

# IMPACT ANALYSIS OF CROSSOVERS IN MULTIOBJECTIVE EVOLUTIONARY ALGORITHM

Wali Khan Mashwani<sup>1</sup>, Abdel Salhi<sup>2</sup>, Muhammad Asif Jan<sup>3</sup>, Rasheeda Adee Khanum<sup>4</sup> and M.Sulaiman<sup>5</sup>

<sup>1,3</sup>Department of Mathematics, Kohat University of Science & Technology (KUST), Pakistan.

<sup>3</sup>Jinnah College for Women, University of Peshawar, KPK, Pakistan.

<sup>4</sup>Department of Mathematical Sciences, University of Essex, Colchester, UK

<sup>5</sup>Department of Mathematics, Abdul Wali Khan University, Mardan, KPK, Pakistan

E-mail: mashwanigr8@gmail.com

**ABSTRACT:** Multi-objective optimization has become mainstream because several real-world problems are naturally posed as a Multi-objective optimization problems (MOPs) in all fields of engineering and science. Usually MOPs consist of more than two conflicting objective functions and that demand trade-off solutions. Multi-objective evolutionary algorithms (MOEAs) are extremely useful and well-suited for solving MOPs due to population based nature. MOEAs evolve its population of solutions in a natural way and searched for compromise solutions in single simulation run unlike traditional methods. These algorithms make use of various intrinsic search operators in efficient manners. In this paper, we experimentally study the impact of different multiple crossovers in multi-objective evolutionary algorithm based on decomposition (MOEA/D) framework and evaluate its performance over test instances of 2009 IEEE congress on evolutionary computation (CEC'09) developed for MOEAs competition. Based on our carried out experiment, we observe that used variation operators are considered to main source to improve the algorithmic performance of MOEA/D for dealing with CEC'09 complicated test problems.

**Keywords:** Multi-objective optimization problems (MOPs), Evolutionary Algorithms (EAs), Decomposition, Crossovers.

## 1. INTRODUCTION

Optimization is process of finding the most or best satiable solution for optimization and search problems. Practical application of optimization begun much before the Second World War where the distribution of forces in battlefield and allocation accessories to them in well organized and optimal manner were quite necessitated at that times. In essence, optimization problems can be divided into two main categories: combinatorial and continuous problems. A problems with continuous variables are called continuous problem and problems with discrete variables are called combinatorial optimization problem. Travelling salesman problem (TSP) [11] and minimum spanning tree problems (MST) [64] are widely used combinatorial optimization problems. Several other paractical application of combinatorial optimization including the development of best airline network of spokes and destinations, deciding which taxis in a fleet to route to pick up fares, network design for telephone, electrical, hydraulic, TV cable, computer and road to deliver packages. Multi-objective optimization is the process of finding a set of optimal solutions for search and optimization problems. Many real-world problems are naturally arise in form of multi-objective optimization problems (MOPs) [9]. These problems offer big challenges for both researchers and practitioners in all discipline of sciences and engineering. Examples of real-world are extensively studied in the existing literature of the evolutionary computing (EC) [59, 6, 8, 73, 54, 2]. In general, MOP can be mathematically formulated as under:

$$\text{Minimize } F(x) = (f_1(x) + f_2(x) + \dots + f_m(x))^T \quad (1)$$

Subjected to  $x \in \Omega$

Where  $\Omega$  is the decision variable space,  $x = (x_1, x_2, \dots, x_n)^T$  is decision variable vector/ individual with  $n$  decision variables,  $R^m$  is the objective space containing  $m$  conflicting objective functions. If  $\Omega$  is closed, connected in region  $R^n$  and all their objective functions are continuous then problem (1) is said to be continuous MOP [32]. In

addition, if  $m > 3$ , then problem (1) is said to be many objective optimization problem.

A solution  $u = (u_1, u_2, \dots, u_n) \in \Omega$  is said to be Pareto optimal if there does not exist another solution  $v = (v_1, v_2, \dots, v_n) \in \Omega$  such that  $f_j(u) \leq f_j(v)$  for all  $j = 1, 2, \dots, m$  and  $f_j(u) < f_j(v)$  for at least one index  $k$ . An objective vector is said to be Pareto optimal if their corresponding decision vector is Pareto optimal in decision space. All the Pareto optimal solutions in the decision space of the targeted problem is called Pareto set (PS) and their image in objective space is called Pareto front (PF) [13]. This idea of Pareto optimality was first proposed by Francis Ysidro Edgeworth in 1881 [17] and then later on generalized by Vilfredo Pareto in 1986 [7]. The primarily goals in tackling MOP is to find their approximated set of optimal solutions that much closer to their true Pareto front (PF), and also the approximated solutions should desirably uniformly distribute along the true PF with high density.

Multi-objective evolutionary algorithms (MOEAs) are well-known population based techniques for dealing with diverse sets of test MOPs and real-world problems. In last two decades, many different types of evolutionary algorithms (EAs) have been developed in the specialized literature of evolutionary computing [73, 39, 31, 78] and they are successfully applied on different complicated optimization and search problems [66, 50, 27, 37, 37, 6, 20, 33]. In general, classical MOEAs can be divided into three main families based on selection rules of candidate solutions: 1) Aggregation functions based MOEAs (i.e., [25, 24, 46, 67, 32, 69]), 2) dominance-based approaches (e.g., [15, 74, 21, 47, 23, 22]), 3) Indicator based EAs (IBEA) [76, 4, 3]. In last mentioned above two groups, decomposition concept is not purely used and they treat a given MOP as a whole or directly. On the other hand, aggregation based EAs associate their solutions of population with scalar optimization problem.

MOEA/D: multi-objective evolutionary algorithm based on decomposition [67] is novel aggregation based MOEAs that decomposes the given MOP into a number of different scalar optimization subproblems (SOPs). It then optimizes all SOPs simultaneously using generic population-based algorithm. Recently, several enhanced versions of MOEA/D have been suggested in the existing literature of EC [35, 38]. In [32], two different neighbourhoods systems along with restricted replacement strategy have been introduced for solving complicated test problems. Different subproblems of MOEA/D [67] were required different amounts of computational resources. A strategy of dynamical resource allocation have been introduced in [69] and with induction of this strategy MOEA/D-DRA [69] has been nominated as the winner of MOEAs competition. Gaussian Process Model has been integrated in original MOEA/D [67] to handle an expensive MOPs [70]. In [51], each subproblem records more than one solution for maintaining search diversity in their population.

In [30, 40, 36, 41], multiple different search operators with novel dynamic resources allocation strategies have been introduced in MOEA/D paradigm for solving commonly used ZDT test problems [75] and recently formulated complicated CEC'09 test instances [71]. In [42, 34], a combination of MOEA/D and NSGA-II has been suggested in the form of multimethod for coping with hard multi-objective optimization problems. Two different aggregation functions have been integrated at the same time in MOEA/D framework [26] for combinatorial MOPs. A new NBI-style Tchebycheff approach has been adopted in [68] coping with portfolio optimization problems. A decomposition-based multi-objective evolutionary algorithm with an ensemble of neighbourhood sizes (ENS-MOEA/D) has been proposed for solving CEC'09 test instance [72]. In ENS-MOEA/D, ensemble strategy of using two neighbourhood sizes (NSs) with online self-adaptation procedure has been proposed for the purpose to overcome the user-specific tuning parameter of neighbourhood size (T) adopted in the MOEA/D framework [69]. In [29], ant colony optimization (ACO) has been incorporated as local search technique in MOEA/D paradigm for solving the multi-objective Knapsack problems (MOKPs) and the multi-objective traveling salesman problem (MTSPs). The strategy of adaptive weight vector adjustment in MOEA/D has introduced and the proposed algorithm has been denoted by (MOEA/D-AWA) in [55]. The original MOEA/D paradigm is based on fixed weight vectors mechanism. In [36, 41], differential evolution and particle swarm optimization have been used for population evolution with adaptive procedure in the framework of original MOEA/D [67] for solving the ZDT test problems [75] and CEC'09 test instances [71]. In [40], a decomposition based hybrid evolutionary algorithm with dynamic resources allocation has been developed for solving the CEC'09 test instances. Recently, several latest versions of MOEA/D have been reviewed in the [30].

Different crossover operator's suite different optimization and search problems. One operator can be suitable for one types of problems that might be not suitable for other types of problems. In general, the performance of EAs are greatly

effected with the employment of different evolutionary operators.

In the last two decades, different types of crossover operators have been proposed such as BLX-a [19], simulated binary crossover (SBX) [28, 45], simplex crossover (SPX) [5, 62], centre of mass crossover (CMX) [60, 61], unimodal normally distributed crossover (UNDX) [48, 49], parent-centric crossover (PCX) [14], and many other real coded crossover operators [13, 53]. Many studies regarding the effects of the use different multiple crossovers have been already studied using single objective problems (SOPs) [58, 63, 65, 16]. However, very few research studies have been conducted with the same line for dealing with MOPs. Therefore, it is reasonable for us to experimentally analysis the impact of different multiple crossover operators. We employ CMX [60, 61], (SPX) [5, 62], adaptive DE [56], PCX [14], modified PCX (MPCX) [1] and quadratic interpolation crossover (QIX) in the framework of the MOEA/D [69] and examine their behaviours using the complicated CEC'09 test instances [71].

In this paper, our main objective is not to develop a novel search algorithm. We empirically examine the behaviours of different multiple crossovers one by one in existing MOEA/D [69] framework using the IEEE CEC '09 test instances [71]. Our hypothesis is that the performance of MOEA/D are greatly dependable on search operators .

The rest of this paper is organized as follows. Section 2 describes mathematical formulation of different employed crossover operators. Section 3 provides the algorithm of MOEA/D. Section 4.3 dedicated to experimental set up and discussion. Section 5 concludes this paper.

