j. port. sci. res.
ISSN: 2616-7441 (Online)
ISSN: 2616-6232 (Print)
ISSN: 2616-7220 (USB)
INTERNATIONAL IDENTIFIER FOR SERIALS

Journal port Science Research
Available online www.jport.co
Volume 6, No:4. 2023

A proposed Hyper-Heuristic optimizer Nesting Grey Wolf Optimizer and COOT Algorithm for Multilevel Task

[1]Afrah U. Mosaa, [2]Waleed A Mahmoud Al-Jawherb

[1]*Informatics Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatics, Baghdad, Iraq*

[2]*College of Engineering, Uruk University, Baghdad, Iraq.*

phd202110688@iips.icci.edu.iq

Abstract It can be extremely difficult to find the optimal solution in many complex optimization problems. The goal of optimization algorithms in such cases is to locate a feasible solution that is as close as possible to the optimal one. These algorithms are called metaheuristic optimization algorithms and the majority of them take their inspiration from nature and work to solve challenging problems in a variety of fields. In this paper, a combination between GWO and Coot algorithm was proposed. The effectiveness of the GWO algorithm has been demonstrated in many fields, including engineering and medicine. However, GWO has a disadvantage: the potential to enter the local minima due to a lack of diversity. GWO and the Coot algorithm were merged to fix this flaw. Ten benchmark functions were used to evaluate the performance of this hybrid technique, and its results were compared to those of other common optimization algorithms, including GWO, Cuckoo Search (CS), and the Shuffled Frog Leaping algorithm (SFLA). The results show that the suggested algorithm can provide results that are both competitive and more consistent than the other algorithms in most test functions.

Crossref 10.36371/port.2023.4.1

Keywords: *metaheuristic; optimization; grey wolf optimization; Coot algorithm, hybrid algorithm.*

1. INTRODUCTION

Optimization algorithms concern with finding the optimal solution to a problem. These algorithms are widely used in various fields, including engineering, economics, finance, and computer science [1]. There are many Optimization algorithms such as Gradient Descent which is used to find the minimum of a function by iteratively adjusting the parameters in the direction of steepest descent[2]. In many cases, finding the optimal solution is almost impossible because it is computationally infeasible or too time-consuming, so the role of optimization algorithms in that case is to find approximate solutions to complex optimization problems. This class of optimization algorithms called metaheuristic optimization algorithms and most of them draw inspiration from nature.[3] Today, These algorithms are widely used in different fields because their simplicity, adaptability, and avoidance of local optimums [3], [4]. Genetic Algorithm, is one of metaheuristic algorithms which is inspired by biological evolution and use a population of candidate solutions to find the optimal solution to a problem[5]. Swarm Optimization algorithms are a class of optimization algorithms that are inspired by the behavior of social animals, such as bees, ants, birds, and fish. These algorithms use the collective behavior of a group of individuals to find the optimal solution to a problem. There are many examples of these types of algorithms such as particle swarm optimization, which is inspired by the behavior of swarms of particles and uses their collective behavior to find the optimal solution to a problem. Ant Colony Optimization (ACO) is an algorithm that is inspired by the behavior of ants when they search for food. In ACO, artificial ants deposit pheromones to mark the path that leads to the optimal solution to a problem[6]. Artificial Bee Colony (ABC) is a swarm optimization algorithm that is inspired by the behavior of honey bees. In ABC, the population of bees collaborates to find the optimal solution to a problem[7]. Cuckoo Search (CS) is another example of Swarm algorithms which is inspired by the behavior of cuckoo birds. In CS, the cuckoo birds lay their eggs in the nests of other birds, and the search process is based on the competition between the cuckoo birds.[8]

Every optimization algorithm has a weaknesses[9], prompting scholars to develop hybrid algorithms that combine multiple algorithm to enhance performance and overcome these drawbacks[10]. The motivation behind algorithm hybridization stems from the desire to harness the complementary strengths of different optimization methods, with the expectation that cooperation among these methods will lead to improvements [11], [12]. For example, in 2007, Shelokar et al. introduced the particle swarm ant colony

Afrah U. Mosaa*, 2Waleed A Mahmoud Al-Jawherb, 2023. A proposed Hyper-Heuristic optimizer Nesting Grey Wolf Optimizer and COOT Algorithm for Multilevel Task.. *Journal port Science Research*, 6(4), pp. 310-317. https://doi.org/10.36371/port.2023.4.1

optimization (PSACO) by merging Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO)[9]. Another hybrid algorithm emerged when Hoseini and Shayesteh combined Genetic Algorithm (GA), Ant Colony Optimization (ACO), and Simulated Annealing (SA)[13]. GAAPI represents a hybrid algorithm that combines a specialized Colony Optimization (API) with the Genetic Algorithm based on the work of.[14]

Mirjalili, Mirjalili, and Lewis (2014) introduced a novel algorithm inspired by Grey Wolves named Grey Wolves Optimizer (GWO) [15]. In spite of the GWO algorithm has demonstrated its efficiency various fields such as medicine, engineering, and machine learning [16]–[18], it has a drawback, which is that it may fall into local minima [19], [20]. To overcome the drawback of GWO, the proposed algorithm hybridizes it with another metaheuristic algorithm that is the coot algorithm.

