International Journal of Computational Intelligence Systems (2024) 17:111
https://doi.org/10.1007/s44196-024-00497-6

RESEARCH ARTICLE

Using Improved Hybrid Grey Wolf Algorithm Based on Artificial Bee Colony Algorithm Onlooker and Scout Bee Operators for Solving Optimization Problems

Ishaq Ahmad[1] · Fawad Qayum[1] · Sami Ur Rahman[1] · Gautam Srivastava[2,3]

Received: 27 November 2023 / Accepted: 3 April 2024

Abstract

Grey Wolf optimization (GWO) is a newly developed stochastic meta-heuristic technique motivated by nature. It shows potential in diverse optimization challenges. It replicates grey wolf hunting behaviour and social hierarchy, exploring the solution space similar to their natural process. The algorithm efficiently explores and converges to the optimal solution. However, a drawback of the standard GWO is its limited exploitation capability due to its exploration-focused iterations. This may hinder finding the optimal solution nearby, leading to lower local convergence rates and degraded solution quality. To address this, the GWO-Employed-Onlooker model suggests incorporating the onlooker and scout bee operators from the artificial bee colony algorithm (ABC) during the position-changing stage of the grey wolves. This enhances exploitation capability, resulting in improved local convergence rates and better solution quality. The proposed method's performance is evaluated on various optimization functions and compared their convergence rate to standard GWO, Genetic Algorithm (GA), Firefly Algorithm (FA), ABC, and Ant Colony Optimization (ACO) techniques. The results demonstrate that the proposed strategy GWO-Employed-Onlooker is better, indicating that it is valuable in solving optimization problems.

Keywords Grey Wolf · Stochastic · Heuristics · Ant colony · Optimization

1 Introduction

Swarm intelligence (SI) is a modern Artificial intelligence (AI) technology that has captured the interest of researchers during the past several years. All of the methodologies covered by SI are used to solve various problems of optimization [1]. The collective activity of various forms of swarming, such as honey-bees termites, ant colonies, groups of fish, and groups of birds is the main source of motivation for SI approaches. However, while trying to accomplish different kinds of challenging objectives while performing different complicated tasks, such swarms create a decentralized system that encourages strong coordination and intercommunication. Each of the mentioned swarms establish a very strong communication system to assist the entire system toward accomplishing many different kinds of complex goals, such as setting nets, looking for food, travelling from one place to a different one, and residing safely in hazardous environments. To achieve these objectives, an effectively managed and mutually intelligible synchronized structure is managed. To maintain a strong coordination system, swarms use two phenomena: exploration and exploitation. Exploration is the process of collecting new information, while exploitation is the process of using existing information to improve coordination. Exploration and exploitation are two crucial concepts utilized in SI techniques for solving optimization problems. Exploration involves expanding the search space to discover new solutions, while exploita-

✉ Gautam Srivastava
srivastavag@brandonu.ca

Ishaq Ahmad
ishaqictm@gmail.com

Fawad Qayum
fawadqayum@uom.edu.pk

Sami Ur Rahman
srahman@uom.edu.pk

1 Department of Computer Science and Information Technology, The University of Malakand, Chakdara Dir (L), Khyber Pakhtunkhwa 18800, Pakistan

2 Department of Math and Computer Science, Brandon University, Brandon, MB, Canada

3 Research Centre for Interneural Computing, China Medical University, 40402 Taichung, Taiwan

tion focuses on refining the best solutions found so far to avoid missing optimal solutions nearby. These processes are employed in optimization to target global and local solutions, respectively. Exploration broadens the search space by varying optimization function values, while exploitation concentrates on known solutions and neighbouring candidates to avoid overlooking the best local solution. Striking a balance between exploration and exploitation is essential for achieving optimal solutions in problems related to optimization [2]. In [3], the authors first proposed the concept of self-organized and decentralized swarm intelligence for optimizing cellular robotic systems, and it has since found applications in domains, such as load balancing, mobile network routing, and problem-solving. SI is a field within artificial intelligence that studies the collective behaviour of decentralized systems. SI algorithms draw inspiration from natural swarms, such as bird flocks, fish schools, and ant colonies. These algorithms have demonstrated effectiveness in solving a variety of optimization problems. Well-known SI techniques include: Ant colony optimization (ACO) [4], Krill herd algorithm [5], Particle swarm optimization (PSO) [6], Firefly algorithm (FA) [7], Artificial bee colony (ABC) [8], Bat algorithm [9], and Grey Wolf algorithm (GWO) [10]. SI algorithms can also be used in conjunction with various other artificial intelligence (AI) techniques to target particular functions. PSO, for example, was used to optimize parameters for support vector machines in goods volume prediction [11]. Additionally, GWO has been paired with Particle Swarm Optimization (PSO) [12], Differential Evolution (DE) [13], Genetic Algorithm (GA) [14], Ant Lion Optimizer (ALO) [15], and Simulated Annealing (SA) [16] for enhanced performance.

1.1 Grey Wolf Optimization Algorithm

Grey Wolf Optimizer (GWO) is a meta-heuristic method motivated by grey wolf social behaviour. It was introduced by Mirjalili et al. in 2014 and has been successfully applied to various optimization challenges. The algorithm simulates a wolf pack hunting for prey, with distinct roles assigned to the wolves: Alpha, Beta, Delta, and Omega. The first, second, and third best solutions are represented by the Alpha, Beta, and Delta wolves, respectively, during the optimization process [17].

GWO consists of three stages: initialization, exploration, and exploitation. During the exploration phase, the locations of the wolves are updated using an equation derived from the hunting behaviour of wolf packs. This equation incorporates individual, social, and prey movements, enabling the exploration of the search space and the identification of potential solutions. In the exploitation stage, Alpha, Beta, and Delta wolves further refine the optimal solution found so far using a greedy search strategy.

GWO is a simple and efficient technique that shows promising results in both continuous and discrete optimization problems [18]. It leverages the hunting behaviour of grey wolves, which operate in groups and employ a searching, encircling, and attacking strategy. The algorithm assigns specific roles to wolves, with the Alpha wolf representing the best, the Beta wolf as the second best, and the Omega wolves representing the remaining members of the pack [19]. The mathematical modelling of this algorithm can be represented by Eq. 1 [20]

$$X(t+1) = X(t) - A.D. \tag{1}$$

The equation for determining the subsequent location of a wolf, denoted as $X(t+1)$, is based on its present location, $X(t)$, a coefficient matrix represented by A, and a vector D that depends on the prey's location (X_p). The mathematical expression for D is given by Eq. 2 [21]

$$D = |C.X_p(t) - X(t)|, \tag{2}$$

where $C = 2.r_2$. r_2 is a vector produced at random from the interval [0, 1]. The preceding two equations simulate the grey wolves' movement speeds and step sizes

$$A = 2a.r_1 - a. \tag{3}$$

In Eq. 3, a is a vector that linearly decreases from 2 to 0 for every iteration. Furthermore, r_1 is a randomly generated vector within the range [0, 1]. To update the parameter, you can utilize the following equation:

$$a = 2 - t\left(\frac{2}{T}\right). \tag{4}$$

In Eq. 4, t shows the most recent iteration, while T denotes the total number of iterations. Equations 1,2,3, and 4 can only be utilized for relocating the wolf to any position in a hypersphere around the prey. When the global optimum is unknown in a problem of optimization, then alpha, beta, and delta are considered to have a better knowledge of their positions. Equation 5 requires other wolves to update their locations [22]

$$X(t+1) = \frac{1}{3}X_1 + \frac{1}{3}X_2 + \frac{1}{3}X_3, \tag{5}$$

where X_1, X_2, and X_3 are calculated in Eq. 6 as

$$\begin{aligned} X_1 &= X_\alpha(t) - A_1.D_\alpha \\ X_2 &= X_\beta(t) - A_2.D_\beta \\ X_3 &= X_\gamma(t) - A_3.D_\gamma; \end{aligned} \tag{6}$$

D_α, D_β, D_γ are calculated using Eq. 7

$$\begin{aligned} D_\alpha &= |C_1.X_\alpha - X| \\ D_\beta &= |C_2.X_\beta - X| \\ D_\gamma &= |C_3.X_\gamma - X|. \end{aligned} \tag{7}$$

