RESEARCH ARTICLE

Battle Royale Optimizer with Ring Neighborhood Topology

Taymaz Akan[1,3], **Tomáš Zálabský**[2], **Kimia Shirini**[4], **Jorge Ramos-Frutos**[5,*], **Diego Oliva**[6] and **Mohammad A. Bhuiyan**[1]

[1] Department of Medicine, Louisiana State University Health Sciences Center, Shreveport, LA 71103, United States
[2] Faculty of Electrical Engineering and Informatics, University of Pardubice, Pardubice, Czech Republic
[3] Software Engineering Department, Istanbul Topkapi University, Istanbul, Turkey
[4] Faculty of Electrical & Computer Engineering, University of Tabriz, Tabriz, Iran
[5] Depto. de Ingeniería Industrial, Tecnológico Nacional de México/Instituto Tecnológico de Jiquilpan, Jiquilpan, Michoacán, México
[6] Depto. de Ingeniería Electro-Fotónica, Universidad de Guadalajara, CUCEI, Guadalajara, Jalisco, México

Abstract

Recently, battle royale optimizer (BRO), a game-based metaheuristic search algorithm, has been proposed for continuous optimization, inspired by a genre of digital games known as "battle royale." In BRO, each individual chooses the nearest opponent as a competitor. For this purpose, the Euclidean distance between individuals is calculated. This interaction corresponds to an increase in computational complexity by a factor of n. To improve the computational complexity of BRO, a modified methodology is proposed using a ring topology, namely, BRO-RT. In the modified version, a set of individuals is arranged in a ring such that each has a neighborhood comprising several individuals to its left and right. Instead of a pairwise comparison with all individuals in the population, the best individual among the left and right neighborhoods is selected as the competitor. We compared the proposed scheme with the original BRO and six popular optimization algorithms. All algorithms are tested in several benchmark functions and engineering optimization problems. Experimental results show that the BRO-RT demonstrates competitive performance compared to nine state-of-the-art methods across most benchmark functions. Additionally, the compression spring design problem was utilized to assess the proposed method's ability to solve real-world engineering problems. These results demonstrate that BRO-RT yields promising results when applied to real-world engineering problems. Finally, while BRO is ranked first and BRO-RT second, they achieved competitive results; BRO-RT has the advantages of lower computational complexity and faster run times than the original BRO algorithm.

Keywords: Battle royale optimization, ring topology, optimization.

Submitted: 26 November 2025
Accepted: 16 January 2026
Published: 02 February 2026

Vol. 2, **No.** 1, 2026.
10.62762/TSEL.2025.751954

***Corresponding author:**
✉ Jorge Ramos-Frutos
jorgearmando.rf@jiquilpan.tecnm.mx

Citation
Akan, T., Zálabský, T., Shirini, K., Ramos-Frutos, J., Oliva, D., & Bhuiyan, M. A. (2026). Battle Royale Optimizer with Ring Neighborhood Topology. *ICCK Transactions on Swarm and Evolutionary Learning*, 2(1), 19–40.

1 Introduction

Optimization refers to the action of selecting the best possible solution among a variety of options. This selection can be performed using either exhaustive or stochastic approaches. Exhaustive methods are guaranteed to find the optimal solution to a problem because they search the entire state space of the problem. However, the costs associated with using these methods are prohibitively high for issues that are of a non-polynomial (NP) nature. Utilizing stochastic methods could be an alternative approach to solving this problem. Despite their widespread use, stochastic methods do not guarantee that the best solution will be identified. Metaheuristic algorithms employ stochastic methods to enhance their chances of finding the optimal solution to a problem. Metaheuristic optimization algorithms, which are used to address various types of problems despite their constraints, have played a significant role in solving numerous practical problems in scientific and industrial applications. In recent years, metaheuristic algorithms have been used widely to solve optimization problems [1]. Numerous optimization algorithms have been employed to address various real-world challenges. In general, the problems can be solved using exhaustive or stochastic schemes. For specific inputs, exhaustive algorithms always give the same outputs, while stochastic algorithms may generate different results because of random operators [2].

Nature-inspired metaheuristic optimization tools can be categorized into three main groups: evolutionary algorithms (EAs), swarm intelligence (SI), and physical phenomena. A fourth category, a new optimization tool called game-based, has recently been proposed by Farshi [3]. EAs mimic the processes in Darwin's theory of biological evolution. These algorithms benefit from common mechanisms of mutation, selection, and recombination operations. Some popular EA paradigms are the Genetic Algorithm (GA) [4], Evolution Strategies (ES) [5], Tabu Search (TS) [6], Simulated Annealing (SA) [7], Biogeography-Based Optimizer (BBO) [8], and Bird Mating Optimizer (BMO) [9], among others. SIs rely on the power of the collective intelligent activity of different living things, such as humans, animals, bacteria, etc. While each individual within this algorithm is not competent, the collective behavior of a group of individuals generates an intelligent mechanism. Some examples of SI-based algorithms include Particle Swarm Optimization (PSO) [10], Ant Colonies (AC) [11], Bee Colonies (BC) [12], Animal Migration Optimization (MGO) [13], Hawks Hunting (HH) [13], Human Mental Search (HMS) [14], and Selfish Herd Optimizer (SHO) [15]. Physics-based methods have been inspired by physical laws such as inertial, electromagnetic, and gravitational forces. Some famous physics-based algorithms are Quantum-Inspired Particle Swarm Optimization (QPSO) [16], Big Bang-Big Crunch (BB-BC) [17], and Hysteretic Optimization (HO) [18]. The difference between the SI algorithms and game-based algorithms is that, unlike the populations in SI algorithms, the population in a game-based algorithm does not have to collaborate to achieve its goal; the individuals enter into conflict with each other instead of cooperating.

Each optimization algorithm has its exploration and exploitation strategies. The exploration and exploitation mechanisms are fundamental in determining the efficiency of an optimization algorithm. A good balance between exploration and exploitation provides an efficient optimization algorithm [19]. Exploration involves probing for solutions across a wide portion of the problem space in a global search that escapes from local optima. In contrast, exploitation involves a local search by exploring solutions in the neighborhood of the best solution so far [20]. If the weight parameter of the inertia is too large, an algorithm can fail to achieve or converge on the optimal solution.

Various algorithms have been proposed over the last two decades for several real-world optimization problems. One of the most frequently asked questions about optimization algorithms is: Is there a need for new optimization algorithms, given the vast number of commonly used and well-established algorithms already available? The no-free-lunch theorem (NFL) [21] claims that no specific optimization algorithm outperforms any other in dealing with all kinds of optimization problems. However, it is also clear that the real world will face various optimization issues that have not yet been identified, let alone addressed. As a result, the performance of existing algorithms in addressing such matters is unknown so any standard algorithm may be superior to any other method. Consequently, the emergence of new algorithms is inevitable. However, the significant shortcoming of many optimization algorithms is that they do not contain novelties [22, 23]. Also, many variants extend the existence of optimization algorithms to improve their search capability [24–27].

The Battle Royale Optimizer (BRO) is a new population-based metaheuristic technique for single-objective optimization across continuous problem spaces [28]. The BRO is not just a population-based algorithm; it also encompasses a new category of algorithms known as game-based optimization algorithms, in which players actively explore their environments and compete against one another to achieve a win. In contrast to SI-based algorithms, players do not work together toward a common goal of victory here. Soldiers, or players, represent the solutions in the BRO algorithm; the algorithm's optimal solution is the solution that defeats all of its neighbors while staying alive. Recently, the binary version of this approach, known as BinBRO, was proposed [28, 29]. BRO has been used to address optimization problems [30, 31]. Our research employed a modified version of the BRO algorithm to enhance computational complexity by incorporating ring neighborhood topology (BRO-RT).

The need for low computational cost when optimizing nondeterministic polynomial (NP) problems is essential in the processes to be calculated [32, 33]. The intrinsic time requirements of algorithms play an important role in computational complexity. One of the strengths of BRO-RT is that it requires less time, which has led to its widespread use to address optimization problems with exponential complexity. For example, using the BRO algorithm to measure the distance between individuals in high-dimensional problems is time-consuming. This shortcoming has been addressed in BRO-RT using a ring topology. For experimental purposes, the BRO-RT has been tested.

The remainder of the sections are organized as follows: Section 2 describes the basic concept of the standard BRO. Section 3 introduces the BRO using ring topology. Section 4 describes the experimental results, and finally, Section 5 discusses some conclusions.

2 Battle royale optimizer

The original Battle Royale game was based on the 2000 Japanese action movie called Battle Royale. Its popularity led to the creation of many more games in the battle royale genre that have also become very popular, such as Player Unknown's Battlegrounds (PUBG), Call of Duty: Warzone, Apex Legends, Counter-Strike: Global Offensive, and Ring of Elysium. In these games, the match begins with a predetermined number of players, who are randomly distributed across a game map. In the simulation, the problem space is considered a game map, and the search agent is viewed as a player. These games can be played in solo, duo, and squad modes. Unlike duo and squad modes, in solo mode, players have no teammates. The most crucial goal of this game has two parts: players must defeat their opponents while also staying within a safe zone on the game map. Players encountering other players are most likely to enter into conflict with them.

Like in Battle Royale games, the BRO algorithm spreads its initial set of candidate solutions across the problem space at random. Moreover, the best solution is kept in a variable called flag. Afterward, the flag will be updated with each iteration to realize elitism. Everybody shoots at everyone else to try to make them hurt. Players who better position themselves can inflict damage on their opponents. Given this information, one might wonder who or what can shoot their opponent while they are playing the game. In the simulation, since each individual faces the closest another individual, the Euclidean distance of each solution is calculated with the rest of the solutions. The solution with a better cost function can overcome the competitor and be recognized as the winner and the other one as the loser. When one player injures another player, the damage level of the injured player (loser) is increased by one. This interaction is simulated as $x_i{\cdot}damage = x_i{\cdot}damage{+}1$, where $x_i{\cdot}damage$ denotes the damage level of the damaged player. However, if a solution injures the opponent, the damage level resets to zero. Meanwhile, injured players look for better positions to avoid being hit. To put it another way, soldiers want to change their position as soon as they take damage so that they can attack opponents from a different side. As a consequence of this, the damaged soldier moves toward a point that is located somewhere between its previous position and the position that has been determined to be the best so far (best player). In the simulation, a solution with a worse fitness value than its nearest neighbor would move towards the best position as follows [34, 35]:

$$x_{dam,d} = x_{dam,d} + r_2(x_{best,d} - x_{dam,d}) \tag{1}$$

However, [28] extends this with a new movement strategy as follows:

$$C_{dam,d} = r_1 x_{best,d} + r_2(C_{dam,d} - x_{dam,d}) \tag{2}$$

and

$$x'_{dam,d} = x_{dam,d} + C_{dam,d} \tag{3}$$

where $x_{best,d}$ is the position of the best solution found so far. $x_{dam,d}$ is the position of the damaged individual,

$C_{dam,d}$ indicates the movement step of the damaged individual in dimension d, $x'_{dam,d}$ is the updated position of the damaged individual in dimension d, and r_1 and r_2 indicate a randomly generated number uniformly distributed in $[0, 1]$. In the original BRO algorithm, r_1 was specified as one. It is clear that if the player is too injured, it will die. In the event of a player's death as a consequence of an opponent, he will respawn on the battleground at a completely random location within the zone. Therefore, if the damage level exceeds a predefined threshold value, the solution may be stuck in local optima. To escape from the local optima and provide exploration, the solution is re-initialized in a feasible problem space as follows:

$$x'_{dam,d} = r(ub_d - lb_d) + lb_d \tag{4}$$

During the game, the safe zone shrinks continually, so players cannot run away from the battle to survive. In the simulation, the balance between exploration and exploitation is provided by this interaction so that the upper- and lower-bound problem space shrinks toward the best solution found so far. Hence, in every Δ iteration, the search boundary of the variables of the problem shrinks toward the global optimum solution. Herein, the initial value Δ is:

$$\frac{MaxCicle}{round(\log_{10}(MaxCicle))} \tag{5}$$

where $MaxCicle$ denotes the maximum number of generations. Consequently, $\Delta = \Delta + \lceil \frac{\Delta}{2} \rceil$ if $i \geq \Delta$, where i is the current iteration number. The reduction of the problem space is performed as follows:

$$\begin{aligned} lb_d &= x_{best,d} - SD(x_d) \\ ub_d &= x_{best,d} + SD(x_d) \end{aligned} \tag{6}$$

where lb_d and ub_d respectively are the lower and upper bounds of dimension d in the problem space and $SD(x_d)$ represents the population's standard deviation in dimension d. Also, another shrink-the-search region mechanism was proposed in [36].

3 BRO using ring topology

In Particle Swarm Optimization (PSO) algorithms, the ring neighborhood topology is a common approach. In this topology, particles are arranged in a circular pattern, with each particle connected to a certain number of its neighbors. This allows particles to exchange information and influence each other's behavior by establishing a network of structured communication [37–39].

