we rank their distances to the high quality child solution in the objective space and replace the n_r solutions with the smallest distances.

C. *Random scale search in DE mutation*

MOEA/D-DE uses the DE/best/1/bin [8] mutation, which is as follows:

$$y' = x^i(t) + F(x^{r1}(t) - x^{r2}(t)) \tag{4}$$

where indices r_1 and r_2 ($r_1, r_2 \in P$) are randomly chosen and mutually different, and also different from the current index *i*. $F \in (0,1]$ is a constant called the scaling factor, which controls the amplification of the differential variation $x^{r1}(t) - x^{r2}(t)$.

DE is with no doubt, a very powerful search engine for single objective optimization. But when it comes to multiobjective problems, it seems to converge very fast to the vicinity of the true PF, but presents problems to actually reach it [11]. A recent study using a scaling factor that is uniformly distributed from 0.5 to 1 is shown to have higher successful rate to reach the global optimization point in single objective problems [12].

In this work, we use a Gaussian distributed random scaling factor with mean value μ and variance σ: $F_{i,k} = norm(\mu, \sigma)$, $i = 1, \ldots, N$ and $k = 1, \cdots, n$. For each variable in the search space, the scaling factor $F_{i,k}$ of each differential variation $x^{r1}(t) - x^{r2}(t)$ is different. $\hat{F}$ is continuously generated randomly in each iteration. Eqn. (4) is changed to:

$$y' = x^i(t) + \hat{F}(x^{r1}(t) - x^{r2}(t)) \tag{5}$$

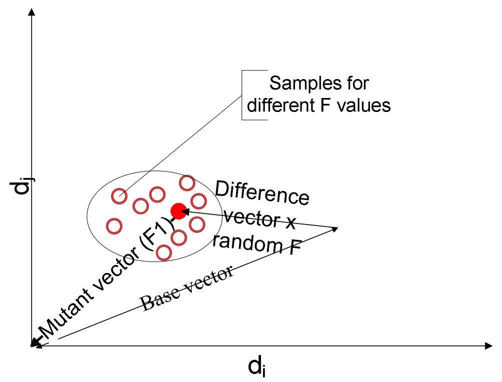


Fig. 2. Illustration of mutant vectors obtained by the random-scale operator

The random amplification induces two advantages: (1) The algorithm has a lower probability of providing premature solutions because of the reasonable diversity; (2) The vicinity of the mutant vector is investigated by the randomized amplification of the differential variation $x^{r1}(t)-x^{r2}(t)$. Even when stagnation appears, a new trial vector has fair chances of pointing at an even better location on the multimodal functional surface. Fig. 2 shows the effect of randomizing *F*. It can be seen that a cloud of potential points centered around the mutant vector could be generated.

IV. Experimental Results

To show the effects of the two mechanisms, we compare original MOEA/D-DE (OD), MOEA/D-DE with new replacement rules (RD), MOEA/D-DE with stochastic scaling factor (FD) and MOEA/D-DE with both new replacement rules and stochastic scaling factor (FRD). The test problem instances are UF1 to UF10 in CEC 2009 competition (2-3 objectives) [13] and a real world problem, sizing of folded-cascode amplifier (4 objectives).

A. Performance Metric

The inverted generational distance (IGD) [14] is used to assess the performance of the algorithms. Let P^* be a set of uniformly distributed points in the objective space along the PF. Let *A* be an approximation to the PF, the inverted generational distance from P^* to *A* is defined as:

$$IGD(A,P^*)=\frac{\sum_{v\in P^*} d(V,A)}{|P^*|} \tag{5}$$

where $d(v,A)$ is the minimum Euclidean distance between *v* and the points in *A*.

B. Test Problems and Parameter Setting

The test problems include benchmark problems and a four objective analog sizing problem. The benchmark problems are UF1 to UF10 in [13]. The multiobjective analog sizing is optimization of a folded-cascode amplifier (Fig.3), where the DC gain, GBW, phase margin and power are the 4 objectives. In the analog sizing problem, there is no analytical formulation of the optimization goals. They are based on the SPICE simulation. There are 11 design variables, 5 of which have a range of 0.24 μm to 100 μm, 5 of which have a range of 0.18 μm to 10 μm and 1 of which has a range of 1 μA to $2.5mA$.

