Applied Soft Computing 12 (2012) 2765–2780

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A decomposition-based hybrid multiobjective evolutionary algorithm with dynamic resource allocation

Wali Khan Mashwani*, Abdellah Salhi

Department of Mathematical Sciences, University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK

ARTICLE INFO

Article history:
Received 17 November 2011
Received in revised form 25 February 2012
Accepted 8 March 2012
Available online 24 May 2012

Keywords:
Multi-objective optimization
Pareto optimality
Decomposition
Crossover
Resource allocation

ABSTRACT

Different crossover operators suit different problems. It is, therefore, potentially problematic to chose the ideal crossover operator in an evolutionary optimization scheme. Using multiple crossover operators could be an effective way to address this issue. This paper reports on the implementation of this idea, i.e. the use of two crossover operators in a decomposition-based multi-objective evolutionary algorithm, but not simultaneously. After each cycle, the operator which has helped produce the better offspring is rewarded. This means that the overall algorithm uses a dynamic resource allocation to reward the better of the crossover operators in the optimization process. The operators used are the Simplex Crossover operator (SPX) and the Center of Mass Crossover operator (CMX). We report experimental results that show that this innovative use of two crossover operators improves the algorithm performance on standard test problems. Results on the sensitivity of the suggested algorithm to key parameters such as population size, neighborhood size and maximum number of solutions to be altered for a given subproblem in the the decomposition process are also included.

1. Introduction

1.1. Problem definition and basic concepts

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

$$\text{minimize } F(x) = (f_1(x), \ldots, f_m(x))^T \quad (1)$$

$$\text{subject to } x \in \Omega, \quad (1)$$

where Ω is the decision variable space, $x = (x_1, x_2, \ldots, x_n)^T$ is the the vector of decision variables, and $F(x) : \Omega \to R^m$ is a function that associates with every decision vector in Ω an m-vector in R^m with each entry corresponding to the value of some objective function of the MOP. R^m is referred to as the objective space. When Ω is a closed and connected region in R^n, and all the objectives are continuous in x, problem (1) is a continuous MOP.

Often, the objectives of problem (1) are in conflict with one another or are incommensurable. This means, a single solution in the search space Ω that minimizes all the objectives functions simultaneously does not exit. Instead, one tries to find the best trade-offs among the different objectives. These tradeoffs are defined in terms of Pareto optimality [10], a formal definition of which can be as follows [5–7,18].

Definition 1. Let $u = (u_1, u_2, \ldots, u_m)^T$ and $v = (v_1, v_2, \ldots, v_m)^T$ be two vectors in R^m; u is said to dominate v, denoted as $u \prec v$, if and only if

1. $u_i \leq v_i$ for every $i \in \{1, 2, \ldots, m\}$
2. $u_j < v_j$ for at least one index $j \in \{1, 2, \ldots, m\}$.

Remark 1. For any two given vectors, u and v, there are two possibilities:

1. Either u dominates v or v dominates u.
2. Neither u dominates v nor v dominates u.

Definition 2. A solution $x^* \in \Omega$ is said to be Pareto optimal for problem (1), if there is no other solution $x \in \Omega$ such that $F(x)$ dominates $F(x^*)$. $F(x^*)$ is called the Pareto optimal (objective) vector.

Remark 2. Any Pareto optimal point improvement in one objective must lead to deterioration in at least one other objective.

Definition 3. The set of all Pareto optimal solutions is called the Pareto set (PS), written as $PS = \{x \in \Omega | \nexists y \in \Omega, F(y) \prec F(x)\}$.

Definition 4. The image of the Pareto optimal set (PS) in the objective space is called the Pareto front (PF), $PF = \{F(x) | x \in PS\}$. In other words, the set of all Pareto optimal solutions forms a tradeoff surface in the objective space.

* Corresponding author.
E-mail address: mashwanigr8@gmail.com (W. Khan Mashwani).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.03.067

Remark 3. In MOP, Pareto domination does not define a complete ordering among the solutions in the objective space. It, also, does not measure how much one solution is better than another.

Over the last two decades, MOP has received growing attention, possibly because it has a wide application. Evolutionary Computation (EC) has also seen significant developments in recent years. Multi-objective evolutionary algorithms (MOEAs) have received a lot of attention due to their ability to find multiple tradeoff solutions in a single run. They are population-based and, therefore, work with a population of candidate solutions.

Most existing MOEAs dealing with MOP are Pareto dominance based, i.e. they determine the fitness of candidate solutions in the population by Pareto dominance relations with other solutions found in previous searches [4,9,13,20,32,34]. To promote diversity, they use various techniques such as fitness sharing, niching, the Kernel approach, the nearest neighbor approach, the histogram technique, crowding/clustering, and a relaxed form of dominance and restricted mating [5]. Non-dominated sorting algorithms (NSGA), such as NSGA-II [9], and SPEA2 [32], are examples of Pareto dominance based MOEAs.

Decomposition is a basic strategy in traditional mathematical programming [11,18]. It has not been widely used in Evolutionary Multi-objective Optimization (EMO). Fairly recently, a multi-objective evolutionary algorithm based on decomposition (MOEA/D) has been developed [29]. It combines decomposition techniques and evolutionary algorithms for dealing with MOP. It converts a MOP into a number of scalar optimization subproblems which it then optimizes simultaneously [16,29]. The key features of MOEA/D are that it solves each subproblem using information from its neighborhood which consists of other subproblems. The neighborhood relationships among the subproblems are based on the distances between their corresponding weight vectors. The population kept by MOEA/D includes the best solution found so far for each subproblem.

MOEA-type algorithms use mainly two genetic operators, crossover and mutation, for creating offspring. Recall that mutation is applied to a single solution, unlike crossover which is generally applied to at least two [1].

A number of crossover operators can be found in the literature such as the blend crossover operator (BLX-α) [12], the simulated binary crossover (SBX) [15,19], the Simplex Crossover (SPX) [2,27], the Center of Mass Crossover (CMX) [25,26], the unimodal normally distributed crossover (UNDX) [21,22], the parent-centric crossover (PCX) [8], and many other real coded crossover operators for dealing with continuous problems [7,23]. Here, we consider the use of multiple crossover operators within the MOEA/D search algorithm [30]. These operators are SPX [2,27], and CMX [25,26]. The result is a hybrid algorithm denoted MOEA/D-DRA-CMX+SPX in which the probability of the use of each crossover operator is automatically adjusted based on its performance. A reward or credit of 1 is assigned whenever the generated offspring solution of the current subproblem successfully replaces at least one solution of a neighboring subproblem. Otherwise, the reward is 0. Hence the concept of dynamic resource allocation (DRA). Note that the resource we are concerned with is really "computing". Computing facilities can be explicitly or implicitly allocated. Here, the facility is implicitly allocated by allowing an operator to be more likely used rather than another, thus making it absorb computing power every time it is used.

The key features of the proposed hybrid algorithm, i.e. MOEA/D-DRA-CMX+SPX, are as follows.

1. It is adaptive.
2. It maintains two different sets of subproblems in its evolutionary process.
3. It uses two crossover operators each of which deals with its own set of subproblems.
4. The number of subproblems in the two different sets is adjusted dynamically according to a parameter (probability) of the used crossover operator.

The performance of MOEA/D-DRA-CMX+SPX is compared with that of MOEA/D-DRA-CMX and that of MOEA/D-DRA-SPX on CEC'09 test instances [31]. Its sensitivity to key parameters is also empirically investigated. To summarize, the paper addresses the following issues.

1. Would the use of multiple crossovers enhance MOEA/D-type algorithms?
2. How is resource allocation used to reward different crossover operators?

The rest of the paper is organized as follows. Section 2 presents the framework of MOEA/D-DRA, and explains how the use of two genetic operators can be implemented by adjusting a probability parameter. Section 3 presents MOEA/D-DRA-CMX+SPX. Section 4 describes the benchmark problems, the parameter settings for the empirical investigation, and the metric used to measure the performance of the suggested hybrid algorithm. Section 5 discusses experimental results. Section 6 discusses the sensitivity of the algorithm performance to key parameters, and finally, Section 7 concludes the paper and points out some worthwhile issues to follow up.

2. The MOEA/D-DRA framework

MOEA/D-DRA builds up on MOEA/D [30], which, itself, builds up on the MOEA framework which is perhaps epitomized by NSGA-II [9]. In the following we provide the different components required by the suggested hybrid algorithm for a successful implementation.

2.1. Decomposition: implementation

MOEA/D implements decomposition to transform the approximation of the PF of the overall problem by that of N single objective subproblems. Decomposition can be implemented in many ways. The approach adopted in this paper is Tchebycheff aggregation [18,29,30].

2.1.1. The Tchebycheff aggregation function

In this approach, the scalar optimization problem is of the form

$$\text{minimize } g^{te}(x|\lambda, z) = \max_{1\le j\le m}\{\lambda_j|f_j(x) - z_j^*|\}, \tag{2}$$

where $x \in \Omega$, $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T$, $\lambda_j \ge 0, j = 1, \ldots, m$, and $\sum_{j=1}^{m}\lambda_j = 1$. $z^* = (z_1^*, \ldots, z_m^*)^T$ is the reference point, i.e., $z_j^* = min\{f_j(x)|x \in \Omega\}$ for each $j = 1, \ldots, m$. It is well known that, under mild conditions, for each Pareto optimal solution there exists a weight vector λ such that it is the optimal solution to (2) and each optimal solution of (2) is a Pareto optimal solution to problem (1) [29]. Therefore, a set of Pareto optimal solutions can be obtained to problem (1) by optimizing a set of single optimization subproblems defined by the Tchebycheff approach with different weight vectors.

2.1.2. Weight vector selection in the decomposition process

A set of N weight vectors, W, is generated according to the following criteria as were adopted in [30].

1. Uniformly randomly generate 5000 weight vectors for forming the set $W1$. Set W is initialized as the set

containing all the weight vectors (1, 0, . . ., 0, 0), (0, 1, . . ., 0, 0), . . ., (0, 0, . . ., 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 W_1.
3. If the size of set W is N, stop and return set W. Otherwise, go to 2.