The BRO algorithm calculates Euclidean distance by comparing each individual to its closest neighbor in each iteration. Computational complexity increases by n due to this interaction. Figure 1 shows how the BRO-RT algorithm uses ring neighborhood topology to avoid extra computation. This figure shows that all individuals have left and right neighbors. Ring topology compares the neighborhood instead of individuals to find the nearest neighbor.

First, the population is arranged in a circular array with a length of N. Next, the length of the neighborhood is set to five. If j is the index of an individual, then the neighborhood is composed of $j-2$,$j-1$,j,$j+1$, and $j+2$ indexes (See Figure 1). As it is clear, a set of neighbors for the j^{th} individual is always the same. In other words, every individual has and maintains its ring as a definition of its local neighborhood. If the neighbor's index exceeds N, it will return to zero, and vice versa: it will return to N if it reaches below zero. Therefore, each individual has some number of individuals to the left and right in the neighborhood. The index connects the ring's members. Additionally, the ring topology may facilitate population information exchange. Finally, instead of a pairwise comparison with all individuals in the population, the best individual from the left and right neighborhoods is chosen as the competitor, whose index will be variables i and $i \neq j$. The competitor is selected solely based on objective function value within the fixed ring neighborhood; no distance-based criterion is employed, and the original BRO update equations remain unchanged. The ring topology generates the positions of exemplars according to the following proposition:

- Let p_j be a proposition stating that "the individual at index j is the current individual in consideration".
- Let L_j be a proposition stating that "the best individual from the left neighborhood of individual j is selected".
- Let R_j be a proposition stating that "the best individual from the right neighborhood of individual j is selected".
- Let B_i be a proposition stating that "the individual at index i is the best competitor selected".
- Let C be a proposition stating that "the circular

Table 1. Unimodal benchmark test functions.

Function	Name	Range	Shift position	f_{min}
$f_1(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	Sphere	$[-100, 100]$	$[-30, \cdots, -30]$	0
$f_2(\mathbf{x}) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	Schwefel 2.20	$[-10, 10]$	$[-3, \cdots, -3]$	0
$f_3(\mathbf{x}) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	Rotated hyper-ellipsoids	$[-100, 100]$	$[-30, \cdots, -30]$	0
$f_4(\mathbf{x}) = \max_{i=1,\cdots,n} \lvert x_i \rvert$	Schwefel 2.21	$[-100, 100]$	$[-30, \cdots, -30]$	0
$f_5(\mathbf{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	Rosenbrock	$[-30, 30]$	$[-15, \cdots, -15]$	0
$f_6(\mathbf{x}) = \sum_{i=1}^{n} (x_i + 0.05)^2$	Step	$[-100, 100]$	$[-750, \cdots, -750]$	0
$f_7(\mathbf{x}) = \sum_{i=1}^{n} i x_i^4 + rand[0, 1]$	Quartic	$[-128, 128]$	$[-25, \cdots, -25]$	0

Table 2. Multimodal benchmark test functions.

Function	Name	Range	Shift position	f_{min}
$f_8(\mathbf{x}) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	Schwefel	$[-500, 500]$	$[-300, \cdots, -300]$	$-418.9829 \times dim$
$f_9(\mathbf{x}) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right)$	Rastrigin	$[-5.12, 5.12]$	$[-2, \cdots, -2]$	0
$f_{10}(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + \exp(1)$	Ackley	$[-32, 32]$		0
$f_{11}(\mathbf{x}) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	Griewank	$[-600, 600]$	$[-400, \cdots, -400]$	0
$f_{12}(\mathbf{x}) = \frac{\pi}{n} \cdot \{10\sin^2(\pi y_i) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})\} + \sum_{i=1}^{n} u(x, 10, 100, 4)$, $y_i = 1 + \frac{1}{4}(x_i + 1)$, $u(x, 10, 100, 4) = \begin{cases} k(x_i - a)^m, & \text{if } x_i > a \\ 0, & \text{if } -a < x_i < a \\ k(-x_i - a)^m, & \text{if } x_i < -a \end{cases}$	Penalized	$[-50, 50]$	$[-30, \cdots, -30]$	0
$f_{13}(\mathbf{x}) = 0.1[\sin^2(3\pi x) + (x_i - 1)^2(1 + \sin^2(3\pi y)) + (x_n - 1)^2(1 + \sin^2(2\pi x_n))] + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	Levy	$[-50, 50]$	$[-100, \cdots, 100]$	0

array wraps around such that if the neighbor's index exceeds N, it will return to zero, and if it reaches below zero, it will return to $N - 1$".

Now, assuming the best individual from either the left or right neighborhood is to be selected as the competitor for the current individual j, we can represent this using the following propositional logic statement:

$$p_j \rightarrow (L_j \vee R_j \wedge B_i) \tag{7}$$

This states that if p_j (we are considering the individual at index j, then either L_j (the best from the left neighborhood is selected) or R_j (the best from the right neighborhood is selected) must be true, and as a result, B_i (individual i is the best competitor selected) is true. For the wrapping around the circular array, we

Table 3. Fixed-dimension multimodal benchmark functions.

Function	Dim	Range	f_{min}
$f_{14}(\mathbf{x}) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i + a_{ij})^6}\right)^{-1}$	2	$[-65, 65]$	1
$f_{15}(\mathbf{x}) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	$[-5, 5]$	3×10^{-4}
$f_{16}(\mathbf{x}) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316
$f_{17}(\mathbf{x}) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	2	$[-5, 5]$	0.3980
$f_{18}(\mathbf{x}) = [1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]$	3
$f_{19}(\mathbf{x}) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	$[1, 3]$	-3.86
$f_{20}(\mathbf{x}) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6	$[0, 1]$	-3.32
$f_{21}(\mathbf{x}) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.1532
$f_{22}(\mathbf{x}) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.4028
$f_{23}(\mathbf{x}) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.5363

could express:

$$C \leftrightarrow ((j < 0 \rightarrow j = N - 1) \vee (j > N - 1 \rightarrow j = 0)) \tag{8}$$

This says that C (the array wraps around) is true if and only if both conditions for wrapping below zero or above $N - 1$ are satisfied.

The ring topology uses four neighboring individuals to produce diversity-enhanced exemplars, which can improve the exploration ability.

4 Experimental results and evaluation

The proposed modified BRO algorithm has been compared with the original BRO and six other well-known metaheuristic optimization algorithms: Genetic Algorithm (GA) [46], Particle Swarm Optimization (PSO) [40], Artificial Bee Colony (ABC) [12], Gravitation Search Algorithm (GSA) [48], Firefly Algorithm (FA) [49], and Differential Evolution (DE) [47]. Since different test functions are capable of challenging an algorithm in different ways, different test functions with diverse characteristics should be used. For this reason, all algorithms have been evaluated using 24 well-known continuous benchmark functions with different properties selected from among classical functions applied by many researchers, CEC2005 [43] and CEC2010 [44]. These functions are listed in Tables 1, 2, 3, and 4. These benchmarks have different properties. One of the properties is the shift position that indicates a fixed shift vector $\mathbf{o}$, where all components take the same constant value. This vector is used to translate the global optimum away from the origin according to $\mathbf{z} = \mathbf{x} - \mathbf{o}$, following standard benchmark definitions. While all test functions are formulated as minimization problems, $f_1 - f_7$ are unimodal, while $f_8 - f_{23}$ are multimodal. Furthermore, f_3, f_5, f_{12}, and f_{13} are non-symmetric, while the rest of the functions are symmetric. Also, f_9,f_{10}, and f_{11} are evenly distributed whereas f_8, f_{12}, and f_{13} are unevenly distributed. Finally, f_{24}-f_{27} are composite

Table 4. Composite benchmark functions.

Function	Dim	Range	f_{min}
f_{24} (CF1):	30	$[-5, 5]$	0
$f_1, f_2, \ldots, f_{10}$ = Sphere Function			
$[\sigma_1, \sigma_2, \ldots, \sigma_{10}] = [1, 1, \ldots, 1]$			
$[\lambda_1, \lambda_2, \ldots, \lambda_{10}] = [5/100, 5/100, \ldots, 5/100]$			
f_{25} (CF2):	30	$[-5, 5]$	0
$f_1, f_2, \ldots, f_{10}$ = Griewank's Function			
$[\sigma_1, \sigma_2, \ldots, \sigma_{10}] = [1, 1, \ldots, 1]$			
$[\lambda_1, \lambda_2, \ldots, \lambda_{10}] = [5/100, 5/100, \ldots, 5/100]$			
f_{26} (CF3):	30	$[-5, 5]$	0
$f_1, f_2, \ldots, f_{10}$ = Griewank's Function			
$[\sigma_1, \sigma_2, \ldots, \sigma_{10}] = [1, 1, \ldots, 1]$			
$[\lambda_1, \lambda_2, \ldots, \lambda_{10}] = [5/100, 5/100, \ldots, 5/100]$			
f_{27} (CF4):	30	$[-5, 5]$	0
f_1, f_2 = Ackley's Function, f_3, f_4 = Rastrigin's Function			
f_5, f_6 = Weierstrass Function, f_7, f_8 = Griewank's Function			
f_9, f_{10} = Sphere Function			
$[\sigma_1, \sigma_2, \ldots, \sigma_{10}] = [1, 1, \ldots, 1]$			
$[\lambda_1, \lambda_2, \ldots, \lambda_{10}] = [5/100, 5/100, \ldots, 5/100]$			

functions. Figure 2 shows the search space for all tested benchmark functions in two-dimensional form.

Since metaheuristics carry out a form of stochastic optimization, the estimated solutions are affected by the set of random variables generated. To alleviate the effect of randomness, the statistical evaluations were performed by taking the average fitness value over 25 independent runs. Furthermore, the compression spring design problem was also used to estimate the proposed method's ability to solve real-world engineering problems. The statistical evaluations for the compression spring design problem were also performed by taking the average fitness value over 25 independent runs. Apart from the mean value, the best, worst, median, standard deviation,

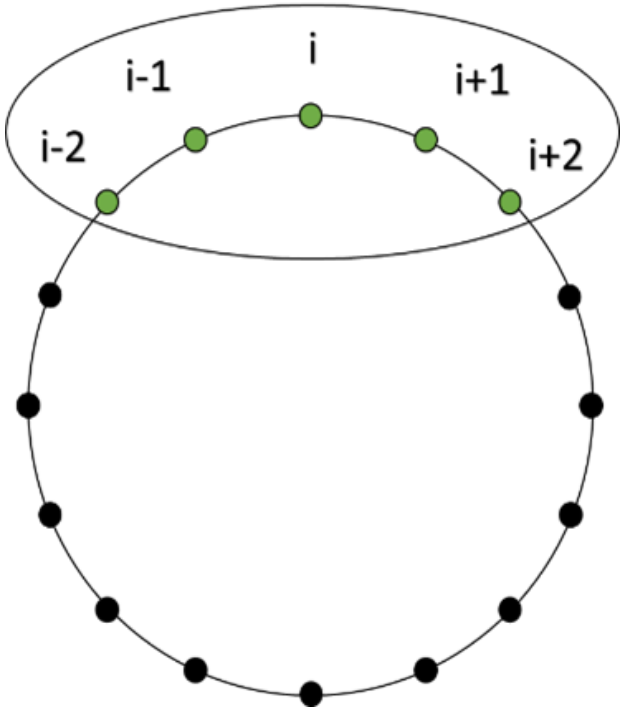


Figure 1. Plot of the local neighborhood of an individual.

and algorithm run times were considered in the evaluations. Experiments were performed on a single machine with a 2.81 GHz Intel Core i7 processor and 32 GB of memory to provide a fair assessment. The algorithms were coded in MATLAB R2020b. A swarm population size of 200 was used for all algorithms to optimize all test functions. Finally, the number of maximum iterations was set to 500. The best control parameters for all algorithms were tuned by trial and error or taken directly from the original papers. These parameters are listed in Table 5.

Tables 6, 8, 10, and 12 summarize all numerical outcomes for all performance criteria for test functions. For the sake of clarity, the best results are in bold. The approaches are then ranked in terms of performance criteria for each function (see Tables (7, 9, 11, and 13)). The best-averaged rank is also in bold. Finally, the average of each algorithm's ranks across all performance criteria is shown in the last row of the table.

As shown in Table 6, all algorithms were tested in seven separate unimodal functions. Based on the overall results, BRO has the most successes across most performance criteria. With respect to f_1, BRO performed better than its competitors except for run time. On the other hand, BRO-RT yielded extremely competitive results with BRO; the run time of BRO-RT is 1.9883 seconds faster than that of BRO. According to this function, GSA ranked third, but this algorithm's

Table 5. The initial values of the parameters of optimization algorithms.

