Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2874528, 17 pages
https://doi.org/10.1155/2020/2874528

Research Article

A Novel Parent Centric Crossover with the Log-Logistic Probabilistic Approach Using Multimodal Test Problems for Real-Coded Genetic Algorithms

Ehtasham ul Haq,[1] **Ishfaq Ahmad**,[1] **and Ibrahim M. Almanjahie**[2,3]

[1]*Department of Mathematics and Statistics, International Islamic University, Islamabad, Pakistan*
[2]*Department of Mathematics, College of Science, King Khalid University, Abha 62529, Saudi Arabia*
[3]*Statistical Research and Study Support Unit, King Khalid University, Abha 62529, Saudi Arabia*

Correspondence should be addressed to Ehtasham ul Haq; ehtasham.malik@iiu.edu.pk

Received 23 July 2020; Revised 24 September 2020; Accepted 8 October 2020; Published 31 October 2020

Academic Editor: Predrag S. Stanimirović

In this paper, a comprehensive empirical study is conducted to evaluate the performance of a new real-coded crossover operator called Fisk crossover (FX) operator. The basic aim of the proposed study is to preserve population diversity as well as to avoid local optima. In this context, a new crossover operator is designed and developed which is linked with Log-logistic probability distribution. For its global performance, a realistic comparison is made between FX versus double Pareto crossover (DPX), Laplace crossover (LX), and simulated binary crossover (SBX) operators. Moreover, these crossover operators are also used in conjunction with three mutation operators called power mutation (PM), Makinen, Periaux, and Toivanen mutation (MPTM), and nonuniform mutation (NUM) for inclusive evaluation. The performance of probabilistic-based algorithms is tested on a set of twenty-one well-known nonlinear optimization benchmark functions with diverse features. The empirical results show a substantial dominance of FX over other crossover operators with authentication of performance index (PI). Moreover, we also examined the significance of the proposed crossover scheme by administrating ANOVA and Gabriel pairwise multiple comparison test. Finally, the statistically significant results of the proposed crossover scheme have a definite edge over the other schemes, and it is also expected that FX has a great potential to solve complex optimization problems.

1. Introduction

In many real-life decision-making problems, it is often the best possible solutions are required. These problems may be anything from engineering, science, economics, and finance [1–3]. When the quality of potential solutions can be modeled mathematically, it may be possible to algorithmically find a better and sometimes optimal solution. In this case, decisions are made by developing optimization models that describe the nature of the problem, and then mathematical procedures are applied to solve these models. Hence, the simplest optimization scenario is based on constrained optimization problems, but the present research study focuses only on unconstrained optimization problems. More generally, unconstrained nonlinear optimization problem may be mathematically defined as

$$\text{Min } f(y), \quad \text{where } f\colon R^n \longrightarrow R, \tag{1}$$

where $y \in S$, and S is a rectangular hypercube with n-dimensions in R^n with limits $a_i \le y_i \le b_i$, $i = 1, 2, 3, \ldots, n$. These are commonly known as bounds which are based on the decision variables. A point $y^+ \in S$ is known as local minima of f if $f(y^+) \le f(y)$ $y \in N_\varepsilon(y^+) \cap S$, where $N_\varepsilon(y^+) = \{y \,\| y - y^+ \| < \varepsilon, \, \varepsilon > 0\}$ is the small neighborhood of the point (y^+). If $f(y^+) \le f(y)$ $y \in S$, then y^+ is said to be the global minima of f.

The optimization techniques for unconstrained problems invariably used the gradient information to locate the optima. Hence, proximal gradient descent is also a gradient-based optimization method, which can be used to solve objective functions with nondifferentiable parts. These techniques cannot efficiently handle discrete variables and are most probably stuck at local optima for multimodal objective functions. Hence, the gradient-based methods often ensure that the local optima will be reached at the global optima.

To preserve population diversity and to avoid local optima, there are different types of population-based probabilistic optimization techniques that do not require continuous or differentiable objective functions. Such techniques are genetic algorithms (GAs) [4, 5], particle swarm optimization (PSO) [6], simulated annealing (SA) [7], differential evolution [8], ant colony optimization (ACO) [9], and Tabu search [10, 11]. All these optimization techniques are hence categorized under guided random search methods [12]. The guided random search methods provide a very efficient solution to the combinatorial problems and can be subcategorized into evolutionary techniques. GA is the most efficient procedure to understand and solve problems for which there is limited information. These algorithms can effectively handle both unconstrained and constrained optimization problems depending on a process of natural selection through biological evolution. The working mechanism of GA is linked with a search space that contains all possible solutions. Each part of the search space represents one sufficient solution and its fitness values will be marked by these sufficient solutions. A strong chromosome can survive, and usually, the weak chromosomes are eliminated from the population. GA is working with a search space that contains all feasible solutions. It means that each of the points in the search space is to obtain one feasible solution that will be marked according to its fitness [13]. The main operators of the GA process are selection, crossover, and mutation, and all these operators make the algorithm more unique compared to other conventional algorithms in the optimization scenario. The ideology of the selection process is to select the good chromosomes which are sent to the mating pool for combining with the other chromosomes to reproduce two new offspring. Meanwhile, the mutation process aims to encourage diversity in the new population with minor probability [14].

In previous studies, the representation of chromosomes was in the binary form which contains 0's (absent) and 1's (present), and initially, it was applied due to its relative simplicity. Binary-coded scheme transforms the continuous search space into discrete nature with grids and the string length depends on the distance between two neighboring grids. The performance of binary encoding is good and also required less precision for the solutions under a limited number of decision variables. On the other way, binary encoding performs unsatisfactorily to solve multidimensional optimization problems where high precision is required. By the use of multiple decision variables in objective functions, the optimization problems may be solved efficiently with the increase in the size of the population. Hence, increment in the string length may be achieved better precision. In earlier optimization studies, Goldberg [12] explored that the increase in string length exponentially increased computation time and with some additional adjustments in the binary encoding will improve the convergence speed in genetic algorithms which were cited in Jin et al. [15].

Generally, GA used binary encoding in the earlier nineteen century, but many researchers [and 16–22] used real numbers for encoding. Mohamed et al. [23] recently proposed gaining sharing knowledge-based algorithm (GSK) for solving naturally inspired optimization problems over continuous space. The designing and development of a real-encoding scheme were naturally suitable to solve optimization problems with continuous variables. As a binary coding scheme, similar genetic operators such as selection, crossover, and mutation are used in real-coded GA. The major advantage of real encoding over binary encoding is to efficiently handle complex nonlinear optimization problems with the continuous domain. Hence, by using real-encoded schemes, many difficulties such as premature convergence are to be solved, and such situations arise because of low genetic diversity in the population, and also it represents the poor exploration of the search space. If all the chromosomes have nearly the same empirical value, then the resulting process may be trapped at local optima. In the above context, a substantial amount of algorithmic work has been carried out to improve the performance of the GA process by increasing exploitation and exploration potential [24]. The exploration strength of the GA process mainly relies on the use of crossover operators due to the utilization of information about the current population. This is one of the key reasons that the majority of researchers pay more attention to performance improvement of the GA process through efficiently designed and developed real-coded crossover operators.

A dynamic class of crossover operators with real encoding is also known as parent centric crossover operators. Deb et al. [25] have shown that these parent centric crossover operators are quite effective to obtain optimum solutions of real parametric problems. In our current research study, a newly proposed real-coded crossover operator is based on log-logistic probability distribution which is empirically defined. The comparative performance evaluation of newly proposed parent centric crossover operator (FX) is carried out with double Pareto crossover (DPX), Laplace crossover (LX), and simulated binary crossover (SBX) operators. Moreover, these crossover operators are also used in conjunction with three main mutation operators called power mutation (PM), nonuniform mutation (NUM), and Makinen, Periaux, and Toivanen mutation (MPTM) proposed by Deep and Thakur [26], Meittinen et al. [27], Deb [5], and Maaranen et al. [28]. To evaluate the significance of the simulated study, the pairwise comparison is carried out for all algorithms based on statistical measures.

The rest of this paper is organized as follows. Concise detail about some existing crossover operators with real coding is discussed in Section 2. Description with the mathematical formulation of previous crossover schemes along with tournament selection is presented in Section 3. Designing and development of newly proposed crossover operator based on log-logistic probability distribution along with the description of some well-known benchmark functions are presented in Section 4. Experimentation detail with statistical analysis is described in Section 5, while simulated results by using twenty-one well-known benchmark functions are revealed and discussed in Section 6. Performance evaluation for comparative study is presented in Section 7. The conclusions of the study are presented in Section 8 of this paper.

2. Literature Review of Crossover Operators with Real Coding

Different classes of crossover operators with real coding have been proposed in the literature related to the GA process. In the earlier expansion of real coding, Wright [18] and Michalewicz et al. [29] suggested a heuristic crossover approach where the selection of a gene's position is purely a random process and the generation of two new offspring is resulting by exchanging the position of their respective genes. Radcliffe [30] explained a Flat Crossover where a random generation of offspring is based on uniform distribution. Muhlebein and Schlierkamp-Voosen [31] proposed an extended form of intermediate crossover and line crossover. Both these crossover operators improved their search competencies by allowing population diversity under a specified interval. Eshelman and Schaffer [32] further extended the ideology of Radcliffe [30] by introducing the theoretical approach of interval schemata in the real-coding scheme. A Blend-α crossover is a hybridization of intermediate crossover and line crossover which was introduced by Muhlebein and Schlierkamp-Voosen [31]. With the benefit of the user-defined parameter, this crossover defined a bond in between genes of the parent as well as equally on either side. Moreover, blend crossover can be transformed into an intermediate crossover with $\alpha = 0.25$.

Michalewicz et al. [29] presented a theory of sandwiched offspring between two parents in an arithmetic crossover operator. Moreover, a random selection of genes for producing offspring which is based on the uniform distribution in single arithmetic crossover and more offspring is generated by duplicating the process of earlier produced offspring through the average measure. The generation process of a single arithmetical crossover is generalized to all arithmetic crossovers. For more logically mathematical extension in the GA process, fuzzy recombination crossover, and fuzzy min-max crossover operators are introduced by Voigt et al. [33, 34]. Herrera and Lozano [21] presented a class of dynamic heuristic real-coded crossover operators that are linked with fuzzy connectives to overcome the shortcomings including premature convergence. These crossover operators focus on maintaining population diversity with the sustainable convergence speed of the genetic process. Another unique idea of a multiparent crossover is presented by Tsutsui et al. [35] in a simplex real-coded crossover operator by using a uniform probability distribution. Deb and Agrawal [19] repeat the development strategy of single-point crossover by using a binary string is continuous while introducing simulated binary crossover. Tutkun [36] is also introduced as a crossover operator linked with Gaussian probability distribution. Kaelo and Ali [37] proposed a hybridization strategy of different crossovers by reviewing the functionality of these crossover operators. Laplace probability distribution is also used in real-coded crossover for producing new offspring by Deep and Thakur [26].