## 2. Crossovers

In essence, crossover operators enhance the exploration search abilities of EAs . On the other hand, mutation operators promoted the diversity in population of EAs to escape its population to get stuck in local optima of the problem. A crossover operator operates on more than one parent solutions while mutation applies to single solution. In past many years, variety of crossover and mutation operators have been designed by many researches in the existing literature of evolutionary computing consist of four classical paradigms: genetic algorithm (GA), genetic programming (GP), evolutionary strategy (ES) and evolutionary programming (EP). In following, we explain some those crossovers which are employed in the study of this paper.

### 2.1 Simplex Crossover (SPX)

SPX operates on three solutions,  $x^1, x^2, x^3$  and generate three new solutions (i.e., offspring) as follows:

$$y^k = (\mathbf{1} + \tau)(x^k - \mathbf{O}), \quad k = 1, 2, 3 \quad (3)$$

Where  $\mathbf{O} = \sum_{k=1}^3 x^k$  is the centre of mass and  $\delta \geq 0$  is the scaling parameter that controls the expansion of the simplex. In our implementation, an offspring solution is produced as follow [60]:

$$z = \sum_{k=1}^3 r^k y^k + \mathbf{O}, \quad (3)$$

$$\sum_{k=1}^3 r^k = \mathbf{1} \quad (4)$$

Where  $r^k$  are  $k$  random numbers and must be greater than zero.

**2.2 Centre of Mass Crossover (CMX)**

Given three parent solutions,  $x^1, x^2, x^3$ , then CMX works as follow

Then create a set of virtual mates by mirroring each parent across the centre of mas as:

$$v^j = 2O - x^i \quad (7)$$

In this paper, we have selected randomly both virtual mate  $v^j$  and parent  $x^i$  to generate an offspring solution as under

$$y = (1 - \alpha)x^i + v^j \quad (8)$$

Here, we have used  $\alpha = 2r - 0.5$ ,  $r$  is the random number belongs to  $[0,1]$ .

**2.3 Differential Evolution**

Differential evolution (DE) is a reliable and versatile optimizer. It was developed by proposed by Storn and Price [56, 57] for continuous problems. Since then, it has been successfully applied to diverse set of test and real-world optimization problems. DE has many enhanced versions which are recently reviewed in [39]. DE has two control parameters,  $F$  and  $CR$  and have had great influence in their process of evolution. DE was came up with the idea of using vector differences for perturbing the vector population. A simple mutation strategies of DE is formulated as follow:

$$y \sim = x^1 + F(x^2 - x^3) \quad (2)$$

Where  $x^2$  and  $x^3$  are two random solutions different from  $x^1$  and  $F$  is the scaling factor which controls the difference of two solutions  $x^2$  and  $x^3$ . An offspring solution  $y$  is generated as under

$$y = \begin{cases} y \sim & \text{if } u_r \leq CR \\ x^1 & \text{otherwise} \end{cases} \quad (3)$$

Here  $u_r \in [0,1]$  is a uniformly random number. The values of  $F$  and  $CR$  are adaptively settled in our implementation (i.e.  $F = CR = 0.5(1+r)$ ,  $r$  denotes random number. For our convenience, we called it adaptive DE (ADE) in this paper.

**2.4 Parent Centric Crossover (PCX)**

PCX [14] is based on the formulation of the UNDX. It generates an offspring solutions as follows:

$$y = x^p + w_\zeta |d^p| + \sum_{i=1, i \neq p} w_\eta \bar{D} e^i \quad (4)$$

$$d^p = x^p - g \quad (5)$$

Where  $x^p$  is a parent solution that selects with an equal probability for each offspring  $y$ ,  $g$  is mean of  $\mu$  parents,  $\bar{D}$  is the average perpendicular distance of the  $D^i$  perpendicular distances of  $\mu - 1$  parents,  $e^i$  are  $\mu - 1$  orthonormal bases that span the subspace perpendicular to direction vector  $d^p$  and  $w_\zeta = w_\eta = 0.001$  are two parameters.

**2.5 Modified Parent Centric Crossover (MPCX)**

A modified PCX creates an offspring individual as follow [1]:

$$y = \begin{cases} x^p + w_\zeta |d^p| + \sum_{i=1, i \neq p} w_\eta \bar{D} e^i & \text{if } u > (1 - \frac{1}{k}) \\ m + w_\zeta |d^p| + \sum_{i=1, i \neq p} w_\eta \bar{D} e^i & \text{otherwise} \end{cases} \quad (11)$$

- Compute the centre of mass

$$O = \frac{1}{3} \sum_{i=1}^3 x^i \quad (6)$$

Where  $m$  is the mean of the entire population,  $k = 1$ ,  $w_\zeta = 0.00$ ,  $w_\eta = \frac{w_\zeta}{2}$  and  $u \in [0,1]$  is distributed random number.

**Quadratic Interpolation Crossover**

Quadratic Interpolation Crossover (QIX) has been implemented in particle swarm optimization (PSO) [52]. It works as follows:

$$y = \frac{[x^{2j} - x^{2k}]g(x^1) + [x^{2i} - x^{2k}]g(x^j) + [x^{2i} - x^{2j}]g(x^{2k})}{[x^j - x^k]g(x^1) + [x^i - x^k]g(x^j) + [x^i - x^j]g(x^k)} \quad (13)$$

Where  $g(x^1), g(x^j), g(x^{ki})$  are the single values of the solutions,  $x^i, x^j, x^k$  respectively.

**3 Multi-objective Evolutionary Algorithm Based on Decomposition: MOEA/D**

MOEA/D [67] normally applies the conventional aggregation functions such as weighted sum function, Tchebycheff function for converting the problem of approximation of the Pareto front (PF) into a number of scalar optimization problems To date, MOEA/D have many enhanced versions and they are successfully applied to MOPs with many objective functions, discrete decision variables and complicated Pareto sets [32, 69, 38, 35, 41, 36, 34].

There are many existing aggregation functions such as weighted sum approach [44], Tchebycheff function [44], Normal-boundary intersection method [10], Normalized Normal Constraint Method [43] and many others [18]. MOEA/D can use any of the aforementioned aggregation function. In this paper, we have used the Tchebycheff Function as described in equation (12).

$$\text{Minimize } g^T = \max_{1 \leq j \leq m} \{ \lambda_j |f_j(x) - z_j^*| \} \quad (12)$$

where  $x \in \Omega, \lambda = (\lambda_1, \lambda_2, \dots, \lambda_m), \lambda_j \geq 0, j = 1, \dots, m, \sum_{j=1}^m \lambda_j = 1, z = (z_1^*, z_2^*, \dots, z_m^*)^T$  is the reference point (i.e.,  $z_j^* = \min\{f_j(x) | x \in \Omega\}$ ) for each  $j = 1, 2, \dots, m$ .

It is well known that, under mild conditions, for each Pareto optimal solution there exists a weight vector  $\lambda$  such that it is the optimal solutions to (12) and each optimal solution of (11) is a Pareto optimal solution of the problem (1). Let  $\lambda^1, \lambda^2, \lambda^3, \dots, \lambda^N$  be the set of  $N$  weight vectors. Correspondingly, we have  $N$  single objective optimization sub-problems (SOPs), where the  $i^{th}$  sub-problem is in equation (12) with  $\lambda = \lambda^i$ . If  $N$  is reasonably large and the weight vectors, Let  $\lambda^1, \lambda^2, \lambda^3, \dots, \lambda^N$  are properly selected, then Pareto optimal solution to the SOPs defined by the Tchebycheff function or any other decomposition function will provide a good set of final optimal to the given problem (1). The framework of MOEA/D is herewith explain in Algorithm 1 and Algorithm 2, respectively.

**Algorithm 1: Framework of MOEA/D-DRA****Input:**

1. MOP (1);
2. a stopping criterion;
3.  $N$ : the number of the sub-problems;
4. a uniform spread of  $N$  weight vectors:  $\lambda$
5.  $T$ : the number of the weight vectors in the neighbourhood of each weight vector.
6.  $\pi^i = 1$ , the utility function

**Output:**  $PS = \{x^1, x^2, x^3, \dots, x^N\}$  &  $PF = \{F(x^1), F(x^2), F(x^3), \dots, F(x^N)\}$

**Step 1 Initialization:**

**Step 1.1:** Compute the Euclidean distances between any two weight vectors to each weight vector.

For each  $i = 1, 2, \dots, N$ , set  $B(i) = \{i_1, i_2, \dots, i_T\}$ , where  $\lambda^{i_1}, \lambda^{i_2}, \dots, \lambda^{i_T}$  and  $T$  are closest weight vectors to  $\lambda^i$ .

**Step 1.2:** Generate an initial population of size  $N$ ,  $x^1, x^2, x^3, \dots, x^N$ , uniformly and randomly sampling in the search space of the MOP (1).

**Step 1.3:** Initialize  $z = (z_1, z_2, \dots, z_m)^T$  by setting  $z_i = \min\{f_i(x^1), f_i(x^2), f_i(x^3), \dots, f_i(x^N)\}$

**Step 1.4:** set  $gen = 0$  for all  $i = 1, 2, \dots, N$

If  $\text{mod}(gen, 50) == 0$ ;

Update  $\pi^i$  (i.e., for details See algorithm 2)

else  $\pi^i = 1$

end

**Step 2 Selection of Sub-problems for Search:** the indices of the sub-problems whose objectives are MOP individual objectives  $f$  are selected to form initial  $I$ . By using 10-tournament selection based  $\pi^i$  select other  $\left\lfloor \frac{N}{5} \right\rfloor - m$  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  $\text{rand}$  from  $(0, 1)$ . Then set

$$P = \begin{cases} B(i) & \text{if } \text{rand} < \delta \\ \{1, 2, \dots, N\} & \text{otherwise} \end{cases}$$

**Step 3.2: Reproduction:** Set  $r_1 = i, r_2, r_3 \in P$ . then apply crossover operator on three parent individuals,  $x^{r_1}, x^{r_2}, x^{r_3}$  to generate an offspring solution  $y := y^i$ .