Algorithms	Parameter	Value
BRO & BRO-RT		
	Max Damage Level	4
GA	Crossover probability	0.8
	Mutation probability	0.1
	Selection mechanism	Roulette wheel
	Crossover type	Whole Arithmetic Crossover
	Mutation type	Uniform mutation
PSO	Inertia weights range	$[0.9, 0.6]$
	acceleration coefficients	2.1 and 2.1
ABC	Limit	8
GSA		20
	Gravitational constant	100
	Velocity range	$rand[0, 1]$
WOA	linearly decrease from	[2 to 0] $rand[0, 1]$
DE	Crossover probability	0.55
	Differential weight	0.5

run time was not acceptable. For f_2, similar to the results for f_1, BRO outperformed the others, but BRO-RT closely followed the results of BRO with a better run time. Therefore, despite satisfactory results, the algorithm's run time was not acceptable. It is evident that the same situation observed for these two functions exists in the other functions as well.

Upon closer analysis, the mean and best BRO tests were successful five out of seven times. As shown in Table 8, all algorithms were put to the test in six different multimodal functions. The results indicate that the outcomes using BRO-RT were competitive. The data in Table 9 demonstrate that BRO-RT ranks first overall, while BRO and FA rank second and third, respectively. PSO ranks fourth alongside GSA. ABC, DE, and GA rank sixth to last, respectively. It is clear that the outcomes of BRO and BRO-RT are competitive; however, BRO-RT runs faster due to the ring topology. Although the FA algorithm performed well, it is too slow, rendering its performance unacceptable in practical terms. In addition, Table 10 demonstrates that BRO-RT is the overall winner, with BRO, FA, PSO, ABC, GSA, DE, and GA ranking second to last, respectively.

In terms of f_8, it is clear that PSO outperformed its competitors. FA ranked second, but the run time was not acceptable. BRO and BRO-RT yielded extremely competitive results and ranked third and fourth, respectively. With respect to f_9, it is clear that BRO and BRO-RT performed far better than the other algorithms. The mean values of BRO-T and BRO are $1.36859e-01$ and 1.636288, respectively. The corresponding values for GA, DE, PSO, ABC, FA, and GSA are 375.3837, 391.2683, 151.578, 125.359, 73.7860, and 7.24330. It is worth noting that BRO-RT obtained better results than BRO in less time. Also, in terms of f_{10} and f_{11}, BRO and BRO-RT outperformed others, followed by GSA. For f_{12} and f_{13}, FA surpassed the other algorithms, but the running time of this algorithm is considerably longer than that of the others.

The convergence curves of the eight algorithms for the thirteen benchmark functions are shown in Figure 3. This figure demonstrates that, in almost every case, the convergence curve of BRO plunges rapidly. Furthermore, it is clear that DE and ABC are very competitive and behave quite similarly. As shown in Table 8, the performance of the optimization algorithms varies across different objective functions. While the BRO algorithm performs well in some cases, it shows higher variability. Other algorithms like BRO-RT, GA, DE, PSO, ABC, and FA also demonstrate competitive performance, with slight differences in their best solutions and execution times. The choice of the most suitable algorithm depends on the specific optimization problem and its characteristics. For Objective Function f_{14}, the BRO-RT, BRO, GA, DE, PSO,

Table 6. Outcomes for unimodal benchmark functions when the dimension is 30.

F	Metrics	BRO-RT	BRO	GA	DE	PSO	ABC	FA	GSA
F1	mean	9.89E-20	**5.87E-23**	4391.778	239.261	10.7531	572.051	2.27E-07	2.87E-18
	median	1.56E-20	**3.35E-23**	3960.613	238.928	6.88792	466.242	2.32E-07	2.81E-18
	best	3.25E-22	**3.11E-25**	2740.399	184.58	1.57468	195.224	1.94E-07	1.75E-18
	worst	1.22E-18	**2.75E-22**	6659.923	301.243	87.857	1594.3	2.56E-07	5.16E-18
	SD	2.56E-19	**8.26E-23**	1113.52	34.8715	17.0463	346.202	1.75E-08	7.24E-19
	time	3.076579	5.06492	1.405496	3.75196	1.40959	**0.55398**	209.252	19.485
F2	mean	1.18E-12	**3.77E-14**	42.5415	16.6453	8.36035	9.71763	2.58E-04	1.18E-08
	median	1.04E-12	**3.05E-14**	40.25567	16.7377	5.72143	9.54642	2.59E-04	1.17E-08
	best	3.65E-13	**1.20E-14**	30.51653	12.9923	3.19271	4.90615	2.21E-04	9.60E-09
	worst	2.72E-12	**8.55E-14**	73.327	19.4763	25.0081	14.3522	2.78E-04	1.38E-08
	SD	6.26E-13	**2.30E-14**	9.180281	1.92487	5.82822	2.29684	1.17E-05	1.10E-09
	time	3.093874	5.021733	1.471881	3.62379	1.39673	**0.60891**	213.424	19.2101
F3	mean	19.77956	22.6468	10866.67	119796.1	6359.11	59579.9	**7.71E-04**	412.063
	median	1.244508	**5.02E-06**	10529.46	120167.8	6219.16	59024.7	6.20E-04	395.923
	best	1.02E-02	**3.48E-09**	5302.346	97353.6	3.952.38	46390.1	1.60E-04	298.493
	worst	1.89E+02	5.23E+02	22029.5	141630.5	1.1553.7	76673	**2.21E-03**	590.187
	SD	4.60E+01	1.04E+02	3769.457	10719.85	1706.471	6346.64	**6.13E-04**	81.9464
	time	3.013068	6.35402	1.344457	3.844894	1.336824	**0.56482**	230.617	19.279
F4	mean	0.4692125	**3.70E-05**	29.08226	74.9011	24.2813	67.1112	2.13E-04	1.31569
	median	7.25E-03	**4.76E-06**	27.84905	75.083	23.3449	67.6384	2.15E-04	1.34119
	best	1.10E-04	**2.15E-08**	20.54251	66.2556	16.2397	58.6671	1.90E-04	0.20446
	worst	7.770726	3.49E-04	48.85631	79.6231	33.6071	72.0583	**2.37E-04**	2.51112
	SD	1.647391	**8.33E-05**	6.079352	2.94754	4.12026	3.43772	1.29E-05	0.53355
	time	6.941026	5.067939	1.373684	3.37252	1.41172	**0.50484**	201.168	19.2023
F5	mean	47.446	**46.9953**	1216223	246158.6	991.961	48805.12	57.08101	48.3232
	median	47.28329	**46.8696**	1164298	233895.9	610.214	38016.14	36.77909	47.6299
	best	46.6425	**46.327**	406005.7	151182.9	304.939	12857.43	35.33959	47.1232
	worst	48.75226	**48.0083**	2552054	418990.6	6782.41	172208.1	95.65826	61.8891
	SD	0.566056	**0.553441**	528944.7	59713.12	1307.14	36542.03	28.1458	2.87154
	time	3.761625	6.150595	1.383168	3.8866	1.362195	**1.187242**	259.6817	19.1355
F6	mean	4.266958	3.50669	4999.388	251.863	10.1068	542.131	2.28E-07	**2.74E-18**
	median	4.374491	3.373831	4651.235	249.14	8.48404	437.848	2.29E-07	**2.74E-18**
	best	2.314128	2.772362	1730.207	201.74	1.28578	212.998	1.88E-07	**2.05E-18**
	worst	6.180951	5.002485	8500.485	300.979	30.3469	1357.14	2.57E-07	**3.73E-18**
	SD	0.924035	0.575598	1518.946	25.3621	8.51881	314.533	1.81E-08	**4.42E-19**
	time	3.081463	5.095228	1.370656	3.32608	1.25716	**0.51172**	205.053	19.2512
F7	mean	1.09E-03	**3.13E-04**	1.432707	0.497741	0.195812	2.73182	8.81E-04	5.03E-03
	median	8.87E-04	**2.09E-04**	1.33887	0.474839	0.183945	2.57166	8.39E-04	4.30E-03
	best	2.87E-04	**1.79E-05**	0.573331	0.324987	0.121315	1.42377	4.87E-04	1.94E-03
	worst	3.03E-03	**1.24E-03**	2.323878	0.628954	0.334519	4.81694	1.24E-03	1.18E-02
	SD	7.43E-04	3.22E-04	0.49384	0.086247	0.053036	0.74299	**2.16E-04**	2.59E-03
	time	4.176776	5.055593	2.167874	4.540451	2.0153	**1.63726**	336.88	19.9882

Table 7. Ranks of algorithms for unimodal benchmark functions.

Metrics	BRO-RT	BRO	GA	DE	PSO	ABC	FA	GSA
mean	2	1	8	6	5	6	3	4
median	3	1	8	6	5	6	2	4
best	2	1	8	6	5	6	3	4
worst	3	1	8	6	5	6	2	4
SD	3	1	8	6	5	7	2	4
time	4	6	3	5	2	1	8	7
Average rank	2,833,333	1,833,333	7,166,667	5,833,333	4,5	5,333,333	3,333,333	4,5
Overall rank	2	**1**	8	7	4	6	3	4

ABC, and FA algorithms show similar performance, with mean and median values close to each other. They all achieve the same best solution, except for the BRO algorithm, which exhibits higher variability. The convergence curves indicate that BRO-RT is competitive in terms of convergence speed, but its advantage is not consistent across all functions.

The standard deviation is small for most algorithms, except for GSA. The BRO-RT algorithm has the fastest execution time. Moving on to Objective Function F15, BRO-RT, BRO, GA, DE, PSO, ABC, and FA algorithms

Table 8. Outcomes for multimodal benchmark functions when dimension is 30.

F		BRO-RT	BRO	GA	DE	PSO	ABC	FA	GSA
F8	mean	-5.71E+03	-4.99E+03	-5382.12	-8534.37	**-3.53E+08**	-15025.1	-3.73E+05	-4241.5
	median	-5.70E+03	-4.73E+03	-5253.44	-8454.39	**-1.97E+07**	-14870.7	-3.69E+05	-4249.24
	best	-6.97E+03	-6.74E+03	-6584.45	-9474.46	**-3.54E+09**	-16271.3	-5.13E+05	-5034.36
	worst	-4.45E+03	-4.31E+03	-4624.65	-7938.29	**-2.29E+02**	-14291.2	-2.45E+05	-3532.09
	SD	6.15E+02	6.07E+02	430.7446	**361.6365**	8.15E+10	451.9381	8.38E+04	391.6468
	time	3.236725	5.02087	1.442667	3.572773	1.35358	**0.895489**	231.5539	19.3615
F9	mean	**1.37E-01**	1.636288	375.3837	391.2683	151.578	125.359	73.786	7.2433
	median	**0**	**0**	375.101	395.9235	154.244	126.667	68.6521	6.96471
	best	**0**	**0**	314.1879	358.9837	115.496	93.7395	44.7731	0.994959
	worst	**2.41392**	4.09E+01	442.6328	411.4047	198.525	153.82	137.3039	11.93951
	SD	**5.15E-01**	8.181439	34.31645	13077453	21.2774	17.8855	21.27878	2.821781
	time	3.620391	4.98046	1.495178	3.62615	1.31506	**0.93893**	221.576	19.28205
F10	mean	2.45E-11	**7.41E-13**	11.4562	5.23008	4.12624	9.73673	8.68E-05	1.01E-09
	median	1.57E-11	**6.83E-13**	11.47951	5.23397	4.04327	9.94132	8.74E-05	9.84E-10
	best	6.94E-12	**1.11E-13**	9.86301	4.67643	2.73075	7.17064	7.79E-05	8.29E-10
	worst	1.29E-10	**1.92E-12**	14.02094	5.6451	6.6873	11.6284	9.17E-05	1.21E-09
	SD	2.52E-11	**4.24E-13**	0.894059	0.21023	0.9119	1.12242	3.64E-06	1.09E-10
	time	3.661268	5.157007	1.429408	3.6979	1.28834	**0.84873**	222.009	19.2007
F11	mean	**0**	**0**	19.00958	3.22116	1.04902	5.68854	2.78E-07	71.0519
	median	**0**	**0**	18.92544	3.16782	1.05459	4.55041	2.76E-07	71.15
	best	**0**	**0**	13.17232	2.70994	0.59566	2.41973	2.12E-07	58.0763
	worst	**0**	**0**	24.94573	3.89793	1.29285	10.3612	3.41E-07	89.7934
	SD	**0**	**0**	3.530459	0.31253	0.14579	2.31052	3.18E-08	7.4951
	time	3.909983	5.535054	1.550043	4.04207	1.54122	**1.044**	239.424	19.4032
F12	mean	0.148087	0.2722615	12578.28	69894.36	44.705	3.39743	**3.12E-10**	3.96732
	median	0.135403	0.2807726	4434.572	61623.55	46.3028	2.88565	**3.10E-10**	4.00574
	best	9.63E-02	0.1518162	24.09496	986.1339	29.2854	0.38921	**2.72E-10**	2.02784
	worst	0.250838	0.3991045	98544.08	193054	62.3999	12.1385	**3.73E-10**	5.23723
	SD	3.96E-02	0.0635199	20716.89	50368.31	8.87854	2.35947	**2.72E-11**	0.86778
	time	5.889444	7.506429	1.953383	5.87513	**1.56256**	2.89062	436.174	19.7423
F13	mean	2.725685	1.541302	493830.3	443010.3	97.3785	1290.9	**8.12E-09**	16.5465
	median	2.775046	1.54433	377427.8	429121.8	95.7855	45.9665	**8.17E-09**	16.9436
	best	1.827252	6.74E-01	24980.59	199280.4	66.3891	3.42733	**6.85E-09**	11.0854
	worst	20.539404	2.943649	1259385	781729.6	122.482	11094.2	**9.41E-09**	21.4251
	SD	4.46E-01	6.27E-01	330224.6	147595.9	15.2666	2876.67	**5.97E-10**	3088704
	time	5.21362	7.194583	1.853909	5.95661	**1.54294**	2.90933	462.095	19.9154

Table 9. Ranks of algorithms for multimodal benchmark functions.