2.2. Genetic operators

Two main operators are implemented in evolutionary algorithms. They are *mutation* and *crossover*. The literature of EA also refers to a third one, *reproduction* which is the process of copying individuals from the current population into the next one without alteration. Here, however, *reproduction* refers to the concept of generating new offspring.

2.2.1. The polynomial mutation operator

The polynomial mutation operator of [7] is used in the implementation of MOEA/D-DRA. Let y_k be the decision variable to be mutated. Then y_k is computed as follows.

$$y_k = \begin{cases} y_k + \sigma_k(u_k - l_k) & \text{with probability } p_m, \\ y_k & \text{with probability } 1 - p_m, \end{cases} \tag{3}$$

where l_k and u_k are the lower and upper bounds of the kth decision variable, respectively. And

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

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

2.2.2. The crossover operators

As mentioned earlier, there are many crossover operators. Here we are concerned with two of them, namely the Center of Mass Crossover (CMX) [25,26], and Simplex Crossover (SPX) [2,27].

2.2.2.1. The Center of Mass Crossover. Given three parents, $x^{(1)}$, $x^{(2)}$, $x^{(3)}$, the CMX operator works as follows:

- Compute the center of mass as

$$o = \frac{1}{3}\sum_{i=1}^{3} x^i \tag{4}$$

and then create a set of virtual mates by mirroring each parent across the center of mass

$$v_i = 2o - x^i. \tag{5}$$

In our implementation, we randomly select a virtual mate v^j and a parent x^i and then generate one offspring as:

$$y = (1 - \alpha)x^i + \alpha v^j, \tag{6}$$

where $\alpha = 2r - 0.5$, r is a random number.

CMX is expected to have the following characteristics:

1. It is the natural generalization of the 2-parent crossover operator.
2. It is simple and easy to implement.
3. It has one control parameter which changes dynamically.
4. It tends to generate offspring uniformly around the parents.

2.2.2.2. The Simplex Crossover. Given three parents, $x^{(1)}$, $x^{(2)}$, $x^{(3)}$, SPX generates three offspring as follows:

$$y^{(k)} = (1 + \epsilon)(x^{(k)} - o), \quad k = 1, 2, 3. \tag{7}$$

where $o = (1/3)\sum_{k=1}^{3} x^{(k)}$ is the centroid of the parents and $\epsilon \geq 0$ is the scaling parameter which controls the expansion of the simplex. In our implementation, we generate one offspring as in [25]:

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

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

Following are some key features of SPX [27]:

1. SPX is simple and easy to implement.
2. It has only one control parameter, ϵ.
3. SPX is based on simplex. It can generate offsprings inside the expanded region of the simplex.
4. SPX is independent of the coordinate system.

2.3. Resource allocation: a probabilistic approach

Let p_t^1 and p_t^2 be the probabilities of the use of crossovers C_1 and C_2 at generation t. These probabilities are updated as follows.

1. p_1^1 and p_1^2 are initialized to 0.5 in generation 1.
2. In Step 3 of the algorithm (the details are given in the next section), in $\lfloor p_t^1 \times 100\% \rfloor$ of the subproblems, C_1 will be used for generating new solutions. The rest of the subproblems will use C_2.
3. Compute the total reward, r^1 and r^2 of each crossover operator at the current generation. Then set:

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

for $i = 1, 2$.

A crossover is said to be successful if its new generated solution can replace at least one solution. Whenever a crossover operator is successful it will get a reward of 1. Otherwise, it will get a reward of 0. This means, the more successful crossover operator will be applied to more subproblems than the less successful one.

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

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

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

MOEA/D-DRA minimizes all of these N objective functions simultaneously, in a single run. In [17,29], all the subproblems are treated equally; each one of them receives the same amount of computational time. However, it is reasonable to assign different computational efforts to different problems. Indeed, MOEA/D-DRA, as in [30], does just that for different subproblems. It defines and computes 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 (2) maintains:

- a population of N points $x^1, \ldots, x^N \in \Omega$, where x^i is the current solution to the ith 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 each objective f_i;
- utilities $\pi^1, \ldots, \pi^N$ for subproblem i, $i = 1, \ldots, N$.
- *gen*, the current generation number.

3. A new hybrid algorithm: MOEA/D-DRA-CMX+SPX

Here we describe the hybrid algorithm MOEA/D-DRA-CMX+SPX and how it is implemented. Note that the general framework, as already mentioned earlier, is that of MOEA/D which can be found in many sources such as [17,29]. The core of the contribution of this paper is in Step 3.2 where new offspring are generated according to either CMX or SPX. Initially, they have the same probability of being used. Thereafter, their use is dependent on how successful they are at generating good offspring that are used to replace others previously generated. The MOEA/D framework is repeated here for completeness and the benefit of the novice in MOEA matters.

Algorithm
Input:
1: MOP (1);
2: a stopping criterion;
3: N, the number of subproblems to be considered in MOEA/D-DRA;
4: a uniform spread of N weight vectors $\lambda^1, \ldots, \lambda^N$;
5: T, the number of 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 $i = 1, \ldots, N$, 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 in the search space.
Step 1.3 Initialize $z = (z_1, \ldots, z_m)^T$ by setting $z_i = \min\{f_i(x^{(1)}), f_i(x^{(2)}), \ldots, f_i(x^{(N)})\}$, for $i = 1, \ldots, m$.
Step 1.4 Set $gen = 0$ and $\pi^i = 1$ for all $i = 1, \ldots, N$.
Step 2 Selection of Subproblems for Search: the indices of the subproblems whose objectives are MOP individual objectives f_i are selected to form initial set I. By using 10-tournament selection based on π^i, select other $[N/5] - m$ indices 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 indices r_2 and r_3 from the current population P. Then apply SPX or CMX to generate a new solution $\bar{y}$ using the selected parent solution, x^{r_1}, x^{r_2} and x^{r_3}. Then apply the polynomial mutation operator [7], to $\bar{y}$ with probability of mutation p_m to get a new solution y.
Step 3.3 Repair: If the elements of y are not within the specified domain in the parameter space Ω, reset them inside this domain.
Step 3.4 Update z**:** For each $j = 1, \ldots, m$, if $z_j > f_j(y)$, then set $z_j = f_j(y)$.
Step 3.5 Update 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 $g(y|\lambda^j, z) \leq g(x^j|\lambda^j, z)$, then set $x^j = y$, $FV^j = F(y)$ and $c = c + 1$.
(3) Delete j from P and go to (1).

Table 1
Details of the used benchmark functions.

Function	Search space range	Characteristics of PF
UF1	$[0, 1] \times [-1, 1]^{n-1}$	Concave
UF2	$[0, 1] \times [-1, 1]^{n-1}$	Concave
UF3	$[0, 1]^n$	Concave
UF4	$[0, 1] \times [-2, 2]^{n-1}$	Convex
UF5	$[0, 1] \times [-1, 1]^{n-1}$	21 point front
UF6	$[0, 1] \times [-1, 1]^{n-1}$	One isolated point and two disconnected parts
UF7	$[0, 1] \times [-1, 1]^{n-1}$	Continuous straight line
UF8	$[0, 1]^2 \times [-2, 2]^{n-2}$	Parabolic
UF9	$[0, 1]^2 \times [-2, 2]^{n-2}$	Planar
UF10	$[0, 1]^2 \times [-2, 2]^{n-2}$	Parabolic

Step 4 Stopping Criteria: If the stopping criteria are 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 multiple of 50, then
Compute Δ^i, the relative decrease in the objective value of each subproblem i during the last 50 generations, as

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

Update utility π^i using

$$\pi^i = \begin{cases} 1 & \text{if } \Delta^i > 0.001 \\ (0.95 + 0.05\dfrac{\Delta^i}{0.001})\pi^i & \text{otherwise.} \end{cases} \quad (8)$$

Endif
Go to **Step 2**.

In the 10-tournament selection of **Step 2**, the index with highest π^i value from 10 uniformly randomly selected indices is chosen to enter I. We should do this selection $[N/5] - m$ times.

In **Step 3.2**, we used two crossover operators for generating new solutions to the subproblems. The probability of the use of each crossover is updated according to their individual successes. Each new individual solution undergoes the polynomial mutation [7] operation which is formulated in Eq. (3). The probabilities of the crossover operators in MOEA/D-DRA-CMX+SPX are updated as explained in Section 2.3.

4. Empirical investigation: benchmark, settings and performance measure

4.1. The benchmark problems: the CEC'09 test set

Three versions of the new hybrid algorithm based on MOEA/D enhanced with resource allocation and one of two crossovers, i.e. MOEA/D-DRA-CMX and MOEA/D-DRA-SPX, and MOAE/D enhanced with resource allocation and both crossovers, i.e. MOEA/D-DRA-CMX+SPX, are tested on CEC'09 benchmark problems [31]. These problems, their search space ranges and the characteristics of their Pareto fronts are given in Table 1.

4.2. Parameter settings

The following parameters settings were used in our experiments.

- $N = 600$, the population size for 2-objective test instances (UF1–UF7),
- $N = 1000$, the population size for 3-objective test instances (UF8–UF10),

Table 2
The IGD-metric values of the three algorithms over 30 independent runs on UF1, . . ., UF5.

