Mathematics

AIMS Mathematics, 9(10): 29250–29268.
DOI: 10.3934/math.20241418
Received: 02 August 2024
Revised: 16 September 2024
Accepted: 26 September 2024
Published: 15 October 2024

https://www.aimspress.com/journal/Math

Research article

Enhancing probabilistic based real-coded crossover genetic algorithms with authentication of VIKOR multi-criteria optimization method

Jalal-ud-Din[1,*], Ehtasham-ul-Haq[2], Ibrahim M. Almanjahie[3] and Ishfaq Ahmad[4]

[1] Department of Mathematics and Statistics, International Islamic University, 44000 Islamabad, Pakistan, Email: jalaluddin.phdst18@iiu.edu.pk
[2] Department of Mathematics and Statistics, Faculty of Basic and Applied Sciences, International Islamic University, 44000 Islamabad, Pakistan, Email: ehtasham.malik@iiu.edu.pk
[3] Department of Mathematics, College of Science, King Khalid University, Abha 62223, Saudi Arabia, Email: imalmanjahi@kku.edu.sa
[4] Department of Mathematics and Statistics, Faculty of Basic and Applied Sciences, International Islamic University, 44000 Islamabad, Pakistan, Email: Ishfaq.ahmad@iiu.edu.pk

* **Correspondence:** Email: jalaluddin.phdst18@iiu.edu.pk.

Abstract: To improve the performance of genetic algorithms (GAs) in complex optimization settings, this work offered two novel real-coded crossover operators: one based on the Gumbel distribution (GX) and the other on the Rayleigh distribution (RX). These innovative operators, when combined with three different mutation techniques, created a significant improvement in GA methodology. Our meticulous simulations showed that the GX operator significantly outperformed RX and other traditional operators, demonstrating its superior capacity to address complex optimization problems. The GX operator's unusual robustness was further validated through detailed performance analysis utilizing the VlseKriterijuska Optimizacija I Komoromisno Resenje (VIKOR) multi-criteria decision-making technique, setting a new standard in crossover operator design and significantly improving the state of the art in GAs.

Keywords: genetic algorithms; Gumbel crossover; Rayleigh crossover; global optimization; real-coded crossover operator; VIKOR method
Mathematics Subject Classification: 60-xx, 68W50

1. **Introduction**

In numerous broad fields of study, optimization is a useful strategy when choosing the most beneficial choice from a range of options is essential. These broad fields include computer science, applied mathematics, engineering, and management science, as discussed by Deb [1], Goldberg [2], Michalewicz, and Arabas [3]. Effective crossover operators are essential for solving complicated problems with genetic algorithms (GA), especially for addressing information technology (IT) outsourcing schedule concerns. Traditional crossover approaches frequently struggle to balance exploration and exploitation F Lu [4]. Complex systems, such as IT outsourcing timetables, have become increasingly difficult to manage in an era of globalization due to inherent risks and uncertainties. Just as supply chain optimization entails resolving multi-echelon difficulties and decreasing costs using advanced heuristics, improving IT outsourcing schedules necessitates sophisticated methodologies to efficiently manage numerous risks Nahangi and Awwad [5]. In computing and engineering, the aim is to maximize system or application performance while minimizing runtime and resource consumption as much as possible Chu and Beasley [6]. Decisions are made by creating optimization models containing the problem's core and then applying mathematical approaches to address these models Wasserkrug et al [7]. Optimization algorithms for unconstrained issues typically employ gradient information to locate the optimal solution. As such, the gradient-based optimization method can solve objective functions with non-differentiable components Ahmadianfar et al. [8]. Deterministic approaches are a type of local optimization where the search process and its outcome primarily depend on an initial guess. Several population-based stochastic techniques, including particle swarm optimization (PSO), simulated annealing (SA), GAs, and others, have been created and are employed to address optimization issues with constraints, as discussed in Eberhart et al. [9], Kirkpatrick et al. [10] and Deb [3].

The guided random search approaches comprise all of these optimization techniques Goldberg and Deb [11]. GAs are based on Charles Darwin's principle that only the offspring of the fittest parents can survive Holland [12]. GA is a reliable and effective evolutionary search technique for locating the best potential solutions to complicated multi-modal situations De-Jong [13]. Natural phenomena suggest that genetic inheritance is stored in chromosomes composed of genes Haq et al. [14]. As the mutation operator aids in preserving population diversity and preventing premature convergence, the crossover operator uses genetic information from different chromosomes to explore new search spaces Haq et al. [15]. The continuous search space is changed into a discrete one using a binary-coded scheme, where the string length is determined by the separation between two adjacent grids. Under a small number of decision variables, binary encoding performs well and requires less precision for the solutions Katoch et al. [16]. However, when high precision is needed to solve multi-dimensional optimization problems, binary encoding schemes perform unsatisfactorily Haq et al. [15]. The concept of real encoding first surfaced in the early 1990s, when a vector of real-coded GA was used to represent a chromosome Wright [17].

The crossover and mutation operators both have a big effect on how well GAs perform. As a result, a lot of research is focused on improving these operators' performance. Laplace crossover (LX) is used to locate the offspring and is linked with a Laplace probability distribution Deep et al. [18]. Therefore, the two offspring generated by the LX operator are symmetrical in terms of their parental position and did not automatically locate close to the better of the two parents. To create a simulated binary crossover (SBX), Deb et al. [19] modified a single-point binary crossover. When two parents are chosen, SBX produces two offspring. These offspring are positioned in a straight line that is connected to their parents. The main drawback of SBX is that it is unable to control the size of the parameter value adaptively. Essentially, the crossover operator uses the current population information to guide the search in the other search space. In this context, the crossover operator is essential to exploring the unique aspect of GA Naqvi et al. [20].

2. **Crossover operators used in the previous studies**

2.1. LX

At first, Deep and Thakur [18] proposed a self-parent-centric crossover operator based on the Laplace distribution. This is the Laplace distribution's distribution function:

$$F(y) = \begin{cases} \frac{1}{2} exp\left(\frac{|y-a|}{b}\right), & y \leq a, \\ 1 - \frac{1}{2} exp\left(-\frac{|y-a|}{b}\right), & y > a, \end{cases} \tag{1}$$

where, respectively, a and b represent the location and shape parameters for Laplace distribution. Utilizing LX, from two parents $y^{(1)} = \left(y_1^{(1)}, y_2^{(1)}, y_3^{(1)}, \ldots, y_n^{(1)}\right)$ and $y^{(2)} = \left(y_1^{(2)}, y_2^{(2)}, y_3^{(2)}, \ldots, y_n^{(2)}\right)$, two offspring, ϑ = (ϑ1, ϑ2, ϑ3,…, ϑn) and τ = (τ1, τ2, τ3,..., τn.

2.2. Double Pareto crossover (DPX)

Based on the double Pareto probability distribution; the double Pareto crossover (DPX) is a parent-centric operator Thakur [21]. The distribution function of the double Pareto distribution, which this crossover operator uses, is given as

$$F(y) = \begin{cases} \frac{1}{2}\left(1 - \frac{y}{ab}\right)^{-a}, & y < 0 \\ \frac{1}{2}\left[1 - \left(1 + \frac{y}{ab}\right)^{-a}\right], & y \geq 0 \end{cases}. \tag{2}$$

The Double Pareto probability distribution has two parameters, i.e., a & b, where a belongs to a real number and b is greater than zero. Here a = location parameter and b = scale parameter of the double Pareto probability distribution.

2.3. Fisk crossover (FX)

The Fisk crossover (FX) is a parent-centric crossover operator using a log-logistic distribution ul Haq et al. [22]. The FX operator uses the cumulative distribution function of the log-logistics distribution, as shown below:

$$F(y) = \begin{cases} \frac{1}{1+(y|\beta)^{-\alpha}}, & y < 0 \\ 1 - \frac{1}{1+(y|\beta)^{-\alpha}}, & y \geq 0 \end{cases}, \tag{3}$$

where β > 0 and α > 0 are scale and shape parameters respectively.

2.4. Logistic crossover (LogX)

This specific operator is proposed by Naqvi et al. [20], which is based on logistic distribution. The cumulative distribution function (CDF) of the logistic distribution is given as

$$F(y) = \begin{cases} \frac{1}{1+e^{-\alpha\frac{-(y-\alpha)}{s}}}, & y < 0 \\ 1 - \frac{1}{1+e^{-\alpha\frac{-(y-\alpha)}{s}}}, & y \geq 0 \end{cases}, \tag{4}$$

where α > 0 and s > 0 are location and scale parameters respectively.

