The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances

Qingfu Zhang, Wudong Liu and Hui Li

***Abstract*—This paper describes the idea of MOEA/D and proposes a strategy for allocating the computational resource to different subproblems in MOEA/D. The new version of MOEA/D has been tested on all the CEC09 unconstrained MOP test instances.**

***Index Terms*—MOEA/D, Test problems, Multiobjective optimization.**

I. Introduction

A *multiobjective optimization problem* (MOP) can be stated as follows:

$$\begin{array}{ll} \text{minimize} & F(x) = (f_1(x), \ldots, f_m(x))^T \\ \text{subject to} & x \in \Omega \end{array} \quad (1)$$

where Ω is the *decision (variable) space*, $F : \Omega \rightarrow R^m$ consists of m real-valued objective functions and R^m is called the *objective space*. If $x \in R^n$, all the objectives are continuous and Ω is described by

$$\Omega = \{x \in R^n | h_j(x) \leq 0, j = 1, \ldots, k\},$$

where h_j are continuous functions, we call (1) a *continuous MOP*. Very often, since the objectives in (1) contradict one another, no point in Ω can minimize all the objectives simultaneously. One has to balance them. The best tradeoffs among the objectives can be defined by Pareto optimality.

Let $u, v \in R^m$, u is said to *dominate* v if and only if $u_i \leq v_i$ for every $i \in \{1, \ldots, m\}$ and $u_j < v_j$ for at least one index $j \in \{1, \ldots, m\}$. A point $x^* \in \Omega$ is *Pareto optimal* if there is no point $x \in \Omega$ such that $F(x)$ dominates $F(x^*)$. $F(x^*)$ is then called a *Pareto optimal (objective) vector*. In other words, any improvement in a Pareto optimal point in one objective must lead to deterioration to at least one other objective. The set of all the Pareto optimal points is called the *Pareto set* (PS) and the set of all the Pareto optimal objective vectors is the *Pareto front* (PF) [1].

Recent years have witnessed significant progress in the development of evolutionary algorithms (EAs) for dealing with MOPs. Multiobjective evolutionary algorithms (MOEAs) aim at finding a set of representative Pareto optimal solutions in a single run. Most MOEAs are Pareto dominance based, they adopt single objective evolutionary algorithm frameworks and the fitness of each solution at each generation is mainly determined by its Pareto dominance relations with other solutions in the population. NSGA-II [2], SPEA-II [3] and PAES [4] are among the most popular Pareto dominance based MOEAs.

A Pareto optimal solution to an MOP could be an optimal solution of a single objective optimization problem in which the objective is a linear or nonlinear aggregation function of all the individual objectives. Therefore, approximation of the PF can be decomposed into a number of single objective optimization problems. some MOEAs such as MOGLS [5]–[7] and MSOPS [8] adopt this idea to some extent. MOEA/D [9] (MultiObjective Evolutionary Algorithm based on Decomposition) is a very recent evolutionary algorithm for multiobjective optimization using the decomposition idea. It has been applied for solving a number of multiobjective optimization problems [10]–[14].

The rest of this paper is organized as follows. Section II introduces a new version of MOEA/D. Section III presents experimental results of MOEA/D on the 13 unconstrained MOP test instances for CEC 2009 MOEA competition [15]. Section IV concludes the paper.

II. MOEA/D

MOEA/D requires a decomposition approach for converting the problem of approximation of the PF into a number of scalar optimization problems. In this paper, we use the Tchebycheff approach.

A. Tchebycheff Approach [1]

In this approach, the scalar optimization problems are in the form

$$\begin{array}{ll} \text{minimize} & g^{te}(x|\lambda, z^*) = \max_{1 \leq i \leq m}\{\lambda_i | f_i(x) - z_i^*|\} \\ \text{subject to} & x \in \Omega \end{array} \quad (2)$$

where $z^* = (z_1^*, \ldots, z_m^*)^T$ is the reference point, i. e. $z_i^* = \min\{f_i(x) | x \in \Omega\}$ for each $i = 1, \ldots, m$. Under some mild conditions [1], for each Pareto optimal point x^*, there exists a weight vector λ such that x^* is the optimal solution of (2) and each optimal solution of (2) is a Pareto optimal solution of (1). Therefore, one is able to obtain different Pareto optimal solutions by solving a set of single objective optimization problem defined by the Tchebycheff approach with different weight vectors.