Generally, the crossover scheme creates offspring with the help of those parents who are selected through the selection operator. It combines the parents' characteristics to procedure new offspring, which may have new sequences compared to those of their parents and play a vital role in the GAs. In the literature, a lot of crossover schemes have been introduced with their significant importance. Chen and Wang [38] proposed a real-coded crossover (UNDX) by using unimodal normal probability distribution where three parents produce multiple offspring through the crossover process. After that Ono and Kobayashi et al. [39] integrated a uniform crossover operator to enhance the working capability of UNDX in the genetic process. A generalized form of UNDX operator having multiple parents was introduced by Kita et al. [40]. Another multiple parent crossover operator, i.e., parent-centric crossover (PCX) operator was reported by Deb et al. [25] with some enhanced features and a modified class of PCX was also introduced by Sinha et al. [41]. Furthermore, Chandra et al. [42] proposed a novel cooperative coevolution working strategy by incorporating PCX with a memetic framework for neural networks without adding to the computational cost in the subpopulations. After the hybridization with the neural network, island-based cooperative coevolutionary algorithms are presented by Bali et al. [43] to solve large-scale fully separable continuous optimization problems under the framework of PCX. Rolland and Chandra [44] also used PCX to efficiently handle the forward kinematics problem (FKP) for parallel manipulators.

By the integration of bound crossover and average crossover, Ling and Leung [45] suggested an average-bound crossover operator where two parents generated four offspring by the selection of two best out of four offspring. It is also noted in the literature that some of the various types of crossover operators including hybrid crossover operators are more beneficial to enhance the performance of the genetic process by introducing a fair amount of population diversity and controlling selection pressure. Hence, Herrera et al. [46] conducted a comprehensive performance evaluation study among real-coded crossover operators to explore the effectiveness of the genetic process. Real-coded crossover operators are parent centric crossover, fuzzy recombination, parent centric blend crossover, simulated binary crossover, and XLM crossover [47], and Laplace real-coded crossover are parent-centric crossovers which by using a unimodal normal probability distribution, blend crossover, and simplex crossover are all based on mean-centric crossover approaches.

3. Some Presently Used GA Operators with Real Coding in the Study

In the current section, we descriptively delineate selection, crossover, and mutation operators that are tournament selection (TS), double Pareto crossover (DPX), Laplace crossover

(LX), and simulated-binary crossover (SBX) along with different mutation operators which are power mutation operator (PM), Makinen, Periaux, and Toivanen mutation (MPTM), and nonuniform mutation (NUM). Hence, the empirical findings are to be obtained through the simulated study.

3.1. Selection Operators

3.1.1. Tournament Selection (TS). TS method is a ranked-based selection procedure that is simple to execute. This selection technique can be classified as binary and large tournament selection. The binary tournament selection is based on the randomized selection of two individuals, conducting a competition to decide which chromosome will win and get selected for the mating pool on the basis of the highest fitness value [14]. On the contrary, there will be no place in the mating pool for the weakest chromosomes in terms of the least fitness value, and then compare it to a predetermined selection probability p_i. Hence, the predetermine selection probability for individual p_i for $(t-1)$ tournament is given as

$$p_i = \frac{1}{W^t}\left((W-i+1)^t - (W-i)^t\right), \quad i \in \{1, 2, \ldots, W\}, \tag{2}$$

where W is defined as population size and t is the size of the tournament. For the binary tournament $t=2$ and large tournament $t>2$, the TS can also be further extended by including more than two individuals if desired [48].

3.2. Crossover Operators

3.2.1. Double Pareto Crossover (DPX). DPX operator [49] is another kind of parent centric crossover operator that uses the double Pareto probability distribution whose cumulative distribution function is expressed as

$$F(y) = \begin{cases} \dfrac{1}{2}\left(1 - \dfrac{y}{\alpha\beta}\right)^{-\alpha}, & y < 0, \\[2ex] \dfrac{1}{2}\left[1 - \left(1 - \dfrac{y}{\alpha\beta}\right)^{-\alpha}\right], & y \leq 0, \end{cases} \tag{3}$$

where $\alpha \in R$ is represented as the location parameter where $\beta > 0$ is known as a scale parameter of the distribution. By using DPX, pair of offspring $z^{\{1\}} = (z_1^{\{1\}}, z_2^{\{1\}}, \ldots, z_n^{\{1\}})$ and $z^{\{2\}} = (z_1^{\{2\}}, z_2^{\{2\}}, \ldots, z_n^{\{2\}})$ are generated from two parents $y^{\{1\}} = (y_1^{\{1\}}, y_2^{\{1\}}, \ldots, y_n^{\{1\}})$ and $y^{\{2\}} = (y_1^{\{2\}}, y_2^{\{2\}}, \ldots, y_n^{\{2\}})$ in a subsequent stepwise procedure:

Step 1: generate a random number (r_i) from a uniform probability distribution, where $r_i \in [0, 1]$.

Step 2: calculate the value of the parameter β_i by generating random numbers from double Pareto probability distribution by simply finding the inverse of the cumulative distribution function of double Pareto distribution as

$$\beta_i = \begin{cases} \left(\alpha\beta\left(1 - (2r_i)^{-1/\alpha_i}\right)\right), & r_i \leq 0.5, \\ \left(\alpha\beta\left(1 - (2r_i)^{-1/\alpha_i}\right) - 1\right), & r_i > 0.5. \end{cases} \tag{4}$$

Step 3: now, the offspring are in equations (5) and (6) for $i = 1, 2, \ldots, n$:

$$z_i^{\{1\}} = \frac{\left(y_i^{\{1\}} + y_i^{\{2\}}\right) + \beta_i\left|y_i^{\{1\}} - y_i^{\{2\}}\right|}{2}, \tag{5}$$

$$z_i^{\{2\}} = \frac{\left(y_i^{\{1\}} + y_i^{\{2\}}\right) - \beta_i\left|y_i^{\{1\}} - y_i^{\{2\}}\right|}{2}. \tag{6}$$

If the generation of offspring in DPX is outside the variable bound, i.e., $y_i < y_i^l$ or $y_i < y_i^{ri}$, then random values are given to y_i in an interval $[y_i^l, y_i^{ri}]$.

3.2.2. Laplace Crossover (LX). LX crossover operator [26] is a class of self-adaptive crossover operators with real encoding. This crossover is linked with Laplace probability distribution and the cumulative distribution function is expressed subsequently as

$$F(y) = \begin{cases} \dfrac{1}{2}e^{(|y-a|/b)}, & y \leq a, \\[2ex] \left[1 - \dfrac{1}{2}e^{(|y-a|/b)}\right], & y > a. \end{cases} \tag{7}$$

In the Laplace probability distribution, $b > 0$ is defined as the scale parameter and $a \in R$ is known as the location parameter. By using LX, from a pair of parents $y^{\{1\}} = (y_1^{\{1\}}, y_2^{\{1\}}, \ldots, y_n^{\{1\}})$ and $y^{\{2\}} = (y_1^{\{2\}}, y_2^{\{2\}}, \ldots, y_n^{\{2\}})$, two offspring $z^{\{1\}} = (z_1^{\{1\}}, z_2^{\{1\}}, \ldots, z_n^{\{1\}})$ and $z^{\{2\}} = (z_1^{\{2\}}, z_2^{\{2\}}, \ldots, z_n^{\{2\}})$ are generated through a stepwise approach:

Step 1: generate a random number (r_i) with a range of unity from a uniform distribution.

Step 2: obtain the parametric value β_i by generating random numbers from Laplace probability distribution by simply inverting the cumulative distribution function as

$$\beta_i = \begin{cases} a - b\log_e(r_i), & r_i \leq \dfrac{1}{2}, \\[2ex] a + b\log_e(r_i), & r_i > \dfrac{1}{2}. \end{cases} \tag{8}$$

Step 3: now, the offspring are obtained in the subsequent equations (9) and (10), respectively, for $i = 1, 2, \ldots, n$:

$$z_i^{\{1\}} = y_i^{\{1\}} + \beta_i \left| y_i^{\{1\}} - y_i^{\{2\}} \right|, \quad (9)$$

$$z_i^{\{2\}} = y_i^{\{2\}} + \beta_i \left| y_i^{\{1\}} - y_i^{\{2\}} \right|. \quad (10)$$

If the generation of offspring is outside the variable limits, i.e., $y_i < y_i^l$ or $y_i < y_i^{ri}$ in LX, then random values are given to y_i in certain bound $[y_i^l, y_i^{ri}]$ and assigned "0" value to its location parameter.

3.2.3. Simulated Binary Crossover (SBX). The SBX is an important crossover operator with real coding. This operator was proposed by Deb and Agrawal [19] with a unique feature of binary transformation to continuous search space. At the first step, random number r_i is generated from a uniform distribution with range 0 to 1. After that, the parametric value β_i is obtained from the following mathematical expression as

$$\beta_i = \begin{cases} (2r_i)^{1/(n_c+1)}, & \text{if } r_i \leq \dfrac{1}{2}, \\ \dfrac{1}{(2-2r_i)^{1/(n_c+1)}}, & \text{otherwise,} \end{cases} \quad (11)$$

where $n_c \in [0, \infty]$ is known as the distribution index.

Hence, from the two parents $y^{\{1\}} = (y_1^{\{1\}}, y_2^{\{1\}}, \ldots, y_n^{\{1\}})$ and $y^{\{2\}} = (y_1^{\{2\}}, y_2^{\{2\}}, \ldots, y_n^{\{2\}})$, an offspring $z = (z_1, z_2, \ldots, z_n)$ is produced in the following equation:

$$z_i = \frac{1}{2}\left[\left(y_i^{\{1\}} + y_i^{\{2\}}\right) - \beta_i \left| y_i^{\{1\}} - y_i^{\{2\}} \right|\right]. \quad (12)$$

3.3. Mutation Operators

3.3.1. Power Mutation (PM). The mutation process prevents the new population from stopping at the local optima and keeps the fair amount of diversity in the population. In other words, this process will generate a unique and fit offspring in the population. Goldberg [1] proposed a mutation clock to overcome the problem of the computational complexity in the mutation process. He used the exponential distribution to find the next location to change the string by using the first changed string location. The mutation in this process is aiming to explore more searching space while the crossover tries to converge on some point. This is because the role of the mutation is to solve the local minimum problem. Thus, this process is one way to prevent local minimum solutions at the expense of exploring more areas. Here, we subsequently represent the PM mutation operator by Deep and Thakur [26].

The PM is originated from power distribution and *p.d.f* of power distribution is as with cumulative distribution function in equations (13) and (14), respectively:

$$f(y) = py^{p-1}, \quad 0 \leq y \leq 1, \quad (13)$$

$$F(y) = y^p, \quad 0 \leq y \leq 1, \quad (14)$$

where p is represented as distribution index and PM is used to obtain an offspring $z = (z_1, z_2, \ldots, z_n)$ from a parent $y = (y_1, y_2, \ldots, y_n)$ in the following stepwise procedure:

Step 1: obtain a random value (r_i) from a uniform distribution, where $r_i \in [0, 1]$.

Step 2: calculate a random value s_i by using power distribution:

$$s_i = (r_i)^{1/p}. \quad (15)$$

Now, use the following mathematical expressions to obtain offspring:

$$z_i = \begin{cases} y_i - s_i\left(y_i - y_i^l\right), & \text{if } \dfrac{y_i - y_i^l}{y_i^{ri} - y_i} < r_i, \\ y_i + s_i\left(y_i - y_i^l\right), & \text{if } \dfrac{y_i - y_i^l}{y_i^{ri} - y_i} \geq r_i, \end{cases} \quad (16)$$

where y_i^l and y_i^{ri} are the lower and upper bound of the ith decision variable. In the context of the above mathematical formulation, it is stated that perturbation in offspring is proportional to the "p" parameter. Hence, the probability of obtaining a mutated offspring is proportional to the distance from parametric bound which always resulted in feasible solution.