The remainder of the paper is structured as follows: Section 2 provides a brief overview of the COOT algorithm, while Section 3 outlines the fundamental principles of GWO. Section 4 describes the proposed algorithm, and in Section 5 present the results alongside the employed benchmark functions. Section 6 encompasses the discussion and conclusion.

2. COOT ALGORITHM

Coots are little birds that belong to the rail family. These birds exhibit a variety of behaviors and motions. The coot algorithm simulates the movements of flocks of these birds on the surface of the water [21]. These movements consists of three movements which are

1. Random movement or a disordered movement to the sides.
2. Chain movement where each bird follows the one before it.
3. Changing the position in accordance with the group leaders.
4. Leading the group by the leaders towards the optimal area.

The algorithm starts by generating random individuals using (1) to constitute the population.

CootPos(i)= rand(1,d).*(ub- lb)+lb (1)

Where CootPos(i) refers to the coot position, d is the dimensions of the problem, lb is the lower bound, and ub is the upper bound of the search space.

After the population is formed, each individual in it follows one of the movements mentioned. These four movements are represented through mathematical equations, which will be outlined in the following.

2.1 Random movement

In this movement, a random position within the search space is generated using (2). Once we have a random position, it will be the target for the coot movement.

$$Q = rand(1,d).*(ub + lb) + lb \qquad (2)$$

This coot movement investigates different areas of the search space. This movement will enable the algorithm to avoid getting stuck in a local optimal. The new position of the coot will be calculating as the following (equation 3)

$$CootPos(i) = CootPos(i) + A \times R2 \times \left(Q - CootPos(i)\right) \qquad (3)$$

Where R2 is a random number in the interval [0, 1] and A is calculated by (4).

$$A = 1 - L \times (1\, Iter) \quad (4)$$

Where Iter is the number of iterations and L is the number of the current iteration.

2.2 Chain Movement

In a sequential movement, each bird follows the one before it. To implement this movement, the average distance between the two coots is calculated as shown in (5).

$$CootPos(i) = 0.5 \times \left(CootPos(i-1) + CootPos(i)\right) \qquad (5)$$

Where CootPos(i) and CootPos(i- 1) are the positions of two successive coots.

2.3 Adjusting the position based on the group leaders

In this movement, the leader must be determined for each coot, and this is done by (6).

$$K = 1 + (iMODNL) \qquad (6)$$

Where K represents the leader's index, *i* is the index of the coot and NL the number of leaders.

After defining the leader for each coot, the coot position is updated with (7).

$$CootPos(i) = LeaderPos(k) + 2 \times R1 \times \cos(2R\pi) \times \left(LeaderPos(k) - CootPos(i)\right) \quad (7)$$

Where LeaderPos(k) is the position of the leader, R1 is a random number between [0,1] while R a random numbers from -1 to 1.

Leading the group by the leaders towards the optimal area (leader movement)

Afrah U. Mosaa*, 2Waleed A Mahmoud Al-Jawherb, 2023. A proposed Hyper-Heuristic optimizer Nesting Grey Wolf Optimizer and COOT Algorithm for Multilevel Task.. *Journal port Science Research*, 6(4), pp. 310-317. https://doi.org/10.36371/port.2023.4.1

j. port. sci. res.
ISSN: 2616-7441 (Online)
ISSN: 2616-6232 (Print)
ISSN: 2616-7220 (USB)
INTERNATIONAL IDENTIFIER FOR SERIALS
Journal port Science Research
Available online www.jport.co
Volume 6, No:4. 2023

Leaders in the group update their positions according to the global optimum position to lead the swarm toward it.

$$LeaderPos(i) = \begin{cases} B \times R3 \times \cos(2R\pi) \times \left(gBest - LeaderPos(i)\right) + gBest & R4 < 0.5 \\ B \times R3 \times \cos(2R\pi) \times \left(gBest - LeaderPos(i)\right) - gBest & R4 >= 0.5 \end{cases} \quad (8)$$

R3 and R4 are random numbers in the interval [0, 1], R is a random number in the interval [1, 1], and B is calculated in accordance with . In coot algorithm, the coot follows one of three movements, and one of these movements is chosen by generating a random number, as well as for leaders, as the leaders approach the global optimum or move away according to a random number. The following figure describe the coot algorithm (Fig 1).