**Step 3.3: Repair Method:** If the elements  $y := y^i$  are out of the boundary of  $\Omega$ , its value is reset to be a randomly selected value inside the boundary of problem.

**Step 3.4: Update of  $z$ :** For each  $j = 1, 2, \dots, m$ , if  $z_j > f_j(y)$ , then set  $z_j = f_j(y)$ .

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

- a) if  $c = n_r$  or  $P = \{\}$  go to **Step 4** else randomly pick an index  $j$  from  $P$ ;
- b) if  $g(y|\lambda^j, z) \leq g(x^j|\lambda^j, z)$ , then set  $x^j = y$ ;  $F(x^j) = F(y)$ ;  $c = c + 1$ ;
- c) Delete  $j$  from  $P$  and go to a).

**Step 4**  $gen = gen + 1$ ;

**Step 5** If stopping criteria is satisfied, then stop and give output

**Algorithm 2: Dynamic Resources Allocation to Sub-problems**

If gen is a multiple of 50, then compute  $\Delta^i$ , the relative decrease in the single objective values for each sub-problem  $i$  during the last 50 generations are updated according to the utility function as under:

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

end if

Go to Step 2 of the Algorithm 1

In Step 4 of the Algorithm 1, the relative decrease is defined as

$$\Delta^i = \frac{Diff}{old}, \text{ where } old = \text{old function value}$$

and  $Diff = \text{old function value} - \text{new function value}$ . If  $\Delta^i$  is smaller than 0.001, the value of  $\pi^i$  will be reduced.

In 10-tournament selection in Step 2 of the Algorithm 1, the index with the highest  $\pi^i$  values from 10-uniformly randomly selected indexes are chosen to enter I. We should do this selection  $\left\lceil \frac{N}{5} \right\rceil - m$  times.

In Step 3.2, different crossover operators followed by the polynomial mutation are used for creating an offspring population in the algorithm 1. The polynomial mutation [59] is formulated as follow:

$$y_k = \begin{cases} y_k + \sigma_k(u_k - l_k), & \text{with } p_m \\ y_k, & \text{with probability } 1 - p_m \end{cases} \quad (13)$$

$$\sigma_k = \begin{cases} (2 \times rand)^{\frac{1}{\eta-1}} - 1, & \text{if } rand < 0.5 \\ 1 - (2 - 2 \times rand)^{\frac{1}{\eta-1}}, & \text{otherwise} \end{cases} \quad (14)$$

Where  $l_k$  and  $u_k$  are the lower and upper bound of the  $k^{th}$  decision variable,  $\eta$  is the distribution index,  $p_m$  is the probability of mutation and  $rand \in [0,1]$  is uniformly random number.

**3 SIMULATION RESULTS AND COMPARATIVE ANALYSIS**

The selection of test suite functions with multi-modality, deception, isolation and particularly the location of true Pareto-optimal front are important for the comparative analysis of algorithms. The global optima of the benchmark functions formulated in [75, 12] which are either lying in the centre of the search range or along their respective bounds. Mostly, the functions in this test suite are comparatively very simple compared to the test instances which are recently designed in the special session of MOEAs competition of the 2009 IEEE Congress on Evolutionary Computation (CEC'09). The CEC'09 test instances [71].

have been covered an extension, stretching and rotation in their respective objective functions. We have chosen ten unconstrained problems from this test suite in our comparative analysis. All these problems should be treated as black-box problems, i.e., the mathematical formulations of these problems could not use in our suggested algorithms. The nature of the Pareto optimal fronts these problems are described in the Table 1. The reference data sets and their Matlab source code can be downloaded freely from the both links: <http://www.ntu.edu.sg/home/EPNSugan>.

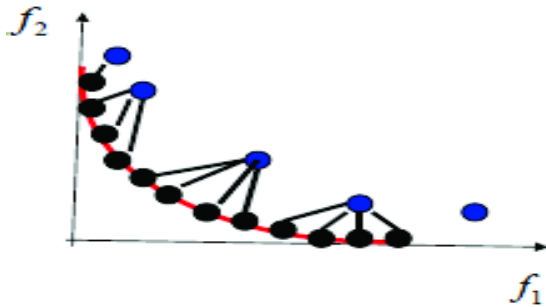
Table 1: Characteristics of the CEC'09 Test Instances.

CEC'09	Objectives	Search space range	Characteristics of PF
UF1	2	$[0, 1] \times [-1, 1]^{n-1}$	Concave
UF2	2	$[0, 1] \times [-1, 1]^{n-1}$	Concave
UF3	2	$[0, 1]^n$	Concave
UF4	2	$[0, 1] \times [-2, 2]^{n-1}$	Convex
UF5	2	$[0, 1] \times [-1, 1]^{n-1}$	21 point front
UF6	2	$[0, 1] \times [-1, 1]^{n-1}$	One isolated point and two disconnected parts
UF7	2	$[0, 1] \times [-1, 1]^{n-1}$	Continuous straight line
UF8	3	$[0, 1]^2 \times [-2, 2]^{n-2}$	Parabolic
UF9	3	$[0, 1]^2 \times [-2, 2]^{n-2}$	Planar
UF10	3	$[0, 1]^2 \times [-2, 2]^{n-2}$	Parabolic

**4.1 Performance Metrics**

In general, the quality of final approximated set of non-dominated solutions produced by specific MOEA are measure in terms of proximity and diversity improvements with help of different performance indicators. Proximity depicts the closeness of approximated non-dominated solutions against true Pareto front (PF), whereas diversity

measures that whether the final approximated set of multiple solutions are uniformly distributive and spread all along the whole true PF of problem at hand. In our experimental comparative analysis, we have chosen the Inverted generational distance (IGD) [77] as a performance indicator as shown in the figure 1 because it shown both convergence and diversity at the same time subject to availability of reference data set of MOPs test suites.



**Figure 1: Inverted generational distance (IGD).**

All black points (i.e., down) are Pareto solutions uniformly distributed along the RPF. All the blue solution points (i.e., above) are produced by an algorithm.

Let  $P^*$  be a set of uniformly distributed points along the true/real PF (RPF). Let  $A$  be an approximate set to the RPF, the average distance from  $P$  to  $A$  calculated as under:

$$D(A, P) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|} \quad (13)$$

Where  $d(v, A)$  is the minimum Euclidean distance between  $v \in P^*$  and the points belongs to approximated set  $A$ . If  $P^*$  is large enough, then it will be represent the PF very well,  $D(A, P)$  could measure both the diversity and convergence of  $A$  in a sense. In this study, we have used  $P^*=1000$  of uniformly spaced solutions to measure the Euclidean distance between approximate Pareto-optimal and true PF using equation (13). A smaller IGD-metric values implies a better convergence toward the Pareto-optimal front.

#### 4.1 Weight Vectors Selection

A set of  $N$  weight vectors are generated as per criteria given below.

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  
 $(1, 0, \dots, 0, 0), (0, 1, \dots, 0, 0), \dots, (0, 0, \dots, 0, 1)$ .
2. Find the weight vector in set  $W1$  with the largest distance to set  $W$ , add it to set  $W$  and delete it from set  $W1$ .

If the size of set  $W$  is  $N$ , stop and return set  $W$ . Otherwise, go to 2).

#### 4.2 Algorithmic Parameters Settings

- Operating system: Windows XP Professional.
- Programming language of the algorithms: Matlab

- CPU: Core 2 Quad 2.4 GHz. RAM: 4 GB DDR2 1066 MHz
- Execution: 30 independent runs with different random seeds.

- $N = 600$ : Population size for 2-objective test instances.
- $N = 1000$ : Population size for 3-objective test instances.
- $T = 0.1N$ : The neighbourhood size for each sub-problem.
- $p_m = \frac{1}{n}$ : is the probability mutation, where  $n = 30$ , the size of decision variables.
- $\eta = 20$ : the distribution index, used in polynomial mutation.
- Function evaluations: Each algorithm stops after 300,000;
- The maximum number of solutions for IGD-metric values measurements: 100 for 2-objective test instances and 150 for 3-objective test instances.
- $F = CR = 0.5(1+r)$ , where  $r$  is the random number.
- PCX's parameters:  $w_\zeta = w_\eta = 0.001$ .
- MPCX's parameters:  $w_\zeta = 0.002$  and  $w_\eta = \frac{w_\zeta}{2}$