2 Literature Review

GWO has been effectively used for a variety of optimization problems since its launch in 2014. Researchers have explored hybridization strategies to further improve the speed and effectiveness of GWO [23]. One popular hybrid approach is combining GWO with Particle Swarm Optimization (PSO). Kumar et al. [24] proposed a hybrid algorithm called GPSO–GWO, which utilizes PSO to generate the initial population and employs GWO to enhance the PSO solutions. Experimental results demonstrate that GPSO–GWO outperforms both PSO and GWO in terms of solution quality and convergence speed [24]. Another hybridization method involves combining GWO with Genetic Algorithm (GA). Li et al. [25] introduced a hybrid algorithm named GGA-GWO, where GA generates the initial population and GWO refines the solutions obtained by GA. Experimental findings indicate that GGA-GWO outperforms both GA and GWO in terms of solution quality and resilience [25]. Furthermore, Zhang et al. [26] proposed a hybrid method called GWO–PSO–SVM, which combines GWO, PSO, and a Support Vector Machine (SVM) for breast cancer diagnosis. GWO and PSO are used to create the initial population of SVM models, and an optimal SVM model is implemented as the final solution. Experimental results demonstrate that in terms of classification accuracy and sensitivity, GWO–PSO–SVM beats numerous state-of-the-art approaches [26]. Similarly, Wang et al. [27] developed a hybrid technique named GWO–DE by combining GWO with Differential Evolution (DE) to solve the image thresholding problem. GWO generates the initial population, and DE refines the solutions produced by GWO. The experimental results suggest that GWO–DE produces higher solution quality and faster convergence than GWO and DE alone [27]. Other hybrid approaches include combining GWO with Artificial Bee Colony (ABC) in GWO–ABC proposed by Sahu and Pati [28] for multi-objective optimization. The hybrid algorithm utilizes GWO and ABC to generate the initial population and select the best solutions using a non-dominated sorting technique. Experimental results demonstrate that GWO–ABC outperforms various state-of-the-art algorithms in terms of solution quality and diversity [28]. Additionally, Tabrizi et al. [29] combined GWO with the artificial bee colony (ABC) technique to tackle the economic emission dispatch (EED) problem. This hybrid approach outperformed other optimization algorithms in less computational time and best solution quality [29]. Mozafari et al. [30] introduced a new hybrid approach that combines the Grey Wolf Optimizer (GWO) with the Genetic Algorithm (GA) to solve constrained optimization problems. The suggested approach performance and quality are examined in comparison to other algorithms using three benchmark functions. The outcomes demonstrate that the hybrid approach performs better than other methods in convergence rate and solution quality [30]. Karthikeyan et al. [31] published a study where they combined GWO with the Particle Swarm Optimization (PSO) algorithm to address the feature selection problem in intrusion detection systems. The hybrid algorithm was tested on two datasets in comparison with other feature selection techniques. The outcomes demonstrated that the hybrid method achieved better classification accuracy and feature subset size compared to other techniques [31]. Dhillon et al. [32] researched parameter identification of photovoltaic models and proposed a hybrid algorithm that combines GWO with the Cuckoo Search (CS) algorithm. The hybrid method was evaluated and compared to other optimization techniques using a real-world photovoltaic system. The outcome demonstrated that the hybrid approach outperformed other methods in accuracy and resilience [32]. Subramanyam et al. [33] developed a hybrid technique that combines GWO with the Teaching Learning-Based Optimization (TLBO) approach to tackle the optimal power flow (OPF) problem. The hybrid algorithm was tested and compared to other optimization algorithms using the IEEE 30-bus and 118-bus test systems. The results demonstrated that the hybrid approach is efficient in computational time and solution quality as compared with other methods [33]. Wang et al. [7] developed a hybrid GWO algorithm with Differential Evolution (DE) to overcome the parameter identification problem of photovoltaic cell models. When the proposed hybrid algorithm's performance was compared to regular GWO and DE algorithms, it outperformed them [34]. Kumar and Singh [34] proposed a hybrid GWO algorithm with the Shuffled Frog Leaping Algorithm (SFLA) to address the feature selection problem in intrusion detection systems. The hybrid algorithm outperforms the traditional GWO and SFLA algorithms in terms of performance [35]. Sharma and Singh [35] presented a hybrid GWO technique with Particle Swarm Optimization (PSO) to handle the simultaneous placement and routing problem of analog circuits. The proposed hybrid algorithm performed better when compared to standard GWO and PSO algorithms [36]. Sahu and Swain [28] proposed a hybrid GWO–ABC algorithm to tune the Proportional-Integral-Derivative (PID) controller for an automated voltage regulator (AVR) system. The efficiency of the hybrid approach was compared to standard GWO and Artificial Bee Colony (ABC) algorithms, and it achieved better results [37]. Pakhira and Das [37] developed a hybrid GWO method with Differential Evolution (DE) to address the feature selection problem in microarray data classification. The proposed hybrid algorithm outperformed standard GWO and DE algorithms [38]. In conclusion, combining GWO with different optimization techniques has shown considerable promise in terms of enhancing GWO performance. The hybrid algorithms described in these studies outperformed the original GWO algorithm and other modern algorithms.

3 Inspiration

Meta-heuristic algorithms utilize two key concepts to evaluate the solution search space: exploration and exploitation. Exploration aims to diversify problem solutions, expanding the search field to include alternatives with greater variations. Exploitation on the other side works to narrow down the solutions by minimizing the differences between distinct solutions. Exploration involves globalizing the solution search space, while exploitation involves localizing it [39]. Balancing these two properties is crucial for optimization algorithms to achieve higher solution quality. If the solution domain is more explorative but lightly exploitative, a considerable divergence between the previous iteration's solutions and the current iteration's solutions is found which results in missing the best solution. Ideally, the solutions should have a smaller difference between them to ensure the best outcome. The success of each meta-heuristic algorithm in finding a solution depends on effectively balancing exploration and exploitation [40]. Similarly, the Gray Wolf Optimizer (GWO) operates as a meta-heuristic algorithm with compromises between exploration (identifying new search areas) and exploitation (locating results within the neighbourhood search region). However, this compromise can result in a slow convergence rate and low-quality solutions. To address this issue and tackle various optimization problems, researchers have proposed different variants of GWO and hybridizations with other optimization algorithms. These approaches aim to strike a balance between exploration and exploitation. Despite these efforts, there is still ample room for further improvement [41].

4 Proposed Techniques

4.1 Problem Formulation

GWO is a meta-heuristic technique that takes inspiration from the social structure and hunting strategies observed in grey wolves. The GWO algorithm consists of three main stages, i.e., initialization, wolf's position changing, and termination. In the initialization stage, four wolves are initialized randomly within the solution search space. The wolves

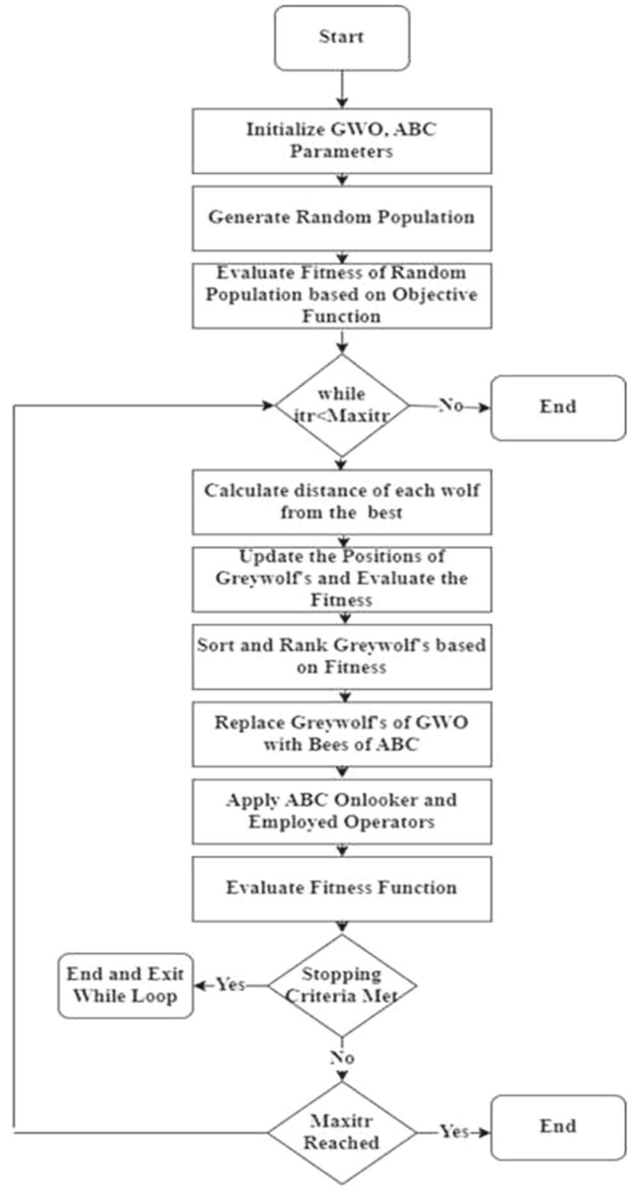


Fig. 1 Proposed TECHNIQUE GWO-Employed-Onlooker

are ranked according to their fitness values, with the alpha wolf striving for the utmost level of fitness and the omega wolf aiming for the minimal fitness value. In Wolf's position-changing stage, each wolf moves towards the location of the alpha wolf. The amount of movement is determined by a randomization factor. The algorithm terminates when either a pre-specified limit of execution is achieved or the fitness of the alpha wolf has remained unchanged for a specific number of iterations. When the wolf's position changes, the randomization element is significant. It controls the amount of randomness in the movement of the wolves. A high value of the randomization factor will cause the wolves to move more randomly, while a low value of the randomization factor will cause the wolves to move more deterministically.