3.3.2. Makinen, Periaux, and Toivanen Mutation (MPTM). The MPTM mutation operator was suggested by Makinen et al. [50], which is used to solve some multidisciplinary shape-related optimization problems in GA especially in the field of electromagnetics and aerodynamics. Meittinen et al. [27] also solved constrained optimization problems under the GA process. Deep and Thakur [22] tested and evaluated its results on multimodal nonlinear optimization problems. From a point $y = (y_1, y_2, \ldots, y_n)$, the mutated point $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)$ is obtained in the following way.

Let a random value (r_i) be from a uniform distribution, where $r_i \in [0, 1]$. Hence, the muted solution is in the following equations:

$$\hat{y}_i = (1 - \hat{t})y_i^l + (\hat{t})y_i^{ri}, \quad (17)$$

where

$$\hat{t}_i = \begin{cases} t_i - t_i\left(\dfrac{t_i - r_i}{t_i}\right)^b, & \text{if } r_i < t_i, \\ t_i, & \text{if } r_i = t_i, \\ t_i + (1 - t_i)\left(\dfrac{t_i - r_i}{1 - t_i}\right)^b, & \text{if } r_i \geq t_i, \end{cases} \tag{18}$$

$$t = \frac{y - y_i^l}{y_i^{ri} - y}. \tag{19}$$

Hence, the y_i^l and y_i^{ri} are the lower and upper limits of ith decision variable, respectively.

3.3.3. Nonuniform Mutation (NUM). The most extensively used mutation operators are Michalewicz's NUM mutation operator in real-coded GAs. The working mechanism of nonuniform mutation could be initiated by Michalewicz et al. [29, 51]. For implementation context, by increasing the number of generations in the simulation process, the strength of the mutation might be reduced so that it can search uniformly for the initial generations of the process while it may search locally for later generations. For a point $y^{\{1\}} = (y_1^{\{p\}}, y_2^{\{p\}}, \ldots, y_n^{\{p\}})$, a muted point $y^{\{p+1\}} = (y_1^{\{p+1\}}, y_2^{\{p+1\}}, \ldots, y_n^{\{p+1\}})$ is developed subsequently:

Step 1: generate a random value u_i from a uniform probability distribution, $u_i \in [0, 1]$.

Step 2: obtain a muted solution by following mathematical expression:

$$y_i^{p+1} = \begin{cases} \left(y_i^p + \left(x_i^u - y_i^p\right)\left(1 - u_i^{(1-(p/P))}\right)^b\right), & r_i \leq 0.5, \\ \left(y_i^p - \left(y_i^p - x_i^l\right)\left(1 - u_i^{(1-(p/P))}\right)^b\right), & \text{otherwise}, \end{cases} \tag{20}$$

where "b" is a parametric value that determines the working capability of the mutation operator. P and p represent the number of maximum generations and the number of the current generation. x_i^u and x_i^l are the upper and lower limits of the ith decision variable, respectively.

4. The Proposed Fisk Crossover (FX) Operator Based on Log-Logistic Probability Distribution

The log-logistic probability distribution has been used in networking to model the transmission times of data considering both the network and the software, in hydrology to model streamflow and precipitation, and in economics as a simple model of the distribution of wealth or income. This distribution efficiently deals with the modeling of the delays in the transmission of sensory data coming from a networked telerobot, which would allow us to predict the future times of arrival and provide assurance on the time requirements of these systems [52]. Besides, log-logistics distribution is also used to estimate low-flow frequency based on the analysis on extreme low flow event within a specific time interval [53].

So, with diversified application in many optimization related fields, the presently designed novel real-coded crossover scheme is naturally suitable to solve optimization problems with continuous variables. The major advantage of real encoding in GA over binary encoding is to efficiently handle complex nonlinear optimization problems with continuous domain and also overcome the fundamental issues such as exploration (population diversity) and exploitation (selection pressure). Hence, the binary encoding scheme inherited many difficulties such as premature convergence which can be solved by real encoding in the GA process. Hence, we present a Fisk crossover (FX), and this is a parent centric crossover operator that is linked with log-logistic distribution [52].

Figure 1 reveals the working mechanism of a mixture probability distribution-based real-coded crossover operator. Initially, we generate a random population of size "N" chromosomes with predefined encoding followed by evaluating each chromosome according to its fitness value. Afterword, approximately half of the chromosomes is selected through tournament selection. In the next step, recombine new offspring using FX, DPX, LX, and SBX crossover operators within the conjunction of three mutation operators, i.e., MTPM, PM, and NUM with predefined probability for minor changes in the chromosomes. In the final step, the whole algorithmic process will be continued until the required optimum solution is obtained.

The mathematical working principle of Fisk crossover operator is formulated by using the probability density function of log-logistics distribution, which is given below in

$$f(x) = \frac{(\beta/\alpha)(x/\alpha)^{\beta-1}}{\left(1 + (x/\alpha)^\beta\right)^2}. \tag{21}$$

The cumulative distribution function is also depicted as subsequent:

$$F(x) = y = \begin{cases} \dfrac{1}{1 + (x/\alpha)^{-\beta}}, & x > 0, \\ 1 - \dfrac{1}{1 + (x/\alpha)^{-\beta}}, & x \leq 0, \end{cases} \tag{22}$$

where $\alpha > 0$ is the scale parameter and $\beta > 0$ is the shape parameter.

By using FX, a pair of offspring $z^{\{1\}} = (z_1^{\{1\}}, z_2^{\{1\}}, \ldots, z_n^{\{1\}})$ and $z^{\{2\}} = (z_1^{\{2\}}, z_2^{\{2\}}, \ldots, z_n^{\{2\}})$ is generated from a pair of parents $y^{\{1\}} = (y_1^{\{1\}}, y_2^{\{1\}}, \ldots, y_n^{\{1\}})$ and $y^{\{2\}} = (y_1^{\{2\}}, y_2^{\{2\}}, \ldots, y_n^{\{2\}})$ in a subsequent stepwise procedure.

Step 1: generate a random number (r_i) from a uniform probability distribution, where $r_i \in [0, 1]$.

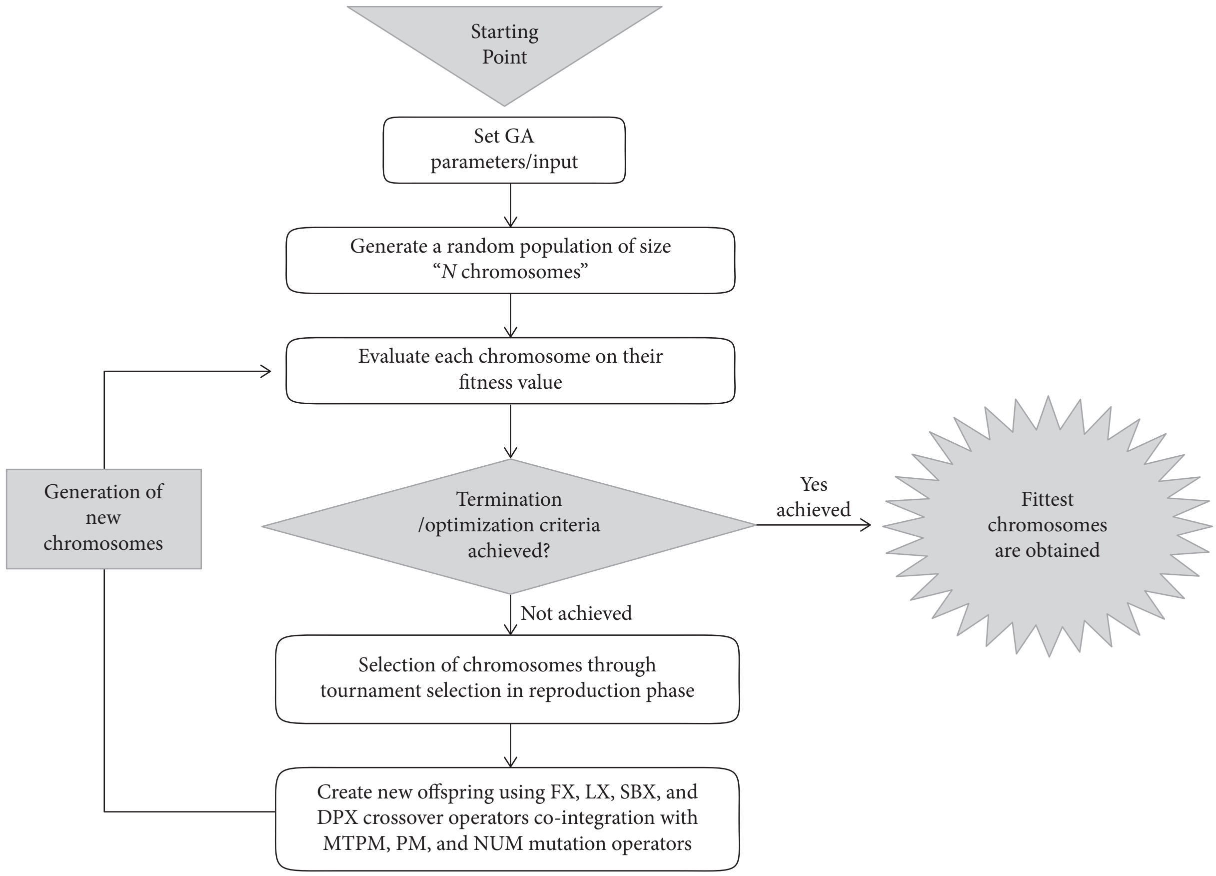

Figure 1: Visual framework of Fisk/log-logistic distribution crossover.

Step 2: calculate the value of the parameter β_q by generating random numbers from Fisk/Log-logistic probability distribution by simply finding the inverse of the cumulative distribution function of Fisk/Log-logistic distribution as

$$\beta_q = \begin{cases} \alpha\left(\dfrac{1-y}{y}\right)^{-1/\beta}, & r_i \leq 0.5, \\ \alpha\left(\dfrac{y}{1-y}\right)^{-1/\beta}, & r_i > 0.5. \end{cases} \quad (23)$$

Step 3: finally, the offspring are generated through subsequent equations (24) and (25), respectively, for $i = 1, 2, \ldots, n$:

$$z_i^{\{1\}} = \frac{\left(y_i^{\{1\}} + y_i^{\{2\}}\right) + \beta_q \left|y_i^{\{1\}} - y_i^{\{2\}}\right|}{2}, \quad (24)$$

$$z_i^{\{2\}} = \frac{\left(y_i^{\{1\}} + y_i^{\{2\}}\right) - \beta_q \left|y_i^{\{1\}} - y_i^{\{2\}}\right|}{2}. \quad (25)$$

In the context of the two equations (24) and (25), it is visualized that the smaller parametric value of β produces offspring close to the parents while the large parametric value of β generates offspring away from the parents for the fixed parametric value of α in Figure 2. Hence, from Figure 3, it is also observed that, with the smaller parametric value of α which produces offspring which are close to parents conversely for the larger value of α, the generation of offspring is away from the parents for the fixed parametric value of β. There is another matter of fact that the generation of offspring through FX crossover is visually symmetrical about the location of the parents as depicted in Figure 4. Therefore, the above mathematical formulation for $z_i^{\{1\}}$ and $z_i^{\{2\}}$ reveals that the distance between the two offspring is proportional to the distance between parents under fixed parametric values of α and β. Hence, proposed FX crossover operator shows self-adaptive behavior with the violation of variable limit constraint, i.e., $y_i < y_i^{\{l\}}$ or $y_i < y_i^{\{ri\}}$, during the generation process of offspring in FX then the random value given to y_i in certain bound $[y_i^{\{l\}}, y_i^{\{ri\}}]$ and assigns "0" value to its location parameter.