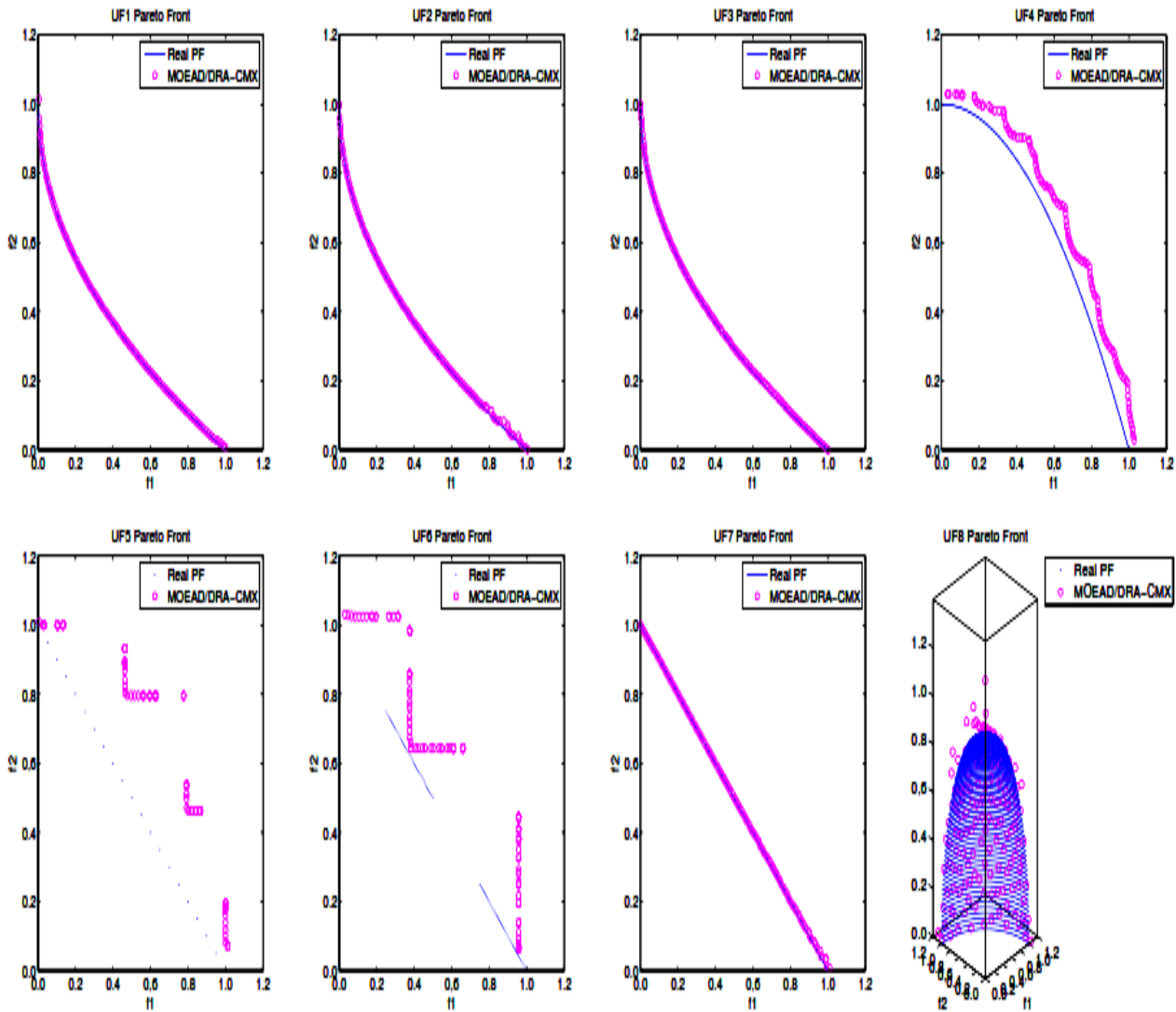
#### 4.3 Discussion of the Experimental Results

Several existing algorithms have been applied on the CEC'09 test MOPs [71]. In this paper, We have suggested six different versions of MOEA/D [69] by employing six crossover only abbreviated as 1) MOEA/D-CMX, 2) MOEA/D-SPX, 3) MOEA/D-ADE, 4) MOEA/D-PCX, 5) MOEA/D-MPCX 6) MOEA/D-QIX.

We have executed all our suggested algorithms 30 times independently to solve each CEC'09 test instance [71] with different random seeds. The IGD-metric values in 30 independent runs found by MOEA/D-CMX are listed in TABLE 3 for each CEC'09 test instance [71]. TABLE 2 provides the IGD-metric values obtained by MOEA/D-SPX for each test problem. The parameter settings regarding SPX are given in the last column of the TABLE 2. Tables 4 and TABLE 5 furnish the IGD-metrics in terms of smallest (minimum/best), median, mean, standard deviation (std) and largest (maximum/worst) with respect to MOEA/D-ADE and MOEA/D-PCX, respectively. TABLE 5 also provides parameter settings which are used in PCX implementation and formulation. The control parameters  $F$  and  $CR$  of DE are settled adaptively as like (i.e.,  $F = CR = 0.5(1 + \text{rand})$ ).

TABLE 6 and TABLE 7 summarise the IGD-metric values got by MOEA/D-MPCX and MOEA/D-QIX for each CEC'09 test instance [71]. From TABLE 7, one can easily conclude that the algorithmic performances MOEA/D [69] are quite deteriorated by integration QIX in its framework over almost all test problems. The same problems was happened with PCX and MPCX as well as compared to other used crossovers. Figures 2-3-4 demonstrate the best approximated PF brought forth by MOEA/D-CMX, MOEA/D-SPX, MOEA/D-ADE and MOEA/D-QIX for each CEC'09 test instance [71], respectively. A Figure 5 provides the best PF approximated by MOEA/D-QIX obtained for UF1-UF3 and UF7. MOEA/D-QIX has not properly tackled the rest problems, UF4-UF6 and UF8-UF10. Therefore, we did not include the figures of their IGD-metric values in this paper.





**Figure 2: Plots of the approximated Pareto Front display by MOEA/D-CMX in its best run among 30 independent runs over CEC'09 test instances.**

In Figures 6-7, we have plotted all 30 PFs altogether granted by MOEA/DCMX and MOEA/D-SPX. These figures clearly demonstrate the distribution ranges accomplished by MOEA/D-CMX and MOEA/D-SPX are much better than other candidates on almost all CEC'09 test instance. The figures of the UF5 and UF6 are included in this paper. Figure 8-9 depict the less average evolution in IGD- metric values in MOEA/DCMX,

MOEA/D-SPX and MOEA/D-ADE and MOEA/D-PCX over UF1UF10.

Based on the experimental results presented in this paper, we are confident by saying that CMX, SPX and ADE more better operators as compared to PCX, MPCX and QIX dealing with CEC'09 test instances [71].

Table 2: The IGD-metric values of the MOEA/D-SPX for UF1-UF10.

MOEA/D with SPX (MOEA/D-SPX)						
CEC'09	min	median	mean	std	max	parameters
UF1	0.0110	0.0620	0.0730	0.0480	0.1957	$2 + \sqrt{rand(1)}$
	0.0071	0.0739	0.0788	0.05579	0.2228	$1 + \sqrt{n - 26}$
	0.0287	0.1345	0.1356	0.0584	0.2422	$1 + \sqrt{n - 27}$
	0.005077	0.008203	0.008652	0.004079	0.026533	$\sqrt{n + 1}$
UF2	0.0053	0.0069	0.0069	0.0011	0.0096	$1 + \sqrt{(n - 27)}$
	0.0076	0.0175	0.0184	0.0092	0.0405	$1 + \sqrt{rand(1)}$
	0.004952	0.005596	0.005717	0.000487	0.007131	$\sqrt{n + 1}$
UF3	0.0148	0.0296	0.0356	0.0205	0.0864	$1 + \sqrt{(n - 27)}$
	0.005455	0.022374	0.030286	0.03823	0.091197	$\sqrt{n + 1}$
UF4	0.04572	0.05558	0.05568	0.0049	0.0666	$1 + \sqrt{rand(1)}$
	0.0502	0.0573	0.0573	0.0038	0.0657	$\sqrt{\sqrt{n + 1}}$
	0.049804	0.053895	0.054925	0.004188	0.069964	$\sqrt{n - 5}$
UF5	0.1938	0.3814	0.0679	0.0679	0.5326	$1 + \sqrt{rand(1)}$
	0.2186	0.3889	0.3963	0.0742	0.5704	$\sqrt{\sqrt{n + 1}}$
	0.202465	0.381695	0.389094	0.095218	0.583975	$\sqrt{n + 1}$
UF6	0.1953	0.4441	0.3896	0.0997	0.5962	$1 + \sqrt{rand(1)}$
	0.1807	0.4490	0.4101	0.1011	0.6212	$\sqrt{\sqrt{n + 1}}$
	0.153636	0.443824	0.373217	0.119556	0.586167	$\sqrt{n + 1}$
UF7	0.01663	0.3568	0.3546	0.1284	0.5519	$1 + \sqrt{rand(1)}$
	0.0109	0.3911	0.3259	0.1509	0.5488	$2 + \sqrt{rand(1)}$
	0.01107	0.3805	0.3384	0.1788	0.5684	$\sqrt{\sqrt{n + 1}}$
	0.005776	0.010898	0.062154	0.152973	0.533897	$\sqrt{n + 1}$
	0.006947	0.010454	0.121337	0.184387	0.523075	$\sqrt{n - 5}$
UF8	0.05552	0.0668	0.0724	0.0251	0.1951	$2 + \sqrt{rand(1)}$
	0.0557	0.1604	0.1448	0.0453	0.1901	$1 + \sqrt{n - 27}$
	0.0583	0.0786	0.0871	0.0215	0.1345	$\sqrt{\sqrt{n + 1}}$
UF9	0.4246	0.4967	0.4988	0.0450	0.5849	$1 + \sqrt{n - 27}$
	0.0404	0.1607	0.1502	0.03992	0.1845	$2 + \sqrt{n - 27}$
	0.0593	0.1727	0.1422	0.0507	0.1913	$\sqrt{\sqrt{n + 1}}$
UF10	0.4048	0.4794	0.4831	0.0472	0.5818	$2 + \sqrt{rand(1)}$
	0.4243	0.4668	0.4666	0.0270	0.5484	$\sqrt{\sqrt{n + 1}}$

Table 3: The IGD-metric values of the MOEA/D-CMX for UF1-UF10.

