Performance Assessment of DMOEA-DD with CEC 2009 MOEA Competition Test Instances

Minzhong Liu, Xiufen Zou, Yu Chen, Zhijian Wu

Abstract—In this paper, the DMOEA-DD, which is an improvement of DMOEA[1, 2] by using domain decomposition technique, is applied to tackle the CEC 2009 MOEA competition test instances that are multiobjective optimization problems (MOPs) with complicated Pareto set (PS) geometry shapes. The performance assessment is given by using IGD [3, 4] as performance metric.

I. INTRODUCTION

During the last 20 years, evolutionary computation techniques have been successfully used to solve a number of multiobjective optimization problems (MOPs). However, a lot of multiobjective evolutionary algorithms (MOEAs) may loose their effectiveness when MOPs are scalable with respect to the objectives, the decision variables and constraints with complex Pareto shape in the decision space [3, 4]. It forces us to develop new algorithms to face these great challenges. In this paper, we propose an improved algorithm, named as "DMOEA-DD". The domain decomposition technique is used to divide the feasible domain of decision variables into several subdomains, which is similar to literatures [5, 6]. In each subdomain, the DMOEA[1, 2] is used to search the pareto optimal solutions and each subdomain can exchange the information by genetic operators. The DMOEA-DD is applied to solve thirteen unconstraint problems and ten constraint problems, which are proposed in the CEC 2009 test suite [7].

II. DESCRIPTION OF ALGORITHM

Without loss of generality, we consider the following multi-objective optimization problem (MOP), and assume that all objectives are to be minimized

$$\underset{X \in D}{Minimize}\; G(X) = \{g_1(X), g_2(X), \cdots, g_M(X)\} \qquad (1)$$

where D is feasible domain of decision variables, M is the number of objectives.

A. The Dynamical Multiobjective Evolutionary Algorithm (DMOEA)

(1) The Fitness Assignment Strategy

The dynamical multiobjective evolutionary algorithm (DMOEA), i.e. the method based on principle of the minimal free energy in thermodynamics, was first described by Zou *et al*. [1, 2]. DMOEA adopted an aggregating function that combined ranking with entropy and density. DMOEA focused on non-dominated solutions and simultaneously maintained diversity in the solutions that converted the original multiple objectives into a new fitness to a solution. The fitness was composed of three parts:

$$fitness(i) = R(i) - TS(i) - d(i) \qquad (2)$$

where $fitness(i)$ denotes the fitness of the individual i in the population.

The first part $R(i)$ is the Pareto-rank value of individual i, which is equal to the number of solution n_i that dominates solution $\boldsymbol{i}$. The Pareto-rank values can be computed as follows:

$$R(i) = |\Omega_i| \qquad (3)$$

where $\Omega_i = \{X_j \mid X_j \succ_p X_i, 1 \le j \le N, j \ne i\}$.

The second part $S(i) = -p_T(i)\log p_T(i)$, where $T > 0$ is the analog of temperature. $p_T(i) = 1/Z\exp(-R(i)/T)$ is the analogue of the Gibbs distribution, $Z = \sum_{i=1}^{N}\exp(-R(i)/T)$ is called the partition function, and N is the population size.

The third part $d(i)$ is the crowding distance, which is computed by using a density estimation technique that is described in Deb's NSGA-II [8].

In the proposed algorithm, the new fitness of all individuals are sorted in increasing order, and the individuals whose fitness are biggest are recorded, called as "worst individuals", however, the rank values that are equal to zero are considered as "best individuals". Concepts of best and worst individuals in the proposed algorithm are different from those in conventional MOEAs. Moreover, only the worst individuals are eliminated through selections in each step of DMOEA, which can ensure the good diversity of population.

(2) The Selection Criterion

During the selection process, the new individuals only compared with the worst individuals, and used the Metropolis criterion of simulated annealing algorithm (SA) and the crowding distance to guide the select process, that is,

(1) If $R(X_{new}) < R(X_{worse})$, then $X_{worst} := X_{new}$;

(2) If $R(X_{new}) = R(X_{worse})$ and $d(X_{new}) > d(X_{worse})$, then $X_{worst} := X_{new}$;

Manuscript received December 3, 2008. This work was supported by the Natural Science Foundation of China under Grants 50677046 and 60573168.
M. Liu is with School of Computer Sciences, Wuhan University, Wuhan , 430072, China.