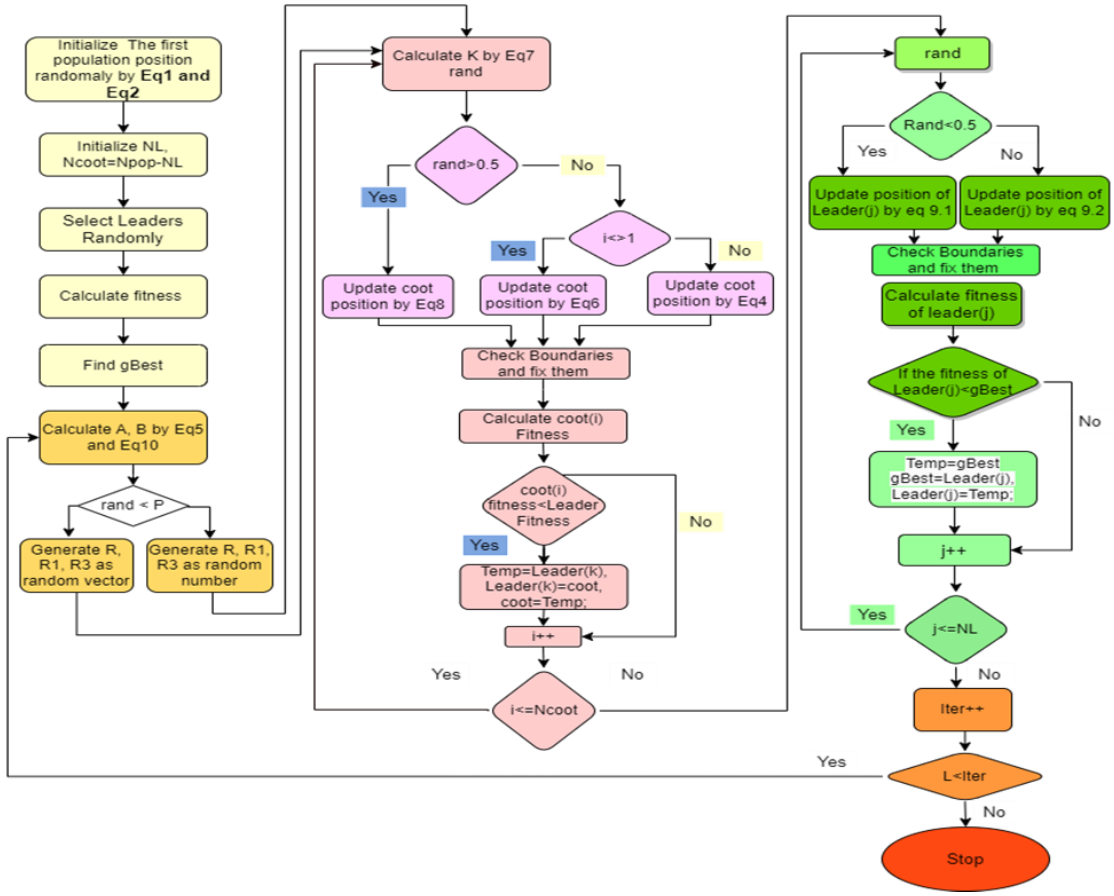

Fig. 1. .*Coot algorithm diagram*

3. GREY WOLF OPTIMIZER (GWO)

The Grey Wolf Optimization (GWO) algorithm is inspired by the social hierarchy and hunting behavior of grey wolves, where the alpha wolf is the leader of the pack and makes decisions related to hunting and other pack activities [15]. The GWO algorithm is a population-based metaheuristic optimization algorithm that mimics the hunting behavior of grey wolves in finding the optimal solution to a given problem. In this algorithm, the position of each wolf in the pack represents a potential solution to the optimization problem, and the hunting behavior of the pack is simulated through the search for the optimal solution. The alpha wolf plays a crucial role in guiding the search process. It does not have to be the strongest in the pack, but it must be the most capable of managing. All pack members must follow the alpha wolf's instructions.

The social hierarchy of grey wolves, the alpha wolf is assisted by beta wolves, which are subordinate to the alpha but have a higher rank than the other wolves in the pack. Beta wolves help the alpha wolf in making decisions related to the pack's activities, such as hunting, and they also help in maintaining the social order of the pack. The beta wolf follows the orders of the alpha wolf. The rest of the pack members follow the orders of both the alpha and beta wolves, and the beta wolf is considered a potential future alpha wolf and may replace the alpha wolf when it is no longer capable of leading the pack.