Metrics	BRO-RT	BRO	GA	DE	PSO	ABC	FA	GSA
mean	1	3	8	7	4	5	2	5
median	1	2	8	7	4	5	2	6
best	1	1	8	7	5	4	3	6
worst	1	3	8	7	4	6	1	5
SD	1	2	8	6	7	5	3	4
time	4	6	3	5	1	1	8	7
Average rank	1.5	2.833333	7.166667	6.5	4.166667	4.333333	3.166667	5.5
Overall rank	**1**	2	8	7	4	5	3	6

exhibit comparable performance with similar mean and median values. The BRO-RT algorithm finds the best solution slightly lower than other algorithms. The standard deviation is relatively high for GA and ABC. PSO has the lowest execution time among the algorithms. For Objective Function f_{16}, BRO-RT, BRO, GA, DE, PSO, ABC, and FA algorithms all converge to the same solution. The standard deviation is relatively small across the algorithms. GA, PSO, and ABC have the highest execution times. Analyzing Objective Function f_{17}, BRO-RT, BRO, GA, DE, PSO, ABC, and FA algorithms produce similar mean, median, and best solutions. The standard deviation is relatively small for all algorithms. GA, PSO, and ABC have the highest execution times. Considering the Objective Function f_{18}, BRO-RT, BRO, DE, PSO, and FA algorithms find the same best solution, while GA and ABC deviate slightly. The standard deviation is relatively small across all algorithms. GA, PSO, and ABC have the highest execution times. Lastly, for Objective

Table 10. Outcomes for fixed-dimension multimodal benchmark functions.

F		BRO-RT	BRO	GA	DE	PSO	ABC	FA	GSA
F14	mean	9.98E-01	9.98E-01	1.020103	**9.98E-01**	**9.98E-01**	**9.98E-01**	**9.98E-01**	2.445938
	median	9.98E-01	9.98E-01	9.98E-01	**9.98E-01**	**9.98E-01**	**9.98E-01**	**9.98E-01**	2.355065
	best	9.98E-01	9.98E-01	9.98E-01	**9.98E-01**	**9.98E-01**	**9.98E-01**	**9.98E-01**	1.014279
	worst	9.98E-01	9.98E-01	1.523844	**9.98E-01**	**9.98E-01**	**9.98E-01**	**9.98E-01**	3.96825
	SD	1.34E-07	2.59E-07	1.05E-01	**0**	**0**	1.11E-16	1.13E-16	9.90E-01
	time	1.73E+00	1.56078	4.300697	1.750402	2.016239	**1.048118**	2.09E+02	6.321202
F15	mean	3.89E-04	3.16E-04	1.35E-03	**3.07E-04**	5.82E-04	6.25E-04	3.07E-04	1.49E-03
	median	3.56E-04	3.16E-04	1.15E-03	**3.07E-04**	**3.07E-04**	6.12E-04	3.07E-04	1.41E-03
	best	3.13E-04	3.15E-04	4.77E-04	**3.07E-04**	3.07E-04	4.27E-04	3.07E-04	9.13E-04
	worst	5.30E-04	3.16E-04	4.41E-03	**3.08E-04**	1.22E-03	8.52E-04	3.07E-04	2.08E-03
	SD	9.92E-05	**9.27E-07**	8.94E-04	1.08E-08	4.42E-04	1.22E-04	2.31E-15	4.53E-04
	time	7.11E-01	8.31E-01	1.692333	9.86E-01	5.61E-01	**2.03E-01**	7.54E+01	5.21724
F16	mean	**-1.031628**	**-1.031628**	**-1.03161**	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**
	median	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**
	best	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**
	worst	**-1.031626**	**-1.031628**	**-1.031448**	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031628**	**-1.031627**
	SD	1.18E-06	3.72E-07	5.19E-05	6.80E-16	**0**	5.36E-16	3.51E-16	1.00E-06
	time	5.63E-01	8.44E-01	1.764582	8.75E-01	5.83E-01	**1.55E-01**	6.31E+01	3.012975
F17	mean	3.98E-01	3.98E-01	3.98E-01	**3.98E-01**	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	median	3.98E-01	3.98E-01	3.98E-01	**3.98E-01**	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	best	3.98E-01	3.98E-01	3.98E-01	**3.98E-01**	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	worst	3.98E-01	3.98E-01	4.02E-01	**3.98E-01**	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	SD	2.32E-04	8.58E-05	7.69E-04	**0**	**0**	2.50E-15	3.56E-16	**0**
	time	5.60E-01	7.48E-01	1.726655	8.57E-01	5.30E-01	**1.27E-01**	6.79E+01	3.160106
F18	mean	**3**	**3**	3.003973	**3**	**3**	3.000123	**3**	3.000001
	median	**3**	**3**	**3**	**3**	**3**	3.000002	**3**	**3**
	best	**3**	**3**	**3**	**3**	**3**	3	**3**	**3**
	worst	**3**	**3**	3.025455	**3**	**3**	3.001199	**3**	3.000002
	SD	1.56E-07	1.57E-07	7.25E-03	7.53E-16	**7.40E-16**	3.18E-04	9.75E-15	9.75E-15
	time	5.97E-01	8.17E-01	1.754178	8.34E-01	5.64E-01	**1.31E-01**	6.30E+01	3.154308
F19	mean	-3.86212	-3.858598	**-3.861157**	-3.862782	-3.862782	-3.862782	-3.862782	-3.862781
	median	-3.86217	-3.858598	**-3.861739**	-3.862782	-3.862782	-3.862782	-3.862782	-3.862782
	best	-3.862782	-3.862532	-3.862683	-3.862782	-3.862782	-3.862782	-3.862782	**-3.86242**
	worst	-3.862218	-3.854664	-3.856577	-3.862782	-3.862782	-3.862782	-3.862782	**-3.861779**
	SD	3.14E-04	5.56E-03	1.78E-03	2.27E-15	**9.36E-16**	1.89E-15	1.01E-15	9.36E-09
	time	8.03E-01	1.334279	2.153489	1.052812	6.52E-01	**3.21E-01**	9.64E+01	3.781642
F20	mean	-3.247418	-3.25873	**-3.159494**	-3.293461	-3.262549	-3.321995	-3.321995	-3.321994
	median	-3.276975	-3.25873	**-3.165188**	-3.321995	-3.262549	-3.321995	-3.321995	-3.321995
	best	-3.294015	**-3.290655**	-3.299126	-3.321995	-3.321995	-3.321995	-3.321995	-3.321995
	worst	**-3.141706**	-3.226805	-2.946335	-3.203102	-3.203102	-3.321995	-3.321995	-3.321992
	SD	7.11E-02	4.51E-02	8.41E-02	5.18E-02	6.27E-02	1.56E-11	3.35E-14	**4.68E-16**
	time	8.47E-01	1.152988	2.141157	1.097036	7.68E-01	**3.39E-01**	9.89E+01	5.368243
F21	mean	-7.633255	-6.896374	-8.497846	-1.02E+01	-4.818715	-1.02E+01	**-1.02E+01**	-7.321239
	median	-1.02E+01	-7.626986	-9.130951	-1.02E+01	-4.818715	-1.02E+01	**-1.02E+01**	-7.339522
	best	-1.02E+01	-1.02E+01	-1.02E+01	-1.02E+01	-6.466687	-1.02E+01	**-1.02E+01**	-8.292549
	worst	-2.788623	-2.630472	-2.68159	-1.02E+01	-3.170744	-1.02E+01	**-1.02E+01**	-6.313362
	SD	3.281182	2.330584	1.842866	**5.44E-15**	3.546332	5.92E-04	2.65E-12	1.089776
	time	9.50E-01	1.181533	2.370532	1.12655	8.74E-01	**3.85E-01**	1.10E+02	4.290946
F22	mean	-7.18753	-7.675265	-8.319781	-1.04E+01	**-6.286392**	-1.04E+01	-1.04E+01	-1.04E+01
	median	-7.494097	-7.675265	-9.042958	-1.04E+01	**-5.108247**	-1.04E+01	-1.04E+01	-1.04E+01
	best	-8.690756	-8.048532	-1.04E+01	-1.04E+01	**-1.04E+01**	**-1.04E+01**	**-1.04E+01**	**-1.04E+01**
	worst	-5.071172	-7.301998	-2.715277	-1.04E+01	-2.751934	-1.04E+01	**-1.04E+01**	-1.19E+01
	SD	1.524091	5.28E-01	2.199572	**2.76E-15**	3.656552	2.71E-03	2.90E-12	1.87E-09
	time	9.92E-01	1.413426	2.64555	1.200651	9.98E-01	**4.54E-01**	1.15E+02	4.634844
F23	mean	-8.700753	-9.48943	-8.324636	-1.05E+01	-7.394579	-1.05E+01	**-1.05E+01**	-1.05E+01
	median	-1.05E+01	-9.48943	-9.373038	-1.05E+01	-7.139911	-1.05E+01	**-1.05E+01**	-1.05E+01
	best	-1.05E+01	-1.03E+01	-1.05E+01	-1.05E+01	-8.920549	-1.05E+01	**-1.05E+01**	-1.05E+01
	worst	-4.871143	-8.727387	-3.602712	-1.05E+01	-6.377942	-1.05E+01	**-1.05E+01**	-1.05E+01
	SD	3.220772	1.077692	2.291127	1.81E-15	1.227986	1.26E-03	3.14E-12	**1.57E-15**
	time	1.346778	1.288167	2.563596	1.323928	1.214182	**5.40E-01**	1.43E+02	5.175969

Function f_{19}, BRO-RT, BRO, GA, DE, PSO, ABC, and FA algorithms exhibit similar mean, median, and best solutions. The BRO-RT algorithm achieves the lowest best solution. The standard deviation is relatively small for all algorithms. GA, PSO, and ABC have the highest execution times.

While the Firefly Algorithm (FA) may have shown

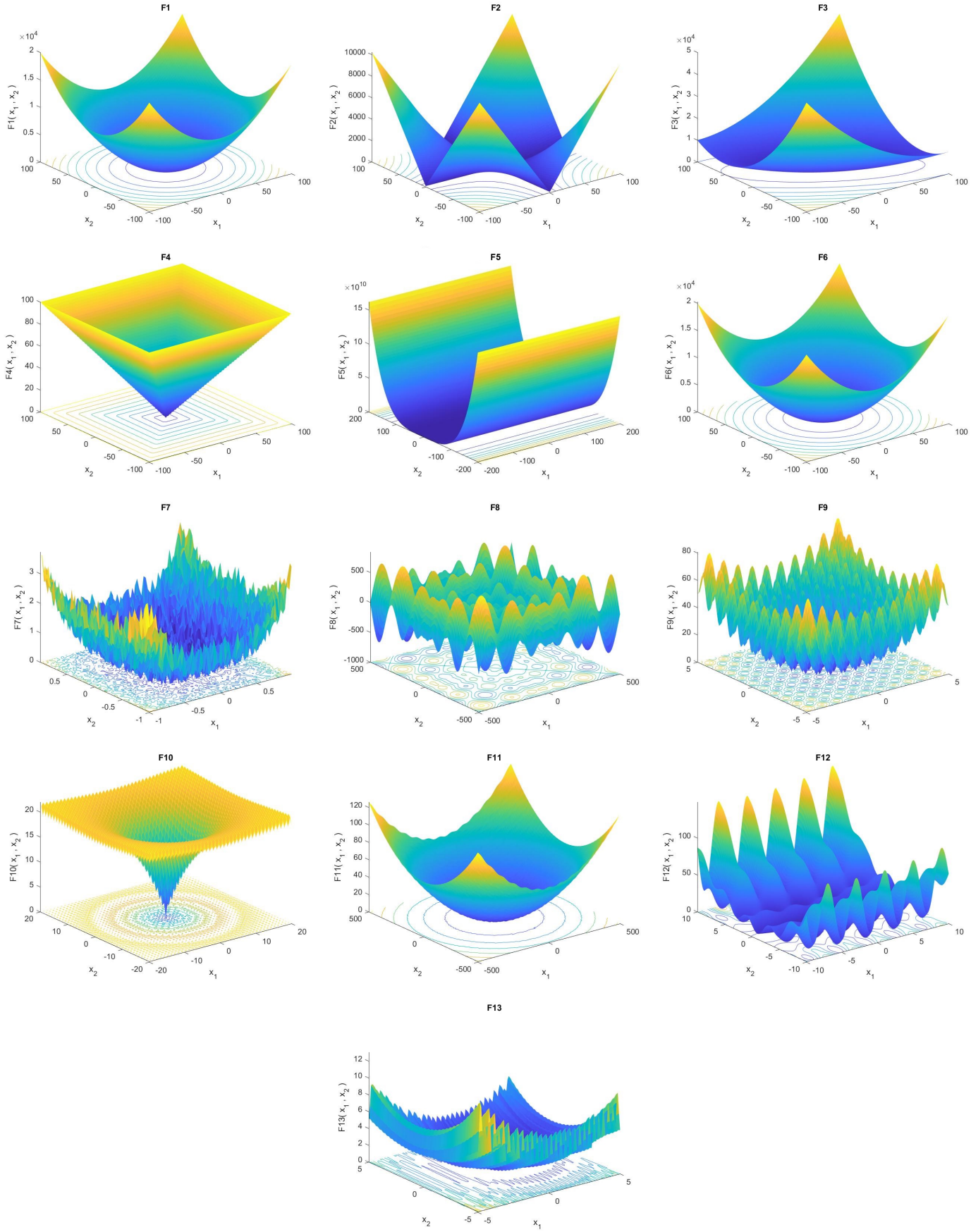


Figure 2. The plots of the problem instance of all benchmark functions.