4.1. Benchmark Functions. The optimization process is focused on obtaining the global optimum point; consequently, the regions nearby local optima should be circumvented because the optimization process might be stuck at local optima, and then local optima are considered to be as

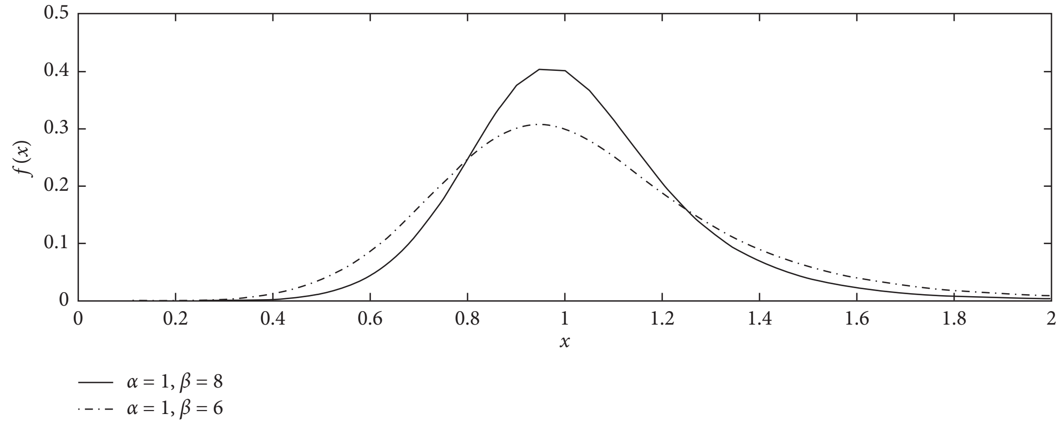

Figure 2: p.d.f of Fisk/Log-logistic distribution for fix α.

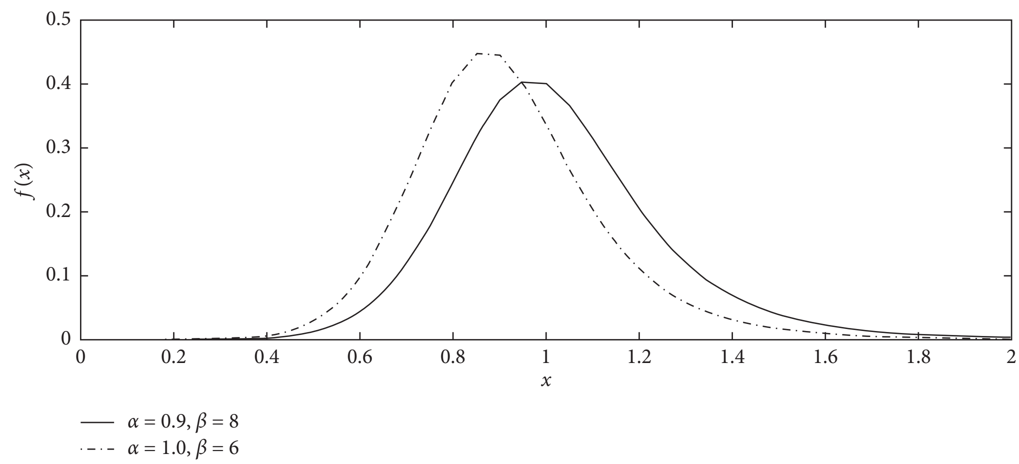

Figure 3: p.d.f of Fisk/Log-logistic distribution for fix β.

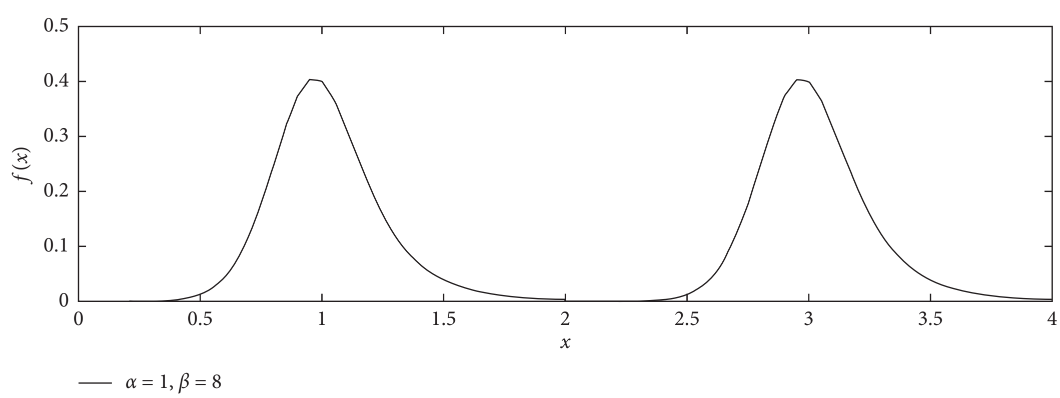

Figure 4: Distribution of offspring.

global optima. To evaluate the performance and sustainability of the proposed real-coded crossover operators, we will use twenty-one unimodal, multimodal, separable or nonseparable, convex, and continuous benchmark functions. Table 1 presents the list of benchmark functions [54] utilized to appraise the efficiency of suggested evolutionary methods. Hence, the benchmark function name, fitness function, search limits, and theoretical optimum value are present in Table 1. These benchmark functions have varying complexities that are most commonly applied in many comparative studies. The necessary details regarding these benchmarks are given in Table 1.

Table 1: Detail of benchmark functions for comparison.

Benchmark	Fitness function	Search limits	Optimum value
Ackley's	$-20\exp(-0.2\sqrt{1/n\sum_{i=1}^{n} x_i^2}) - \exp(1/n\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e(1)$	[−32.768, 32.768]	0
Axis parallel ellipsoid	$f(x) = \sum_{i=1}^{n} i x_i^2$	[−5.12, 5.12]	0
Cigar	$f(x) = x_i^2 + 1000000\sum_{i=1}^{n} x_i^2$	[−10, 10]	0
Cosine mixture	$f(x) = \sum_{i=1}^{n} x_i^2 - 0.1\sum_{i=1}^{n}\cos(5\pi x_i)$	[−1, 1]	0.1 n
De-Jong	$f(x) = \sum_{i=1}^{n}(x_i^4 + \text{rand}(0,1))$	[−10, 10]	0
Drop-wave	$f(x) = \sum_{i=1}^{n} 1 + \cos(12\sqrt{x_i^2 + x_{i+1}^2})/0.5(x_i^2 + x_{i+1}^2) + 2$	[−5.12, 5.12]	−1
Ellipsoidal	$f(x) = \sum_{i=1}^{n}(x_i - i)^2$	[−n, n]	0
Brown	$f(x) = \sum_{i=1}^{n}(x_i^2(x_{i+1}^2 + 1) + x_{i+1}^2(x_{i+1}^2 + 1))$	[−1, 4]	0
Generalized penalized-1	$f(x) = \pi/n(10\sin^2(\pi z_i) + \sum_{i=0}^{n-1}(z_i - 1)^2[1 + 10\sin^2(\pi z_{i+1})] + (z_n - 1)^2) + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ where $z_i = (1 + (x_i + 1))/4$, $u = \begin{cases} k * \text{pow}((x_i - a), m), & \text{if } x > a, \\ k * \text{pow}((-x_i - a), m), & \text{if } x < -a, \\ 0, & \text{otherwise.} \end{cases}$	[−50, 50]	0
Generalized penalized-2	$f(x) = 0.1((sin^2 3\pi x_i + \sum_{i=0}^{n-1}(x_i - 1)^2[1 + sin^2(3\pi x_i)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)] + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ where $u = \begin{cases} k * \text{pow}((x_i - a), m), & \text{if } x > a, \\ k * \text{pow}((-x_i - a), m), & \text{if } x < -a, \\ 0, & \text{otherwise.} \end{cases}$	[−50, 50]	0
Levy and Mantalvo	$f(x) = 0.1(\sin^2(3\pi x_i) + \sum_{i=0}^{n-1}(x_i - 1)^2[1 + \sin^2(3\pi x_i)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)])$	[−5, 5]	0
Matyas	$f(x) = \sum_{i=1}^{n}(0.26(x_i^2 + x_{i+1}^2)) - 0.48x_i x_{i+1})$	[−10, 10]	0
Neumaier	$f(x) = \sum_{i}^{n}(x_i - 1)^2 + \sum_{i=1}^{n-1} x_i x_{i-1}$	$[-n^2, n^2]$	$n(n+4)(n-1)/6$
New function	$f(x) = \sum_{i=0}^{n}[0.2x_i^2 + 0.1x_i^2\sin(2x_i)]$	[−10, 10]	0
Rastrigin	$f(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)]$	[−5.12, 5.12]	0
Rosenbrock	$f(x) = \sum_{i=1}^{n}(100(x_i^2 - x_{i+1})^2) + (1 - x_i)^2$	[−30, 30]	0
Sum of Power	$f(x) = \sum_{i=1}^{n} \lvert x_i\rvert^{(n+1)}$	[−1, 1]	0
Schwefel-1	$f(x) = \sum_{i=1}^{n} x_i\sin(\sqrt{\lvert x_i\rvert})$	[−500, 500]	0
Schwefel-2	$f(x) = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n} x_i$	[−10, 10]	0
Styblinski	$f(x) = 1/2\sum_{i=0}^{n}(x_i^4 - 16x_i 2 + 5x_i)$	[−5, 5]	−39.16599 n
Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	[−5.12, 5.12]	0

5. Experimental Setup

In the current research study, a newly proposed parent centric crossover (FX) is used to enhance the performance of the genetic process and make a close comparison with existing real-coded crossover operators which comprise of DPX, LX, and SBX. These four crossover operators co-integrated with MPTM, PM, and NUM mutation operators for the evaluation of their global performance. A simulated study of twelve algorithmic combinations along with their respective crossover and mutation probabilities is summarized and final parametric values are represented in Table 2. The suitable adjustment of these parametric values is helpful in obtaining optimum results in the empirical study.

The size of the population for all these algorithms is ten times the number of decision variables and the simulated results are obtained through thirty independent runs for each algorithm. Tournament selection along with Elitism with size one is applied in the whole GA algorithmic process. All experiments are terminated when the number of generations achieved the 500 generations and the optimum results regarding the GA process were obtained through trial run and screening experimentation. To evaluate the efficiency, compatibility, and effectiveness of the simulation process, all algorithms were executed thirty times and the mean and standard deviation along average execution time in seconds are taken as final results. The performance of the newly proposed real-coded crossover scheme linked with logistic probability distribution is evaluated on twenty-one benchmark functions by using MATLAB version R2015a.

