BMR and BWR: Two simple metaphor-free optimization algorithms for solving constrained and unconstrained problems

Ravipudi Venkata Rao* and Ravikumar Shah
Sardar Vallabhbhai National Institute of Technology
Ichchanath, Surat, Gujarat – 395 007, India
Phone: 91-261-2201773; 91-9925207027
*Corresponding author (Email: ravipudirao@gmail.com)

Abstract: This paper presents two simple yet powerful optimization algorithms named Best-Mean-Random (BMR) and Best-Worst-Randam (BWR) algorithms to handle both constrained and unconstrained optimization problems. These algorithms are free of metaphors and algorithm-specific parameters. The BMR algorithm is based on the best, mean, and random solutions of the population generated for solving a given problem; and the BWR algorithm is based on the best, worst, and random solutions. The performances of the proposed two algorithms are investigated on 12 constrained engineering problems and the results are compared with the very recent algorithms (in some cases compared with more than 30 algorithms). Furthermore, computational experiments are conducted on 30 unconstrained standard benchmark optimization problems including 5 recently developed benchmark problems having distinct characteristics. The results proved the better competitiveness and superiority of the proposed simple algorithms. The optimization research community may gain an advantage by adapting these algorithms to solve various constrained and unconstrained real-life optimization problems across various scientific and engineering disciplines.

Keywords: Optimization; BMR algorithm; BWR algorithm; Constrained engineering problems; Unconstrained problems; New benchmarks.

1. Introduction

Population-based metaheuristic algorithms are very adaptable and are used to solve complex optimization problems in a variety of domains. They are especially helpful in situations when traditional optimization techniques—such as deterministic techniques or gradient-based methods—prove inappropriate because of certain factors like large search spaces, non-linearity, multimodality, or complex problem domains. Through a series of iterative procedures, the metaheuristic algorithms methodically investigate the solution space, improving the initial solution or solution population over time. Metaheuristics offer several advantages such as versatility, gradient independence, global search capability, multi-objective problem-solving capability, exploration and exploitation capability, configurability, practical applicability, etc. On the other hand, there are certain limitations of metaheuristics such as the absence of the global optimum guarantee, difficulty in achieving convergence in the case of high-dimensional or complex solution spaces, the requirement of tuning of common control parameters, and the algorithm-specific parameters, black-box nature, etc.

Nearly all algorithms that rely on population are probabilistic in nature and necessitate common parameters such as the number of generations and size of the population. With a few exceptions (e.g., Jaya algorithm, and Rao algorithms), each algorithm needs its own set of control parameters apart from common parameters. Inadequate adjustment of algorithm-specific parameters results in the local optimal solution or escalates the computing effort.

https://doi.org/10.32388/6EGLFW

The body of literature on metaheuristics has expanded significantly in recent years. The recent review papers on metaheuristics give a clear idea to the readers about various metaheuristics, their working principles, applications, limitations, future directions, etc. To date, more than 600 metaheuristic algorithms have been developed, with over 400 of them being developed during the past ten years. Many new optimization algorithms based on metaphors are released each month, with the authors claiming that their algorithms are “novel” and are better than those of the other algorithms.

A profusion of "novel" population-based metaheuristic algorithms, inspired by metaphors based on diverse natural phenomena including floods, disasters, animals (animals on earth as well as in the ocean), birds, insects, reptiles, fishes, viruses, diseases, matings, humans, human activities, societies, cultures, planets, heavenly bodies, plants, trees, swamps, deserts, musical instruments, sports, household items, physics, chemistry, mathematics, etc. have emerged in the last 15 years. The developers of these algorithms make an analogy of the equations proposed by them with any of the metaphors related to the phenomena mentioned above and try to justify the analogy. It is ironic that in almost all such algorithms there is no such real relation between the phenomena and the equations shown by them. This kind of research may be seen as risky and detrimental to the development of the optimization field. Several researchers have questioned the contentious subject of the exponential increase in new algorithms. Regretfully, a sizeable portion of the scientific community resorted to believing that the development of so-called “novel” optimization algorithms based on ever more bizarre analogies (in the name of metaphors) can advance science. Arguably the most dubious features of these techniques can be found in the literature such as meaningless and unfair metaphors, poor experimental validation, comparison, and lack of novelty.

Regretfully, over the past 10 years, we have seen the emergence of a new trend in which hundreds of metaphor-based metaheuristics have been proposed. These metaheuristics incorporate the greatest variety of natural, man-made, social, and sometimes even paranormal occurrences and actions, and their authors have not provided a clear rationale for their proposals other than the desire to get their papers published. Sörensen [1] opined that the current research trajectory in metaheuristics threatens to deviate from a rigorous scientific approach, and it appears that no concept is too ridiculous to serve as motivation to launch yet another metaheuristic algorithm. Sörensen et al. [2] described the development of metaheuristics over the course of five separate eras, beginning well before the name was coined and concluding far into the future. They commented that a sizable portion of the research community has fooled itself into believing that the development of so-called "novel" approaches that rely on ever-more bizarre analogies may advance science. By the time these metaphor-based ideas are suppressed, they expect that the scientific community will have suffered great injury, even though science will ultimately win out.

Campelo and Aranha [3] compiled a long list of "novel" algorithms and showed that developing a metaheuristic that just approximates a real-world process is a fruitless exercise and should not be added to the corpus of scientific literature. Moreover, when metaheuristics are used, the mathematical models obtained from the metaphors are often modified or omitted since they result in subpar implementations. Aranha et al. [4] opined that the emergence of publications that suggest metaphor-based algorithms that are influenced by often absurd processes that are not optimized at all, show poor scientific housekeeping, and reflect poorly on the metaheuristics research community.

The concerning thing is that a large number of metaphor-based metaheuristics include three (nearly wholly) distinct entities: the metaphor, the mathematical model "derived" from the metaphor, and the algorithm itself [5,6]. Rao [7] expressed concern that the flood of metaphor-based metaheuristics might threaten the optimization field's scientific viability and suggested that rather than concentrating on creating metaphor-based algorithms, researchers should concentrate on creating simple optimization strategies that can solve complex optimization problems.

The primary metaheuristic techniques and their diversification mechanisms were explained by Sarhani et al. [8]. They suggested a new classification for the current initialization techniques after reviewing and analyzing them. Rajwar et al. [9] reviewed about 540 metaheuristics and provided statistical information. The authors raised an important question: If the search properties of an optimization algorithm are altered or almost identical to those of current methods, can it still be considered "novel"? The authors categorized metaheuristics based on the number of control parameters, which is a new taxonomy in the field. They also presented a few limitations and open challenges.

Salgotra et al. [10] classified the metaheuristics as physics-based, human-based, swarm-based, and evolutionary-based. A large number of metaphor-based algorithms including some bizarre metaphors were mentioned in the classification. Different benchmark test functions related to existing metaheuristics were reviewed. It can be observed from the metaheuristics listed under each category of classification that the researchers have touched on almost all the “nature inspirations” and trying to make analogies irrespective of whether that metaphor has anything to do with the equations shown by them. Sharma and Raju [11] presented a comprehensive overview of metaheuristic optimization algorithms and the classification of benchmark test functions

Velasco et al. [12] examined 111 recent articles which proposed "new, hybrid, or improved optimization algorithms". A significant observation that was mostly ignored by the academics developing new algorithms was that only 43% of the reviewed articles referenced the No Free Lunch (NFL) theorem. The Black Widow Optimization and Coral Reef Optimization metaheuristics were examined to show how algorithms with little innovation can mistakenly be regarded as novel frameworks. These algorithms were found to be nothing more than inadequate combinations of various evolutionary operators.

Benaissa et al. [13] explained the core ideas and elements of metaheuristics, focusing on the utilization of search references and the careful balancing act between exploration and exploitation. They opined that, although intuitively appealing, metaphor-based optimization algorithms have generated controversy because of possible oversimplification and inflated expectations, and the names of the algorithms don't always correspond to the guiding ideas or methods they use. Sometimes, researchers use names that are fashionable or catchy, but these names cannot accurately represent the algorithm's originality or uniqueness.

It is clarified here that the objective of this paper is NOT to insult the researchers who have developed (and who are developing) metaphor-based optimization algorithms till now. The objective is to prove that there is no need to depend on metaphors to develop new optimization algorithms. The equations developed by the researchers in their papers may be able to solve the benchmark functions and other problems, but the analogies made with some natural or synthetic phenomena are not at all necessary. Developing straightforward optimization approaches that can solve complex optimization problems more effectively would be a better course of action for researchers than trying to develop metaphor-based algorithms. In light of this, the objectives of the work presented in this paper are listed below.

1. To prove that there is no need to depend on metaphors to develop optimization algorithms.
2. To develop two simple basic metaphor-free and algorithm-specific parameter-free optimization algorithms.
3. To test the performance of the proposed algorithms on 12 constrained engineering problems that have been recently attempted by *many* latest algorithms (in some cases more than 30 algorithms).

4. To demonstrate how well the proposed algorithms perform on a range of standard unconstrained optimization problems, including the most recent benchmark functions, each with unique characteristics.

The next section explains the proposed optimization algorithms.

2. Proposed best-mean-random (BMR) and best-worst-random (BWR) algorithms

2.1 BMR algorithm

Let $f(x)$, the objective function, be the function to be minimized or maximized. Assume that there are 'm' design variables and 'n' candidate solutions (i.e., population size, k=1,2,...,n) for every iteration i. The candidate with the best overall performance gets the best value of $f(x)$ (i.e., $f(x)_{\text{best}}$), while the candidate with the poorest overall performance gets the worst value of $f(x)$ (i.e., $f(x)_{\text{worst}}$) in all candidate solutions. Let r_1, r_2, r_3, and r_4 be four random numbers and each can take any value randomly from 0 to 1 and U_j and L_j be the upper and lower values of j^{th} variable. Additionally, let $V_{j,k,i}$ represent the j^{th} variable's value for the k^{th} candidate in the i^{th} iteration and T is a factor that randomly takes either 1 or 2 during an iteration.

