IMPACT ANALYSIS OF Crossovers IN MULTIOBJECTIVE
EVOLUTIONARY ALGORITHM

Wali Khan Mashwani¹, Abdel Salhi², Muhammad Asif Jan³, Rasheeda Adeeb Khanum⁴ and M.Sulaiman⁵
¹,³Department of Mathematics, Kohat University of Science & Technology (KUST), Pakistan.
³Jinnah College for Women, University of Peshawar, KPK, Pakistan.
⁴Department of Mathematical Sciences, University of Essex, Colchester, UK
⁵Department 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 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 experiments, 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 practical 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:

\[
\text{Minimize } F(x) = (f_1(x) + f_2(x) + \cdots + f_m(x))^T
\]

Subjected to \( x \in \Omega \)

Where \( \Omega \) is the decision variable space, \( x = (x_1, x_2, \ldots, x_n)^T \) 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 \( u = (u_1, u_2, \ldots, u_n) \in \Omega \) is said to be Pareto optimal if there does not exist another solution \( v = (v_1, v_2, \ldots, v_n) \in \Omega \) such that \( f_j(u) \leq f_j(v) \) for all \( j = 1, 2, \ldots, m \) 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.

Nov.-Dec
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 multithread 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 neighborhood 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 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 [30].

Different crossover operators 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 affected 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 researchers 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:

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

Where $O = \sum_{k=1}^{3} x^k$ 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]:

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

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

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 mass as:

\[ v^j = 2O - x^j \quad (7) \]

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 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. A simple mutation strategies of DE is formulated as fallow:

\[ y^- = x^i + F(x^2 - x^3) \quad (2) \]

Where \( x^2 \) and \( x^3 \) are two random solutions different from \( x^i \) 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

\[ y = \begin{cases} y^- & \text{if } u_r \leq CR \\ x^i & \text{otherwise} \end{cases} \quad (3) \]

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:

\[ y = x^p + \omega_{c} |d|^p + \sum_{i=1}^{l} \omega_{i} D e^i \quad (4) \]

\[ d^p = y^p - \bar{g} \quad (5) \]

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 \( d^p \) perpendicular distances of \( \mu \) -1 parents, \( e^i \) are \( \mu \)-1 orthonormal bases that span the subspace perpendicular to direction vector \( d^p \) and \( \omega_{c} = \omega_{\eta} = 0.001 \) are two parameters.

2.5 Modified Parent Centric Crossover (MPCX)

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

\[ y = \begin{cases} \left(x^p + \omega_{c} |d|^p \right) + \sum_{i=1}^{l} \omega_{i} D e^i & \text{if } u > \left(1 - \frac{1}{k} \right) \\ m + \omega_{c} |d|^p + \sum_{i=1}^{l} \omega_{\eta} D e^i & \text{otherwise} \end{cases} \quad (11) \]

- 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, \( \omega_{c} = 0.00 \), \( \omega_{\eta} = \frac{\omega_{c}}{2} \) and \( u \in [0, 1] \) is distributed random number.

2.6 Quadratic Interpolation Crossover

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

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

Where \( g(x^1), g(x^2), g(x^3) \) are the single values of the solutions, \( x^1, x^i, x^k \) 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).

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

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

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 (11) is a Pareto optimal solution of the problem (1). Let \( \lambda^1, \lambda^2, \lambda^3, ..., \lambda^N \) 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 \( \lambda^1, \lambda^2, \lambda^3, ..., \lambda^N \) 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 neighborhood of each weight vector.
6. \( \pi^{i} = 1 \), the utility function

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

Step 1 Initialization:

Step 1.1: Compute the Euclidean distances between any two weight vectors to each weight vector.
For each \( i = 1, 2, \ldots, N \), set \( B(i) = \{ i_{1}, i_{2}, \ldots, i_{T} \} \), where \( \lambda^{i_{1}}, \lambda^{i_{2}}, \ldots, \lambda^{i_{T}} \) and \( T \) are closest weight vectors to \( \lambda^{i} \).

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

Step 1.3: Initialize \( z = (z_{1}, z_{2}, \ldots, z_{m})^{T} \) by setting \( z_{i} = \min[f_{i}(x^{1}), f_{i}(x^{2}), f_{i}(x^{3}), \ldots, f_{i}(x^{N})] \)

Step 1.4: set \( \pi^{i} = 0 \) for all \( i = 1, 2, \ldots, N \), if \( \text{mod}(\text{gen}, 50) == 0 \); Update \( \pi^{i} \); (i.e., for details See algorithm 2)

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 \( \left[ \frac{N}{2} \right] - m \) Indices and add them to \( I \).