In the context of probabilistic search algorithms in the GA process, we applied a one-way analysis of variance (ANOVA) as hypothesis testing [55] for comparison of four real-valued crossover operators including the proposed one. The experimental results seem to be statistically significant if it is considered unlikely to have occurred by chance, assuming the significance of the null hypothesis. The statistically significant results justify the rejection of the null hypothesis when a probability (p value) is less than a prespecified threshold level (5% level of significance). The F test is applied to make a comparing between crossover operators through a ratio of between and within variation. Hence, the statistical significance of the crossover schemes is tested through the F test statistic in

$$F = \frac{\text{Variance between crossover operators}}{\text{Variance within crossover operators}}, \quad (26)$$

$$F = \frac{\text{MS}_{\text{crossover operators}}}{\text{MS}_{\text{error}}}. \quad (27)$$

On the basis of a significant F test, we have determined that crossover operators' means are significantly different from each other. Hence, the pairwise multiple comparisons can determine the difference between each pair of means. The Gabriel pairwise comparison test [56] is used, which is based on the studentized maximum modulus and generally

Table 2: Details about parametric settings for all algorithms.

Crossover operators	Mutation operators	Selection operators	Crossover probability	Mutation probability
DPX	MTPM	Tournament	0.70	0.02
FX	MTPM	Tournament	0.70	0.02
LX	MTPM	Tournament	0.70	0.02
SBX	MTPM	Tournament	0.70	0.02
DPX	PM	Tournament	0.65	0.005
FX	PM	Tournament	0.65	0.005
LX	PM	Tournament	0.65	0.005
SBX	PM	Tournament	0.65	0.005
DPX	NUM	Tournament	0.70	0.01
FX	NUM	Tournament	0.70	0.01
LX	NUM	Tournament	0.70	0.01
SBX	NUM	Tournament	0.70	0.01

more powerful under simulated studies. The test statistic of the Gabriel test is as follows:

$$Gb_{\text{cal}} = \frac{\overline{x}_{\max} - \overline{x}_{\min}}{\sqrt{\text{MS}_{\text{error}}/2\left(1/n_i + 1/n_j\right)}}, \quad (28)$$

where MS_{error} = Mean square error from ANOVA, $\overline{x}_{\max}$ and $\overline{x}_{\min}$ are two means for comparison, and n_i and n_j are respective sample sizes from population i and j.

6. Results and Discussion

In the current empirical study, our main contribution is introducing a new real-coded crossover operator (FX), and the focus of our study is to evaluate the performance of proposed crossover operators in the context of simulation results. So, according to the results of Table 3, we make a comparison of newly proposed (FX) with Double Pareto crossover (DPX) operator [49], Laplace (LX) crossover operator [26], and simulated binary crossover (SBX) [19] by the co-integration with MTPM mutation operator [50] based on mean, standard deviation (SD), and average execution time. It is observed that the Fisk crossover operator outperformed in fifteen out of twenty-one test problems/benchmark functions under diverse features. In most of the multimodal test problems, the empirical results are considered close to the theoretical optimum value which reflects the improved performance of the newly proposed crossover scheme. The performance of FX in terms of mean values, standard deviation, and average execution time is exceptionally ideal and also helpful to overcome the shortcomings of the GA process including exploitation and exploration. Hence, the smallest mean values with lesser SD along with minimal average execution time reflect better control over the selection pressure and have a sufficient impact on loss of population diversity.

The empirical comparison of novel Fisk distribution- (FX-) based crossover operator with Double Pareto crossover (DPX) operator [49], Laplace (LX) crossover operator [26], and simulated binary crossover (SBX) [19] in Table 4 was administered by incorporating the power mutation proposed by Deep and Thakur [26] in the algorithmic framework. The resulted mean and standard deviation with average executing time are lowest in most of the benchmark functions, i.e., Generalized-2, Ackley, Axis, Brown, Ellipsoidal, Levy-Mont, etc. by using the power mutation operator by [26]. Overall, FX performs best in thirteen out of twenty-one benchmark functions which show a complete dominance over the other three crossover operators.

The co-integration of power mutation (PM) with crossover operators is also helpful to enhance the performance and also have better control over the deficiencies of the GA process. The considerable closeness of empirical results with theoretical optimum value delineates better control on the loss of population diversity and the least execution time demonstrates sustainable selection pressure. In general, FX has a better success rate for obtaining the optimum results.

The empirical results of Table 5 reveal an ample dominance of the FX crossover operator over the Double Pareto crossover (DPX) operator [49], Laplace (LX) crossover operator [26], and simulated binary crossover (SBX) [19] algorithmic approaches with co-integration of nonuniform mutation (NUM) operator [51] for obtaining an optimum value. It is also observed that the FX crossover attained optimum results in thirteen benchmark functions, whereas SBX crossover performs better in 8 benchmark functions regarding optimum mean with fewer SD and lesser average execution time under nonuniform mutation operators. The closeness with theoretical value expresses the superlative performance of the newly proposed real-coded crossover scheme. The empirical results in Table 5 also show sustainable control over the weaknesses of the GA process.

According to the results of Table 6 show that there is a highly significant difference between DPX, FX, LX, and SBX under Ellipsoidal and Rastrigin benchmark functions by using ANOVA. Besides, there is also a significant difference in thirteen benchmark functions out of twenty-one. Gabriel multiple pairwise comparison test is applied for close comparison between real-coded crossover operators. In this regard, there is a highly significant difference between FX and DPX in

Table 3: Statistical results with average execution time for real-coded crossover operators under Makinen, Periaux, and Toivanen mutation (MTPM) operator.

Benchmark functions		DPX_MTPM		FX_MTPM		LX_MTPM		SBX-MTPM	
		Statistical measures	Average execution time (sec)	Statistical measures	Average execution time (sec)	Statistical measures	Average execution time (sec)	Statistical measures	Average execution time (sec)
Generalized_1	Mean	0.2313	566.675352	0.0041	246.48479	**0.0027**	268.06119	0.0043	285.407911
	S.D	0.1152		0.0058		0.0048		0.0044	
Generalized_2	Mean	0.6163	408.737251	**0.022**	224.97533	0.0243	241.49021	0.0323	230.671076
	S.D	0.2315		0.0323		0.0297		0.0429	
Ackley	Mean	8.079	375.733917	**1.3087**	206.83753	1.7639	198.06534	2.0945	199.727056
	S.D	0.9474		0.8369		1.2447		1.0181	
Axis	Mean	45.9862	367.017767	**0.5095**	204.02592	0.6617	199.18445	1.0493	200.972761
	S.D	16.5777		0.5655		0.8451		1.5094	
Brown	Mean	0.1441	430.309508	**0.0046**	280.6953	0.0048	276.82815	0.0197	281.898095
	S.D	0.0644		0.006		0.0072		0.0179	
Cigar	Mean	1.20E+07	349.927297	2.24E+05	178.78084	**1.02E+05**	182.02777	2.14E+05	172.44999
	S.D	4.57E+06		3.70E+05		1.29E+05		2.32E+05	
Ellipsoidal	Mean	9.63E+02	430.620612	**6.01E+02**	238.30767	6.21E+02	177.53345	6.17E+02	175.823523
	S.D	1.31E+02		1.51E+02		1.20E+02		8.77E+01	
LevyMont	Mean	6.68E−01	371.496581	**1.86E−02**	219.7111	2.43E−02	195.46777	3.39E−02	196.40783
	S.D	2.64E−01		2.27E−02		3.07E−02		4.36E−02	
Neumaier	Mean	1.50E+05	377.978013	**2.25E+03**	215.68475	2.35E+03	198.76745	3.09E+03	203.739667
	S.D	5.87E+04		4.35E+03		2.76E+03		4.28E+03	
Powersums	Mean	2.47E−20	376.7447	**1.67E−51**	216.90121	2.92E−48	241.85019	6.36E−43	203.469996
	S.D	1.23E−19		7.41E−51		1.60E−47		2.90E−42	
Rastrigin	Mean	2.00E+02	496.604286	**1.71E+01**	203.92601	8.54E+01	222.86203	2.25E+01	182.554729
	S.D	1.64E+01		2.55E+01		6.02E+01		5.47E+00	
Rosenbrok	Mean	1.48E+05	352.491989	7.24E+01	190.13846	**4.22E+01**	184.65858	1.84E+02	174.230471
	S.D	9.40E+04		1.36E+02		4.07E+01		3.22E+02	
Comix	Mean	−1.80E+00	359.558982	**−2.99E+00**	194.00672	−2.98E+00	191.51063	−2.97E+00	187.656013
	S.D	2.42E−01		2.90E−02		2.29E−02		4.20E−02	
Dejong	Mean	3.75E+01	379.64551	1.74E+01	216.62535	1.63E+01	201.89812	**1.45E+01**	207.739321
	S.D	1.73E+01		8.68E+00		8.29E+00		8.84E+00	
Dropwave	Mean	−3.38E−01	392.756396	**−9.15E−01**	201.66234	−8.64E−01	204.18529	−8.87E−01	200.597559
	S.D	8.25E−02		6.02E−02		1.06E−01		9.25E−02	
Matyas	Mean	−1.17E+48	413.750804	−3.22E+47	364.89539	−7.11E+53	299.16512	**−8.82E+53**	279.741252
	S.D	6.38E+48		1.04E+48		2.06E+54		2.00E+54	
Schwefel_1	Mean	−2.04E+04	384.2278	**−7.12E+04**	345.73165	−6.93E+04	290.69073	−7.17E+04	302.46965
	S.D	2.42E+03		1.48E+04		1.03E+04		9.78E+03	
Schwefel_2	Mean	−1.20E+40	412.899538	−1.06E+46	390.8193	−2.28E+46	299.34147	**−8.45E+48**	263.902044
	S.D	4.26E+40		5.79E+46		1.17E+47		1.77E+49	
Sphere	Mean	3.14E+00	415.745488	**2.97E−02**	251.85282	2.99E−02	208.42148	9.89E−02	195.014598
	S.D	1.03E+00		3.40E−02		3.16E−02		8.33E−02	
New function	Mean	2.56E+00	375.682437	**2.23E−02**	201.64129	2.42E−02	215.59478	5.84E−02	216.957449
	S.D	7.68E−01		3.80E−02		4.40E−02		8.27E−02	
Styblin_30	Mean	−1.17E+03	464.240474	**−1.18E+03**	240.64107	−1.17E+03	284.35864	−1.17E+03	234.250931
	S.D	2.77E+00		3.20E−01		3.37E−01		6.34E−01	

two benchmark functions while also have a significant impact on 12 benchmark functions at 5% and 10% significance level respectively. When we compare FX with LX then the differences between these real coded crossover operators are statistically significant at 1%, 5%, and 10% under 15 benchmark functions. Gabriel multiple pairwise comparisons show an insignificant difference between FX and SBX under most of the benchmark functions. The overall statistical results represent the numerical uniqueness of FX over DPX, LX, and SBX crossover operators.

The performance of the algorithmic procedure is usually examined through optimum value and execution time required to get an optimal solution. Four real-coded crossover schemes

Table 4: Statistical results with average execution time for real-coded crossover operators under Power Mutation (PM) operator.

