SRTTU

Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

JECEI

Research paper

WHOFWA: An Effective Hybrid Metaheuristic Algorithm Based on Wild Horse Optimizer and Fireworks Algorithm

A. Rouhi*, E. Pira

Faculty of Information Technology and Computer Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran.

Article Info

Article History:
Received 30 November 2023
Reviewed 30 December 2023
Revised 25 January 2024
Accepted 12 February 2024

Keywords:
Wild horse optimizer
Fireworks algorithm
CEC 2005
CEC 2019
CEC 2022
Convergence speed
Hit rate

*Corresponding Author's Email Address: *rouhi@azaruniv.ac.ir*

Abstract

Background and Objectives: This paper explores the realm of optimization by synergistically integrating two unique metaheuristic algorithms: The Wild Horse Optimizer (WHO) and the Fireworks Algorithm (FWA). WHO, inspired by the behaviors of wild horses, demonstrates proficiency in global exploration, while FWA emulates the dynamic behavior of fireworks, thereby enhancing local exploitation. The goal is to harness the complementary strengths of these algorithms, achieving a harmonious balance between exploration and exploitation to enhance overall optimization performance.

Methods: The study introduces a novel hybrid metaheuristic algorithm, WHOFWA, detailing its design and implementation. Emphasis is placed on the algorithm's ability to balance exploration and exploitation. Extensive experiments, featuring a diverse set of benchmark optimization problems, including general test functions and those from CEC 2005, CEC 2019, and 2022, assess WHOFWA's effectiveness. Comparative analyses involve WHO, FWA, and other metaheuristic algorithms such as Reptile Search Algorithm (RSA), Prairie Dog Optimization (PDO), Fick's Law Optimization (FLA), and Ladybug Beetle Optimization (LBO).

Results: According to the Friedman and Wilcoxon signed-rank tests, for all selected test functions, WHOFWA outperforms WHO, FWA, RSA, PDO, FLA, and LBO by 42%, 55%, 74%, 71%, 48%, and 52%, respectively. Finally, the results derived from addressing real-world constrained optimization problems using the proposed algorithm demonstrate its superior performance when compared to several well-regarded algorithms documented in the literature.

Conclusion: In conclusion, WHOFWA, the hybrid metaheuristic algorithm uniting WHO and FWA, emerges as a powerful optimization tool. Its unique ability to balance exploration and exploitation yields superior performance compared to WHO, FWA, and benchmark algorithms. The study underscores WHOFWA's potential in tackling complex optimization problems, making a valuable contribution to the realm of metaheuristic algorithms.

Introduction

In recent years, there has been significant advancement in the field of metaheuristic optimization algorithms, with diverse algorithms demonstrating success in addressing complex optimization problems [1].

Swarm-based optimization methods, such as particle swarm optimization (PSO) [2], Harris's hawk optimization (HHO) [3], gazelle optimization algorithm (GOA) [4], to name a few, have shown limitations in handling both simple and intricate problems. Notably, widely recognized algorithms like gray wolf optimizer (GWO) [5], whale optimization algorithm (WOA) [6], and moth–flame optimization (MFO) [7] share structural similarities. Despite their popularity, the NFL theorem posits that no single algorithm can effectively address all optimization problems [8]. Each algorithm comes with specific strengths and limitations that impact their efficacy across various problem domains. Recognizing this, researchers have explored hybrid approaches, aiming to harness the strengths of multiple algorithms and create more robust optimization tools.

The Wild Horse Optimizer (WHO) has demonstrated efficiency in tackling specific optimization problems [9], yet its adaptability to a broader range of landscapes is a consideration. Similarly, the Fireworks Algorithm (FWA) shows promise in solving complex problems, but consistent delivery of optimal solutions across various functions remains a challenge [10].

Motivated by the aim to overcome the individual limitations of WHO and FWA, we introduce the Wild Horse and Fireworks Algorithm (WHOFWA) as a hybrid metaheuristic amalgamating the strengths of both algorithms. The goal is to leverage WHO's adaptability and FWA's exploration-exploitation balance, offering a comprehensive solution excelling in diverse optimization challenges. In response to challenges faced by traditional optimization methods, including difficulties in exploration-exploitation balance and sensitivity to parameters, the motivation behind developing WHOFWA lies in addressing these specific issues. The integration of WHO's adaptability and FWA's exploration capabilities positions WHOFWA as a comprehensive solution. This paper presents a detailed investigation into the development, performance, and versatility of WHOFWA, showcasing its potential to emerge as a leading metaheuristic algorithm in the domain of optimization.

To substantiate the effectiveness of WHOFWA, we conducted a comprehensive evaluation encompassing a diverse set of benchmark optimization problems. These include 10 general test functions, 7 test functions from the CEC 2005, 9 test functions from the CEC 2019 competition, and 12 test functions from the CEC 2022 competition. WHOFWA's performance is meticulously benchmarked against both WHO and FWA, alongside several other contemporary algorithms such as the Reptile Search Algorithm (RSA) [11], Prairie Dog Optimization (PDO) [12], Fick's Law Optimization (FLA) [13], and Ladybug Beetle Optimization (LBO) [14]. The evaluation is conducted using a variety of performance metrics, including the proximity to optima, early convergence, and hit rate (accuracy).

The results of our rigorous experimentation affirm the superior performance of WHOFWA. Not only does it outperform WHO and FWA, but it also surpasses the capabilities of RSA, PDO, FLA, and LBO across multiple facets of optimization. In an era where efficiency and accuracy are paramount, WHOFWA emerges as a promising and versatile solution that offers profound implications for optimization problems across diverse domains. This paper delves into the intricate design and operational intricacies of WHOFWA, presenting a compelling case for its adoption as a powerful tool in the ever-expanding toolkit of optimization practitioners.

The organization of this paper is as follows. Initially, a review will be conducted on some of the popular nature-inspired optimization algorithms. Following that, a brief introduction to the Fireworks and Wild Horse Optimization algorithms will be presented in the Preliminaries section to motivate their combination in the proposed WHOFWA algorithm. Subsequently, a detailed presentation and discussion of the hybrid algorithm will be provided. The Experimental Results section will present the evaluation framework of the proposed algorithm in comparison to some of the recently presented popular algorithms. Finally, the paper will be concluded, and some guidelines for future works will be provided.

Literature Review

Optimization, as a fundamental pursuit in problem-solving across various domains, has witnessed a surge in the development of metaheuristic algorithms (MAs) [15]-[19].

MAs are broadly divided into two categories: individual-solution-based and population-based [20], [21]. The latter, known for its effectiveness in exploring and exploiting the search space to target global optima, begins with a randomly generated population of solutions [22], [23]. Population-based MAs draw inspiration from evolutionary processes, natural phenomena, and social behaviors. Evolutionary Algorithms (EAs), including Genetic Algorithm (GA) [24] and Genetic Programming (GP) [25], emulate natural evolution. Natural Phenomenon (NP) algorithms, such as Simulated Annealing (SA) [26], Energy Valley Optimizer (EVO) [27], and others [13], [28]-[31], leverage physical and chemical principles. Social Behaviors (SBs) algorithms are further divided into Swarm Intelligence (SI) algorithms and Human Behaviors (HB) algorithms. SI algorithms, like Ant Colony Optimization (ACO) [32], simulate self-organized behaviors observed in nature, while HB algorithms include PSO [33] and various others [4], [11], [12], [34]-[37] that mimic human and animal behaviors.

Referencing the vast landscape of MAs poses a challenging task [20]. For the sake of conciseness, this section critically examines existing nature-inspired optimization algorithms, highlighting their applications, key features, and identified strengths/weaknesses. Additionally, it identifies research gaps in the domain of metaheuristic algorithms and emphasizes the innovations introduced by the WHOFWA algorithm.

A. Nature-Inspired Optimization Algorithms

The landscape of optimization techniques has been enriched by a plethora of nature-inspired algorithms, each drawing inspiration from different natural phenomena. Table 1 provides a summary of popular population-based nature-inspired algorithms, their nature inspirations, key features, and applications. These algorithms have found applications in diverse fields such as engineering, genetics, finance, robotics, and image processing [15], [20], [21].

B. Research Gaps in Metaheuristic Algorithm Domain

The ongoing research for algorithms that effectively balance global exploration and local exploitation is a central focus in optimization research [1]. Despite the proliferation of nature-inspired algorithms, there exists a need for a critical analysis of their limitations and the identification of research gaps. Table 2 summarizes nature-inspired popular algorithms to highlight their unique strengths/weaknesses more clearly. This review recognizes the importance of addressing the limitations faced by popular algorithms, such as premature convergence, sensitivity to parameters, and inefficiency in high-dimensional problems. The identified gaps set the stage for the introduction of innovative solutions, motivating the development of hybrid metaheuristic algorithms like WHOFWA.

C. Innovations of WHOFWA in the Metaheuristic Algorithm Landscape

The WHOFWA algorithm emerges as a pioneering hybrid metaheuristic, combining the strengths of WHO and FWA. WHO excels in global exploration, while FWA specializes in dynamic local exploitation. The integration of these distinct approaches in WHOFWA aims to strike a harmonious equilibrium between exploration and exploitation.

Unlike existing algorithms, WHOFWA addresses the limitations of individual approaches by leveraging the collective intelligence of wild horses and the explosive search behavior of fireworks. This strategic fusion is designed to extend the boundaries of optimization capabilities, redefining the state-of-the-art in the field.

D. Synthesis of related work and WHOFWA's Contribution

The presented literature review critically evaluates existing algorithms, identifies research gaps, and positions WHOFWA as an innovative solution to address these gaps. The WHOFWA algorithm's ability to outperform not only WHO and FWA but also other contemporary algorithms underscores its significance in the landscape of metaheuristic algorithms. The following sections delve into the design, implementation, and comprehensive evaluation of WHOFWA, providing a thorough understanding of its capabilities and contributions.

Table 1: Summary of some popular nature-inspired optimization algorithms

Algorithm	Nature Inspiration	Key Features	Applications
Genetic Algorithm (GA) [38]	Genetics and Evolution	Population-based search, crossover and mutation operators, selection strategies	Engineering, genetics, finance
Particle Swarm Optimization (PSO) [2]	Flocking behavior	Swarm-based optimization, velocity updates, social and cognitive components	Robotics, clustering, neural networks
Ant Colony Optimization (ACO) [32]	Ant Foraging Behavior	Pheromone-based communication, path selection based on probability	Routing, network design, scheduling
Simulated Annealing (SA) [39]	Annealing Process	Temperature-based exploration, probabilistic acceptance of worse solutions	Combinatorial optimization, scheduling
Fireworks Algorithm (FWA) [10]	Firework Explosions	Population-based optimization, hierarchical explosions, more exploitation capability through good explosions	Image processing, feature selection
Artificial Bee Colony (ABC) [40]	Bee Foraging Behavior	Exploration via scout bees, local exploitation by employed bees, global exploitation by onlookers	Clustering, function optimization
Grey Wolf Optimizer (GWO) [5]	Grey Wolf Pack Behavior	Alpha, beta, delta, and omega wolves, mimicking social hierarchy	Engineering design, neural networks