$r_1, r_2, r_3, r_4 \sim \text{Uniform}(0, 1)$
$T \sim \text{Choice}(\{1, 2\})$
if $r_4 > 0.5$, the value of $V_{j,k,i}$ is changed as per Eq. (1).

$$V'_{j,k,i} = V_{j,k,i} + r_{1,j,i}\,(V_{j,best,i} - T^* V_{j.mean,i}) + r_{2,j,i}\,(V_{j,best,i} - V_{j,random,i}) \tag{1}$$

Else, $V'_{j,k,i} = \text{R} = U_j - (U_j - L_j)r_3$ (2)

The modified value of $V_{j,k,i}$ is $V'_{j,k,i}$. The best value of $f(x)$ during the i^{th} iteration is $V_{j,best,i}$ for the j^{th} variable. The mean value of j^{th} variable is $V_{j.mean,i}$ during the i^{th} iteration. The randomly picked up value, during the i^{th} iteration, for the j^{th} variable is $V_{j,random,i}$. The BMR algorithm's exploitation and exploration capabilities are explained in Eqs. (1) and (2).

2.2 BWR algorithm

With the same description of the terms given in sub-section 2.1, the BWR algorithm is described below.

$r_1, r_2, r_3, r_4 \sim \text{Uniform}(0, 1)$
$T \sim \text{Choice}(\{1, 2\})$
if $r_4 > 0.5$, the value of $V_{j,k,i}$ is changed as per Eq. (3).

$$V'_{j,k,i} = V_{j,k,i} + r_{1,j,i}\,(V_{j,best,i} - T^* V_{j,random,i}) - r_{2,j,i}\,(V_{j,worst,i} - V_{j,random,i}) \tag{3}$$

Else, $V'_{j,k,i} = \text{R} = U_j - (U_j - L_j)r_3$ (4)

The modified value of $V_{j,k,i}$ is $V'_{j,k,i}$. The best value of $f(x)$ during the i^{th} iteration is $V_{j,best,i}$ for the j^{th} variable. The worst value of $f(x)$ during the i^{th} iteration for the j^{th} variable is $V_{j,worst,i}$. The randomly picked up value of j^{th} variable during the i^{th} iteration is $V_{j,random,i}$. The BWR algorithm's exploitation and exploration capabilities are explained in Eqs. (3) and (4). It can be noted that both BMR and BWR algorithms are not based on any metaphors. Fig. 1 shows the flow diagram of the proposed BMR and BWR optimization algorithms.

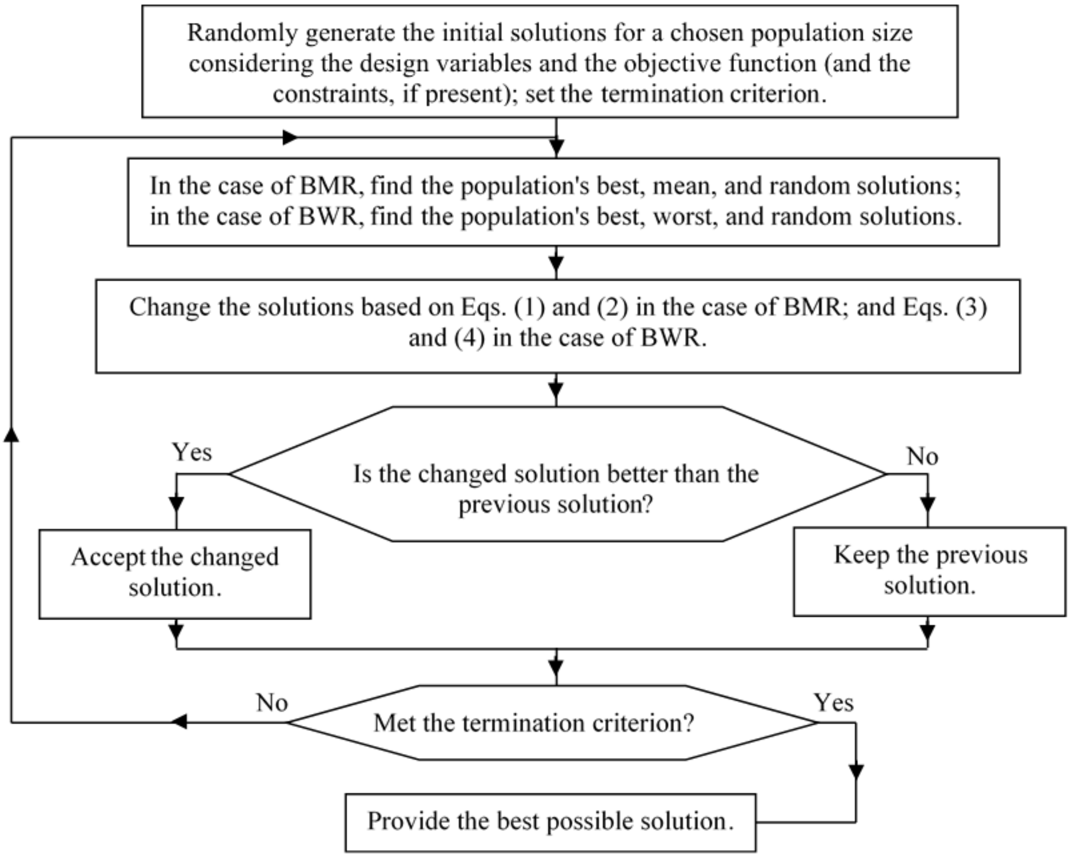


Fig. 1. Flow diagram of the BMR and BWR algorithms.

3. Illustration of the functionality of the proposed algorithms

3.1. Illustration of the functionality of the BMR algorithm

To illustrate the functionality of the BMR algorithm, we explore an unconstrained standard benchmark function of Sphere. The objective function is to determine the values of x_i that minimize the Sphere function.

Minimize, $$f(x_i) = \sum_{i=1}^{n} x_i^2 \tag{5}$$

Bounds of the variables: $-100 \leq x_i \leq 100$

This benchmark function's known solution is 0 for all x_i values of 0. To illustrate the BMR algorithm, let us use the following: two design variables, x_1 and x_2; an iteration serving as the termination criterion; and a population size of five (i.e., five solutions). Table 1 shows the values of the objective function corresponding to the initial population, which is created at random within the bounds of the variables. Since *f(x)* is a minimization function, the best solution is defined as having the lowest value, and the worst solution as having the greatest value.

Table 1
Randomly generated initial solutions.

Solution	x_1	x_2	$f(x)$	*Status*
1	-5	18	349	
2	14	33	1285	*worst*
3	30	-6	936	
4	7	-12	193	*best*
5	-18	8	388	
	Mean of x_1 = 5.6	Mean of x_2 = 8.2		

It is evident from Table 1 that solution 4 offers the best solution, while solution 2 offers the worst solution. Eq. (1) is used to determine the new values of the variables for x_1 and x_2 and are included in Table 2, considering random numbers $r_1 = 0.30$ and $r_2 = 0.10$ for x_1 and $r_1 = 0.60$ and $r_2 = 0.30$ for x_2. Assuming T=1 and random interaction with solution 5, the new values of x_1 and x_2 for solution 1 are computed as follows during the first iteration.

$$V'_{1,1,1} = V_{1,1,1} + r_{1,1,1}\,(V_{1,4,1} - 1\text{*Mean of } x_1) + r_{2,1,1}\,(V_{1,4,1} - V_{1,5,1})$$
$$= -5 + 0.30\,(7- 1\text{*}5.6) + 0.10\,(7- (-18)) = -2.08$$
$$V'_{2,1,1} = V_{2,1,1} + r_{1,2,1}\,(V_{2,4,1} - 1\text{*Mean of } x_2) + r_{2,2,1}\,(V_{2,4,1} - V_{2,5,1})$$
$$= 18 + 0.60\,(-12- 1\text{*}8.2) +0.30\,(-12-8) = -0.12$$

The new values of x_1 and x_2 for the remaining solutions are determined similarly. The new values of x_1 and x_2, along with the corresponding values of the objective function, are displayed in Table 2. For illustration purposes, solutions 2, 3, 4, and 5 are taken into consideration for their random interactions with 4, 2, 1, and 3, respectively.

Table 2
New values of x_1 and x_2 and $f(x)$ during the first iteration of the BMR algorithm.

Solution	x_1	x_2	$f(x)$
1	-2.08	-0.12	4.3408
2	14.42	20.88	643.9108
3	29.72	-31.62	1883.103
4	8.62	-33.12	1171.239
5	-19.88	-5.92	430.2608

After comparing the values of $f(x)$ in Tables 1 and 2, Table 3 is prepared and it contains the updated values of $f(x)$ based on fitness comparison. The first iteration of the BMR algorithm is now complete.

Table 3
Updated values of x_1 and x_2, and $f(x)$ after the first iteration of BMR algorithm.

Solution	x_1	x_2	$f(x)$	*Status*
1	-2.08	-0.12	4.3408	*best*
2	14.42	20.88	643.9108	
3	30	-6	936	*worst*
4	7	-12	193	
5	-18	8	388	

Table 3 illustrates that solution 1 is the best, while solution 3 is the worst. Additionally, it is evident that in just one iteration, the objective function's value drops from 193 to 4.3408. If the number of iterations is increased, the known value of the objective function, or 0, can be obtained in a few iterations. It is important to keep in mind that in cases of maximization problems, the highest value of the objective function is referred to as the best value, and calculations must be done accordingly. This means that problems involving either minimization or maximization can be handled using the proposed BMR algorithm.