Benchmark functions		DPX_PM		FX_PM		LX_PM		SBX-PM	
		Statistical measures	Average execution time (sec)	Statistical measures	Average execution time (sec)	Statistical measures	Average execution time (sec)	Statistical measures	Average execution time (sec)
Generalized_1	Mean	0.0285	518.721058	0.001	367.863727	**8.45E − 04**	258.531662	0.0012	246.16973
	S.D	0.0108		0.0011		0.0015		0.0012	
Generalized_2	Mean	0.6359	385.276104	**0.0152**	269.044215	0.0158	253.092402	0.0362	267.80063
	S.D	0.2068		0.0238		0.0215		0.0374	
Ackley	Mean	7.467	478.659304	**1.0848**	279.067055	1.8357	242.515118	2.3116	226.51509
	S.D	1.1872		0.9074		0.9342		1.1955	
Axis	Mean	45.8074	425.004108	**0.7551**	222.356894	0.8955	210.639859	1.1532	237.43432
	S.D	16.4635		0.9722		1.3225		1.441	
Brown	Mean	0.1876	441.397977	**0.0067**	315.156832	0.0076	306.27855	0.0114	262.25005
	S.D	0.0758		0.0071		0.0083		4.08E − 03	
Cigar	Mean	1.12E + 07	516.123492	3.98E + 05	2.34E + 05	**1.76E + 05**	219.306763	2.63E + 05	210.98199
	S.D	2.34E + 05		2.98E + 05		3.06E + 05		1.09E + 02	
Ellipsoidal	Mean	8.78E + 02	432.251909	**5.62E + 02**	399.667168	6.12E + 02	346.677706	6.01E + 02	292.1226
	S.D	2.46E − 01		2.23E − 02		1.90E − 02		3.84E − 02	
LevyMont	Mean	5.88E − 01	466.362675	**1.49E − 02**	268.877354	1.51E − 02	260.511992	2.77E − 02	298.59981
	S.D	4.60E + 04		5.54E + 03		2.65E + 03		3.40E + 03	
Neumaier	Mean	1.23E + 05	450.195382	**2.14E + 03**	273.851317	3.18E + 03	277.065924	4.07E + 03	261.22039
	S.D	4.56E − 19		5.03E − 43		3.57E − 42		8.68E − 47	
Powersums	Mean	9.65E − 20	477.403985	9.20E − 44	288.434361	6.52E − 43	291.37676	**1.59E − 47**	276.99917
	S.D	1.85E + 01		6.39E + 00		4.99E + 01		5.14E + 00	
Rastrigin	Mean	1.93E + 02	411.950127	**1.85E + 01**	225.617841	1.11E + 02	328.539715	2.40E + 01	310.21191
	S.D	8.72E + 04		8.47E + 01		4.09E + 01		5.08E + 02	
Rosenbrok	Mean	1.29E + 05	364.809901	6.76E + 01	229.118435	**4.77E + 01**	220.049887	1.99E + 02	285.28473
	S.D	3.03E − 01		1.47E − 01		4.06E − 01		1.04E − 01	
Comix	Mean	−1.83E + 00	389.120664	**−3.40E + 00**	367.165338	−2.53E − 01	210.895496	−2.61E + 00	219.33614
	S.D	1.62E + 01		1.01E + 03		3.55E + 02		8.73E + 00	
Dejong	Mean	4.70E + 01	370.296456	3.49E + 03	230.889114	7.27E + 02	228.377338	**2.37E + 01**	237.55929
	S.D	7.10E − 02		1.07E − 01		1.69E − 01		1.16E − 01	
Dropwave	Mean	−3.16E − 01	389.633546	**−8.55E − 01**	261.323633	−7.99E − 01	211.82917	−8.21E − 01	208.57591
	S.D	3.58E + 48		1.74E + 54		3.24E + 54		2.76E + 54	
Matyas	Mean	−6.56E + 47	465.717775	−8.37E + 53	385.474899	−1.78E + 54	317.153294	**−1.82E + 54**	308.17748
	S.D	4.50E + 03		1.67E + 04		1.70E + 04		4.52E + 03	
Schwefel_1	Mean	−1.98E + 04	447.263322	**−8.94E + 04**	342.631783	−6.72E + 04	360.494851	−7.88E + 04	425.09822
	S.D	1.48E + 43		2.39E + 47		3.78E + 47		9.23E + 50	
Schwefel_2	Mean	**−2.69E + 42**	465.913381	−8.59E + 46	542.754243	−8.29E + 46	325.887074	−2.01E + 50	403.50187
	S.D	1.48E + 43		2.39E + 47		3.78E + 47		9.23E + 50	
Sphere	Mean	4.91E + 00	346.340363	4.35E + 01	220.556922	1.93E + 01	213.039734	**1.46E + 00**	210.44449
	S.D	1.21E + 00		8.36E + 00		3.69E + 00		5.16E − 01	
New function	Mean	2.35E + 00	479.065646	**3.99E − 02**	216.028312	4.32E − 02	237.68754	4.31E − 02	242.30062
	S.D	8.03E − 01		7.48E − 02		7.76E − 02		4.45E − 02	
Styblin_30	Mean	−1.16E + 03	509.537899	**−1.17E + 03**	327.468554	−1.17E + 03	316.823763	−1.17E + 03	331.77431
	S.D	4.65E + 00		4.87E − 01		4.40E − 01		3.89E − 01	

are visually compared with the integration of three mutation operators in Figure 5. Hence, Figure 5 visualized the dominance of FX over other crossover operators. In the MTPM mutation operator, FX obtained the optimum value in 15 benchmark functions while LX and SBX outperformed in three benchmark functions. Similarly, FX distinctly achieved an optimum solution in thirteen benchmark functions on the other hand SBX obtained optimum value in four and three benchmark functions under PM with NUM mutation operators, respectively, while the graphical description also depicted the limited performance of DPX and LX operators in the whole algorithmic procedure.

7. Performance Index (PI)

After statistically examining the performance of FX real-coded crossover operator with others, we make a comparison between GAs' real-coded crossover schemes based on the performance

Table 5: Statistical results with average execution time for real-coded crossover operators under nonuniform (NUM) operator.

Benchmark functions		DPX_NUM		FX_ NUM		LX_ NUM		SBX- NUM	
		Statistical measures	Average execution time (sec)	Statistical measures	Average execution time (sec)	Statistical measures	Average execution time (sec)	Statistical measures	Average execution time (sec)
Generalized_1	Mean	0.5225	478.916752	3.0341	308.5493	$1.45E+00$	280.0871	**0.0487**	255.9563
	S.D	0.2516		0.9671		0.5452		0.0225	
Generalized_2	Mean	0.6291	391.052693	**0.1521**	259.1522	1.4837	243.2173	0.1806	246.4749
	S.D	0.1512		0.0201		0.4423		0.0465	
Ackley	Mean	9.1248	413.71599	**4.9445**	240.5565	12.3698	266.037	5.4122	247.9398
	S.D	0.8869		0.9155		1.4621		0.5585	
Axis	Mean	64.4259	537.250458	**6.9984**	345.104	65.8394	354.8941	18.2763	209.7847
	S.D	20.0796		2.7297		21.6322		6.2694	
Brown	Mean	1.492	456.516034	4.8687	467.5696	4.3687	448.0076	**0.4106**	276.9796
	S.D	0.3779		3.1271		1.7771		0.1248	
Cigar	Mean	$1.76E+07$	343.362341	**3.61E + 06**	$2.19E+02$	$1.61E+07$	217.4726	$4.51E+06$	209.4604
	S.D	$4.03E+06$		$1.34E+06$		$4.57E+06$		$1.50E+06$	
Ellipsoidal	Mean	$9.50E+02$	367.615066	**3.00E + 02**	239.8889	$4.55E+02$	226.9789	$5.82E+02$	208.8561
	S.D	$1.27E+02$		$1.32E+02$		$1.17E+02$		$9.51E+01$	
LevyMont	Mean	$6.44E-01$	346.025872	**1.30E − 01**	212.774	$1.49E+00$	211.1474	$1.75E-01$	211.4208
	S.D	$1.75E-01$		$6.74E-02$		$5.44E-01$		$6.47E-02$	
Neumaier	Mean	$1.34E+05$	339.924576	$5.25E+05$	209.1727	$2.50E+05$	214.5617	**4.98E + 04**	212.3191
	S.D	$4.05E+04$		$1.42E+05$		$5.44E+04$		$1.63E+04$	
Powersums	Mean	$3.01E-17$	374.391225	$7.66E-11$	247.9908	$2.56E-14$	238.4452	**2.33E − 24**	236.0081
	S.D	$1.65E-16$		$2.24E-10$		$4.24E-14$		$1.12E-23$	
Rastrigin	Mean	$2.03E+02$	370.518653	**1.40E + 01**	232.7837	$1.28E+02$	220.9383	$2.32E+01$	218.6061
	S.D	$1.75E+01$		$2.10E+01$		$2.07E+00$		$5.21E+00$	
Rosenbrok	Mean	$2.63E+05$	353.134843	$6.39E+06$	226.0087	$3.85E+05$	243.1898	**4.31E + 04**	217.7927
	S.D	$1.80E+05$		$2.40E+06$		$2.01E+05$		$2.36E+04$	
Comix	Mean	$-1.44E+00$	358.87612	**−2.72E + 00**	221.4712	$-7.35E-01$	216.1618	$-2.63E+00$	214.7786
	S.D	$2.60E-01$		$1.35E-01$		$3.41E-01$		$1.03E-01$	
Dejong	Mean	$4.80E+01$	377.17503	$8.48E+02$	246.0532	$6.43E+01$	237.6296	**1.93E + 01**	245.2073
	S.D	$1.91E+01$		$3.95E+02$		$2.41E+01$		$7.77E+00$	
Dropwave	Mean	$-2.61E-01$	367.673226	**−9.80E − 01**	215.0356	$-1.13E-01$	202.1959	$-4.67E-01$	208.7195
	S.D	$4.47E-02$		$1.16E-02$		$2.48E-02$		$6.93E-02$	
Matyas	Mean	$-2.36E+29$	520.190243	$-4.84E+29$	301.1081	$-4.80E+29$	247.7128	**−4.65E + 29**	223.1851
	S.D	$1.33E+29$		$6.92E+26$		$2.64E+17$		$-4.65E+29$	
Schwefel_1	Mean	$-1.09E+04$	394.179023	**−1.33E + 04**	277.8657	$-1.21E+04$	252.6488	$-1.24E+04$	242.9953
	S.D	$8.80E+02$		$8.15E+02$		$5.83E+02$		$3.24E+01$	
Schwefel_2	Mean	$-1.74E+28$	406.560342	**−6.21E + 29**	316.0881	$-2.03E+28$	203.0264	$-4.73E+29$	238.3114
	S.D	$9.04E+28$		$1.02E+29$		$5.81E+28$		$1.74E+29$	
Sphere	Mean	$4.26E+00$	346.884419	$2.08E+01$	218.3084	$4.10E+00$	213.0483	**1.27E + 00**	205.879
	S.D	$1.19E+00$		$5.63E+00$		$1.44E+00$		$4.09E-01$	
New function	Mean	$3.80E+00$	405.187371	**3.80E − 01**	298.5285	$8.14E+00$	257.0114	$8.99E-01$	217.5631
	S.D	$3.80E+00$		$7.80E-01$		$3.62E+00$		$3.03E-01$	
Styblin_30	Mean	$-1.12E+03$	443.314611	**−7.80E + 03**	288.4661	$-8.30E+02$	217.5405	$-1.17E+03$	213.2046
	S.D	$1.77E+01$		$5.14E+00$		$4.12E+01$		$2.92E+00$	

index (PI) that was used by HaqHussainn and Ahmad [14]. This performance index was precisely applied to examine the behavior of various controlled stochastic search methods. This index is a widely used procedure for making a comparison between different heuristic algorithms [50, 57]. The mathematical derivation of PI is given below in the following equation:

$$PI = \frac{1}{Np}\sum_{i=1}^{Np}\left(\gamma_1\delta_1^i + \gamma_2\delta_2^i + \gamma_3\delta_3^i\right), \tag{29}$$

where

$$\delta_1^i = \frac{MF^i}{LMF^i}, \quad \delta_2^i = \frac{SF^i}{LSF^i}, \quad \delta_1^i = \frac{CVF^i}{LCVF^i}, \quad \text{for } i = 1, 2, \ldots, N_p. \tag{30}$$

Table 6: Statistical results about ANOVA and Gabriel multiple pairwise comparisons of real coded crossover operators.