2.5. SBX

The SBX is a real-coded crossover operator, which was first proposed by Deb and Agarwal [19]. The binary transformation to continuous search space is one of its special features. Following are the steps for generating offspring, $\vartheta = (\vartheta 1, \vartheta 2, \vartheta 3, \ldots, \vartheta n)$ by two parents $y^{(1)} = \left(y_1^{(1)}, y_2^{(1)}, y_3^{(1)}, \ldots, y_n^{(1)}\right)$ and $y^{(2)} = \left(y_1^{(2)}, y_2^{(2)}, y_3^{(2)}, \ldots, y_n^{(2)}\right)$ are as follows:

1st step: Generate a random number εi between zero and one.

2nd step: Then, obtain a parameter β_i as:

$$\beta_i = \begin{cases} (2\varepsilon_i)^{\frac{1}{(n_c+1)}}, & if\ \varepsilon_i \leq \frac{1}{2} \\ \frac{1}{(2-2\varepsilon_i)^{\frac{1}{(n_c+1)}}}, & if\ \varepsilon_i > \frac{1}{2} \end{cases}, \tag{5}$$

where the distribution index is denoted by n_c and $n_c \in [0, \infty]$.

Thus, both parents $y^{(1)} = \left(y_1^{(1)}, y_2^{(1)}, y_3^{(1)}, \ldots, y_n^{(1)}\right)$ and $y^{(2)} = \left(y_1^{(2)}, y_2^{(2)}, y_3^{(2)}, \ldots, y_n^{(2)}\right)$, and an offspring $\vartheta = (\vartheta 1, \vartheta 2, \vartheta 3, \ldots, \vartheta n)$ is produced in the following Eq (6):

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

3. Proposed real-coded crossover operators

By taking a balanced approach to exploration and exploitation, the GX (Gumbel-based) and RX (Rayleigh-based) crossover operators will improve the performance of GAs. The Gumbel distribution, which is well-known for modeling extreme values, is utilized by the GX operator to enhance the algorithm's exploratory power. The GX operator lets the algorithm escape local optima and fully search the solution space by producing offspring with features that can differ greatly from the parent population. This makes the method especially useful in complicated or misleading landscapes. Conversely, the RX operator, derived from the Rayleigh distribution, highlights moderate deviations and helps refine the search in areas of potential interest, hence improving exploitation.

3.1. First proposed real-coded crossover operator based on Gumbel distribution

Extreme value distributions are frequently modeled by using the Gumbel distribution, especially when a set of random variables has maximum or minimum values Gumbel [23]. The

Gumbel distribution is appropriate for modeling extreme events because it exhibits heavy tails, and a bell-shaped, double-exponential probability function describes it. The maximum or minimum of an objective function is a common problem in optimization. Modeling the distribution of extreme values with the Gumbel distribution gives statistical insight into and ability to predict rare but important events. So in optimization theory, Gumbel distribution plays an important role. It is favored for capturing extreme values in extremely complex and multimodal scenarios where extreme events are extremely important Kamel et al. [24]. The decision is based on the optimization problem and the particular features of the data. The GX operator has been proposed here. The density function of the Gumbel distribution is as follows:

$$f(y) = \frac{1}{\beta} e^{\left[-\frac{y-\mu}{\beta} - e^{\left(-\frac{y-\mu}{\beta}\right)}\right]}, \tag{7}$$

where the location parameter is μ and the scale parameter is β.

The cumulative distribution function of Gumbel distribution is as follows:

$$F(y) = e^{-e^{-\left(\frac{y-\mu}{\beta}\right)}}. \tag{8}$$

Main steps for generating two offspring, $\vartheta = (\vartheta 1, \vartheta 2, \vartheta 3, \ldots, \vartheta n)$ and $\tau = (\tau 1, \tau 2, \tau 3, \ldots, \tau n)$ by two parents $y^{(1)} = \left(y_1^{(1)}, y_2^{(1)}, y_3^{(1)}, \ldots, y_n^{(1)}\right)$ and $y^{(2)} = \left(y_1^{(2)}, y_2^{(2)}, y_3^{(2)}, \ldots, y_n^{(2)}\right)$, are as follows:

1st step: Random number ε_i is generated in between zero and one.

2nd step: The Gumbel distribution function is inverted to obtain the parameter βi, which is obtained by equating the randomly generated number ε to the area under the curve from $-\infty$ to βi.

$$F(y) = e^{-e^{-\left(\frac{y-\mu}{\beta}\right)}}, \tag{9}$$

$$\varepsilon = e^{-e^{-\left(\frac{\beta_i-\mu}{\beta}\right)}}, \tag{10}$$

$$\log(\varepsilon) = -e^{-\left(\frac{\beta_i-\mu}{\beta}\right)}, \tag{11}$$

$$\log[-\log(\varepsilon)] = -\left(\frac{\beta_i-\mu}{\beta}\right), \tag{12}$$

$$\beta_i = \mu - \beta * \left[\log[-\log(\varepsilon)]\right]. \tag{13}$$

3rd step: The generation of offspring is based on the following Eqs (14) and (15):

$$\vartheta_i = \frac{(y_i^{\{1\}} + y_i^{\{2\}}) + \beta_i \left|y_i^{\{1\}} - y_i^{\{2\}}\right|}{2} \tag{14}$$

and

$$\tau_i = \frac{(y_i^{\{1\}} + y_i^{\{2\}}) - \beta_i \left|y_i^{\{1\}} - y_i^{\{2\}}\right|}{2}. \tag{15}$$

3.2. Second proposed real-coded crossover operator based on Rayleigh distribution

A continuous probability distribution for random variables with nonnegative values is called the Rayleigh distribution. The distribution of a two-dimensional vector's magnitude is modeled using the Rayleigh distribution Grimmett & Stirzaker [25]. Rayleigh distribution describes the distribution of vector magnitudes rather than being specifically made for modeling extremes. In optimization theory, the Rayleigh distribution is important, especially in cases where squared magnitudes or the magnitudes of two-dimensional vector components are essential. Depending on the unique characteristics of the optimization problem and the type of data being modeled, the Rayleigh or Gumbel distributions can be very important in optimization theory. Each of the distributions is useful in various situations and has advantages of its own. The crossover operator RX, which is based on Rayleigh distribution, has also been proposed in this study. The density function of the Rayleigh distribution is as follows:

$$f(y) = \frac{y}{\sigma^2} e^{-\frac{y^2}{2\sigma^2}}, \tag{16}$$

where the distribution's scale parameter is $\boldsymbol{\sigma}$.

The cumulative distribution function of Rayleigh distribution is as follows:

$$F(y) = 1 - e^{-\frac{y^2}{2\sigma^2}}. \tag{17}$$

Prerequisite steps for generating two offspring, $\vartheta = (\vartheta 1, \vartheta 2, \vartheta 3, \ldots, \vartheta n)$ and $\tau = (\tau 1, \tau 2, \tau 3, \ldots, \tau n)$ by two parents $y^{(1)} = \left(y_1^{(1)}, y_2^{(1)}, y_3^{(1)}, \ldots, y_n^{(1)}\right)$ and $y^{(2)} = \left(y_1^{(2)}, y_2^{(2)}, y_3^{(2)}, \ldots, y_n^{(2)}\right)$, are as follows:

1st step: Begin with generating random number ε_i in between zero and one.

2nd step: Calculating a parameter β_i, which follows Rayleigh distribution, by inverting the distribution function of Rayleigh distribution as follows:

$$F(y) = 1 - e^{-\frac{y^2}{2\sigma^2}}, \tag{18}$$

$$\varepsilon = 1 - e^{-\frac{\beta_i{}^2}{2\sigma^2}}, \tag{19}$$

$$\log(1 - \varepsilon) = -\frac{\beta_i{}^2}{2\sigma^2}, \tag{20}$$

$$\beta_i = \sigma * [2\log(1 - \varepsilon)]^{\frac{1}{2}} \ . \tag{21}$$

3rd step: Offspring are generated by using the following Eqs (22) and (23):

$$\vartheta_i = \frac{(y_i^{\{1\}} + y_i^{\{2\}}) + \beta_i \left| y_i^{\{1\}} - y_i^{\{2\}} \right|}{2} \tag{22}$$

and

$$\tau_i = \frac{(y_i^{\{1\}} + y_i^{\{2\}}) - \beta_i \left| y_i^{\{1\}} - y_i^{\{2\}} \right|}{2}. \tag{23}$$

4. Experimental setup

A set of benchmark test problems has been used to assess the performance of novel crossover operators. The two novel parent-centric crossovers GX and RX improve genetic process performance that is closely compared to considered real-coded operators, such as LX, DPX, and SBX. Along with nonuniform mutation (NUM), Makinen, Periaux, and Toivanen mutation (MPTM), and power mutation (PM), these seven crossover operators (LX, DPX, SBX, FX, LogX, RX, and GX) have been utilized for evaluating the global optimal performance. The population size has been set to be three hundred and thirty independent runs performed to obtain the simulated results. The selection criteria used by all GAs is tournament selection. Size-one elitism refers to the idea that the most esteemed individuals are retained in the present generation. Final results are considered in terms of mean values, standard deviation, and average execution time. The algorithm stops after five hundred generations. Trial runs and screening experimentation have produced the most optimal outcome for the GA process. Final parametric values are displayed in Figure 1, which summarizes a simulated analysis of fifteen algorithmic combinations and their corresponding crossover and mutation probabilities.