The value of the randomization factor must be balanced to achieve good performance. If the randomization factor is too high, the wolves will move too randomly and will be less likely to produce the best solution. If the randomization factor is too low, the wolves will move too deterministically and will be more likely to get stuck in local optima. The optimal value of the randomization factor will depend on the problem being solved. However, in general, it is important to find a value that balances exploration and exploitation in a search to identify the best answer. Figure 1 represents the data flow diagram of the proposed method GWO-Employed-Onlooker.

of each respective solution. The employed and onlooker bees represent different candidate solutions being evaluated, while the scout bees represent a search for new candidate solutions. The algorithm updates the candidate solutions based on their quality, with higher quality solutions being more likely to generate new candidate solutions [44]. Even lower quality solutions have a chance to contribute by generating new candidate solutions with some degree of randomness. ABC combines local search through employed and onlooker bees with global search through scout bees to efficiently explore

Proposed Approach GWO-Employed-Onlooker Pseudo Code

```
1. Start
2. Initialize the Grey Wolf population
3. Initialize the Onlooker and Employed Bee populations
4. Calculate the fitness of each Grey Wolf
5. Set the maximum number of iterations
6. Initialize the iteration counter to 1
7. While the termination criteria are not met and the maximum number of iterations
 has not been reached, do the following:
    a. For the first half of iterations, update the positions of Grey Wolves using
     Onlooker and Employed Bee operators of ABC:
        i. Calculate the fitness of each Onlooker and Scout Bee
        ii. Select the best Onlooker Bee and update the position of a Grey Wolf
        iii. If the fitness of the Employed Bee is better than the worst Grey Wolf,
        replace the worst Grey Wolf with the Employed Bee
    b. For the second half of iterations, update the positions of Grey Wolves using
    the standard GWO algorithm:
        i. Calculate the fitness of each Grey Wolf
        ii. Update the positions of Alpha, Beta, and Delta Grey Wolves
        iii. Increment the iteration counter
8. End while loop
9. Output the best solution found
```

4.2 Artificial Bee Colony Algorithm Working Mechanism

ABC is an optimization algorithm that draws inspiration from the foraging behaviour of honeybees. This algorithm simulates the behaviours of three distinct types of bees: employed, onlooker, and scout [42]. Employed bees investigate their immediate surroundings for food sources and share their information about quality with onlooker bees. Onlooker bees utilize this information to determine which food sources to visit and also engage in communication with one another to exchange relevant details. Scout bees, on the other hand, actively explore uncharted areas in search of new food sources [43]. In the context of optimization, the food sources represent potential solutions to the problem, with their quality reflecting the fitness or objective function value

the solution space and find high-quality solutions to the optimization problem [45].

4.3 Grey Wolf Algorithm Working Mechanism

The Grey Wolf Optimization (GWO) algorithm is an optimization technique inspired by grey wolf social structure and hunting behaviour [46]. In this algorithm, a population of grey wolves is initially generated randomly within the search space, with every wolf representing a probable solution to the problem of optimization. Alpha, beta, and delta are then identified based on their fitness values, which link to the best, 2nd best, and 3rd best optimal solution in the population [47]. The positions of the remaining wolves are updated using equations based on the positions of the alpha, beta, and delta wolves. If any updated positions exceed the boundaries of the search space, they are adjusted to the nearest boundary. The fitness values of the modified positions are then analyzed, and the population is updated by picking the wolves with the best fitness values [48]. By utilizing the alpha wolf to explore new regions of the search space, the GWO algorithm strikes a balance between exploration and exploitation [49]. The algorithm concludes when a termination criterion is met, either by hitting the upper bound of iterations or attaining the desired level of convergence. Overall, the standard GWO algorithm is known for its simplicity, efficiency, and satisfactory performance across various optimization problems [50].

5 Assessment of the Proposed Model

The proposed model's effectiveness was evaluated by considering various metrics, including the rate at which it converged, its ability to generate the best possible solutions, the average quality of the solutions it generated, the worst quality solution it generated, and the degree of variation or deviation among the solutions it produced.

5.1 Simulation Results and Discussion

In this part of this paper, we provide an overview of the hardware and software employed for conducting the research. The experiments were carried out using an HP EliteBook 850 G4 laptop equipped with a 7th-generation Intel Core i7 processor and 8 GB of RAM. To facilitate efficient execution, we utilized MATLAB R2017a to implement the proposed model as well as other comparable methods.

5.2 Minimization Benchmark Functions

A set of eight benchmark functions for minimization purposes was utilized in this study. Equation 8 represents the mathematical expression for function f_1. It is a combination of exponential terms and cosine functions involving x_1 and x_2. Equation 9 describes function f_2, which involves absolute values and quadratic terms of x_1 and x_2. Equation 10 represents function f_3, a combination of quadratic and squared terms of x_1 and x_2. Equation 11 describes f_4, a combination of quadratic and linear terms of x_1 and x_2. Equation 12 represents f_5, a function involving a complex combination of exponential, cosine, and quadratic terms of x_1 and x_2. Equation 13 describes f_6, a function involving a quadratic term and cosine functions of x_1 and x_2. Equation 14 represents f_7, a function with quadratic and product terms of x_1 and x_2. Equation 15 describes f_8, a function involving polynomial terms of x_1 and x_2. Eqs. 8–15 and Figs. 2 and 3 together form a comprehensive set of minimization benchmark functions and their corresponding convergence behaviours, providing valuable insights into the optimization process

$$f_1 = e^{-(x_1-4)^2-(x_2-4)^2} + e^{-(x_1-4)^2-(x_2-4)^2} + 2e^{-(x_1)^2-(x_2-4)^2} + 2e^{-(x_1)^2-(x_2)^2}, \tag{8}$$

where $x_i \in \{-5, 5\}$

$$f_2 = |x_1| - 5 + (|x_2| + 5)^2, \tag{9}$$

where $x_i \in \{-10, 10\}$

$$f_3 = 100\left((x_2 - x_1)^2\right)^2 + \left(6.4(x_2 - 0.5)^2 - x_1 - 0.6\right)^2, \tag{10}$$

where $x_i \in \{-5, 5\}$

$$f_4 = 100\left((x_2 - x_1)^2\right)^2 + (x_1 - 1)^2, \tag{11}$$

where $x_i \in \{-2.48, 2.48\}$

$$f_5 = \left(x_2 - \frac{5.1}{4\pi^2}(x_1)^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10, \tag{12}$$

where $x_i \in \{-5, 15\}$

$$f_6 = (x_1 + 2x_2 - 7)^2 - \cos(18x_1) - \cos(18x_2), \tag{13}$$

where $x_i \in \{-1, 1\}$

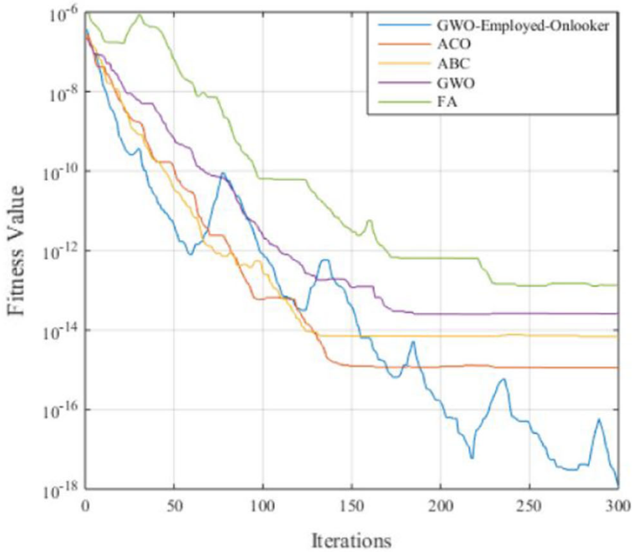
$$f_7 = (x_1 + 2x_2 - 7)^2 + (2(x_1)(x_2) - 5)^2, \tag{14}$$