CEC'09	Min	Median	Mean	St.dev.	max	Algorithm	Ranks
MOEA/D-DRA-CMX (a), MOEA/D-DRA-SPX (b), MOEA/D-DRA-CMX+SPX (c)							
UF1	0.004084	0.0046905	0.0051250	0.0034	0.0079980	a	2
	0.005077	0.008203	0.008652	0.004079	0.026533	*b*	3
	0.003985	**0.004171**	**0.004292**	**0.000263**	**0.005129**	**c**	1
UF2	*0.0050340*	*0.0066460*	*0.0071868*	*0.00190157*	*0.012719*	*a*	3
	0.004952	0.005596	0.005717	0.000487	0.007131	b	2
	0.005149	**0.005472**	**0.005615**	**0.000412**	**0.006778**	**c**	1
UF3	*0.0049430*	*0.0432895*	*0.04153040*	*0.0239478*	*0.0855450*	*a*	3
	0.005455	0.022374	0.030286	0.03823	0.091197	b	2
	0.004155	**0.005313**	**0.011165**	**0.013093**	**0.068412**	**c**	1
UF4	0.0508390	0.0632330	0.0628110	0.00476440	0.0734990	a	2
	0.0502	**0.0573**	**0.0573**	**0.0038**	**0.0657**	**b**	1
	0.055457	0.063524	0.064145	0.004241	0.075361	*c*	3
UF5	**0.1676280**	**0.3791160**	**0.363792**	**0.0813385**	**0.5085780**	**a**	1
	0.2186	0.3889	0.3963	0.0742	0.5704	b	2
	0.211058	0.379241	0.418508	0.135554	0.707093	*c*	3

- $T = 0.1N$, the number of neighbors for each subproblem,
- $n_r = 0.01N$, the maximal number of solutions to update successfully by a new solution of subproblem,
- $\delta = 0.9$, the probability for the mating pool to select solutions from the neighborhood,
- $\eta = 20$, the distribution index used in real mutation,
- $p_m = 1/n$, the probability of mutation, where $n = 30$ denoted the number of parametric variables,
- $\epsilon = \sqrt{n+1}$, the control parameter of the simplex, used in SPX,
- The algorithm stops after 300,000 function evaluations; this is the stopping criterion for all tested versions of MOEA/D-DRA in our experimental work.

4.3. Performance metric

A performance metric or indicator is used to measure the quality of the approximated set of Pareto optimal solutions. In multi-objective optimization, the quality of the obtained set of solutions can be measured in several ways, for example, the closeness of the solutions to true Pareto optimal front (i.e., convergence property), the uniformity of the solutions (i.e., diversity preservation) and the dominance relationship between two sets of solutions. Limitations and suitability of various performance indicators/metrics have been discussed in the specialized literature such as [24,33,35].

The inverted generational distance (IGD)-metric [35] is used in assessing the performance of the proposed versions of the MOEA/D-DRA [30], in this paper. It measures both the convergence and the spread of the obtained solutions; 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 PF. Let P be an approximate set to the PF, the average distance from P^* to P is defined [29,31], as

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

where $d(v, P)$ is the minimum Euclidean distance between v and the points in P. If P^* is large enough to represent the PF very well, $D(P^*, P)$ could measure both the diversity and convergence of P in a sense. To have a lower value of $D(P^*, P)$, P must be very close to the PF and cannot miss any part of it.

In our experiments, P^* is 1000 evenly distributed points in the PF of 2-objective problems and it is 10,000 for problems with 3 objective functions.

The IGD-metric is the inverted version of the widely used generational distance (GD) performance metric [28]. GD represents the average distance of the points in an approximation to the true PF, which cannot effectively measure the diversity of the approximation. There are several different ways of computing and averaging the distances in the GD performance metrics [9,28]. The version of the IGD-metric used in this thesis inverts the Υ version of GD as described in [9].

5. Experimental results

The three versions of MOEA/D-DRA algorithm, namely, MOEA/D-DRA-CMX, MOEA/D-DRA-SPX, and MOEA/D-DRA-CMX+SPX, are tested on UF1 through to UF10 [31]. All were run 30 times independently on each test instance. The same parameter settings are used by all algorithms for a fair comparison.

5.1. Discussion of IGD-metric statistical results for MOEA/D-DRA-CMX+SPX

Tables 2 and 3 present the best (i.e., minimum), median, mean, standard deviation (std), and worst (i.e., maximum) of the IGD-metric values of the final approximation over 30 independent executions of MOEA/D-DRA-CMX, MOEA/D-DRA-SPX, and MOEA/D-DRA-CMX+SPX on each test instance.

The last columns of Tables 2 and 3 show the performance ranks of the individual algorithms; bold data highlight rank 1 (best), teletype data, rank 2 and Italic data, rank 3 (i.e., worst). Based on these rankings, MOEA/D-DRA-CMX+SPX comes first on nine out of ten test problems. This clearly shows that the incorporation of two crossover operators is a good idea. It, indeed, improves the performance of MOEA/D-DRA, on nearly all problems of the CEC'09 collection [31].

5.2. Discussion of Pareto front graphical results

Figs. 1–4 plot the final approximate solutions in the objective space with the best (i.e., smallest) IGD-metric value over 30 independent runs of MOEA/D-DRA-SPX (1st panel), MOEA/D-DRA-CMX (2nd panel) and MOEA/D-DRA-CMX+SPX (3rd panel), respectively, on all test problems. These figures clearly show that MOEA/D-DRA-CMX+SPX is more effective based on the IGD-metric values found.

It is well known, [14], that no single metric can always provide the necessary information to judge the performance of MOEAs. For this reason, the Pareto fronts generated by the algorithms are plotted in the objective space to show their distribution range for a given problem. Figs. 5 and 6 show such plots. These figures clearly indicate that MOEA/D-DRA-CMX+SPX has produced better PFs in each of 30 runs, in the objective space for all CEC'09 test instances considered except for UF5 and UF6. These two problems are much

Table 3
The IGD-metric values of the 3 algorithms over 30 independent runs on UF6–UF10.

CEC'09	Min	Median	Mean	St.dev.	Max	Algorithm	Ranks
MOEA/D-DRA-CMX (a), MOEA/D-DRA-SPX (b), MOEA/D-DRA-CMX+SPX (c)							
UF6	0.0710280	0.44046750	0.3665337	0.12022382	0.60331490	a	2
	0.445548	0.507600	0.532296	0.070171	0.670946	*b*	3
	0.056972	**0.248898**	**0.327356**	**0.185717**	**0.792910**	**c**	1
UF7	0.0045910	0.006100	0.00762145	0.00526699	0.0317640	a	2
	0.005776	0.010898	0.062154	0.152973	0.533897	*b*	3
	0.003971	**0.004745**	**0.006262**	**0.003307**	**0.014662**	**c**	1
UF8	0.0559350	0.070072250	0.077114571	0.02153013	0.1310650	a	2
	0.078084	*0.142313*	*0.136907*	*0.053748*	*0.272108*	*b*	3
	0.051800	**0.056872**	**0.057443**	**0.003366**	**0.065620**	**c**	1
UF9	*0.2075150*	*0.2875470*	*0.2807007*	*0.046701797*	*0.343310*	*a*	3
	0.064034	0.181923	0.159370	0.053077	0.216422	b	2
	0.033314	**0.144673**	**0.097693**	**0.054285**	**0.151719**	**c**	1
UF10	0.4175320	0.4898737	0.4945660	0.0460605756	0.5991810	a	2
	0.445548	0.507600	*0.532296*	*0.070171*	*0.670946*	*b*	3
	0.391496	**0.467715**	**0.462653**	**0.038698**	**0.533234**	**c**	1

harder compared to the others; indeed, each algorithm has found a weak set of final solutions for both of them.

5.3. Discussion of the evolution of the IGD-metric values

Fig. 7 plots the evolution of the average IGD-metric values found over 30 independent runs against the number of generations spent by MOEA/D-DRA-SPX+CMX on each of the CEC'09 test instances [31]. These indicates that our proposed hybrid algorithm effectively handle all tested problems in terms of reducing the IGD-metric value compared to one crossover based algorithm.

Fig. 8 plots the evolution of the IGD-metric values of the best run among 30 independent runs against the number of generations for all tested algorithms, MOEA/D-DRA-CMX, MOEA/D-DRA-SPX and MOEA/D-DRA-CMX+SPX on each of the CEC'09 test instance. These figures visually show that MOEA/D-DRA-CMX+SPX performs significantly better in terms of convergence towards the true the PF of all CEC'09 test instances except on UF5 and UF6. These two problems are quite hard for all versions of MOEA/D-DRA.

5.4. Discussion of the dynamic allocation of resources to CMX and SPX

The resource, here *computing*, is in terms of the probability with which a given crossover is called during the search process. It is easy to see that when the probability of using a crossover is high, that of the other crossover must be low. This follows from their respective performances. In other words, the crossover that performed well will get the higher probability of use. Plots of the allocated probability values of CMX and SPX versus the number of iterations of MOEA/D-DRA-CMX+SPX on each of the CEC'09 test instances show just that. The important conclusion to draw is that where a crossover is not so good, the other one will get used instead. This means that overall, the search process uses the most suitable crossover, therefore, increasing the search effectiveness of the process. It can also be seen as if the two crossovers work synergistically.

Note that the combination of well performing crossover operators will not necessarily lead to a better performance on all problems. Indeed, MOEA/D-DRA-CMX+SPX is more effective than the versions of MOEA/D-DRA with individual crossovers overall, but not on UF5 and UF6. This may be explained by the fact that for some problems, a given crossover operator is more effective throughout the search. The combination may distract the overall search from concentrating on the use of that one crossover. There is also the cost of overheads involved in the combined version.

Another way to explain this shortcoming is perhaps that the PF of UF5 is discontinuous. Note that the combined version still performed as well as the SPX-based version and only slightly worse than the CMX-based version. This may point, as said above, to the fact that the CMX crossover operator is well adapted to that problem. The same goes for UF6.

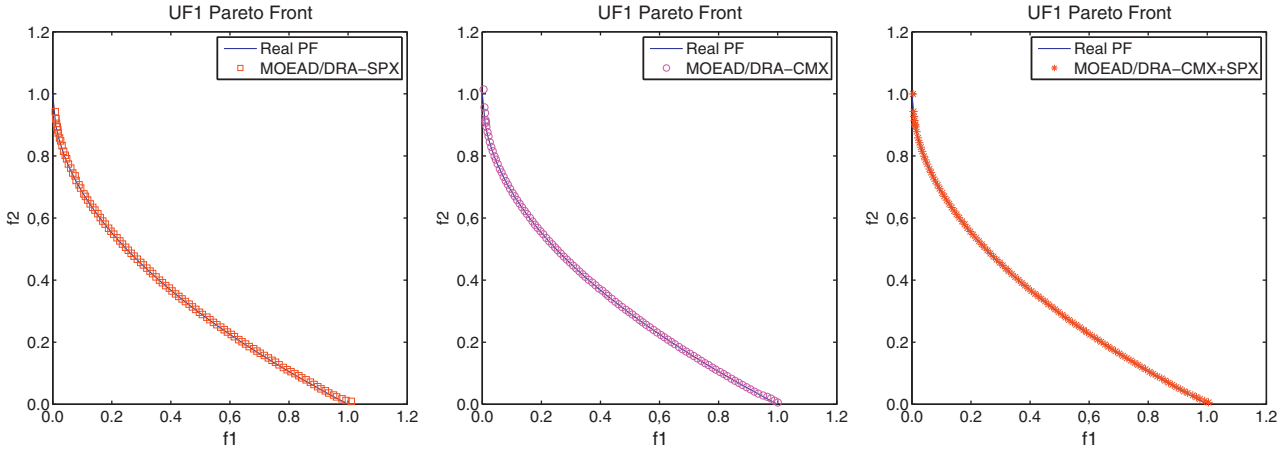

Fig. 1. Plots of the final non-dominated solutions with the lowest IGD-metric value over 30 independent runs in the objective space of UF1. The first figure is for MOEA/D-DRA-SPX, the second figure is for MOEA/D-DRA-CMX and the third figure is for MOEA/D-DRA-CMX+SPX.