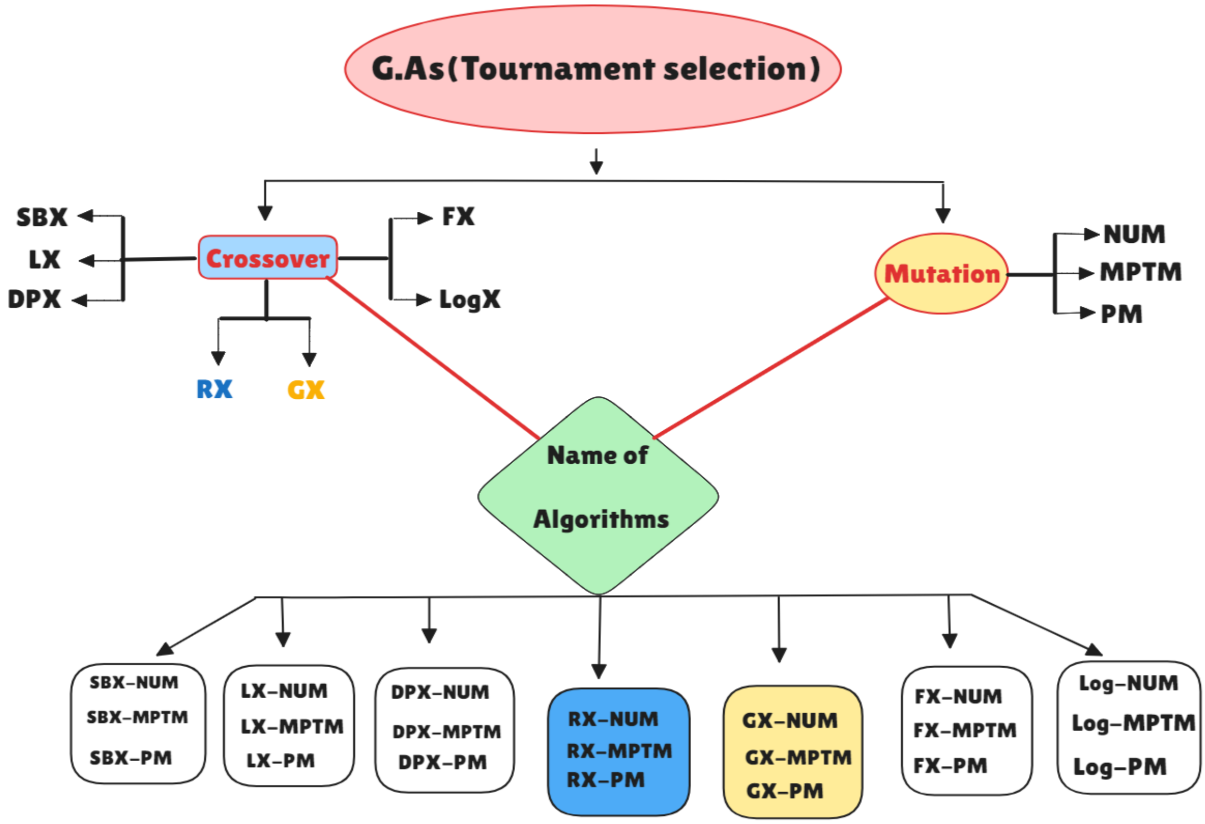

Figure 1. Visual framework of different operators used in the study.

4.1. Test problems

Benchmark functions are authentic tools used to assess the effectiveness of real-coded algorithms in optimization problems. In this study, we take a set of fifteen well-known benchmark functions with different complexity levels. This set of benchmark functions also has different levels of multimodality. The search technique that efficiently eliminates local optima and continues its journey to find global optima is considered an efficient search technique for optimization problems because it does not stick at local optima Mahajan et al. [26]. Table 1 details fifteen benchmark functions used in this study to judge the efficiency of proposed evolutionary methods.

Table 1. Detail of test problems.

Sr. #	Test problem	Objective function	Limits
1	Levy and Montalvo-2 function	$\min_x f(x) = 0.1(sin^2\,(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2[1 + sin^2\,(3\pi x_{i+1})] + (x_n - 1)^2[1 + sin^2(2\pi x_n)])$	{-5, 5}
2	Neumair function	$\min_x f(x) = \sum_i^n (x_i\text{-}1)^2 + \sum_{i=1}^{n-1} x_i\, x_{i-1}$	{-n^2, n^2}
3	Griewank function	$\min_x f(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\,(\frac{x_i}{\sqrt{i}})$	{-10, 10}
4	Brown3 function	$\min_x f(x) = \sum_{i=1}^{n-1}[\,(x_i^2)\text{^}(x_i^2\text{+}1) + (x_i\text{+}1^2)\text{^}(x_{i+1}^2\text{+}1)]$	{-1, 4}
5	Ellipsoidal function	$\min_x f(x) = \sum_{i=1}^{n}(x_i - i)^2$	{-n, n}
6	Cigar function	$\min_x f(x) = x_1^2 + 10000000\sum_{i=2}^{n} x_i^2$	{-10, 10}
7	Axis Parallel hyper ellipsoid function	$\min_x f(x) = \sum_{i=1}^{n} i\, x_i^2$	{-5.12, 5.12}
8	Ackley function	$\min_x f(x) = -20\, e^{-0.2\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}}} - e^{\frac{\sum_i^n \cos(2\pi x_i)}{n}} + 20 + \mathrm{e}$	{-30, 30}
9	Rosenbrock function	$\min_x f(x) = \sum_{i=1}^{n}[100\,(x_{i+1} - x_i^2) + (x_i - 1)^2]$	{-30, 30}
10	New function	$\min_x f(x) = (\frac{\pi}{n})\ 10\sin^2(\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2\,[1 + 10\sin^2(\pi x_i + 1)] + (x_n - 1)^2$	{-10, 10}
11	C01	$Min\, f(x) = \sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2$, z=x-0 g(x) = $\sum_{i=1}^{D}\{z_i^2 - 5000\cos(0.1\pi\, z_i) - 4000\} \leq 0$	xϵ{-100,100}^{D}
12	C02	$Min\, f(x) = \sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2$, z=x-0, y=M*z g(x) = $\sum_{i=1}^{D}\{y_i^2 - 5000\cos(0.1\pi\, y_i) - 4000\} \leq 0$	xϵ{-100,100}^{D}
13	C03	$Min\, f(x) = \sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2$, z=x-0 g(x) = $\sum_{i=1}^{D}\{z_i^2 - 5000\cos(0.1\pi\, z_i) - 4000\} \leq 0$ h(x) = $-\sum_{i=1}^{D} z_i \sin(0.1\pi z_i) = 0$	xϵ{-100,100}^{D}
14	C04	$Min\, f(x) = \sum_{i=1}^{D}\{z_i^2 - 10\cos(2\pi\, z_i) + 10\}$,z=x-0 $g_1(x) = -\sum_{i=1}^{D} z_i \sin\,(2z_i) \leq 0$ $g_2(x) = -\sum_{i=1}^{D} z_i \sin\,(z_i) \leq 0$	xϵ{-10,10}^{D}
15	C05	$Min\, f(x) = \sum_{i=1}^{D}(\,100(z_i^2 - z_{i+1})^2 - (z_i - 1)^2)$,z =x-0, y = M_1*z, w = M_2*z $g_1(x) = \sum_{i=1}^{D}\{y_i^2 - 50\cos(2\pi\, y_i) - 40\} \leq 0$ $g_2(x) = \sum_{i=1}^{D}\{w_i^2 - 50\cos(2\pi\, w_i) - 40\} \leq 0$	xϵ{-10,10}^{D}

5. Results and discussion

Our primary contribution to this research effort is the introduction of two novel real-coded crossover operators such as GX and RX. The main objective is to assess the proposed crossover operators' performance in light of the simulation results. As GX-NUM, RX-NUM, GX-MPTM, RX-MPTM, GX-PM, and RX-PM are the proposed operators that are compared to other crossover operators, such as LX-NUM, DPX-NUM, SBX-NUM, FX-NUM, LogX-NUM LX-MPTM, DPX-MPTM, SBX-MPTM, FX-MPTM, LogX-MPTM, LX-PM, DPX-PM, SBX-PM, FX-PM, and LogX-PM.