where $x_i \in \{-10, 10\}$

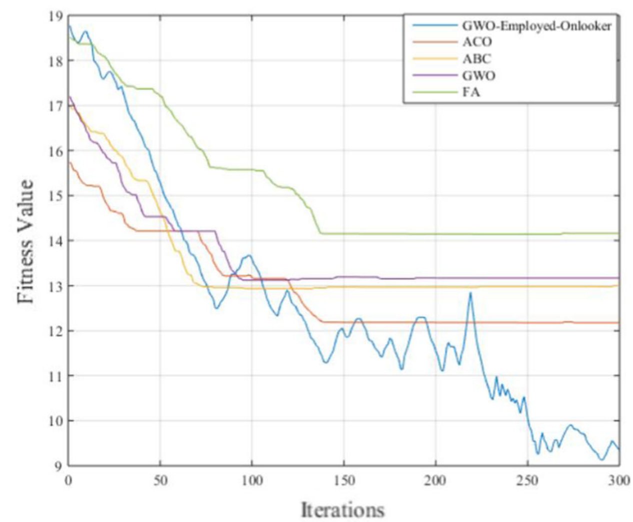
$$f_8 = 4(x_1)^2 + 2.1(x_1)^4 + \frac{1}{3}(x_1)^6 + (x_1)(x_2) - 4(x_1)^2 + 4(x_1)^4, \tag{15}$$

where $x_i \in \{-5, 5\}$

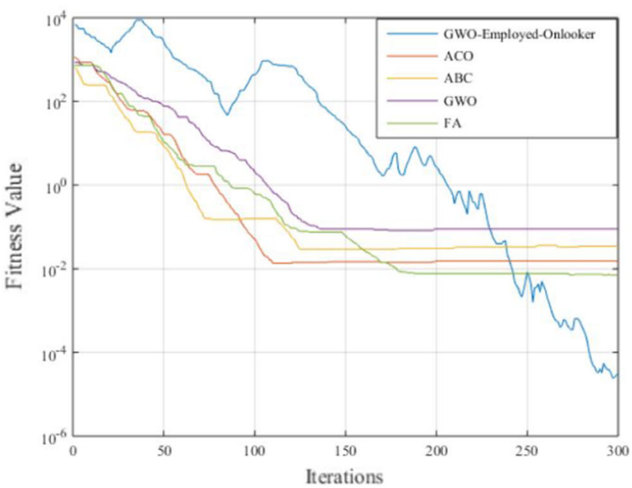
According to Figs. 2a–d and 3a–d, when it comes to minimizing functions, the suggested method is optimal than standard GWO, standard FA, ABC, and ACO in terms of how quickly it converges. Figure 2a shows that the convergence graph of the GWO-Employed-Onlooker method has many fluctuations before the 150th iteration, but after that, the proposed model's convergence rate is better than GWO, FA, ABC, and ACO. Figure 2b shows similar fluctuations, but after the 200th iteration, the proposed model's solution



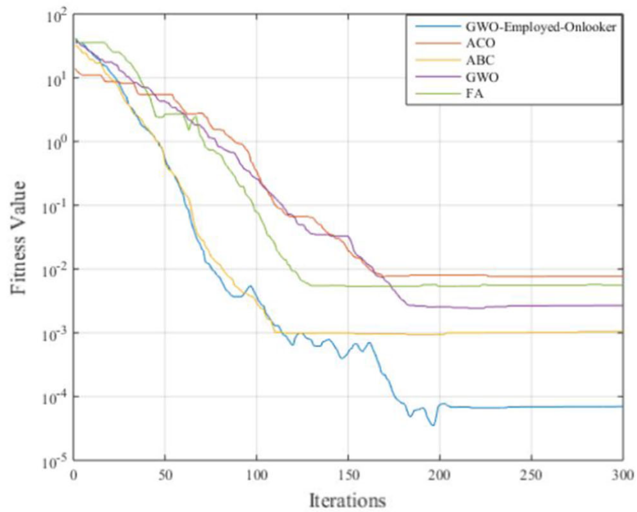
(a) F1 Convergence



(b) F2 Convergence



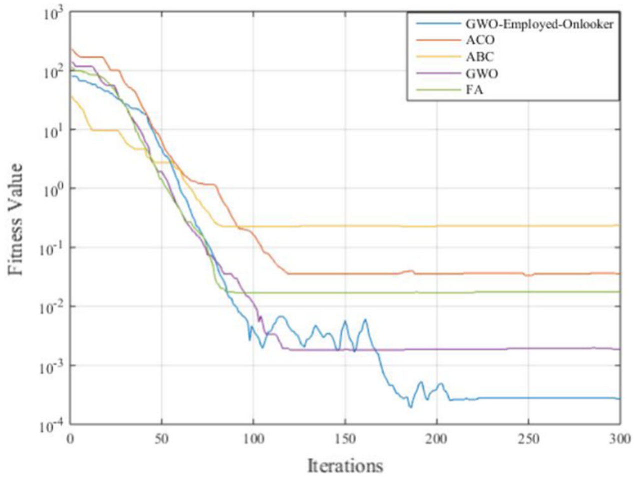
(c) F3 Convergence



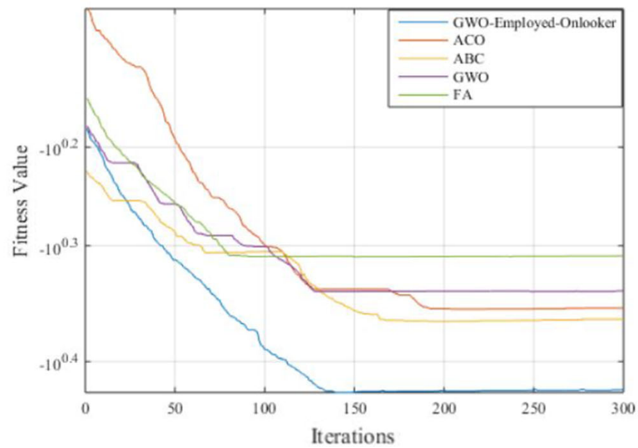
(d) F4 Convergence

Fig. 2 Convergence graphs F1, F2, F3, F4

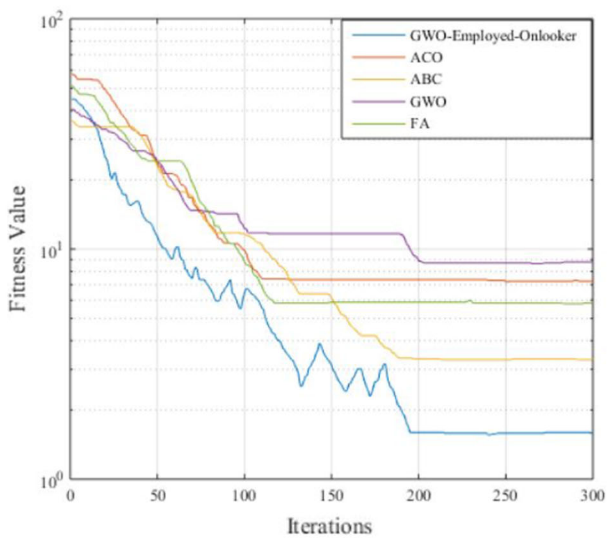
quality is better than all other considered algorithms. Figure 2c his figure illustrates the convergence graph for the optimization process of function f_3. The graph reveals an improvement in solution quality over iterations. Specifically, the GWO-Employed-Onlooker method consistently outperforms standard GWO, FA, ABC, and ACO in terms of both convergence speed and solution quality for the function f_3. This suggests that the proposed method is particularly effective in minimizing f_3, offering a more efficient optimization solution. Figure 2d is similar to the trend observed in Fig. 3c, this graph depicts the convergence behaviour for function f_4. The GWO-Employed-Onlooker method continues to outperform the standard algorithms (GWO, FA, ABC, and ACO) in terms of convergence speed and solution quality for f_4. The consistent improvement over iterations further supports the efficiency of the proposed method in minimizing f_4. Figure 3a convergence graph for function f_5 demonstrates a similar pattern of improvement in solution quality over iterations. The proposed GWO-Employed-Onlooker method consistently outshines GWO, FA, ABC, and ACO, indicating its efficacy in achieving faster convergence and superior solution quality for f_5. Figure 3b graph, representing the convergence behaviour for function f_6, the GWO-Employed-Onlooker method again exhibits a superior performance compared to standard GWO, FA, ABC, and ACO. The trend of improving solution quality over iterations is evident, emphasizing the effectiveness of the proposed method in minimizing f_6. The convergence graph for function f_7 shows