Step 3 for each \( \epsilon \in I \), do:

Step 3.1: Selection of Mating/Update Range: Uniformly randomly generate a number \( \text{rand} \) from \( (0, 1) \). Then set
\[
P = \begin{cases} 
B(i) & \text{if } \text{rand} < \delta \\
\{1, 2, \ldots, N\} & \text{otherwise}
\end{cases}
\]

Step 3.2: Reproduction: Set \( r_{1}, r_{2}, r_{3} \in P \). 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 \( j = 1, 2, \ldots, m \), 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 \( c = n_{e} \) or \( P = \{ j \} \) go to Step 4 else randomly pick an index \( j \) from \( P \);

b) If \( g(y) < g(x^{j}) \), set \( x^{j} = y \); \( F(x^{j}) = F(y) \); \( c = c + 1 \);

c) Delete \( j \) from \( P \) and go to a).

Step 4 \( \text{gen} = \text{gen} + 1 \);

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:

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

end if

Go to Step 2 of the Algorithm 1

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

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

and Diff = old function value – 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 $\left\lfloor \frac{N}{5} \right\rfloor - m$ 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 [39] is formulated as follow:

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

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

Where $l_k$ and $u_k$ are the lower and lower 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>
<thead>
<tr>
<th>CEC'09</th>
<th>Objectives</th>
<th>Search space range</th>
<th>Characteristics of PF</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF1</td>
<td>2</td>
<td>$[0,1] \times [-1,1]^{n-1}$</td>
<td>Concave</td>
</tr>
<tr>
<td>UF2</td>
<td>2</td>
<td>$[0,1] \times [-1,1]^{n-1}$</td>
<td>Concave</td>
</tr>
<tr>
<td>UF3</td>
<td>2</td>
<td>$[0,1]^{n}$</td>
<td>Concave</td>
</tr>
<tr>
<td>UF4</td>
<td>2</td>
<td>$[0,1] \times [-2,2]^{n-1}$</td>
<td>Convex</td>
</tr>
<tr>
<td>UF5</td>
<td>2</td>
<td>$[0,1] \times [-1,1]^{n-1}$</td>
<td>21 point front</td>
</tr>
<tr>
<td>UF6</td>
<td>2</td>
<td>$[0,1] \times [-1,1]^{n-1}$</td>
<td>One isolated point and two disconnected parts</td>
</tr>
<tr>
<td>UF7</td>
<td>2</td>
<td>$[0,1] \times [-1,1]^{n-1}$</td>
<td>Continuous straight line</td>
</tr>
<tr>
<td>UF8</td>
<td>3</td>
<td>$[0,1] \times [-2,2]^{n-1}$</td>
<td>Parabolic</td>
</tr>
<tr>
<td>UF9</td>
<td>3</td>
<td>$[0,1] \times [-2,2]^{n-1}$</td>
<td>Planar</td>
</tr>
<tr>
<td>UF10</td>
<td>3</td>
<td>$[0,1] \times [-2,2]^{n-1}$</td>
<td>Parabolic</td>
</tr>
</tbody>
</table>

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 wether 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).](image)

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:

$$D(A, P) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|}$$  \hspace{1cm} (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 $W_1$. Set $W$ is initialized as the set containing all the weight vectors $(1, 0, \ldots, 0, 0), (0, 1, \ldots, 0, 0), (0, 0, \ldots, 0, 1)$.
2. Find the weight vector in set $W_1$ with the largest distance to set $W$, add it to set $W$ and delete it from set $W_1$.
   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: $\omega_c = \omega_q = 0.001$.
- MPCX’s parameters: $\omega_c = 0.002$ and $\omega_q = \frac{\omega_c}{2}$.

### 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 crossovers 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., $F = CR = 0.5(1 + r)$).

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.
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 instances. 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.