X. Zou and Y. Chen are with the School of Mathematics and Statistics, Wuhan University, Wuhan, 430072, China (X. Zou, corresponding author, 8627-13397177528, e-mail: xfzou@whu.edu.cn).

Z. Wu is with the State Key Laboratory of Software Engineering, Wuhan University, Wuhan , 430072, China.

(3) else if $\exp((R(X_{worst}) - R(X_{new}))/T) > random(0,1)$, then $X_{worst} := X_{new}$;

where $R(X_{worst})$ and $R(X_{new})$ are respectively the Pareto-rank value of the worst individuals and the new individuals that are computed by using equation (3).

Algorithm 1 (DMOEA)

1 Randomly generate the initial population $P(0)$, and set generation index $t = 0$;

2 REPEAT

(a) Calculate the Pareto-rank values $\{R_1(t), R_2(t), \ldots, R_L(t)\}$ of all individuals in $P(t)$ according to equation (3), L is the population size;

(b) Evaluate the fitness of each individual in $P(t)$ according to equation (2) , sort them in increasing order, and record the worst individuals;

(c) Save the individuals whose Pareto-rank values are equal to zero to form Pareto front.

(d) Apply multi-parent crossover to generate new individuals. Select new individuals based on the selection criterion in this section;

(e) Use uniform mutation to each individual in the population, if the new individual is better than the original individual, then the original individual is replaced;

(f) Form the new population $P(t+1)$ by (d) and (e);

(g) $t{:=}t+1$;

UNTIL the stopping criteria fulfilled;

3 Output the Pareto front.

B. The Parallel Evolutionary Algorithm (DMOEA-DD)

The whole feasible domain of decision variables D is divided into N subdomains $D_1, D_2, \cdots, D_N$, and $D_i \cap D_{i+1} = S_i$ ($S_i=\phi$ represents that there is no overlapping between D_i and D_{i+1}). The communication between different subdomains can be implemented by generic operation between these subdomains. The detailed algorithm is described as follows.

Algorithm 2 (DMOEA-DD)

1 Split the feasible domain D into between different subdomains;

2 REPEAT

(a) Do i=1,2,…, N in parallel

Use **Algorithm 1 (DMOEA)** for each D_i

(b) Select m individuals from each subdomain D_i to do multi-parent crossover to generate new individuals;

(c) If new individuals belongs to the overlapping domains, they will be sent to each subdomain, or they will be sent to the specific subdomain.

UNTIL the stopping criteria fulfilled.

Remark1: the whole domain are equally divided and the number of subdomains can be determined according to the number of the parallel virtual machine.

TABLE I

THE MEAN, STANDARD DEVIATION, THE SMALLEST AND THE LARGEST VALUES OF THE IGD VALUES OF THE 30 FINAL APPROXIMATIONS OBTAINED AND THE AVERAGE CPU TIME USED FOR EACH UNCONSTRAINED TEST INSTANCE

Problem	Mean and Std.of IGD	the smallest IGD	the largest IGD	Average CPU time(s)
UF1	0.010384 ±0.002367	0.006467	0.015170	27.786458
UF2	0.006791 ±0.002017	0.004568	0.014616	29.720312
UF3	0.033370 ±0.005680	0.024710	0.046058	27.223958
UF4	0.042686 ±0.001386	0.039194	0.046727	30.476563
UF5	0.314543 ±0.046598	0.230466	0.395479	24.941146
UF6	0.066732 ±0.023803	0.043126	0.179191	24.892187
UF7	0.010324 ±0.002254	0.006178	0.015156	28.204167
UF8	0.068419 ±0.009462	0.052962	0.089636	39.398438
UF9	0.048965 ±0.009115	0.038484	0.077495	38.975521
UF10	0.322118 ±0.022272	0.286431	0.372509	35.352083
UF11	1.203282 ±0.071333	1.039853	1.347115	114.844792
UF12	477.656326 ±93.47138	350.6535	652.112515	118.859896
UF13	1.997199 ±0.011466	1.973544	2.014546	107.572396