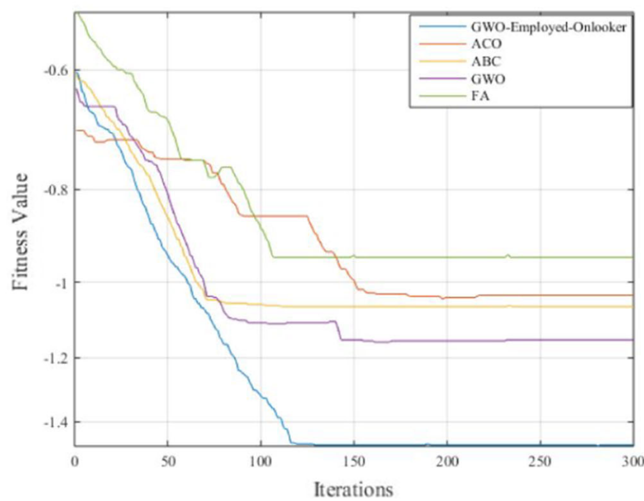
(a) F5 Convergence



(b) F6 Convergence



(c) F7 Convergence



(d) F8 Convergence

Fig. 3 Convergence graphs F5, F6, F7, F8

a consistent trend of the GWO-Employed-Onlooker method outperforming other algorithms (GWO, FA, ABC, and ACO) in terms of both convergence speed and solution quality. This reinforces the conclusion that the proposed method is particularly efficient in minimizing f_7. Finally, the convergence graph for function f_8 follows a similar pattern observed in the previous figures. The GWO-Employed-Onlooker method consistently surpasses the standard algorithms (GWO, FA, ABC, and ACO) in achieving faster convergence and better solution quality for f_8. This collective evidence across multiple functions supports the conclusion that the suggested method is more effective in minimizing various functions compared to the standard algorithms. According to Table 1, the proposed model's best, average, and worst-case solutions in minimization functions are better than GWO, ACO, FA, and ABC. Nevertheless, the proposed model exhibits a higher standard deviation compared to other models. This disparity can be attributed to result variations resulting from the use of different parameters. The process of tuning these parameters involved a trial-and-error approach, as there are no established rules for their configuration.

5.3 Maximization Benchmark Functions

$$f_9 = -(20 + (x_1)^2 - 10\cos(2\pi x_1) + (x_2)^2 \\ -10\cos(2\pi x_1)), \tag{16}$$

where $x_i \in \{-2.048, 2.048\}$.

Equation 16 represents the mathematical formulation of function f_9 for maximization purposes. It involves a combination of quadratic terms, cosine functions, and constants.

Table 1 GWO-Employed-Onlooker model comparison for minimization functions with other techniques

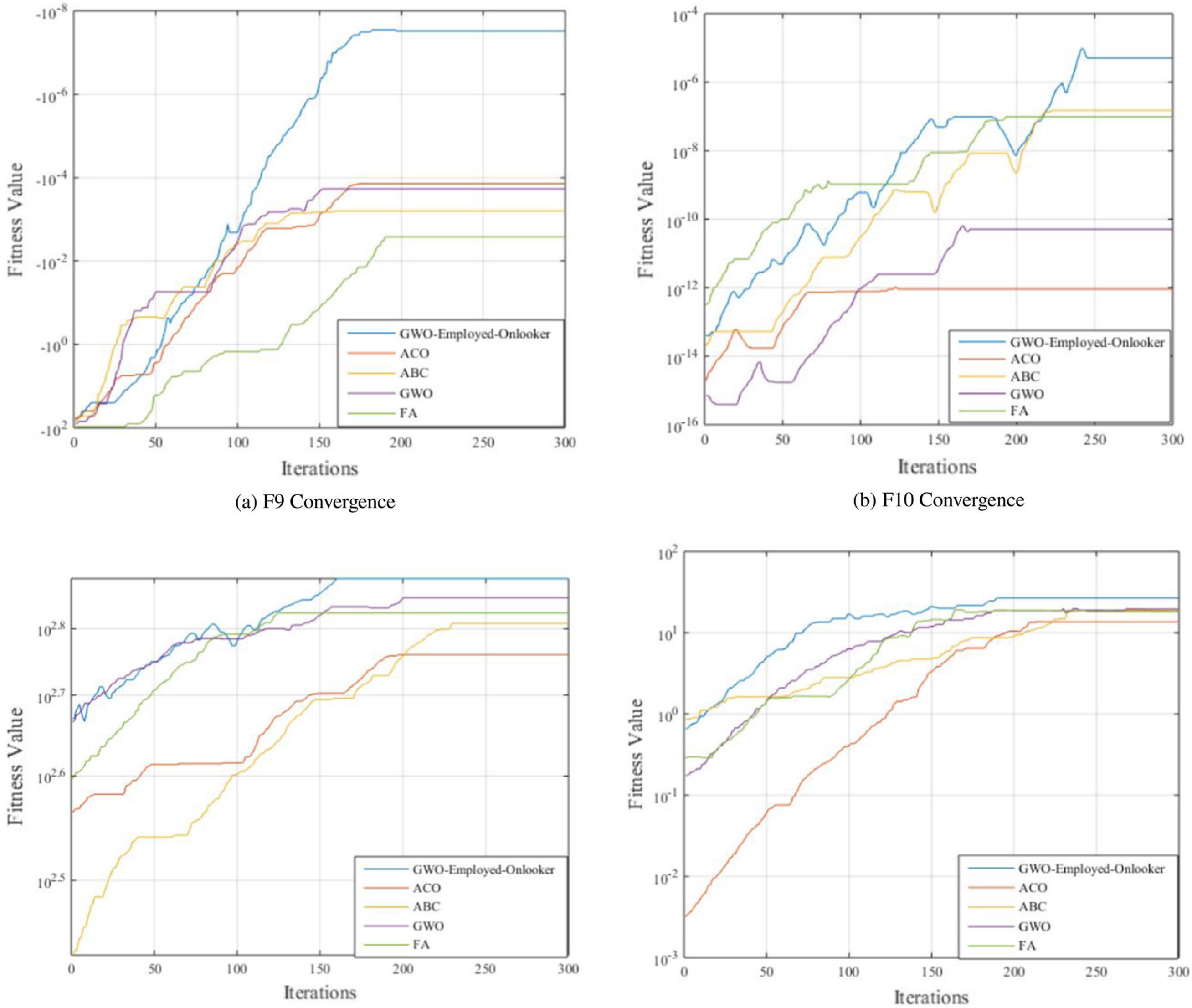
Function	Technique	Best minimum	Worst minimum	Average minimum	Standard deviation
F1	ACO	7.56E−14	6.57E−09	7.78E−10	1.75045E−09
	FA	6.56E−12	6.67E−06	7.05E−07	1.44094E−06
	GWO	7.09E−13	6.11E−10	3.30E−11	1.22123E−10
	ABC	8.32E−14	9.65E−09	1.65E−09	3.37546E−09
	GWO-Emp-Onl	**6.34E−17**	**8.45E−11**	**4.67E−12**	**1.80048E−11**
F2	ACO	1.20E+02	1.31E+02	1.21E+02	5.45E−02
	FA	1.20E+02	1.41E+02	1.29E+02	2.64E−02
	GWO	1.20E+02	1.36E+02	1.25E+02	4.43E−02
	ABC	1.18E+02	1.29E+02	1.20E+02	2.67E−02
	GWO-Emp-Onl	**9.10E+01**	**1.10E+02**	**1.05E+02**	**6.49E−02**
F3	ACO	3.43E−03	9.88E−02	1.70E−02	2.41E−02
	FA	7.34E−03	9.61E−02	3.06E−02	3.30E−02
	GWO	7.12E−03	9.64E−03	8.56E−03	8.11E−03
	ABC	3.43E−03	8.23E−02	2.05E−02	2.50E−02
	GWO-Emp-Onl	**6.46E−06**	**6.79E−04**	**5.80E−05**	**1.41E−04**
F4	ACO	6.78E−04	4.33E+01	2.72E+01	1.10E+01
	FA	6.78E−04	5.15E+01	2.40E+01	1.03E+01
	GWO	2.34E−04	4.13E+01	1.83E+01	1.37E+01
	ABC	1.56E−04	4.13E+01	1.72E+01	1.16E+01
	GWO-Emp-Onl	**6.45E−06**	**8.36E−03**	**6.19E−04**	**1.79E−03**
F5	ACO	2.12E−02	2.13E+02	8.79E+01	7.11E+01
	FA	1.57E−03	1.23E+02	6.31E+01	4.09E+01
	GWO	1.64E−04	9.46E+01	3.39E+01	3.39E+01
	ABC	2.09E−02	7.29E+01	3.11E+01	2.22E+01
	GWO-Emp-Onl	**2.57E−05**	**3.13E+01**	**8.71E−02**	**1.11E+01**
F6	ACO	− 2.11E+01	− 1.69E+01	− 2.11E+01	9.29E−03
	FA	− 1.81E+01	− 1.81E+01	− 1.79E+01	3.69E−03
	GWO	− 2.11E+01	− 1.69E+01	− 1.79E+01	8.49E−03
	ABC	− 2.19E+01	− 1.71E+01	− 2.11E+01	1.51E−02
	GWO-Emp-Onl	**− 2.81E+01**	**2.21E+01**	**− 2.31E+01**	**1.11E+01**
F7	ACO	7.09E+01	1.56E+02	1.12E+02	2.07E+01
	FA	5.68E+01	1.13E+02	8.83E+01	2.03E+01
	GWO	8.46E+01	2.35E+02	1.19E+02	3.51E+01
	ABC	7.09E+01	1.03E+02	7.29E+01	1.74E+01
	GWO-Emp-Onl	**1.55E+01**	**7.51E+01**	**4.79E+01**	**1.44E+01**
F8	ACO	− 1.21E+01	1.07E+01	− 1.05E+01	5.16E−02
	FA	− 1.15E+01	− 8.33E−02	− 9.07E−02	7.44E−03
	GWO	− 1.31E+01	− 1.12E+01	− 1.22E+01	3.22E−03
	ABC	− 1.19E+01	− 8.63E−02	− 9.63E−02	1.05E−02
	GWO-Emp-Onl	**− 1.49E+01**	**− 1.29E+01**	**− 1.51E+01**	**4.41E−03**