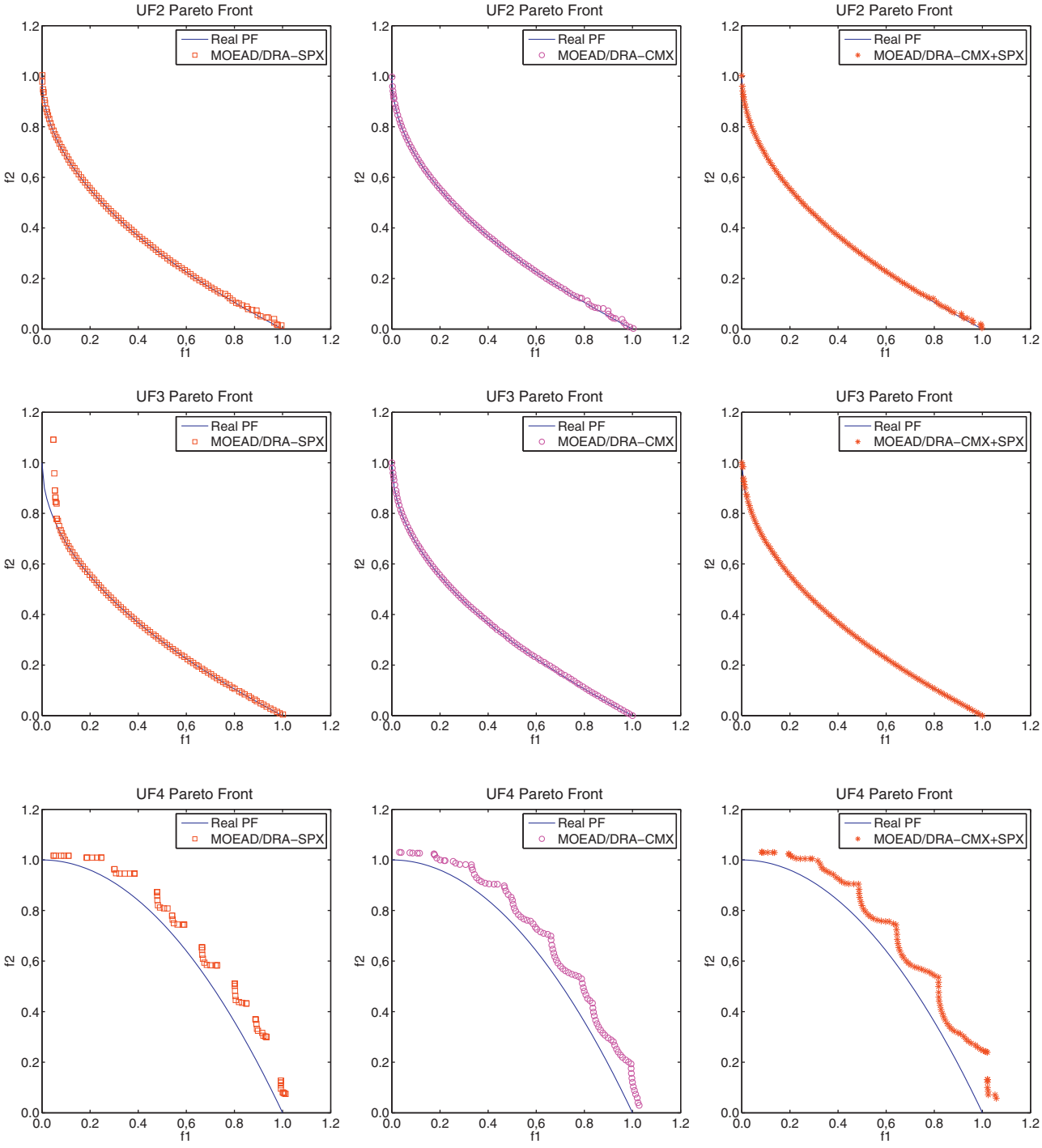

Fig. 2. Plots of the final non-dominated solutions with the lowest IGD-metric value over 30 independent runs in the objective space of UF2, . . ., UF4. The 1st panel is for MOEA/D-DRA-SPX, the 2nd is for MOEA/D-DRA-CMX, and the 3rd is for MOEA/D-DRA-CMX+SPX.

6. Impact of parameter settings on MOEA/D-DRA-CMX+SPX

In the following, we report on our experiences with MOEA/D-DRA-CMX+SPX when solving standard test problems with different settings of key parameters such as the population size (N), the neighborhood size (T), and the maximum number of subproblem solutions replaced (n_r) after each search cycle. The impact of these parameter settings on the performance of the suggested hybrid algorithm is in terms of the calculated IGD-metric values from the corresponding search results.

6.1. Performance of MOEA/D-DRA-CMX+SPX with different population sizes

MOEA/D-DRA-CMX+SPX was tested with population sizes $N = 200, 300, 400$, and 500 on 2-objective problems and with population sizes $N = 250, 500, 600$, and 800 on 3-objective problems. The results were compared with those obtained with MOEA/D-DRA-CMX and MOEA/D-DRA-SPX. The test problems are from the CEC'09 test suit.

Tables 4 and 5 show the IGD-metric values including best, median, mean, standard deviation and maximum obtained for

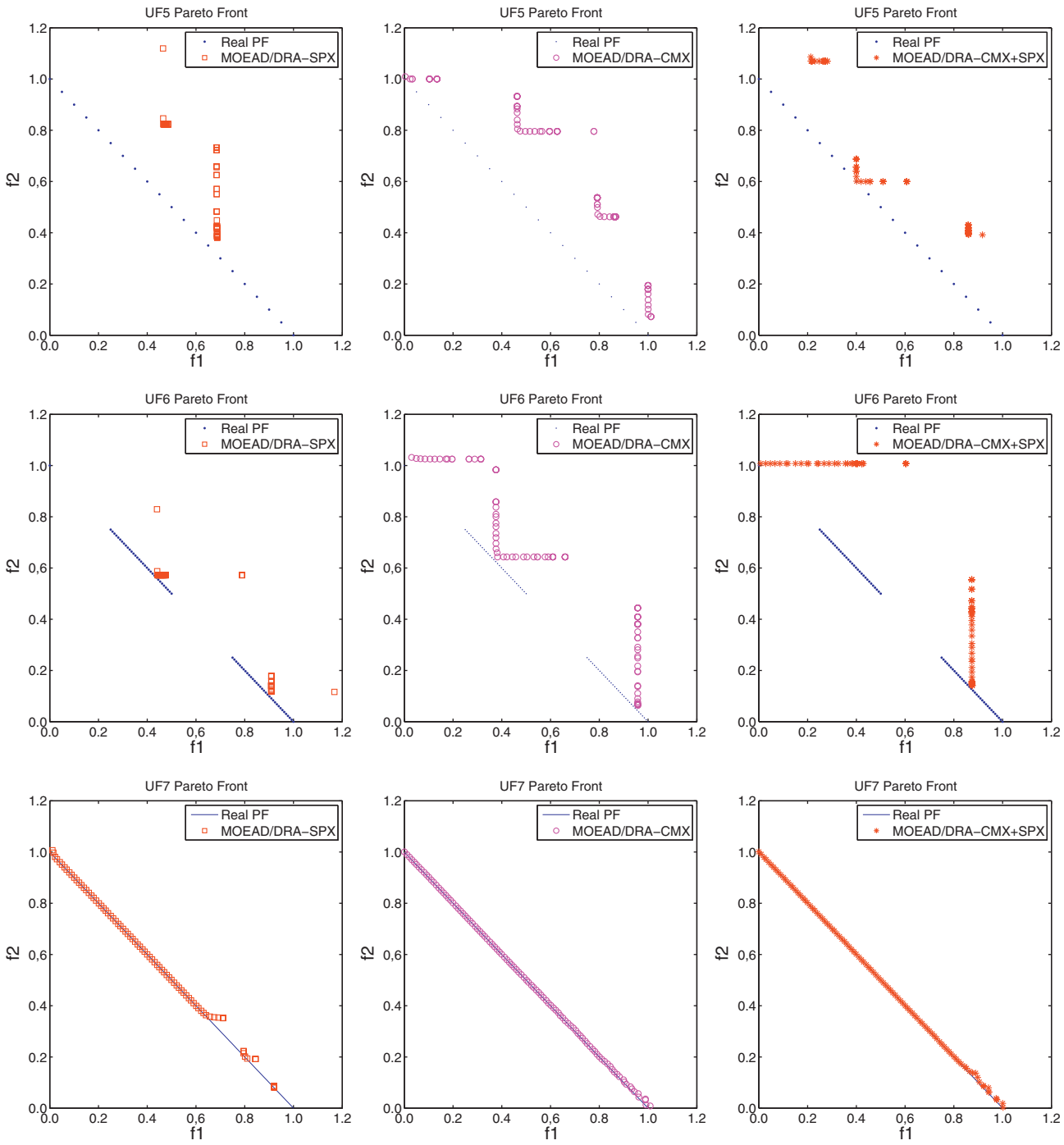

Fig. 3. Plots of the final non-dominated solutions with the lowest IGD-metric value over 30 independent runs in the objective space of UF5, . . ., UF7. The 1st panel is for MOEA/D-DRA-SPX, the 2nd is for MOEA/D-DRA-CMX, and the 3rd is for MOEA/D-DRA-CMX+SPX.