3.2. Illustration of the functionality of the BWR algorithm

To illustrate the functioning of the BWR algorithm, the same Sphere function is considered. For a fair comparison, the same values of random numbers, the same values of *T*, and the same random interactions are considered. The values are calculated accordingly. Assuming T=1 and random interaction with solution 5, the new values of x_1 and x_2 for solution 1 are computed as follows during the first iteration.

$$V'_{1,1,1} = V_{1,1,1} + r_{1,1,1} (V_{1,4,1} - 1^* V_{1,5,1}) - r_{2,1,1} (V_{1,2,1} - V_{1,5,1})$$
$$= -5 + 0.30 (7- 1^*(-18)) - 0.10 (14- (-18)) = -0.70$$
$$V'_{2,1,1} = V_{2,1,1} + r_{1,2,1} (V_{2,4,1} - 1^* V_{2,5,1}) + r_{2,2,1} (V_{2,4,1} - V_{2,5,1})$$
$$= 18 + 0.60 (-12- 1^*8) -0.30 (33-8) = -1.50$$

The new values of x_1 and x_2 for the remaining solutions are determined similarly. The new values of x_1 and x_2, along with the corresponding values of the objective function, are displayed in Table 4.

Table 4
New values of x_1 and x_2, and *f(x)* during the first iteration of the BWR algorithm.

Solution	x_1	x_2	$f(x)$
1	-0.7	-1.5	2.74
2	13.3	19.5	557.14
3	27.9	-33	1867.41
4	8.7	-34.5	1265.94
5	-23.3	-7.3	596.18

After comparing the values of *(x)* in Tables 1 and 4, Table 5 is prepared and it contains the updated values of *f(x)* based on fitness comparison. The first iteration of BWR algorithm is now complete.

Table 5
Updated values of x_1 and x_2, and *f(x)* after the first iteration of BWR algorithm.

Solution	x_1	x_2	$f(x)$	*Status*
1	-0.7	-1.5	2.74	*best*
2	13.3	19.5	557.14	
3	30	-6	936	*worst*
4	7	-12	193	
5	-18	8	388	

Table 5 illustrates that solution 1 is the best solution, while solution 3 is the worst. Additionally, it is evident that in just one iteration, the objective function's value drops from 193 to 2.74. The known value of the objective function, or 0, can be reached in a few iterations if the number of iterations is increased. Problems involving either minimization or maximization can be handled by the BWR.

This illustration pertains to an unconstrained optimization problem. Nonetheless, the same procedures can be employed when dealing with constrained optimization problems. The primary distinction is that in the constrained optimization problem, each violation of a constraint is handled by a penalty function that is applied to the objective function.

The experimentation of the proposed algorithms on 12 constrained benchmark problems given in [21] is explained in the following section. In the present work, MATLAB r2024a has been used to implement the BMR and BWR algorithms. A laptop with Microsoft Windows 10 operating system with AMD Ryzen 7 CPU and 24 GB RAM has been used for doing the computational experiments.

4. Experiments on 12 constrained engineering optimization problems

Very recently in June 2024, Ghasemi et al. [21] proposed a metaphor-based algorithm named "Flood Algorithm (FLA)" and compared its performance with *so many* optimization algorithms (more than 30 algorithms in some cases) in solving certain Congress on Evolutionary Computation (CEC) 2005 and 2014 functions along with 12 constrained engineering problems. The decision variables, objective functions, constraints, and the bounds of the decision variables are available in Ghasemi et al. [21] and hence are not reproduced here for space reasons and to avoid plagiarism issues.

Now the proposed BMR and BWR algorithms are applied to the same 12 constrained engineering problems under the same conditions with 30 runs as those used by FLA [21] and other *so many* other optimization algorithms. A static penalty method is used to deal with constraint violations. For example, in the case of problem 1 (i.e., a minimization problem related to welded beam design) which has seven constraints $g_1(x), \ldots\ldots,$ and $g_7(x)$, the penalized value of $f(x)$ is calculated as, Penalized $f(x)= f(x)+10*g_1(x)^2+10*g_2(x)^2+\ldots\ldots\ldots+10*g_7(x)^2$. If there is no constraint violation, then there will not be any penalty, and the Penalized $f(x)= f(x)$. It may be noted here that the user can decide which type of penalty can be imposed for constraint violation. Table 6 presents the so *many optimization algorithms* with which the FLA was compared by Ghasemi et al. [21].

Table 6
List of the optimization algorithms* attempted previously on 12 constrained engineering problems.

Problem numbers and the optimization algorithms used											
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
CPO	SCHO	BLPSO	mGWO	SCHO	YDSE	WOA	YDSE	AD-IFA	PSO	AEFA-C	MPDO
IAS	PSA	MBWO	BES	PSA	VCO	SSA	SRS	LS-LF-FA	DE	FPSA	MGO
SCHO	AMO	CCEO	GOA	KOA	BP-εMAg-ES	MBA	CPA	LF-FA	GA	AD-IFA	RAO-3
LSO	DSA	IAS	EBS	DSA	COLSHA	GWO	SOS	FA	HPSO	LS-LF-FA	PSA
KOA	ESOA	MPDO	UPSO	EEFO	DE−QL	ER-WCA		DO	HPSO-Q	LF-FA	WOA

SWO	iLSHADEε	PSA		VCO	VMCH	ALO		KOA	SNS	FA	SSA
GSO	RL-BA	EEFO		GGO	UPSO	LFD		DBB-BC			MBA
DSA	AD-IFA	AD-IFA		ESOA	G-QPSO	ACVO					WCA
VCO	LS-LF-FA	LS-LF-FA		WO	CPSO	EChOA					ER-WCA
SAO	LF-FA	LF-FA		DE−QL	mGWO	I-GWO					ALO
OA	FA	FA		VMCH	RFO	HFPSO					MFO
MMLA	SFO	GCHHO		EnMODE	EO	HEAA					T-CSS
AD-IFA	mGWO	GOA		QS	CDE	SHO					CSS
LS-LF-FA	PSO-HBF	MFO		GCHHO	DHOA	SETO					FACSS
LF-FA		WOA		SMA-AGDE		LFD					
FA		SMA		COOT		SELO					
WCA		m-SCA		SDO		AHA					
SFO				CPSO		AO					
EPSO				mGWO		MBWO					
FSA				PFA		CCEO					
CPSO				G-QPSO		MPDO					
TEO				WCA		SCHO					
CDE				DDAO		GAO					
UPSO				CDE		YDSE					
PFA				(l + λ)-ES		LEA					
HGS				HPSO		CSA					
EO				EO		SCA					
GWO				INFO		MVO					
IPSO				NRBO		MFO					
HMS				IMSCSO		RSA					
POA				LSO		hHHO-SCA					
CPO				EBS		AOA					
				HGA							
				TDO							
				UPSO							
				CSA							
				SCA							
				MVO							
				MFO							

*The abbreviations of the optimization algorithms are available in Ghasemi et al. [21].

Table 7 presents the results of BMR and BWR algorithms along with the results of FLA. *The results of so many other algorithms are not included in Table 7, as FLA has already claimed its supremacy over those algorithms, and it is felt that comparison with FLA is sufficient to check the performance of BMR and BWR algorithms.*

Table 7
Statistical results obtained by BMR and BWR algorithms and FLA for 12 constrained engineering problems.