TABLE II

THE MEAN, STANDARD DEVIATION, THE SMALLEST AND THE LARGEST VALUES OF THE IGD VALUES OF THE 30 FINAL APPROXIMATIONS OBTAINED AND THE AVERAGE CPU TIME USED FOR EACH CONSTRAINED TEST INSTANCE

CF1	0.011311 ±0.002758	0.007061	0.016932	21.526563
CF2	0.002100 ±0.000453	0.001579	0.003063	28.985417
CF3	0.056305 ±0.007573	0.038062	0.070731	26.063021
CF4	0.006995 ±0.001457	0.005523	0.011552	27.846354
CF5	0.015773 ±0.006662	0.007872	0.039396	24.404167
CF6	0.015020 ±0.006462	0.006186	0.031120	26.489062
CF7	0.019051 ±0.006123	0.010424	0.033821	24.213542
CF8	0.047501 ±0.006387	0.038785	0.065014	35.584896
CF9	0.143432 ±0.021416	0.119107	0.206549	35.484896
CF10	0.162128 ±0.031621	0.098377	0.239399	32.967188

Remark2: Average CPU time in the Table I and Table II include the time that calculates the all IGD values in each 50 generations.

III. PERFORMANCE ASSESSMENTS

In this paper, we consider the 13 unconstraint problems and 10 constraint problems, UF1-UF13 and CF1-CF10, which are proposed in the CEC 2009 test suite [7].

A. Performance Metric (IGD)

Let P^* be a set of uniformly distributed points along the PF (in the objective space). Let A be an approximate set to the PF, the average distance from P^* to A is defined as [3]:

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

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

B. PC Configuration

– System: Windows XP_SP3
– RAM: 1.59GHz,1.96GB
– CPU: Intel Core(TM)2 DUO CPU T8100 2.10GHz
– Computer Language:C++

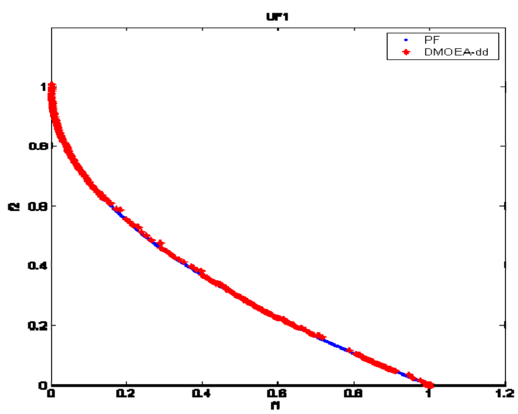

Fig.1. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on UF1.

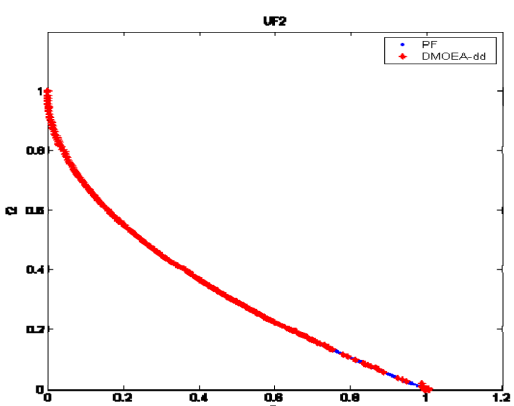

Fig.2. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on UF2.

C. Algorithmic Parameter Setting

The parameter settings are as follows.
Number of decision variables: It is set to be 30 for all test instances.
Number of population size: The number of the whole population size is set to be 300 for biobjective and triobjective problems, and is set to be 600 for five objective problems.

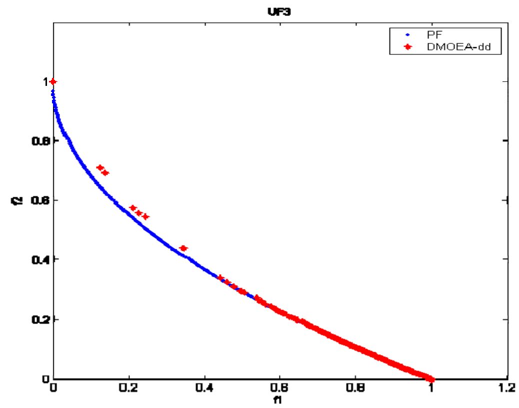

Fig.3. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on UF 3.