Table 2. Results for real-coded crossover operators under NUM operator.

Benchmark functions	Statistics	LX-NUM	DPX-NUM	SBX-NUM	RX-NUM	GX-NUM	FX-NUM	LogX-NUM
	Mean	0.00014	0.000087	0.0087	0.00013	0.0000557	0.377	0.000814
LevyMont	SD	0.00011	0.0000559	0.0033	0.000061	0.0000375	0.1015	0.000376
	Time	107.712624	83.698256	78.64462	85.495419	126.712031	127.4357	205.604031
	Mean	8846.9	3539.3	35852	10210	5980.5	610000	8701.3
Neumair	SD	4935.7	1248.4	13880	6067.3	3736.1	168000	4459.3
	Time	181.629503	130.1451	141.17727	91.065393	126.993684	87.9393	207.780145
	Mean	0.0024	0.0016	0.2108	0.0037	0.0016	1.0103	0.0198
Griewank	SD	0.0017	0.000841	0.0652	0.0035	0.0033	0.0301	0.0084
	Time	81.457873	82.578837	80.650737	91.321104	125.847594	137.349	150.965613
	Mean	0.004	0.0033	0.3586	0.0064	0.0014	1580000	0.0382
Brown	SD	0.003	0.0022	0.1069	0.0102	0.000874	4740000	0.0137
	Time	166.842948	341.123536	165.873626	155.008723	276.27197	178.4728	188.646094
	Mean	3.3488	8.9061	591.2497	8.4815	3.2852	629.444	79.9827
Ellipsoidal	SD	1.9276	3.038	92.7802	4.889	3.1216	129.164	33.5738
	Time	96.062947	96.814591	88.736142	125.60401	122.285082	155.3865	146.618962
	Mean	30023	27657	4612800	36789	12376	84000000	398740
Cigar	SD	22476	11481	1324100	21110	9082.3	23079000	218480
	Time	106.53924	87.859938	181.107231	89.925312	91.538608	165.17616	134.638317
	Mean	0.1682	0.2142	17.8474	0.2078	0.0636	286.565	1.3887
Axis	SD	0.1884	0.1156	5.3991	0.1938	0.0463	60.3644	0.8416
	Time	76.02274	119.661781	89.444963	127.6703	95.830265	178.415314	131.44787
	Mean	3.9109	4.1854	5.4239	3.4013	3.1305	15.4012	3.9752
Ackley	SD	0.9224	0.7202	0.548	0.6378	0.7961	0.8268	0.8615
	Time	137.023015	96.202567	80.377248	131.106759	97.026432	103.415305	111.027993
	Mean	462.872	372.023	33105	484.198	330.1029	7020000	87.2238
Rosenbrock	SD	283.913	189.3726	16914	228.506	181.1197	2900000	5425.4
	Time	127.652415	75.60977	75.820732	121.291021	74.967119	82.75117	56140
	Mean	1.6991	0.6085	12.2297	1.1294	0.5286	285.369	129.454239
Newfunc	SD	2.1336	0.662	3.8633	1.2487	0.671	63.8885	3.4262
	Time	89.410394	90.788672	120.692209	87.656918	86.152792	127.3678	129.454239
	Mean	56648	55988	55724	56450	52836	54708	56140
C01	SD	5587.1	5544.5	4672.1	4952.3	5160.3	5318	5425.4
	Time	92.524	97.2278	95.3304	85.2393	112.855	120.235	87.2238
	Mean	56517	56260	57668	54954	54649	56583	56217
C02	SD	5902	4713.6	5189.1	6172.8	4855	6067.4	5874.8
	Time	111.096	114.066	127.537	96.6299	144.089	92.7783	111.117
	Mean	54512	56250	57072	56927	53875	56375	57188
C03	SD	4990.4	4679.4	3841.3	5095	4661.5	4960.3	5078.8
	Time	104.291	87.1485	105.457	88.2495	113.392	77.6518	134.316
	Mean	190.419	191.344	189.738	186.129	192.465	187.933	194.472
C04	SD	11.8573	14.8278	18.3649	15.5929	10.7845	15.5167	12.3093
	Time	101.72	95.2251	105.39	122.696	133.913	96.3571	94.5518
	Mean	57207	57123	57054	56962	55723	57351	57011
C05	SD	4797.4	5436.8	4645.6	4301.5	5239.4	6195.3	6341.4
	Time	79.2145	84.4875	83.0697	93.0925	70.9508	107.106	102.261

Table 3. Results for real-coded crossover operators under MPTM operator.

Benchmark functions	Statistics	LX-NUM	DPX-NUM	SBX-NUM	RX-NUM	GX-NUM	FX-NUM	LogX-NUM
	Mean	0.000021	0.000018	0.0011	0.000035	9.86E-06	0.0013	0.00027297
LevyMont	SD	0.000036	0.0000228	0.00076	0.000049	0.0000103	0.0026	0.0001875
	Time	102.36276	104.78486	102.9611	108.5233	96.859274	113.1815	215.58379
	Mean	540.05	892.727	3059.3	936.661	484.8504	2730	1781.6
Neumair	SD	666.967	1056.1	2636.2	1271.1	842.9453	3530	1233.7
	Time	82.433744	92.002206	87.96502	84.6824	153.12932	138.1825	144.79124
	Mean	0.00033	0.00032	0.0135	0.00077	0.000251	0.0068	0.003
Griewank	SD	0.00071	0.00061	0.0166	0.0018	0.00043	0.0061	0.002
	Time	94.864538	93.857005	87.56846	83.61188	78.106502	80.57535	103.5461
	Mean	0.00018	0.00036	0.0153	0.00068	0.000183	0.0046	0.0029
Brown	SD	0.00021	0.000359	0.0093	0.0009	0.000249	0.0051	0.002
	Time	161.53716	189.79598	131.5321	211.5596	131.20721	165.9176	159.56679
	Mean	4.9484	8.0844	616.77	9.7785	3.3263	887.3165	87.6828
Ellipsoidal	SD	4.6375	6.2504	88.5754	8.2256	4.9136	178.292	25.1313
	Time	92.944851	97.119153	89.22244	114.6436	91.098581	196.0281	137.74008
	Mean	4213.4	3641	261720	9597.2	3664.7	171170	54229
Cigar	SD	4299.1	3481.1	351380	18292	7043.9	204670	39044
	Time	72.341756	75.668512	75.01477	82.10584	76.012442	183.34413	149.30796
	Mean	0.0146	0.0182	1.1316	0.0595	0.007	0.7543	0.2631
Axis	SD	0.0207	0.018	1.4478	0.1935	0.0086	1.2248	0.2826
	Time	78.913791	98.06796	75.77991	103.4081	163.50531	103.3628	129.63321
	Mean	1.0033	1.4194	2.0116	0.8519	1.0428	2.2011	1.2607
Ackley	SD	1.0086	1.4095	1.2068	0.8057	1.0207	0.9361	0.7002
	Time	84.841513	77.599839	76.55104	101.5414	79.835305	175.81324	162.0749
	Mean	21.3398	30.3796	77.6897	28.9044	26.7186	37.9367	133.5898
Rosenbrock	SD	43.5137	32.7939	91.6669	25.1465	30.9383	40.0273	154.0104
	Time	75.457313	76.27202	74.91132	95.12053	75.433864	81.01464	133.62013
	Mean	0.1065	0.0101	1.8897	0.148	0.0194	1.0756	0.5968
Newfunc	SD	0.5688	0.0117	2.7021	0.4442	0.0909	2.9493	1.6975
	Time	85.437091	91.872155	83.41255	91.01216	87.218181	95.16292	132.18987
	Mean	56707	55594	55073	56904	56334	55767	56016
C01	SD	4854.4	5396.6	6041.8	4364.4	4246.7	5265.7	6602.3
	Time	108.809	89.4791	101.773	101.017	164.761	92.5819	106.041
	Mean	56580	53596	56727	55552	55575	56129	54544
C02	SD	4696.4	5711.1	5959.8	5404.5	4622.2	6283.8	6550.7
	Time	106.335	127.417	103.112	119.922	106.425	100.304	115.498
	Mean	56653	55562	55815	55986	54708	56841	55989
C03	SD	4381.7	5404.4	7171.9	6356.9	4178.6	4765.8	5389.3
	Time	86.4351	85.5492	79.178	63.6457	63.5677	88.315	91.4974
	Mean	186.933	188.32	190.307	189.828	185.832	186.731	190.316
C04	SD	12.637	14.3215	19.8155	14.583	12.7689	17.5531	10.8582
	Time	121.466	103.941	146.528	122.123	135.113	98.9032	152.392
	Mean	57657	59380	56877	57588	56805	56952	57101
C05	SD	4773.1	6000.9	5475	5109.5	7164.5	4838.1	6639.3
	Time	114.003	119.092	102.944	123.649	110.352	176.427	108.11

Table 4. Results for real-coded crossover operators under the PM operator.

