MOEA/D-DRA with Two Crossover Operators

Wali Khan and Qingfu Zhang

***Abstract*—Different genetic operators suit different problems. Using several crossover operators should be an effective approach for improving the performance of an evolutionary algorithm. This paper studies the effect of the use of two crossover operators on MOEA/D-DRA for multi-objective optimization. It considers two crossover operators, namely, simplex crossover operator (SPX) and center of mass crossover operator (CMX). The probability of the use of each operator at each generation is updated in a adaptive way. The preliminary experimental results have indicated that this approach is promising.**

I. Introduction

In this paper, we consider the following multiobjective optimization problem (MOP) in its general form:

$$\begin{aligned} &\text{minimize } F(x) = (f_1(x), f_2(x) \ldots, f_m(x)) \\ &\text{such that } x \in X,\ X \subseteq R^n \end{aligned} \tag{1}$$

where $x = (x_1, \ldots, x_n)^T \in X$ is an n-dimensional vector of the decision variables, X is the *decision (variable) space*, and F is the objective vector function that contains m real valued functions.

Multiobjective optimization based on decomposition (MOEA/D) [1], [2], [3] is an evolutionary algorithm framework for multiobjective and MOEA/D-DRA [4], one of its recent versions, has exhibited some good performance on some hard multiobjective optimization problems [5].

Crossover operators are a major operator in most evolutionary algorithms for generating new solutions. It is well-known that different crossovers suit different problems. Recently, using multiple crossover operators in one algorithm has been received some attention and produced very good results [6], [7], [8].

In this paper, we employ two crossover operators, namely, simplex crossover (SPX) [9] and center of mass crossover (CMX) [10] in the framework of MOEA/D-DRA [4]. The probability of the use of each operator is adjusted in an adaptive way. We study the performance of the resultant implementation on UF1-UF10, the test instance in CEC 2009 algorithm competition [5].

The rest of this paper is organized as follows. Section II introduces the two crossover operators SPX, CMX and the Tchebycheff aggregation function. Section III describes the proposed probabilistic approach for the selection of SPX and CMX in the reproduction step of MOEA/D-DRA, Section IV presents the major steps of the algorithm. Section V gives the experimental settings and results. In Section VI, a brief conclusion and future work outline is given.

W. Khan is with the Department of Mathematical Sciences, University of Essex, Colchester CO4 3SQ, UK. phone: +44-1206-873032; fax: +44-1206-3040; email: wkhan@essex.ac.uk.

Q. Zhang is with the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: qzhang@essex.ac.uk).

II. Crossover Operators and Tchebycheff Aggregation Function

A. Simplex Crossover (SPX) [9], [11], [10]

Given three parents, x^1, x^2, x^3, the SPX generates three offsprings as follows :

$$y^{(k)} = (1 + \epsilon)(x^{(k)} - o), k = 1, 2, 3. \tag{2}$$

where $o = \frac{1}{3}\sum_{k=1}^{3} x^{(k)}$ is the center of the parents and $\epsilon \geq 0$ is a scaling parameter. In our implementation, we generate one offspring as in [9]:

$$z = \sum_{i=1}^{3} \alpha^i y^i + o,$$

where $\alpha^i > 0$ are randomly which satisfy $\sum_{i=1}^{3} \alpha^i = 1$.

B. Center of Mass Crossover (CMX) [10], [9], [12]:

Given three parents, x^1, x^2, x^3, the CMX operator in our experiments works as follows

- compute
$$o = \frac{1}{3}\sum_{i=1}^{3} x^i \tag{3}$$
and
$$v_i = 2o - x^i \tag{4}$$
- randomly select a v^j and a parent x^i and then apply the simulated binary crossover operator [13] on then to generate one offspring.

C. Tchebycheff Aggregation Function

MOEA/D decomposes a given MOP into N single objective subproblems by using an aggregation function. In this paper, we use the Tchebycheff aggregation function for this purpose, which is given as follows [14]:

$$\text{minimize } g(x|\lambda, z) = max_{1 \leq j \leq m} \lambda_j |f_j(x) - z_j^*|\} \tag{5}$$

$$\text{where} \begin{cases} x \in X \\ z^* = \{z_1, z_2, \ldots, z_m\} \\ z_j = min\{f_j(x)|x \in X\} \\ \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m) \\ \lambda_j \geq 0 \\ \sum_{j=1}^{m} \lambda_j = 1 \end{cases}$$