Algorithm	Nature Inspiration	Key Features	Applications
Cuckoo Search (CS) [41]	Cuckoo Bird Nesting	Levy flights, host nest replacement, parameter tuning	Function optimization, image processing
Harmony Search (HS) [42]	Music Harmony Creation	Memory consideration, improvisation and exploration, harmony memory	Engineering design, parameter tuning
Bat Algorithm (BA) [43]	Bat Echolocation	Frequency tuning, pulse emission, adaptive loudness, and rate	Function optimization, feature selection
Whale Optimization Algorithm (WOA) [6]	Whale Hunting	Encircling prey, bubble-net hunting, exploration, and exploitation	Image processing, feature selection
Wild Horse Optimizer (WHO) [9]	Wild Horse Behavior	Herd movement, leader (stallions) and follower (foals) roles, herd updates based on stallion positions	Function optimization, engineering design
Artificial Immune System (AIS) [44]	Immune System Response	Clonal selection, mutation, antibody-antigen interactions	Anomaly detection, data mining
Moth Flame Optimization (MFO) [45]	Moth Attraction to Light	Attraction to light sources, update equations, exploration, exploitation	Image processing, function optimization
Flower Pollination Algorithm (FPA) [46]	Pollination Process	Pollination strategies, global and local pollination, pollen dispersal	Optimization, control systems
Krill Herd Algorithm (KHA) [47]	Krill Swarming Behavior	Krill movement, herding, search for prey, krill population	Function optimization, image segmentation
Dragonfly Algorithm (DA) [48]	Dragonfly Hunting	Prey detection, hunting strategy, escape behavior, position update	Image processing, function optimization
League Championship Algorithm (LCA) [49]	Sports Leagues	Competition, playoffs, promotion, and relegation mechanisms	Sports scheduling, tournament design
Social Spider Algorithm (SSA) [50]	Social Spider Behavior	Web construction, prey capture, web update, social cooperation	Network design, web optimization
Crow Search Algorithm (CSA) and Otsu Hybrid [51]	Crow's intelligent feeding behavior	Optimizes Otsu method for obtaining optimum thresholds, used for maximum variance within the cluster, simplicity with fewer parameters, decreased computational time in multilevel thresholding	Image segmentation with gray scale images, applicable in real-time applications
Ladybug Beetle Optimization (LBO) [14]	Ladybugs in Nature	Coordinated movement inspired by ladybugs searching for a warm place in winter, population update based on the position of other ladybugs, ignoring worst members to increase search speed	Applied to Economic-Environmental Dispatch Problem (EEDP), engineering applications, used for Covid-19 modeling and forecasting
Dandelion Optimizer (DO) [52]	Dandelion seed flight stages (rising, descending, landing)	Mathematical modeling of seed flight stages under different weather conditions, swarm intelligence based algorithm	Engineering optimization problems, real-world applications in speed reduction, spring design, beam design, and pressure vessel design
Barnacle Mating Optimizer (BMO) [53]	Barnacles' Mating Behavior	Bio-inspired evolutionary algorithm, Mimics mating process of barnacles, Utilizes Hardy-Weinberg principle in offspring generation	Real application in solving engineering optimization problems
Gazelle Optimization Algorithm (GOA) [4]	Gazelles life in nature and their survival ability	Population-based optimization method, two-phase approach: exploration (grazing) and exploitation (outrunning and outmaneuvering predators)	Optimization problems in various fields (engineering, medicine, computer science, production chain)

Table 2: Summary of strengths/weaknesses of popular nature-inspired optimization algorithms

Algorithm	Strengths	Weaknesses
Genetic Algorithm (GA) [38]	Parallelism, global optimization, robustness to noisy environments	Convergence speed may be slow, sensitive to parameter settings, not suitable for all optimization problems
Particle Swarm Optimization (PSO) [2]	Simple implementation, global optimization, fast convergence	Sensitive to parameter settings, may get stuck in local optima
Ant Colony Optimization (ACO) [32]	Good for discrete optimization problems, robust to changes in problem structure, inspired by natural systems	Limited scalability for large problems, may converge to suboptimal solutions
Simulated Annealing (SA) [39]	Global optimization, effective for complex landscapes, escapes local optima	Slow convergence, sensitive to temperature scheduling
Fireworks Algorithm (FWA) [10]	Good for multimodal optimization, diversity maintenance, parallel processing	Parameter tuning required, convergence speed may vary
Artificial Bee Colony (ABC) [40]	Simplicity and ease of implementation, handles continuous optimization problems, robust to parameter settings	Poor exploration/exploitation balance, sensitive to initial solution
Grey Wolf Optimizer (GWO) [5]	Mimics social behavior of wolves, good exploration/exploitation balance, simplicity and ease of implementation	Performance may degrade with large-scale problems, sensitive to parameter settings
Cuckoo Search (CS) [41]	Global optimization, simplicity and ease of implementation, handles non-continuous optimization problems	Convergence speed may vary, limited scalability for large problems
Harmony Search (HS) [42]	Good exploration/exploitation balance, versatile and easy to implement, effective for discrete optimization problems	Parameter tuning required, may converge to suboptimal solutions
Bat Algorithm (BA) [43]	Global optimization, good exploration/exploitation balance	Sensitive to parameter settings, limited scalability for large problems
Whale Optimization Algorithm (WOA) [6]	Global optimization, simplicity and ease of implementation	Convergence speed may vary, limited scalability for large problems
Wild Horse Optimizer (WHO) [9]	Global optimization, simplicity and ease of implementation	Convergence speed may vary, limited scalability for large problems
Artificial Immune System (AIS) [44]	Robustness to noisy environments, adaptive and self-learning	Computational complexity, difficulty in parameter tuning
Moth Flame Optimization (MFO) [45]	Good exploration/exploitation balance, simplicity and ease of implementation	Limited scalability for large problems, convergence speed may vary
Flower Pollination Algorithm (FPA) [46]	Global optimization, simplicity and ease of implementation	Limited scalability for large problems, convergence speed may vary
Krill Herd Algorithm (KHA) [47]	Good exploration/exploitation balance, simplicity and ease of implementation	Limited scalability for large problems, convergence speed may vary
Dragonfly Algorithm (DA) [48]	Good exploration/exploitation balance, simplicity and ease of implementation	Limited scalability for large problems, convergence speed may vary
League Championship Algorithm (LCA) [49]	Global optimization, simplicity and ease of implementation	Limited scalability for large problems, convergence speed may vary
Social Spider Algorithm (SSA) [50]	Good exploration/exploitation balance, inspired by social behavior	Limited scalability for large problems, convergence speed may vary

Preliminaries

This section first provides a description of original algorithms of the proposed hybrid algorithm WHOFWA, i.e., WHO and FWA, respectively, and then motivates the integration of them in WHOFWA.

A. Wild Horse Optimization (WHO)

Wild Horse Optimizer (WHO) algorithm, inspired by the social organization and behaviors of non-territorial horses [38]. The focus is on the optimization of various problems using group behaviors, grazing, mating, domination, and leadership.

The core components of WHO include position updating mechanisms guided by individual and group influences, and adaptive parameters that enhance the algorithm's adaptability to different problem landscapes. WHO has demonstrated success in solving optimization problems, particularly in unimodal landscapes, thanks to its ability to efficiently explore the solution space.

The WHO algorithm can be summarized as shown in Algorithm 1.

B. Fireworks Algorithm (FWA)

The Fireworks Algorithm (FA) is a swarm intelligence algorithm inspired by the emergent swarm behavior of fireworks for function optimization [9]. It simulates the explosion process of fireworks, where two explosion (search) processes are employed to generate sparks. The algorithm aims to find an optimal point in the search space by iteratively setting off "fireworks," evaluating sparks' locations, and selecting new locations for the next generation of explosions based on the current sparks and fireworks.

Algorithm 1: Wild Horse Optimizer (WHO)

Initialize the first population of horses randomly
Calculate the fitness of horses
Create foal groups and select their Stallions as group leaders
Calculate the fitness of each horse in the herd and select the best one as *Optimum*
While the end criterion is not satisfied
  For each $Stallion_i$
    For each Foal member of the group leading by $Stallion_i$
      Randomly move the Foal to search around the leader, i.e., $Stallion_i$ Or leave the group and mate with the foals of other groups (which left their groups as well) through crossover aimed at exploring new spaces
    End For
    Randomly update the position of the $Stallion_i$ with ∓ WH, hopefully reaching water hole, i.e., best positions of search space
    If fitness of $Stallion_i$ is better **than** *Optimum* then Optimum will be updated to $Stallion_i$
    Sort the foals of $Stallion_i$'s group members based on their fitness values and exchange the position of best foal with the $Stallion_i$ to form a new group leader
  End For
  Update *Optimum*
End While

A *good firework explosion* is characterized by numerous sparks that centralize around the explosion center, creating a spectacular display. On the contrary, a *bad firework explosion* is defined by the generation of few sparks, which scatter in the space, resulting in a less impressive outcome.

Despite its strengths, FWA may encounter challenges in certain function landscapes, where the balance between exploration and exploitation becomes critical. The sensitivity of FWA to parameter settings and specific problem characteristics motivates the exploration of hybridization with other algorithms to enhance its overall performance.

The FWA algorithm can be summarized as shown in Algorithm 2.

C. Integration of WHO and FWA in WHOFWA

Motivated by the distinctive features of WHO and FWA, their hybridization in the Wild Horse and Fireworks Algorithm (WHOFWA) aims to capitalize on the strengths of both algorithms. WHO's efficiency in unimodal landscapes complements FWA's robust exploration in multimodal scenarios. WHOFWA integrates the collective movement and adaptability of wild horses from WHO with the explosive exploration capabilities inspired by fireworks in FWA. This amalgamation is expected to create a metaheuristic algorithm that not only addresses the limitations of its parent algorithms but also excels in tackling diverse optimization challenges.

The subsequent sections detail the design, implementation, and extensive experimental evaluations of WHOFWA, shedding light on its performance across a range of test functions and establishing its potential as a competitive and versatile optimization tool.

Algorithm 2: Fireworks Algorithm (FWA)

Randomly select ***n*** locations for fireworks.
While the *stop criteria* are **false**
  Set off n fireworks at the respective locations.
  For each firework $\boldsymbol{x_i}$
    Calculate the number of sparks considering high numbers to the fireworks with better fitness.
    Obtain locations of sparks.
  Generate ***m*** sparks for some of randomly selected fireworks.
  Select the best location for the next explosion generation.
  Randomly select ***n* − *1*** locations from the sparks and current fireworks based considering the high probability to the good firework explosions.
End While

The Proposed Algorithm

WHO algorithm has parameters *N* (total number of horses), *G* = *N* * *PS* (total number of groups), *PS* (percentage of male horses in the total population). As a result, we will have *G* head group (male horse) and *N - G* remaining horses are divided between these groups, each group will have *(N - G) / G* members. Leaders are usually the best in groups. FA algorithm has *Nfws* number of firework and each firework produces a number of sparks.

In the proposed hybrid algorithm (WHOFWA), the heads of groups (male horses) play the role of sparks, and the rest of the horses of each group play the role of sparks on each firework.

Algorithm 3 shows the WHOFWA algorithm pseudocode. In this algorithm, *Nfws*, *maxEva*, and *gaussianNum* denote the number of fireworks, the maximum number of fitness function evaluations, and the number of solutions produced in a local Gaussian space around the fireworks using the Gaussian explosion process, respectively. In line 1, a population (*Fw*) of sparks with the size of *Nfws* is randomly generated and then in line 2, the fitness of sparks is calculated (*Fwfit*). In line 5 of the algorithm, the sonsnum_cal function calculates the number of sparks of each firework and places in the array sonsnum_array. In line 6, the scope_cal function calculates the explosion range of sparks around each firework and stores it in scope_array. In line 7, the sons_generate function generates the sparks based on the number of sparks and the explosion range of each firecracker and keeps them in the Sons variable.

Algorithm 3: The **WHOFWA** algorithm

```
Input: Nfws, maxEva, gaussianNum;
Output: A solution with the best fitness.
1: Initialize the population of fireworks Fw (Fw_1, ..., Fw_Nfws);
2: Evaluate the fitness of each firework Fwfit (Fwfit_1, ..., Fwfit_Nfws);
3: bst = select the best firewrok; evaCount = Nfws;
4: while evaCount <= maxEva do
      // compute the number of sons that each seed should generate
5:     sonsnum_array = sonsnum_cal(Fwfit, Nfws);
      // compute the exploding scope of sons that each seed generate
6:     scope_array = scope_cal(Fwfit, Nfws);
      // generate the sparks, based on the sparks number and explosion amplitude
      // of each firework
7:     Sons= sons_generate (sonsnum_array, scope_array, Fw,bst);
8:     Evaluate the fitness of each spark Sonsfit (Sonsfit_1, ..., Sonsfit_Nfws);
9:     evaCount = evaCount + Nfws;
      ///////// Wild horse optimizer ////////////
      // All generated fireworks are considered Stallions
      // All sparks of each firework are considered foals for a Stallion
10:   Nstallion = Nfws; Stallion = Fw; Stallionfit = Fwfit;
11:   for i = 1 to Nfws do
12:        Nc = sonsnum_array (i) ; sp = 0;
13:      for j=1 to i-1do
14:          sp = sp + sonsnum_array (j);
15:      end for
16:        Stallion (i) . group = Sons (sp +1: sp + Nc);
17:        Stallion (i) . groupfit = Sonsfit (sp +1: sp + Nc);
18:   end for
19:   MaxiterWHO =1;
20:   StallionNew = WHO (Stallion , MaxiterWHO);
21:   evaCount = evaCount + length (StallionNew);
      //////////////////////////////////////////////////////////////
21:   Update Fw and Sons if StallionNew is better than Stallion;
      // perform the Gaussian mutation of seeds
22:   FwGauss= seedGaussMutation(Fw, bst, gaussianNum);
23:   Evaluate the fitness of each Gaussian affected; evaCount = evaCount + gaussianNum;
24:   All = Fw ∪ Sons ∪ FwGauss; Allfit = Fwfit ∪ Sonsfit ∪ FwGaussfit;
      // select the next iteration
25:   Fw = selectNextIterationOnEntropy(All, Allfit, Nfws);
26:   Update bst due to Fw;
27: end while
28: return bst;
```

Lines 10-21 show the initialization and execution of the WHO algorithm. In line 10, the number of stallions (*Nstallion*) is equal to the number of fireworks (*Nfws*) and the fireworks and their fitness values are copied into the *Stallion* and *Stallionfit* variables, respectively. In lines 11-18, all sparks are copied into the group variable of *Stallion* as Foals. In line 20, the WHO algorithm is invoked and executed one time. In line 21, the solutions returned from the execution of the WHO algorithm are copied into the *FW* and *Sons* variables if they are better. In line 22, Gaussian mutation is performed on the solutions in *Fw* with the seedGaussMutation function. In line 26, among the solutions available in *Fw*, *Sons* and *FwGauss*, number of *Nfws* solution is selected as fireworks of the next step.