B. MOEA/D with Dynamical Resource Allocation

Let $\lambda^1, \ldots, \lambda^N$ be a set of even spread weight vectors and z^* be the reference point. As shown in Section II, the problem of approximation of the PF of (1) can be decomposed into N

Q. Zhang and W. Liu are with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, CO4 3SQ, U.K {qzhang, wliui}@essex.ac.uk.

H. Li is with Department of Computer Science, University of Nottingham, Nottingham, NG8 1BB, U. K

978-1-4244-2959-2/09/$25.00 © 2009 IEEE

scalar optimization subproblems and the objective function of the j-th subproblem is:

$$g^{te}(x|\lambda^j, z^*) = \max_{1\le i\le m}\{\lambda_i^j|f_i(x) - z_i^*|\} \quad (3)$$

where $\lambda^j = (\lambda_1^j, \ldots, \lambda_m^j)^T$, and $j = 1, \ldots, N$.

MOEA/D minimizes all these N objective functions simultaneously in a single run. Neighborhood relations among these single objective subproblems are defined based on the distances among their weight vectors. Each subproblem is optimized by using information mainly from its neighboring subproblems. In the versions of MOEA/D proposed in [9] and [10], all the subproblems are treated equally, each of them receives about the same amount of computational effort. These subproblems, however, may have different computational difficulties, therefore, it is very reasonable to assign different amounts of computational effort to different problems. In MOEA/D with Dynamical Resource Allocation (MOEA/D-DRA), the version of MOEA/D proposed in this paper, we define and compute a utility π^i for each subproblem i. Computational efforts are distributed to these subproblems based on their utilities.

During the search, MOEA/D-DRA with the Tchebycheff approach maintains:

- a population of N points $x^1, \ldots, x^N \in \Omega$, where x^i is the current solution to the i-th subproblem;
- $FV^1, \ldots, FV^N$, where FV^i is the F-value of x^i, i.e. $FV^i = F(x^i)$ for each $i = 1, \ldots, N$;
- $z = (z_1, \ldots, z_m)^T$, where z_i is the best (lowest) value found so far for objective f_i;
- $\pi^1, \ldots, \pi^N$: where π^i utility of subproblem i.
- gen: the current generation number.

The algorithm works as follows:

Input:
- MOP (1);
- a stopping criterion;
- N : the number of the subproblems considered in MOEA/D;
- a uniform spread of N weight vectors: $\lambda^1, \ldots, \lambda^N$;
- T : the number of the weight vectors in the neighborhood of each weight vector.

Output: $\{x^1, \ldots, x^N\}$ and $\{F(x^1), \ldots, F(x^N)\}$

Step 1 Initialization

Step 1.1 Compute the Euclidean distances between any two weight vectors and then find the T closest weight vectors to each weight vector. For each $i = 1, \ldots, N$, set $B(i) = \{i_1, \ldots, i_T\}$ where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the T closest weight vectors to λ^i.

Step 1.2 Generate an initial population $x^1, \ldots, x^N$ by uniformly randomly sampling from the search space.

Step 1.3 Initialize $z = (z_1, \ldots, z_m)^T$ by setting $z_i = \min\{f_i(x^1), f_i(x^2), \ldots, f_i(x^N)\}$.

Step 1.4 Set $gen = 0$ and $\pi^i = 1$ for all $i = 1, \ldots, N$.

Step 2 Selection of Subproblems for Search: the indexes of the subproblems whose objectives are MOP individual objectives f_i are selected to form initial I. By using 10-tournament selection based on π^i, select other $[\frac{N}{5}] - m$ indexes and add them to I.

Step 3 For each $i \in I$, do:

Step 3.1 Selection of Mating/Update Range: Uniformly randomly generate a number $rand$ from $(0, 1)$. Then set

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

Step 3.2 Reproduction: Set $r_1 = i$ and randomly select two indexes r_2 and r_3 from P, and then generate a solution $\bar{y}$ from x^{r_1}, x^{r_2} and x^{r_3} by a DE operator, and then perform a mutation operator on $\bar{y}$ with probability p_m to produce a new solution y.