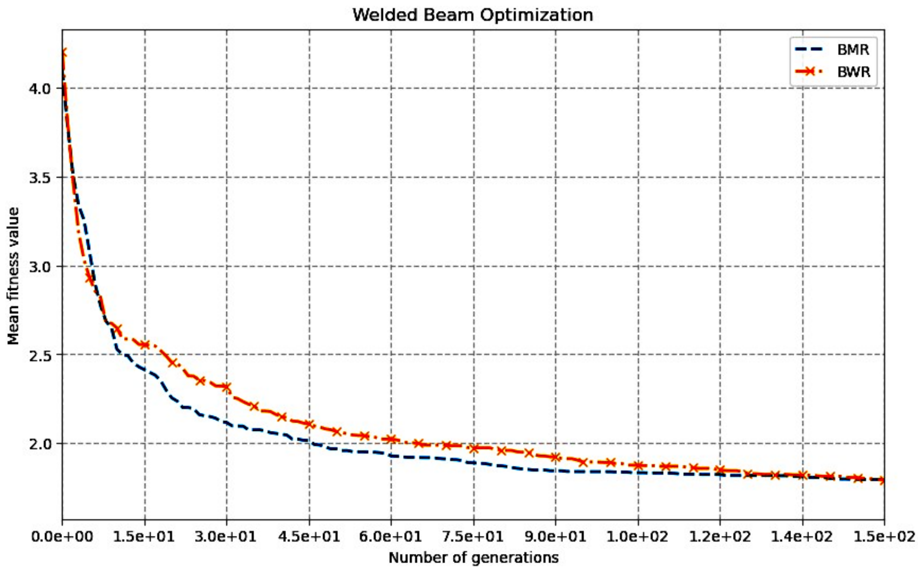
No.	Name of the problem	Algorithm	Best	Mean	Worst	Std. dev.
1	Welded beam optimization	BMR	1.6981	1.7010	1.7032	1.5674*E* − 03
		BWR	**1.6979**	**1.6979**	**1.6979**	2.7043*E* − 10
		FLA [21]	1.7248523	1.7248527	1.7248536	3.08E-06
2	Three-bar truss optimization	BMR	**1.085211*E*+02**	**1.085211*E*+02**	**1.085211*E*+02**	1.4504*E* − 14
		BWR	**1.085211*E*+02**	**1.085211*E*+02**	**1.085211*E*+02**	1.8346*E* − 14
		FLA [21]	263.89584	263.89586	263.89665	7.10E-05
3	Cantilever beam optimization	BMR	**1.3351**	**1.3351**	**1.3351**	1.9342*E* − 11
		BWR	**1.3351**	**1.3351**	**1.3351**	1.5367*E* − 14
		FLA [21]	1.339956	1.339958	1.339963	6.48E-07
4	Optimal design of gear train	BMR	4.287642E−22	3.4497E − 18	3.4131E − 17	8.21E − 18
		BWR	**7.3856E−25**	**7.1755E − 21**	**4.3061E − 20**	1.121E − 20
		FLA [21]	2.700857E-12	8.7526E-10	1.4069E-09	2.76E-09
5	Tension/compression spring optimization	BMR	**0.012648**	**0.012648**	**0.012648**	8.0429E − 14
		BWR	**0.012648**	**0.012648**	**0.012648**	**0.012648**
		FLA [21]	0.0126652	0.012666	0.012667	6.29E-07
6	Pressure vessel optimization	BMR	**4.840545E+02**	**4.840545E+02**	**4.840545E+02**	1.6409E − 13
		BWR	**4.840545E+02**	**4.840545E+02**	**4.840545E + 02**	1.6409E − 13
		FLA [21]	6.059714E+03	6.06021E+03	6.09052E+03	3.86
7	Speed reducer optimization	BMR	**2.35748E+03**	**2.357481E+03**	**2.35748E+03**	9.2825E−13
		BWR	**2.35748E+03**	**2.35748E+03**	**2.35748E+03**	9.2825E−13
		FLA [21]	2.99447E+03	2.994471E+03	2.994473E+03	2.09E-04
8	I-beam vertical deflection	BMR	**0.0016369**	**0.0016369**	**0.0016369**	6.6394E − 19
		BWR	**0.0016369**	**0.0016369**	**0.0016369**	6.6394E − 19
		FLA [21]	0.013074	0.01307445	0.01307579	6.91E-06
9	Tubular column optimal design	BMR	1.03168E+01	1.03168E+01	1.03168E+01	9.2825E−13
		BWR	**1.03168E+01**	**1.03168E+01**	**1.03168E+01**	9.2825E−13
		FLA [21]	2.64995E+01	2.64995E+01	2.651003E+01	1.41E-04
10	Piston lever optimal design	BMR	**7.585**	**7.585**	**7.5851**	2.4052E−05
		BWR	**7.585**	**7.585**	**7.585**	2.054E-14
		FLA [21]	8.412698	23.821251	167.232196	47.2
11	Corrugated bulkhead optimal design	BMR	**6.5795**	**6.5795**	6.5795	2.7195E − 15
		BWR	**6.5795**	**6.5795**	6.5795	2.7195E − 15
		FLA [21]	6.842958	6.8429676	6.8432916	1.25E-05

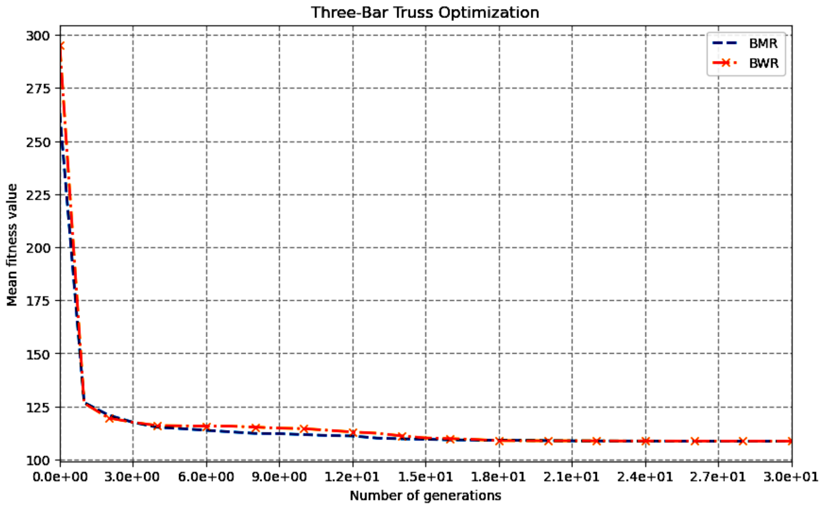
12	Car side impact optimization	BMR	**2.22857E+01**	**2.22857E+01**	**2.22857E+01**	1.0534E − 14
		BWR	**2.22857E+01**	**2.22857E+01**	**2.22857E+01**	1.0534E − 14
		FLA [21]	2.284297E+01	2.288914E+01	2.317638E+01	7.38E-03

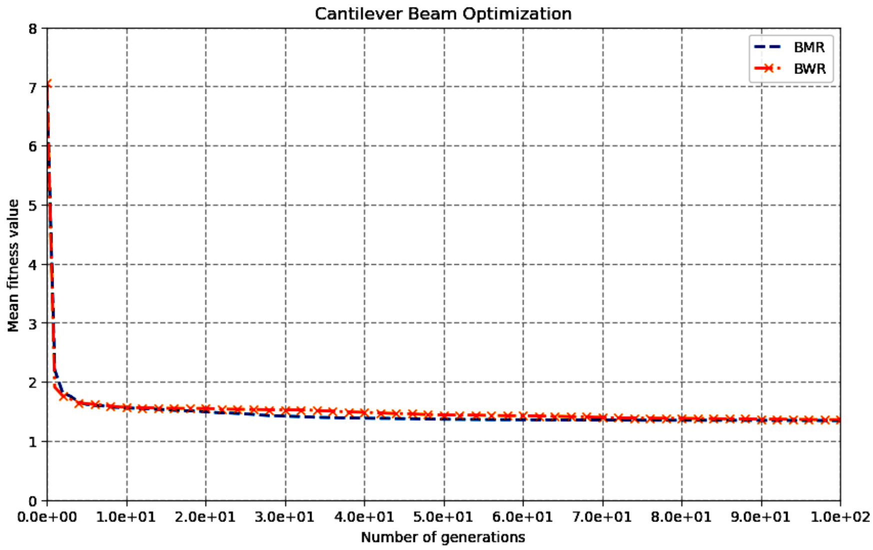
The **bold** numbers denote better values in comparison to the similar values provided by the FLA [21].

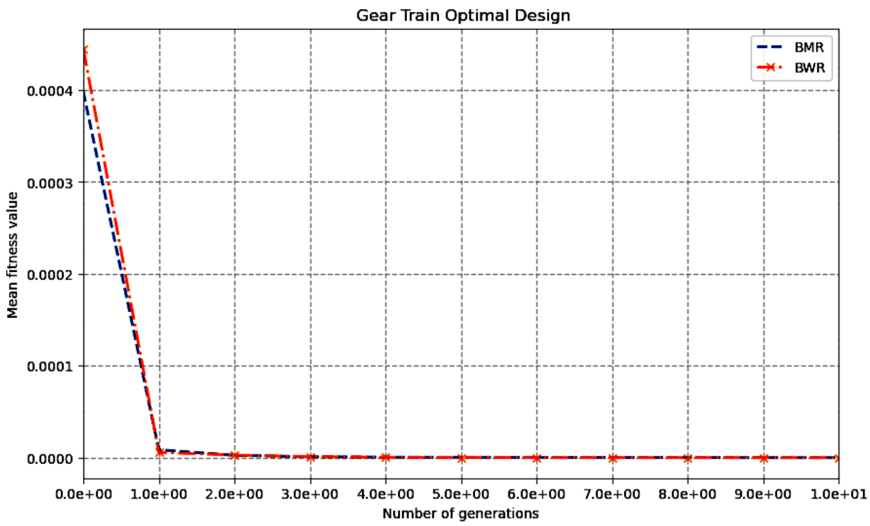
It can be noted that the BMR and BWR algorithms have outperformed the very recently published FLA [21]. **It is very interesting to note that the FLA was shown by Ghasemi et al. [21] as superior to 32 other algorithms in the case of problem 1; 14 other algorithms in the case of problem 2; 17 other algorithms in the case of problem 3; 5 other algorithms in the case of problem 4; 39 other algorithms in the case of problem 5; 14 other algorithms in the case of problem 6; 32 other algorithms in the case of problem 7; 4 other algorithms in the case of problem 8; 7 other algorithms in the case of problem 9; 6 other algorithms in the case of problem 10; 6 other algorithms in the case of problem 11; and 14 other algorithms in the case of problem 12. Now the proposed BMR and BWR algorithms have shown better performance in all 12 engineering problems, compared to FLA [21] which was recently published in June 2024.**