Bold values represent the optimal values obtained during simulation process

Under normal circumstances, x_1 and x_2 are within the range $[-2.048, 2.048]$. Function f_9 is formulated to be maximized, incorporating both quadratic and cosine terms. The specified range for variables ensures a bounded domain.

$$f_{10} = \cos x_1 \cos x_2 \exp((x_1 - \pi)^2 - (x_2 - \pi)^2), \tag{17}$$

where $x_i \in \{-20, 20\}$.



(a) F9 Convergence

(b) F10 Convergence

(c) F11 Convergence

(d) F12 Convergence

Fig. 4 Convergence graphs F9, F10, F11, F12

Equation 17 defines function f_{10} designed for maximization. It involves cosine functions, exponential terms, and constants. Both x_1 and x_2 are allowed to vary within the range $[-20, 20]$. Function f_{10} captures a more complex relationship between variables, with a broader range. The inclusion of exponential and cosine terms adds intricacy to the optimization landscape.

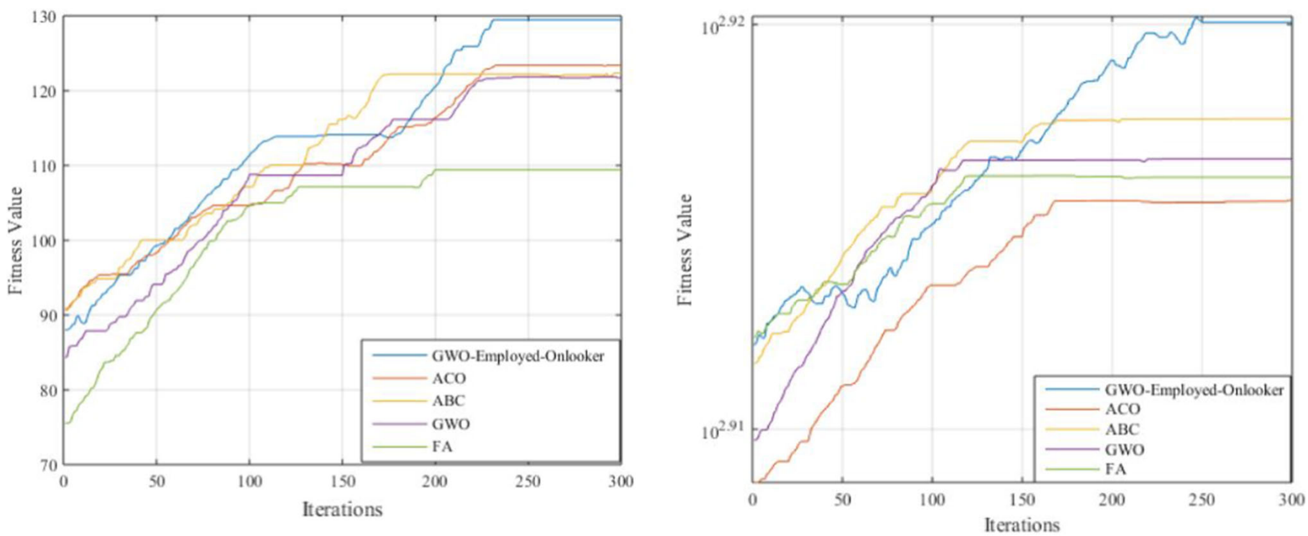
$$f_{11} = 280 - \frac{1}{2}((x_1)^4 - 16(x_1)^2 - 5x_1) - \frac{1}{2}((x_2)^4 - 16(x_2)^2 - 5x_2), \tag{18}$$

where $x_i \in \{-5, 5\}$

Equation 18 defines function f_{11} for maximization, comprising polynomial terms with coefficients and constants. Both x_1 and x_2 are limited to the range $[-5, 5]$. Function f_{11} introduces polynomial terms with specific coefficients. The limited variable range suggests a focused exploration of the optimization landscape.

$$\begin{aligned} f_{12} = \ln \Bigg[& \left(\sin\left(\cos x_1 + \cos x_2\right)^2\right)^2 \\ & - \left(\cos\left(\sin x_1 + \sin x_2\right)^2\right)^2 + x_1 \Bigg]^2 \\ & -0.1\left((x_1 - 1)^2 + (x_2 - 1)^2\right), \end{aligned} \tag{19}$$

where $x_i \in \{-10, 10\}$.



(a) F13 Convergence

(b) F14 Convergence

Fig. 5 Convergence graphs F13, F14

Equation 19 characterizes function f_{12} for maximization. It involves logarithmic, trigonometric, and polynomial terms. Both x_1 and x_2 are constrained within the range $[-10, 10]$. Function f_{12} incorporates a logarithmic term and a combination of trigonometric functions, adding complexity to the optimization landscape.

$$f_{13} = \left[\sum_{i=1}^{5} \operatorname{cis}((i+1)x_1 + i)\right]\left[\sum_{i=1}^{5} \operatorname{cis}((i+1)x_2 + i)\right], \tag{20}$$

where $x_i \in \{-10, 10\}$

Equation 20 defines function f_{13} for maximization, involving the sum of cosine functions. Both x_1 and x_2 are allowed to vary within the range $[-10, 10]$. Function f_{13} captures a complex relationship between variables through the summation of cosine functions. The broad variable range suggests a more extensive exploration of the optimization landscape,

$$f_{14} = 660 - \left((x_1)^2 + x_2 - 11\right)^2 - \left(x_1 + (x_2)^2 - 7\right)^2, \tag{21}$$

where $x_i \in \{-6, 6\}$

Equation 14 characterizes function f_{14} designed for maximization. It involves polynomial terms with coefficients and constants. Both x_1 and x_2 are constrained within the range $[-6, 6]$. Function f_{14} introduces polynomial terms with specific coefficients and constants, providing a more focused exploration of the optimization landscape.