MOEA/D-DRA-CMX+SPX with different population settings (given in the last columns of these tables), when applied to each of the CEC'09 test instances [31].

Table 7 presents the IGD-metric statistics (i.e., best, median, mean, standard deviation and maximum) for MOEA/D-DRA-CMX with two different population sizes $N = 400, 500$, when applied to 2-objective problems, and $N = 500, 600$ when applied to 3-objective problems in the CEC'09 collection. Table 6 records the statistics of MOEA/D-DRA-SPX with two different population sizes, 400 and 500, when applied to 2-objective problems and sizes 500, and 600, when applied to 3-objective problems of the same collection. Note that, apart from N, the population size, the rest of parameters is the same as set in Section 4.2.

Tables 4 and 5 clearly indicate that MOEA/D-DRA-CMX+SPX is not very sensitive to population size compared to MOEA/D-DRA-CMX and MOEA/D-DRA-SPX. Furthermore, Tables 4 and 5 show that the performance of MOEA/D-DRA-CMX+SPX improves with increasing population size unlike that of MOEA/D-DRA-CMX and MOEA/D-DRA-SPX, in terms of reducing the average IGD-metric values on seven (i.e., UF1–UF7) out of the ten test instances.

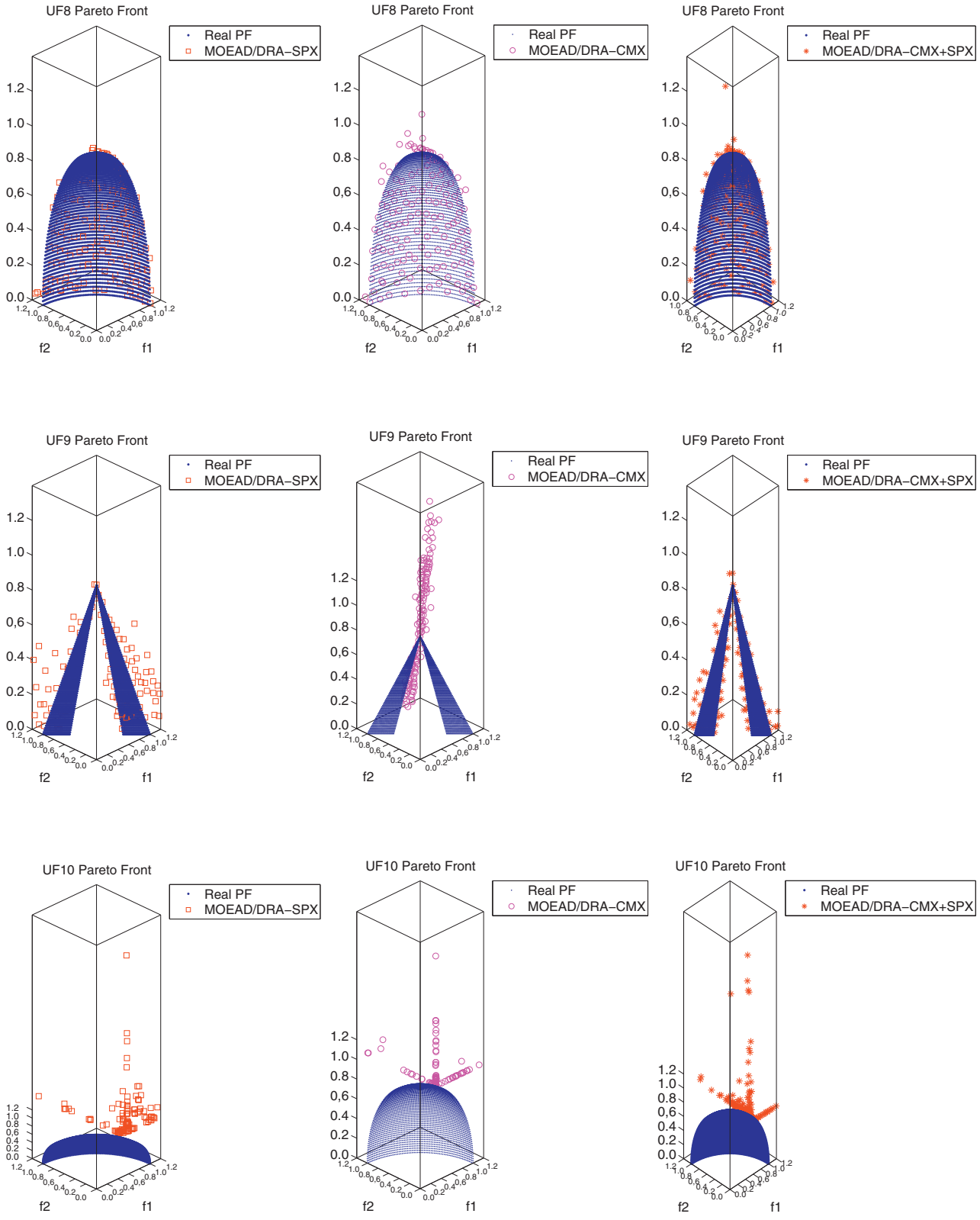

Fig. 4. Plots of the final non-dominated solutions with the lowest IGD-metric value of the three algorithms over 30 independent runs in the objective space of UF8, . . ., UF10. The 1st panel is for MOEA/D-DRA-SPX, the 2nd is for MOEA/D-DRA-CMX, and the 3rd is for MOEA/D-DRA-CMX+SPX.

6.2. *Impact of the neighborhood size on MOEA/D-DRA-CMX+SPX*

The neighborhood concept is one of the key features of the MOEA/D paradigm. Its size T is the most important control parameter in MOEA/D and its setting should be smaller than N as recommended in [29].

In this section, we study the sensitivity of MOEA/D-DRA-CMX+SPX to parameter T. All other parameters are kept the same as

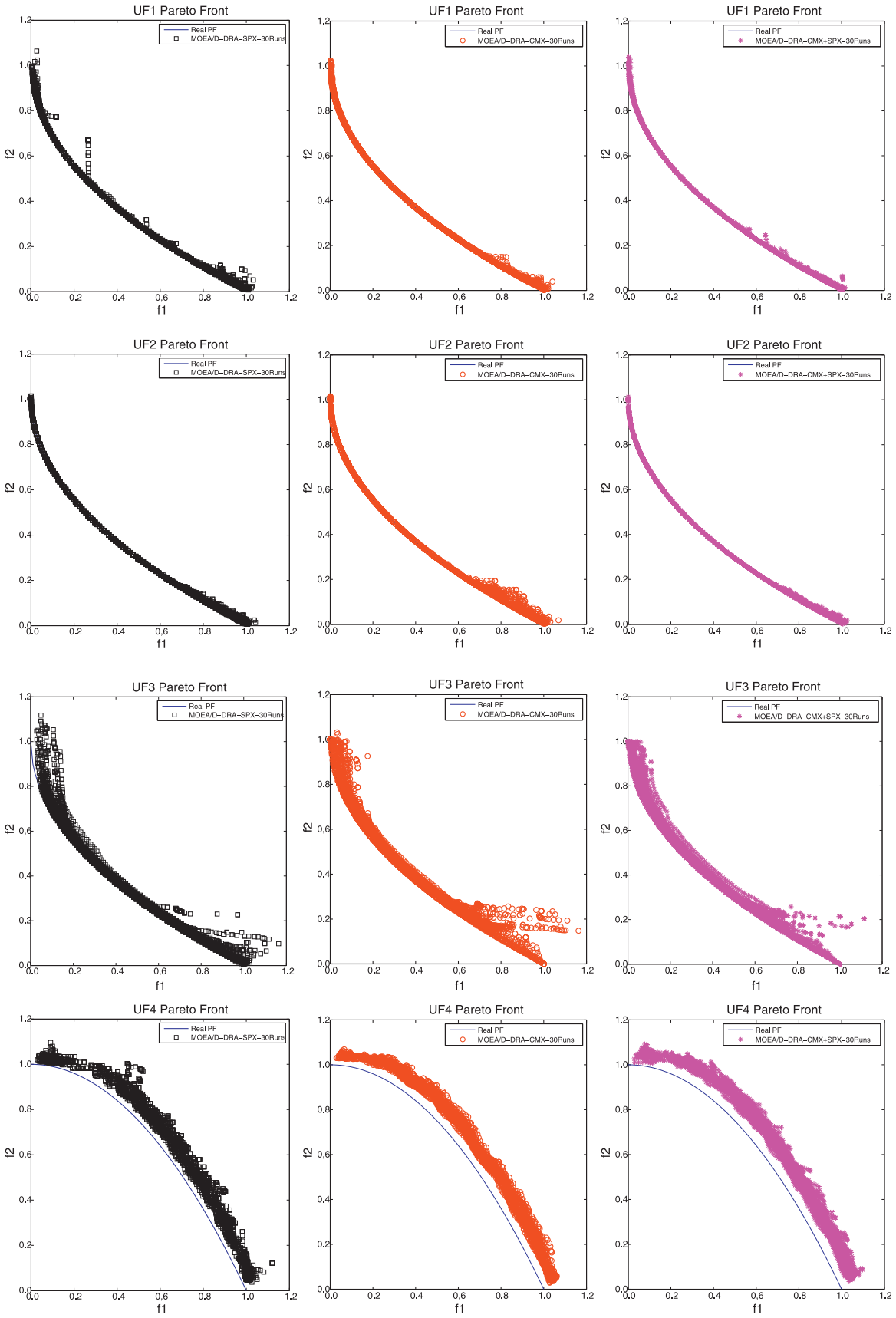

Fig. 5. Plots of all 30 Pareto fronts in the objective space found by MOEA/D-DRA-SPX for UF1, . . ., UF4. The 1st panel is for MOEA/D-DRA-SPX, the 2nd is for MOEA/D-DRA-CMX, and the 3rd is for MOEA/D-DRA-CMX+SPX.