MOEA/D with CMX (MOEA/D-CMX)					
CEC'09	min/Best	median	mean	Std	max/worst
UF1	0.0041	0.0045	0.0053	0.0034	0.0228
UF2	0.0052	0.0068	0.0069	0.0011	0.0090
UF3	0.0044	0.0223	0.0326	0.0271	0.0913
UF4	0.0535	0.0606	0.0614	0.0047	0.0739
UF5	0.2224	0.3837	0.4254	0.1546	0.7124
UF6	0.1500	0.4436	0.3981	0.1538	0.7619
UF7	0.0048	0.0063	0.0085	0.0055	0.0317
UF8	0.0588	0.0685	0.0752	0.0166	0.1199
UF9	0.0491	0.1515	0.1280	0.0608	0.3256
UF10	0.4280	0.4780	0.4844	0.0351	0.5531

Table 4: The IGD-metric values of the MOEA/D-ADE for UF1-UF10.

MOEA/D-ADE with parameters $F = 0.5 * (1 + rand) = CR$ of DE.					
CEC'09	min	median	mean	std	max
UF1	0.004847	0.010675	0.01099	0.003732	0.019072
UF2	0.005804	0.008230	0.008306	0.00155	0.012082
UF3	0.006955	0.04685	0.044923	0.027546	0.093627
UF4	0.039672	0.048962	0.051182	0.007402	0.064670
UF5	0.127310	0.270832	0.281138	0.087448	0.480059
UF6	0.041672	0.172887	0.196197	0.155070	0.754536
UF7	0.004898	0.008514	0.009190	0.003596	0.023703
UF8	0.068072	0.088657	0.087818	0.006782	0.102548
UF9	0.043128	0.069057	0.087238	0.042770	0.177565
UF10	0.215141	0.344499	0.372537	0.107854	0.627219

Table 5: The IGD-metric values gathered by MOEA/D-PCX with parameters (i.e.  $w_\zeta = w_\eta = 0.001$ )

MOEA/D-PCX					
CEC'09	min/Best	median	mean	Std	max/worst
UF1	0.0312	0.0638	0.0749	0.0373	0.1886
UF2	0.0101	0.0180	0.0206	0.0107	0.0476
UF3	0.1148	0.2066	0.2042	0.0389	0.2693
UF4	0.0496	0.0560	0.0581	0.0075	0.0785
UF5	0.1709	0.3797	0.3787	0.01313	0.6431
UF6	0.0775	0.2046	0.2848	0.2056	0.8327
	0.1609	0.3736	0.4118	0.2073	0.8313
UF7	0.0129	0.1146	0.1909	0.1950	0.6199
	0.0132	0.0247	0.1431	0.1717	0.5134
UF8	0.891378	1.016922	1.009308	0.066704	1.117835
UF9	0.0968	0.1876	0.1702	0.0362	0.2193
UF10	0.1957	0.4029	0.3820	0.1478	0.7615

Table 6: The IGD-metric values found by MOEA/D-MPCX with parameters  $w_\zeta = 0.002, w_\eta = w_\zeta/2$  for UF1-UF10.

MOEA/D-MPCX.					
CEC'09	min/Best	median	mean	Std	max/worst
UF1	0.038334	0.081224	0.091323	0.045432	0.260354
UF2	0.014111	0.023313	0.033506	0.020654	0.095736
UF3	0.204872	0.227166	0.232482	0.021263	0.280650
UF4	0.053795	0.063551	0.122970	0.301495	1.716160
UF5	0.182220	0.442678	0.437333	0.096366	0.643224
UF6	0.181398	0.446682	0.417977	0.125357	0.780010
UF7	0.013394	0.227361	0.225019	0.198278	0.557805
UF8	0.450662	0.636792	0.663861	0.111034	0.946083
UF9	0.131885	0.236197	0.2288897	0.039954	0.2922767
UF10	0.205368	0.453790	0.442067	0.055197	0.503839

Table 7: The IGD-metric values of the MOEA/D-QIX for UF1-UF10.

MOEA/D with QIX (MOEA/D-QIX)					
CEC'09	min/Best	median	mean	Std	max/worst
UF1	0.0305	0.0837	0.0885	0.0406	0.1809
UF2	0.0319	0.0421	0.0433	0.0067	0.0580
UF3	0.0943	0.1818	0.1822	0.0426	0.2610
UF7	0.0125	0.0483	0.2200	0.2604	0.6014
UF8	1.1413	1.3027	1.2868	0.0697	1.4228
UF10	0.6433	1.0829	1.1249	0.2695	1.7226



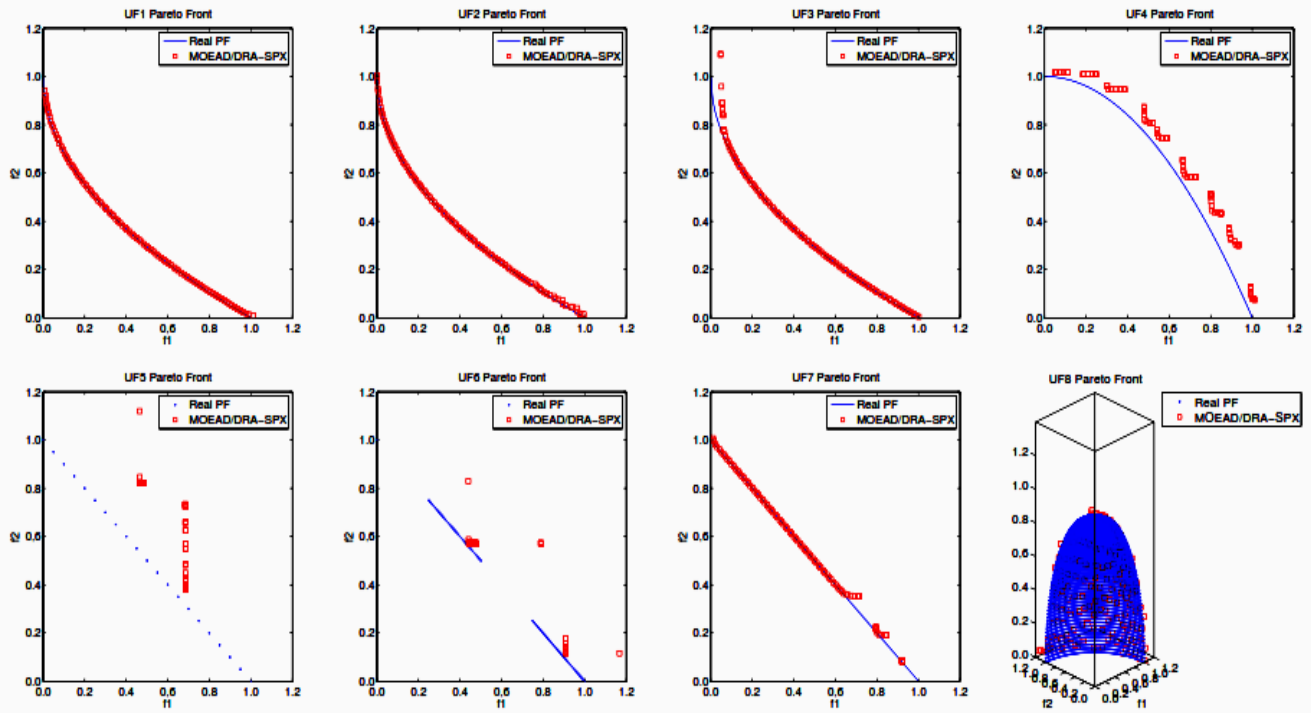


Figure 3: Plots of the approximated Pareto Front display by MOEA/D-SPX in its best run among 30 independent runs over CEC'09 test instances.

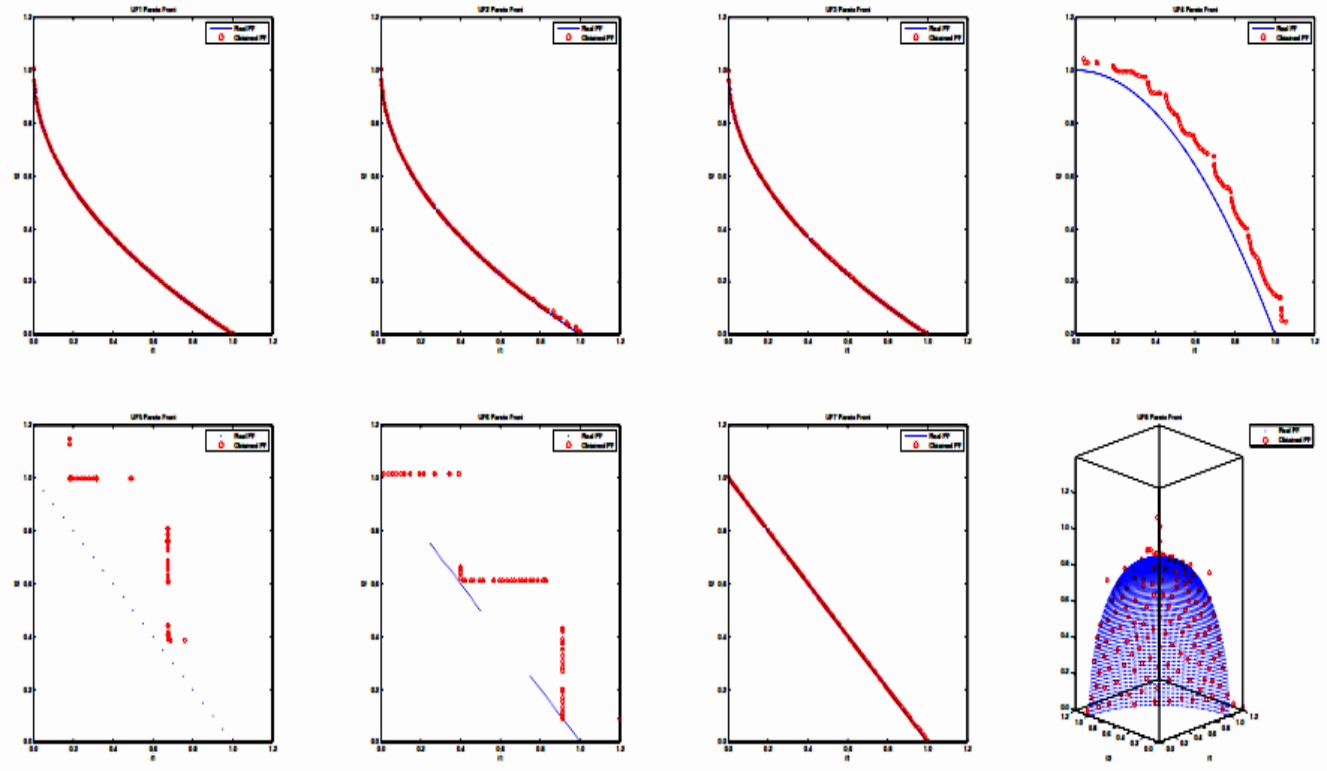


Figure 4: Plots of the approximated Pareto Front display by MOEA/D-ADE in its best run among 30 independent runs over CEC'09 test instances.