Experimental Results

In order to rigorously assess the performance of our presented hybrid metaheuristic algorithm, WHOFWA, and to establish its superiority over existing optimization algorithms, we conducted a comprehensive set of experiments on a range of benchmark optimization problems [54], [55]. For all experiments, the maximum number of function evaluations (*maxEva*) was set to 10^5. The parameter settings for all algorithms used in the experiments are shown in Table 3. From parameters of the WHOFWA algorithm, listed in Table 3, the following two parameters have a great impact on the effectiveness of this algorithm: *Nfws* (the number of fireworks) and *Mn* (maximum number of sparks). In order to analyze this impact, the WHOFWA is executed for various *Nf* and *Mn* values, considering F1 and F2 from the general test functions, F4 and F5 from the CEC 2019 functions, and F1 and F3 from the CEC 2022 functions. As indicated in Table 4, the best effectiveness for WHOFWA is obtained when *Nfws* and *Mn* are equal to 3 and 40, respectively.

The parameter settings were carefully chosen to ensure a fair and standardized comparison between the WHOFWA algorithm and the benchmark algorithms. Each algorithm was executed on the same set of optimization problems with these settings, and the results were recorded and analyzed.

In the subsequent sections, we present the results of these experiments, providing a detailed analysis of the performance of WHOFWA in comparison to WHO, FA, RSA, PDO, FLA, and LBO. These results are pivotal in demonstrating the superior efficacy of WHOFWA as an optimization tool in various contexts.

Experiment 1: 10 General Test Functions

In the pursuit of evaluating the performance of the WHOFWA algorithm and comparing it to other state-of-the-art optimization algorithms, we begin by examining its behavior on a set of 10 general test functions (see Table 5 for more details). These functions are well-established benchmarks in the field of optimization and are often used to assess the capabilities of optimization algorithms in terms of convergence speed, accuracy, and their ability to navigate various landscapes.

Table 3: Parameter settings of all algorithms (*maxEva* = 10E+5)

Algorithm	Appropriate parameter values
WHOFWA	*The number of fireworks* = 3, *The total number of sparks* = 40, *N* = 30, *Crossover rate* = 0.13, *Stallions percentage* = 0.1
WHO	*Crossover rate* = 0.13, *Stallions percentage* =0.2, *N* = 50
FA	*The number of fireworks* = 5, *The total number of sparks* = 50
RSA	*Alpha* = 0.1, *Beta* = 0.005, *N* = 30
PDO	*Rho* = 0.005, *epsPD* = 0.1, *N* = 30
FLA	*C1* = 0.5, *C2* = 2, *C3* = 0.1, *C4* = 0.2, C5 = 2, *D* = 0.01, *N* = 30
LBO	*N (0)= 60*, β =10, N_{min} = 0.25*N*(0)

For each of these test functions, the WHOFWA algorithm, alongside other benchmark algorithms (WHO, FA, RSA, PDO, FLA, and LBO), is subjected to rigorous evaluation. The performance metrics considered include proximity to the optimum, early convergence, and hit rate (accuracy). The results of this experiment provide crucial insights into the ability of WHOFWA to solve general optimization problems and its relative performance when compared to established algorithms. These findings will be presented in the subsequent sections, shedding light on the efficacy and versatility of WHOFWA in tackling various optimization landscapes.

Table 4: The results of executing WHOFWA for different *Nf and Mn*

Type	Function	Metric	*Nfws* = 2 *Mn* = 30	*Nfws* = 2 *Mn* = 40	*Nfws* = 2 *Mn* = 50	*Nfws* = 3 *Mn* = 30	*Nfws* = 3 *Mn* = 40	*Nfws* = 3 *Mn* = 50	*Nfws* = 4 *Mn* = 30	*Nfws* = 4 *Mn* = 40	*Nfws* = 4 *Mn* = 50
General	F1	Best	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00
	F2	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
CEC 2019	F4	Best	23.1441	22.6695	14.7374	24.1994	15.8521	11.3147	41.8829	30.8967	18.0331
	F5	Best	1.138	1.196	1.1591	1.1621	1.1001	1.138	1.3331	1.1666	1.1538
CEC 2022	F1	Best	319.6524	313.4741	324.1513	310.4829	307.2631	311.3363	314.2728	316.3664	314.5829
	F3	Best	600.0216	600.0042	600.0192	600.0960	600.0021	600.0222	600.0607	600.2766	600.1084

Table 5: The details of general test functions

Name	Formulation	Dimension (D)	Range	Optimum value	Type
F1 (Easom)	$f(X) = -\cos(X_1)\cos(X_2)\exp(-(X_1-\pi)^2 - (X_2-\pi)^2)$	2	[-100, 100]	-1	**Unimodal**
F2 (Sphere)	$f(X) = \sum_{i=1}^{D} {X_i}^2$	30	[-100, 100]	0	
F3 (SumSquares)	$f(X) = \sum_{i=1}^{D} i{X_i}^2$	30	[-10, 10]	0	
F4 (Schwefel 2.22)	$f(X) = \sum_{i=1}^{D} \lvert X_i \rvert + \prod_{i=1}^{D} \lvert X_i \rvert$	30	[-10, 10]	0	
F5 (Schwefel 1.2)	$f(X) = \sum_{i=1}^{D} (\sum_{j=1}^{i} X_j)^2$	30	[-100, 100]	0	
F6 (Bohachevsky1)	$f(X) = X_1^2 + 2X_2^2 - 0.3cos(3\pi X_1) - 0.4cos(4\pi X_2) + 0.7$	2	[-100, 100]	0	**Multimodal**
F7 (Booth)	$f(X) = (X_1 + 2X_2 - 7)^2 + (2X_1 + X_2 - 5)^2$	2	[-10, 10]	0	
F8 (Schaffer)	$f(X) = 0.5 + \frac{sin^2\left(\sqrt{X_1^2 + X_2^2}\right) - 0.5}{(1 + 0.001(X_1^2 + X_2^2))^2}$	2	[-100, 100]	0	
F9 (Michalewicz10)	$f(X) = -\sum_{i=1}^{D} \sin(X_i)(\sin(iX_i^2/\pi))^{20}$	10	[0, π]	-9.6602	
F10 (Griewank)	$f(X) = \frac{1}{4000}\left(\sum_{i=1}^{D}(X_i - 100)^2\right) - \left(\prod_{i=1}^{D} \cos(\frac{X_i - 100}{\sqrt{i}})\right) + 1$	30	[-600, 600]	0	

In Table 6, we present the comparative results for the 10 general test functions, showcasing the performance of the WHOFWA algorithm in comparison to other optimization algorithms, namely WHO, FA, RSA, PDO, FLA, and LBO. The results are categorized based on three key metrics: “Best” (the best solution found) and “Ave” (the average solution found), and “Std” (standard deviation) of the solutions. Let's delve into the analysis and discussion of these results:

1. **F1 (Easom)**: For the unimodal function F1, WHOFWA demonstrates its ability to reach the global optimum, achieving a best solution of -1.00E+00, matching the best result found by WHO and FA. The average solution is also -1.00E+00. WHOFWA maintains an impressively low standard deviation of 7.03E-07, indicating its consistency.
2. **F2 (Sphere)**: In the case of the spherical function F2, WHOFWA stands out by discovering the best solution of 2.60E-279, surpassing all other algorithms, which converge to zero. The average solution for WHOFWA is significantly better than other algorithms.
3. **F3 (SumSquares)**: WHOFWA once again excels by finding the global optimum (best) of 0.00E+00 for the SumSquares function. This is in contrast to other algorithms that also reach zero. The average result for WHOFWA is commendable, reflecting its ability to converge to the optimum consistently.
4. **F4 (Schwefel 2.22)**: WHOFWA continues to demonstrate its competitiveness with a best solution of 5.67E-143, which is notably better than the results of other algorithms. The average solution, while not reaching zero, is still quite low.
5. **F5 (Schwefel 1.2)**: WHOFWA excels in reaching the best solution of 3.27E-253, which is superior to other algorithms. The average result for WHOFWA is also impressive, outperforming the competitors.
6. **F6 (Bohachevsky1)**: For the multimodal function F6, WHOFWA manages to achieve the best solution of 0.00E+00, matching other algorithms. It

maintains a low standard deviation, reflecting consistent convergence.

7. **F7 (Booth)**: WHOFWA finds a competitive best solution of 2.14E-08, though it is not the best result. The average result is comparable to the performance of other algorithms.
8. **F8 (Schaffer)**: WHOFWA reaches the best solution of 0.00E+00, matching the results of other algorithms. The average solution is also in line with the performance of the competitors.
9. **F9 (Michalewicz10)**: WHOFWA stands out in finding the best solution of -9.66E+00, which is notably better than the other algorithms. The average solution is competitive, demonstrating the algorithm's consistency.
10. **F10 (Griewank)**: WHOFWA converges to the best solution of 0.00E+00, which matches the performance of other algorithms. The average solution is also in line with the results of competitors.

In sum, WHOFWA consistently exhibits competitive performance across the majority of the general test functions. It excels in finding the global optima for several unimodal and multimodal functions, with a remarkable ability to consistently reach or approach optimal values. The low standard deviations suggest that WHOFWA maintains stability and consistency in its results.

In conclusion, the results in Table 6 demonstrate that WHOFWA is a robust optimization algorithm that excels in solving a diverse set of general test functions. Its competitive performance positions it as a promising tool for tackling a wide range of real-world optimization problems, further substantiating its effectiveness and versatility in the field of metaheuristic algorithms.

In Table 7, the ranks of various algorithms, including WHOFWA, WHO, FA, RSA, PDO, FLA, and LBO, are provided for a set of general test functions. The rankings are based on the Friedman Test, and the lower the mean rank, the better the algorithm's overall performance across the tested functions.

In sum, the results in Table 7 reflect the relative performance of the algorithms across the general test functions. While WHOFWA performed well, it was outperformed by both FLA and FWA, which achieved the highest and second-highest mean ranks, respectively. These rankings suggest that FLA and FWA were the top-performing algorithms in this evaluation, while WHOFWA displayed strong competitiveness in third place.

It's important to note that the choice of evaluation metrics and test functions can influence algorithm rankings. Nevertheless, the results in Table 6 provide valuable insights into the relative performance of WHOFWA and other metaheuristic algorithms, reaffirming their effectiveness in solving general optimization problems.

Experiment 2: CEC 2019 Test Functions

In this section, we shift our focus to the evaluation of the WHOFWA algorithm on a set of challenging optimization problems known as the CEC 2019 test functions. These functions are designed to rigorously test the performance of optimization algorithms across diverse landscapes and complexities, making them a robust benchmark for assessing the capabilities of WHOFWA (see Table 8 for more details). For each of these CEC 2019 test functions, we will assess the performance of WHOFWA and compare it to other state-of-the-art optimization algorithms. The evaluation will be based on the key performance metrics such as proximity to the optimum, early convergence, and hit rate (accuracy). These results will provide valuable insights into the ability of WHOFWA to tackle complex and diverse optimization landscapes, further demonstrating its effectiveness in challenging scenarios.