The third layer is occupied by the delta wolf (δ), which follows the commands of the alpha and beta wolves and also leads the rest of the pack. The delta wolf is responsible for tasks such as hunting, scouting, and taking care of weaker or injured wolves. The rest of the pack members outside the alpha, beta, and delta wolves follow the commands of the delta wolf.

Afrah U. Mosaa*, 2Waleed A Mahmoud Al-Jawherb, 2023. A proposed Hyper-Heuristic optimizer Nesting Grey Wolf Optimizer and COOT Algorithm for Multilevel Task.. *Journal port Science Research*, 6(4), pp. 310-317. https://doi.org/10.36371/port.2023.4.1

j. port. sci. res.
ISSN: 2616-7441 (Online)
ISSN: 2616-6232 (Print)
ISSN: 2616-7220 (USB)
INTERNATIONAL IDENTIFIER FOR SERIALS
Journal port Science Research
Available online www.jport.co
Volume 6, No:4. 2023

At the base of the hierarchy are the omega wolves (ω), which are the lowest-ranking members of the pack and are dominated by the alpha, beta, and delta wolves. The omega wolves often receive the least amount of food.

In the third layer of the pack hierarchy is the delta wolf (δ), which follows the alpha and beta commands, but the wolves outside of these types follow delta wolf. The tasks of delta wolves include hunting, scouting, and taking care of weaker wolves. At the base of the hierarchy are the omega wolves (ω), which are dominated by the other three types. Figure 1 shows the social hierarchy of the grey wolf pack.

The social hierarchy and hunting strategy of the grey wolf pack have been mathematically simulated and modeled in the GWO algorithm to guide the optimization process. In the GWO algorithm, the alpha wolf represents the best solution found so far, while the beta and delta wolves represent the second and third-best solutions, respectively. The other solutions in the population are modified to approach the three best solutions represented by the alpha, beta, and delta wolves.

The hunting technique of grey wolves typically involves three phases: encircling, harassing, and capturing the prey. In the first phase, the wolves work together to encircle the prey, cutting off its escape routes and forcing it to stay within a smaller area. This phase is modeled mathematically in the following equations:

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \qquad (12)$$

$$\vec{C} = 2.\vec{r}_2 \qquad (13$$

Where t is the current iteration. X represents the position of the wolf and Xp represents the position of the prey. A and C are the coefficients and they can be calculated by the following equations:

A ⃗=2(a) ⃗.(r) ⃗_1 - (a) ⃗ (12)

C ⃗=2.(r) ⃗_2 (13)

(a) ⃗ values decrease from 2 to 0 over the iterations. (r) ⃗_1 and (r) ⃗_2 are random in range.(0,1)

To find the best position according to the prey position, the wolves' positions are changed by modifying A and C.

During the second phase, the wolves begin to harass the prey, making it more vulnerable and weakening its defenses. This phase often involves chasing and nipping at the prey, trying to wear it down and separate it from the herd if it is a herd animal.

The second phase after encircling is the hunting phase. The dominant wolf (α) directs the hunt. Beta and delta wolves occasionally take part. The best solution is considered as alpha by the GWO algorithm, while the next two best solutions are beta and delta. These three positions are saved by the algorithm, which then modifies the other individuals' positions according to them. These steps in the hunting process are modeled as follows:

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1.\vec{D}_\alpha \qquad (14)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2.\vec{D}_\beta \qquad (15)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3.\vec{D}_\delta \qquad (16)$$

$$\vec{X}_{t+1} = \frac{\vec{X}_1+\vec{X}_2+\vec{X}_3}{3} \qquad (17)$$

Where X ⃗_(t+1) is a new position and it is the mean value of three values which are calculated depending on α, δ, and ω .

The last phase in hunting is attacking prey. This phase is modeled mathematically by decreasing the value of (a) ⃗ from 2 to 0.