III. Proposed Probabilistic Approach for the Selection of SPX and CMX

We use two crossover operators, i.e., SPX and CMX, in MOEA/D to generate offspring. Let p_t^1 and p_t^2 be the probability of the use of SPX and CMX at generation t, respectively. We update p_t^1 and p_t^2 and select two crossovers as follows:

1) p_1^1 and p_1^2 is initialized to be 0.5 at generation 1.
2) In Step 3 in the algorithm (the details are given in the next section), in $p_t^1 \times 100\%$ of the subproblems, SPX will be used for generating new solutions. The rest of subproblems will use CMX.
3) Compute the success rate of each crossover operators at the current generation: r^1 and r^2 [1]. Then set:

$$p_{t+1}^i = 0.5 \times p_t^i + 0.5 \times \frac{r^i}{r^1 + r^2}$$

for $i = 1, 2$.

IV. Framework of MOEA/D-DRA

Let $\lambda^1, \ldots, \lambda^N$ be a set of even spread weight vectors and z^* be the reference point. The problem of approximation of the PF of (1) can be decomposed into N scalar optimization subproblems and the objective function of the j-th subproblem is:

$$g^{te}(x|\lambda^j, z^*) = \max_{1 \le i \le m} \{\lambda_i^j |f_i(x) - z_i^*|\} \tag{6}$$

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

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

- a population of N points $x^1, \ldots, x^N \in X$, 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 SPX or CMX. Then apply the polynomial mutation operator with probability p_m on it.

Step 3.3 Repair: If an element of y is out of the boundary of X, 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

[1]A crossover operator is successful if its offspring can replace at least one old solution

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 **Step 3.2**, Two crossover operators are employed as explained in Section III and polynomial mutation as explained below:

$$y_k = \begin{cases} y_k + \sigma_k(u_k - l_k) & \text{with probability } p_m \\ y_k & \text{with probability } 1 - p_m \end{cases} \quad (7)$$

Where l_k, u_k are the lower and upper bound of the the kth decision variable respectively.

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

Where $rand \in [0, 1]$ is a uniformly random number. The mutation rate p_m and distribution index η are the two control parameter.

V. Experimental Settings and Results

A. Parameters setting

In this paper, we use the same parameters settings as in [4] with the exception of DE parameters and are give as under:

- $N = 600$ for two objective test instances.
- $N = 1000$ for three objective test instances.
- $T = 0.1N$, it defines the number of neighbors of one single optimization problem (SOP);
- $n_r = 0.01N$, it restricts the maximum time of the successful updates;
- $\delta = 0.9$, it is the probability which decide that wether the candidate of parents are coming from the whole population or from the neighbors of SOP;
- $\eta = 20$ and $p_m = 1/n$ in the polynomial mutation operator, where n is parameter space dimension of MOP and η is the distribution index;
- The parameter for SPX is $\epsilon = \sqrt{n+1}$ for each problems except UF4 while for UF4, $\epsilon = \sqrt{\sqrt{n+1}}$ in both single and double crossover operators strategies.
- The algorithm stops after $300,000$ function evaluations;

B. Weight vectors selection

A set of N weight vectors, W, is generated by using the following criteria [4]:

1) Uniformly randomly generate $5,000$ weight vectors for forming the set $W1$. Set W is initialized 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 set W_1 with the largest distance to set W, add it to set W and delete it from set $W1$.
3) If the size of set W is N, stop and return set W. Otherwise, go to 2).

C. Performance metric

The performance metric IGD [15] used to measure the algorithm performance. Let P^* be a set of uniformly distributed points along the PF. Let A be an approximate set to the PF, the average distance from P^* to A is defined as [1], [5]:

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

where $d(v, A)$ is the minimum Euclidean distance between v and the points in A. If P^* is large enough to represent the PF very well, $D(A, P)$ could measure both the diversity and convergence of A in a sense.

D. Results

We tested three versions of MOEA/D-DRA: 1) using SPX only, 2) using CMX only, and 3) using both SPX and CMX as in Section III. The IGD statistics based on 30 independent runs of these three implementation are given in Table I. As can be seen from the table, the performance of using two crossover operators is the best on most of test instances.

Fig. 1 presents the distribution of the final solutions with the smallest IGD value found by each algorithm on each instances. Fig.2 presents the variation in IGD values by each algorithm on each instances.

Furthermore, in Fig.2 the upper curve (i.e, the red color), the middle curve (i.e., magenta color) and the lower curve (i.e., the green one) represents the variation in IGD value for MOEA/D-DRA-SPX, MOEA/D-DRA-CMX, MOEA/D-DRA-SPX+CMX, respectively, for each problem.