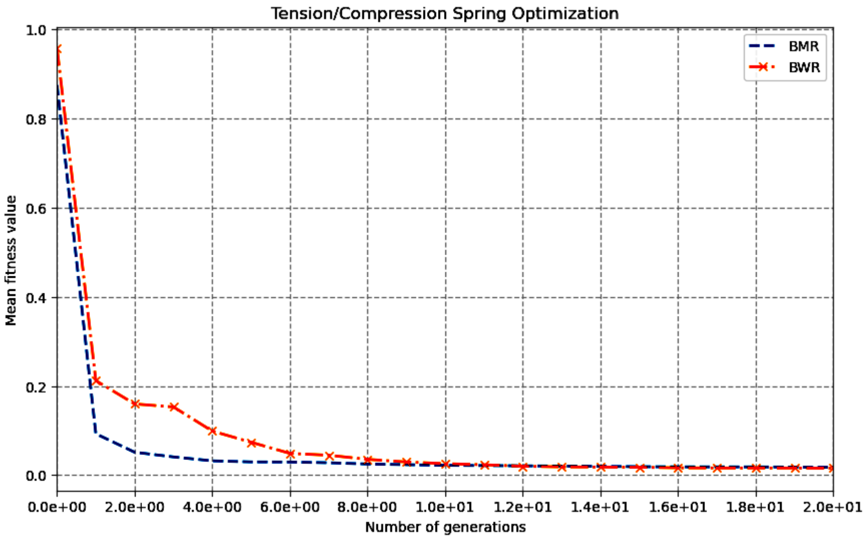
The convergence behavior of the BMR and BWR algorithms is shown in Fig. 2. It may be noted that the 0e+00 shown at the origin of the graphs indicates the iteration during which the population is randomly generated. *Complete convergence till the end is not clearly visible in the graphs in certain cases (because of the scale step size taken on x- and y-axes) but the readers may understand that the convergence occurred at the mean function values shown in Table 7.*

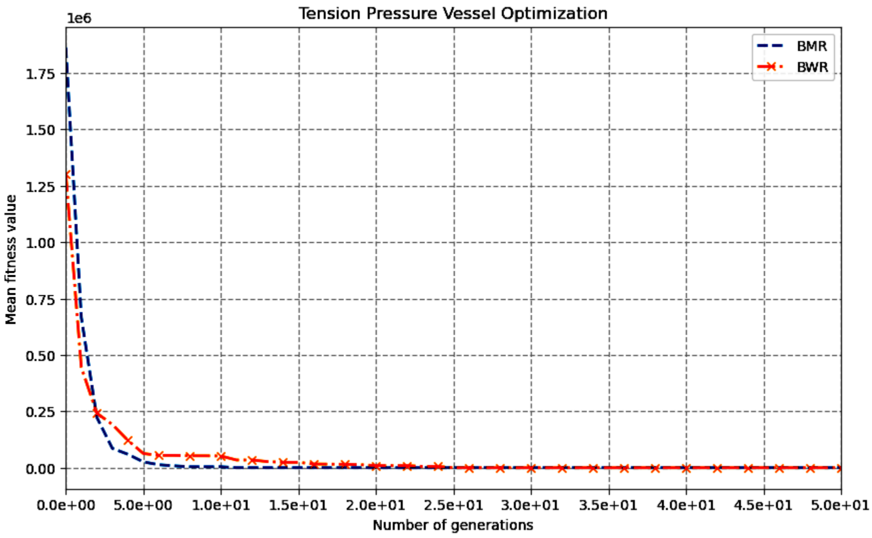


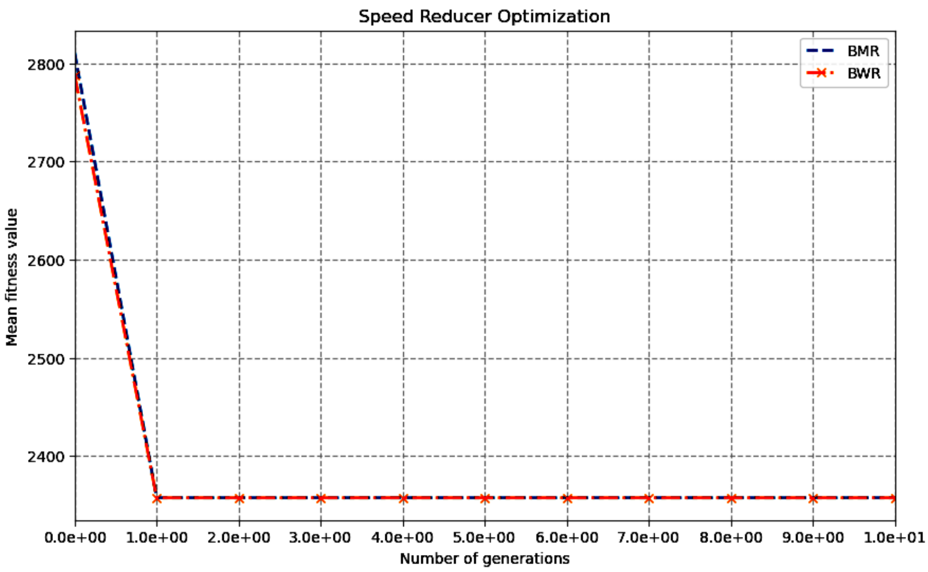

Three-Bar Truss Optimization
Mean fitness value
Number of generations
BMR
BWR

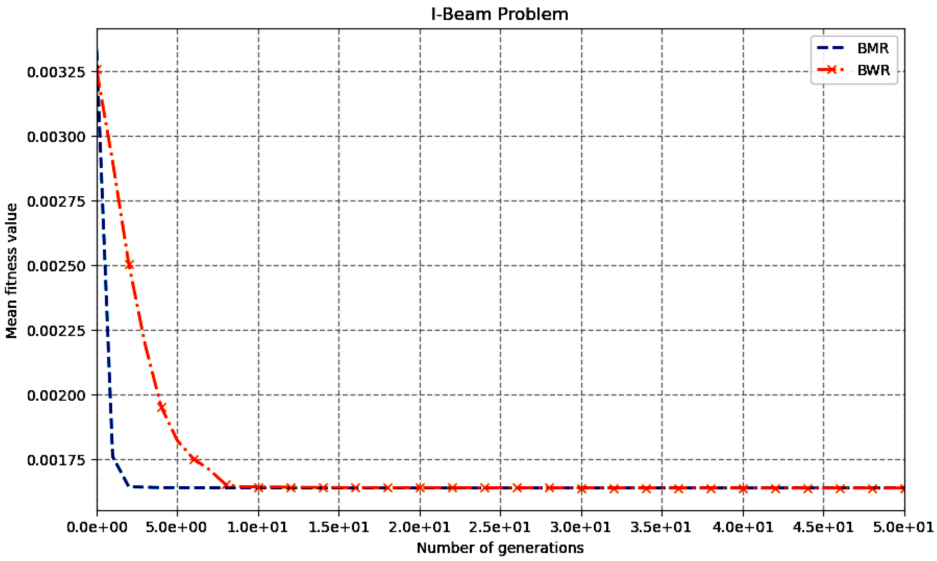

Cantilever Beam Optimization
Mean fitness value
Number of generations
BMR
BWR

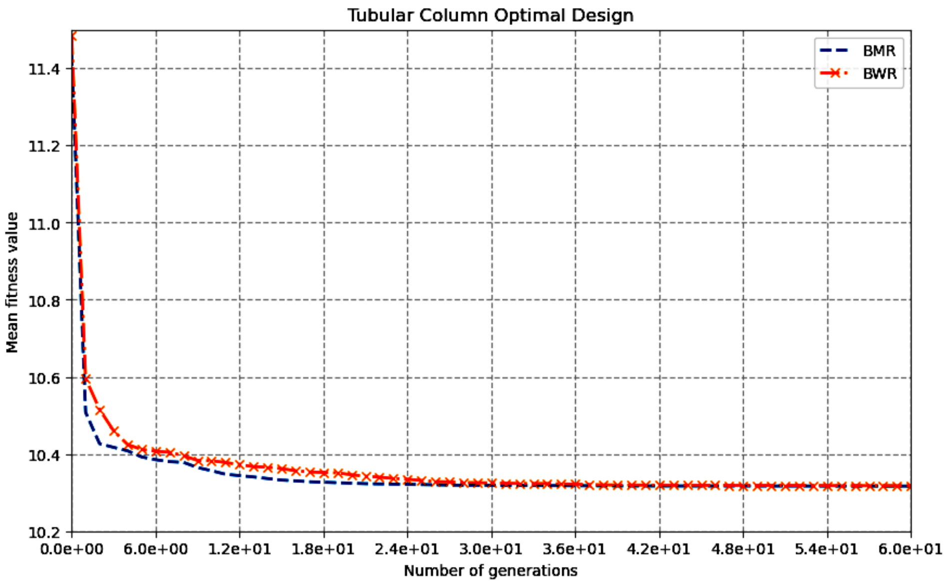

Gear Train Optimal Design
Mean fitness value
Number of generations
BMR
BWR

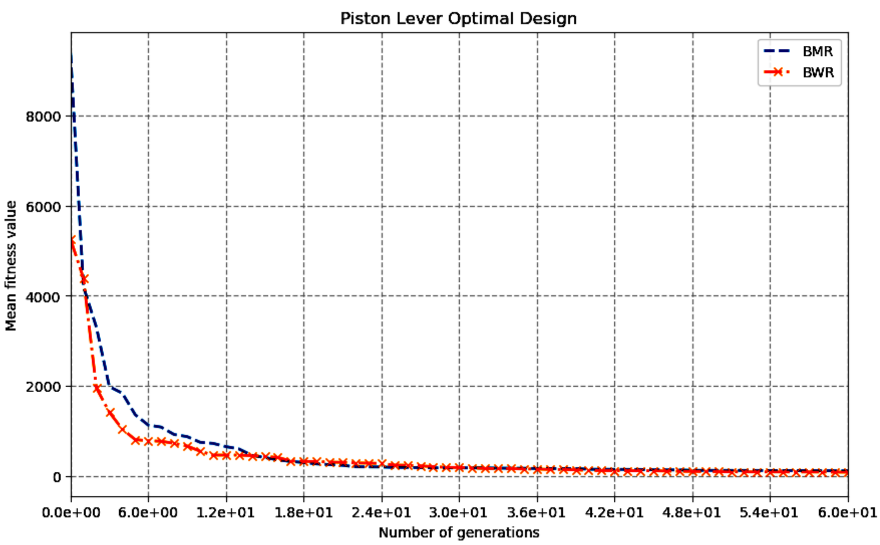

Tension/Compression Spring Optimization
Mean fitness value
Number of generations
BMR
BWR
0.0
0.2
0.4
0.6
0.8
1.0
0.0e+00
2.0e+00
4.0e+00
6.0e+00
8.0e+00
1.0e+01
1.2e+01
1.4e+01
1.6e+01
1.8e+01
2.0e+01