Table 11. Ranks of algorithms for fixed-dimension Ranks of algorithms for composite functions.

Metrics	BRO-RT	BRO	GA	DE	PSO	ABC	FA	GSA
mean	5	4.8	6.6	2.2	3.7	2.6	**1.1**	4.7
median	3.9	5.3	5.4	1.9	3	2.8	**1.1**	2.7
best	3.6	5.8	3.7	2.5	2	2.2	**1.3**	3.7
worst	5.7	4.2	7.5	2.1	3.6	2.6	**1.1**	4.9
SD	6.1	5.4	7.2	**2.3**	3.8	4	2.5	4.1
time	2.7	4.1	6	4.4	2.8	1	8	7
Average rank	4,5	4.93	6.06	2.56	3.15	2.53	**2.51**	4.51
Overall rank	5	7	8	3	4	2	**1**	6

Table 12. Outcomes for composite benchmark functions

F		BRO-RT	BRO	GA	DE	PSO	ABC	FA	GSA
F24	mean	4.247777	6.961867	5.81E+01	6.97E-06	6.107698	9.81E-03	-	**1.18E-18**
	median	4.315002	6.961867	5.77E+01	2.28E-10	4.733647	7.95E-03	-	**1.26E-18**
	best	1.56858	6.721801	2.79E+01	3.43E-21	0	2.60E-03	-	**6.00E-19**
	worst	7.257057	7.201933	1.09E+02	7.68E-05	1.95E+01	2.58E-02	-	**1.61E-18**
	SD	1.792177	3.40E-01	1.92E+01	1.96E-05	6.277467	5.62E-03	-	**2.89E-19**
	time	5.59E+01	5.85E+01	1.10E+01	6.74E+01	**5.795678**	6.05E+01	-	1.63E+01
F25	mean	3.044667	7.965057	8.30E+01	1.39E-05	1.03E+01	7.84E-02	-	**3.62E-17**
	median	2.785652	7.965057	8.99E+01	4.71E-16	8.933087	5.55E-02	-	**0**
	best	1.630956	7.709063	3.34E+01	7.46E-22	3.55E-01	6.97E-03	-	**0**
	worst	4.808106	8.221052	1.19E+02	1.37E-04	2.07E+01	2.87E-01	-	**3.61E-16**
	SD	8.50E-01	3.62E-01	2.36E+01	3.70E-05	5.789537	6.48E-02	-	**1.14E-16**
	time	7.14E+01	7.82E+01	9.756016	7.92E+01	**5.856811**	8.80E+01	-	1.68E+01
F26	mean	4.99E+01	6.56E+01	5.40E+02	**0**	3.42E+02	3.36E-01	-	**0**
	median	4.62E+01	6.56E+01	5.73E+02	**0**	3.53E+02	2.83E-01	-	**0**
	best	3.23E+01	6.41E+01	2.60E+02	**0**	5.19E+01	5.64E-02	-	**0**
	worst	1.31E+02	6.71E+01	7.36E+02	**0**	4.98E+02	8.76E-01	-	**0**
	SD	2.10E+01	2.137125	1.40E+02	**0**	1.56E+02	2.40E-01	-	**0**
	time	5.88E+01	6.23E+01	1.19E+01	6.26E+01	**5.931329**	6.75E+01	-	**1.21E+01**
F27	mean	3.99E+01	3.31E+01	2.86E+02	4.31E-13	1.09E+02	9.05E-01	-	**3.96E-15**
	median	2.90E+01	3.31E+01	2.93E+02	1.12E-13	1.39E+02	8.54E-01	-	**4.40E-15**
	best	2.51E+01	2.93E+01	1.16E+02	1.98E-14	2.56E+01	8.55E-02	-	**0**
	worst	2.61E+02	3.69E+01	3.86E+02	4.29E-12	1.82E+02	2.267773	-	**8.81E-15**
	SD	4.65E+01	5.393872	6.74E+01	8.75E-13	6.15E+01	5.72E-01	-	**2.90E-15**
	time	7.67E+01	7.70E+01	1.31E+01	8.06E+01	**6.262372**	6.26E+01	-	1.14E+01

Table 13. Ranks of algorithms for composite functions.

Metrics	BRO-RT	BRO	GA	DE	PSO	ABC	FA	GSA
mean	4.25	5	7	1.75	5.75	3	-	1
median	4	5.25	7	1.75	5.75	3	-	1
best	4.5	6	7	1.75	3.75	3.25	-	1.5
worst	5	4.25	7	1.75	5.75	3	-	1
SD	5	4	6.75	1.75	6.25	3	-	1
time	4.25	5.25	2.25	6.5	1	6	-	2.75
Average rank	4.5	4.958333	6.166667	2.541667	4.708333	3.541667	-	1.375
Overall rank	4	6	7	2	5	3	-	1

better performance in terms of optimization results for certain objective functions, it is important to consider the execution time as a crucial factor in practical applications. Despite its success in achieving favorable solutions, if the execution time of FA is deemed unacceptable or impractical for an optimization task, it can limit its usefulness in real-world scenarios. In such cases, alternative algorithms that strike a balance between solution quality and execution time, such as BRO-RT, GA, DE, PSO, or ABC, might be more suitable choices. These algorithms have demonstrated competitive performance and relatively faster execution times, making them more practical options for optimization tasks with time constraints.

As it is clear from Table 11, among the algorithms

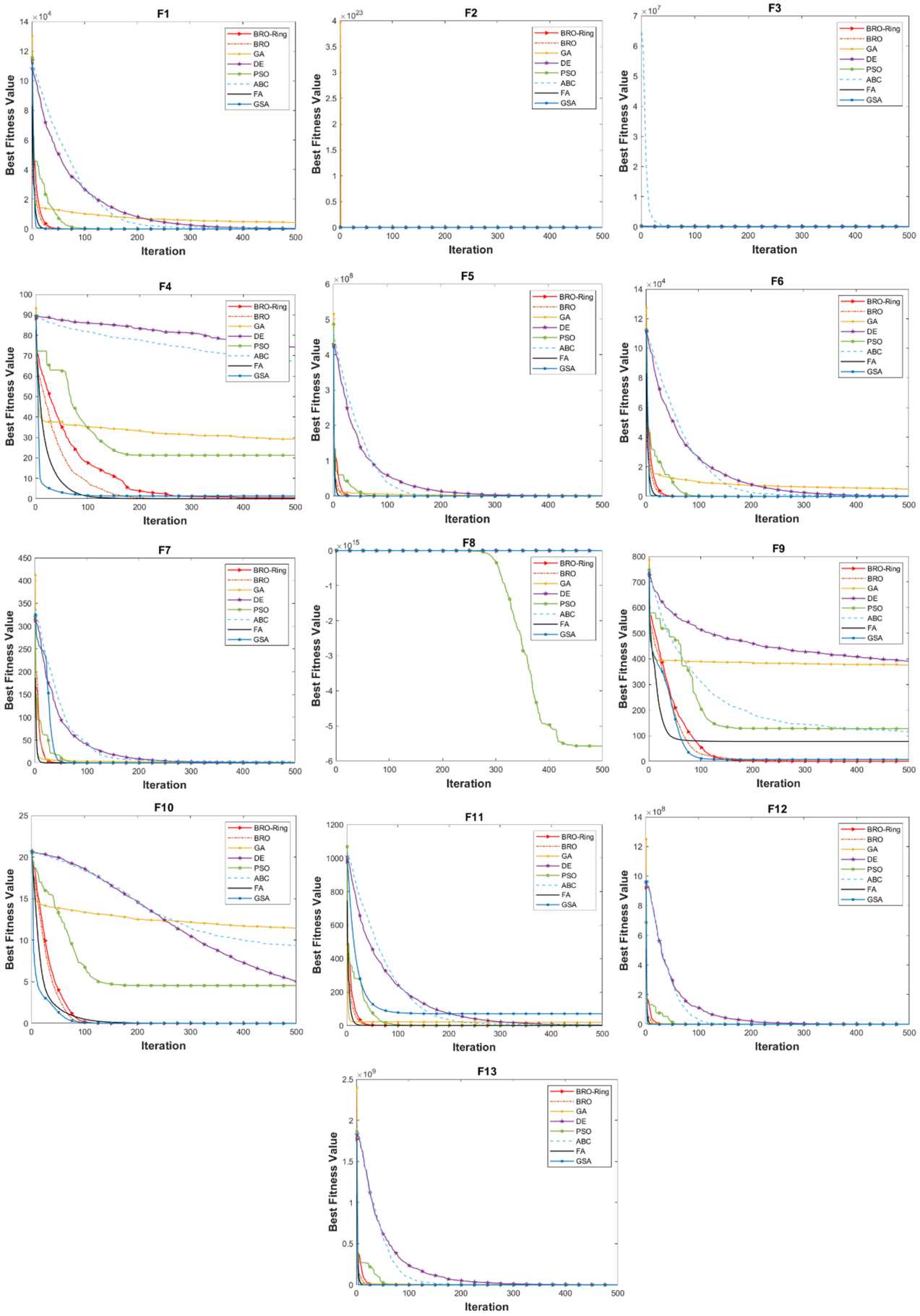


Figure 3. Average convergence curves over 25 runs.

evaluated, FA emerges as the top-performing algorithm overall, achieving the lowest average rank and securing the first position. However, if the execution time of FA is deemed unacceptable or impractical for an optimization task, it can limit its usefulness in real-world scenarios. DE follows closely behind in the third position, while ABC claims the second position. BRO-RT and BRO hold the fifth and seventh positions, respectively, indicating relatively good performance. GA, PSO, and GSA occupy the sixth, fourth, and sixth positions, respectively, in terms of overall rank. The Firefly Algorithm (FA) is excluded from the reported results (Table 12) due to its unacceptable execution time, as the evaluation of each function using FA takes more than one day. The impracticality of FA's execution time makes it challenging to obtain timely and efficient results within the scope of the benchmarking process. As a result, the focus of the reported results is on the other algorithms that offer more reasonable execution times while still providing valuable insights for the composite benchmark functions. Table 13 lists the rank of each algorithm. The table shows that the BRO-RT is not only faster than BRO but also provides competitive results. Across all benchmark categories, the reported execution times (Tables 6– 13) consistently indicate that BRO-RT is faster than the original BRO, confirming that the proposed ring topology effectively reduces computational cost

In Table 14, we present the results of a comprehensive pairwise statistical analysis conducted to evaluate the comparative performance of the BRO algorithm against a suite of established optimization algorithms: GA, DE, PSO, ABC, FA, and GSA. The performance metrics assessed in this analysis are derived from a set of standardized benchmark functions, denoted as f_1 through f_{13}, which are commonly used within the field to test the efficacy of optimization algorithms. Each function provides a unique landscape to challenge the algorithms' ability to locate global optima within a multidimensional search space. The Wilcoxon Signed-Rank Test, with a significance level of $\alpha = 0.05$, was utilized to determine if there are statistically significant differences in the algorithms' performance with respect to the quality of solutions found, as opposed to processing time or convergence speed. The resulting p-values and corresponding effect sizes (denoted as h^2) provide a robust statistical basis to infer the relative performance advantages of the BRO algorithm. Where p-values fall below the threshold of α, we interpret this as indicative of a statistically significant difference in solution quality, favoring the BRO algorithm over its counterparts for the corresponding benchmark function.