Benchmark functions	Statistics	LX-NUM	DPX-NUM	SBX-NUM	RX-NUM	GX-NUM	FX-NUM	LogX-NUM
	Mean	0.00029188	0.00044951	0.01	0.00053266	0.00012793	0.5802	0.00075139
LevyMont	SD	0.00028392	0.00024785	0.0036	0.00047348	0.00011685	0.1658	0.00040303
	Time	84.624551	87.889195	84.79806	96.266321	128.015248	78.136	114.231003
	Mean	17660	5829.3	48169	17680	12722	1100000	10913
Neumair	SD	10570	1994.4	17482	9224.2	10172	262000	4147.2
	Time	76.387883	77.273559	74.559053	80.738701	93.808228	132.1672	115.832336
	Mean	0.011	0.0076	0.22	0.0133	0.005	1.0365	0.0166
Griewank	SD	0.0104	0.0032	0.0643	0.0106	0.0086	0.0114	0.0074
	Time	77.444941	79.57467	73.455623	84.317306	108.64987	102.2386	95.016283
	Mean	0.0171	0.0182	0.4092	0.0275	0.0067	150000000	0.0369
Brown	SD	0.0157	0.015	0.1433	0.0297	0.0039	216000000	0.0216
	Time	118.097805	130.299253	141.748471	202.023482	138.965081	152.3581	156.981702
	Mean	12.8189	27.0527	582.962	17.1602	6.7218	894.8881	37.4548
Ellipsoidal	SD	9.5745	12.8489	87.0768	10.9878	8.4446	183.4455	14.0708
	Time	81.374196	87.473469	83.054335	107.340013	83.439545	179.9395	102.88153
	Mean	146650	122430	5178300	207810	56580	160000000	330020
Cigar	SD	120740	91644	1861900	147350	48021	30300000	171040
	Time	103.470988	102.063039	94.859803	69.272894	89.969564	146.1098	117.330069
	Mean	0.8665	0.5906	19.1424	0.7692	0.2216	561.8476	1.2993
Axis	SD	0.862	0.2811	6.9989	0.5294	0.1483	115.6398	0.6751
	Time	80.632743	70.974762	71.756398	91.537621	67.093254	85.751672	133.283326
	Mean	4.585	4.6023	5.7192	4.3573	3.7683	16.2949	4.3009
Ackley	SD	0.6772	0.6189	0.765	0.7281	0.8069	0.8198	0.7688
	Time	73.641746	74.941498	71.834179	124.518549	71.533324	96.644492	134.60261
	Mean	2133	795.5493	55823	3162.6	1138.9	26300000	2607.7
Rosenbrock	SD	1351.1	540.5518	39463	5062.1	1618.7	10900000	1524
	Time	87.658367	70.738393	69.854953	86.866803	112.408897	74.95865	112.957973
	Mean	2.8305	1.8898	15.9042	2.4014	1.385	430.0245	3.8883
Newfunc	SD	2.8807	1.7127	4.9163	2.2697	1.3594	83.8202	3.2206
	Time	83.868899	97.802163	73.286121	130.844641	150.45537	85.07991	119.062673
	Mean	54985	56467	56475	57416	54458	55304	55516
C01	SD	6121.2	5738.8	5533.9	4805.2	5416.3	5941.9	5384.8
	Time	106.927099	64.585727	64.741079	65.951868	76.611812	73.597283	65.504372
	Mean	55247	55788	56889	56668	53980	57005	54505
C02	SD	5603.3	5843.2	6943.1	3946	4933.3	5285.3	5023.1
	Time	78.033709	75.670182	68.064948	74.584962	97.248741	66.907757	97.391049
	Mean	55459	55087	55942	57314	54092	55747	56519
C03	SD	6990	6642.3	5305.5	5372.2	5346.1	5669.7	5530.9
	Time	97.438107	92.050652	107.485536	62.965519	75.8755	55.902367	72.503046
	Mean	189.4933	194.6566	192.249	194.8476	189.024	189.3816	191.455
C04	SD	13.3131	12.6488	12.2808	15.641	10.129	15.8216	11.9106
	Time	65.09012	71.011633	88.869256	63.046407	59.316266	67.058653	82.14324
	Mean	56232	55251	56948	59842	57325	58569	54250
C05	SD	7084.4	6620	5632.7	5197.8	4800.2	5381.2	5936.7
	Time	81.343016	51.506021	68.586363	81.39364	77.857029	75.532882	72.242855

Based on the results presented in Table 2, GX-NUM performs extremely efficiently against almost all other real-coded operators having the least mean value in all benchmark functions except ‘Neumair’. The empirical results also indicate that other novel operators i.e. RX-NUM did not perform well. This suggests that GX-NUM has a clear-cut dominant capacity for handling selection pressure and population diversity as compared to RX-NUM and other considered operators. According to Table 3, the results show that GX-MPTM performs distinctly as compared to RX-MPTM and other considered operators under many benchmark functions. Similarly from Table 4, GX-PM can also efficiently handle the selection pressure and population diversity. This all suggests that the Gumbel distribution outperformed in terms of efficiently handling selection pressure, and preservation of population diversity than the Rayleigh distribution in the search for global maxima.

6. Performance index

The behavior of several controlled stochastic search techniques was carefully examined by using the performance index (PI) in Figures 2–4. It is an approach that is frequently used to compare various heuristic algorithms Ul Haq et al. [22]. The equation that follows provides the mathematical derivation of PI:

$$PI = \frac{1}{N_q} \sum_{i=1}^{N_q} (\vartheta_1 \eta_1^i + \vartheta_2 \eta_2^i + \vartheta_3 \eta_3^i), \tag{24}$$

where

$$\vartheta_1 \eta_1^i = \frac{M^i}{m^i}, \ \vartheta_2 \eta_2^i = \frac{S^i}{s^i} \text{ and } \vartheta_3 \eta_3^i = \frac{C^i}{c^i} \qquad \text{for } i = 1,2,3,\ldots,N_q.$$

Three statistics were taken into consideration, with weights as ϑ_1, ϑ_2, and ϑ_3 respectively ($\sum_{i=1}^{3} \vartheta_i = 1$ and $0 \leq \vartheta_i \geq 1$). Hence PI is the function of ϑ_i, for making a clear visualization of all seven algorithms and avoiding overlapping. Two terms in PI expression are given equal weights at a time, thus PI becomes a function of a single variable. Following are the three resulting cases:

1) $\vartheta_1 = weight\ (w),\ \vartheta_2 = \vartheta_3 = 0.5(1 - weight(w)),$
2) $\vartheta_2 = weight\ (w),\ \vartheta_1 = \vartheta_3 = 0.5(1 - weight(w)),$
3) $\vartheta_3 = weight\ (w),\ \vartheta_1 = \vartheta_2 = 0.5(1 - weight(w)),$

where for all cases $0 \leq weight(w) \leq 1$.