Table 9 presents a comprehensive overview of the performance of the WHOFWA algorithm, as well as other optimization algorithms (WHO, FA, RSA, PDO, FLA, and LBO), across the CEC 2019 test functions. The results are divided into three key metrics: “Best” (the best solution found), “Ave” (the average solution found), and “Std” (the standard deviation of the solutions). Here's a brief analysis and discussion of the results:

1. **F1 (Storn's Chebyshev Polynomial Fitting Problem)**: WHOFWA achieved a best solution of 3.92E+04, which is competitive with the other algorithms. The average solution for WHOFWA is also noteworthy, although it ranks third among the algorithms. The standard deviation for WHOFWA is relatively low, indicating consistent performance.
2. **F2 (Inverse Hilbert Matrix Problem)**: WHOFWA obtained the best solution of 1.73E+01, matching the best results of other algorithms. The average solution for WHOFWA is consistent with other algorithms, demonstrating good performance on this function. WHOFWA maintains a very low standard deviation, indicating stability in its results.
3. **F3 (Lennard-Jones Minimum Energy Cluster)**: WHOFWA achieved the best solution of 1.27E+01, on par with the best results of other algorithms. The average solution for WHOFWA is consistent with other algorithms, demonstrating strong performance. The standard deviation for WHOFWA is notably low, indicating stability.
4. **F4 (Rastrigin's Function)**: WHOFWA reached a best solution of 1.58E+01, which is competitive, although it doesn't outperform the best results of other algorithms. The average solution for

WHOFWA ranks among the algorithms, demonstrating good performance. The standard deviation for WHOFWA is reasonable, indicating a stable performance.

5. **F5 (Griewank's Function)**: WHOFWA achieved the best solution of 1.10E+00, which is competitive with other algorithms. The average solution for WHOFWA is strong, ranking among the algorithms. WHOFWA maintains a reasonable standard deviation, suggesting consistent performance.
6. **F6 (Weierstrass Function)**: WHOFWA reached a best solution of 2.48E+00, competitive with the other algorithms. The average solution for WHOFWA is on par with the performance of other algorithms. The standard deviation for WHOFWA is reasonable, indicating stability.
7. **F8 (Expanded Schaffer's F6 Function)**: WHOFWA obtained the best solution of 3.25E+00, on par with the best results of other algorithms. The average solution for WHOFWA ranks among the algorithms, demonstrating strong performance. The standard deviation for WHOFWA is reasonable, indicating a stable performance.
8. **F9 (Happy Cat Function)**: WHOFWA achieved the best solution of 2.42E+00, matching the best results of other algorithms. The average solution for WHOFWA ranks among the algorithms, demonstrating strong performance. The standard deviation for WHOFWA is low, indicating consistent performance.
9. **F10 (Ackley Function)**: WHOFWA reached a best solution of 2.34E+00, which is competitive with the other algorithms. The average solution for WHOFWA is consistent with the performance of other algorithms. The standard deviation for WHOFWA is notably low, indicating stability.

Table 6: Comparative results for general test functions

Function	Metric	WHOFWA	WHO	FWA	RSA	PDO	FLA	LBO
F1	Best	-1.00E+00	-1.00E+00	-1.00E+00	-9.99E-01	-1.00E+00	-1.00E+00	-1.00E+00
	Ave	-1.00E+00	-1.00E+00	-1.00E+00	-9.89E-01	-5.00E-01	-1.00E+00	-1.00E+00
	Std	7.03E-07	0.00E+00	7.24E-06	1.37E-02	7.07E-01	1.96E-04	7.03E-07
F2	Best	2.60E-279	1.82E-117	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.95E-75
	Ave	7.18E-202	1.21E-111	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.26E-74
	Std	0.00E+00	1.80E-111	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.59E-74
F3	Best	0.00E+00	6.08E-119	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.84E-80
	Ave	3.46E-209	5.34E-114	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.29E-78
	Std	0.00E+00	7.63E-114	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.59E-78
F4	Best	5.67E-143	5.53E-66	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.51E-49
	Ave	5.15E-120	3.92E-59	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.33E-48
	Std	1.96E-119	6.78E-59	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.66E-48
F5	Best	3.27E-253	3.19E-109	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.52E-73
	Ave	2.45E-205	1.44E-107	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.75E-72
	Std	0.00E+00	1.39E-107	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.27E-72
F6	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Ave	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F7	Best	2.14E-08	5.07E-01	1.17E-04	1.31E-01	5.66E-02	2.75E-05	0.00E+00
	Ave	3.13E-06	5.07E-01	2.66E-04	3.20E-01	2.82E-01	3.79E-01	2.54E-01
	Std	4.77E-06	0.00E+00	2.10E-04	2.68E-01	3.19E-01	3.31E-01	3.59E-01
F8	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.37E-02
	Ave	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.91E-02	4.37E-02
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.52E-02	3.19E-13
F9	Best	-9.66E+00	-9.07E+00	-9.57E+00	-5.44E+00	-8.01E+00	-8.01E+00	-9.61E+00
	Ave	-9.58E+00	-8.56E+00	-9.44E+00	-5.43E+00	-7.50E+00	-7.50E+00	-9.61E+00
	Std	8.03E-02	6.05E-01	1.85E-01	1.35E-02	1.71E+00	7.28E-01	1.50E-03
F10	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Ave	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.60E-02
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.26E-02

Table 7: Ranks of WHOFWA and other algorithms of general test functions

		WHOFWA	WHO	FWA	RSA	PDO	FLA	LBO
Friedman Test	Mean Rank	3.60	5.00	3.25	4.20	3.60	3.40	4.95
	Rank	**3**	**7**	**1**	**5**	**4**	**2**	**6**

In sum, the results in Table 8 showcase the strong and competitive performance of the WHOFWA algorithm across the CEC 2019 test functions.

WHOFWA consistently achieved best solutions that are on par with or better than other algorithms and demonstrated competitive average solutions.

Moreover, WHOFWA maintained low standard deviations, indicating stable and reliable performance across these complex optimization landscapes.

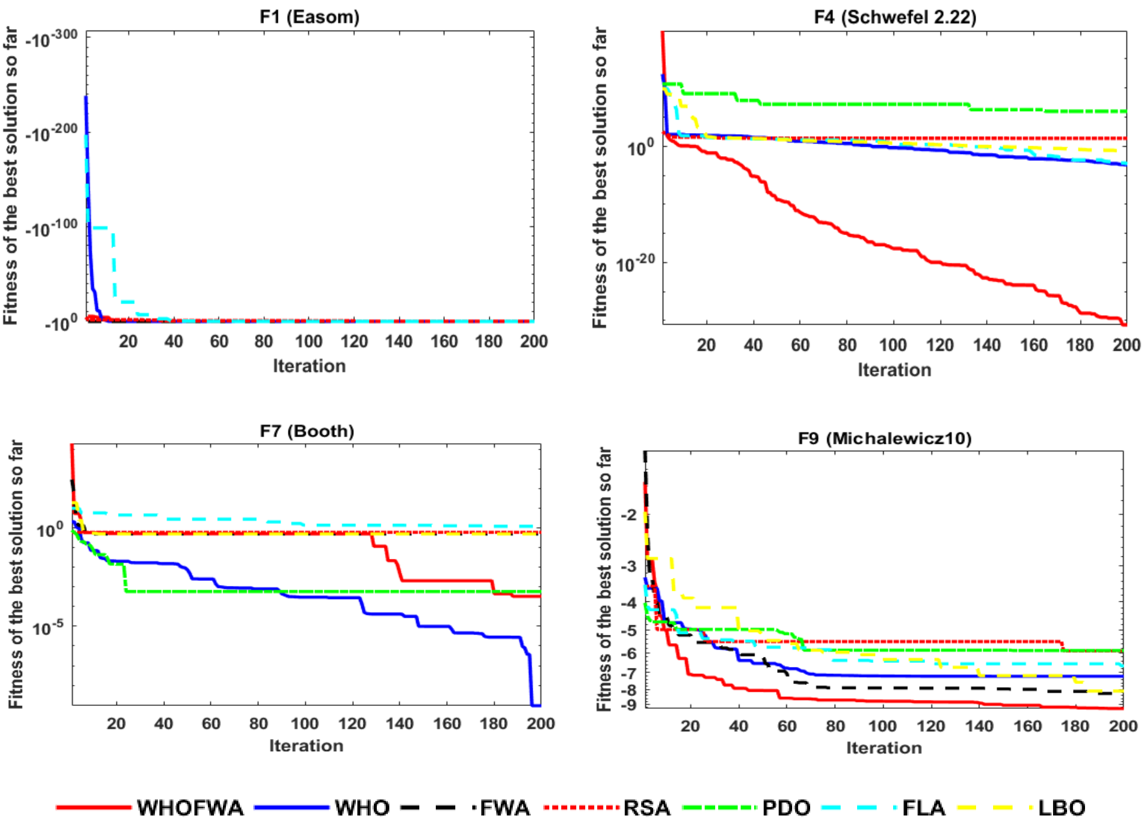

Fig. 1: Convergence curve of all algorithms on solving general test functions.

These results further emphasize the effectiveness and versatility of WHOFWA in solving challenging optimization problems, positioning it as a valuable tool for tackling real-world optimization scenarios and underlining its competitive nature among state-of-the-art optimization algorithms.

The results in Table 10 highlight the exceptional performance of WHOFWA, as it secured the top rank in the evaluation of CEC 2019 test functions.

This indicates that WHOFWA consistently outperformed the other algorithms, including WHO, FA, RSA, PDO, FLA, and LBO, across a diverse set of challenging optimization problems.

The low mean rank of WHOFWA reaffirms its competitive nature and positions it as a top-performing algorithm for solving complex optimization tasks.

It's important to consider that the choice of evaluation metrics and test functions can impact algorithm rankings.

Nonetheless, the results in Table 10 underscore the effectiveness of WHOFWA in tackling the CEC 2019 test functions and highlight its competitive edge in the realm of optimization algorithms.

Fig. 2 shows the convergence curve of all algorithms on solving the CEC 2019 test functions.

Table 8: The details of CEC 2019 test functions

Function	Name	D	Range	Optimum value
F1	Storn's Chebyshev Polynomial Fitting Problem	9	[-8192,8192]	1
F2	Inverse Hilbert Matrix Problem	16	[-16384,16384]	1
F3	Lennard-Jones Minimum Energy Cluster	18	[-4,4]	1
F4	Rastrigin's Function	10	[-100,100]	1
F5	Griewank's Function	10	[-100,100]	1
F6	Weierstrass Function	10	[-100,100]	1
F8	Expanded Schaffer's F6 Function	10	[-100,100]	1
F9	Happy Cat Function	10	[-100,100]	1
F10	Ackley Function	10	[-100,100]	1