VI. Conclusion and Future Work

Different genetic operators suit different problems. Using several crossover operators should be a effective approach for improving the performance of an evolutionary algorithm. This paper have proposed a approach for using two crossover operators in MOEA/D for multiobjective optimization. The experimental results are very promising.

In the future, we will improve our algorithm by considering other crossovers and more intelligent approach for combining them.

References

[1] Q. Zhang and H. Li, "A Multiobjective Evolutionary Algorithm Based on Decomposition: MOEA/D," *IEEE Transaction on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, December 2007.

[2] H. Li and Q. Zhang, "Multiobjective Optimization Problems With Complicated Pareto Sets: MOEA/D and NSGA-II," *IEEE Transsation On Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, April 2009.

[3] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *GECCO*, M. Pelikan and J. Branke, Eds. Genetic and Evolutionary Computation Conference, GECCO 2010, Proceedings, Portland, Oregon, USA: ACM, July 7-11 2010, pp. 519–526.

[4] 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 (IEEE CEC 2009), Trondheim, Norway*, pp. 1–6, May, 18-21 2009.

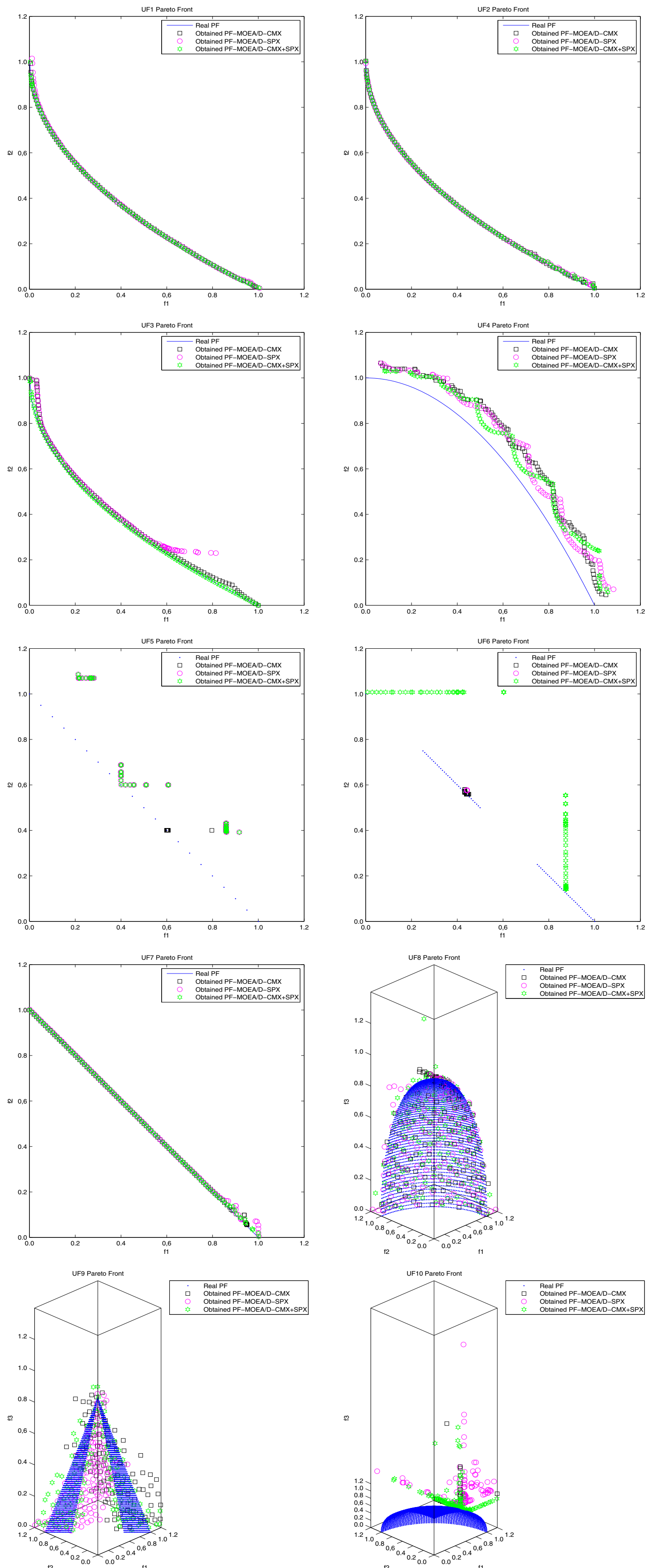

Fig. 1. Plots of the final approximated solutions with the lowest IGD value in the objective space found by MOEA/D-DRA with CMX, SPX, SPX+CMX in 30 independent runs.