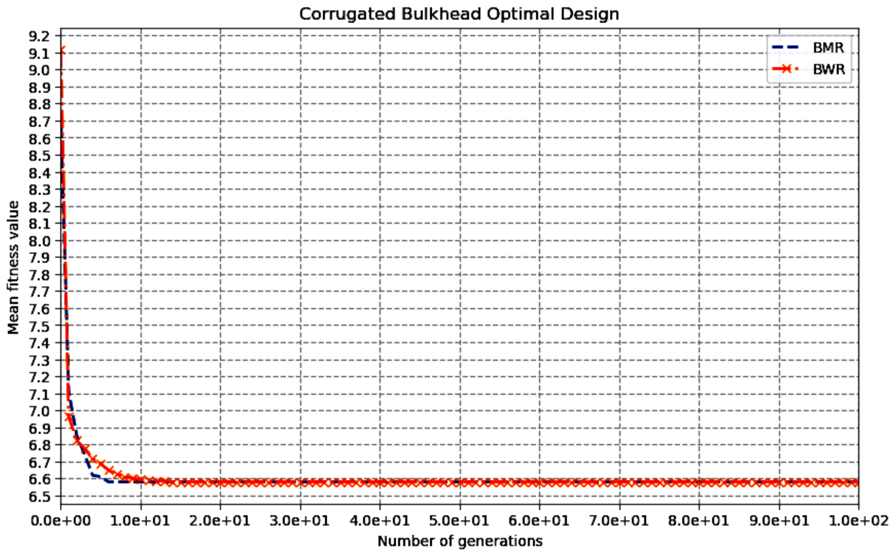

Tension Pressure Vessel Optimization
1e6
Mean fitness value
Number of generations
BMR
BWR
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
0.0e+00
5.0e+00
1.0e+01
1.5e+01
2.0e+01
2.5e+01
3.0e+01
3.5e+01
4.0e+01
4.5e+01
5.0e+01


Speed Reducer Optimization
Mean fitness value
Number of generations
BMR
BWR
2400
2500
2600
2700
2800
0.0e+00
1.0e+00
2.0e+00
3.0e+00
4.0e+00
5.0e+00
6.0e+00
7.0e+00
8.0e+00
9.0e+00
1.0e+01


I-Beam Problem
BMR
BWR
Mean fitness value
Number of generations


Tubular Column Optimal Design
BMR
BWR
Mean fitness value
Number of generations


Piston Lever Optimal Design
BMR
BWR
Mean fitness value
Number of generations



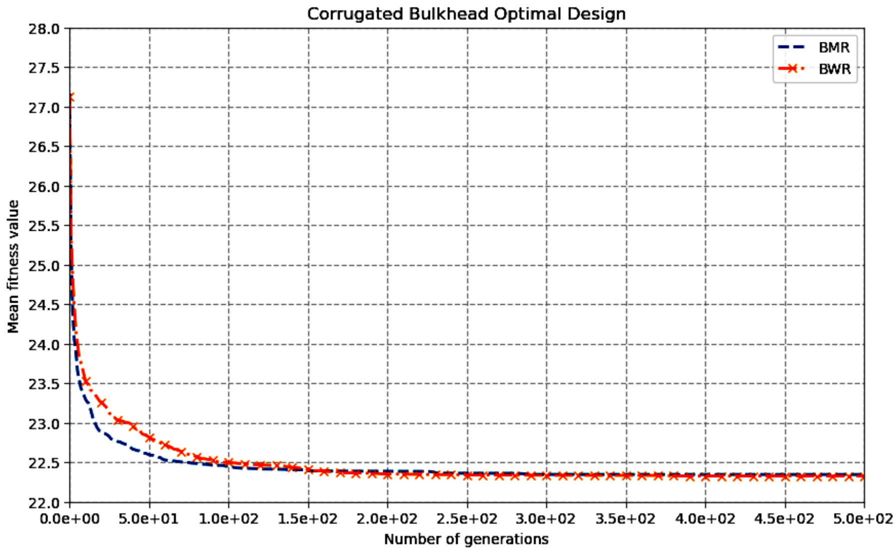


Fig. 2. Convergence graphs of the BMR and BWR algorithms for 12 engineering problems.

5. Experiments on 30 unconstrained optimization problems

5.1 Experiments on 25 unconstrained standard benchmark functions

To test the performance of BMR and BWR algorithms on unconstrained optimization problems, 25 standard benchmark functions frequently used by the researchers are considered. These benchmark functions are separable, non-separable, multimodal, and unimodal. The algorithms are coded in Python 3.11.5. Thirty separate runs of each function and a maximum of 500000 function evaluations are used in the computational studies. Table 8 displays the "Best", "Mean", "Worst", "Standard Deviation (Std. dev.)", and "Mean function evaluations (MFE)" for the BMR and BWR algorithms.

Table 8
Statistical results obtained by BMR and BWR algorithms for 25 unconstrained standard benchmark problems.

No.	Unconstrained function	Optimum	Algorithm	Best	Mean	Worst	Std. dev.	MFE
F 1	Sphere	0	BMR	0	0	0	0	125018
			BWR	0	0	0	0	68256
F 2	SumSquares	0	BMR	0	0	0	0	124709
			BWR	0	0	0	0	62936
F 3	Beale	0	BMR	0	0	0	0	10317
			BWR	0	0	0	0	4535
F 4	Easom	-1	BMR	0	0	0	0	5174
			BWR	0	0	0	0	2891
F 5	Matyas	0	BMR	0	0	0	0	13610
			BWR	0	0	0	0	23663
F 6	Colville	0	BMR	0	0	0	0	23195
			BWR	0	0	0	0	14469
F 7	Trid 6	-50	BMR	-50	-50	-50	0	18496
			BWR	-50	-50	-50	0	13793
F 8	Trid 10	-210	BMR	-210	-210	-210	0	55635
			BWR	-210	-210	-210	0	52834
F 9	Zakharov	0	BMR	0	0	0	0	128387
			BWR	0	0	0	0	79267
F 10	Schwefel 1.2	0	BMR	0	0	0	0	129580
			BWR	0	0	0	0	80000
F 11	Rosenbrock	0	BMR	0	4.62E-29	1.09E-29	1.44E-29	434010
			BWR	0	0	0	0	167089
F 12	Dixon-Price	0	BMR	0.24906	0.24906	0.24906	0	19000
			BWR	0.24906	0.24906	0.24906	0	14300
F 13	Branin	0.397887	BMR	0.397887	0.397887	0.397887	0	22330
			BWR	0.397887	0.397887	0.397887	0	11080
F 14	Bohachevsky 1	0	BMR	0	0	0	0	2746
			BWR	0	0	0	0	1788
F 15	Bohachevsky 2	0	BMR	0	0	0	0	2738
			BWR	0	0	0	0	1761
F 16	Bohachevsky 3	0	BMR	0	0	0	0	2757
			BWR	0	0	0	0	1749
F 17	Booth	0	BMR	0	0	0	0	7862
			BWR	0	0	0	0	3910
F 18	Michalewicz 2	-1.8013	BMR	-1.8013	-1.8013	-1.8013	0	1819
			BWR	-1.8013	-1.8013	-1.8013	0	1157
F 19	Michalewicz 5	-4.6877	BMR	-4.6877	-4.6877	-4.6877	5.76E-07	180120
			BWR	-4.6877	-4.6877	-4.6877	1.38E-15	23600

F 20	GoldStein-Price	3	BMR	3	3	3	1.94E-14	12517
			BWR	3	3	3	1.87E-14	4317
F 21	Perm	0	BMR	0	0	0	0	55635
			BWR	0	0	0	0	38393
F 22	Ackley	0	BMR	4.44E-16	4.44E-16	4.44E-16	0	11350
			BWR	4.44E-16	4.44E-16	4.44E-16	0	2300
F 23	Foxholes	0.998004	BMR	0.998004	0.998004	0.998004	0	741
			BWR	0.998004	0.998004	0.998004	0	600
F 24	Hartmann 3	-3.86278	BMR	-3.86278	-3.86278	-3.86278	0	1784
			BWR	-3.86278	-3.86278	-3.86278	0	780
F 25	Penalized 2	0	BMR	1.50E-33	1.50E-33	1.50E-33	0	402120
			BWR	1.50E-33	1.50E-33	1.50E-33	0	150000

Recently, Rao and Pawar [20] used I-Rao algorithm for solving the above 25 unconstrained functions and proved that I-Rao performed better than the three Rao algorithms reported by Rao [7]. Hence, the results of BMR and BWR algorithms are compared now with those of the I-Rao. Table 9 presents the summary of the performance of BMR and BWR algorithms compared to the I-Rao algorithm. The comparison summary is in terms of how many times the BMR and BWR algorithms performed "Better" or "Similar or equal" or "Inferior" to the I-Rao algorithm. The "Success %" is calculated similarly as explained in section 4.