Benchmark functions	Analysis of variance (ANOVA)		Gabriel multiple pairwise comparison		
	F	p value	FX vs. DPX	FX vs. LX	FX vs. SBX
Generalized_1	0.5650	0.6530	0.9060	0.9800	0.7570
Generalized_2	1.3850	0.3160	0.5340	0.7390	1.0000
Ackley	4.7240	0.0239*	0.0304*	0.0866[a]	0.0897
Axis	3.6680	0.0630*	0.0860[a]	0.0787[a]	0.0987
Brown	0.3980	0.7580	0.9820	1.0000	0.9070
Cigar	3.6350	0.0640*	0.0990[a]	0.0899[a]	0.0967
Ellipsoidal	12.1300	0.0020**	0.0030**	0.0090**	0.0658[a]
LevyMont	4.4330	0.0300*	0.5140	0.7240	0.2360
Neumaier	0.4920	0.6970	0.8765	0.9790	0.8100
Powersums	1.0000	0.4410	0.6570**	0.6570	0.6570
Rastrigin	178.4140	0.0000**	0.0000	0.0000**	0.9660
Rosenbrok	0.9020	0.4810	0.7300	0.7100	0.6650
Comix	3.9740	0.0210*	0.0240*	0.0120*	0.1960
Dejong	3.6090	0.0262*	0.0406*	0.0569[a]	0.0900[a]
Dropwave	3.4520	0.0420*	0.0700[a]	0.0512[a]	0.0890[a]
Matyas	2.9290	0.0364*	0.0274*	0.0120*	0.0960[a]
Schwefel_1	3.0600	0.0418*	0.0553[a]	0.0762[a]	0.7865
Schwefel_2	4.1310	0.0393*	0.0340*	0.0720[a]	0.6030
Sphere	5.6940	0.0245*	0.0528[a]	0.0672[a]	0.9320
Newfunction	3.873	0.0379*	0.0657[a]	0.0790[a]	0.5674
Styblin	3.044	0.0424*	0.0490*	0.0613[a]	0.7821

**significant at 1%, *significant at 5%, and [a]significant at 10%.

MF^i: mean of the objective function for the ith optimization problem

LMF^i: least mean value of objective function obtained by all algorithms for the ith optimization problem

SF^i: standard deviation of the objective function for the ith optimization problem

LSF^i: least standard deviation value of objective function obtained by all algorithms for the ith optimization problem

CVF^i: the value of the coefficient of variation linked with the objective function for the ith optimization problem

$LCVF^i$: least coefficient of variation value of objective function obtained by all algorithms for the ith optimization problem

N_p: the total population to be analyzed

The γ_1, γ_2, and γ_3 ($\gamma_1 + \gamma_2 + \gamma_3 = 1$ and $0 \le \gamma_1, \gamma_2, \gamma_3 \le 1$) are weights assigned to three statistics were considered, respectively.

In regards of the above description, the performance index is a mathematical formulation of γ_1, γ_2, and γ_3, respectively. Hence, $\gamma_1 + \gamma_2 + \gamma_3 = 1$ and one of γ_i, for ($i = 1$, 2, 3), could be excluded by reducing the number of dependent variables from the mathematical formulation of the performance index. However, it is still difficult to visually evaluate the trend of all GAs' real-coded crossover schemes because of the overlapping of the surface plot of PI. So, we adopt the modified process by assigning the same weights to any two terms in PI. Hence, the PI becomes a function of a single variable. Now, resultant cases are below:

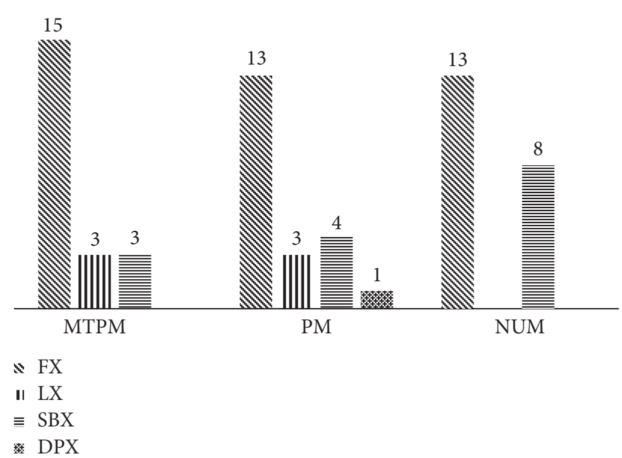

Figure 5: Graphical description of real-coded crossover operators with mutation operators.

$$(\textbf{Case}-1)\ \gamma_1 = wt,\ \gamma_2 = \gamma_3 = \frac{1-Wt}{2}, \text{where } 0 \le wt \le 1,$$
$$(\textbf{Case}-2)\ \gamma_2 = wt,\ \gamma_1 = \gamma_3 = \frac{1-Wt}{2}, \text{where } 0 \le wt \le 1,$$
$$(\textbf{Case}-3)\ \gamma_3 = wt,\ \gamma_1 = \gamma_2 = \frac{1-Wt}{2}, \text{where } 0 \le wt \le 1. \quad (31)$$

The visual representation for cases (1–3) in Figures 6–8 reveals that the horizontal axis represents weights (wt) and scaled value of performance index (PI) defined on the vertical axis. The PI of FX is outperformed

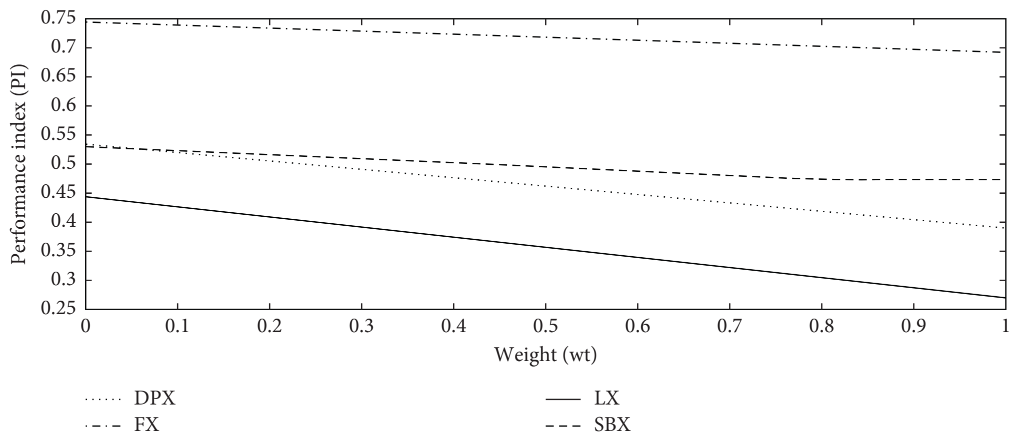

Figure 6: Graphical representation of performance index for real-coded crossover schemes for case 1.

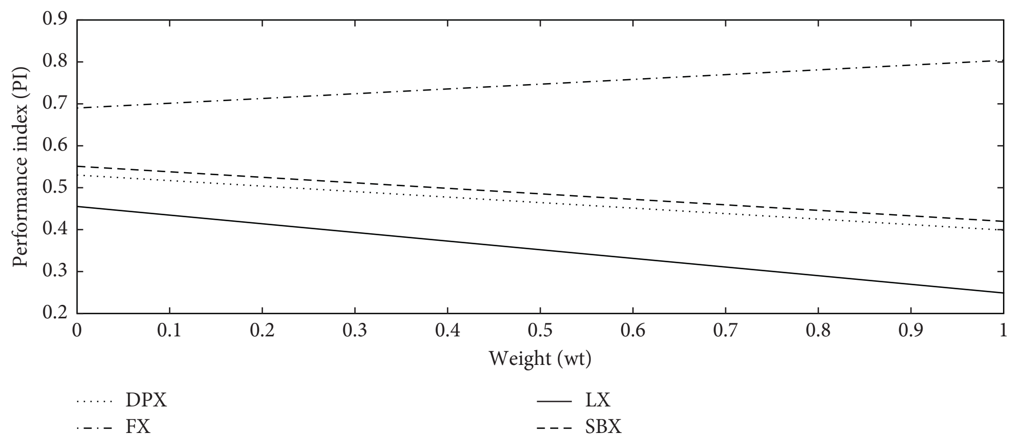

Figure 7: Graphical representation of performance index for real-coded crossover schemes for case 2.

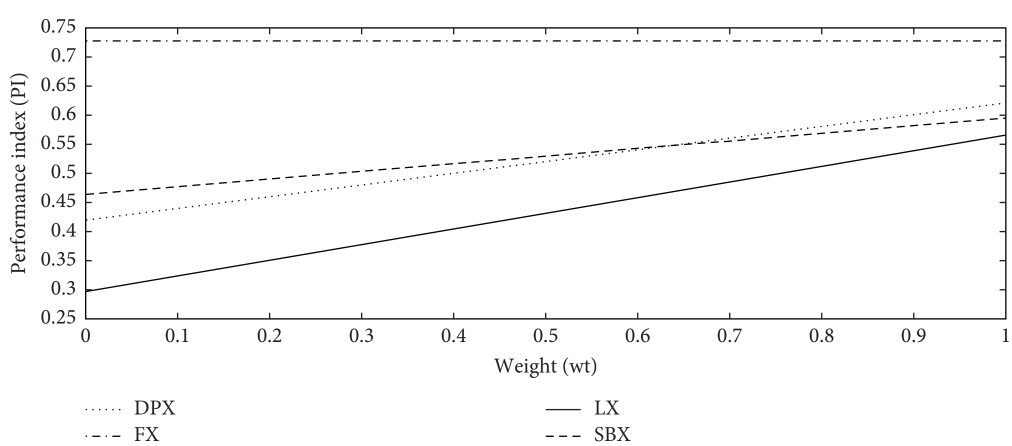

Figure 8: Graphical representation of performance index for real-coded crossover schemes for case 3.

in all figures instead of other parent-centric crossover operators which shows a substantial dominance towards perfection. More specifically, the graphical depiction of PI endorses the performance improvement in the FX crossover operator.

8. Conclusions

In this paper, an improved real-coded crossover operator called the Fisk crossover operator (FX) is introduced to enhance the performance of the GA process with a fine tradeoff between selection pressure and population diversity. This newly proposed parent centric crossover operator has a characteristic of self-adaption and the development ideology of FX is linked with Log-logistic probability distribution. FX and three crossover operators (DPX, LX, and SBX) are attached with three well-known mutation operators (MPTM, PM, and NUM), which are compared, and evaluated their algorithmic performance under twenty-one benchmark functions with diverse features. All considered test problems are scalable with the varied number of decision variables.