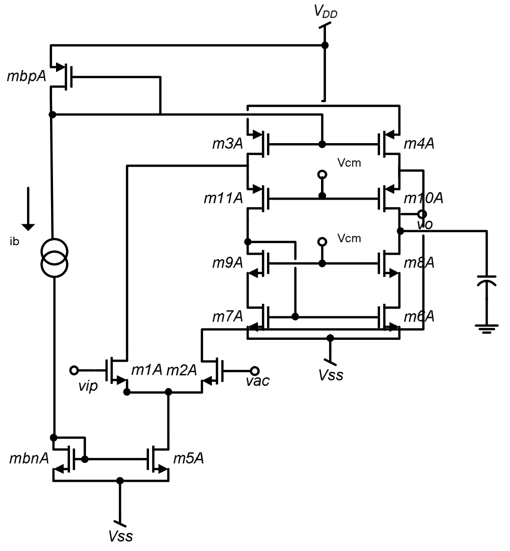


Fig. 3. Folded-cascode amplifier

For UF1 to UF10 in [13], the number of decision variables is 30. For the analog sizing problem, the number of design variables is 11. The number of sub-problems (population size), ***N***, is 300 for 2 objective problems, 500 for three objective problems and 148 for the analog sizing problem (though 4 objectives, considering the computational effort, *N* is reduced to 148). T is set to $0.1N$, n_r is set to $0.01N$, δ is set to 0.9. In DE operators, CR is set to 1, F is a Gaussian distributed vector with a mean of 0.5 and a variance of 0.15. In GA operators, η and p_m are the same as MOEA/D-DE. For benchmark problems, the algorithm stops after 1000 generations for 2 objective problems, and 1200 generations for 3 objective problems. For the analog sizing problem, the algorithm stops after 200 iterations.

C. Results

For UF1 to UF10 in [13], the set $P^*\in PF$ is available. For the analog sizing problem, 30 runs are first performed using each method, whose results are combined to approximate the P^* using the method in [10]. Table I shows the mean values of IGD results for each problem in 20 runs. The runs with smallest IGD values are drawn in Fig. 4.

Table I. The IGD statistics based on 20 runs (average values)

Tests	FRD	FD	RD	OD
UF1	0.0096	0.0064	0.0025	0.0027
UF2	0.0084	0.0072	0.0094	0.0098
UF3	0.0472	0.0311	0.0093	0.0105
UF4	0.0592	0.0788	0.0881	0.0858
UF5	0.5577	0.7650	0.8476	0.9247
UF6	0.1795	0.2726	0.2381	0.2665
UF7	0.0056	0.0063	0.0054	0.0032
UF8	0.0660	0.0611	0.0569	0.0562
UF9	0.1304	0.1299	0.1170	0.1501
UF10	0.4035	0.4370	0.4119	0.4781
Analog	9.4572	9.5344	9.5199	9.6079

D. *Discussions*

Here are some observations of the results. For each problem, we can rank the different methods according to the IGD values and get Table II and Table III.

Table II. Ranking of the IGD values

Tests	FRD	FD	RD	OD
UF1	Rank 4	Rank 3	Rank 1	Rank 2
UF2	Rank 2	Rank 1	Rank 3	Rank 4
UF3	Rank 4	Rank 3	Rank 1	Rank 2
UF4	Rank 1	Rank 2	Rank 4	Rank 3
UF5	Rank 1	Rank 2	Rank 3	Rank 4
UF6	Rank 1	Rank 4	Rank 2	Rank 3
UF7	Rank 3	Rank 4	Rank 2	Rank 1
UF8	Rank 4	Rank 3	Rank 2	Rank 1
UF9	Rank 3	Rank 2	Rank 1	Rank 4
UF10	Rank 1	Rank 3	Rank 2	Rank 4
Analog	Rank 1	Rank 2	Rank 3	Rank 4

Table III. Statistics of the ranking

Methods	Rank 1	Rank 2	Rank 3	Rank 4
FRD	5	1	2	3
FD	1	4	4	2
RD	3	4	3	1
OD	2	2	2	5

It can be seen that the improvement of the new replacement mechanism is obvious. In 7 cases out of 11, the RD (MOEA/D-DE with new replacement) method ranks 1 or 2, FRD (RD plus random scaling factor) method has 6 cases with rank 1 or 2, FD (MOEA/D-DE with random-scale F) have 5 cases with rank 1 or 2 and the original MOEA/D-DE has 4 cases. If only considering the rank 1 column, it can be seen that RD and FRD have more distinct advantages.