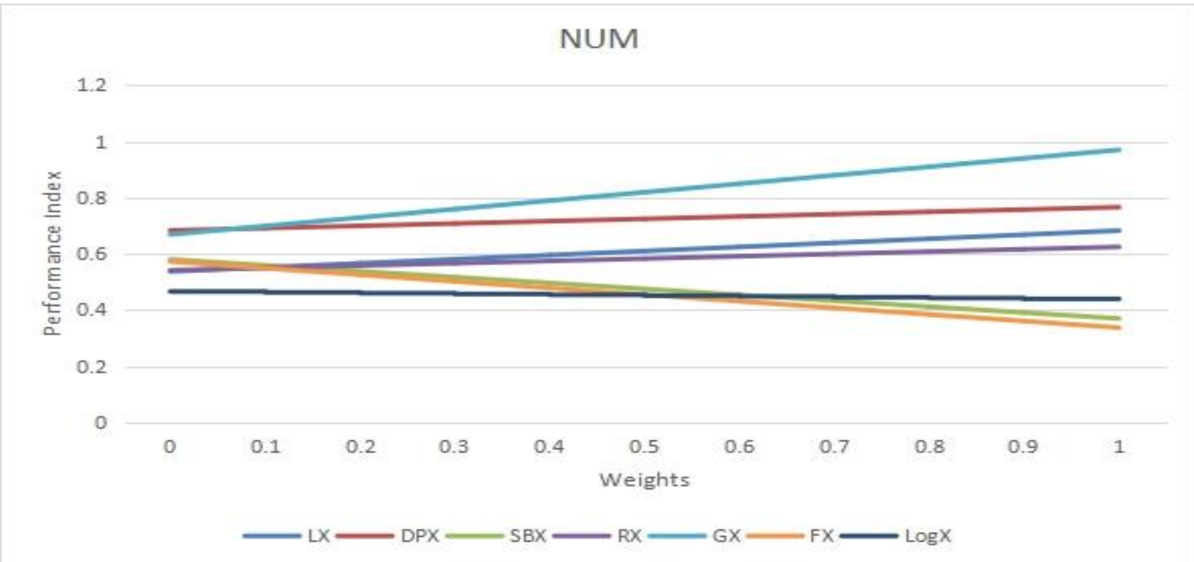

Figure 2. PI graphically represented for real-coded crossover schemes in the first case.

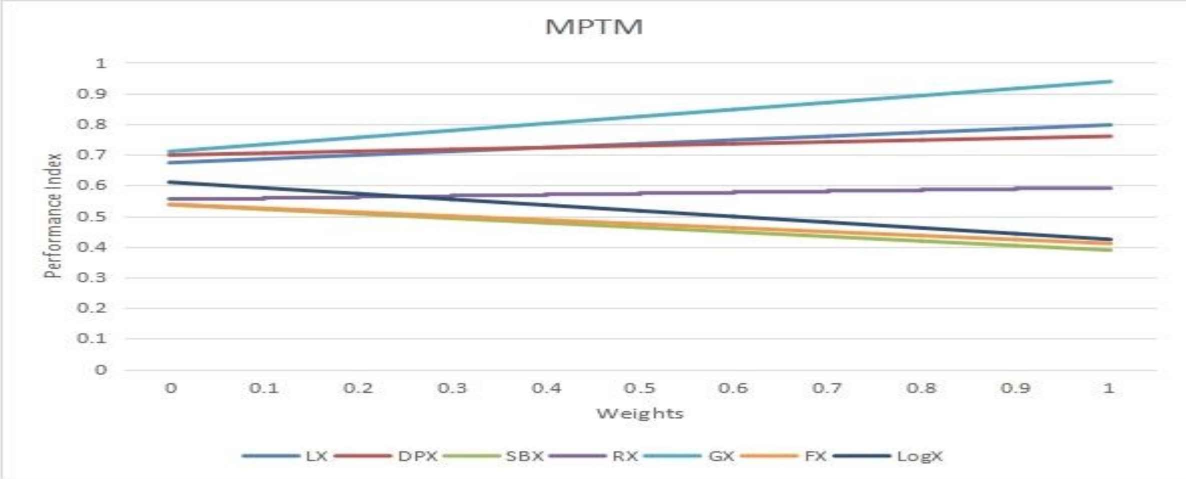

Figure 3. PI graphically represented for real-coded crossover schemes in the second case.

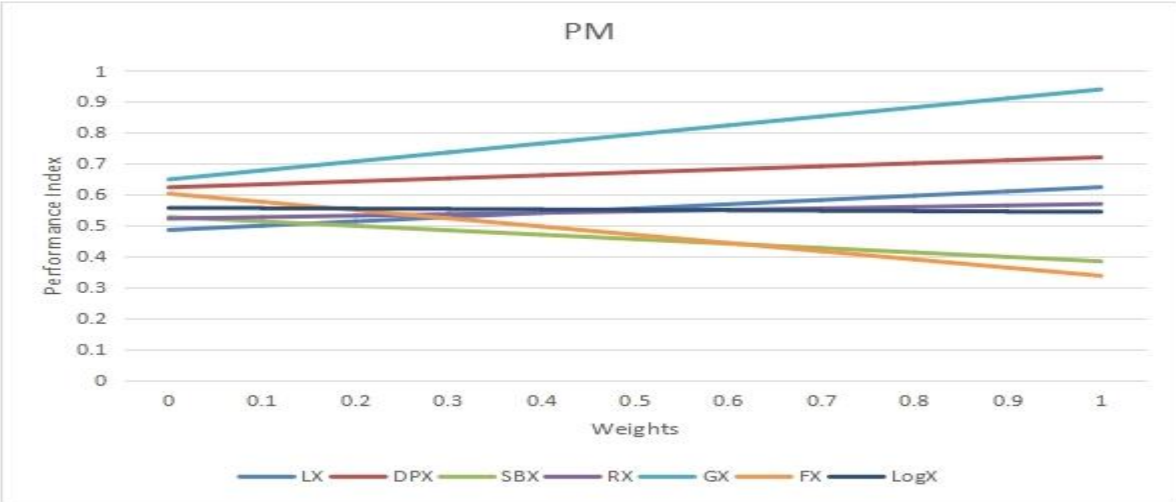

Figure 4. PI graphically represented for real-coded crossover schemes in the third case.

When a line chart exhibits a consistent upward trending line above all other real-coded operators, it indicates in the form of a plotted metric line, and the corresponding crossover operator is outperforming the others. Here in Figures 2–4, we can observe that the line of novel crossover operator GX is initially below the line of considered crossover operator DPX, but it continuously rises and outperforms all. This increasing trend indicates a convergence toward the global solution for the proposed crossover operator (GX). The novel crossover operator (GX) allows for finding global solutions more successfully as the algorithm develops its search. Examine the rate at which each operator converges. It is possible to argue that the novel crossover (GX) is more efficient because it produces better results more quickly. Thus, in the context of the obtained optimum mean values in fifteen benchmark functions, seven real-coded crossover operators and three mutation operators are visually compared in Figure 5.

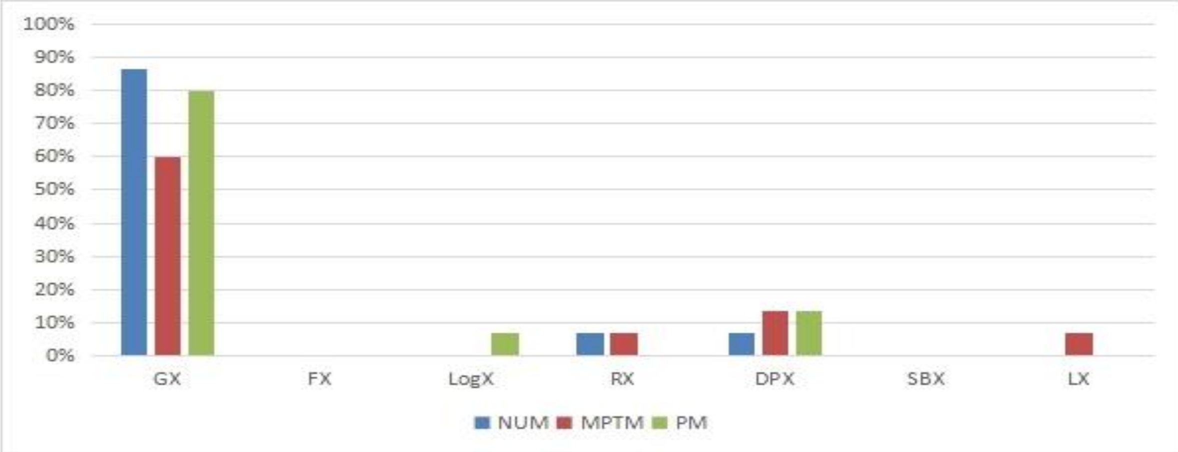

Figure 5. Visual representation of crossovers with various mutation operators regarding benchmark functions.