The tournament selection is applied in the reproduction phase for the newly proposed algorithm and performance is examined under an identical simulation strategy. Two different strategies are used to evaluate the performance of GA. The first strategy is to make a comparison between FX and three other crossover operators for obtaining optimal solutions on the basis of mean, SD, and average execution time. The empirical results show a complete dominance of FX over other crossover operators. Hence, FX outperformed in between thirteen to fifteen benchmark functions under MTPM, PM, and NUM mutation operators, respectively. Furthermore, relevant statistical techniques including ANOVA and Gabriel pairwise multiple comparison test are also administrated which indicate the significance of the proposed crossover scheme.

The second strategy is to evaluate the performance by comparing all algorithms through performance index (PI) with the use of three statistical measures. The graphical representation of PI reveals the optimal performance of FX instead of other crossover operators. Finally, the statistically significant results of the proposed crossover technique have a definite edge over the others and have great potential to solve more complex optimization problems.

Data Availability

The data used to support the findings of the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The authors are very grateful to the deanship of scientific research at King Khalid University, Abha, Kingdom of Saudi Arabia, for funding this work through the Research Groups Program under the project number R.G.P-2/67/41.

References

[1] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA, USA, 1989.

[2] K. Deb, *Multi-Objective Optimization Using Eevolutionary Algorithms*, John Wiley & Sons, New York, NY, USA, 2001.

[3] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Science & Business Media, Berlin, Germany, 2013.

[4] J. H. Holland, "Adaptation in natural and artificial systems," 1991.

[5] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Eengineering*, vol. 186, no. 2–4, pp. 311–338, 2000.

[6] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*, Elsevier, Berlin, Germany, 2001.

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[8] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Eevolution: A Practical Approach to Global Optimization*, Springer Science & Business Media, Berlin, Germany, 2006.

[9] M. Dorigo, A. Colorni, and V. Maniezzo, "Distributed optimization by ant colonies," 1991.

[10] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.

[11] J. Knox, "Tabu search performance on the symmetric traveling salesman problem," *Computers & Operations Research*, vol. 21, no. 8, pp. 867–876, 1994.

[12] D. E. Goldberg, "Real-coded genetic algorithms, virtual alphabets, and blocking," *Complex Systems*, vol. 5, no. 2, pp. 139–167, 1991.

[13] A. H. Haq and I. M. Almanjahie, "A novel selection approach for genetic algorithms for global optimization of multimodal continuous functions," *Computational Intelligence and Neuroscience*, vol. 5, 2019.

[14] A. H. Hussain and I. Ahmad, "Performance evaluation of novel selection processes through hybridization of k-means clustering and genetic algorithm," *Applied Ecology and Environmental Research*, vol. 17, no. 6, pp. 14159–14177, 2019.

[15] J. L. Jin, X. H. Yang, and J. Ding, "An improved simple genetic algorithm-accelerating genetic algorithm," *Systems Engineering Theory & Practice*, vol. 4, no. 4, pp. 8–13, 2001.

[16] L. Davis, "Handbook of genetic algorithms," 1991.

[17] C. Z. Janikow and Z. Michalewicz, "An experimental comparison of binary and floating-point representations in genetic algorithms," *ICGA*, vol. 4, pp. 31–36, 1991.

[18] A. H. Wright, "Genetic algorithms for real parameter optimization," *Foundations of genetic algorithms*, vol. 1, pp. 205–218, 1991.

[19] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.

[20] F. Herrera, M. Lozano, and J. L. Verdegay, "Dynamic and heuristic fuzzy connectives-based crossover operators for controlling the diversity and convergence of real-coded genetic algorithms," *International Journal of Intelligent Systems*, vol. 11, no. 12, pp. 1013–1040, 1996.

[21] F. Herrera and M. Lozano, "Adaptation of genetic algorithm parameters based on fuzzy logic controllers," *Genetic Algorithms and Soft Computing*, vol. 8, pp. 95–125, 1996.

[22] I. G. Tsoulos, "Modifications of real code genetic algorithm for global optimization," *Applied Mathematics and Computation*, vol. 203, no. 2, pp. 598–607, 2008.

[23] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, "Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm," *International Journal of Machine Learning and Cybernetics*, vol. 20, pp. 1–29, 2019.

[24] Q. Yuan, F. Qian, and W. Du, “A hybrid genetic algorithm with the Baldwin effect,” *Information Sciences*, vol. 180, no. 5, pp. 640–652, 2010.

[25] K. Deb, A. Anand, and D. Joshi, “A computationally efficient evolutionary algorithm for real-parameter optimization,” *Evolutionary Computation*, vol. 10, no. 4, pp. 371–395, 2002.

[26] K. Deep and M. Thakur, “A new crossover operator for real coded genetic algorithms,” *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 895–911, 2007.

[27] K. Miettinen, M. M. Mäkelä, and J. Toivanen, “Numerical comparison of some penalty-based constraint handling techniques in genetic algorithms,” *Journal of Global Optimization*, vol. 27, no. 4, pp. 427–446, 2003.

[28] H. Maaranen, K. Miettinen, and M. M. Mäkelä, “Quasi-random initial population for genetic algorithms,” *Computers & Mathematics with Applications*, vol. 47, no. 12, pp. 1885–1895, 2004.

[29] Z. Michalewicz, T. Logan, and S. Swaminathan, “Evolutionary operators for continuous convex parameter spaces,” in *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, pp. 84–97, New York, NY, USA, 1994.

[30] N. J. Radcliffe, “Equivalence class analysis of genetic algorithms,” *Complex Systems*, vol. 5, no. 2, pp. 183–205, 1991.

[31] H. Mühlenbein and D. Schlierkamp-Voosen, “Predictive models for the breeder genetic algorithm I. Continuous parameter optimization,” *Evolutionary Computation*, vol. 1, no. 1, pp. 25–49, 1993.

[32] L. J. Eshelman and J. D. Schaffer, “Real-coded genetic algorithms and interval-schemata,” *Foundations of Genetic Algorithms*, vol. 2, pp. 187–202, 1993.

[33] H. M. Voigt, J. Born, and I. Santibánez-Koref, “A multivalued eevolutionary algorithm,” 1993.

[34] H. M. Voigt, H. Mühlenbein, and D. Cvetkovic, “Fuzzy recombination for the breeder genetic algorithm,” 1995.

[35] S. Tsutsui, M. Yamamura, and T. Higuchi, “Multi-parent recombination with simplex crossover in real coded genetic algorithms,” in *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation*, vol. 1, pp. 657–664, New York, NY, USA, 1999.

[36] N. Tutkun, “Optimization of multimodal continuous functions using a new crossover for the real-coded genetic algorithms,” *Expert Systems with Applications*, vol. 36, no. 4, pp. 8172–8177, 2009.

[37] P. Kaelo and M. M. Ali, “Integrated crossover rules in real coded genetic algorithms,” *European Journal of Operational Research*, vol. 176, no. 1, pp. 60–76, 2007.

[38] Z. Q. Chen and R. L. Wang, “An efficient real-coded genetic algorithm for real-parameter optimization,” in *Proceedigs of the 2010 Sixth International Conference on Natural Computation*, vol. 5, pp. 2276–2280, IEEE, Berlin, Germany, 2010.

[39] I. Ono and S. Kobayashi, “A real-coded genetic algorithm for function optimization using the unimodal normal distribution,” in *Proceedings of the International Conference on Genetic Algorithms*, pp. 246–253, Berlin, Germany, 1999.

[40] H. Kita, “The multi-parent unimodal normal distribution crossover for real-coded genetic algorithms,” in *Proceedings of the 1999 Congress on Evolutionary Computation*, vol. 2, pp. 1588–1595, London, UK, 1999.

[41] A. Sinha, S. Tiwari, and K. Deb, “A population-based, steady-state procedure for real-parameter optimization,” in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 514–521, IEEE, London, UK, 2005.

[42] R. Chandra, M. Frean, and M. Zhang, “Crossover-based local search in cooperative co-evolutionary feedforward neural networks,” *Applied Soft Computing*, vol. 12, no. 9, pp. 2924–2932, 2012.

[43] K. K. Bali, R. Chandra, and M. N. Omidvar, “Competitive island-based cooperative coevolution for efficient optimization of large-scale fully-separable continuous functions,” in *Proceedings of the International Conference on Neural Information Processing*, pp. 137–147, New York, NY, USA, 2015.

[44] L. Rolland and R. Chandra, “The forward kinematics of the 6-6 parallel manipulator using an evolutionary algorithm based on generalized generation gap with parent-centric crossover,” *Robotica*, vol. 34, no. 1, p. 1, 2016.

[45] S. H. Ling and F. H. F. Leung, “An improved genetic algorithm with average-bound crossover and wavelet mutation operations,” *Soft Computing*, vol. 11, no. 1, pp. 7–31, 2007.

[46] F. Herrera, M. Lozano, and A. M. Senchez, “Hybrid crossover operators for real-coded genetic algorithms: an experimental study,” *Soft Computing*, vol. 9, no. 4, pp. 280–298, 2005.

[47] O. Takahashi and S. Kobayashi, “An adaptive neighboring search using crossover-like mutation for multimodal function optimization,” in *Proceedings of the 2001 IEEE International Conference on Systems, Man and Cybernetics e-Systems and e-Man for Cybernetics in Cyberspace*, vol. 1, pp. 261–267, IEEE, London, UK, 2001.

[48] S. N. Sivanandam and S. N. Deepa, *An Introduction to Genetic Algorithms*, Springer, Berlin, Heidelberg, 2008.

[49] M. Thakur, “A new genetic algorithm for global optimization of multimodal continuous functions,” *Journal of Computational Science*, vol. 5, no. 2, pp. 298–311, 2014.

[50] R. A. E. Mäkinen, J. Périaux, and J. Toivanen, “Multidisciplinary shape optimization in aerodynamics and electromagnetics using genetic algorithms,” *International Journal for Numerical Methods in Fluids*, vol. 30, no. 2, pp. 149–159, 1999.

[51] Z. Michalewicz, D. Dasgupta, and R. G. Le Riche, “Evolutionary algorithms for constrained engineering problems,” *Computers & Industrial Engineering*, vol. 30, no. 4, pp. 851–870, 1996.

[52] A. Schoenauer, J.-A. Fernández-Madrigal, and A. Cruz-Martín, “Log-logistic modeling of sensory flow delays in networked telerobots,” *IEEE Sensors Journal*, vol. 13, no. 8, pp. 2944–2953, 2013.

[53] F. Ashkar and S. Mahdi, “Fitting the log-logistic distribution by generalized moments,” *Journal of Hydrology*, vol. 328, no. 3-4, pp. 694–703, 2006.

[54] M. Jamil and X. S. Yang, “A literature survey of benchmark functions for global optimization problems,” 2013.

[55] I. Rojas, J. González, H. Pomares, J. J. Merelo, P. A. Castillo, and G. Romero, “Statistical analysis of the main parameters involved in the design of a genetic algorithm,” *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 32, no. 1, pp. 31–37, 2002.

[56] K. R. Gabriel, “A simple method of multiple comparisons of means,” *Journal of the American Statistical Association*, vol. 73, no. 364, pp. 724–729, 1978.

[57] C. Mohan and H. T. Nguyen, “A controlled random search technique incorporating the simulated annealing concept for solving integer and mixed-integer global optimization problems,” *Computational Optimization and Applications*, vol. 14, no. 1, pp. 103–132, 1999.