If only adding a random scaling factor, slight improvements have been observed in high rank region (rank 1 or 2). But we can see that the FD method ranks 3 in 4 cases and ranks 4 in 2 case, while the original MOEA/D-DE (OD) ranks 3 in 2 cases, and ranks 4 in 5 cases.

When the two mechanisms are combined together, it can be seen that FRD have 5 cases with rank 1, which has distinct advantage compared with other methods. On the other hand, it has 3 cases with rank 4. Therefore, we can conclude that FRD is a method which can obtain very good result, and RD method is more stable.

E. *NSGA-II Result of the folded-cascode amplifier*

NSGA-II is also implemented for the analog sizing problem using the same population size, η and p_m. The distribution index in SBX is set to 20. The average IGD value is 15.8656, which is much larger than MOEA/D-based methods.

V. Conclusions

In this paper, two extensions of the MOEA/D-DE framework are investigated. The new replacement mechanism uses the good information from high quality child solution better than the original random selection replacement mechanism, while at the same time keep the diversity by not increasing n_r. The random scaling factor improves the search ability of the MOEA/D-DE. When these two techniques are combined, higher performance can be obtained. Future works will concentrate on applying the enhanced MOEA/D-DE framework(s) to more real world applications.

Acknowledgment

This research was supported by a special bilateral agreement scholarship of Katholieke Universiteit Leuven, Belgium and Tsinghua University, P. R. China, and by the TIC-2532 Project, funded by Consejeria de Innovación, Ciencia y Empresa, Junta de Andalucia, Spain. We sincerely thank Prof. Dr. Günhan Dündar, Bogazici University, Turkey.

References

[1] C. Hwang and A. Masud, "Multiple objective decision making—methods and applications", Springer-Verlag, 1979.

[2] K. Deb, "Multiobjective optimization using evolutionary algorithms", New York: Wiley, 2001.

[3] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, 2002. pp. 182-197.

[4] E. Zitzler, M. Laumanns and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm", Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Switzerland, 103,2001.

[5] Q. Zhang, H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition", IEEE Transactions on Evolutionary Computation, 2007. pp. 1-20.

[6] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II", IEEE Transactions on Evolutionary Computation, 2008. pp. 1-19.

[7] K. Deb and R. Agrawal, "Simulated binary crossover for continuous search space", Complex Systems 9, 1995, pp 115-148.

[8] K. Price, R. Storn and J. Lampinen, Differential Evolution. A Practical Approach to Global Optimization. Springer, 2005.

[9] K. Miettinen, "Nonlinear multiobjective optimization", Norwell, MA: Kluwer, 1999.

[10] Q. Zhang W. Liu and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances", IEEE Congress on Evolutionary Computation, 2009. pp. 203-208.

[11] C. Coello Coello, G. Lamont and D. Weldhuizen, Evolutionary algorithms for solving multiobjective problems, Springer, 2007.

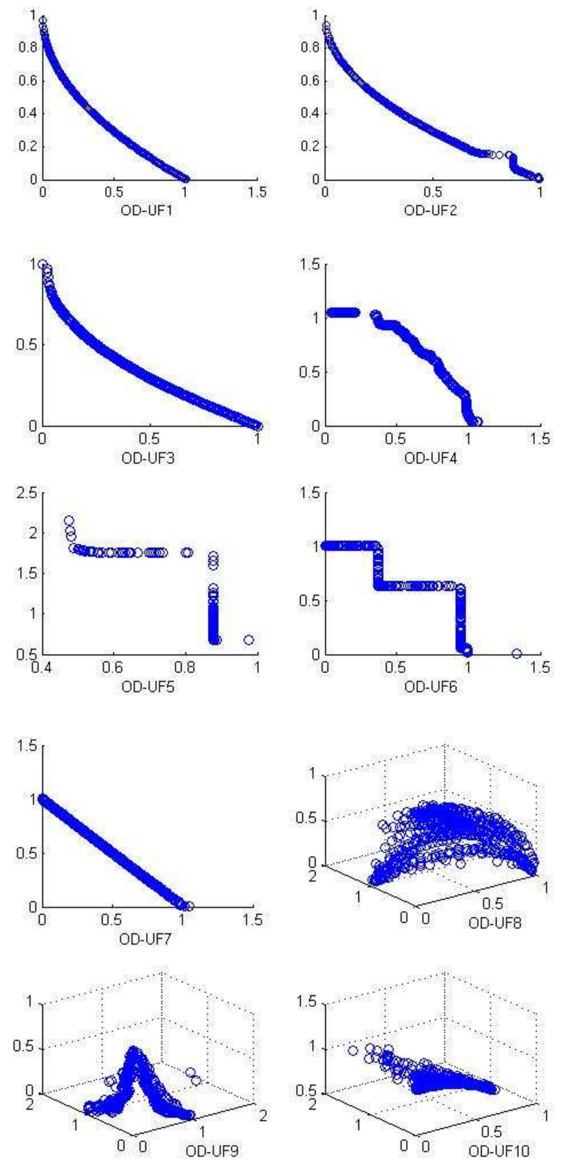
[12] S. Das, A. Konar and U. Chakraborty, "Two improved differential evolution schemes for faster global search", Genetic and Evolutionary Computation Conference, 2005. pp. 991-998.