Step 3.3 Repair: If an element of y is out of the boundary of Ω, its value is reset to be a randomly selected value inside the boundary.

Step 3.4 Update of z: For each $j = 1, \ldots, m$, if $z_j > f_j(y)$, then set $z_j = f_j(y)$.

Step 3.5 Update of Solutions: Set $c = 0$ and then do the following:

(1) If $c = n_r$ or P is empty, go to **Step 4**. Otherwise, randomly pick an index j from P.

(2) If $g(y|\lambda^j, z) \le g(x^j|\lambda^j, z)$, then set $x^j = y$, $FV^j = F(y)$ and $c = c + 1$.

(3) Delete j from P and go to (1).

Step 4 Stopping Criteria If the stopping criteria is satisfied, then stop and output $\{x^1, \ldots, x^N\}$ and $\{F(x^1), \ldots, F(x^N)\}$.

Step 5 $gen = gen + 1$.

If gen is a multiplication of 50, then compute Δ^i, the relative decrease of the objective for each subproblem i during the last 50 generations, update

$$\pi^i = \begin{cases} 1 & \text{if } \Delta^i > 0.001; \\ (0.95 + 0.05\frac{\Delta^i}{0.001})\pi^i & \text{otherwise.} \end{cases}$$

endif

Go to **Step 2**.

In 10-tournament selection in **Step 2**, the index with the highest π^i value from 10 uniformly randomly selected indexes are chosen to enter I. We should do this selection $[\frac{N}{5}] - m$ times.

In **Step 5**, the relative decrease is defined as

$$\frac{\text{old function value-new function value}}{\text{old function value}}$$

If Δ^i is smaller than 0.001, the value of π^i will be reduced.

In the DE operator used in **Step 3.2**, each element $\bar{y}_k$ in $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_n)^T$ is generated as follows:

$$\bar{y}_k = \begin{cases} x_k^{r_1} + F \times (x_k^{r_2} - x_k^{r_3}) \text{ with probability } CR, \\ x_k^{r_1}, \qquad\qquad \text{with probability } 1 - CR, \end{cases} \quad (4)$$

where CR and F are two control parameters.

The mutation operator in Step 3.2 generates $y = (y_1, \ldots, y_n)^T$ from $\bar{y}$ in the following way:

$$y_k = \begin{cases} \bar{y}_k + \sigma_k \times (b_k - a_k) & \text{with probability } p_m, \\ \bar{y}_k & \text{with probability } 1 - p_m, \end{cases} \quad (5)$$

with

$$\sigma_k = \begin{cases} (2 \times rand)^{\frac{1}{\eta+1}} - 1 & \text{if } rand < 0.5, \\ 1 - (2 - 2 \times rand)^{\frac{1}{\eta+1}} & \text{otherwise,} \end{cases}$$

where $rand$ is a uniformly random number from $[0, 1]$. The distribution index η and the mutation rate p_m are two control parameters. a_k and b_k are the lower upper bounds of the k-th decision variable, respectively.

III. Experimental Results

MOEA/D has been tested on all the 13 unconstrained test instances in CEC 2009 [15].

The parameter settings are as follows:

- N: 600 for two objectives, 1000 for three objectives, and 1500 for five objectives;
- $T = 0.1N$ and $n_r = 0.01N$;
- $\delta = 0.9$;
- In DE and mutation operators: CR=1.0 and F=0.5, $\eta = 20$ and $p_m = 1/n$.
- Stopping condition: the algorithm stops after 300,000 function evaluations for each test instance.

A set of N weight vectors W are generated using as follows:

1) Uniformly randomly generate 5,000 weight vectors for forming the set W_1. W is initialize as the set containing all the weight vectors $(1, 0, \ldots, 0, 0), (0, 1, \ldots, 0, 0), \ldots, (0, 0, \ldots, 0, 1)$.
2) Find the weight vector in W_1 with the largest distance to W, add it to W and delete it from W_1.
3) If the size of W is N, stop and return W. Otherwise, go to 2).