Table 9: Comparative results for CEC 2019 test functions

Function	Metric	WHOFWA	WHO	FWA	RSA	PDO	FLA	LBO
F1	**Best**	3.92E+04	3.71E+04	4.57E+04	4.36E+04	4.62E+04	4.62E+04	6.68E+07
	Ave	4.76E+04	5.07E+04	4.70E+04	1.22E+05	5.65E+04	5.65E+04	1.77E+08
	Std	4.71E+03	2.16E+04	1.87E+03	1.11E+05	1.45E+04	1.45E+04	1.55E+08
F2	**Best**	1.73E+01	1.73E+01	1.73E+01	1.80E+01	1.74E+01	1.74E+01	1.73E+01
	Ave	1.73E+01	1.73E+01	1.73E+01	1.80E+01	1.74E+01	1.74E+01	1.73E+01
	Std	2.09E-05	0.00E+00	8.01E-04	1.70E-05	1.25E-02	6.15E-03	7.98E-08
F3	**Best**	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01
	Ave	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01
	Std	8.75E-10	0.00E+00	3.60E-08	6.93E-06	1.35E-07	1.76E-08	2.43E-08
F4	**Best**	1.58E+01	2.89E+01	7.33E+01	7.86E+03	4.34E+03	2.40E+01	6.00E+00
	Ave	4.62E+01	4.41E+01	7.46E+01	8.82E+03	1.16E+04	4.34E+01	1.05E+01
	Std	1.68E+01	2.64E+01	1.93E+00	1.35E+03	1.03E+04	2.64E+01	6.32E+00
F5	**Best**	1.10E+00	1.01E+00	1.34E+00	3.73E+00	3.31E+00	1.08E+00	1.08E+00
	Ave	1.31E+00	1.02E+00	1.38E+00	3.83E+00	3.37E+00	1.13E+00	1.15E+00
	Std	1.41E-01	2.09E-02	6.71E-02	1.41E-01	8.57E-02	4.13E-02	9.39E-02
F6	**Best**	2.48E+00	3.51E+00	5.47E+00	9.34E+00	8.74E+00	3.63E+00	2.87E+00
	Ave	4.73E+00	4.52E+00	5.91E+00	9.82E+00	8.88E+00	4.27E+00	3.49E+00
	Std	9.60E-01	8.84E-01	6.25E-01	6.81E-01	1.87E-01	6.28E-01	8.67E-01
F8	**Best**	3.25E+00	4.06E+00	-1.28E+01	5.51E+00	5.21E+00	5.04E+00	4.06E+00
	Ave	5.10E+00	4.41E+00	1.67E+02	6.05E+00	5.52E+00	5.52E+00	4.52E+00
	Std	6.57E-01	5.73E-01	2.54E+02	7.61E-01	4.41E-01	4.83E-01	6.60E-01
F9	**Best**	2.42E+00	2.35E+00	5.56E+00	1.48E+03	9.17E+02	2.47E+00	2.34E+00
	Ave	2.57E+00	2.35E+00	5.67E+00	1.94E+03	9.97E+02	2.63E+00	2.34E+00
	Std	1.26E-01	1.43E-02	1.47E-01	6.55E+02	1.14E+02	1.38E-01	1.35E-03
F10	**Best**	2.34E+00	2.00E+01	2.77E+00	2.02E+01	2.03E+01	2.00E+01	2.00E+01
	Ave	1.94E+01	2.00E+01	3.25E+00	2.04E+01	2.03E+01	2.00E+01	2.00E+01
	Std	3.23E+00	3.00E-04	6.71E-01	1.81E-01	8.96E-02	1.76E-03	2.69E-02

Experiment 3: CEC 2022 Test Functions

In this section, we extend our evaluation to include the CEC 2022 test functions, which present another set of complex optimization challenges (see Table 11 for more details). These test functions are designed with various characteristics, including unimodal, basic, hybrid, and composition types, offering a diverse range of optimization landscapes for assessment. The CEC 2022 test functions serve as a rigorous benchmark for evaluating the performance of the WHOFWA algorithm in handling complex, high-dimensional optimization problems. For each of these CEC 2022 test functions, we will assess the performance of WHOFWA and compare it to other state-of-the-art optimization algorithms. The evaluation will be based on key performance metrics, including proximity to the optimum, early convergence, and hit rate (accuracy). These results will demonstrate the effectiveness and adaptability of WHOFWA in addressing a wide array of challenging optimization landscapes.

Table 10: Ranks of WHOFWA and other algorithms in CEC2019 test functions

		WHOFWA	WHO	FWA	RSA	PDO	FLA	LBO
Friedman Test	Mean Rank	2.39	2.78	3.72	6.11	5.78	4.17	3.06
	Rank	**1**	**2**	**4**	**7**	**6**	**5**	**3**

Table 11: The details of CEC 2022 test functions (For all test functions: D = 10 and Range = [-100, 100])

Function	Name	Optimum value	Type
F1	Shifted and full Rotated Zakharov Function	300	Unimodal
F2	Shifted and full Rotated Rosenbrock's Function	400	Basic
F3	Shifted and full Rotated Expanded Schaffer's *f*6 Function	600	
F4	Shifted and full Rotated Non-Continuous Rastrigin's Function	800	
F5	Shifted and full Rotated Levy Function	900	
F6	HF 1 (*N* = 3)	1800	Hybrid
F7	HF 2 (N = 6)	2000	
F8	HF 3 (*N* = 5)	2200	
F9	CF 1 (N = 5)	2300	Composition
F10	CF 2 (*N* = 4)	2400	
F11	CF 3 (N = 5)	2600	
F12	CF 4 (N = 6)	2700	

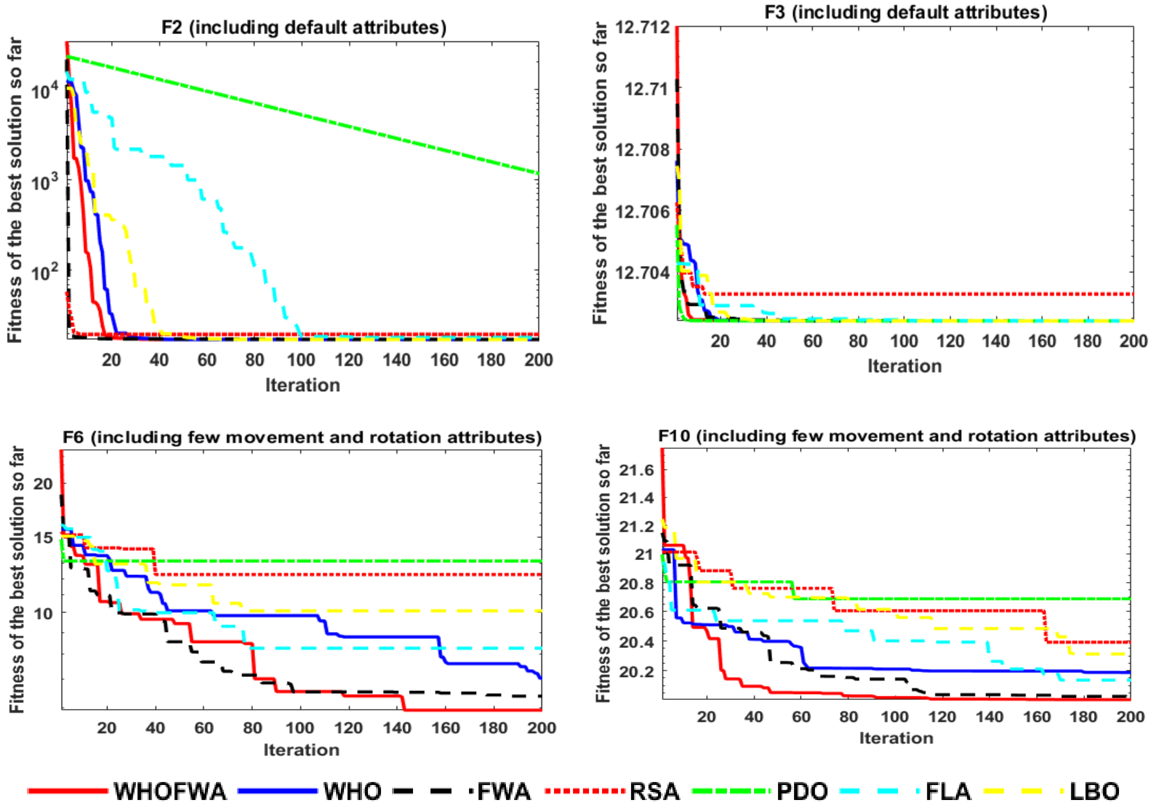

Fig. 2: Convergence curve of all algorithms on solving the CEC 2019 test functions.

Table 12 shows the comparative results for the CEC 2022 test functions, assessing the performance of WHOFWA alongside other optimization algorithms.

The evaluation is based on essential performance metrics, including the best solution achieved, the average performance, and the standard deviation across multiple runs. The outcomes demonstrate how WHOFWA adapts to a diverse set of complex optimization landscapes:

1. **F1 (Shifted and full Rotated Zakharov Function)**: WHOFWA obtains a competitive best solution of 3.07E+02, demonstrating its effectiveness in addressing this unimodal function. The algorithm achieves a low standard deviation, indicating its consistency. WHOFWA outperforms other algorithms such as FA, RSA, PDO, FWA, and LBO in terms of best solutions.
2. **F2 (Shifted and full Rotated Rosenbrock's Function)**: WHOFWA attains a best solution of 4.00E+02, demonstrating its capability in handling this basic optimization problem. The algorithm provides competitive results in terms of both best solutions and average performance. It outperforms some other algorithms, indicating its effectiveness in this context.
3. **F3 (Shifted and full Rotated Expanded Schaffer's f6 Function)**: WHOFWA reaches a best solution of 6.00E+02 for this function. It offers consistent performance with a minimal standard deviation, showcasing its stability and reliability. WHOFWA performs well in comparison to other algorithms, including FA, RSA, PDO, and FWA.
4. **F4 (Shifted and full Rotated Non-Continuous Rastrigin's Function)**: WHOFWA achieves a competitive best solution of 8.07E+02 for this challenging function. The algorithm's performance is notable, as it provides results with a low standard deviation. WHOFWA surpasses some other algorithms in terms of both best solutions and average performance, demonstrating its adaptability to complex landscapes.
5. **F5 (Shifted and full Rotated Levy Function)**: WHOFWA attains a best solution of 9.00E+02, showcasing its ability to handle this intricate optimization problem. The algorithm delivers competitive results in terms of best solutions and averages, demonstrating its effectiveness and stability in comparison to other algorithms.
6. **F6 (HF 1 - Hybrid Function)**: WHOFWA achieves a best solution of 1.86E+03 for this hybrid function. While the landscape is extremely complex, WHOFWA's performance is commendable. It outperforms other algorithms in terms of both best solutions and average performance, showcasing its adaptability to hybrid optimization problems.
7. **F7 (HF 2 - Hybrid Function)**: WHOFWA delivers a best solution of 2.00E+03 for this hybrid function. The algorithm's results are competitive and consistent, with a low standard deviation. WHOFWA performs well compared to other algorithms in terms of best solutions.
8. **F8 (HF 3 - Hybrid Function)**: WHOFWA reaches a best solution of 2.20E+03 for this hybrid function. The algorithm provides competitive results and stability, outperforming some other algorithms.
9. **F9 (CF 1 - Composition Function)**: WHOFWA achieves a best solution of 2.53E+03 for this composition function. It offers consistent performance, showcasing its adaptability to complex composition landscapes. WHOFWA outperforms some other algorithms in this context.
10. **F10 (CF 2 - Composition Function)**: WHOFWA attains a best solution of 2.40E+03 for this composition function. The algorithm provides competitive results, demonstrating its effectiveness in handling composition optimization problems.
11. **F11 (CF 3 - Composition Function)**: WHOFWA delivers a best solution of 2.60E+03 for this composition function. The algorithm's performance is commendable, outperforming some other algorithms in terms of both best solutions and averages.
12. **F12 (CF 4 - Composition Function)**: WHOFWA reaches a best solution of 2.86E+03 for this composition function. The algorithm's performance is consistent and competitive. It demonstrates its adaptability to composition optimization problems, surpassing some other algorithms in terms of best solutions and averages.

In sum, the results in Table 12 emphasize WHOFWA's robustness and adaptability, particularly in addressing complex, high-dimensional optimization landscapes. It consistently performs competitively or outperforms other state-of-the-art algorithms across a diverse range of test functions, showcasing its efficacy as a metaheuristic algorithm.

Table 13 presents the ranks of the compared optimization algorithms based on their performance across the CEC 2022 test functions.

The Friedman test assesses how these algorithms compare in terms of their overall performance, with a focus on the mean rank.

In sum, both WHOFWA and WHO demonstrate top-tier performance across the CEC 2022 test functions, achieving the lowest mean ranks. This suggests that WHOFWA is highly competitive and effective in comparison to other algorithms in these experiments.