4. THE PROPOSED ALGORITHM

The proposed algorithm is a combination of GWO and COOT algorithms. The algorithm starts with GWO steps and gives initial values for a, A, and C. Then the algorithm moves to the COOT part, where it begins with generating a random population and determines the number of leaders. The steps of coot algorithm continue until the specified number of iterations ends. After the coot algorithm ends, it returns the final value of the global optimum to the GWO algorithm where this value will be the prey for it. The proposed algorithm steps are:

1. Initialize a, A and C randomly.
2. Determine the prey position by COOT algorithm.
   - **2.1** Initialize the first population position randomly by (1) and (2) and P.
   - **2.2** Initialize Number of Leaders (NL) , Ncoot=Npop-NL.
   - **2.3** Select Leaders Randomly.
   - **2.4** Calculate fitness.
   - **2.5** Find gBest.
   - **2.6** Calculate A, B by (5) and (9)
   - **2.7** Generate a random number (rand),
   - **2.8** if rand <P
   - **2.9** Generate R, R1, R3 as random vector
   - **2.10** Else
   - **2.11** Generate R, R1, R3 as random number
   - **2.12** Calculate K by (7) and generate rand.
   - **2.13** If rand>0.5
   - **2.14** Update coot position by (8)
   - **2.15** Else
   - **2.16** if the index of coot (i) =1
   - **2.17** Update coot position by (6)
   - **2.18** Else

Afrah U. Mosaa*, 2Waleed A Mahmoud Al-Jawherb, 2023. A proposed Hyper-Heuristic optimizer Nesting Grey Wolf Optimizer and COOT Algorithm for Multilevel Task.. *Journal port Science Research*, 6(4), pp. 310-317. https://doi.org/10.36371/port.2023.4.1

j. port. sci. res.
ISSN: 2616-7441 (Online)
ISSN: 2616-6232 (Print)
ISSN: 2616-7220 (USB)
INTERNATIONAL IDENTIFIER FOR SERIALS

Journal port Science Research
Available online www.jport.co
Volume 6, No:4. 2023

2.19 Update coot position by (4)
2.20 Check Boundaries and fix them
2.21 Calculate coot(i) Fitness
2.22 If the fitness of coot <fitness of its leader switch them.
2.23 For Number of leaders
2.24 Generate rand
2.25 If rand<0.5 Update position of Leader(j) by (9.1)
2.26 Else Update position of Leader(j) by (9.2)
2.27 Check Boundaries and fix them
2.28 Calculate fitness of leader
2.29 If the fitness of leader <fitness of gbest switch them
2.30 Return the final value of global optimum(gbest) to GWO algorithm and stop.

3. Determine Xα, Xβ and Xδ
4. While T<No. of iterations
5. Select new a, A and C randomly
6. Update each individual position
7. Update Xα, Xβ and Xδ
8. Return the final Xα, Xβ and Xδ and stop.

As mentioned earlier, GWO may converge to a local minimum. Therefore, in the proposed algorithm, the coot algorithm has been integrated with it to enhance exploration. In this approach, the population in the coot algorithm is divided into subgroups that explore promising areas within the search space, thereby increasing exploration. Fig.2 shows the algorithm steps.