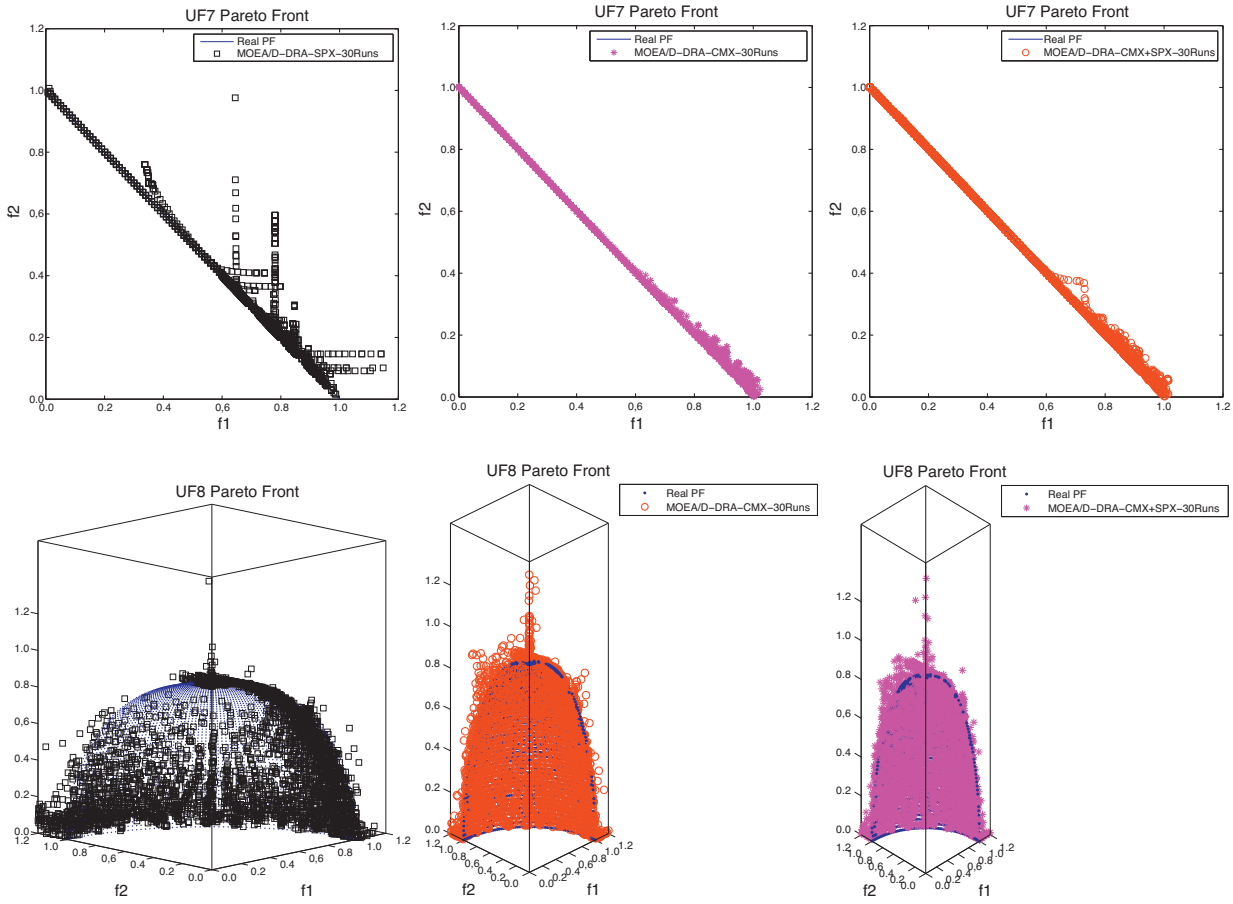

Fig. 6. Plots of all 30 Pareto fronts in the objective space found by MOEA/D-DRA-SPX+CMX for UF7, UF8. The 1st panel is for MOEA/D-DRA-SPX, the 2nd is for MOEA/D-DRA-CMX, and the 3rd is for MOEA/D-DRA-CMX+SPX.

set in Section 4.2, except T, of course. Fig. 9(center) plots the evolution of the average IGD-metric value against different settings of T in MOEA/D-DRA-CMX+SPX. This figure clearly shows that this algorithm performs very well with T in the range 50–100. It also works very well for all values of T except the very small ones and $T = N$. Thus, we can say that MOEA/D-DRA-CMX+SPX is not sensitive to T on at least the CEC'09 problems that we have considered here. Note that we have only included an exemplar graph of the results obtained for UF1.

The IGD-metric values in Tables 4 and 5 are obtained with MOEA/D-DRA-CMX+SPX with different population sizes (i.e., as explained in Section 6) and with different neighborhood sizes, $T = 20, 30, 40, 50$ for 2-objective and $T = 25, 50, 60, 80$ for 3-objective CEC'09 test instances [31], respectively.

Table 4
The IGD-metric values of MOEA/D-DRA-CMX+SPX over 30 independent runs with different population sizes N on the 2-objective CEC'09 test instances [31].

CEC'09	Min	Median	Mean	St.dev.	Max	N
UF1	0.012823	0.026487	0.041614	0.038193	0.153849	200
	0.004129	0.005037	0.006873	0.007852	0.046767	300
	0.005343	0.009221	0.026405	0.045996	0.172712	400
	0.004223	0.005579	0.006141	0.002063	0.013711	500
UF2	0.008142	0.010616	0.011191	0.002201	0.016848	200
	0.004974	0.005656	0.005798	0.000633	0.007276	300
	0.006223	0.007343	0.007427	0.000985	0.011199	400
	0.005426	0.006348	0.006437	0.000676	0.008557	500
UF3	0.047194	0.112486	0.141920	0.081489	0.325030	200
	0.004188	0.009407	0.013399	0.015863	0.086123	300
	0.005075	0.043814	0.047143	0.030288	0.112878	400
	0.004654	0.010830	0.011821	0.006879	0.034609	500
UF4	0.066841	0.080706	0.081409	0.008061	0.105816	200
	0.004188	0.009407	0.013399	0.015863	0.086123	300
	0.057589	0.070794	0.071627	0.005536	0.083088	400
	0.050508	0.056913	0.057854	0.005591	0.071978	500
UF5	0.368941	0.606737	0.623653	0.138410	0.882580	200
	0.214192	0.381650	0.399385	0.090454	0.707107	300
	0.329791	0.417681	0.432884	0.083484	0.631771	400
	0.223677	0.373221	0.421708	0.154778	0.707106	500

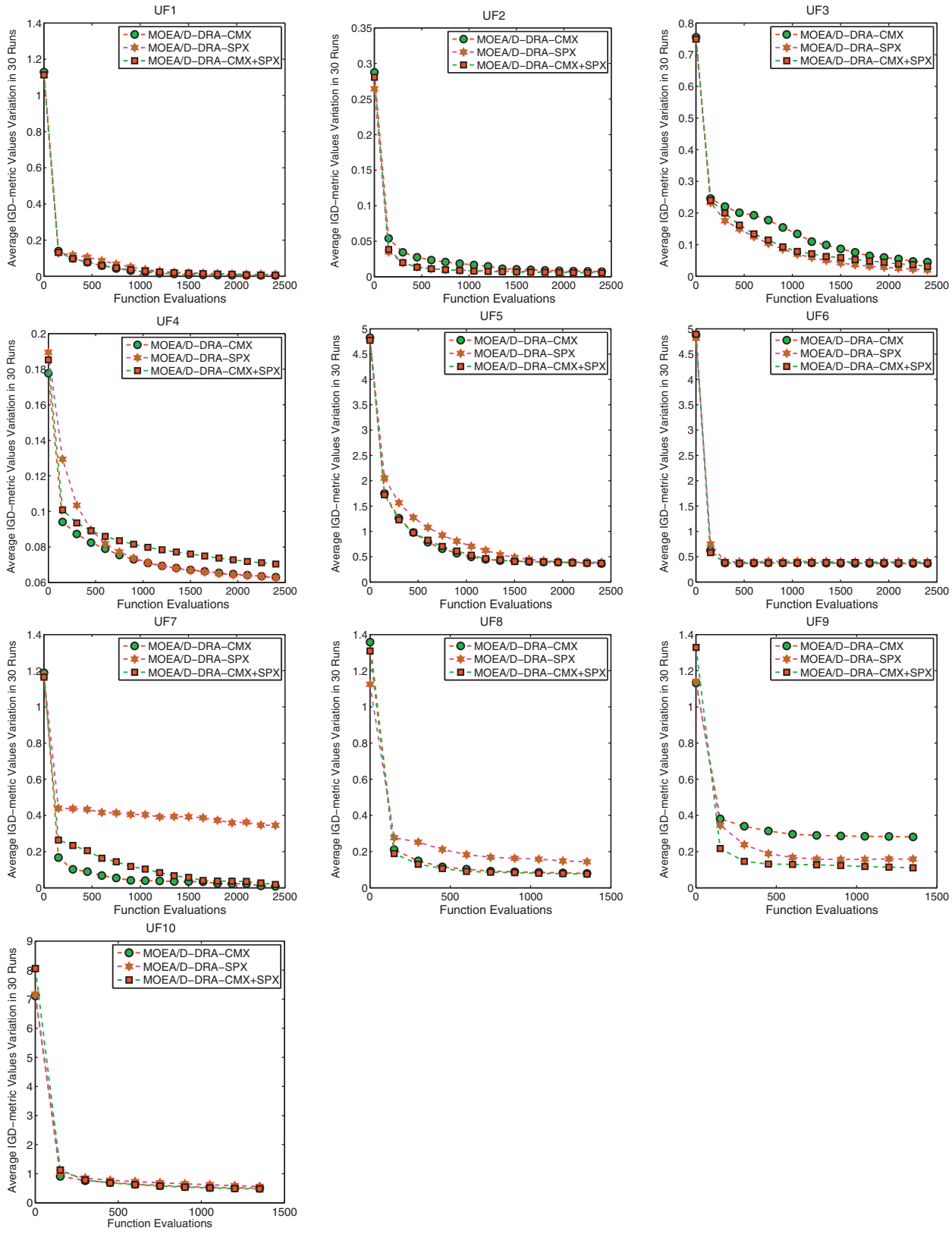

Fig. 7. Evolution of the average IGD-metric values versus the number of generations spent by MOEA/D-DRA-CMX, MOEA/D-DRA-SPX, and MOEA/D-DRA-CMX+SPX on UF1, . . ., UF6.