<table>
<thead>
<tr>
<th>CEC'09</th>
<th>min</th>
<th>median</th>
<th>mean</th>
<th>std</th>
<th>max</th>
<th>parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF1</td>
<td>0.0110</td>
<td>0.0620</td>
<td>0.0730</td>
<td>0.0480</td>
<td>0.1957</td>
<td>$2 \cdot \sqrt{n+1}$</td>
</tr>
<tr>
<td>UF2</td>
<td>0.0071</td>
<td>0.0739</td>
<td>0.0788</td>
<td>0.0579</td>
<td>0.2228</td>
<td>$1 + \sqrt{n-26}$</td>
</tr>
<tr>
<td>UF3</td>
<td>0.0287</td>
<td>0.1345</td>
<td>0.1356</td>
<td>0.0884</td>
<td>0.2422</td>
<td>$1 + \sqrt{n-27}$</td>
</tr>
<tr>
<td>UF4</td>
<td>0.0050</td>
<td>0.008203</td>
<td>0.008635</td>
<td>0.004070</td>
<td>0.026503</td>
<td>$\sqrt{n+1}$</td>
</tr>
<tr>
<td>UF5</td>
<td>0.0053</td>
<td>0.00699</td>
<td>0.00699</td>
<td>0.00111</td>
<td>0.00896</td>
<td>$1 + \sqrt{n-27}$</td>
</tr>
<tr>
<td>UF6</td>
<td>0.0067</td>
<td>0.0175</td>
<td>0.0184</td>
<td>0.0092</td>
<td>0.0405</td>
<td>$1 + \sqrt{n-27}$</td>
</tr>
<tr>
<td>UF7</td>
<td>0.004952</td>
<td>0.005596</td>
<td>0.005717</td>
<td>0.004871</td>
<td>0.070131</td>
<td>$\sqrt{n+1}$</td>
</tr>
<tr>
<td>UF8</td>
<td>0.0148</td>
<td>0.0296</td>
<td>0.0356</td>
<td>0.0205</td>
<td>0.0864</td>
<td>$1 + \sqrt{n-27}$</td>
</tr>
<tr>
<td>UF9</td>
<td>0.005455</td>
<td>0.022374</td>
<td>0.030286</td>
<td>0.03823</td>
<td>0.091197</td>
<td>$\sqrt{n+1}$</td>
</tr>
<tr>
<td>UF10</td>
<td>0.04572</td>
<td>0.05558</td>
<td>0.05568</td>
<td>0.0049</td>
<td>0.0666</td>
<td>$1 + \sqrt{n-27}$</td>
</tr>
</tbody>
</table>

Table 4: The IGD-metric values of the MOEA/D-AD with parameters $F = 0.5 \cdot (1 + \text{rand})$. |