Parameter settings in DMOEA: the number of multi-parent crossover is 5 and the number of the new individuals by crossover is 10 at each generation. The mutation rate is 1/n (that is, the one variable of each individual is mutated, n is number of decision variables). Temperature is $T=10000$. The number of subdomains N is 2.
Initialization: Initial populations in all the algorithms are randomly generated.
Stopping condition: The two algorithms stop after 300, 000 function evaluations for each test instance.

All the following results are based on 30 independent runs of DMOEA-DD on each test instance. We select at most 100 final nondominated solutions for the calculation of IGD metric values for two-objective problems, and 150 solutions for three-objective problems, 800 solutions for five-objective problems.

D. Experimental Results

Table I and Table II show the statistical results of IGD and the average UPU time by DMOEA-DD on UF1-UF13, CF1-CF10 respectively. Comparisons of the final approximation set with the smallest IGD value by DMOEA-DD in 30 independent runs and Pareto front on UF1-UF10, CF1-CF10 are drawn in Figs. 1 to 20 respectively.

IV. CONCLUSION

In this paper, we use CEC 2009 MOEA competition test instances and the performance metric (IGD) to assess the performance of DMOEA-DD. The results indicated that DMOEA-DD could successfully solve some problems. But some problems are difficult to solve. Future work will improve DMOEA-DD or find new algorithm to deal with a class of MOPs presented in Ref. [7].

REFERENCES

[1] X. Zou, Y. Chen, M.Liu and L.Kang, “A New Evolutionary Algorithm for Solving Many-objective Optimization Problems,” *IEEE Trans. on System, Man and Cybernetics, Part B,* vol.38, pp.1402-1412, 2008.

[2] X. Zou, M. Liu, L. Kang, and J. He, “A high performance multi-objective evolutionary algorithm based on the principles of thermodynamics”, in *Proc. Parallel Problem Solving from Nature 8th Int. Conf.*, vol. 3242, LNCS, X. Yao, E. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J. Rowe, P. Tino, A. Kabán, and H.-P.Schwefel, Eds., Berlin, Germany: Springer-Verlag, Sep. 2004, pp.922-931.

[3] H. Li and Q. Zhang, “Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGAII,” *IEEE Trans. Evol. Comp., Accepted*, 2008.

[4] A. Zhou, Q. Zhang and Y. Jin, “Approximating the Set of Pareto Optimal Solutions in Both the Decision and Objective Spaces by an Estimation of Distribution Algorithm,” Working Report CES-485, Dept of CES, University of Essex, 06/2008.

[5] Q. Zhang and H. Li, “MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition, IEEE Trans. on Evolutionary Computation”, vol.11, no. 6, pp712-731, 2007.

[6] Z. Yang, K. Tang and X. Yao, "Differential Evolution For High-Dimensional Function Optimization," Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, pp.3523-3530, 25-28 Sept. 2007

[7] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, “Multiobjective optimization test instances for the CEC 2009 special session and competition,” University of Essex and Nanyang Technological University, Tech. Rep. CES-487, 2008.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multi-objective genetic algorithm: NSGA-II,” IEEE Trans. Evol. Comput., vol. 6, no. 2, pp. 182–197, Apr. 2002.

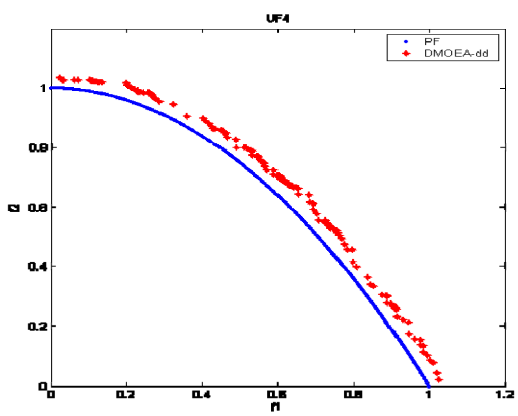

Fig.4. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on UF4.