The left of Fig. 9 shows the evolution of the best run IGD-metric values over 30 runs on the CEC'09 instances. To address the issue of the impact on the performance of MOEA/D-DRA-CMX+SPX, we set $T = N$. The center of Fig. 9 shows that the performance of MOEA/D-DRA-CMX+SPX is worst when $T = 600$ in terms of the increase in the IGD-metric value (here displayed for instance UF1) compared to other T values. These results indicate that T should be set less than the population size for better performance on the IGD-metric.

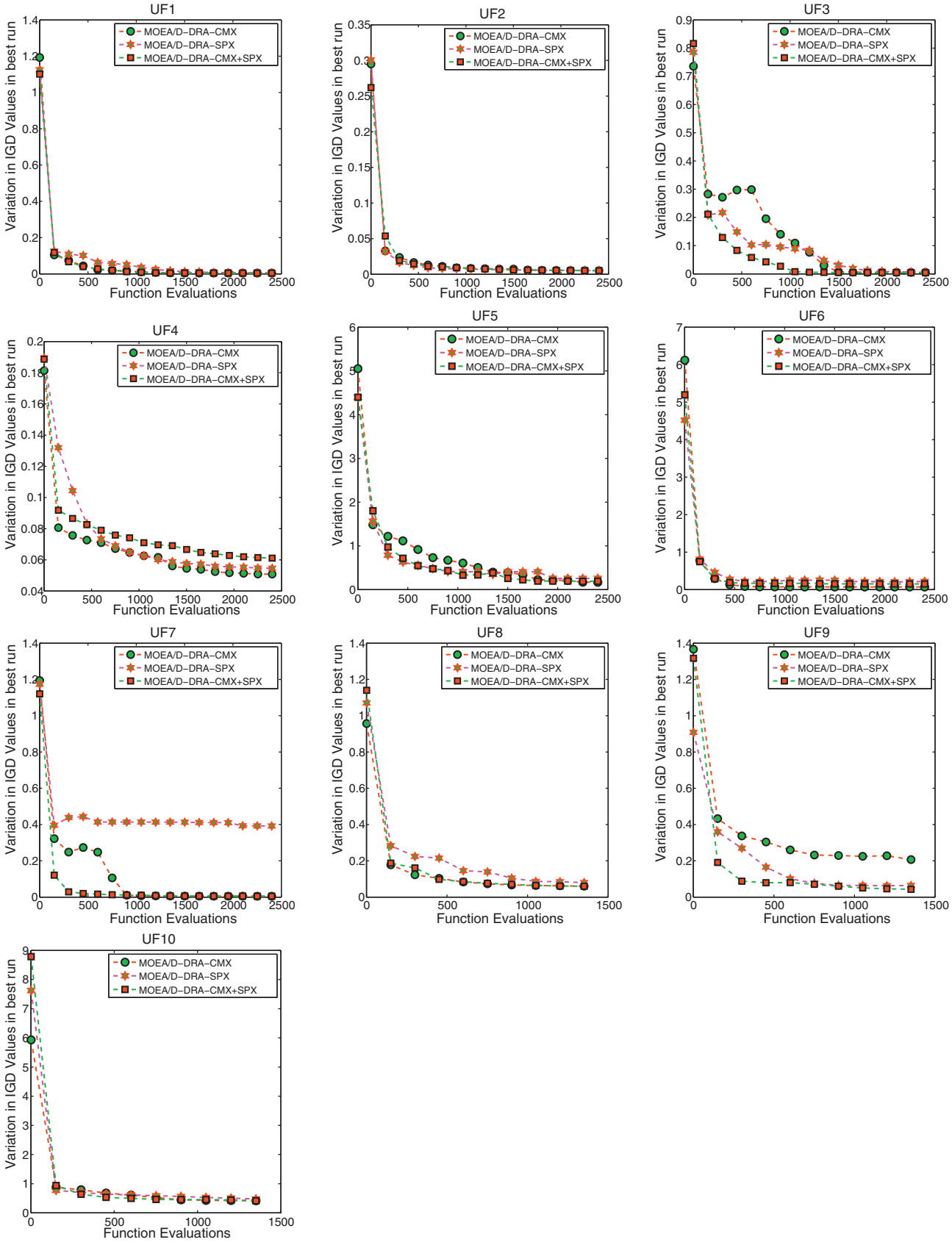

Fig. 8. Evolution of the IGD-metric values versus the number of generations of the best run over 30 independent runs of MOEA/D-DRA-CMX, MOEA/D-DRA-SPX, and MOEA/D-DRA-CMX+SPX on UF1, . . ., UF6.

Table 5
The IGD-metric values of MOEA/D-DRA-CMX+SPX over 30 independent runs with different population sizes on the 3-objective CEC'09 test instances [31].

CEC'09	Min	Median	Mean	St.dev.	Max	*N*
UF6	0.173843	0.443582	0.400633	0.134066	0.759079	200
	0.164252	0.428657	0.374936	0.139382	0.722266	300
	0.167471	0.444579	0.415866	0.130556	0.716430	400
	0.143108	0.442482	0.0382341	0.153729	0.804836	500
UF7	0.009082	0.019022	0.146966	0.183822	0.556820	200
	0.004466	0.005955	0.006501	0.002456	0.016846	300
	0.005969	0.009837	0.083261	0.162711	0.542298	400
	0.005045	0.006124	0.00690	0.001872	0.014013	500
UF8	0.112062	0.135966	0.148222	0.035861	0.244353	250
	0.060292	0.098290	0.105947	0.035883	0.217259	500
	0.051823	0.067814	0.070392	0.016783	0.110421	600
	0.055175	0.0758446	0.0885577	0.029123	0.182638	800
UF9	0.114800	0.202512	0.202146	0.023488	0.260696	250
	0.048576	0.162500	0.148970	0.039373	0.189787	500
	0.059098	0.152641	0.111974	0.032441	0.118181	600
	0.053043	0.079842	0.07828	0.020178	0.15234	800
UF10	0.428427	0.514462	0.518051	0.057482	0.611057	200
	0.364067	0.457048	0.459822	0.045823	0.546396	500
	0.406185	0.460197	0.465373	0.037577	0.530965	600
	0.314962	0.446823	0.451374	0.045025	0.530874	800

Table 6
The IGD-metric values of MOEA/D-DRA-SPX over 30 independent runs with population sizes *N* used on CEC'09 test instances [31].

CEC'09	Best	Median	Mean	St.dev.	Worst	*N*
UF1	0.010524	0.024126	0.032884	0.030600	0.167038	300
	0.007577	0.013552	0.025604	0.041755	0.210340	400
UF2	0.007600	0.008815	0.008911	0.000829	0.010604	300
	0.006783	0.008209	0.008298	0.000699	0.010667	400
UF3	0.012746	0.058641	0.059622	0.030882	0.140573	300
	0.007767	0.037966	0.041621	0.026521	0.108513	400
UF4	0.061380	0.067608	0.067832	0.004257	0.077741	300
	0.057845	0.063680	0.063797	0.004520	0.074215	400
UF5	0.350164	0.448279	0.459738	0.079113	0.630960	300
	0.266307	0.406658	0.421402	0.073858	0.577894	400
UF6	0.174771	0.419950	0.370526	0.096431	0.462408	300
	0.185385	0.443906	0.383862	0.097282	0.475299	400
UF7	0.008347	0.191911	0.190371	0.162191	0.526301	300
	0.007253	0.022904	0.173042	0.213450	0.574394	400
UF8	0.064486	0.093309	0.091978	0.018175	0.131712	500
	0.053873	0.064945	0.071863	0.017748	0.116492	600
UF9	0.048333	0.151325	0.120079	0.047077	0.193034	500
	0.040426	0.148211	0.131620	0.039884	0.158120	600
UF10	0.385331	0.493508	0.489788	0.050616	0.574868	500
	0.245726	0.506193	0.493731	0.077428	0.635953	600

Table 7
The IGD-metric values of MOEA/D-DRA-CMX over 30 independent runs on CEC'09 test instances [31].

CEC'09	Best	Median	Mean	St.dev.	Worst	*N*
UF1	0.010524	0.024126	0.032884	0.030600	0.167038	300
	0.086596	0.113851	0.115079	0.014754	0.162955	400
UF2	0.022165	0.027052	0.027051	0.001940	0.030585	300
	0.020010	0.022568	0.022696	0.001549	0.025608	400
UF3	0.082230	0.119640	0.230061	0.195813	0.594848	300
	0.065553	0.114643	0.269749	0.227045	0.595418	400
UF4	0.081325	0.097801	0.097550	0.010116	0.116335	300
	0.077429	0.095759	0.094752	0.010616	0.112952	400
UF5	1.889568	2.794576	2.726526	0.406303	3.513249	300
	1.249339	2.392959	2.364813	0.401596	3.123953	400
UF6	0.619284	1.364154	1.280091	0.363662	1.862737	300
	0.590681	1.148391	1.165530	0.278422	1.799595	400
UF7	0.036360	0.050242	0.065372	0.088558	0.532118	300
	0.024297	0.033658	0.047508	0.054138	0.320080	400
UF8	0.095381	0.195370	0.192308	0.045898	0.281929	500
	0.103358	0.192468	0.182471	0.041294	0.242652	600
UF9	0.061337	0.169943	0.143623	0.049655	0.193465	500
	0.055389	0.172894	0.153690	0.046937	0.199654	600
UF10	0.184626	0.196215	0.320123	0.215095	0.787695	500
	0.178995	0.197305	0.258797	0.150136	0.778593	600