In calculating the IGD values, 100 nondominated solutions selected from each final population were used in the case of two objectives, 150 in the case of three objectives and 800 in the case of five objectives. The final solution set A is selected from the output $O = \{F(x^1), \ldots, F(x^N)\}$ of the algorithm is as follows:

- For the instances with two objectives, the set 100 final solutions are the solution set consisting of the best solutions in O for the subproblems with weights $(0, 1), (1/99, 98/99) \ldots, (98/99, 1/99), (1, 0)$.
- For the instances with more than two objectives:
  1) Randomly select an element e from O and set $O_1 = O \setminus \{e\}$ and $A = \{e\}$.
  2) Find the element in O_1 with the largest distance to A, and delete it from O_1 and add it to A.
  3) If the size of A is 150 for three objectives and 800 for five objectives, stop. Otherwise, go to 2).

The experiments were performed on a 1.86GHz Intel PC with 2GB RAM. The programming languages are MATLAB and C++.

TABLE I
The IGD statistics Based on 30 independent runs

Test Instances	Mean	Std	Best	Worst
UF01	0.00435	0.00029	0.00399	0.00519
UF02	0.00679	0.00182	0.00481	0.01087
UF03	0.00742	0.00589	0.00394	0.02433
UF04	0.06385	0.00534	0.05687	0.08135
UF05	0.18071	0.06811	0.08028	0.30621
UF06	0.00587	0.00171	0.00342	0.01005
UF07	0.00444	0.00117	0.00405	0.01058
UF08	0.05840	0.00321	0.05071	0.06556
UF09	0.07896	0.05316	0.03504	0.14985
UF10	0.47415	0.07360	0.36405	0.64948
R2DTLZ2M5	0.11032	0.00233	0.10692	0.11519
R2DTLZ3M5	146.7813	41.8281	66.1690	214.2261
WFG1M5	1.8489	0.0198	1.8346	1.8993

The IGD values are listed in table III. The distributions of the final populations with the lowest IGD values among the 30 runs for the first 10 test instances are plotted in figures 1-10.

For seven biobjective instances, MOEA/D found good approximations to UF1, UF2, UF3, UF6 and UF7 but performed poorly on UF4 and UF5. For three 3-objective instances, MOEA/D had better performance on UF8 than the other two. For three 5-objective instances, the IGD value found by MOEA/D on R2-DTLZ3-M5 was very large while those on R2-DTLZ2-M5 and WFG1-M5 were smaller.

IV. Conclusion

This paper described the basic idea and framework of MOEA/D. A dynamic computational resource allocation strategy was proposed. It was tested on the 13 unconstrained instances for CEC09 algorithm competition.

The source code of the algorithm can be obtained from its authors.

References

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

[2] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[3] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, Eds., Athens, Greece, 2001, pp. 95–100.

[4] J. D. Knowles and D. W. Corne, "The pareto archived evolution strategy: A new baseline algorithm for multiobjective optimisation," in *Proc. of Congress on Evolutionary Computation (CEC'99)*, Washington D.C., 1999, pp. 98–105.

[5] H. Ishibuchi and T. Murata, "Multiobjective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 28, no. 3, pp. 392–403, 1998.

[6] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - a comparative experiment," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 4, pp. 402–412, Aug. 2002.

[7] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evolutionary Computation*, vol. 7, no. 2, pp. 204–223, Apr. 2003.

[8] E. J. Hughes, "Multiple single objective pareto sampling," in *Proc. of Congress on Evolutionary Computation (CEC'03)*, Canberra, 2003, pp. 2678–2684.