The first proposed crossover operator (GX) shows considerable dominance with 87% in NUM, 60% in MPTM, and 80% in PM mutation. But in the same visual representation, the second novel crossover operator (RX) shows limited performance.

7. Multi-criteria decision-making technique

The process of finding and selecting the best option from multiple options by considering the decision maker's expectations is known as decision-making. The diversity of benchmarks used to evaluate the solutions makes the decision-making process the most challenging. Therefore, the term multi-criteria decision-making (MCDM) describes decision-making when faced with several, frequently opposing criteria.

7.1. VIKOR method

Several strategies for MCDM, including VlseKriterijuska Optimizacija I Komoromisno Resenje (VIKOR) mean "multi-criteria optimization and compromise solution". Opricovic developed the main VIKOR research in a 1979 PhD dissertation and subsequently in an application in 1980 Mardani et al. [27], Zheng and Wang [28]. The VIKOR approach is necessary to create the appropriate evaluation or decision matrix, which displays how well the crossover operators perform on several test problems. Let, X_{ij} represent the performance measure of the i^{th} alternative (crossover operators) for the j^{th} criterion (test problems). The Lp-metric used as an aggregating function in a compromise programming technique, is then utilized to build the multi-criteria measure for compromise ranking Zeleny [29].

$$L_{p,i} = \left\{ \sum_{j=1}^{M} [w_j \left(\frac{[(X_{ij})_{max} - X_{ij}]}{[(X_{ij})_{max} - (X_{ij})_{min}]} \right)]^p \right\}^p \qquad i = 1, 2, \ldots, N;\ 1<p<\infty, \qquad (25)$$

where w_j denotes weights for j^{th} criteria, M is the number of criteria (test problems), and N is the number of alternatives (crossover operators). Applying the VIKOR approach, values of sum (Si) and maximum row (Ri) are first calculated for each crossover operator that is considered for a v = 0.5. The appropriate values of least quantity (Qi)are subsequently determined in Tables 5, 7, and 9 for each case of mutations (NUM, MPTM, and PM), respectively.

Furthermore, Tables 6 and 8 demonstrate that as the v values change, the rankings of the crossover operators (alternatives) with the highest and lowest rankings remain unchanged, and moderate changes in the intermediate ranking order in Table 6 are observed. In Table 10, variation is observed in the best-ranked positions, but GX crossover operators hold the best ranking position.

Table 5. Ranking of crossover operators (alternatives) in the VIKOR method (in case of NUM mutation).

Crossover operators (Alternatives)	Si	Ri	Qi	Rank
LX-NUM	0.37346	0.1	0.04365	4
DPX-NUM	0.42009	0.086	0.03548	3
SBX-NUM	1.42196	0.62594	0.57167	6
RX-NUM	0.30299	0.09481	0.03369	2
GX-NUM	0.07863	0.07594	0	1
FX-NUM	6.40965	0.6666	1	7
LogX-NUM	0.57215	0.1	0.05934	5

Table 6. Variations in rankings with different values of "v" in the VIKOR method (in case of NUM mutation).

Crossover operators (Alternatives)	v=0.1	v=0.2	v=0.3	v=0.4	v=0.5	v=0.6	v=0.7	v=0.8	v=0.9
LX-NUM	0.04131 (4)	0.0419 (4)	0.04248 (4)	0.04306 (4)	0.04365 (4)	0.04423 (4)	0.04481 (4)	0.0454 (4)	0.0459 (3)
DPX-NUM	0.02071 (2)	0.0244 (2)	0.02809 (2)	0.03178 (2)	0.03548 (3)	0.03917 (3)	0.04286 (3)	0.04655 (3)	0.0502 (4)
SBX-NUM	0.85926 (6)	0.78736 (6)	0.71547 (6)	0.64357 (6)	0.57167 (6)	0.49977 (6)	0.42788 (6)	0.35598 (6)	0.2841 (6)
RX-NUM	0.03228 (3)	0.03263 (3)	0.03299 (3)	0.03334 (3)	0.03369 (2)	0.03404 (2)	0.03439 (2)	0.03474 (2)	0.0350 (2)
GX-NUM	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)
FX-NUM	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)
LogX-NUM	0.04445 (5)	0.04817 (5)	0.0519 (5)	0.05562 (5)	0.05934 (5)	0.06306 (5)	0.06679 (5)	0.07051 (5)	0.07423 (5)

Table 7. Ranking of crossover operators (alternatives) in the VIKOR method (in case of MPTM mutation).

Crossover operators (Alternatives)	Si	Ri	Qi	Rank
LX-MPTM	0.44436	0.09531	0.01816	2
DPX-MPTM	0.68949	0.28039	0.20272	4
SBX-MPTM	5.62095	0.6666	1	7
RX-MPTM	0.62367	0.11699	0.05382	3
GX-MPTM	0.25889	0.09432	0	1
FX-MPTM	4.3809	0.6666	0.88437	6
LogX-MPTM	2.22905	0.6666	0.68371	5

Table 8. Variations in rankings with different values of "v" in the VIKOR method (in case of MPTM mutation).

Crossover operators (Alternatives)	v=0.1	v=0.2	v=0.3	v=0.4	v=0.5	v=0.6	v=0.7	v=0.8	v=0.9
LX-MPTM	0.00501 (2)	0.0083 (2)	0.01158 (2)	0.0149 (2)	0.01816 (2)	0.02144 (2)	0.02473 (2)	0.02802 (2)	0.0313 (2)
DPX-MPTM	0.30065 (4)	0.2762 (4)	0.25168 (4)	0.2272 (4)	0.20272 (4)	0.17824 (4)	0.15375 (4)	0.12927 (4)	0.10479 (4)
SBX-MPTM	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)
RX-MPTM	0.04245 (3)	0.0453 (3)	0.04814 (3)	0.0509 (3)	0.05382 (3)	0.05666 (3)	0.0595 (3)	0.06235 (3)	0.06519 (3)
GX-MPTM	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)	0 (1)
FX-MPTM	0.97687 (6)	0.9538 (6)	0.93062 (6)	0.9075 (6)	0.88437 (6)	0.86124 (6)	0.83812 (6)	0.81499 (6)	0.79186 (6)
LogX-MPTM	0.93674 (5)	0.8735 (5)	0.81023 (5)	0.7469(5)	0.68371(5)	0.62045(5)	0.5572 (5)	0.49394 (5)	0.43068 (5)

Table 9. Ranking of crossover operators (alternatives) in the VIKOR method (in case of PM mutation).

Crossover operators (Alternatives)	Si	Ri	Qi	Rank
LX-PM	0.20644	0.04346	0.01168	2
DPX-PM	0.33571	0.09672	0.06467	4
SBX-PM	1.08619	0.43249	0.39361	6
RX-PM	0.54265	0.1	0.08371	5
GX-PM	0.0592	0.05499	0.00925	1
FX-PM	6.36274	0.6666	1	7
LogX-PM	0.23595	0.07533	0.03959	3

Table 10. Variations in rankings with different values of "v" in the VIKOR method (in case of PM mutation).

Crossover operators (Alternatives)	v=0.1	v=0.2	v=0.3	v=0.4	v=0.5	v=0.6	v=0.7	v=0.8	v=0.9
LX-PM	0.0023 (1)	0.0048 (1)	0.0070 (1)	0.0093 (1)	0.0117 (2)	0.0140 (2)	0.0164 (2)	0.0187 (2)	0.0210 (2)
DPX-PM	0.0813 (4)	0.0772 (4)	0.0729 (4)	0.0688 (4)	0.0647 (4)	0.0605 (4)	0.0564 (4)	0.0522 (4)	0.0480 (4)
SBX-PM	0.5782 (6)	0.5320 (6)	0.4859 (6)	0.4398 (6)	0.3936 (6)	0.3475 (6)	0.3013 (6)	0.2552 (6)	0.20906 (6)
RX-PM	0.0893 (5)	0.0879 (5)	0.0865 (5)	0.0851 (5)	0.0837 (5)	0.0823 (5)	0.0809 (5)	0.0795 (5)	0.0781 (5)
GX-PM	0.0167 (2)	0.0148 (2)	0.0129 (2)	0.0111 (2)	0.0093 (1)	0.0074 (1)	0.0056 (1)	0.0037 (1)	0.0019 (1)
FX-PM	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)	1 (7)
LogX-PM	0.0488 (3)	0.0465 (3)	0.0442 (3)	0.0419 (3)	0.0396 (3)	0.0373 (3)	0.0349 (3)	0.0327 (3)	0.0304 (3)

8. Conclusions

This study introduces two novel crossover operators, the GX operator and the RX operator. In comparison with three existing real-coded algorithms (LX, DPX, and SBX), the performance of GX and RX are evaluated to assess their effectiveness. Furthermore, six new algorithmic combinations, GX-NUM, GX-MPTM, GX-PM, RX-NUM, RX-MPTM, and RX-PM, are proposed by integrating GX and RX with three mutation operators (NUM, MPTM, and PM).