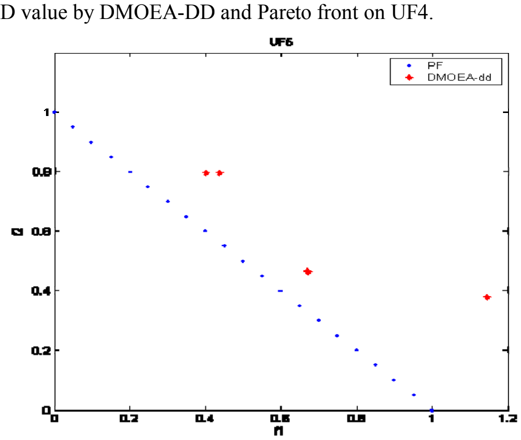

Fig. 5. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on UF5.

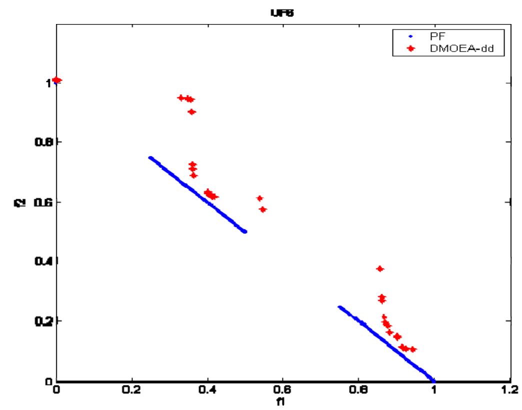

Fig. 6. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on UF6.

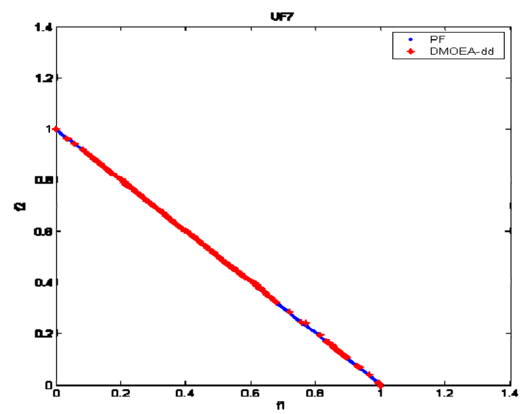

Fig. 7. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on UF7.

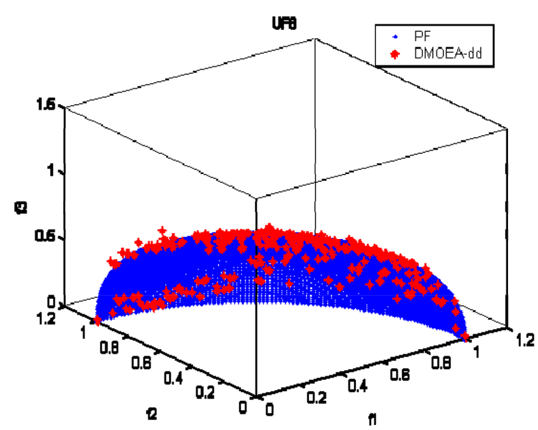

Fig. 8. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on UF8.

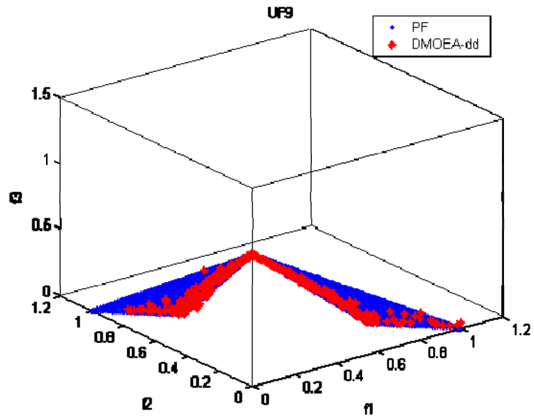

Fig.9. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on UF9.