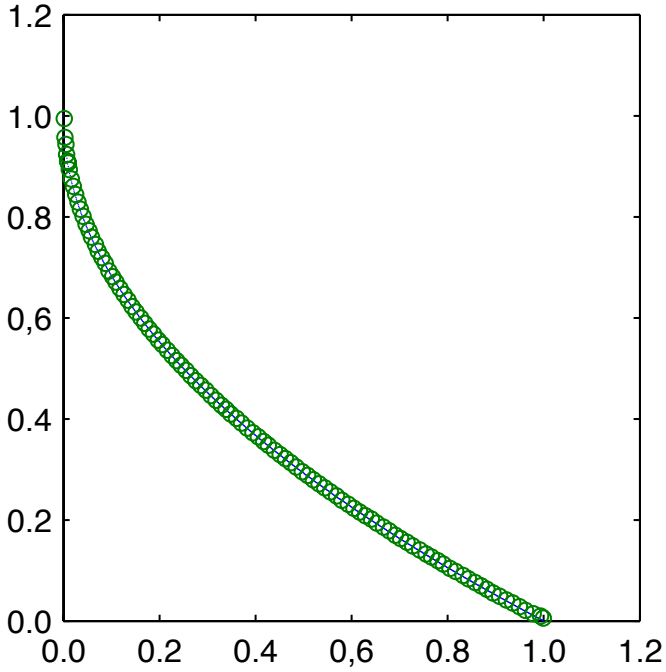

Fig. 1. The best approximation to UF1

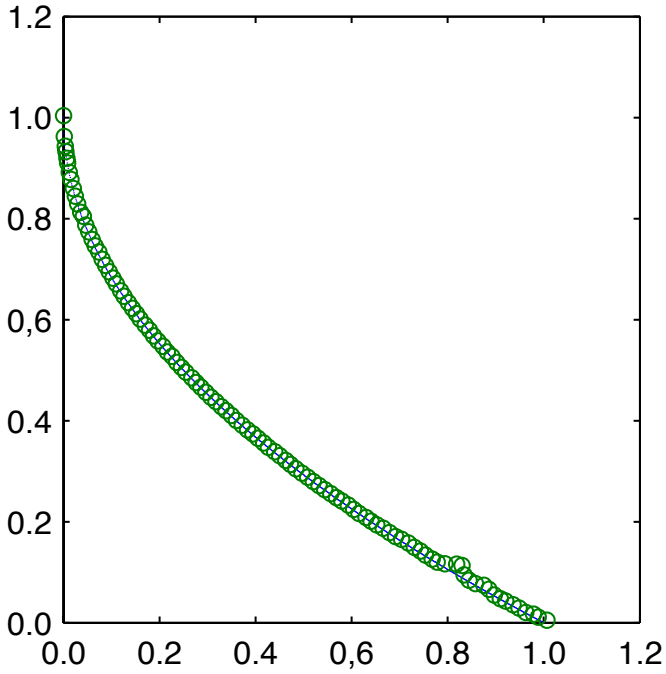

Fig. 2. The best approximation to UF2

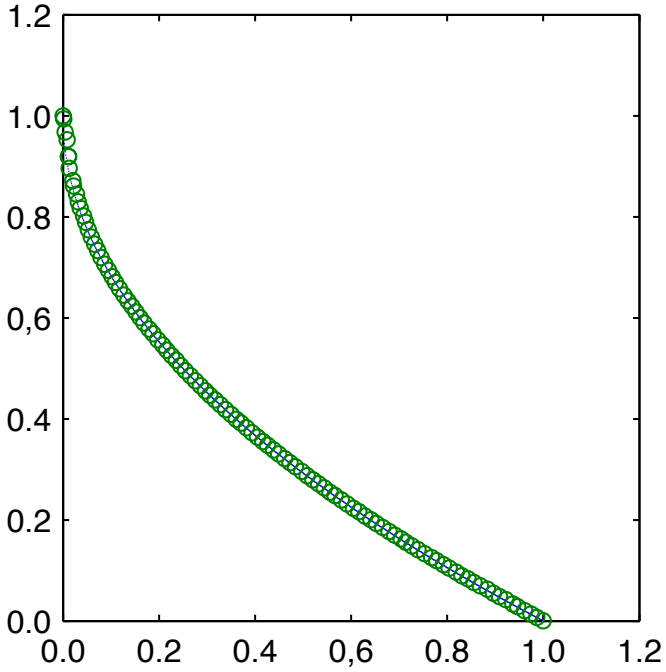

Fig. 3. The best approximation to UF3

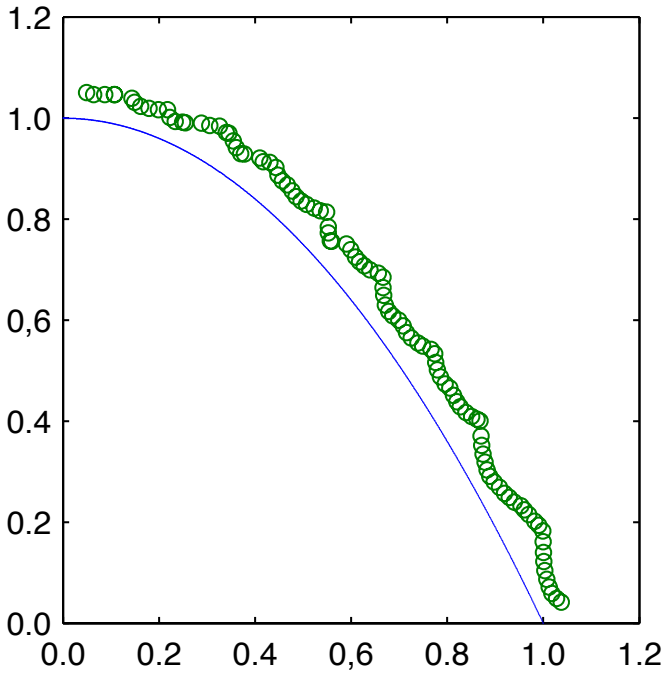

Fig. 4. The best approximation to UF4

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

[10] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto set, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, 2009, in press.

[11] P. C. Chang, S. H. Chen, Q. Zhang, and J. L. Lin, "MOEA/D for flowshop scheduling problems," in *Proc. of Congress on Evolutionary Computation (CEC '08)*, Hong Kong, 2008, pp. 1433–1438.

[12] W. Peng, Q. Zhang, and H. Li, "Comparision between MOEA/D and NSGA-II on the multi-objective travelling salesman problem," in *Multi-Objective Memetic Algorithms*, ser. Studies in Computational Intelligence, C.-K. Goh, Y.-S. Ong, and K. C. Tan, Eds. Heidelberg, Berlin: Springer, 2009, vol. 171.

[13] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with gaussian process model," Technical Report CES-489, the School of Computer Science and Electronic Engineering, University of Essex, Tech. Rep., 2009.