In terms of algorithmic procedures, tournament selection is employed during the reproduction phase, while a simulation-based approach is utilized to analyze the efficacy of the algorithms. A comprehensive evaluation uses fifteen benchmark functions sourced from existing literature to authenticate the performance of the novel algorithms introduced in this study. The comparison metrics encompass mean value, standard deviation, and execution time (measured in seconds) to gauge the efficiency of each algorithm.

Empirical findings, graphical representations of performance indices, and the MCDC VIKOR method indicate that GX outperforms RX and other operators. Notably, the real-coded crossover operator GX enhances the performance of the GA by fine-balancing population diversity and selection pressure. Consequently, GX exhibits significant potential in tackling increasingly complex optimization challenges compared to the existing real-coded operators. Moreover, future work should concentrate on several important areas, including evaluating these operators in practical settings, investigating their efficacy in dynamic and multi-objective scenarios, and extending their applicability to other optimization procedures. Furthermore, it's critical to improve the operators' effectiveness and adaptability across a range of problem kinds. However, this study has certain limitations, such as the scope of the benchmark functions utilized and the need for a broader variety of performance indicators. Addressing these constraints and introducing more evaluation criteria will result in a more comprehensive understanding of the operators' capabilities, paving the way for future breakthroughs in optimization techniques.

Author contributions

Jalal-ud-Din: Conceptualization, writing original draft, writing and editing, formal analysis, software; Ehtasham-ul-Haq: Conceptualization, investigation, methodology, supervision; Ibrahim M. Almanjahie: Investigation, resources; Ishfaq Ahmad: Data curation, formal analysis, validation. All authors have read and approved the final version of the manuscript for publication.

Data availability

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

Acknowledgments

The authors thank and extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through a Large Research Project under grant number RGP2/338/45.

Conflict of interest

The authors declare that they have no conflicts of interest.

Reference

1. K. Deb, Multi-objective optimization using evolutionary algorithms: An introduction, In: *Multi-objective evolutionary optimisation for product design and manufacturing*, London: Springer, 2011, 3–34. https://doi.org/10.1007/978-0-85729-652-8_1
2. D. E. Goldberg, *Genetic algorithms in search, optimization, machine learning*, Boston: Addison-Wesley, 1989.
3. Z. Michalewicz, J. Arabas, Genetic algorithms for the 0/1 knapsack problem, In: *Methodologies for intelligent systems*, Berlin: Springer, **869** (1994), 134–143. https://doi.org/10.1007/3-540-58495-1_14
4. F. Lu, T. Yan, H. Bi, M. Feng, S. Wang, M. Huang, A bilevel whale optimization algorithm for risk management scheduling of information technology projects considering outsourcing, *Knowl. Based Syst.*, **235** (2022), 107600. https://doi.org/10.1016/j.knosys.2021.107600
5. A. Nahangi, M. Awwad, A tabu search heuristic for the outsourcing risk management problem in multi-echelon supply chains, *Int. J. Oper. Res.*, **50** (2024), 490–520. https://doi.org/10.1504/IJOR.2024.140483
6. P. C. Chu, J. E. Beasley, *A genetic algorithm for the set partitioning problem*, London: Imperial College, 1995.
7. S. Wasserkrug, O. Davidovich, E. Shindin, D. Subramanian, P. Ram, P. Mural, et al., Ensuring the quality of optimization solutions in data generated optimization models, In: *IJCAI Decision science for optimization workshop*, 2021.
8. I. Ahmadianfar, O. Bozorg-Haddad, X. Chu, Gradient-based optimizer: A new metaheuristic optimization algorithm, *Inform. Sci.*, **540** (2020), 131–159. https://doi.org/10.1016/j.ins.2020.06.037
9. R. C. Eberhart, Y. Shi, J. Kennedy, *Swarm intelligence (Morgan Kaufmann series in evolutionary computation)*, Morgan Kaufmann Publishers, 2001,
10. S. Kirkpatrick, C. D. Gelatt Jr, M. P. Vecchi, Optimization by simulated annealing, *Science*, **220** (1983), 671–680. https://doi.org/10.1126/science.220.4598.671
11. D. E. Goldberg, K. Deb, A comparative analysis of selection schemes used in genetic algorithms, *Found. Genet. Algorithms*, **1** (1991), 69–93. https://doi.org/10.1016/B978-0-08-050684-5.50008-2
12. J. H. Holland, Adaptation in natural and artificial systems, The MIT Press, 1992. https://doi.org/10.7551/mitpress/1090.001.0001
13. K. A. De Jong, An analysis of the behavior of a class of genetic adaptive systems, University of Michigan, 1975.
14. E. U. Haq, A. Hussain, I. Ahmad, Performance evaluation of novel selection processes through hybridization of k-means clustering and genetic algorithm, *Appl. Ecology Environ. Res.*, **17** (2019), 14159–14177. http://dx.doi.org/10.15666/aeer/1706_1415914177
15. E. U. Haq, I. Ahmad, A. Hussain, I. M. Almanjahie, A novel selection approach for genetic algorithms for global optimization of multimodal continuous functions, *Comput. Intell. Neurosci.*, 2019. https://doi.org/10.1155/2019/8640218

16. S. Katoch, S. S. Chauhan, V. Kumar, A review on genetic algorithm: Past, present, and future, *Multimed Tools Appl.*, **80** (2021), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6
17. A. H. Wright, Genetic algorithms for real parameter optimization, *Founda. Genet. Algorithms*, **1** (1991), 205–218. https://doi.org/10.1016/B978-0-08-050684-5.50016-1
18. K. Deep, M. Thakur, A new crossover operator for real coded genetic algorithms, *Appl. Math. Comput.,* **188** (2007), 895–911. https://doi.org/10.1016/j.amc.2006.10.047
19. K. Deb, R. B. Agrawal, Simulated binary crossover for continuous search space, *Complex Syst.*, **9** (1995), 115–148.
20. F. B. Naqvi, M. Y. Shad, S. Khan, A new logistic distribution based crossover operator for real-coded genetic algorithm, *J. Stat. Comput. Simul.*, **91** (2021), 817–835. https://doi.org/10.1080/00949655.2020.1832093
21. M. Thakur, S. S. Meghwani, H. Jalota, A modified real coded genetic algorithm for constrained optimization, *Appl. Math. Comput.*, **235** (2014), 292–317. https://doi.org/10.1016/j.amc.2014.02.093
22. E. ul Haq, I. Ahmad, I. M. Almanjahie, A novel parent centric crossover with the log-logistic probabilistic approach using multimodal test problems for real-coded genetic algorithms, *Math. Probl. Eng.*, **2020** (2020), 2874528. https://doi.org/10.1155/2020/2874528
23. E. J. Gumbel, *Statistics of extremes*, Columbia university press, 1958. https://doi.org/10.7312/gumb92958
24. A. R. Kamel, A. A. Alqarni, M. A. Ahmed, On the performance robustness of artificial neural network approaches and Gumbel extreme value distribution for prediction of wind speed, *Int. J. Sci. Res. Math. Statist. Sci.*, **9** (2022), 5–22.
25. G. Grimmett, D. Stirzaker, *Probability and random processes*, Oxford university press, 2020.
26. S. Mahajan, L. Abualigah, A. K. Pandit, Hybrid arithmetic optimization algorithm with hunger games search for global optimization, *Multimed. Tools Appl.*, **81** (2022), 28755–28778. https://doi.org/10.1007/s11042-022-12922-z
27. A. Mardani, E. K. Zavadskas, K. Govindan, A. A. Senin, A. Jusoh, VIKOR technique: A systematic review of the state-of-the-art literature on methodologies and applications, *Sustainability*, **8** (2016), 37. https://doi.org/10.3390/su8010037
28. G. Zheng, X. Wang, The comprehensive evaluation of renewable energy system schemes in tourist resorts based on VIKOR method, *Energy*, **193** (2020), 116676. https://doi.org/10.1016/j.energy.2019.116676
29. M. Zeleny, *Multiple criteria decision making*, 1982.