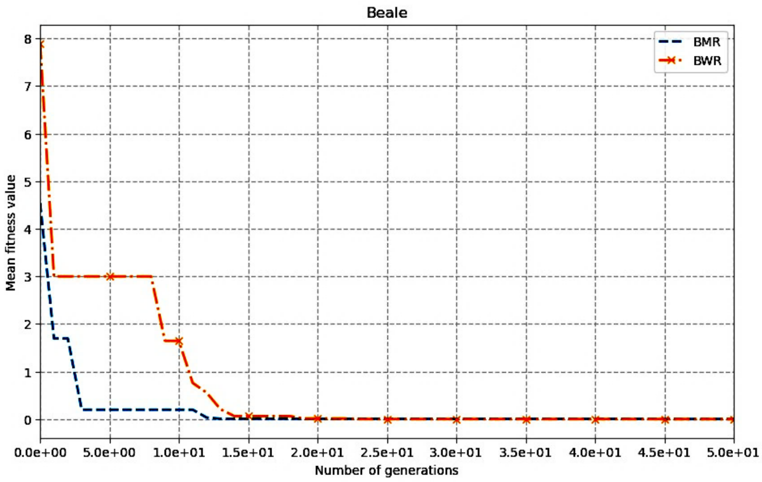
Table 9

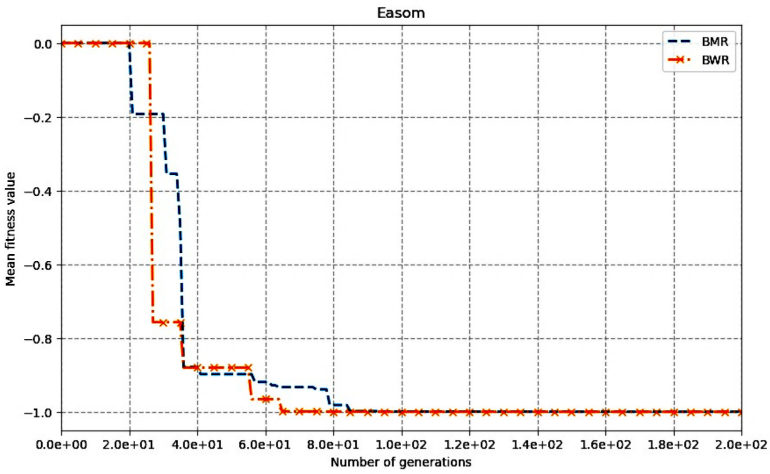
Summary of the performance of BMR and BWR algorithms for 25 unconstrained problems.

Criterion	Best	Mean	Worst	MFE
		BMR vs. I-Rao*		
Better	3	5	5	17
Similar or equal	21	20	20	0
Inferior	1	0	0	8
Success %	96	100	100	68
		BWR vs. I-Rao*		
Better	3	5	5	22
Similar or equal	21	20	20	0
Inferior	1	0	0	3
Success %	96	100	100	88
		BWR vs. BMR		
Better	0	1	1	24
Similar or equal	25	24	24	0
Inferior	0	0	0	1
Success %	100	96	96	96

*Results of I-Rao are taken from [20].

The convergence behavior of BMR and BWR algorithms for 4 selected unconstrained functions is shown in Fig. 3. These graphs give an idea about the convergence behavior. The convergence graphs for the remaining 21 unconstrained problems are not shown for space reasons.





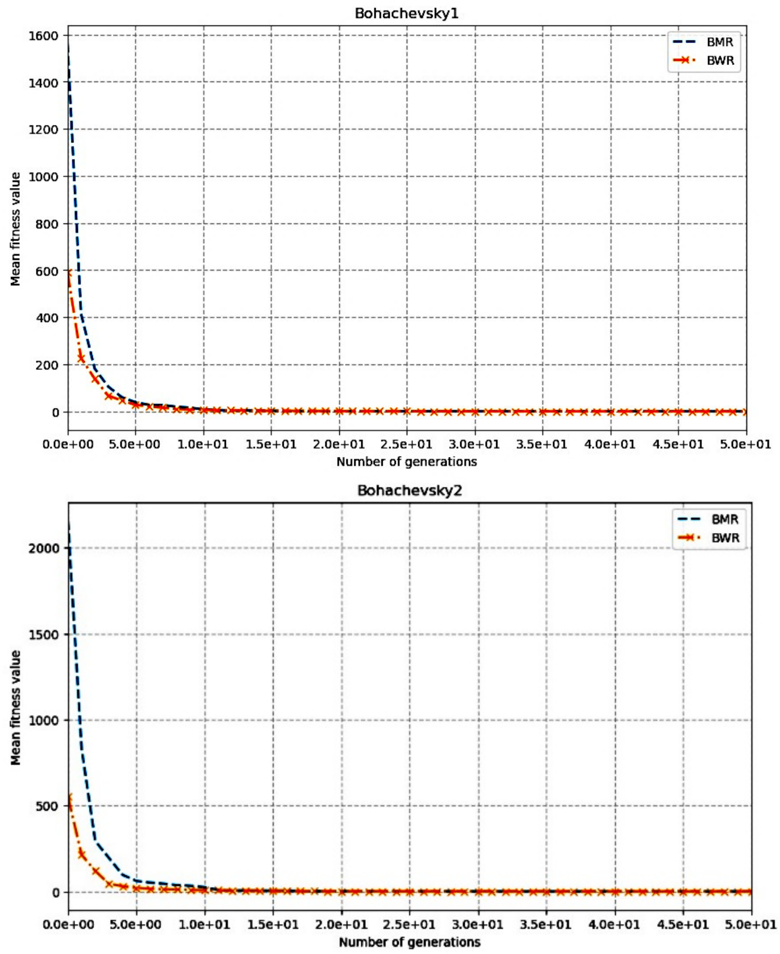


Fig. 3. Convergence graphs of the BMR and BWR algorithms for 4 unconstrained functions.

5.2 Experiments on 5 new unconstrained standard benchmark functions

To further demonstrate the potential of the proposed BMR and BWR algorithms on unconstrained optimization problems, 5 out of 10 latest benchmark functions recently proposed by Yang [23] are considered. Thirty separate runs of each function and a maximum of 500000 function evaluations are used in the computational studies. The "Best", "Mean", "Worst", "Standard deviation (std. dev.)", and "Mean function evaluations (MFE)" corresponding to BMR and BWR algorithms are shown in Table 10.

Table 10
Statistical results of BMR and BWR algorithms for the latest benchmark functions of Yang [23].

S. No.	New benchmark function	Optimum	Algorithm	Best	Mean	Worst	Std. dev.	MFE
1	Complex Noisy Function	-1	BMR	-1	-1	-1	0	2787
			BWR	-1	-1	-1	0	2772
2	Non-differentiable function	0	BMR	3.21228E-06	3.21228E-06	3.21228E-06	0	250380

			BWR	1.8488E-07	1.8488E-07	1.8488E-07	0	221840
3	Hyperboloid Function	1	BMR	1	1	1	0	471400
			BWR	1	1	1	0	222030
4	Non-Smooth Multi-Layered Function (D=1)	0	BMR	0	0	0	0	155
			BWR	0	0	0	0	232
5	Shortest-Path Problem	1	BMR	1	1	1	0	400
			BWR	1	1	1	0	209

To understand the convergence behavior, the convergence graph for “non-smooth multi-layered function” is shown in Fig. 4.

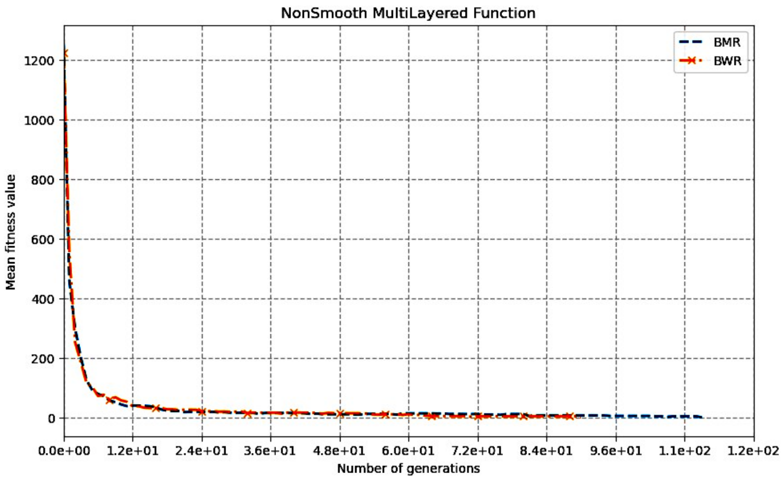


Fig. 4. Convergence graph for the non-smooth multi-layered function.

6. Discussion on the results obtained for constrained engineering problems, and unconstrained problems

6.1 Constrained engineering problems

In the case of constrained engineering problems, it is evident from Table 7 that when compared to the FLA [21] and *many more algorithms* which FLA had outperformed, the BMR and BWR algorithms performed much better in terms of "Best," "Mean," and "Worst". Both BMR and BWR algorithms performed equally well on these 12 problems. The convergence graphs shown in Fig. 4 indicate the better convergence behavior of BMR and BWR algorithms.

In another preprint [22], the results of the application of BMR and BWR algorithms on 26 real-life non-convex constrained optimization problems of CEC 2020 [14] were presented. The proposed algorithms were found better than the IUDE, εMAgES, iLSHADE$_{\varepsilon}$, COLSHADE, EnMODE, and I-Rao algorithms in terms of “Best”, “Median”, “Mean”, “Feasibility Rate (FR)”, “Mean Constraint Violation (MV)”, and “Success Rate (SR)”. However, the performance of BWR was found slightly better than the BMR algorithm.

6.2 Unconstrained optimization problems

6.2.1 Standard unconstrained optimization problems

In the case of 25 standard unconstrained optimization problems, the selected convergence graphs shown in Fig. 5 for Beale, Easom, Bohachevsky 2, and Bohachevsky 2 indicate the better convergence behavior of BMR and BWR algorithms. In the case of other unconstrained problems also, the convergence behavior is found appreciable (however, those graphs are not shown in this paper for space reasons).