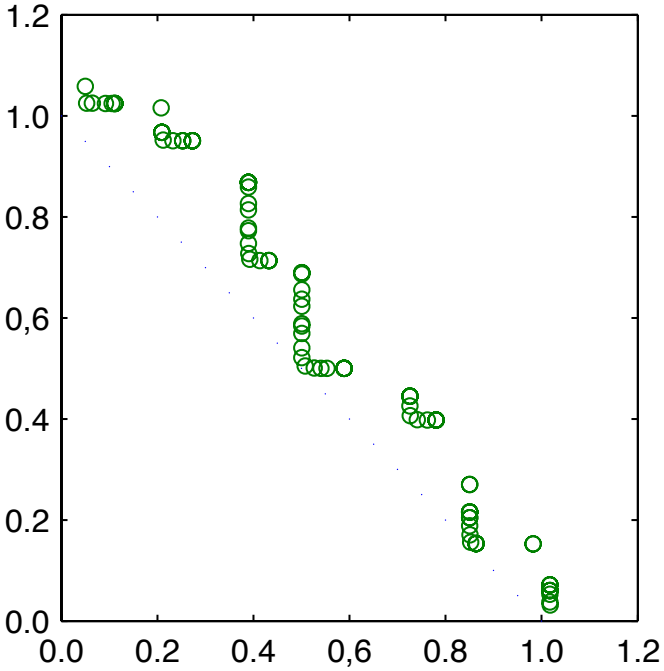

Fig. 5. The best approximation to UF5

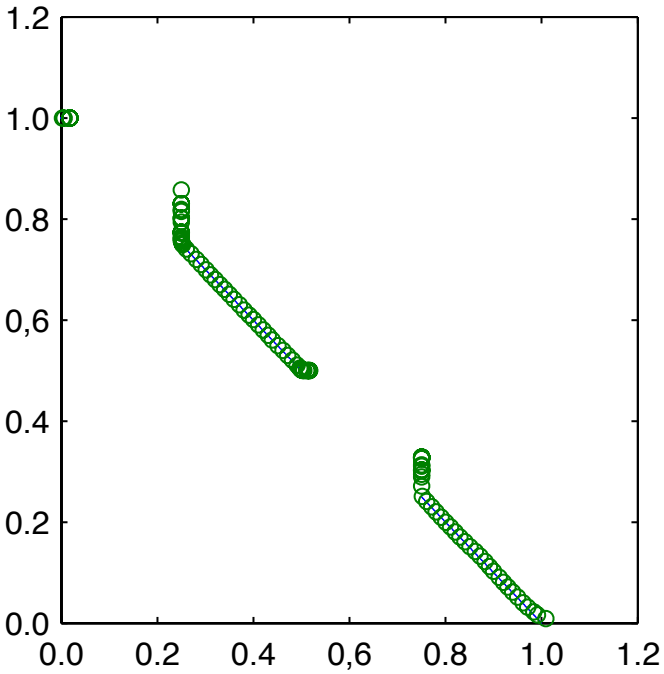

Fig. 6. The best approximation to UF6

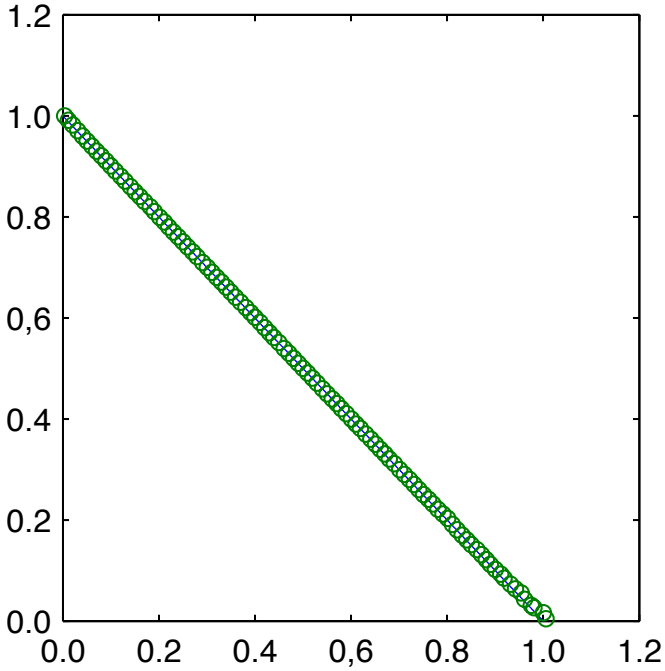

Fig. 7. The best approximation to UF7

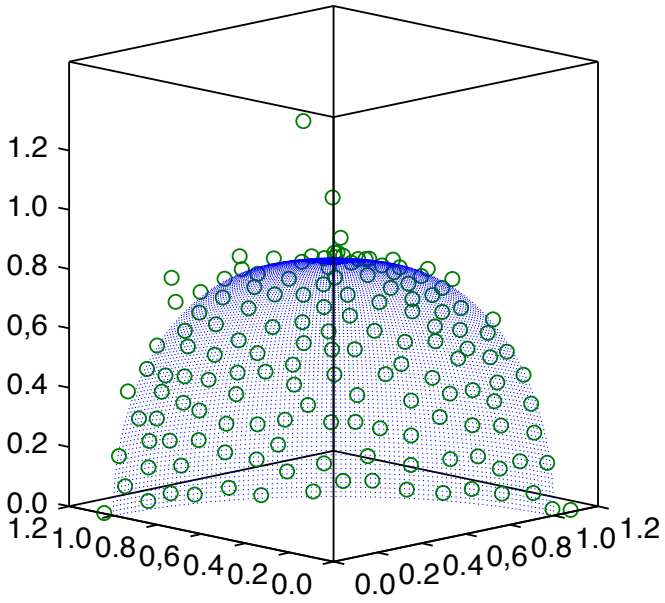

Fig. 8. The best approximation to UF8

[14] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm," in *Proc. of the 5th International Conference devoted to Evolutionary Multi-Criterion Optimization (EMO '09)*, Nantes, France, Apr. 2009.