The convergence graph for function f_9 displays a notable pattern. The GWO-Employed-Onlooker method exhibits a steady improvement in convergence over iterations, indicating an effective exploration of the optimization landscape. While some fluctuations are visible, the overall trend demonstrates the algorithm's ability to converge towards the maximization of f_9. The convergence graph for function f_{10} reveals a consistent and significant improvement in convergence as the iterations progress. The GWO-Employed-Onlooker method outperforms standard GWO, FA, ABC, and ACO, showcasing its efficiency in maximizing f_{10}. The convergence rate suggests a robust exploration of the solution space. The convergence graph for function f_{11} indicates a smooth and steady improvement in convergence over iterations. The GWO-Employed-Onlooker method consistently outpaces standard algorithms, reflecting its effectiveness in maximizing f_{11}. The convergence rate suggests a stable and efficient exploration of the optimization landscape. The convergence graph for function f_{12} exhibits a pattern of continuous improvement in convergence. The GWO-Employed-Onlooker method displays superior performance compared to standard algorithms, indicating its effectiveness in maximizing f_{12}. The convergence rate suggests a robust and efficient exploration of the solution space for this function.

The convergence graph for function f_{13} reveals a consistent improvement in convergence over iterations. The GWO-Employed-Onlooker method outperforms standard algorithms, showcasing its efficiency in maximizing f_{13}. The convergence rate suggests a robust and effective exploration of the optimization landscape for this function. The convergence graph for function f_{14} shows a steady improvement in convergence over iterations. The GWO-Employed-Onlooker method consistently outpaces standard algorithms, indicat-

Table 2 GWO-Employed-Onlooker model comparison for maximization functions with other techniques

Function	Technique	Best maximum	Worst maximum	Average maximum	Standard deviation
F9	ACO	− 5.51E−03	− 1.34E−05	− 3.23E−04	1.13E−03
	FA	− 4.39E−02	− 1.51E−02	− 1.05E−01	9.49E−02
	GWO	− 4.57E−03	− 3.17E−04	− 1.29E−03	1.18E−03
	ABC	− 4.38E−02	− 1.39E−03	− 1.19E−02	1.25E−02
	GWO-Emp-Onl	**− 6.46E−06**	**− 3.46E−07**	**− 1.31E−06**	**1.70E−06**
F10	ACO	3.46E−12	6.24E−11	1.47E−11	1.66E−11
	FA	3.46E−08	9.56E−07	3.22E−07	3.53E−07
	GWO	3.46E−08	9.86E−09	5.35E−09	3.34E−09
	ABC	4.44E−08	9.73E−07	2.77E−07	2.78E−07
	GWO-Emp-Onl	**3.57E−07**	**9.46E−05**	**3.62E−05**	**3.55E−05**
F11	ACO	5.23E+01	4.64E+01	5.02E+01	2.03E+00
	FA	5.19E+01	5.59E+01	6.11E+01	1.74E+00
	GWO	6.59E+01	6.11E+01	6.29E+01	2.05E+00
	ABC	6.09E+01	5.59E+01	5.79E+01	1.37E+00
	GWO-Emp-Onl	**7.76E+01**	**7.54E+01**	**7.66E+01**	**5.76E+00**
F12	ACO	1.53E+02	1.11E+02	1.31E+02	1.71E+01
	FA	1.73E+02	1.31E+02	1.49E+02	1.31E+01
	GWO	1.61E+02	1.21E+02	1.49E+02	1.79E+01
	ABC	1.60E+02	1.11E+02	1.31E+02	1.73E+01
	GWO-Emp-Onl	**2.16E+02**	**1.69E+02**	**2.05E+02**	**1.19E+01**
F13	ACO	1.09E+03	9.46E+02	1.07E+03	7.29E+01
	FA	1.05E+03	8.56E+02	9.83E+02	7.56E+01
	GWO	1.09E+03	1.59E+02	1.05E+03	2.13E+02
	ABC	1.08E+03	1.03E+03	1.03E+03	6.06E+01
	GWO-Emp-Onl	**1.19E+03**	**1.09E+03**	**1.17E+03**	**2.11E+01**
F14	ACO	8.13E+03	8.06E+03	8.07E+03	2.12E+01
	FA	8.13E+03	8.04E+03	8.08E+03	2.17E+01
	GWO	8.14E+03	8.05E+03	8.08E+03	3.03E+01
	ABC	8.16E+03	8.07E+03	8.11E+03	2.67E+01
	GWO-Emp-Onl	**8.22E+03**	**8.16E+03**	**8.19E+03**	**1.79E+01**

Bold values represent the optimal values obtained during simulation process

ing its effectiveness in maximizing f_{14}. The convergence rate suggests a stable and efficient exploration of the solution space for this function.

Figures 4 and 5 reveal that the proposed model experiences fluctuations in solution quality during the initial iterations, but eventually demonstrates a convergence rate superior to that of other algorithms. Table 2 shows that the proposed model outperforms GWO, FA, ABC, and ACO in terms of best, average, and worst-case solutions for all maximization functions, although it has a higher standard deviation compared to the other models. To summarize the entire research work, including the development, representation, operation, and implementation of the GWO and ABC hybrid model, this section provides a comprehensive overview.

6 Conclusive Remarks

The paper presents a solution to a significant issue that affects the standard Grey Wolf Optimization (GWO) algorithm, resulting in a reduction in the quality of its solutions. The proposed solution involves incorporating two operators from the Artificial Bee Colony (ABC) algorithm into the exploration phase of GWO. Specifically, the GWO algorithm suffers from poor exploitation during the initial half of the exploration phase. During each iteration, there may be significant differences between the present solution and the preceding solution, leading to an expanded search space that encompasses a wide range of distinct solutions. As a result, the optimal solution may be overlooked, even if it is nearby previously found solutions. To address this issue, the paper proposes using the employed and onlooker operators from the ABC algorithm as exploitative operators, in the current search domain. Once the specified operations have been carried out on the bees, the fitness values of functions are assessed. Once its fitness value matches the required value, the algorithm terminates. If the condition is not met, the wolves assume the role of the bees in the subsequent iteration, this looping procedure is repeated unless the upper bound of iterations has been achieved and the termination requirements have been fulfilled.

Author Contributions Conceptualization: Ishaq Ahmad and Fawad Qayum; methodology: Fawad Qayum, Sami Ur Rehman, and Gautam Srivastava; validation: Fawad Qayumi, Sami Ur Rehman and Gautam Srivastava; writing: Sami Ur Rehman, and Gautam Srivastava; writing—review and editing: Fawad Qayum, Sami Ur Rehman, and Gautam Srivastava; supervision: Ishaq Ahmad and Gautam Srivastava

Funding This research received no funding.

Data Availability The data are available upon request from the authors.

Declarations

Conflict of interest The authors declare that there are no conflicts of interest in this paper.

Ethical Approval This article does not contain any studies with human participants performed by any of the authors.