Table 8 shows the summary of the performance of BMR and BWR algorithms for 25 problems compared to the recently published I-Rao algorithm of Rao and Pawar [20]. The comparison summary is in terms of how many times the BMR and BWR algorithms performed "Better" or "Similar or equal" or "Inferior" to the other algorithms. It is clear from Table 10 that the success % of BMR and BWR algorithms is very high such as more than 90% (i.e., 100% and 96%). In the case of MFE, compared to the I-Rao algorithm, the BMR and BWR algorithms' success % are 68 and 88 respectively. Further, it can be noted that both BMR and BWR algorithms performed well on these 25 unconstrained problems. However, the performance of BWR may be found somewhat better than the BMR algorithm.

6.2.2 New unconstrained optimization problems of Yang [23]

Statistical results of BMR and BWR algorithms for the 5 latest benchmark functions of Yang [23] presented in Table 12 show that the proposed BMR and BWR algorithms produced the optimum results. It can be noted that the MFE required by BWR is less than the BMR algorithm. The convergence behavior is also found good.

Normally statistical tests such as the Friedman test, Home-Sidak test, etc. are conducted to find the significance and to rank the competing optimization algorithms. However, these tests are not necessary here, as for the constrained and unconstrained problems presented in this paper, the BMR and BWR algorithms have well established their competitiveness by providing better Best, Median, Mean, and MFE values (with the performance of BWR algorithm slightly better than BMR algorithm in some problems).

7. Conclusions

The proposed BMR algorithm is based on "Best", "Mean", and "Random" values in the population of a given iteration, and the proposed BWR algorithm is based on "Best", "Worst", and "Random" values. *These two algorithms are developed in the present work without using any metaphors (as explained in section 1) and proved that there is no need to depend on metaphors to develop the new optimization algorithms.* The metaphor-free and algorithm-specific parameter-free BMR and BWR algorithms are simple to understand and easy to implement. The efficiency of the proposed algorithms is demonstrated in terms of convergence and results on 12 constrained engineering problems, and a range of standard unconstrained optimization problems, including the most recent benchmark functions, each with unique characteristics. Thus, the objectives mentioned in Section 1 are met.

Once again it is clarified here that the objective of this paper is NOT to insult the researchers who have developed (and who are developing) metaphor-based optimization algorithms till now. The objective is to prove that there is no need to depend on metaphors to develop new optimization algorithms.

It is important to understand that the proposed BMR and BWR algorithms are not claimed as the "best" optimization algorithms available from all of the algorithms published in the optimization literature. It is plausible that an "optimal" algorithm may not exist for every type of optimization problem! However, the BMR and BWR algorithms demonstrate a great deal of potential for tackling optimization problems that are both constrained and unconstrained. As of right now, we can say that the BMR and BWR algorithms produce the best results in a comparatively small number of function evaluations, are simple to comprehend and apply, and have no algorithm-specific parameters. The codes of BMR and BWR algorithms are available at https://sites.google.com/view/bmr-bwr-optimization-algorithm/home.

The preliminary investigations serve as the foundation for the proposed algorithms' outcomes, that are given in this work. In-depth investigations are planned to be conducted in the upcoming days on more real-life constrained and unconstrained engineering problems. Testing the effectiveness of the proposed algorithms on a range of intricate and computationally demanding problems involving high dimensions as well as investigating the convergence behavior will be part of these investigations. The results will be compared with those of other well-known and well-established optimization algorithms, and statistical analyses will also be carried out. The application of BMR and BWR algorithms for fine-tuning and training deep neural networks in machine learning will also be investigated.

The optimization community researchers may attempt to enhance these two algorithms to make them significantly more potent. We hope that researchers from various technical and scientific fields—including the physical, biological, and social sciences—will find the BMR and BWR algorithms to be effective instruments for optimizing systems and processes. If certain flaws in these algorithms are found, the researchers may offer suggestions to get around the drawbacks.

Acknowledgment

The research is supported by the Department of Science and Technology (DST) of the Government of India under the Mathematical Research Impact Centric Scheme (MATRICS) with the project number MTR/2023/000071.

References

1. K. Sörensen, “Metaheuristics – the metaphor exposed”, *International Transactional in Operational Research*, vol. 22, pp. 3-18, 2015.
2. K. Sörensen, M. Sevaux and F. Glover, “A history of metaheuristics. In: Martí R, Pardalos P, Resende M (eds), *Handbook of heuristics*, Springer, pp. 791-808, 2018.
3. F. Campelo and C. Aranha, “Evolutionary computation bestiary”, *https:// github. com/ fcamp elo/ ECBestiary*, 2021, Version visited last on 8 July 2024.
4. C. L. C. Aranha, F. Villalón, M. Dorigo, R. Ruiz, M. Sevaux, K. Sörensen, and T. Stützle, “Metaphor-based metaheuristics, a call for action: the elephant in the room”, *Swarm Intelligence,* Vol. 16, pp. 1-6, 2021.
5. C. L. C. Villalón, T. Stützle, and M. Dorigo, “Grey wolf, firefly and bat algorithms: Three widespread algorithms that do not contain any novelty”, In: *International Conference on Swarm Intelligence*, Springer, pp. 121-133, 2020.
6. C. L. C. Villalón, T. Stützle, and M. Dorigo, “Cuckoo search ≡ (μ+λ)–evolution strategy — A rigorous analysis of an algorithm that has been misleading the research community for more than 10 years and nobody seems to have noticed”, *Technical Report TR/IRIDIA/2021-006*, IRIDIA, Université Libre de Bruxelles, Belgium, 2021.
7. R. V. Rao, “Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems”, *International Journal of Industrial Engineering Computations*, vol. 11, pp. 107-130, 2020.

8. M. Sarhani, S. Voß, and R. Jovanovic, “Initialization of metaheuristics: comprehensive review, critical analysis, and research directions”, *International Transactions in Operational Research*, vol. 30, pp. 3361-3397, 2023.
9. Rajwar, K. Deep, and S. Das, “An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges”, *Artificial Intelligence Review*, vol. 56, pp. 13187-13257, 2023.
10. R. Salgotra, P. Sharma, S. Raju, and A. H. Gandomi, “A contemporary systematic review on meta-heuristic optimization algorithms with their MATLAB and Python code reference”, *Archives of Computational Methods in Engineering,* vol. 31, pp. 1749-1822, 2024.
11. P. Sharma, and S. Raju, “Metaheuristic optimization algorithms: a comprehensive overview and classification of benchmark test functions”, *Soft Computing,* vol. 28, pp. 3123-3186, 2024.
12. L. Velasco, H. Guerrero, and A. Hospitaler, “A literature review and critical analysis of metaheuristics recently developed”, *Archives of Computational Methods in Engineering*, vol. 31, pp. 125-146, 2024.
13. B. Benaissa, M. Kobayashi, M. A. Ali, T. Khatir, and M. E. A. E. Elmeliani, “Metaheuristic optimization algorithms: An overview”, *HCMCOUJS-Advances in Computational Structures*, vol. 14, pp. 34-62, 2024.
14. A. Kumar, G. Wu, M. Z. Ali, R. Mallipeddi, P. N. Suganthan, and S. Das, “A test-suite of non-convex constrained optimization problems from the real-world and some baseline results”, *Swarm and Evolutionary Computation*, vol. 56, 100693, 2020.
15. A. Trivedi, D. Srinivasan, and N. Biswas, “An improved unified differential evolution algorithm for constrained optimization problems”, in: 2018 *IEEE Congress on Evolutionary Computation (CEC)*, *IEEE,* pp. 1–10, 2018.
16. M. Hellwig, and H.-G. Beyer, “A matrix adaptation evolution strategy for constrained real-parameter optimization”, in: *2018 IEEE Congress on Evolutionary Computation (CEC)*, *IEEE*, pp. 1–8, 2018.
17. Z. Fan, Y. Fang, W. Li, Y. Yuan, Z. Wang, X. Bian, “LSHADE44 with an improved ϵ constraint-handling method for solving constrained single-objective optimization problems, in: *2018 IEEE Congress on Evolutionary Computation (CEC), IEEE*, pp. 1–8, 2018.
18. J. Gurrola-Ramos, A. Hern´andez-Aguirre, and O. Dalmau-Cede˜no, “COLSHADE for real-world single-objective constrained optimization problems”, In: *2020 IEEE Congress on Evolutionary Computation (CEC), IEEE*, pp.1-8, 2020.
19. K. M. Sallam, S. M. Elsayed, R. K. Chakrabortty, and M. J. Ryan, “Multioperator differential evolution algorithm for solving real-world constrained optimization problems. In: *2020 IEEE Congress on Evolutionary Computation (CEC), IEEE*, pp. 1-8, 2020.
20. R. V. Rao and R. B. Pawar, “Improved Rao algorithm: A simple and effective algorithm for constrained mechanical design optimization problems”, *Soft Computing*, vol. 27, 3847-3868, 2022.
21. M. Ghasemi, K. Golalipour, M. Zare, S. Mirjalili, P. Trojovsky, L. Abbuligah, and R. Hemmati, Flood algorithm (FLA): an efcient inspired meta-heuristic for engineering optimization”, *Journal of Supercomputing*, doi: https://doi.org/10.1007/s11227-024-06291-7, 2024.
22. http://arxiv.org/abs/2407.11149, 2024.
23. X-S. Yang, “Ten new benchmarks for optimization”, in: Benchmarks and Hybrid Algorithms in Optimization and Applications (Ed. X-S Yang), *Springer Tracts in Nature-Inspired Computing*, pp. 19 – 32, 2023 (arXiv:2309.00644v1).