A score of 1 indicates that BRO-RT was more successful than its competitors. On the other hand, a score of 0 indicates that both algorithms performed equally. This table shows that in most cases, BRO-RT exhibits competitive behavior compared to other algorithms, although several state-of-the-art methods achieve superior convergence on specific benchmark functions or during certain stages of the optimization process.

It is important to validate the results in an application case. The constrained engineering design optimization problem (compression spring design) was used in the evaluations to demonstrate the proposed method's feasibility and effectiveness in addressing real-world applications (see Figure 4). The main objective is to minimize the weight subject to different constraints on the minimum deflection, reduce stress and surge frequency, and place limits on the geometric dimensions. The three design variables to be optimized are wire diameter x_1, mean coil diameter x_2, and the number of active coils x_3. The objective function is expressed mathematically as:

$$f(\mathbf{x}) = (x_3 + 2)x_2x_1^2 \tag{9}$$

Subject to:

$$\begin{aligned} g_1(\mathbf{x}) &= 1 - \frac{x_2^2 x_3}{71785x_1^4} \leq 0 \\ g_2(\mathbf{x}) &= \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \\ g_3(\mathbf{x}) &= 1 - \frac{140.45x_1}{x_2^2 x_3} \leq 0 \\ g_4(\mathbf{x}) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0 \end{aligned} \tag{10}$$

where variables are bounded by: $0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15$.

It is important to mention that constraint violations were handled using a penalty function added to the objective value. All algorithms were executed for 25 independent runs with different random initializations; no common random seed was enforced across algorithms, and results are reported as averaged statistics.

Table 15 lists the outcomes of algorithms for the spring design problem. Clearly, the proposed method achieved competitive results in four out

Table 14. Pairwise statistical comparison between BRO-RT and all other algorithms using the Wilcoxon Signed-Rank Test ($\alpha = 0.05$).

	BRO		GA		DE		PSO		ABC		FA		GSA	
	p-value	*h*	p-value	*h*	p-value	*h*	p-value	*h*	p-value	*h*	p-value	*h*	p-value	*h*
F1	1,23E+09	1	0.000239	1	0.000266	1	0.00172	1	0.000295	1	1,23E+09	1	1,23E+09	1
F2	1,23E+09	1	0.000174	1	0.001078	1	0.004162	1	0.003507	1	1,23E+09	1	1,23E+09	1
F3	0.000664	1	0.000295	1	0.000194	1	0.000445	1	9,04E+09	1	1,23E+09	1	0.000194	1
F4	1,23E+09	1	0.000402	1	0.000664	1	0.000445	1	0.000215	1	1,23E+09	1	0.0001743	1
F5	0.000295	1	0.000295	1	0.00014	1	0.000194	1	0.000215	1	0.000295	1	0.0005451	1
F6	0.287862	0	0.000493	1	9,04E+09	1	0.001569	1	0.000215	1	1,23E+09	1	1,23E+09	1
F7	1,23E+09	1	0.000295	1	1,23E+09	1	1,23E+09	1	0.716423	0	1,23E+09	1	1,23E+09	1
F8	0.000239	1	0.00014	1	0.000266	1	0.000125	1	0.00014	1	0.000239	1	0.000266	1
F9	0.003507	1	0.000194	1	0.000402	1	0.000295	1	0.000445	1	0.000445	1	0.008041	1
F10	1,23E+09	1	0.001885	1	0.02643	1	0.121827	0	0.002469	1	1,23E+09	1	1,23E+09	1
F11	1,23E+09	1	0.001078	1	0.544909	0	3,62E+09	1	0.024657	1	1,23E+09	1	0.000363	1
F12	1,23E+09	1	0.000295	1	0.000328	1	0.000602	1	0.353257	0	1,23E+09	1	0.157769	0
F13	0.001885	1	0.000295	1	0.000295	1	0.000194	1	0.000239	1	1,23E+09	1	0.00098	1

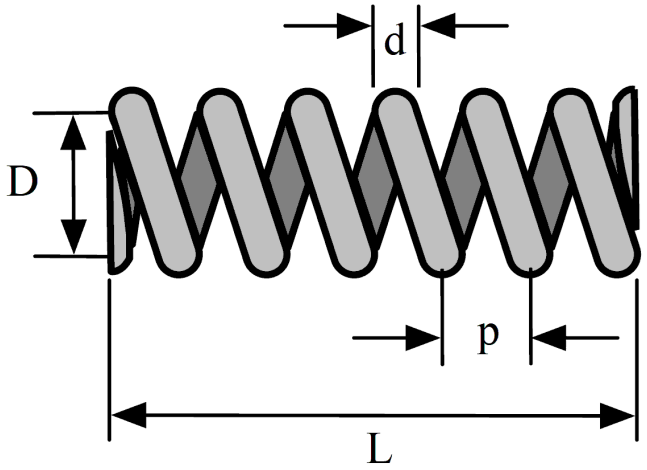


Figure 4. Compression spring design problem.

of six performance criteria and ranked first overall. However, it ranked seventh with respect to time, and it took almost twice as long as BRO-RT. Upon closer analysis, DE ranked first in terms of mean, median, worst, and SD, while BRO-RT ranked first in terms of best. Furthermore, although DE ranked first for most performance criteria, BRO and BRO-RT achieved competitive outcomes. Finally, while BRO and BRO-RT obtained competitive results, BRO-RT had lower computational complexity and ran 1.5 times faster overall than the original BRO algorithm.

To complement the research, the proposed BRO-RT is compared with the original BRO, the six algorithms previously worked on the 13 benchmark functions, and the Liver Cancer Algorithm (LCA) [41] and Dung Beetle Optimizer (DBO) [42] algorithms, using CEC2020 [45] were added to the comparison. Table 16 shows the results using the statistics of mean, median, worst, best, standard deviation, and average execution time. Winning results are placed in bold. The CEC2020 consists of 10 benchmark functions: one unimodal, three basics, three hybrids and three compounds. Table 17 shows the rankings of the algorithms in the CEC2020 for the six statistics. It is observed that the DE is the winning algorithm in the mean, the best, the worst, and the standard deviation, but in the median and the execution time, it does not dominate. The BRO-RT wins in the mean, and the median, in the best it gets second place, the worst and the sd gets the third, but in the execution time it gets the eighth place. Comparing the BRO-RT with the BRO, a substantial improvement is seen in all six statistics.

In f_1, the DE is the winner by far, leaving the BRO-RT in second place, although the BRO-RT achieves an improvement in results with respect to the BRO. The LCA execution times were much better in all CEC2020 functions, and in this particular function, there is no exception. For f_2, the DBO wins in three statistics, but the DE recovers for the best results for the worst obtained and the standard deviation. For f_3 and f_4, DE obtains better results compared to the other algorithms. The BRO-RT is the winner at f_5 for the mean and median, but the BRO wins at the best, for the standard deviation, and the worst, the FA stands out. In f_6, the BRO-RT wins in three of the six indicators (mean, worst, t and sd), the GA in the median, and the FA in the best. At f_7, the FA algorithm obtains the best results, and the BRO-RT follows closely. Finally, in functions f_8, f_9, and f_{10}, the DE resumes leadership, and the BRO-RT maintains closeness regarding the best results. The performance of the proposed algorithm overall ranks

Table 15. Results and ranks of algorithms for spring design problem.

Metrics	BRO-RT	BRO	GA	DE	PSO	ABC	FA	GSA
mean	1.27E-02	1.27E-02	1.62E-02	**1.27E-02**	1.30E-02	1.29E-02	1.28E-02	1.52E-02
median	1.28E-02	1.27E-02	1.63E-02	**1.27E-02**	1.27E-02	1.29E-02	1.28E-02	1.51E-02
best	**1.27E-02**	1.27E-02	1.33E-02	1.27E-02	1.27E-02	1.28E-02	1.27E-02	1.35E-02
worst	1.28E-02	1.27E-02	1.82E-02	**1.27E-02**	1.39E-02	1.32E-02	1.29E-02	1.67E-02
SD	4.50E-05	1.66E-05	1.41E-03	**3.34E-07**	4.93E-04	1.73E-04	3.37E-05	1.14E-03
time	6.143011	8.808807	6.446486	13.01927	**5.508894**	6.542683	1042.801	9.618786
Average rank	2.833,333	2.666667	7	2.166667	4.166667	5.166667	4.833333	7
Overall rank	3	2	7	**1**	4	6	5	7

Table 16. Outcomes for CEC2020 benchmark functions when dimension is 20.