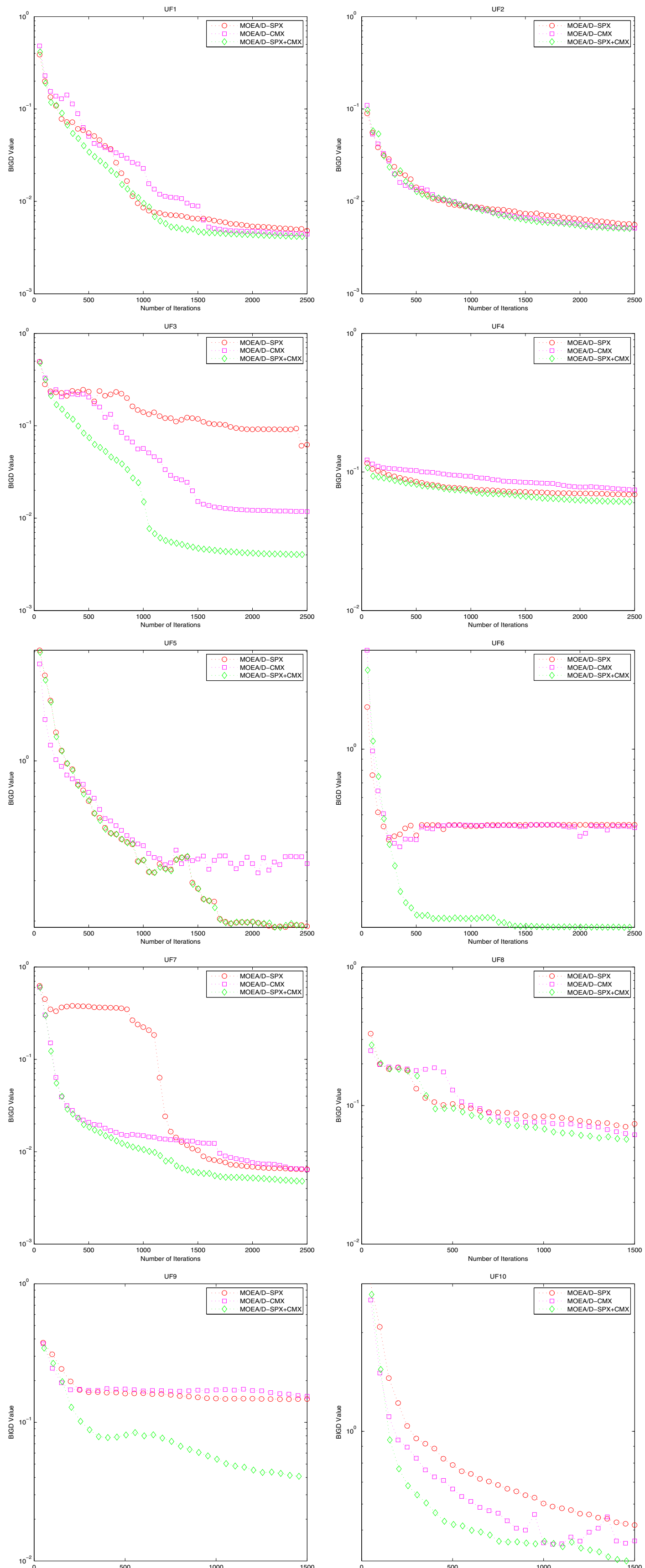

Fig. 2. Plots represents the variation in IGD values found by MOEA/D-DRA with CMX, SPX, SPX+CMX in best run out of 30 independent runs, respectively, for each problem .

TABLE I
THE IGD STATISTICS BASED ON 30 INDEPENDENT RUNS OF MOEA/D-DRA WITH CMX, SPX, SPX+CMX ON UNCONSTRAINED MULTIOBJECTIVE TEST INSTANCES, UF1-UF10.