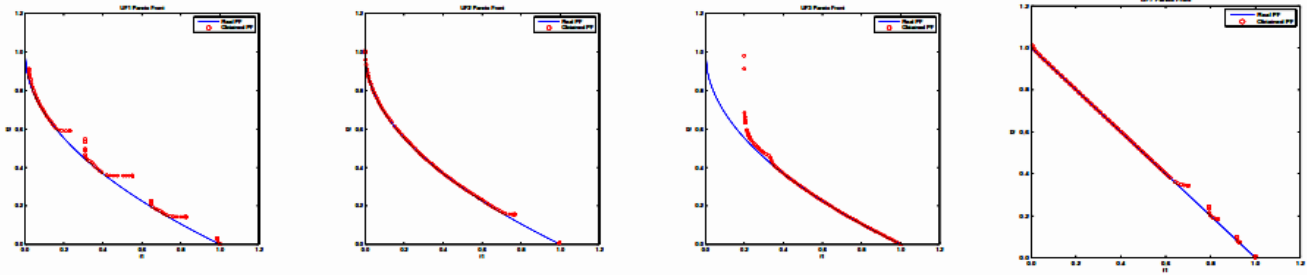


Figure 5: Plots of the approximated Pareto Front display by MOEA/D-QIX in its best run among 30 independent runs over UF1-UF3 and UF7 test instances.

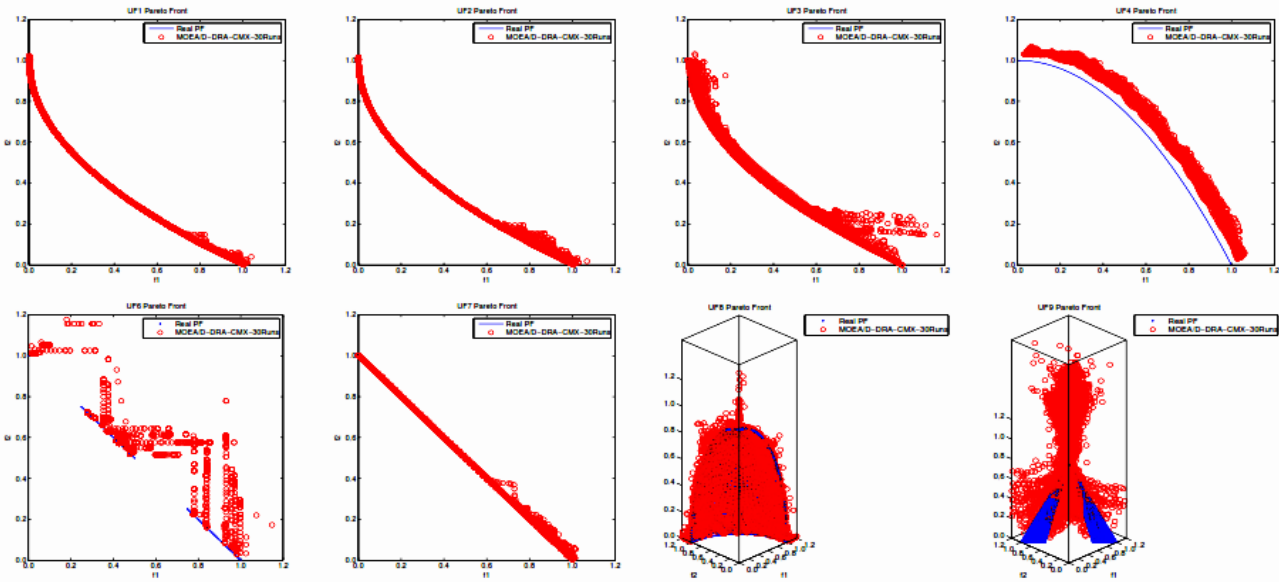


Figure 6: Plots of the 30 approximated Pareto Fronts of MOEA/D-CMX altogether for CEC'09 test instances.

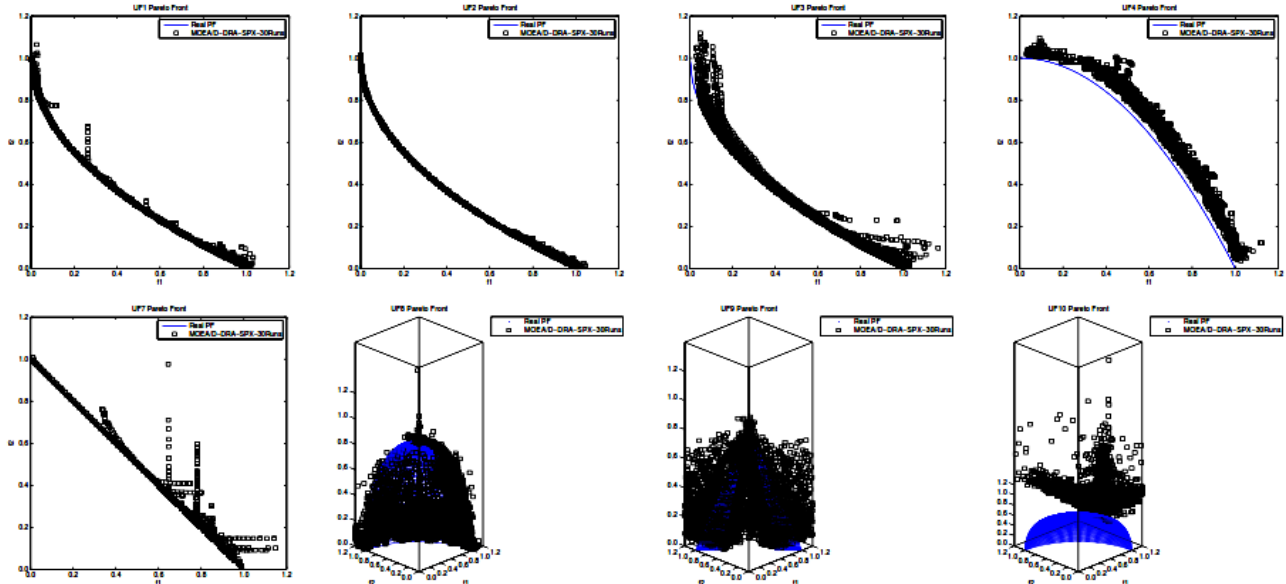


Figure 7: Plots of the 30 approximated Pareto Fronts of MOEA/D-SPX altogether for CEC'09 test instances.

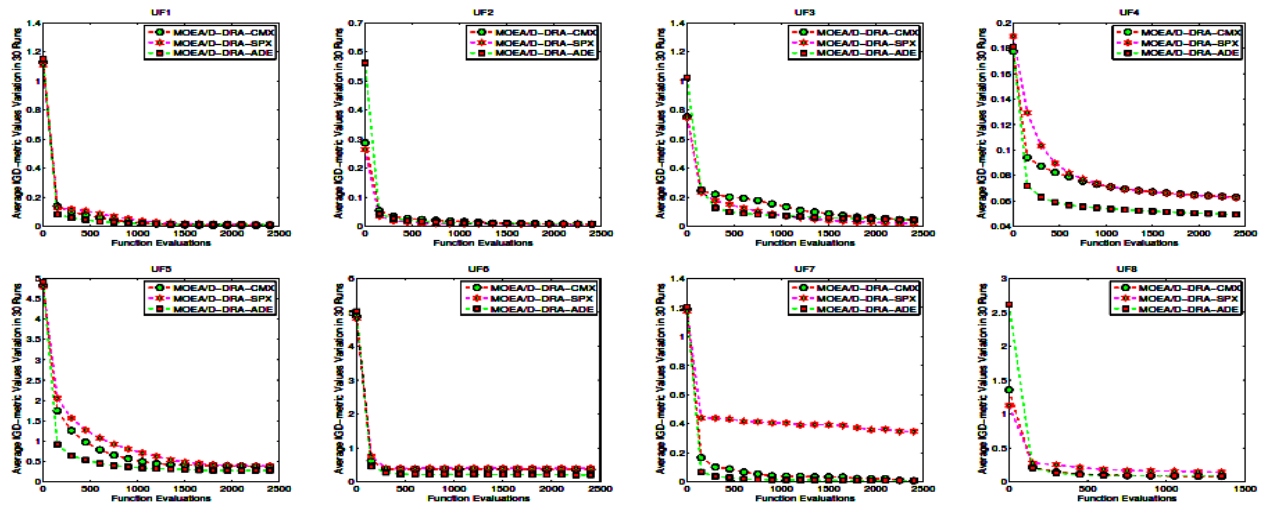


Figure 8: Plots of the average evolution of the IGD-values of the MOEA/DCMX, MOEA/D-SPX and MOEA/D-ADE over CEC'09 test instances.