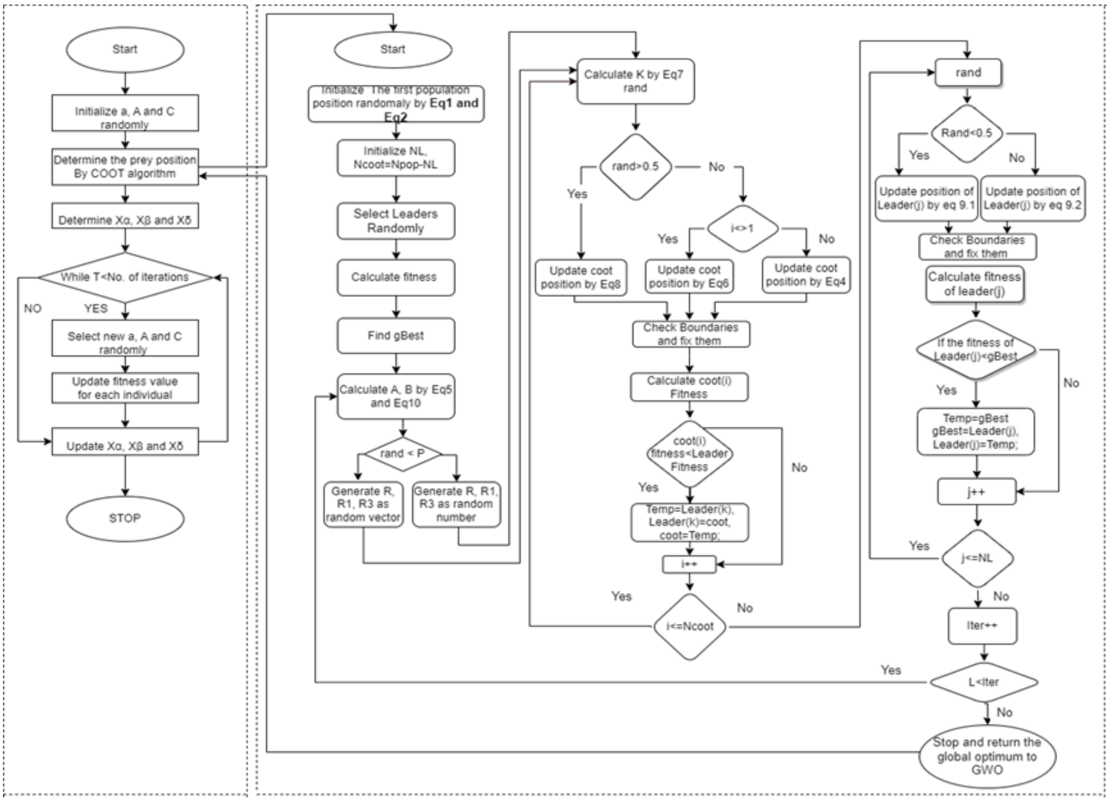

Fig. 2 The proposed algorithm diagram

5. RESULTS

The proposed algorithm was tested by using ten minimization multimodal functions. The chosen functions have various scales to evaluate the algorithm performance and its efficiency with different scales. The used benchmark functions are represented in table (1) and the figure bellow shows their plots (fig 3).

j. port. sci. res.
ISSN: 2616-7441 (Online)
ISSN: 2616-6232 (Print)
ISSN: 2616-7220 (USB)
INTERNATIONAL IDENTIFIER FOR SERIALS

Journal port Science Research
Available online www.jport.co
Volume 6, No:4. 2023

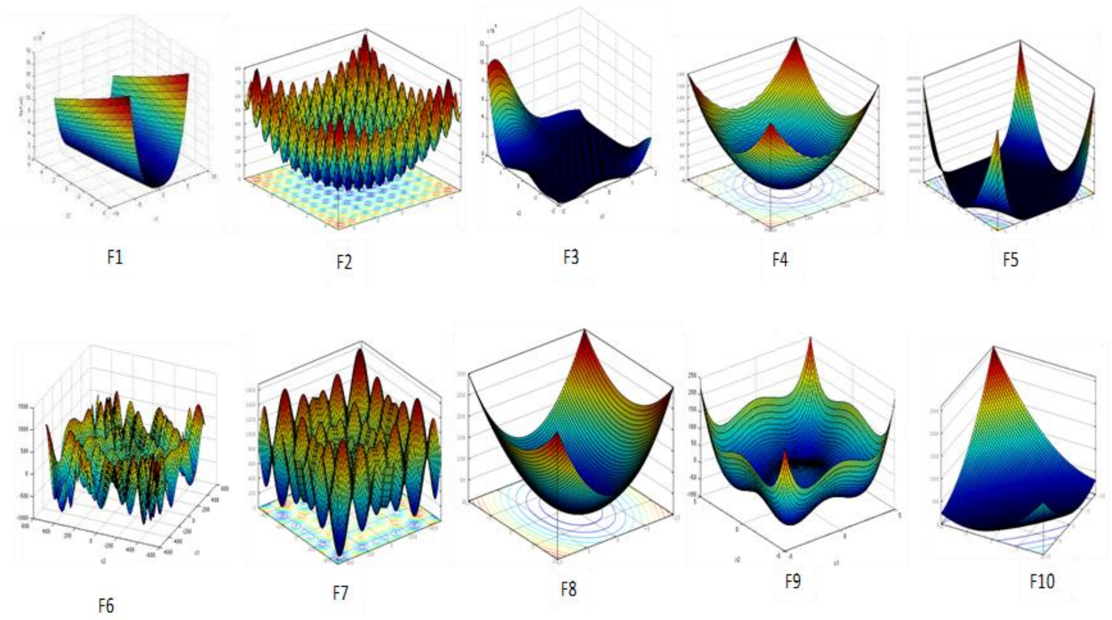

Fig. 3 *Benchmark functions plots*

Table 1 *Benchmark functions*

Function	**Formula**	**d**	**Range**
Rosenbrock function	$F_1(x) = \sum_{i=1}^{d-1}\left[100\left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right]$	2	[-100,100]
Rastrigin function	$F_2(x) = 10d + \sum_{i=1}^{d}\left[x_i^2 - A\cos(2\pi x_i)\right]$	2	[-5.12,5.12]
Goldstein-Price function	$F_3(x,y) = [1 + (x + y + 1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)]\times$ $[30+(2x - 3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 27y^2)\,]$	2	[-2,2]
Griewank function	$F_4(x) = \sum_{i=1}^{d}\frac{x_i^2}{4000} - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	2	[-600, 600]
Beale function	$f_5(x,y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x - xy^3)^2$	2	[-4.5,4.5]
Eggholder function	$F_6(x,y) = -(y + 47)\sin\sqrt{\left\lvert\frac{x}{2} + (y + 47)\right\rvert} - x\sin\sqrt{\lvert x - (y + 47)\rvert}$	2	[-512,512]
Schwefel function	$F_7(x) = 418.982887272433d - \sum_{i=1}^{d} x_i\sin\sqrt{\lvert x_i\rvert}$	2	[-500,500]
Sum Squares function	$F_8(x) = \sum_{i=1}^{d} i x_i^2$	2	[-10,10]