[15] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Mmultiobjective optimization test instances for the CEC 2009 sepcial session and competition," Technical Report CES-487, The Shool of Computer Science and Electronic Engineering, University of Essex, Tech. Rep., 2008.

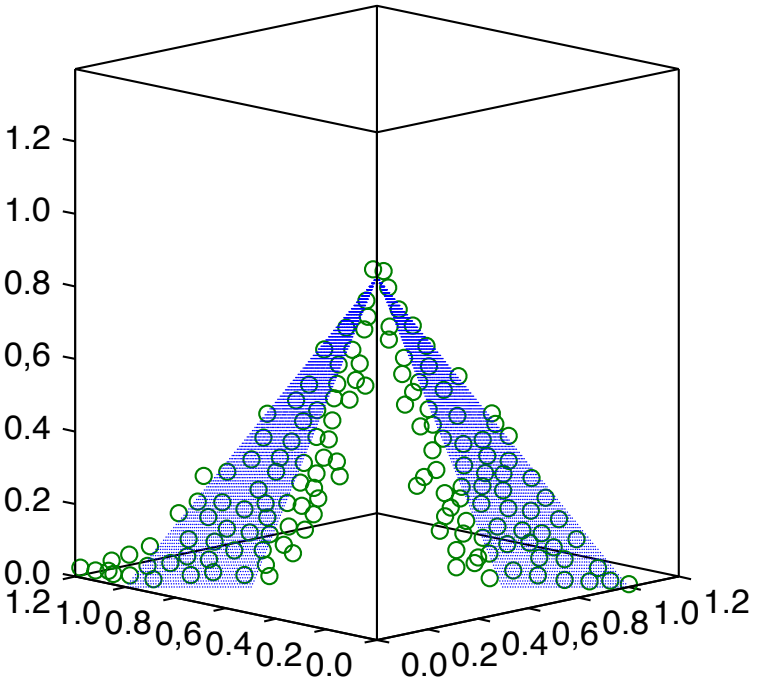

Fig. 9. The best approximation to UF9

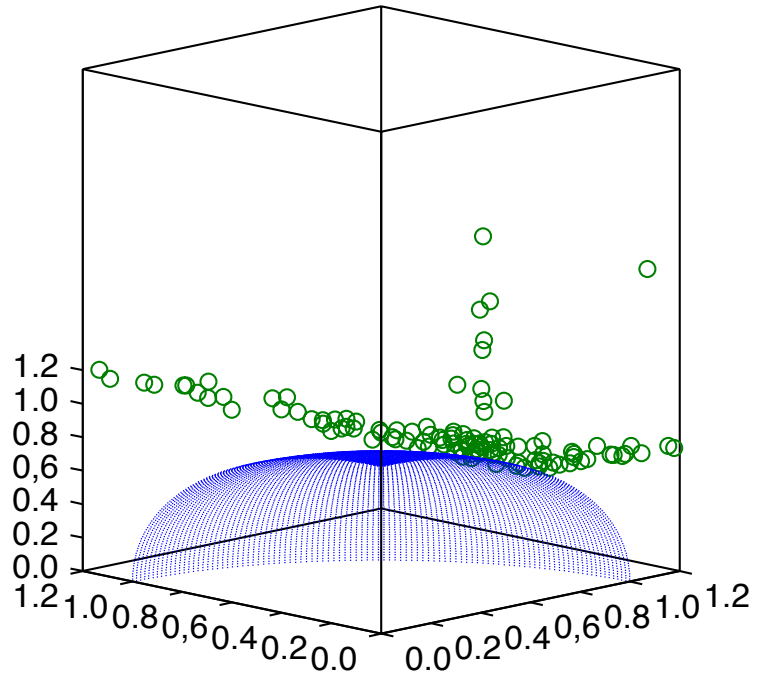

Fig. 10. The best approximation to UF10