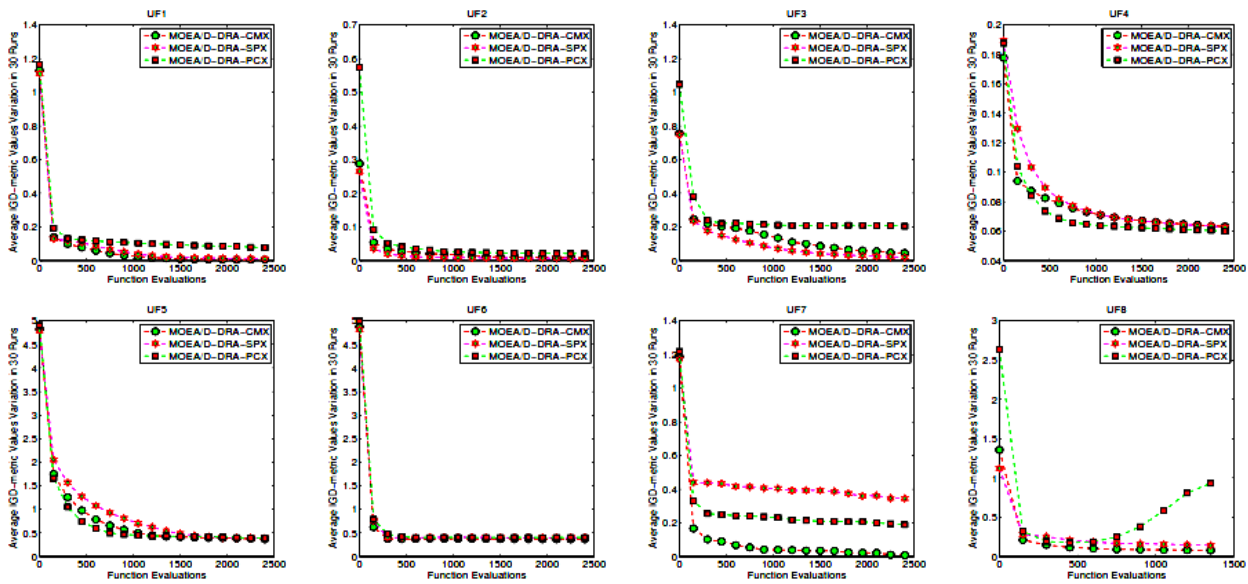


Figure 9: Plots of the average evolution of the IGD-values of the MOEA/DCMX, MOEA/D-SPX and MOEA/D-PCX over CEC'09 test instances.

### 5 CONCLUSION

Crossover and mutation are main intrinsic search operators in the evolutionary process of population based evolutionary algorithms. Crossover operators exploit valuable information of the existing population and influence the search process of in next generation of EAs. More importantly, it directs the search process toward the best search space of underattack problems. Different EAs suite different search operators and different optimization and search problems. In this paper, we mainly experimentally study individual behaviours of different crossover operators in multi-objective optimization context. For this purposes, we have chosen multi-objective evolutionary algorithm based on decomposition [69] as master algorithm. The used crossovers are inlindde the simplex crossover operator (SPX), centre of mass crossover operator (CMX), adaptive differential evolution (ADE), Parent centric crossover (PCX), modified parent centric

crossover (MPCX) and quadratic interpolation crossover (QIX). After examine the individual performance of each crossover in MOEA/D [69] framework by using the CEC'09 test instances, we have observed that QIX, PCX and MPCX are not much appropriate crossover as compared to CMX, SPX and ADE in MOEA/D framework for dealing with most CEC'09 test Instances.

In future, we will be systematically analysis the impact some other crossovers with self-adaptive strategies to tackle the real-world problems. Moreover, we will also examine the impact of different local search optimizers in combination with different multiple crossover in the framework of MOEA/D. In addition, we also intend to establish a close relationship among mutation and crossover operators for the better achievement of multi-objective optimization goals.

## ACKNOWLEDGMENT

The first author is much thankful to Professor Qingfu Zhang, School of Computer Science & Electronic Engineering, University Of Essex, Wivenhoe Park, CO4 3SQ, Colchester, UK for his valuable comments to improve the quality of this paper.

## REFERENCES

- [1] S. T. Ankur Sinha and K. Deb, "A Population-Based, Steady-State Procedure for Real-Parameter Optimization," Kanpur Genetic Algorithms Laboratory (KanGAL), India, Report 2005004.
- [2] A. Arias-Montano, C. A. Coello Coello, and E. Mezura Montes, "Multiobjective Evolutionary Algorithms in Aeronautical and Aerospace Engineering," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 5, pp. 662–694, 2012.
- [3] J. Bader and E. Zitzler, "HypE: An Algorithm for Fast Hypervolume Based Many Objective Optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.
- [4] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective Selection based on Dominated Hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.
- [5] 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.
- [6] T. Caric and H. Gold, *Vehicle Routing Problem*. In-Teh, Vienna, Austria, 2008.
- [7] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.
- [8] C. A. Coello Coello and R. L. Becerra, "Evolutionary Multiobjective Optimization in Materials Science and Engineering," *Materials and Manufacturing Processes*, vol. 24, no.2, pp. 119–129, 2009
- [9] Y. Collette and P. Siarry, *Multiobjective Optimization: Principles and Case Studies*. Springer Science & Business Media, 2003.
- [10] I. Das and J. E. Dennis, "Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multi-criteria Optimization Problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.
- [11] D. Davendra, *Traveling Salesman Problem, Theory and Applications*. INTECH, December 2010.
- [12] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Multi-objective Optimization Test Problems," In *Congress on Evolutionary Computation (CEC2002)*, Piscataway, New Jersey: IEEE Service Centre, vol. 1, pp. 825–830, May 2002.
- [13] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, 2nd ed., S. Ross and R. Weber, Eds. John Wiley and Sons Ltd, 2002. Multiple Crossovers in MOEA/D 23
- [14] K. Deb, A. Anand, and D. Joshi, "A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization," *Evolutionary Computation*, vol. 10, pp. 371–395, 2002.
- [15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multi-objective Genetic Algorithm: NSGA-II," *IEEE Transaction on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [16] D. Thierens, "An Adaptive Pursuit Strategy for Allocating Operator Probabilities," in *Proceedings of the 2005 Genetic and Evolutionary Computation Conference (GECCO'05)*, New York, NY, USA. ACM.2218, 2005, pp. 1539-1546.
- [17] F. Y. Edgeworth, *Mathematical Psychics*. P. Keagan, London, England, 1881.
- [18] M. Ehrgott, *Multi-criteria Optimization*, 2nd ed. Springer, Berlin, 2005.
- [19] L. J. Eshelman and J. D. Schaffer, "Real-Coded Genetic Algorithms and Interval-Schemata," in *Foundations of Genetic Algorithms*, 1993, pp. 187–202.
- [20] E. Zitzler and L. Thiele, "Multi-objective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Evolutionary Algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [21] C. Fonseca and P. Fleming, "Genetic Algorithms for Multi-objective Optimization: Formulation, Discussion and Generalization," in *Proceedings of 5th International Conference on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, pp. 416–423.
- [22] C. Fonseca and P. Fleming "An Overview of Evolutionary Algorithm in Multi-Objective Optimization," *Evolutionary Computation*, vol. 3, no. 1, pp. 1–16, 1995.
- [23] J. Horn, N. Nafpliotis, and D. E. Goldberg., "A Niche Pareto Genetic Algorithm for Multi-objective Optimization," in *Proceedings of the 1<sup>st</sup> IEEE Conference on Evolutionary Computation, CEC'94*, 1994.
- [24] H. Ishibuchi and T. Murata, "Multi-Objective 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.
- [25] H. Ishibuchi and T. Murata., "Multi-Objective Genetic Local Search Algorithm." in *Proceedings of the Third IEEE International Conference on Evolutionary Computation*, I. T. Fukuda and T. Furuhashi, Eds., Nagoya, Japan, 1996, pp. 119–124.
- [26] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of Different Scalarizing Functions in MOEA/D," in *Genetic and Evolutionary Computation Conference, GECCO 2010, Proceedings*, Portland, Oregon, USA, July 7-11, 2010, pp. 519–526.
- [27] A. Jaszkiewicz, "On the Computational Efficiency of Multiple Objective