j. port. sci. res.
ISSN: 2616-7441 (Online)
ISSN: 2616-6232 (Print)
ISSN: 2616-7220 (USB)
INTERNATIONAL IDENTIFIER FOR SERIALS
Journal port Science Research
Available online www.jport.co
Volume 6, No:4. 2023

Styblinski-Tang function	$F_9(x) = \frac{1}{2}\sum_{i=1}^{d}(x_i^4 - 16x_i^2 + 5x_i)$	2	[-100,100]
Booth function	$F_{10}(x,y) = (x+2y-7)^2 + (2x+y-5)^2$	2	[-10,10]

The new algorithm, along with three other algorithms (GWO, Cuckoo Search, and Shuffled Frog Leaping Algorithm - SFLA), was executed 20 times for each function to assess and compare their performance. Table 2 displays the obtained results.

***Table 2** The results of the algorithms*

Functions	*CS*	*SFLA*	*GWO*	*The Proposed Algorithm*
F1	0.014	0.002	4.8×10^{-4}	$3*10^{-4}$
F2	1.605	1.106	0.007	$7*10^{-4}$
F3	3.005	3.125	3.039	3.001
F4	0.025	0.131	0.048	3.369×10^{-5}
F5	0.007	0.002	0.001	0.0007
F6	-704.330	-948.909	-926.361	-932.163
F7	1.480	0.002	0.093	0.012
F8	1.368×10^{-8}	9.010×10^{-5}	5.539×10^{-10}	6.31×10^{-11}
F9	-76.332	-76.163	-71.522	-75.328
F10	0.26	0.001	0.006	4×10^{-4}

The algorithm outperformed other algorithms by delivering better and more consist results. The other algorithms sometimes produced large results. For instance, when retesting the third function (F3), the Grey Wolf Optimizer (GWO) yielded significantly large results in four out of the twenty retest. This indicates that the hybridization was successful and improved the results.

6. CONCLUSION

To assess the effectiveness of the proposed hybrid algorithm, which integrates GWO and the COOT algorithm, its performance was evaluated across ten different test functions. Its performance was compared with the performance of the conventional GWO, to evaluate the benefits of the hybridization.

The results indicated that the combination of these two algorithms led to enhance the performance across the majority of benchmark functions. For example, its performance in the Griewank function (F4) was the best, with a result of 3.369×10-5, surpassing GWO, which result 0.048, and significantly outperforming the Cuckoo Search and Frog Leaping algorithms, which their results 0.025 and 0.025, respectively. The same holds true for functions F1, F2, F3 and F5 as well.

The Frog Leaping algorithm performed better in functions F6, F7 and F9, while CS algorithm was better in F9 only.

As the results indicate, the combination of GWO and coot algorithms has succeeded in delivering better results in most of the test functions. Additionally, the algorithm's performance across 20 test runs was consistently close, with results that do not deviate significantly from the optimal solution.

The aut☒hors now working on using the proposed hyper-heuristic is a heuristic search method that seeks to automate, often by the incorporation of machine learning techniques, the process of selecting, combining, generating or adapting several transforms (like Wavelet, Multiwavelet, Walidlet and Hybrid) to efficiently solve novel feature search problems. One of the motivations for studying hyper-heuristics is to build new OFDM systems which can handle classes of problems rather than solving just one problem in data communication [22-33].

REFERENCES

[1] A. H. Halim and I. I. Swagatam, *Performance assessment of the metaheuristic optimization algorithms : an exhaustive review*, vol. 54, no. 3. Springer Netherlands, 2021. doi: 10.1007/s10462-020-09906-6.

[2] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization Methods for Large-Scale Machine Learning," *SIAM Rev.*, vol. 60, no. 2, pp. 223–311, Jan. 2018, doi: 10.1137/16M1080173.

[3] M. Abd Elaziz *et al.*, "Advanced metaheuristic optimization techniques in applications of deep neural networks: a review," *Neural Comput. Appl.*, vol. 33, no. 21, pp. 14079–14099, Nov. 2021, doi: 10.1007/s00521-021-05960-5.