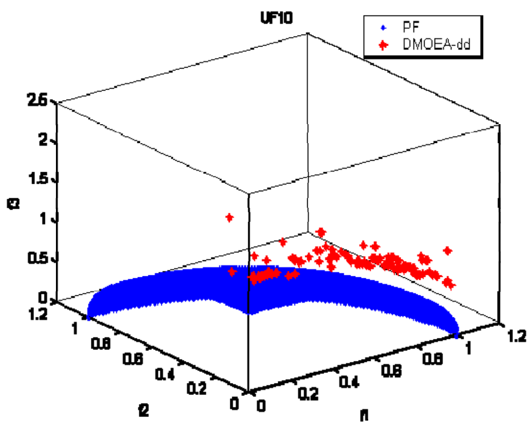

Fig.10. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on UF10.

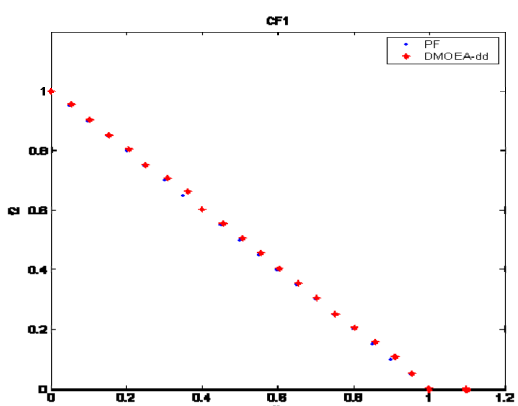

Fig.11. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on CF1.

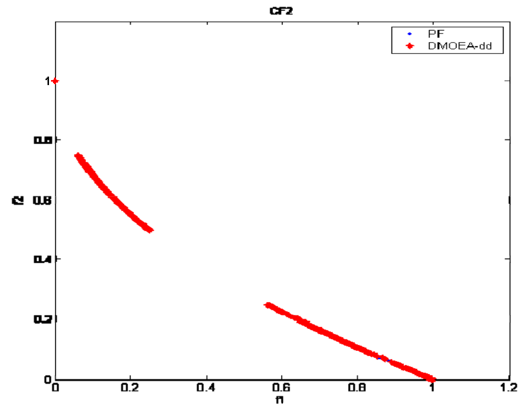

Fig.12. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on CF2.

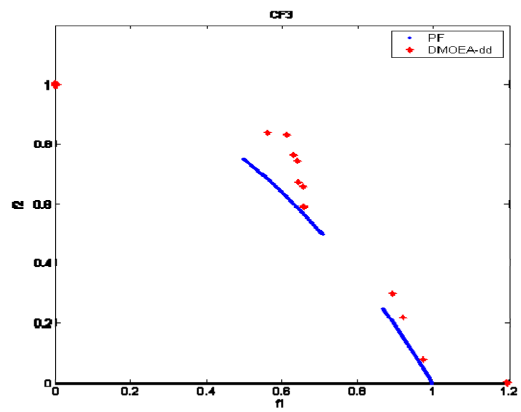

Fig.13. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on CF3.

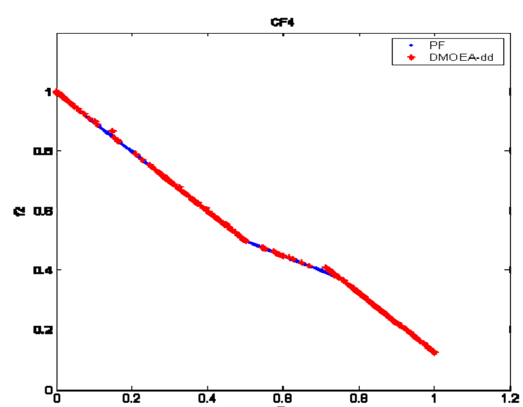

Fig.14. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on CF4.

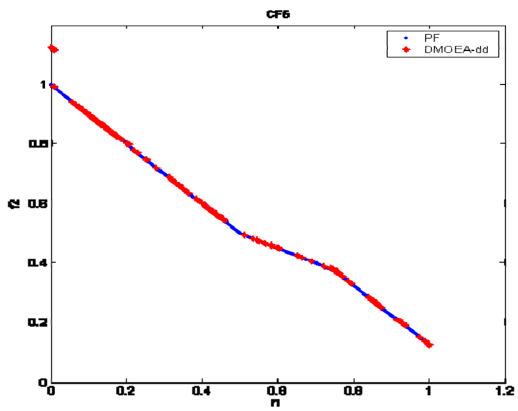

Fig.15. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on CF5.