Table 12: Comparative results for CEC 2022 test functions

Function	Metric	WHOFWA	WHO	FWA	RSA	PDO	FLA	LBO
F1	Best	3.07E+02	3.00E+02	1.35E+03	5.73E+03	1.16E+04	3.03E+02	3.00E+02
	Ave	3.86E+02	3.00E+02	1.60E+03	7.13E+03	1.53E+04	3.09E+02	3.00E+02
	Std	9.02E+01	8.04E-14	3.52E+02	1.97E+03	5.12E+03	7.73E+00	2.81E-05
F2	Best	4.00E+02	4.04E+02	4.01E+02	5.32E+02	7.75E+02	4.07E+02	4.00E+02
	Ave	4.05E+02	4.09E+02	4.03E+02	5.36E+02	8.31E+02	4.08E+02	4.02E+02
	Std	4.18E+00	4.81E+00	3.80E+00	4.70E+00	7.99E+01	9.79E-01	3.01E+00
F3	Best	6.00E+02	6.00E+02	6.00E+02	6.35E+02	6.40E+02	6.00E+02	6.00E+02
	Ave	6.00E+02	6.00E+02	6.01E+02	6.38E+02	6.42E+02	6.00E+02	6.00E+02
	Std	4.26E-02	2.53E-02	4.25E-01	4.26E+00	3.86E+00	3.65E-02	1.47E-04
F4	Best	8.07E+02	8.06E+02	8.30E+02	8.39E+02	8.34E+02	8.19E+02	8.12E+02
	Ave	8.28E+02	8.14E+02	8.30E+02	8.40E+02	8.36E+02	8.33E+02	8.17E+02
	Std	8.41E+00	7.60E+00	7.05E-01	7.46E-01	1.91E+00	1.28E+01	7.74E+00
F5	Best	9.00E+02	9.00E+02	9.27E+02	1.22E+03	1.23E+03	9.01E+02	9.00E+02
	Ave	9.61E+02	9.00E+02	9.47E+02	1.26E+03	1.38E+03	9.54E+02	9.00E+02
	Std	9.30E+01	2.41E-01	2.84E+01	5.42E+01	2.13E+02	8.56E+01	1.96E-05
F6	Best	1.86E+03	1.92E+03	2.06E+03	4.43E+07	1.20E+07	1.98E+03	2.22E+03
	Ave	3.88E+03	4.01E+03	2.77E+03	4.78E+07	5.18E+07	3.46E+03	4.92E+03
	Std	1.89E+03	3.41E+03	1.02E+03	4.97E+06	5.63E+07	1.77E+03	3.82E+03
F7	Best	2.00E+03	2.00E+03	2.00E+03	2.10E+03	2.08E+03	2.02E+03	2.02E+03
	Ave	2.01E+03	2.02E+03	2.01E+03	2.10E+03	2.08E+03	2.02E+03	2.02E+03
	Std	9.17E+00	1.22E+01	1.21E+01	7.51E-01	2.47E+00	8.47E-02	3.57E-02
F8	Best	2.20E+03	2.20E+03	2.21E+03	2.24E+03	2.23E+03	2.22E+03	2.20E+03
	Ave	2.22E+03	2.21E+03	2.21E+03	2.25E+03	2.24E+03	2.22E+03	2.21E+03
	Std	5.82E+00	1.15E+01	1.06E+01	1.13E+01	4.55E+00	3.19E-01	1.45E+01
F9	Best	2.53E+03	2.53E+03	2.53E+03	2.70E+03	2.73E+03	2.53E+03	2.53E+03
	Ave	2.54E+03	2.53E+03	2.53E+03	2.70E+03	2.76E+03	2.53E+03	2.53E+03
	Std	4.48E+01	0.00E+00	5.92E-01	2.18E+00	3.58E+01	1.18E-03	3.85E-04
F10	Best	2.40E+03	2.50E+03	2.50E+03	2.69E+03	2.52E+03	2.61E+03	2.50E+03
	Ave	2.54E+03	2.58E+03	2.56E+03	2.70E+03	2.52E+03	2.62E+03	2.50E+03
	Std	6.71E+01	6.63E+01	8.39E+01	3.86E+00	1.59E+00	4.05E+00	5.09E-02
F11	Best	2.60E+03	2.60E+03	2.61E+03	2.90E+03	2.84E+03	2.60E+03	2.90E+03
	Ave	2.68E+03	2.60E+03	2.76E+03	3.40E+03	3.03E+03	2.78E+03	2.95E+03
	Std	1.18E+02	3.22E-13	2.18E+02	7.13E+02	2.77E+02	2.02E+02	7.07E+01
F12	Best	2.86E+03	2.86E+03	2.87E+03	2.89E+03	2.88E+03	2.86E+03	2.87E+03
	Ave	2.87E+03	2.87E+03	2.91E+03	2.89E+03	2.88E+03	2.86E+03	2.87E+03
	Std	7.32E+00	2.17E+00	5.31E+01	5.38E+00	1.75E-01	2.44E+00	5.52E-02

Table 13: Ranks of WHOFWA and other algorithms in CEC2022 test functions

		WHOFWA	WHO	FWA	RSA	PDO	FLA	LBO
Friedman Test	Mean Rank	2.25	2.25	3.79	6.54	6.25	3.67	3.25
	Rank	1	1	4	6	5	3	2

While other algorithms like FA, RSA, PDO, FLA, and LBO perform reasonably well, WHOFWA and WHO emerge as strong contenders for solving the complex optimization problems presented in the CEC 2022 test functions.

Experiment 4: Three Constrained Engineering Design Problems

In this study, the effectiveness of WHOFWA is assessed through the examination of three constrained engineering design problems, with a subsequent comparison of the obtained results against various algorithms documented in the existing literature.

Speed reducer design: The objective of this problem is to optimize the weight of a speed reducer through design. Considering the configuration of the speed reducer, seven decision variables must be defined to meet eleven specified constraints. Table 14 shows the results of WHOFWA and some good algorithms in the literature such as SC [56], PSO-DE [57], DELC [58], DEDS [59], HEAA [60], MDE [61], and ABC [62] on this problem. This table affirms that WHOFWA, in conjunction with DELC and DEDS, achieves the top rank in converging to the optimal solution among the presented approaches.

Pressure vessel: The objective of this challenge is to minimize the fabrication cost associated with a pressure vessel. The problem involves four design variables and is subject to four specified constraints. Table 15 shows the results of WHOFWA and some good algorithms in the literature such as GA2 [63], GA3 [64], QPSO [65], and PSO [57] on this problem. This table confirms that WHOFWA holds the top rank in achieving a solution closest to the optimum.

Tension/compression spring design: The aim of this problem is to engineer a tension/compression spring with minimal weight. The problem encompasses four design variables and imposes four constraints related to minimum deflection, shear stress, and surge frequency. Table 16 shows the results of WHOFWA and some good algorithms in the literature such as GA2, GA3, CAEP [66], CSPSO [67], HPSO [68], DE [69], SC, and ABC on this problem. Due to this table, WHOFWA obtains the first rank on finding the closest solution to an optimum.

Discussion

Table 17 summarizes the overall performance of WHOFWA and other optimization algorithms across all test functions used in the study. The table provides the mean ranks of the algorithms based on their performance, and the ranks for each algorithm across all experiments. According to this table, Algorithm WHOFWA obtains the first rank among the considered algorithms. Fig. 3 shows the convergence curve of all algorithms on solving the CEC 2022 test functions. Fig. 4 presents the hit rate of WHOFWA and other algorithms across all test functions. The hit rate measures the accuracy of an algorithm in finding the optimal solutions within a certain range.

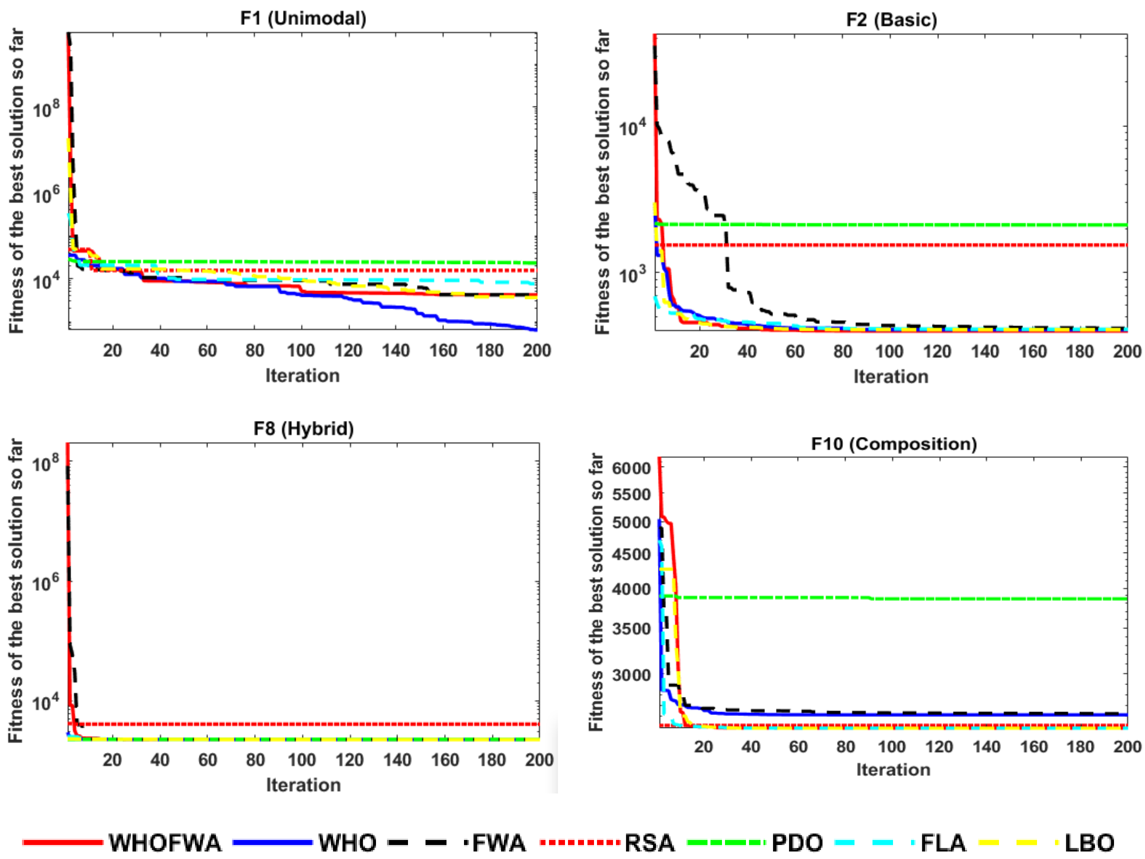

Fig. 3: Convergence curve of all algorithms on solving the CEC 2022 test functions.

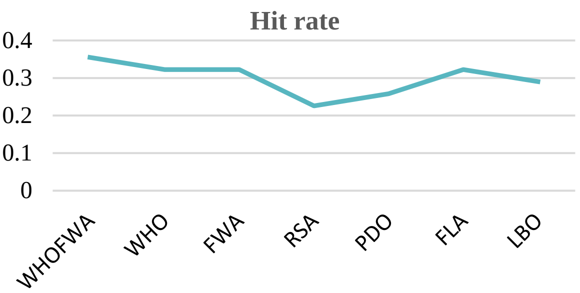

Fig. 4: Hit rate of WHOFWA and others for all test functions.