References

1. Mafarja, M., Awadallah, M.A., Mirjalili, S., Aljarah, I.: A novel multi-objective cuckoo search algorithm for feature selection. Swarm Intell. **15**(2), 89–106 (2021)
2. Hassan, M.J., Azmi, R., Shamsuddin, S.A., Alrajeh, N.A.: Exploration vs. exploitation in swarm intelligence: a comprehensive survey. IEEE Access **8**, 92545–92571 (2020)
3. Blum, C., Dorigo, M., Maniezzo, V., Stützle, D.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, Oxford (2019)
4. Zhang, Y., Zhang, W., Yu, Y., Tian, Y.: An improved ant colony algorithm for multi-objective optimization in complex networks. IEEE Access **9**, 39963–39973 (2021)
5. Zhou, S., Zhang, Q., Jiao, L.: A multi-objective particle swarm optimization algorithm based on adaptive search radius control. IEEE Trans. Evol. Comput. **25**(2), 174–189 (2021)
6. Oliveira, T.H.S., de Carvalho, L.A.V., Gonçalves, G.R.: An artificial bee colony algorithm for the resource-constrained project scheduling problem with resource transfers. IEEE Access **9**, 24458–24476 (2021)
7. Wang, M., Liu, X., Wang, Z.: A hybrid discrete grey wolf optimizer for the permutation flow-shop scheduling problem. IEEE Access **9**, 40726–40738 (2021)
8. Azad, M.A.K., Hassan, M.F., Noor, R.W.JMd.: Performance analysis of multi-objective krill herd algorithm on robot path planning problem. IEEE Access **9**, 65817–65835 (2021)
9. Tran, T.T., Nguyen, V.H., Nguyen, H.L., Nguyen, D.H.: A hybrid bat algorithm for the economic load dispatch problem with valve-point effects. IEEE Access **9**, 112147–112161 (2021)
10. Maayah, B., Abu Arqub, O.: Uncertain M-fractional differential problems: existence, uniqueness, and approximations using Hilbert reproducing technique provisioner with the case application: series resistor-inductor circuit. Phys. Scr. **99**, 025220 (2024)
11. Sujit, S.S., Thampi, S.: Hybrid grey Wolf optimization algorithm for continuous optimization problems. J. Intell. Syst. **26**(4), 465–475 (2017)
12. Maurya, S.P., Singh, R.K., Singh, D.: A novel differential evolution algorithm based on grey wolf optimization for numerical function optimization. J. Comput. Sci. **27**, 40–54 (2018)
13. Kumar, R., Sharma, R.: Hybrid grey Wolf optimization and genetic algorithm for non-convex economic load dispatch. Int. J. Electr. Power Energy Syst. **83**, 212–220 (2016)
14. Gao, H., Yang, J., Huang, Q.: A novel hybrid optimization algorithm based on grey Wolf optimizer and ant lion optimizer. J. Intell. Fuzzy Syst. **32**(6), 4293–4303 (2017)
15. Abu Arqub, O., Mezghiche, R., Maayah, B.: Fuzzy M-fractional integrodifferential models: theoretical existence and uniqueness results, and approximate solutions utilizing the Hilbert reproducing kernel algorithm. Front. Phys. **11**, 1252919 (2023)
16. Rao, T.V.M., Palakollu, M.L., Kumar, N.S.N.: Grey wolf optimizer based multi-objective optimal power flow considering voltage stability. In: 2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia). Chengdu, China (2019)
17. Alam, M., Khare, A., Gupta, N.: Fuzzy rule-based decision support system for assessing a student's performance using grey Wolf optimization algorithm. In: 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India (2020)
18. Lu, H., Li, Y., Zhang, Y., Li, F.: An efficient feature selection algorithm based on improved grey wolf optimization for hyperspectral image classification. In: 2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Kunming, China (2018)
19. Liu, J., Wang, J., Wu, C., Zeng, Z.: Fault diagnosis of rolling bearings using grey Wolf optimizer and wavelet packet transform. In: 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), Xi'an, China (2019)
20. Taha, R.A., Taha, S.A.A., Taha, M.H.: Grey Wolf optimization algorithm for reducing downtime in a semiconductor manufacturing industry. In: 2020 10th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), Tlemcen, Algeria (2020)
21. Abo-Hammour, Z., Abu Arqub, O., Momani, S., Shawagfeh, N.: Optimization solution of Troesch's and Bratu's problems of ordinary type using novel continuous genetic algorithm. Discrete Dyn. Nat. Soc. **2014**, 1–15 (2014)
22. Abu Arqub, O., Abo-Hammour, Z.: Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. Inf. Sci. **279**, 396–415 (2014)
23. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey Wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)

24. Kumar, A., Kumar, S., Singh, S.K., Anand, R.S.: A hybrid GPSO-GWO algorithm for economic dispatch problem. Neural Comput. Appl. **31**, 671–687 (2019)
25. Li, X., Yang, L., Wang, X., Chen, X., Guo, Y.: A hybrid genetic algorithm and grey Wolf optimizer for permutation flowshop scheduling problem. J. Ambient Intell. Humaniz. Comput. **11**, 3143–3155 (2020)
26. Zhang, Y., Li, H., Shen, Y., Wei, L.: A hybrid GWO–PSO–SVM algorithm for breast cancer diagnosis. J. Ambient Intell. Humaniz. Comput. **11**, 1237–1248 (2020)
27. Wang, S., Tang, Y., Li, X., Guo, X.: A hybrid algorithm based on grey wolf optimizer and differential evolution for image thresholding. J. Ambient Intell. Humaniz. Comput. **9**, 1197–1208 (2018)
28. Sahu, S.K., Pati, U.K.: A hybrid GWO–ABC algorithm for multi-objective optimization. J. Ambient Intell. Humaniz. Comput. **12**, 8497–8515 (2021)
29. Tabrizi, H.B., Hosseini, S.H., Hooshmand, R.: Hybrid grey Wolf optimizer and artificial bee colony algorithm for solving economic emission dispatch problem. Energy **126**, 479–494 (2017)
30. Mozafari, M.R., Meybodi, M.R., Fazel Zarandi, M.H.: A novel hybrid algorithm based on grey Wolf optimizer and genetic algorithm for solving constrained optimization problems. Appl. Intell. **48**, 3576–3592 (2018)
31. Karthikeyan, A., Sugumaran, V., Arulselvi, S.: A hybrid grey wolf optimizer and particle swarm optimization algorithm for feature selection in intrusion detection systems. Appl. Soft Comput. **84**, 105722 (2019)
32. Dhillon, Y.K., Kumar, S., Kumar, A.: A hybrid grey Wolf optimizer and cuckoo search algorithm for parameter identification of photovoltaic models. J. Ambient Intell. Humaniz. Comput. **11**, 2497–2509 (2020)
33. Subramanyam, S.V., Chanda, R.K., Panigrahi, B.K.: A hybrid grey Wolf optimizer and teaching learning based optimization algorithm for optimal power flow problem. Electr. Power Energy Syst. **132**, 15–28 (2021)
34. Kumar, R., Singh, S.: A hybrid GWO algorithm with shuffled frog leaping algorithm for feature selection in intrusion detection systems (2021)
35. Sharma, S.K., Singh, A.K.: A hybrid GWO algorithm with particle swarm optimization for simultaneous placement and routing of analog circuits (2021)
36. Sahu, S.K., Swain, S.C.: A hybrid GWO algorithm with artificial bee colony algorithm for tuning of PID controller for AVR system (2021)
37. Pakhira, M.R., Das, R.K.: A hybrid GWO algorithm with differential evolution for feature selection in microarray data classification (2020)
38. Maity, S.K., Rakshit, A.: A hybrid differential evolution and grey Wolf optimizer algorithm for optimal power flow problem. In: 2017 IEEE 7th International Conference on Power Systems (ICPS), pp. 1–6 (2017)
39. Chen, P., Xu, X., Tang, X., Liu, J.: An improved grey Wolf optimizer with dimension-wise search strategy for feature selection. IEEE Access **9**, 5097–5110 (2021)
40. Kachhap, P.K., Jana, P.K.: An improved grey Wolf optimizer for optimal feature selection and classification in medical diagnosis. Expert Syst. Appl. **170**, 114590 (2021)
41. Khan, M.R., Islam, M.A.: A modified grey Wolf optimizer for large-scale optimization problems. IEEE Access **9**, 67162–67171 (2021)
42. Yang, S., Zhang, J.: An enhanced grey Wolf optimizer with scale factor and parameter adaptation for numerical optimization problems. IEEE Access **9**, 7076–7085 (2021)
43. Rana, R.K., Acharjya, D.P.: A survey on the working mechanism of artificial bee colony algorithm. IEEE Access **7**, 32369–32387 (2019)
44. El-Hawary, M.E., El-Shorbagy, R.M.: A hybridization of firefly algorithm and artificial bee colony algorithm for global optimization. IEEE Access **7**, 105126–105146 (2019)
45. Rezoug, A.M., Abdi, H., Boutaib, M.S., Bensaali, F.Z.: A novel combination of artificial bee colony and particle swarm optimization algorithms for efficient feature selection. In: 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA), Abu Dhabi, United Arab Emirates (2019)
46. Yang, X.S., Deb, S.: Improved grey Wolf optimizer: a comprehensive analysis of operator types and working mechanism. Expert Syst. Appl. **174**, 114690 (2021)
47. Wang, X., Wang, L.: A new gray Wolf optimizer based on the study of its working mechanism. Soft Comput. **2021**, 1–13 (2021)
48. Liu, X., Guo, Y., Xie, Y.: Analysis of the working mechanism of the grey Wolf optimizer based on mutation operator. Soft Comput. **2021**, 1–15 (2021)
49. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: An analytical investigation of the working mechanism of the grey Wolf optimizer. Inf. Sci. **574**, 192–207 (2021)
50. Zhang, X., Wang, B., Sun, J.: Working mechanism analysis of the grey Wolf optimizer based on attractor selection. IEEE Access **9**, 63657–63670 (2021)