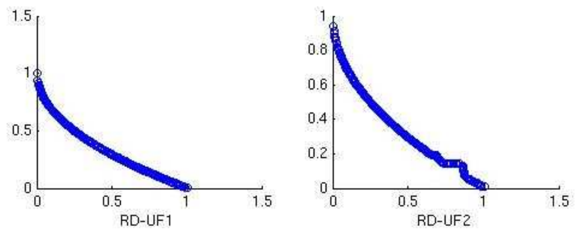
[13] CEC 09 MOEA Competition, http://dces.essex.ac.uk/staff/qzhang/moeacompetition09.htm

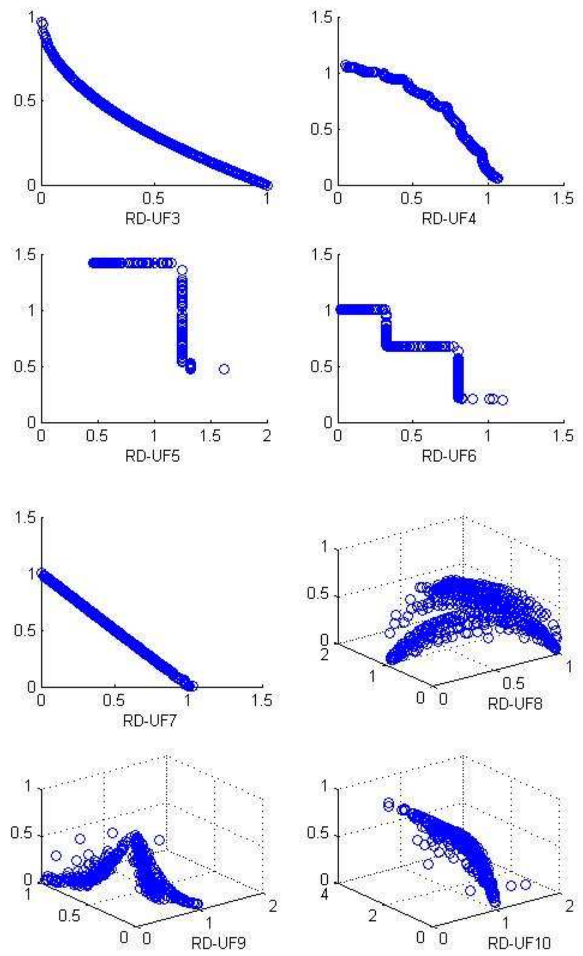
[14] Z. Zitzler, L. Thiele, M. Laumanns, C. Fonseca and V. Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review", IEEE Transactions on Evolutionary Computation, 2003, pp. 117-132.