- Metaheuristics. The Knapsack Problem Case Study,” *European Journal of Operational Research*, vol. 158, no. 2, pp. 418–433, 2004.
- [28] K. Deb and R. Agrawal, “Simulated Binary Crossover for Continuous Search Space,” *Complex System*, vol. 9, pp. 115–148, 1995
- [29] L. Ke, Q. Zhang, and R. Batti “MOEA/D-ACO: a Multi-objective Evolutionary Algorithm using Decomposition and Ant Colony Optimization,” *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1845–1859, 2013.
- [30] W. Khan and Q. Zhang, “MOEA/D-DRA with Two Crossover Operators,” in *Proceedings of the UK Workshop on Computational Intelligence (UKCI 2010)*, 8<sup>th</sup>–10<sup>th</sup> September 2010, pp. 1–6.
- [31] C. Lcken, B. Barn, and C. Brizuela, “A Survey on Multi-Objective Evolutionary Algorithms for Many-objective Problems,” *Computational Optimization and Applications*, vol. 58, no. 3, pp. 707–756, 2014.
- [32] H. Li and Q. Zhang, “Multi-objective Optimization Problems With Complicated Pareto Sets: MOEA/D and NSGA-II,” *IEEE Transaction on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, April 2009.
- [33] X. Liao, Q. Li, X. Yang, W. Zhang, and W. Li, “Multi-objective Optimization for Crash Safety Design of Vehicles using Stepwise Regression Model,” *Structural and Multidisciplinary Optimization*, vol. 35, no. 6, pp. 561–569, 2008.
- [34] W. K. Mashwani, “A Multimethod Search Approach Based on Adaptive Generations level,” in *Seventh International Conference on Natural Computation (ICNC’11)*, Shanghai, China, 26-28 July, 2011, pp. 23–27.
- [35] Wali Khan Mashwani, “Hybrid Multiobjective Evolutionary Algorithms: A Survey of state-of-the-art,” *International Journal of Computer Science Issues*, vol. 8, no. 6, pp. 374–392, 2011.
- [36] Wali Khan Mashwani, “MOEA/D with DE and PSO: MOEA/D-DE+PSO,” in *Proceeding of 31<sup>th</sup> SGA International Conference on Innovative Techniques and Applications of Artificial Intelligence*, Cambridge, UK, December, 2011, pp. 217–221.
- [37] Wali Khan Mashwani, “Comparison of Evolutionary Algorithm over Multi-objective Optimization Problems,” in *Proceeding of International Conference on Modelling and Simulation (ICOMS)*, Air University Islamabad, Pakistan, 2013.
- [38] Wali Khan Mashwani, “Comprehensive survey of the hybrid evolutionary algorithms,” *International Journal of Applied Evolutionary Computation (IJAEC)*, vol. 4, no. 2, pp. 1–19, July 2013.
- [39] Wali Khan Mashwani, “Enhanced Versions of Differential Evolution: state-of-the-art Survey,” *International Journal of Computing Science and Mathematics*, vol. 5, no. 2, pp. 107–126, 2014.
- [40] W. K. Mashwani and A. Salhi, “A Decomposition-based Hybrid Multi-objective Evolutionary Algorithm with Dynamic Resource Allocation,” *Applied Soft Computing*, vol. 12, no. 9, pp. 2765–2780, 2012.
- [41] Wali Khan Mashwani, “Multi-objective Memetic Algorithm Based on Decomposition,” *Applied Soft Computing*, vol. 21, no. 1, pp. 221–243, 2014.
- [42] Y. Mei, K. Tang, and X. Yao, “Decomposition-Based Memetic Algorithm for Multi-objective Capacitated Arc Routing Problem,” *IEEE Trans. Evolutionary Computation*, vol. 15, no. 2, pp. 151–165, 2011.
- [43] A. Messac, A. Ismail-Yahaya, and C. A. Mattson, “The Normalized Normal Constraint Method for Generating the Pareto Frontier,” *Structural and Multidisciplinary Optimization*, vol. 25, no. 2, pp. 86–98, July 2003.
- [44] K. M. Miettinen, *Nonlinear Multi-objective Optimization*, ser. Kluwer’s International Series. Norwell, MA: Academic Publishers Kluwer, 1999.
- [45] M.M.Raghuwanshi, P.M.Singru, U.Kale, and O.G.Kakde, “Simulated Binary Crossover with Lognormal Distribution,” *Complexity International*, vol. 12, pp. 1–10, 2005.
- [46] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba, “MOCeL: A Cellular Genetic Algorithm for Multi-objective Optimization,” *International Journal of Intelligent Systems*, pp. 25–36, 2007.
- [47] N.Srinivas and K. Deb, “A Multi-objective Optimization using Non-dominated Sorting in Genetic Algorithms,” *Journal Evolutionary Computing*, vol. 2, no. 3, pp. 221–248, 1994.
- [48] I. Ono and S. Kobayashi, “A Real-Coded Genetic Algorithm for Function Optimization Using the Unimodal Normal Distribution Crossover,” in *7<sup>th</sup> International Conference on Genetic Algorithms*, 1997, pp. 246–253.
- [49] I. Ono, H. Kita, and S. Kobayashi, *A Real-coded Genetic Algorithm using the Unimodal Normal Distribution Crossover*. New York, NY, USA: Springer-Verlag New York, Inc., 2003, pp. 213–237.
- [50] B. B. Onwubolu, Godfrey C., *New Optimization Techniques in Engineering*. Springer, 2004, vol. 141.
- [51] P. Palmers, T. McConnaghy, M. Steyaert, and G. Gielen, “Massively Multi-topology Sizing of analog Integrated Circuits,” in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE’09. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2009, pp. 706–711.
- [52] M. Pant, R. Thangaraj, and Ajith Abraham, “A New PSO Algorithm with Crossover Operator for Global Optimization Problems,” *E.Corchado et al. (Eds.): Innovations in Hybrid Intelligent Systems*, vol. ASC 44, pp. 215–222, 2007.
- [53] R. Peltokangas and A. Sorsa, “Real Coded Genetic Algorithm and Nonlinear Parameter Identification,” *Control Engineering Laboratory University of OULU*, P.O. BOX 4300 FIN-90014, Report A No 34 34, April 2008.
- [54] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, “A Survey on Multi-objective Evolutionary Algorithms for the Solution of the Portfolio Optimization Problem and other Finance and Economics Applications,” *IEEE Transaction on Evolutionary Computation*, vol. 17, no. 3, pp. 321–344, 2013.



- [55] Y. Qi, X. Ma, L. J. Fang Liu, J. Sun, and J. Wu, "MOEA/D with Adaptive Weight Adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, May 2014.
- [56] R. Storn and K. V. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," ICSI, Technical Report TR-95-012, 1995.
- [57] R. Storn and K. V. Price, "Differential Evolution - a Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *J. Global Opt.*, vol. 11, no. 4, pp. 341–359, December 1997.
- [58] W. M. Spears, "Adapting Crossover in Evolutionary Algorithms," in *Proceedings of the Fourth Annual Conference on Evolutionary Programming*. MIT Press, 1995, pp. 367–384.
- [59] M. G. C. Tapia and C. A. C. Coello, "Applications of Multi-Objective Evolutionary Algorithms in Economics and Finance: A Survey." *IEEE Congress on Evolutionary Computation*, vol. 7, pp. 532–539, 2007.
- [60] S. Tsutsui, "Multi-parent Recombination in Genetic Algorithms with Search Space Boundary Extension by Mirroring," in *PPSN V: Proceedings of the 5<sup>th</sup> International Conference on Parallel Problem Solving from Nature*. London, UK: Springer-Verlage, 1998, pp. 428–437.
- [61] S. Tsutsui and A. Ghosh, "A Study on the Effect of Multi-parent Recombination in Real Coded Genetic Algorithms," *Proc. of the IEEE ICEC*, pp. 828–833, 1998.
- [62] S. Tsutsui, M. Yamamura, and T. Higuchi, "Multi-Parent Recombination with Simplex Crossover in Real coded Genetic Algorithms," in *Proceeding of GECCO-99*, 1999, pp. 657–374.
- [63] A. Tuson and P. Ross, "Adapting Operator Settings in Genetic Algorithms," *Evolutionary Computation*, vol. 6, pp. 161–184, 1998.
- [64] B. Y. Wu and K.-M. Chao, *Spanning Trees and Optimization Problems*. CRC PRESS Boca Raton London New York Washington, D.C., December 2003.
- [65] H.-S. Yoon and B.-R. Moon, "An Empirical Study on the Synergy of Multiple Crossover Operators," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 212–223, 2002.
- [66] T. J. Yuen and R. Ramli, "Comparison of Computational Efficiency of MOEA/D and NSGA-II for Passive Vehicle Suspension Optimization," in *Proceeding of 24th EUROPEAN Conference on Modelling and Simulation (ECMS'10)*. Kuala Lumpur, Malaysia: University of Nottingham, Malaysia Campus, June 1st–4th 2010, pp. 219–215.
- [67] Q. Zhang and H. Li, "MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition," *IEEE Transaction on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, December 2007.
- [68] Q. Zhang, H. Li, D. Maringer, and E. P. K. Tsang, "MOEA/D with NBI style Tchebycheff Approach for Portfolio Management," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'10)* Barcelona, 2010, pp. 1–8.
- [69] Q. Zhang, W. Liu, and H. Li, "The Performance of a New Version of MOEA/D on CEC'09 Unconstrained MOP Test Instances," *IEEE Congress on Evolutionary Computation (IEEE CEC 2009)*, Trondheim, Norway, pp. 203–208, May, 18–21 2009.
- [70] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive Multi-objective Optimization by MOEA/D with Gaussian Process Model," *Transaction. on Evolutionary Computing*, vol. 14, pp. 456–474, June 2010.
- [71] Q. Zhang, A. Zhou, S. Zhaoy, P. N. Suganthany, W. Liu, and S. Tiwariz, "Multi-objective Optimization Test Instances for the CEC 2009 Special Session and Competition," Technical Report CES-487, 2009.
- [72] S.-Z. Zhao, P. N. Suganthan and Q. Zhang, "Decomposition-Based Multi-objective Evolutionary Algorithm with an Ensemble of Neighbourhood Sizes," *Evolutionary Computation*, *IEEE Transactions on*, vol. 16, no. 3, pp. 442–446, 2012.
- [73] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang., "Multi-objective Evolutionary Algorithms: A survey of the state-of-the-art," *Swarm and Evolutionary Computation*, vol. 1, pp. 32–49, 16 March 2011, online Published.
- [74] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," *Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, TIK Report 103*, 2001.
- [75] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multi-objective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [76] E. Zitzler and S. Knzli, "Indicator-based Selection in Multi-objective Search," in *in Proceeding of 8<sup>th</sup> International Conference on Parallel Problem Solving from Nature (PPSN VIII)*. Springer, 2004, pp. 832–842.
- [77] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance Assessment of Multi-objective Optimizers: An Analysis and Review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, April 2003, pp: 117-132
- [78] Wali Khan Mashwani and Abdel Salhi, "Multi-objective Evolutionary Algorithm Based on Multimethod with Dynamic Resources Allocation", Accepted for publication in *upccmng Issue of Applied Soft Computing Journal*, 2016.