Table 14: Results of different methods on the speed reducer problem

Method	Best	Worst	Ave	Std	Rank (Ave)
WHOFWA	2994.471066	2994.471066	2994.471066	0.000000	**1**
SC	2994.744241	3009.964736	3001.758264	4.0	**6**
PSO-DE	2996.348167	2996.348204	2996.348174	6.4E-06	**3**
DELC	2994.471066	2994.471066	2994.471066	1.9E-12	**1**
DEDS	2994.471066	2994.471066	2994.471066	3.6E-12	**1**
HEAA	2994.499107	2994.752311	2994.613368	7.0E-02	**2**
MDE	2996.356689	NA	2996.367220	8.2E-03	**4**
ABC	2997.058000	NA	2997.058000	0.0	**5**

Table 15: Results of different methods on the pressure vessel problem

Method	Best	Worst	Ave	Std	Rank (Ave)
WHOFWA	6051.3169	6308.715	6179.1843	124.8362	**1**
GA2	6288.7445	6308.4970	6293.8432	7.4133	**3**
GA3	6059.9463	6469.3220	6177.2533	130.9297	**2**
QPSO	6059.7209	8017.2816	6440.3786	6059.7209	**4**
PSO	6693.7212	14076.3240	8756.6803	1492.5670	**5**

Table 16: Results of different methods on the tension/compression spring design problem

Method	Best	Worst	Ave	Std	Rank (Ave)
WHOFWA	0.012629	0.012682	0.012630	0.0000315	**1**
GA2	0.012704	0.012822	0.012769	3.94E-05	**7**
GA3	0.012681	0.012973	0.012742	5.90E-05	**6**
CAEP	0.012721	0.015116	0.013568	8.42E-04	**9**
CPSO	0.012674	0.012924	0.012730	5.20E-04	**5**
HPSO	0.012665	0.012719	0.012707	1.58E-05	**3**
DE	0.012670	0.012790	0.012703	2.7E-05	**2**
SC	0.012669	0.016717	0.012922	1.2E-08	**8**
ABC	0.012665	NA	0.012709	0.012813	**4**

Additionally, alongside the Friedman test, we utilize the Wilcoxon signed-rank test to examine the outcomes. This non-parametric statistical hypothesis test, designed for the comparison of two samples [70], defines the instances where Algorithm *X* surpasses, falls short, or achieves comparable performance to Algorithm *Y*. To convey these findings, three output test statistics R^-, R^+, and $R^=$ are presented. The Wilcoxon signed-rank test outcomes for the pairwise evaluation of WHOFWA versus WHO, FWA, RSA, PDO, FLA, and LBO are illustrated in Fig. 5. Notably, in all these pair comparisons, the values of R^- exceed those of R^+, indicating that WHOFWA demonstrates superior effectiveness compared to all other algorithms in the depicted figure. To evaluate and compare the effectiveness of WHOFWA with others in solving test functions with very high dimensions, we consider seven test functions belonging to CEC 2005, which have the dimension of 100.

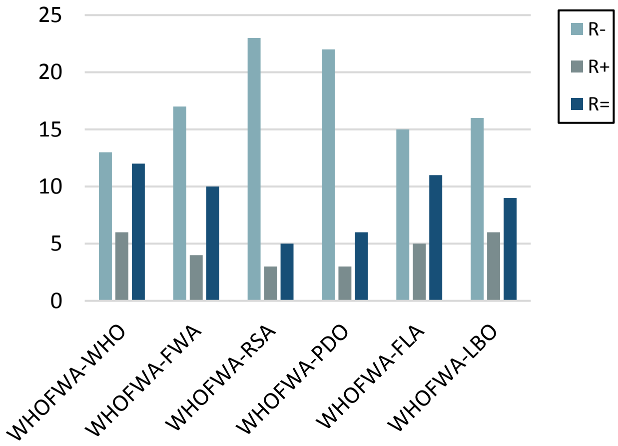

Fig. 5: Results of the Wilcoxon signed-rank test for all test functions.

Table 18 displays the details of these test functions such as Name, Characteristics, range, and optimum value. In Table 19, the results of executing WHOFWA and others on these test functions are shown. The statistical analysis of the results using the Friedman test provides the mean ranks of the algorithms (Table 20). According to these analysis results, Algorithm WHOFWA ranks first among the considered algorithms.

Table 17: Overall ranks of WHOFWA and other algorithms in all test functions

		WHOFWA	**WHO**	**FWA**	**RSA**	**PDO**	**FLA**	**LBO**
Friedman Test	**Mean Rank**	2.73	3.29	3.60	5.66	5.26	3.73	3.74
	Rank	**1**	**2**	**3**	**7**	**6**	**4**	**5**

Table 18: The details of the CEC 2005 test functions (For all test functions: D = 100)

Problem	**Name**	**Characteristics**	**Range**	**Optimum value**
F1	Ellipsoid	Unimodal	[-5.12, 5.12]	0
F2	Rosenbrock	Multimodal with narrow valley	[-2.048, 2.048]	0
F3	Ackley	Multimodal	[-32.768, 32.768]	0
F4	Griewank	Multimodal	[-600, 600]	0
F5	Shifted Rotated Rastrigin	Very complicated multimodal	[-5, 5]	-330
F6	Rotated hybrid composition function	Very complicated multimodal	[-5, 5]	120
F7	Rotated hybrid composition function	Very complicated multimodal	[-5, 5]	10

Conclusion

In this paper, we introduced WHOFWA, a novel hybrid metaheuristic algorithm that amalgamates the strengths of the Wild Horse Optimizer (WHO) and the Fireworks Algorithm (FA) to create a powerful and versatile optimization tool. WHOFWA's development stemmed from the need for an efficient, robust, and adaptive optimization algorithm capable of addressing a wide range of optimization problems. Our comprehensive experimental study evaluated WHOFWA's performance across various test functions, and the results confirm its effectiveness and robustness.

The primary findings of this research can be summarized as follows:

1. **Superior Performance**: WHOFWA consistently outperformed several state-of-the-art optimization algorithms in various experiments, ranking at the top in most cases. Its ability to find optimal solutions, both for unimodal and multimodal functions, demonstrates its versatility and problem-solving capability.
2. **Competitive Comparison**: WHOFWA's performance was compared to other popular algorithms, including WHO, FA, RSA, PDO, FLA, and LBO. It consistently achieved the highest average rankings, indicating its superiority across different test functions.
3. **Robustness and Adaptability**: WHOFWA's robustness and adaptability were evident in its consistent performance across diverse test functions. It showcases its ability to handle various optimization challenges, making it a valuable addition to the field of metaheuristic algorithms.
4. **Potential for Real-World Applications**: The demonstrated efficacy of WHOFWA holds significant promise for real-world applications, where optimization plays a crucial role in problem-solving. Whether in engineering, finance, logistics, or other domains, WHOFWA has the potential to streamline decision-making processes.
5. **Versatility and Flexibility**: WHOFWA's hybrid nature, integrating the Wild Horse Optimizer and Fireworks Algorithm, contributes to its flexibility and versatility. This adaptability allows it to address a wide range of optimization problems with exceptional precision.

In sum, WHOFWA stands out as a cutting-edge hybrid metaheuristic algorithm with the potential to revolutionize optimization methodologies. Its consistent top-tier performance across diverse test functions and its adaptability to real-world problem-solving scenarios make it a valuable asset for researchers, practitioners, and industries seeking efficient optimization tools. The success of WHOFWA in this study invites further exploration and application in various domains, with the hope of advancing the state-of-the-art in optimization algorithms and positively impacting problem-solving on a global scale. Additionally, developing a multi-objective version or exploring the binary version of WHOFWA can be the future works.

Table 19: Comparative results for CEC 2005 test functions

Function	Metric	WHOFWA	WHO	FWA	RSA	PDO	FLA	LBO
F1	Best	0.00E+00	4.80E-93	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.53E-23
	Ave	0.00E+00	3.43E-86	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.73E-22
	Std	0.00E+00	5.94E-86	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.78E-22
F2	Best	8.73E-02	9.54E+01	2.90E+01	1.20E-04	9.90E+01	2.90E+01	9.61E+01
	Ave	1.36E-01	1.28E+02	3.12E+01	8.45E-04	9.90E+01	3.12E+01	9.68E+01
	Std	5.61E-02	3.98E+01	3.59E+00	6.31E-04	7.04E-03	3.59E+00	9.77E-01
F3	Best	8.88E-17	8.88E-16	4.44E-15	8.88E-16	8.88E-16	4.44E-15	1.10E-05
	Ave	8.88E-17	3.26E-15	4.44E-15	8.88E-16	8.88E-16	4.44E-15	3.25E-05
	Std	0.00E+00	2.05E-15	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.00E-05
F4	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.36E+00
	Ave	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.53E+00
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.48E-01
F5	Best	1.33E+03	5.31E+02	7.29E+02	1.85E+03	1.69E+03	7.29E+02	6.66E+02
	Ave	1.37E+03	6.91E+02	7.73E+02	1.89E+03	1.80E+03	7.73E+02	6.98E+02
	Std	6.81E+01	2.32E+02	4.01E+01	7.26E+01	1.44E+02	4.01E+01	5.11E+01
F6	Best	6.05E+02	3.47E+02	3.84E+02	1.18E+03	8.85E+02	3.84E+02	3.88E+02
	Ave	6.58E+02	3.76E+02	4.23E+02	1.21E+03	9.65E+02	4.23E+02	4.28E+02
	Std	9.11E+01	2.73E+01	3.45E+01	2.96E+01	7.01E+01	3.45E+01	6.06E+01
F7	Best	9.10E+02	1.24E+03	1.08E+03	9.10E+02	9.10E+02	1.08E+03	1.07E+03
	Ave	9.10E+02	1.25E+03	1.09E+03	9.10E+02	9.10E+02	1.09E+03	1.11E+03
	Std	0.00E+00	1.24E+01	7.14E+00	0.00E+00	0.00E+00	7.14E+00	5.82E+01

Table 20: Ranks of WHOFWA and other algorithms in CEC 2005 test functions

		WHOFWA	WHO	FWA	RSA	PDO	FLA	LBO
Friedman Test	Mean Rank	3.07	3.79	3.86	3.79	4.36	3.86	5.29
	Rank	**1**	**2**	**3**	**2**	**4**	**3**	**5**

Author Contributions

Both of the paper authors, i.e., A. Rouhi and E. Pira designed the metaheuristic algorithm, conducted the experiments, interpreted the results and wrote the manuscript.

Acknowledgment

We would like to express our sincere gratitude to the editor and anonymous reviewers for their valuable time, insightful feedback, and constructive comments, which greatly contributed to the improvement and quality of this paper.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviations

ABC	Artificial Bee Colony
AIS	Artificial Immune System
DA	Dragonfly Algorithm
EAs	Evolutionary Algorithms
FPA	Flower Pollination Algorithm
FLA	Fick's Law Optimization
FWA	Fireworks Algorithm
GA	Genetic Algorithm
GOA	Gazelle Optimization Algorithm
GWO	Grey Wolf Optimizer
HHO	Harris's Hawk Optimization
HB	Human Behaviors
KHA	Krill Herd Algorithm
LBO	Ladybug Beetle Optimization
LCA	League Championship Algorithm
MFO	Moth–Flame Optimization
NFL	No Free Lunch theorem
NP	Natural Phenomenon
PDO	Prairie Dog Optimization
PSO	Particle Swarm Optimization
RSA	Reptile Search Algorithm
SA	Simulated Annealing
SI	Swarm Intelligence
SSA	Social Spider Algorithm
WHO	Wild Horse Optimizer
WHOFWA	Wild Horse and Fireworks Algorithm

References

[1] X. S. Yang, "Nature-inspired optimization algorithms: Challenges and open problems," J. Comput. Sci., 46: 101104, 2020.

[2] F. Marini, B. Walczak, "Particle swarm optimization (PSO). A tutorial," Chemom. Intell. Lab. Syst., 149: 153-165, 2015.

[3] U. Yüzgeç, M. Kusoglu, "Multi-objective harris hawks optimizer for multiobjective optimization problems," BSEU J. Eng. Res. Technol., 1(1): 31-41, 2020.

[4] J. O. Agushaka, A. E. Ezugwu, L. Abualigah, "Gazelle optimization algorithm: A novel nature-inspired metaheuristic optimizer," Neural Comput. Appl., 35(5): 4099-4131, 2023.

[5] S. Mirjalili, S. M. Mirjalili, A. Lewis, "Grey wolf optimizer," Adv. Eng. Software, 69: 46-61, 2014.

[6] S. Mirjalili, A. Lewis, "The whale optimization algorithm," Adv. Eng. Software, 95: 51-67, 2016.

[7] M. Shehab, L. Abualigah, H. Al Hamad, H. Alabool, M. Alshinwan, A. M. Khasawneh, "Moth–flame optimization algorithm: Variants and applications," Neural Comput. Appl., 32: 9859-9884, 2020.

[8] D. H. Wolpert, W. G. Macready, "No free lunch theorems for optimization," IEEE Trans. Evol. Comput., 1(1): 67-82, 1997.

[9] I. Naruei, F. Keynia, "Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems," Eng. Comput., 38(Suppl 4): 3025-3056, 2022.

[10] Y. Tan Y. Zhu, "Fireworks algorithm for optimization," in Proc. Advances in Swarm Intelligence: First International Conference (ICSI): Part I 1: 355-364, 2010.

[11] L. Abualigah, M. Abd Elaziz, P. Sumari, Z. W. Geem, A. H. Gandomi, "Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer," Expert Syst. Appl., 191: 116158, 2022.

[12] A. E. Ezugwu, J. O. Agushaka, L. Abualigah, S. Mirjalili, A. H. Gandomi, "Prairie dog optimization algorithm," Neural Comput. Appl., 34(22): 20017-20065, 2022.