<table>
<thead>
<tr>
<th>CEC'09</th>
<th>min</th>
<th>median</th>
<th>mean</th>
<th>std</th>
<th>max</th>
<th>parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF1</td>
<td>0.004847</td>
<td>0.010079</td>
<td>0.01099</td>
<td>0.003732</td>
<td>0.019072</td>
<td></td>
</tr>
<tr>
<td>UF2</td>
<td>0.005804</td>
<td>0.008230</td>
<td>0.008306</td>
<td>0.00155</td>
<td>0.012082</td>
<td></td>
</tr>
<tr>
<td>UF3</td>
<td>0.006955</td>
<td>0.046845</td>
<td>0.044923</td>
<td>0.027546</td>
<td>0.093627</td>
<td></td>
</tr>
<tr>
<td>UF4</td>
<td>0.039672</td>
<td>0.048962</td>
<td>0.051182</td>
<td>0.077402</td>
<td>0.064670</td>
<td></td>
</tr>
<tr>
<td>UF5</td>
<td>0.127310</td>
<td>0.270832</td>
<td>0.281138</td>
<td>0.087448</td>
<td>0.480509</td>
<td></td>
</tr>
<tr>
<td>UF6</td>
<td>0.041672</td>
<td>0.172887</td>
<td>0.196197</td>
<td>0.150070</td>
<td>0.754536</td>
<td></td>
</tr>
<tr>
<td>UF7</td>
<td>0.049888</td>
<td>0.008514</td>
<td>0.009190</td>
<td>0.003596</td>
<td>0.23703</td>
<td></td>
</tr>
<tr>
<td>UF8</td>
<td>0.068072</td>
<td>0.088657</td>
<td>0.087818</td>
<td>0.067628</td>
<td>0.102548</td>
<td></td>
</tr>
<tr>
<td>UF9</td>
<td>0.043128</td>
<td>0.060957</td>
<td>0.087238</td>
<td>0.04277</td>
<td>0.177565</td>
<td></td>
</tr>
<tr>
<td>UF10</td>
<td>0.215141</td>
<td>0.344499</td>
<td>0.372537</td>
<td>0.107854</td>
<td>0.627219</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>CEC'09</th>
<th>min</th>
<th>Best</th>
<th>median</th>
<th>mean</th>
<th>Std</th>
<th>max/worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF1</td>
<td>0.0312</td>
<td>0.0638</td>
<td>0.0749</td>
<td>0.0373</td>
<td>0.1886</td>
<td></td>
</tr>
<tr>
<td>UF2</td>
<td>0.0101</td>
<td>0.0180</td>
<td>0.0206</td>
<td>0.0107</td>
<td>0.0476</td>
<td></td>
</tr>
<tr>
<td>UF3</td>
<td>0.1148</td>
<td>0.2066</td>
<td>0.2042</td>
<td>0.0389</td>
<td>0.2693</td>
<td></td>
</tr>
<tr>
<td>UF4</td>
<td>0.0496</td>
<td>0.0560</td>
<td>0.0581</td>
<td>0.0075</td>
<td>0.0785</td>
<td></td>
</tr>
<tr>
<td>UF5</td>
<td>0.1709</td>
<td>0.3797</td>
<td>0.3787</td>
<td>0.0131</td>
<td>0.6431</td>
<td></td>
</tr>
<tr>
<td>UF6</td>
<td>0.0775</td>
<td>0.2046</td>
<td>0.2848</td>
<td>0.2056</td>
<td>0.8327</td>
<td></td>
</tr>
<tr>
<td>UF7</td>
<td>0.1609</td>
<td>0.3736</td>
<td>0.4118</td>
<td>0.2073</td>
<td>0.8313</td>
<td></td>
</tr>
<tr>
<td>UF8</td>
<td>0.0129</td>
<td>0.1146</td>
<td>0.1909</td>
<td>0.1950</td>
<td>0.6199</td>
<td></td>
</tr>
<tr>
<td>UF9</td>
<td>0.0132</td>
<td>0.0247</td>
<td>0.1431</td>
<td>0.1717</td>
<td>0.5134</td>
<td></td>
</tr>
<tr>
<td>UF10</td>
<td>0.891378</td>
<td>1.016922</td>
<td>1.009308</td>
<td>0.066704</td>
<td>1.117835</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>CEC'09</th>
<th>min</th>
<th>Best</th>
<th>median</th>
<th>mean</th>
<th>Std</th>
<th>max/worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF1</td>
<td>0.038334</td>
<td>0.081224</td>
<td>0.091323</td>
<td>0.045432</td>
<td>0.260354</td>
<td></td>
</tr>
<tr>
<td>UF2</td>
<td>0.014111</td>
<td>0.023313</td>
<td>0.033506</td>
<td>0.026654</td>
<td>0.095736</td>
<td></td>
</tr>
<tr>
<td>UF3</td>
<td>0.204872</td>
<td>0.277106</td>
<td>0.232482</td>
<td>0.012183</td>
<td>0.280650</td>
<td></td>
</tr>
<tr>
<td>UF4</td>
<td>0.053795</td>
<td>0.065551</td>
<td>0.122970</td>
<td>0.301495</td>
<td>1.716010</td>
<td></td>
</tr>
<tr>
<td>UF5</td>
<td>0.182200</td>
<td>0.426878</td>
<td>0.437333</td>
<td>0.069366</td>
<td>0.64224</td>
<td></td>
</tr>
<tr>
<td>UF6</td>
<td>0.181908</td>
<td>0.446682</td>
<td>0.417977</td>
<td>0.125357</td>
<td>0.780010</td>
<td></td>
</tr>
<tr>
<td>UF7</td>
<td>0.013394</td>
<td>0.227361</td>
<td>0.225019</td>
<td>0.198278</td>
<td>0.557805</td>
<td></td>
</tr>
<tr>
<td>UF8</td>
<td>0.450662</td>
<td>0.637692</td>
<td>0.668361</td>
<td>0.111094</td>
<td>0.946083</td>
<td></td>
</tr>
<tr>
<td>UF9</td>
<td>0.131885</td>
<td>0.236197</td>
<td>0.228897</td>
<td>0.093954</td>
<td>0.292767</td>
<td></td>
</tr>
<tr>
<td>UF10</td>
<td>0.203688</td>
<td>0.453790</td>
<td>0.442667</td>
<td>0.055197</td>
<td>0.508389</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>CEC'09</th>
<th>min</th>
<th>Best</th>
<th>median</th>
<th>mean</th>
<th>Std</th>
<th>max/worst</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF1</td>
<td>0.0305</td>
<td>0.0387</td>
<td>0.0885</td>
<td>0.0406</td>
<td>0.1809</td>
<td></td>
</tr>
<tr>
<td>UF2</td>
<td>0.0319</td>
<td>0.0421</td>
<td>0.0433</td>
<td>0.0067</td>
<td>0.0580</td>
<td></td>
</tr>
<tr>
<td>UF3</td>
<td>0.0943</td>
<td>0.1818</td>
<td>0.1822</td>
<td>0.0426</td>
<td>0.2610</td>
<td></td>
</tr>
<tr>
<td>UF4</td>
<td>0.0125</td>
<td>0.0483</td>
<td>0.2200</td>
<td>0.2604</td>
<td>0.6014</td>
<td></td>
</tr>
<tr>
<td>UF5</td>
<td>1.1413</td>
<td>1.3027</td>
<td>1.2868</td>
<td>0.0967</td>
<td>1.4228</td>
<td></td>
</tr>
<tr>
<td>UF6</td>
<td>0.6433</td>
<td>1.0829</td>
<td>1.1249</td>
<td>0.2695</td>
<td>1.7226</td>
<td></td>
</tr>
</tbody>
</table>
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.
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 include 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