F		BRO-RT	BRO	GA	DE	PSO	ABC	FA	GSA	LCA	DBO
F1	mean	1.49E+09	2.09E+09	1.30E+11	**1.95E+06**	4.93E+10	2.51E+10	1.61E+09	1.22E+10	3.75E+10	2.46E+10
	median	1.33E+09	2.00E+09	1.27E+11	**1.95E+06**	5.12E+10	2.51E+10	1.64E+09	1.22E+10	3.86E+10	2.49E+10
	best	7.36E+08	9.03E+08	6.18E+10	**4.69E+05**	2.98E+10	2.12E+10	9.17E+08	1.03E+10	1.64E+10	1.64E+10
	worst	3.21E+09	4.11E+09	2.81E+11	**4.29E+06**	6.16E+10	3.15E+10	2.23E+09	1.45E+10	4.76E+10	3.49E+10
	SD	5.73E+08	7.42E+08	4.86E+10	**9.01E+05**	8.05E+09	2.34E+09	3.02E+08	1.23E+09	6.62E+09	5.09E+09
	time	2.92E+00	3.41E+00	1.44E+00	3.18E+00	2.13E+00	2.48E+00	2.74E+00	1.25E+01	**5.45E-01**	2.09E+00
F2	mean	4.57E+03	4.76E+03	8.98E+03	4.19E+03	6.53E+03	5.42E+03	4.76E+03	4.90E+03	6.61E+03	**4.03E+03**
	median	4.53E+03	4.82E+03	9.08E+03	4.20E+03	6.59E+03	5.43E+03	4.84E+03	4.90E+03	6.59E+03	**4.10E+03**
	best	3.54E+03	3.76E+03	7.02E+03	3.72E+03	5.82E+03	4.63E+03	3.86E+03	4.49E+03	6.17E+03	**2.82E+03**
	worst	5.26E+03	5.65E+03	1.00E+04	**4.61E+03**	7.10E+03	5.84E+03	5.16E+03	5.40E+03	7.13E+03	5.14E+03
	SD	4.71E+02	5.08E+02	8.21E+02	**2.05E+02**	3.40E+02	2.44E+02	3.11E+02	2.44E+02	2.46E+02	6.56E+02
	time	3.12E+00	3.58E+00	1.70E+00	3.45E+00	2.43E+00	2.62E+00	2.69E+00	1.29E+01	**5.74E-01**	1.67E+00
F3	mean	8.66E+02	8.70E+02	3.29E+03	**8.14E+02**	1.99E+03	1.50E+03	9.09E+02	8.45E+02	1.10E+03	9.67E+02
	median	8.62E+02	8.69E+02	3.32E+03	**8.17E+02**	2.00E+03	1.50E+03	9.08E+02	8.45E+02	1.11E+03	9.73E+02
	best	8.21E+02	8.27E+02	2.24E+03	**7.98E+02**	1.77E+03	1.34E+03	8.69E+02	8.16E+02	1.05E+03	9.01E+02
	worst	9.07E+02	9.45E+02	4.41E+03	**8.26E+02**	2.23E+03	1.64E+03	9.35E+02	8.83E+02	1.15E+03	1.07E+03
	SD	2.43E+01	2.40E+01	5.91E+02	**6.62E+00**	1.15E+02	7.57E+01	1.44E+01	1.79E+01	2.91E+01	3.75E+01
	time	2.94E+00	3.46E+00	1.59E+00	3.43E+00	2.40E+00	2.40E+00	2.63E+00	1.26E+01	**5.52E-01**	1.59E+00
F4	mean	1.92E+03	1.94E+03	3.80E+09	**1.91E+03**	4.13E+06	3.59E+05	1.99E+03	1.82E+05	2.07E+06	2.05E+05
	median	1.92E+03	1.92E+03	8.00E+08	**1.91E+03**	3.99E+06	2.80E+05	1.99E+03	1.80E+05	1.61E+06	1.92E+05
	best	1.91E+03	1.91E+03	8.12E+06	**1.91E+03**	7.44E+05	7.17E+04	1.92E+03	1.25E+05	2.99E+05	4.56E+03
	worst	1.95E+03	2.12E+03	2.25E+10	**1.91E+03**	1.02E+07	8.95E+05	2.11E+03	2.53E+05	6.44E+06	4.91E+05
	SD	8.37E+00	3.90E+01	6.42E+09	**6.25E-01**	2.27E+06	2.18E+05	4.30E+01	3.76E+04	1.66E+06	1.59E+05
	time	3.01E+00	3.52E+00	1.58E+00	3.39E+00	2.37E+00	2.30E+00	2.58E+00	1.29E+01	**5.66E-01**	1.81E+00
F5	mean	**5.40E+05**	5.92E+05	2.80E+09	1.23E+06	4.75E+07	6.75E+06	5.45E+05	2.22E+06	2.36E+07	1.01E+06
	median	**4.55E+05**	5.11E+05	2.79E+09	1.12E+06	4.01E+07	6.81E+06	5.87E+05	2.18E+06	2.60E+07	9.17E+05
	best	1.59E+05	**8.65E+04**	9.99E+07	4.06E+05	8.03E+06	1.82E+06	1.60E+05	1.25E+06	6.36E+06	1.17E+05
	worst	1.35E+06	2.19E+06	4.70E+09	2.22E+06	1.64E+08	1.12E+07	**1.00E+06**	3.27E+06	4.50E+07	2.99E+06
	SD	3.05E+05	5.03E+05	1.71E+09	4.93E+05	3.10E+07	2.32E+06	**2.09E+05**	5.39E+05	1.25E+07	7.32E+05
	time	3.02E+00	3.57E+00	1.63E+00	3.56E+00	2.38E+00	2.41E+00	2.69E+00	1.32E+01	**7.06E-01**	1.82E+00
F6	mean	**1.73E+03**	1.87E+03	1.77E+03	1.91E+03	1.96E+03	1.93E+03	1.93E+03	1.98E+03	1.95E+03	1.81E+03
	median	1.74E+03	1.82E+03	**1.61E+03**	2.05E+03	2.05E+03	2.02E+03	1.93E+03	2.04E+03	1.97E+03	1.75E+03
	best	1.61E+03	1.62E+03	1.61E+03	1.60E+03	1.61E+03	1.62E+03	**1.60E+03**	1.62E+03	1.60E+03	1.62E+03
	worst	**2.05E+03**	2.10E+03	2.12E+03	2.20E+03	2.67E+03	2.44E+03	2.21E+03	2.41E+03	2.32E+03	2.30E+03
	SD	**1.29E+02**	1.66E+02	2.32E+02	2.31E+02	3.30E+02	2.50E+02	1.79E+02	1.92E+02	2.48E+02	1.74E+02
	time	3.19E+00	3.51E+00	1.46E+00	3.30E+00	2.25E+00	2.81E+00	2.10E+01	1.32E+01	**4.80E-01**	1.74E+00
F7	mean	1.77E+05	1.51E+05	1.12E+09	1.78E+05	2.38E+07	1.60E+06	**1.09E+05**	2.51E+06	3.74E+07	4.32E+05
	median	1.39E+05	1.18E+05	1.36E+09	1.72E+05	2.10E+07	1.70E+06	**1.03E+05**	2.47E+06	2.73E+07	2.85E+05
	best	4.81E+04	3.12E+04	2.34E+08	3.65E+04	3.20E+06	4.76E+05	**2.86E+04**	8.49E+05	1.69E+06	4.96E+04
	worst	1.05E+06	4.22E+05	1.36E+09	3.53E+05	6.34E+07	3.08E+06	**2.32E+05**	5.80E+06	1.28E+08	1.94E+06
	SD	1.80E+05	1.02E+05	3.63E+08	7.63E+04	1.56E+07	7.32E+05	**4.84E+04**	9.36E+05	3.21E+07	4.55E+05
	time	3.83E+00	3.47E+00	1.58E+00	3.37E+00	2.35E+00	2.41E+00	2.74E+00	1.30E+01	**6.41E-01**	1.83E+00
F8	mean	2.69E+03	2.74E+03	1.06E+04	**2.30E+03**	7.61E+03	5.47E+03	2.57E+03	5.37E+03	7.09E+03	4.28E+03
	median	2.45E+03	2.48E+03	1.09E+04	**2.30E+03**	7.66E+03	5.43E+03	2.58E+03	5.43E+03	7.38E+03	4.33E+03
	best	2.39E+03	2.40E+03	9.35E+03	**2.30E+03**	5.68E+03	4.42E+03	2.46E+03	4.09E+03	4.89E+03	2.94E+03
	worst	6.91E+03	6.34E+03	1.10E+04	**2.31E+03**	8.81E+03	6.37E+03	2.70E+03	6.03E+03	8.56E+03	5.24E+03
	SD	9.50E+02	9.39E+02	5.10E+02	**2.87E+00**	7.76E+02	4.33E+02	4.70E+01	3.88E+02	8.30E+02	5.48E+02
	time	4.24E+00	3.96E+00	1.93E+00	3.86E+00	2.68E+00	2.82E+00	3.26E+00	1.37E+01	**9.23E-01**	2.00E+00
F9	mean	2.92E+03	2.93E+03	4.25E+03	**2.90E+03**	3.55E+03	3.15E+03	2.93E+03	3.54E+03	3.80E+03	3.10E+03
	median	2.92E+03	2.93E+03	4.28E+03	**2.90E+03**	3.55E+03	3.16E+03	2.94E+03	3.57E+03	3.83E+03	3.11E+03
	best	2.89E+03	2.90E+03	3.95E+03	**2.88E+03**	3.27E+03	3.07E+03	2.90E+03	3.28E+03	3.39E+03	2.97E+03
	worst	2.96E+03	2.96E+03	4.28E+03	**2.91E+03**	3.79E+03	3.21E+03	2.95E+03	3.68E+03	4.06E+03	3.21E+03
	SD	1.73E+01	1.58E+01	8.75E+01	**7.68E+00**	1.39E+02	3.35E+01	1.22E+01	9.71E+01	1.89E+02	6.16E+01
	time	4.40E+00	4.18E+00	2.07E+00	3.96E+00	2.83E+00	3.07E+00	3.58E+00	1.35E+01	**1.25E+00**	2.13E+00
F10	mean	3.01E+03	3.02E+03	6.10E+04	**2.91E+03**	9.13E+03	5.25E+03	3.05E+03	3.80E+03	7.65E+03	4.00E+03
	median	3.01E+03	3.01E+03	4.24E+04	**2.91E+03**	9.13E+03	5.24E+03	3.04E+03	3.79E+03	7.58E+03	3.89E+03
	best	2.94E+03	2.94E+03	8.79E+03	**2.91E+03**	5.68E+03	4.26E+03	2.99E+03	3.56E+03	4.88E+03	3.42E+03
	worst	3.11E+03	3.11E+03	2.01E+05	**2.91E+03**	1.40E+04	6.27E+03	3.11E+03	3.99E+03	1.09E+04	4.96E+03
	SD	4.72E+01	4.61E+01	4.86E+04	**4.29E-02**	1.96E+03	5.66E+02	2.72E+01	9.67E+01	1.44E+03	4.36E+02
	time	3.94E+00	3.90E+00	1.88E+00	3.79E+00	2.64E+00	2.71E+00	3.20E+00	1.34E+01	**8.93E-01**	2.00E+00

Table 17. Ranks of algorithms for CEC2020 benchmark functions.

	BRO-RT	BRO	GA	DE	PSO	ABC	FA	GSA	LCA	DBO
mean	1	3	10	1	9	7	4	6	8	5
median	1	3	10	2	9	7	4	6	8	5
best	2	3	10	1	9	7	3	6	8	5
worst	3	4	10	1	9	7	2	6	8	5
SD	3	4	10	1	9	7	2	5	8	6
time	8	9	2	7	4	5	6	10	1	3
Average rank	3	4.33333	8.666	2.1666	8.1666	6.666	3.5	6.5	6.833	4.833
Overall rank	**2**	3	8	1	6	4	1	2	2	1

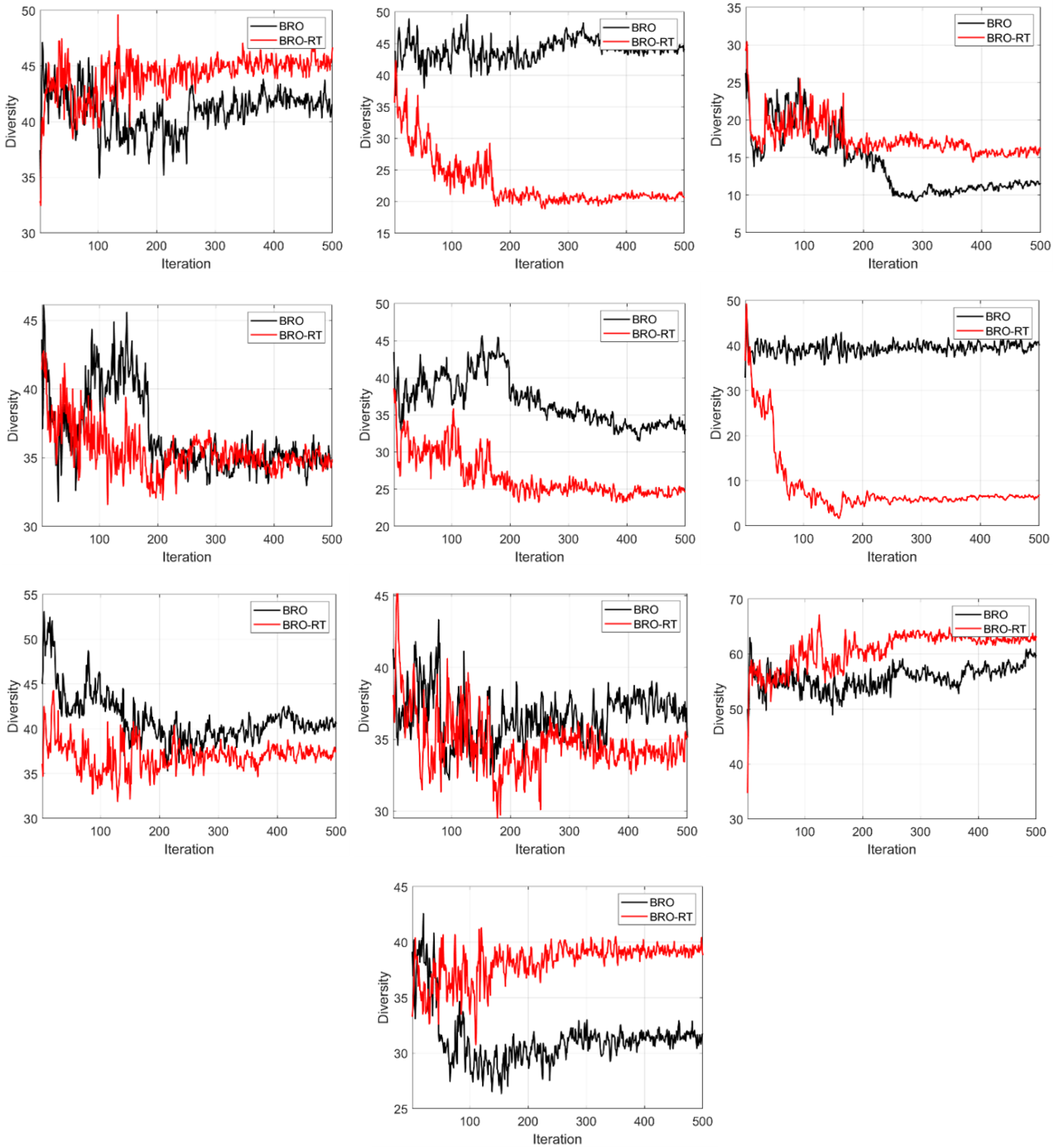


Figure 5. Comparative Analysis of Dimension-Wise Diversity Across Iterations for BRO and BRO-RT.

Table 18. Friedman statistical analysis for CEC2020 results.

Algorithm	Score	Rank
BRO-RT	2.2	1
DE	2.2	1
BRO	3.5	2
FA	3.6	3
DBO	4.7	4
GSA	5.9	5
ABC	6.8	6
LCA	8.2	7
PSO	8.7	8
GA	9.2	9

second, beating the original BRO. To validate the results, the Friedman statistical test (Table 18) is used on the means of the results in Table 16. This test allows obtaining a general ranking of the algorithms. A p-value equal to $2.57E+11$ is observed, so there is a significant difference in the results of the compared algorithms and the BRO-RT with the DE obtains first place. Second place goes to BRO, and third place to FA.

Figure 5 illustrates the dimension-wise diversity trends of two optimization algorithms, BRO and BRO-RT, across various benchmark functions (F_1 to F_{10}) over a series of iterations. Diversity is plotted on the vertical axis, while the number of iterations is represented on the horizontal axis. The diversity measurement reflects the variance of the population in each dimension, highlighting the algorithm's ability to explore the search space without converging prematurely. It can be observed that BRO-RT consistently maintains higher diversity compared to BRO across most benchmark functions, which suggests a greater exploratory capability. Notably, in functions F_2, F_3, F_5, and F_6, BRO-RT exhibits a pronounced advantage in maintaining diversity throughout the iterative process. This robust preservation of diversity is indicative of BRO-RT's potential to escape local optima and explore a broader range of solutions, which could translate to improved optimization performance. The fluctuations in diversity levels reflect the dynamic adjustments made by the algorithms in response to the topology of the search space. These findings highlight the importance of maintaining diversity in optimization algorithms, particularly in complex, multimodal functions where the risk of premature convergence is substantial.