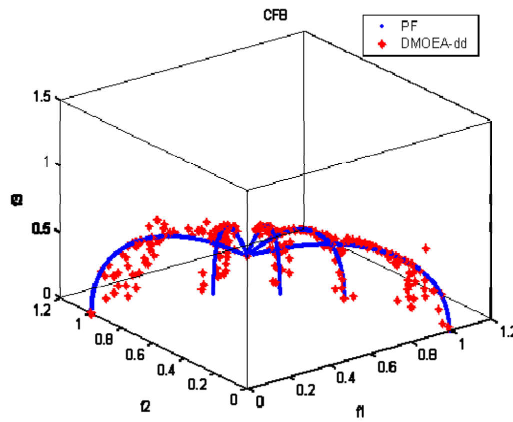

Fig.18. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on CF8.

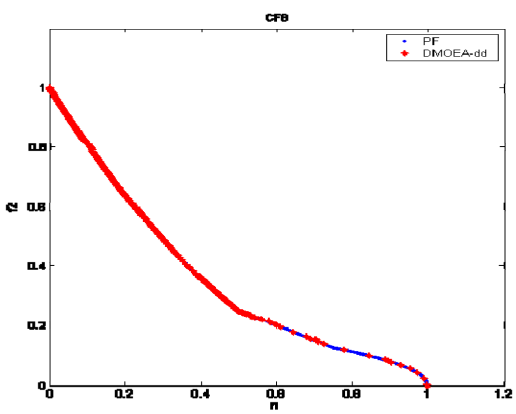

Fig.16. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on CF6.

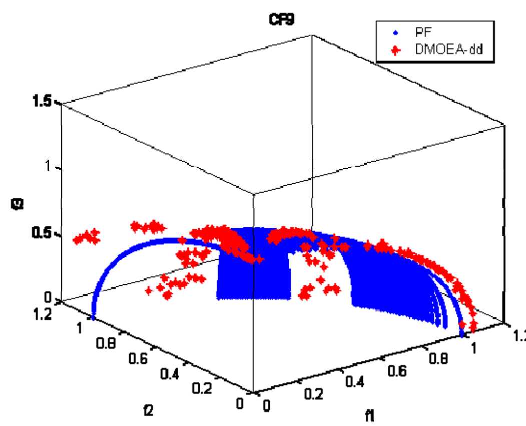

Fig.19. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on CF9.

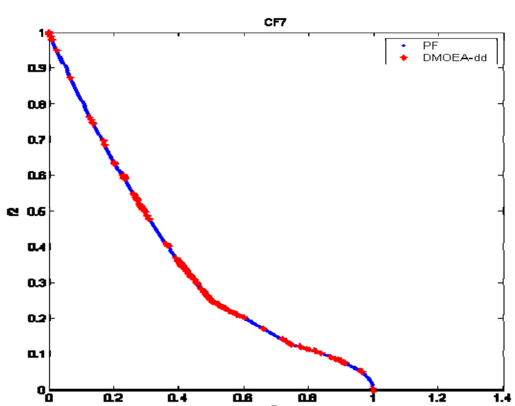

Fig.17. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on CF7.

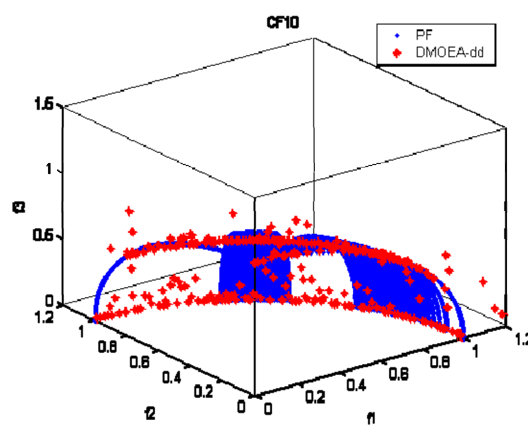

Fig.20. Comparison of the final approximation set with the smallest IGD value by DMOEA-DD and Pareto front on CF10.