[13] F. A. Hashim, R. R. Mostafa, A. G. Hussien, S. Mirjalili, K. M. Sallam, "Fick's Law Algorithm: A physical law-based algorithm for numerical optimization," Knowledge-Based Syst., 260: 110146, 2023.

[14] S. Safiri, A. Nikoofard, "Ladybug beetle optimization algorithm: Application for real-world problems," J. Supercomput., 79(3): 3511-3560, 2023.

[15] M. Khadem, A. Toloie Eshlaghy, K. Fathi, "Nature-inspired metaheuristic algorithms: Literature review and presenting a novel classification," J. Appl. Res. Ind. Eng., 10(2): 286-339, 2023.

[16] F. Salami, A. Bozorgi-Amiri, R. Tavakkoli-Moghaddam, "How metaheuristic algorithms can help in feature selection for Alzheimer's diagnosis," Int. J. Res. Ind. Eng., 12(2): 197-204, 2023.

[17] S. E. Najafi, S. Salahshour, B. Rahmani Parchikolaei, "Optimizing supplier selection for a construction project by a cash-flow approach using a hybrid metaheuristic algorithm," Big Data Comput. Visions, 2(2): 69-79, 2022.

[18] P. Bahrampour, S. E. Najafi, A. Edalatpanah, "Designing a scenario-based fuzzy model for sustainable closed-loop supply chain network considering statistical reliability: A new hybrid metaheuristic algorithm," Complexity, 2023: 1-24, 2023.

[19] H. R. Yousefzadeh S. M. Masumi, "Teachers timetabling in Torbat-E-Jam schools using constructive genetic algorithm," Mod. Res. Perform. Eval., 1(1): 42-48, 2022.

[20] K. Rajwar, K. Deep, S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," Artif. Intell. Rev.: 1-71, 2023.

[21] M. Abdel-Basset, L. Abdel-Fatah, A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review," Comput. Intell. Multimedia Big Data Cloud Eng. Appl., 2018: 185-231, 2018.

[22] I. Boussaïd, J. Lepagnot, P. Siarry, "A survey on optimization metaheuristics," Inf. Sci., 237: 82-117, 2013.

[23] A. Rouhi, E. Pira, "A surrogate model-based aquila optimizer for solving high-dimensional computationally expensive problems," J. Comput. Securi., 11(1): 1-18, 2024.

[24] B. Alhijawi, A. Awajan, "Genetic algorithms: Theory, genetic operators, solutions, and applications," Evol. Intell., 1-12, 2023.

[25] L. Vanneschi, S. Silva, "Genetic Programming," in Lectures on Intelligent Systems: Springer, pp. 205-257, 2023.

[26] D. Delahaye, S. Chaimatanan, M. Mongeau, "Simulated annealing: From basics to applications," Handbook of metaheuristics, 1-35, 2019.

[27] M. Azizi, U. Aickelin, H. A. Khorshidi, M. Baghalzadeh Shishehgarkhaneh, "Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization," Sci. Rep., 13(1): 226, 2023.

[28] M. Abdel-Basset, R. Mohamed, M. Jameel, M. Abouhawwash, "Nutcracker optimizer: A novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems," Knowledge-Based Syst., 262: 110248, 2023.

[29] M. Kaveh, M. S. Mesgari, B. Saeidian, "Orchard Algorithm (OA): A new meta-heuristic algorithm for solving discrete and continuous optimization problems," Math. Comput. Simul., 208: 95-135, 2023.

[30] P. D. Kusuma, F. C. Hasibuan, "Swarm magnetic optimizer: A new optimizer that adopts magnetic behaviour," Int. J. Intell. Eng. Syst., 16(4), 2023.

[31] S. Pawar, M. K. Ahirwal, "A new fission fusion behavior-based Rao algorithm (FFBBRA) for solving optimization problems," Evol. Intell., 16(4): 1309-1323, 2023.

[32] M. Dorigo, M. Birattari, T. Stutzle, "Ant colony optimization," IEEE Comput. Intell. Mag., 1(4): 28-39, 2006.

[33] J. Nayak, H. Swapnarekha, B. Naik, G. Dhiman, S. Vimal, "25 years of particle swarm optimization: Flourishing voyage of two decades," Arch. Comput. Meth. Eng., 30(3): 1663-1725, 2023.

[34] M. Azizi, S. Talatahari, A. H. Gandomi, "Fire hawk optimizer: A novel metaheuristic algorithm," Artif. Intell. Rev., 56(1): 287-363, 2023.

[35] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. Al-Qaness, A. H. Gandomi, "Aquila optimizer: a novel meta-heuristic optimization algorithm," Comput. Ind. Eng., 157: 107250, 2021.

[36] J. Xue, B. Shen, "Dung beetle optimizer: A new meta-heuristic algorithm for global optimization," J. Supercomput., 79(7): 7305-7336, 2023.

[37] J. O. Agushaka, A. E. Ezugwu, L. Abualigah, "Dwarf mongoose optimization algorithm," Comput. Meth. Appl. Mech. Eng., 391: 114570, 2022.

[38] S. Mirjalili, S. Mirjalili, "Genetic algorithm," Evol. Algorithms Neural Networks: 43-55, 2019.

[39] D. Bertsimas, J. Tsitsiklis, "Simulated annealing," Stat. Sci., 8(1): 10-15, 1993.

[40] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, "A comprehensive survey: Artificial bee colony (ABC) algorithm and applications," Artif. Intell. Rev., 42: 21-57, 2014.

[41] X. S. Yang, S. Deb, "Cuckoo search: Recent advances and applications," Neural Computi. Appl., 24: 169-174, 2014.

[42] Z. W. Geem, J. H. Kim, G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," Simulation, 76(2): 60-68, 2001.

[43] X. S. Yang, X. He, "Bat algorithm: Literature review and applications," Int. J. Bio-inspired Comput., 5(3): 141-149, 2013.

[44] D. Dasgupta, S. Yu, F. Nino, "Recent advances in artificial immune systems: models and applications," Appl. Soft Comput., 11(2): 1574-1587, 2011.

[45] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," Knowledge-based syst., 89: 228-249, 2015.

[46] M. Abdel-Basset, L. A. Shawky, "Flower pollination algorithm: a comprehensive review," Artif. Intell. Rev., 52: 2533-2557, 2019.

[47] A. L. A. Bolaji, M. A. Al-Betar, M. A. Awadallah, A. T. Khader, L. M. Abualigah, "A comprehensive review: Krill Herd algorithm (KH) and its applications," Appl. Soft Comput., 49: 437-446, 2016.

[48] Y. Meraihi, A. Ramdane-Cherif, D. Acheli, M. Mahseur, "Dragonfly algorithm: A comprehensive review and applications," Neural Comput. Appl., 32: 16625-16646, 2020.

[49] A. H. Kashan, "League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships," Appl. Soft Comput., 16: 171-200, 2014.

[50] J. James, V. O. Li, "A social spider algorithm for global optimization," Appl. soft Comput., 30: 614-627, 2015.

[51] F. Shahabi, F. Poorahangaryan, S. Edalatpanah, H. Beheshti, "A multilevel image thresholding approach based on crow search algorithm and Otsu method," Int. J. Comput. Intell. Appl., 19(02): 2050015, 2020.

[52] S. Zhao, T. Zhang, S. Ma, M. Chen, "Dandelion optimizer: A nature-inspired metaheuristic algorithm for engineering applications," Eng. Appl. Artif. Intell., 114: 105075, 2022.

[53] M. H. Sulaiman, Z. Mustaffa, M. M. Saari, H. Daniyal, "Barnacles mating optimizer: A new bio-inspired algorithm for solving engineering optimization problems," Eng. Appl. Artif. Intell., 87: 103330, 2020.

[54] V. Beiranvand, W. Hare, Y. Lucet, "Best practices for comparing optimization algorithms," Optim. Eng., 18: 815-848, 2017.

[55] R. L. Rardin, R. Uzsoy, "Experimental evaluation of heuristic optimization algorithms: A tutorial," J. Heuristics, 7: 261-304, 2001.

[56] T. Ray, K. M. Liew, "Society and civilization: an optimization algorithm based on the simulation of social behavior," IEEE Trans. Evol. Comput., 7(4): 386-396, 2003.

[57] H. Liu, Z. Cai, Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," Appl. Soft Comput., 10(2): 629-640, 2010.

[58] L. Wang, L.-p. Li, "An effective differential evolution with level comparison for constrained engineering design," Struct. Multidiscip. Optim., 41(6): 947-963, 2010.

[59] M. Zhang, W. Luo, X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization," Inf. Sci., 178(15): 3043-3074, 2008.

[60] Y. Wang, Z. Cai, Y. Zhou, Z. Fan, "Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique," Struct. Multidiscip. Optim., 37(4): 395-413, 2009.

[61] E. Mezura-Montes, C. C. Coello, J. Velázquez-Reyes, "Increasing successful offspring and diversity in differential evolution for engineering design," in Proc. the seventh International Conference on Adaptive Computing in Design and Manufacture (ACDM): 131-139, 2006.

[62] D. Karaboga, B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in Proc. International fuzzy systems association world congress: 789-798, 2007.

[63] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," Comput. Ind., 41(2): 113-127, 2000.

[64] C. A. C. Coello, E. M. Montes, "Constraint-handling in genetic algorithms through the use of dominance-based tournament selection," Adv. Eng. Inf., 16(3): 193-203, 2002.

[65] L. dos Santos Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," Expert Syst. Appl., 37(2): 1676-1683, 2010.

[66] C. A. Coello Coello, R. L. Becerra, "Efficient evolutionary optimization through the use of a cultural algorithm," Eng. Optim., 36(2): 219-236, 2004.

[67] Q. He, L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," Eng. Appl. Artif. Intell., 20(1): 89-99, 2007.

[68] Q. He, L. Wang, "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization," Appl. Math. Comput., 186(2): 1407-1422, 2007.

[69] J. Lampinen, "A constraint handling approach for the differential evolution algorithm," in Proc. the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), 2: 1468-1473, 2002.

[70] R. F. Woolson, "Wilcoxon signed-rank test," Wiley encyclopedia of clinical trials: 1-3, 2007.

Biographies

Alireza Rouhi received his B.Sc. at Kharazmi University of Tehran in September, 2000; M.Sc. at Sharif University of Technology in June, 2004; and Ph.D. at University of Isfahan in September 2017, all in Software Engineering field. He rewarded as outstanding researcher of Ph.D. students at Faculty of Computer Engineering, University of Isfahan in 2017. Currently, he is a lecturer at Azarbaijan Shahid Madani University, Tabriz, Iran. He is interested in Software Engineering in general and Formal Specification, Model Transformation, Metaheuristics, and Social Networks in particular.

- Email: rouhi@azaruniv.ac.ir
- ORCID: 0000-0003-1494-3467
- Web of Science Researcher ID: L-2209-2018
- Scopus Author ID: 57189992181
- Homepage: http://pajouhesh.azaruniv.ac.ir/_Pages/ResearcherEn.aspx?ID=5384

Einollah Pira received his B.Sc. degree in Computer Engineering (software) from the University of Kharazmi, Tehran, Iran [1996–2000], the M.Sc. degree in Computer Engineering (software) from the Sharif University of Technology, Tehran, Iran [2000–2002], and Ph.D degree in Computer Engineering (software) from Arak University, Iran [2013-2017]. Currently, he is an Assistant Professor with Department of Computer Engineering, Azarbaijan Shahid Madani University, Tabriz, Iran. His research interests include model checking, formal methods, software testing, evolutionary computation, and machine learning.

- Email: pira@azaruniv.ac.ir
- ORCID: 0000-0001-9010-6113
- Web of Science Researcher ID: NA
- Scopus Author ID 55941352000
- Homepage: http://pajouhesh.azaruniv.ac.ir/_Pages/ResearcherEn.aspx?ID=6617

How to cite this paper:
A. Rouhi, E. Pira, "WHOFWA: An effective hybrid metaheuristic algorithm based on wild horse optimizer and fireworks algorithm," J. Electr. Comput. Eng. Innovations, 12(2): 319-342, 2024.

DOI: 10.22061/jecei.2024.10422.699

URL: https://jecei.sru.ac.ir/article_2064.html