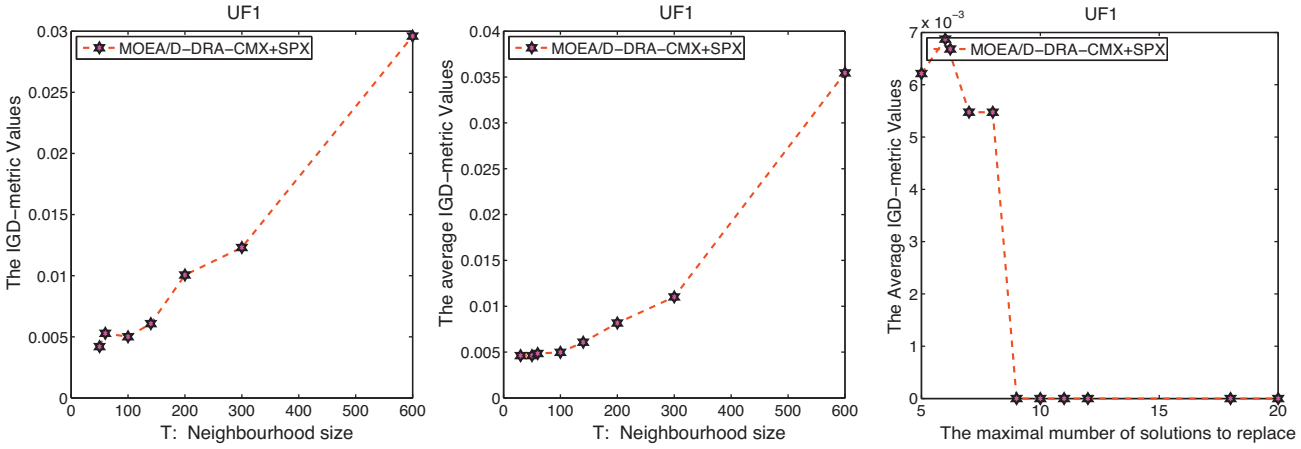

Fig. 9. (Left) Plot of IGD-metric values against T values in MOEA/D-DRA-CMX+SPX on UF1; (center) plot of average IGD-metric values against T values in MOEA/D-DRA-CMX+SPX on UF1; (right) plot of average IGD-metric values against values of n_r in MOEA/D-DRA-CMX+SPX for UF1.

6.3. Impact of the maximal replacement number restriction on MOEA/D-DRA-CMX+SPX

We have also investigated the sensitivity of MOEA/D-DRA-CMX+SPX to parameter n_r, i.e. the maximal number of solutions to be replaced in the neighborhood of each subproblem, to maintain diversity in the current population. Here we only report an exemplar outcome of this investigation in the case problem UF1.

Fig. 9(right) plots the average IGD-metric values of MOEA/D-DRA-CMX+SPX on problem UF1, against n_r values 3, 4, 5, 6, 7, 8, 9, 10, and 20. All other parameters are kept the same as in Section 4.2. It is suggested in [30] that n_r should be set smaller than T in order to maintain good diversity in the population. The experiments were carried out with this suggestion in mind, as can be seen in Fig. 9(right). The figure also clearly indicates that the IGD-metric values of MOEA/D-DRA-CMX+SPX are decreasing from 8 to 20 on UF1.

Note that, since $n_r = 0.01N$, the IGD-metric statistics reported in Table 4 are obtained with $n_r = 2, 3, 4, 5$ for each of the 2-objective test problems and those reported in Table 5 are with $n_r = 3, 5, 6, 8$ for the 3-objective ones.

7. Conclusion and further work

Different crossover operators suit different problems. Using multiple crossover operators should be an effective strategy for avoiding the use of potentially unsuitable crossovers. In this paper, we have studied the combined use of two crossover operators in MOEA/D-DRA [30], and developed a hybrid version of it, referred to as MOEA/D-DRA-CMX+SPX using a probabilistic approach to decide which crossover is to be used in a given generation. The latter aspect is what we refer to throughout as resource allocation. The resource allocation is in terms of the frequency with which a given crossover is used. This is implemented through a reward system which is just the probability of use of each crossover. There are many possible ways to apportion probabilities to crossovers in addition to the one used here. We have already carried out some experiments with an alternative approach. The results will be the subject of a future paper.

Results show that, based on IGD-metric values, MOEA/D-DRA-CMX+SPX is more effective on most of the CEC'09 test instances [31], than either versions of MOEA/D-DRA using CMX or SPX as the crossover operator. These test problems are more difficult than the ones described in [17], for instance. However, it is still sensible to carry out tests on these problems and others with complicated PS shapes [3]. Again, we are looking at carrying out such experiments in the future.

Furthermore, it is perhaps worthwhile considering a generalization of the multi-crossover approach described here, to any number of crossover operators when no obvious one can be chosen.

Acknowledgments

We are grateful to the referees for their useful comments and to Professor Qingfu Zhang for his help.

References

[1] J. Branke, K. Deb, K. Miettinen, R. Slowinski (Eds.), Multiobjective Optimization: Interactive and Evolutionary Approaches, Springer, 2008.
[2] Z. Cai, Y. Wang, A multiobjective optimization-based evolutionary algorithm for constrained optimization, IEEE Transactions on Evolutionary Computation 10 (December (6)) (2006) 658–675.
[3] C.K. Chow, S.Y. Yuen, A multiobjective evolutionary algorithm that diversifies population by its density, IEEE Transactions on Evolutionary Computation (99) (2011) 1–24.
[4] C.A.C. Coello, A comprehensive survey of evolutionary-based multiobjective optimization techniques, Knowledge and Information Systems 1 (1999) 269–308.
[5] C.A. Coello Coello, G.B. Lamont, D.A. Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, New York, March 2002.
[6] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, 1st ed., John Wiley and Sons, Chichester, UK, 2001.
[7] K. Deb, in: S. Ross, R. Weber (Eds.), Multi-Objective Optimization Using Evolutionary Algorithms, 2nd ed., John Wiley and Sons Ltd., 2002.
[8] K. Deb, A. Anand, D. Joshi, A computationally efficient evolutionary algorithm for real-parameter optimization, Evolutionary Computation 10 (2002) 371–395.
[9] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2) (2002) 182–197.
[10] F.Y. Edgeworth, Mathematical Psychics, P. Keagan, London, England, 1881.
[11] M. Ehrgott, Multicriteria Optimization, 2nd ed., Springer, Berlin, 2005.
[12] L.J. Eshelman, J.D. Schaffer, Real-coded genetic algorithms and interval-schemata, in: Foundations of Genetic Algorithms, 1993, pp. 187–202.
[13] J. Horn, N. Nafpliotis, D.E. Goldberg, A niched Pareto genetic algorithm for multiobjective optimization, in: Proceedings of the First IEEE Conference on Evolutionary Computation, CEC'94, 1994.
[14] J. Knowles, D. Corne, Properties of an adaptive archiving algorithm for storing nondominated vectors, IEEE Transactions on Evolutionary Computation 7 (April (2)) (2003) 100–116.
[15] K. Deb, R. Agrawal, Simulated binary crossover for continuous search space, Complex System 9 (1995) 115–148.
[16] H. Li, Combination of Evolutionary Algorithms with Decomposition Techniques for Multiobjective Optimization, Ph.D., Department of Computer Science, University of Essex, Wevehoe Park, Colchester, Essex, UK, 2007.
[17] H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets: MOEA/D and NSGA-II, IEEE Transactions on Evolutionary Computation 13 (April (2)) (2009) 284–302.

[18] K.M. Miettinien, Nonlinear Multiobjective Optimization, Academic Publishers Kluwer, Norwell, MA, 1999 (ser. Kluwer's International Series).
[19] M.M. Raghuwanshi, P.M. Singru, U. Kale, O.G. Kakde, Simulated binary crossover with logonormal distribution, Complexity International 12 (2005) 1–10.
[20] N. Srinivas, K. Deb, A multiobjective optimization using nondominated sorting in genetic algorithms, Journal of Evolutionary Computation 2 (3) (1994) 221–248.
[21] I. Ono, S. Kobayashi, A real-coded genetic algorithm for function optimization using the unimodal normal distribution crossover, in: 7th International Conference on Genetic Algorithms, 1997, pp. 246–253.
[22] I. Ono, H. Kita, S. Kobayashi, A Real-Coded Genetic Algorithm Using the Unimodal Normal Distribution Crossover, Springer-Verlag New York, Inc., New York, NY, USA, 2003, pp. 213–237.
[23] R. Peltokangas, A. Sorsa, Real Coded Genetic Algorithm and Nonlinear Parameter Identification, Control Engineering Laboratory University of Oulu, Report A No. 34, April 2008.
[24] M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J.J.M. Guervós, H.-P. Schwefel (Eds.), Proceedings of the 6th International Conference on Parallel Problem Solving from Nature – PPSN VI, vol. 1917, Paris, France, September 18–20, Springer, 2000, Ser. Lecture Notes in Computer Science.
[25] 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, Springer-Verlag, London, UK, 1998, pp. 428–437.
[26] S. Tsutsui, A. Ghosh, A study on the effect of multi-parent recombination in realcoded genetic algorithms, in: Proc. of the IEEE ICEC, 1998, pp. 828–833.
[27] S. Tsutsui, M. Yamamura, T. Higuchi, Multi-parent recombination with simplex crossover in real coded genetic algorithms, in: Proceeding of GECCO-99, 1999, pp. 657–674.
[28] D.A.V. Veldhuizen, G.B. Lamont, Evolutionary Computation and Convergence to a Pareto Front, Morgan Kaufmann, 1998, Stanford University, California, pp. 221–228.
[29] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (December (6)) (2007) 712–731.
[30] Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA/D on CEC'09 unconstrained MOP test instances, in: IEEE Congress on Evolutionary Computation (IEEE CEC 2009), Trondheim, Norway, May 18–21, 2009, pp. 203–208.
[31] Q. Zhang, A. Zhou, S. Zhaoy, P.N. Suganthany, W. Liu, S. Tiwariz, Multiobjective optimization test instances for the CEC 2009 special session and competition, Technical Report CES-487, 2009.
[32] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength Pareto evolutionary algorithm, in: Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland, TIK Report 103, 2001.
[33] E. Zitzler, M. Laumanns, L. Thiele, C.M. Fonseca, V. Grunert da Fonseca, Why quality assessment of multiobjective optimizers is difficult, in: Genetic and Evolutionary Computation Conference (GECCO 2002), July, Morgan Kaufmann Publishers, New York, NY, USA, 2002, pp. 666–674.
[34] E. Zitzler, L. Thiele, An evolutionary approach for multiobjective optimization: the strength Pareto approach, in: Computer Engineering and Networks Laboratory (TIK), ETH Zurich, TIK Report 43, May 1998.
[35] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Transactions on Evolutionary Computation 7 (2003) 117–132.