[4] B. Arandian, A. Iraji, H. Alaei, S. Keawsawasvong, and M. L. Nehdi, “White-Tailed Eagle Algorithm for Global Optimization and Low-Cost and Low-CO2 Emission Design of Retaining Structures,” *Sustainability*, vol. 14, no. 17, p. 10673, Aug. 2022, doi: 10.3390/su141710673.

[5] J. H. Holland, “Genetic Algorithms,” *Sci. Am.*, vol. 267, no. 1, pp. 66–72, Jul. 1992, doi: 10.1038/scientificamerican0792-66.

[6] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[7] D. Karaboga and B. Basturk, “Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems,” in *Foundations of Fuzzy Logic and Soft Computing*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 789–798. doi: 10.1007/978-3-540-72950-1_77.

[8] X.-S. Yang and Suash Deb, “Cuckoo Search via Lévy flights,” in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, IEEE, 2009, pp. 210–214. doi: 10.1109/NABIC.2009.5393690.

[9] P. S. Shelokar, P. Siarry, V. K. Jayaraman, and B. D. Kulkarni, “Particle swarm and ant colony algorithms hybridized for improved continuous optimization,” *Appl. Math. Comput.*, vol. 188, no. 1, pp. 129–142, May 2007, doi: 10.1016/j.amc.2006.09.098.

[10] A. Bouaouda and Y. Sayouti, “Hybrid Meta-Heuristic Algorithms for Optimal Sizing of Hybrid Renewable Energy System: A Review of the State-of-the-Art,” *Arch. Comput. Methods Eng.*, vol. 29, no. 6, pp. 4049–4083, Oct. 2022, doi: 10.1007/s11831-022-09730-x.

[11] Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, “Grasshopper Optimization Algorithm: Theory, Variants, and Applications,” *IEEE Access*, vol. 9, pp. 50001–50024, 2021, doi: 10.1109/ACCESS.2021.3067597.

[12] J. Li, H. Lei, A. H. Alavi, and G.-G. Wang, “Elephant Herding Optimization: Variants, Hybrids, and Applications,” *Mathematics*, vol. 8, no. 9, p. 1415, Aug. 2020, doi: 10.3390/math8091415.

[13] P. Hoseini and M. G. Shayesteh, “Hybrid Ant Colony Optimization, Genetic Algorithm, and Simulated Annealing for image contrast enhancement,” in *IEEE Congress on Evolutionary Computation*, IEEE, Jul. 2010, pp. 1–6. doi: 10.1109/CEC.2010.5586542.

[14] I. Ciornei and E. Kyriakides, “Hybrid Ant Colony-Genetic Algorithm (GAAPI) for Global Continuous Optimization,” *IEEE Trans. Syst. Man, Cybern. Part B*, vol. 42, no. 1, pp. 234–245, Feb. 2012, doi: 10.1109/TSMCB.2011.2164245.

[15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[16] H. U. Ahmed, R. R. Mostafa, A. Mohammed, P. Sihag, and A. Qadir, “Support vector regression (SVR) and grey wolf optimization (GWO) to predict the compressive strength of GGBFS-based geopolymer concrete,” *Neural Comput. Appl.*, vol. 35, no. 3, pp. 2909–2926, Jan. 2023, doi: 10.1007/s00521-022-07724-1.

[17] M. Banaie-Dezfouli, M. H. Nadimi-Shahraki, and Z. Beheshti, “R-GWO: Representative-based grey wolf optimizer for solving engineering problems,” *Appl. Soft Comput.*, vol. 106, p. 107328, Jul. 2021, doi: 10.1016/j.asoc.2021.107328.

[18] M. Ghalambaz, R. Jalilzadeh Yengejeh, and A. H. Davami, “Building energy optimization using Grey Wolf Optimizer (GWO),” *Case Stud. Therm. Eng.*, vol. 27, p. 101250, Oct. 2021, doi: 10.1016/j.csite.2021.101250.

[19] X. Zhang, X. Wang, H. Chen, D. Wang, and Z. Fu, “Improved GWO for large-scale function optimization and MLP optimization in cancer identification,” *Neural Comput. Appl.*, vol. 32, no. 5, pp. 1305–1325, Mar. 2020, doi: 10.1007/s00521-019-04483-4.

[20] A. A. Heidari and P. Pahlavani, “An efficient modified grey wolf optimizer with Lévy flight for optimization tasks,” *Appl. Soft Comput.*, vol. 60, pp. 115–134, Nov. 2017, doi:10.1016/j.asoc.2017.06.044 .