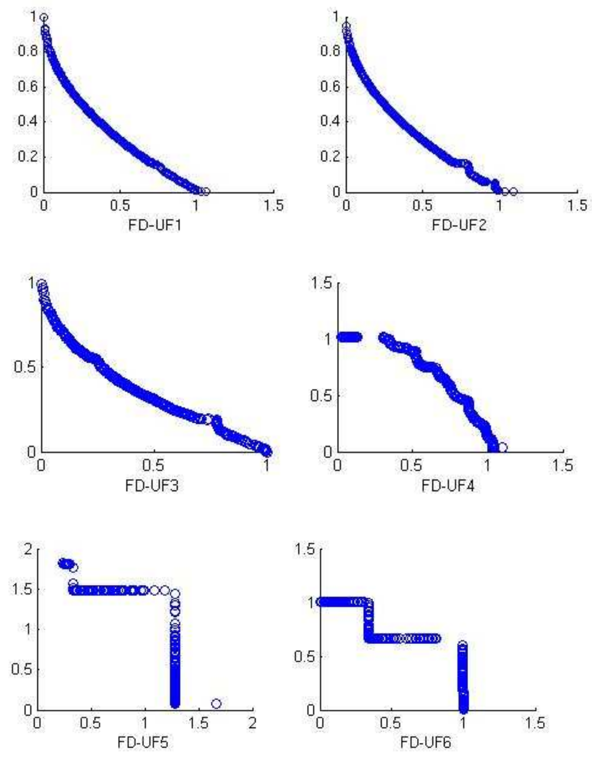
[15] P. Bentley and J. Wakefield, "Finding acceptable solutions in the Pareto-optimal range using multiobjective genetic algorithms", 2nd On-Line World Conference on Soft Computing, Engineering Design and Manufacturing, 1997. pp. 231-240.

[16] M. Gen and R. Cheng, Genetic algorithms and engineering design. New York: Wiley, 1997.


OD-UF1
OD-UF2
OD-UF3
OD-UF4
OD-UF5
OD-UF6
OD-UF7
OD-UF8
OD-UF9
OD-UF10


RD-UF1
RD-UF2


RD-UF3
RD-UF4
RD-UF5
RD-UF6
RD-UF7
RD-UF8
RD-UF9
RD-UF10


FD-UF1
FD-UF2
FD-UF3
FD-UF4
FD-UF5
FD-UF6

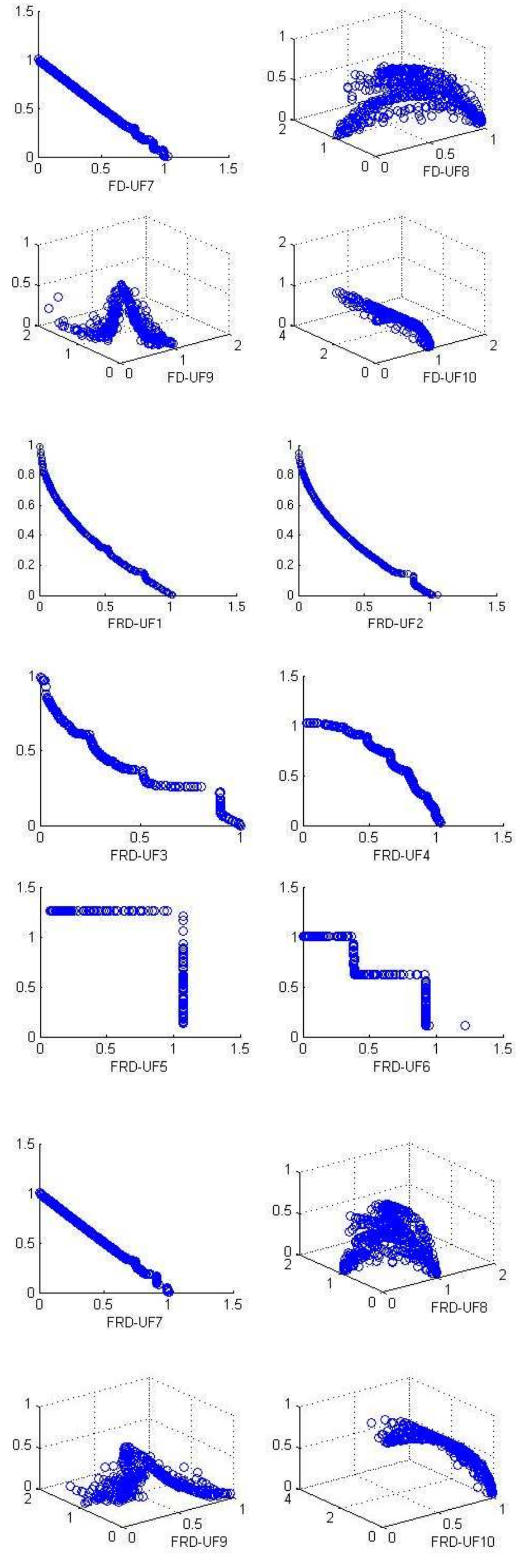


Fig. 4. PF with the smallest IGD values by different methods