MOEA/D-DRA-SPX, MOEA/D-DRA-CMX, MOEA/D-DRA-SPX+CMX						
Prob	min	median	mean	std	max	Crossover
UF1	0.004084	0.0046905	0.0051250	0.0034	0.0079980	CMX
	0.005077	0.008203	0.008652	0.004079	0.026533	SPX
	0.003985	0.004171	0.004292	0.000263	0.005129	SPX+CMX
UF2	0.0050340	0.0066460	0.0071868	0.00190157	0.012719	CMX
	0.004952	0.005596	0.005717	0.000487	0.007131	SPX
	0.005149	0.005472	0.005615	0.000412	0.006778	SPX+CMX
UF3	0.0049430	0.0432895	0.04153040	0.0239478	0.0855450	CMX
	0.005455	0.022374	0.030286	0.03823	0.091197	SPX
	0.004155	0.005313	0.011165	0.013093	0.068412	SPX+CMX
UF4	0.0508390	0.0632330	0.0628110	0.00476440	0.0734990	CMX
	0.0502	0.0573	0.0573	0.0038	0.0657	SPX
	0.055457	0.063524	0.064145	0.004241	0.075361	SPX+CMX
UF5	0.1676280	0.3791160	0.363792	0.0813385	0.5085780	CMX
	0.2186	0.3889	0.3963	0.0742	0.5704	SPX
	0.211058	0.379241	0.418508	0.135554	0.707093	SPX+CMX
UF6	0.0710280	0.44046750	0.3665337	0.12022382	0.60331490	CMX
	0.445548	0.507600	0.532296	0.070171	0.670946	SPX
	0.056972	0.248898	0.327356	0.185717	0.792910	SPX+CMX
UF7	0.0045910	0.006100	0.00762145	0.00526699	0.0317640	CMX
	0.005776	0.010898	0.062154	0.152973	0.533897	SPX
	0.003971	0.004745	0.006262	0.003307	0.014662	SPX+CMX
UF8	0.0559350	0.070072250	0.077114571	0.02153013	0.1310650	CMX
	0.078084	0.142313	0.136907	0.053748	0.272108	SPX
	0.051800	0.056872	0.057443	0.003366	0.065620	SPX+CMX
UF9	0.2075150	0.2875470	0.2807007	0.046701797	0.343310	CMX
	0.064034	0.181923	0.159370	0.053077	0.216422	SPX
	0.033314	0.144673	0.097693	0.054285	0.151719	SPX+CMX
UF10	0.4175320	0.4898737	0.4945660	0.0460605756	0.5991810	CMX
	0.445548	0.507600	0.532296	0.070171	0.670946	SPX
	0.391496	0.467715	0.462653	0.038698	0.533234	SPX+CMX

[5] Q. Zhang, A. Zhou, S. Zhaoy, P. N. Suganthany, W. Liu, and S. Tiwariz, "Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition," Technical Report CES-487, 2009.

[6] J. A. Vrugt, B. A. Robinson, and J. M. Hyman, "Self-Adapotive Multimethod Search for Global Optimization in Real-Parameter Spaces," *IEEE Transsation On Evolutionary Computation*, vol. 13, no. 2, pp. 243–259, April 2009.

[7] J. A. Vrugt and B. A. Robinson, "Improved Evolutionary Optimization from genetically adaptive mutimethod search," *Proceedings of the National Academy of Sciences of the United States of America: PNAS (USA)*, vol. 104, no. 3, pp. 708–701, Jaunuary 16, 2007.

[8] D. Thierens, "An Adaptive Pursuit Strategy for Allocation Operator Probabilities," *GECCO'05*, pp. 1539–1546, June 2005.

[9] S. Tsutsui, "Multi-parent recombination in genetic algorithms with search space boundary extension by mirroring," in *PPSN V: Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*. London, UK: Springer-Verlag, 1998, pp. 428–437.

[10] S. Tsutsui, M. Yamamura, and T. Higuchi, "Multi-parent Recombination with Simplex Crossover in Real Coded Genetic Algorithms," *Proc. of the GECCO-99*, pp. 657–374, 1999.

[11] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Transactions On Evolutionary Computation*, vol. 10, no. 6, pp. 658–675, December 2006.

[12] S. Tsutsui and A. Ghosh, "A Study on the Effect of Multi-parent Recombination in Realcoded Genetic Algorithms," *Proc. of the IEEE ICEC*, pp. 828–833, 1998.

[13] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms.*, First ed. John Wiley and Sons, Chichester, UK, 2001.

[14] K. M. Miettinien, *Nonlinear Multiobjective Optimization*, ser. Kluwer's International Series. Norwell, MA: Academic Publishers Kluwer, 1999.

[15] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," *IEEE Transactions on Evolutionary Computation*, vol. 7, pp. 117–132, 2003.