5 Conclusion and future work

Our research method used a modified version of a recently proposed BRO algorithm, BRO-RT. To improve the computational complexity of BRO, ring neighborhood topology was used. So, instead of calculating the Euclidean distance between all individuals to find a neighbor, the ring topology chooses the proper neighbor. Using ring topology results in a decrease in the computational complexity by a factor of n. Although the computational accuracy of the algorithm is slightly reduced, the results show that the proposed algorithm is very effective in reducing computational time. Overall, the proposed method respectively ranks second, first, and third in unimodal, multimodal, and real-world optimization problems. Furthermore, the simulation and statistical results for the constraint and unconstrained optimization problems reveal that BRO-RT can compete with or exceed the existing optimization algorithms used for comparison in this research. Therefore, for future work, it has enormous potential for use as a tool to hone various optimization tasks in many real-world fields. We also plan to use the same strategy for solving binary optimization problems.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Lazar, A. (2002). Heuristic knowledge discovery for archaeological data using genetic algorithms and rough sets. In *Heuristic and optimization for knowledge discovery* (pp. 263-278). IGI Global. [CrossRef]

[2] Bianchi, L., Dorigo, M., Gambardella, L. M., & Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing, 8*(2), 239-287. [CrossRef]

[3] Rahkar Farshi, T. (2021). Battle royale optimization algorithm. *Neural Computing and Applications, 33*(4), 1139–1157. [CrossRef]

[4] Yang, X. S. (2020). *Nature-inspired optimization algorithms*. Academic Press.

[5] Schwefel, H. P. (1984). Evolution Strategies: A Family of Non-Linear Optimization Techniques Based on Imitating some Principles of Organic Evolution. *Annals of Operations Research, 1*. [CrossRef]

[6] Glover, F., & McMillan, C. (1986). The general employee scheduling problem. An integration of MS and AI. *Computers & operations research, 13*(5), 563–573. [CrossRef]

[7] Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science, 220*(4598), 671-680. [CrossRef]

[8] Simon, D. (2008). Biogeography-based optimization. *IEEE transactions on evolutionary computation, 12*(6), 702-713. [CrossRef]

[9] Askarzadeh, A. (2014). Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. *Communications in Nonlinear Science and Numerical Simulation, 19*(4), 1213–1228. [CrossRef]

[10] Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the sixth international symposium on micro machine and human science* (pp. 39–43). IEEE. [CrossRef]

[11] Talbi, E. G., Roux, O., Fonlupt, C., & Robillard, D. (1999, April). Parallel ant colonies for combinatorial optimization problems. In *International Parallel Processing Symposium* (pp. 239-247). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[12] Bitam, S., Batouche, M., & Talbi, E. G. (2010, April). A survey on bee colony algorithms. In *2010 IEEE international symposium on parallel & distributed processing, workshops and phd forum (ipdpsw)* (pp. 1-8). IEEE. [CrossRef]

[13] Ma, M., Luo, Q., Zhou, Y., Chen, X., & Li, L. (2015). An improved animal migration optimization algorithm for clustering analysis. *Discrete Dynamics in Nature and Society, 2015*(1), 194792. [CrossRef]

[14] Mousavirad, S. J., & Ebrahimpour-Komleh, H. (2017). Human mental search: a new population-based metaheuristic optimization algorithm. *Applied Intelligence, 47*(3), 850-887. [CrossRef]

[15] Fausto, F., Cuevas, E., Valdivia, A., & González, A. (2017). A global optimization algorithm inspired in the behavior of selfish herds. *Biosystems, 160*, 39-55. [CrossRef]

[16] Tu, S., Rehman, O. U., Rehman, S. U., Ullah, S., Waqas, M., & Zhu, R. (2020). A novel quantum inspired particle swarm optimization algorithm for electromagnetic applications. *IEEE Access, 8*, 21909-21916. [CrossRef]

[17] Tabrizian, Z., Afshari, E., Amiri, G. G., Ali Beigy, M. H., & Nejad, S. M. P. (2013). A New Damage Detection Method: Big Bang-Big Crunch (BB-BC) Algorithm. *Shock and Vibration, 20*(4), 633-648. [CrossRef]

[18] Zha, J., Zeng, G. Q., & Lu, Y. Z. (2010, December). Hysteretic optimization for protein folding on the lattice. In *2010 International Conference on Computational Intelligence and Software Engineering* (pp. 1-4). IEEE. [CrossRef]

[19] Doğan, B., & Ölmez, T. (2015). A new metaheuristic for numerical function optimization: Vortex Search algorithm. *Information sciences, 293*, 125-145. [CrossRef]

[20] Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information sciences, 237*, 82-117. [CrossRef]

[21] Wolpert, D. H., & Macready, W. G. (2002). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation, 1*(1), 67-82. [CrossRef]

[22] Camacho Villalón, C. L., Stützle, T., & Dorigo, M. (2020, October). Grey wolf, firefly and bat algorithms: Three widespread algorithms that do not contain any novelty. In *International conference on swarm intelligence* (pp. 121-133). Cham: Springer International Publishing. [CrossRef]

[23] Aranha, C., Camacho Villalón, C. L., Campelo, F., Dorigo, M., Ruiz, R., Sevaux, M., ... & Stützle, T. (2022). Metaphor-based metaheuristics, a call for action: the elephant in the room. *Swarm Intelligence, 16*(1), 1-6. [CrossRef]

[24] Deng, W., Shang, S., Cai, X., Zhao, H., Song, Y., & Xu, J. (2021). An improved differential evolution algorithm and its application in optimization problem. *Soft Computing-A Fusion of Foundations, Methodologies & Applications, 25*(7). [CrossRef]

[25] Seyyedabbasi, A., & Kiani, F. (2021). I-GWO and Ex-GWO: improved algorithms of the Grey Wolf Optimizer to solve global optimization problems. *Engineering with Computers, 37*(1), 509-532. [CrossRef]

[26] Ghosh, A., Das, S., & Das, A. K. (2020). A simple two-phase differential evolution for improved global numerical optimization. *Soft Computing-A Fusion of Foundations, Methodologies & Applications, 24*(8). [CrossRef]

[27] Li, Y., Xiang, R., Jiao, L., & Liu, R. (2012). An improved cooperative quantum-behaved particle swarm optimization. *Soft Computing, 16*(6), 1061-1069. [CrossRef]

[28] Akan, S., & Akan, T. (2022). Battle royale optimizer with a new movement strategy. In *Handbook of nature-inspired optimization algorithms: the state of the art: volume I: solving single objective bound-constrained real-parameter numerical optimization problems* (pp.

265-279). Cham: Springer International Publishing. [CrossRef]

[29] Akan, T., Agahian, S., & Dehkharghani, R. (2022). Binbro: Binary battle royale optimizer algorithm. *Expert systems with applications, 195*, 116599. [CrossRef]

[30] Agahian, S., & Akan, T. (2022). Battle royale optimizer for training multi-layer perceptron. *Evolving Systems, 13*(4), 563-575. [CrossRef]

[31] Wu, H., Zhang, X., Song, L., Su, C., & Gu, L. (2022). A hybrid improved bro algorithm and its application in inverse kinematics of 7r 6dof robot. *Advances in Mechanical Engineering, 14*(3), 16878132221085125. [CrossRef]

[32] Halim, Z., Sargana, H. M., & Waqas, M. (2021). Clustering of graphs using pseudo-guided random walk. *Journal of Computational Science, 51*, 101281. [CrossRef]

[33] Paz, A., & Moran, S. (1981). Non deterministic polynomial optimization problems and their approximations. *Theoretical Computer Science, 15*(3), 251-277. [CrossRef]

[34] Box, M. J. (1965). A new method of constrained optimization and a comparison with other methods. *The Computer Journal, 8*(1), 42-52. [CrossRef]

[35] Krus, P., & Ölvander, J. (2013). Performance index and meta-optimization of a direct search optimization method. *Engineering optimization, 45*(10), 1167–1185. [CrossRef]

[36] Tripathi, D. P., & Jena, U. R. (2017, August). DSSO-directional shrinking search optimization. In *IOP Conference Series: Materials Science and Engineering* (Vol. 225, No. 1, p. 012280). IOP Publishing. [CrossRef]

[37] Zhang, W., Li, G., Zhang, W., Liang, J., & Yen, G. G. (2019). A cluster based PSO with leader updating mechanism and ring-topology for multimodal multi-objective optimization. *Swarm and Evolutionary Computation, 50*, 100569. [CrossRef]

[38] Li, X. (2009). Niching without niching parameters: particle swarm optimization using a ring topology. *IEEE Transactions on Evolutionary Computation, 14*(1), 150–169. [CrossRef]

[39] Wang, Y. X., & Xiang, Q. L. (2008, June). Particle swarms with dynamic ring topology. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)* (pp. 419-423). IEEE. [CrossRef]

[40] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE. [CrossRef]

[41] Houssein, E. H., Oliva, D., Samee, N. A., Mahmoud, N. F., & Emam, M. M. (2023). Liver cancer algorithm: A novel bio-inspired optimizer. *Computers in Biology and Medicine, 165*, 107389. [CrossRef]

[42] Xue, J., & Shen, B. (2023). Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *The Journal of Supercomputing, 79*(7), 7305–7336. [CrossRef]

[43] Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y.-P., Auger, A., & Tiwari, S. (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. *KanGAL report, 2005005*(2005), 2005.

[44] Tang, K., Li, X., Suganthan, P. N., Yang, Z., & Weise, T. (2007). Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization. *Nature inspired computation and applications laboratory, USTC, China, 24*, 1-18.

[45] Premkumar, M., Jangir, P., Kumar, B. S., Sowmya, R., Alhelou, H. H., Abualigah, L., ... & Mirjalili, S. (2021). A new arithmetic optimization algorithm for solving real-world multiobjective CEC-2021 constrained optimization problems: diversity analysis and validations. *IEEE Access, 9*, 84263-84295. [CrossRef]

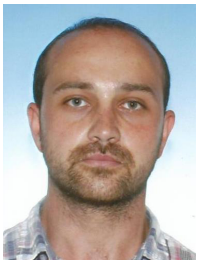
[46] Holland, J. H. (1992). Genetic algorithms. *Scientific american, 267*(1), 66-73.

[47] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization, 11*(4), 341-359. [CrossRef]

[48] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences, 179*(13), 2232-2248. [CrossRef]

[49] Yang, X. S. (2009, October). Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms* (pp. 169-178). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

Taymaz Akan is a postdoctoral fellow in the School of Medicine at Louisiana State University, USA (Shreveport). He received his Ph.D. from the Department of Computer Engineering at Gazi University, Ankara, Turkey, in 2019. He was the Director of the Distance Education Research and Application Center and also the Vice Dean of the Faculty of Engineering at Topkapi University.



Tomáš Zálabský is the head of the research center at the Faculty of Electrical Engineering and Informatics. He is most interested in the area of design, synthesis, simulation, and optimization of antenna elements, arrays, and fields. He also has extensive experience with the design and implementation of passive high-frequency circuits, such as planar and waveguide filters, directional taps, or power dividers.

Kimia Shirini received the Ph.D. degree in Computer Engineering from the University of Tabriz, Tabriz, Iran. Her research interests include optimization, scheduling algorithms, soft computing, and evolutionary algorithms

Jorge Ramos-Frutos received a B.S. degree in industrial engineering from the Instituto Tecnológico de Jiquilpan, Michoacán, Mexico, in 2018. He received his M.Sc. and Ph. D. degrees in Science and Technology from the Centro Innovación Aplicada en Tecnologías Competitivas (CIATEC), León, Guanajuato, Mexico, in 2021 and 2025, respectively. Currently is an Assistant Professor at the National Technological Institute of Mexico, Jiquilpan campus, in Mexico. His research interests are digital image processing, evolutionary computation, and machine learning.

Prof. Diego Oliva received a B.S. degree in Electronics and Computer Engineering from the Industrial Technical Education Center (CETI) of Guadalajara, Mexico, in 2007 and an M.Sc. degree in Electronic Engineering and Computer Sciences from the University of Guadalajara, Mexico, in 2010. He obtained a Ph. D. in Informatics in 2015 from the Universidad Complutense de Madrid. Currently, he is an Associate Professor at the University of Guadalajara in Mexico. He is a member of the Mexican National Research System (SNII), a Senior member of the IEEE, and a member of the Mexican Academy of Computer Sciences (AMEXCOMP). His research interests include evolutionary and swarm algorithms, hybridization of evolutionary and swarm algorithms, and computational intelligence.

Mohammad Alfrad Nobel Bhuiyan PhD joined the Department of Internal Medicine at LSU Health Shreveport in August 2021, where he currently serves as Assistant Professor of Internal Medicine and Director of Biostatistics and Computational Research. He earned his MSc and PhD in the Big Data track from the University of Cincinnati. During his PhD, Dr. Bhuiyan developed a novel Bayesian shape-invariant growth curve model to analyze growth trajectories.