Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 03, September 2021.

WEIGHTED GREY WOLF OPTIMIZER WITH IMPROVED CONVERGENCE RATE IN TRAINING MULTI-LAYER PERCEPTRON TO SOLVE CLASSIFICATION PROBLEMS

Alok Kumar[1], Lekhraj[2] and Anoj Kumar[3]

(Received: 18-May-2021, Revised: 2-Aug.-2021, Accepted: 12-Aug.-2021)

ABSTRACT

The Grey Wolf Optimizer (GWO) is a very recently developed and emerging swarm-intelligent algorithm. The GWO algorithm was inspired by the social dominance hierarchy and hunting strategy of the grey wolves that has been successfully tailored to tackle various discrete and continuous optimization problems. During its practical implementation, however, it may be stuck in sub-optimal solutions (stagnation in local optima) due to its less exploration in the early stages that show the main drawback of this algorithm. Therefore, this research work enhances the hunting and attacking mechanism in order to modify the corresponding position updated equation and exploitation equation, respectively, to propose a novel algorithm, called Weighted Grey Wolf Optimizer with Improved Convergence Rate (WGWOIC). The effectiveness of the proposed algorithm (WGWOIC) is investigated by testing it an 33 different and fairly popular numerical benchmark functions. Although, these test functions are considered from two different benchmark datasets to assess the strength and robustness of the proposed algorithm regarding the unknown search space of the problem. In order to carry out performance analysis, moreover, the WGWOIC's results are compared against many other state-of-the-art meta-heuristic algorithms, such as Particle Swarm Optimization (PSO), Moth-Flame Optimization (MFO), Whale Optimization Algorithm (WOA), Grey Wolf Optimizer (GWO) and very recent variants of GWO. The comparative study for WGWOIC concludes that the proposed algorithm provides very competitive results against other studied meta-heuristic algorithms. Furthermore, the hybridization of the WGWOIC meta-heuristic optimization algorithm with a Multi- Layer Perceptron (MLP) neural network is employed to improve the accuracy of the classification problem. WGWOIC trainer provides the optimal values for weight and biases to the MLP network. Further, the performance is tested in terms of classification accuracy on five popular classification datasets and assesses the efficiency of the WGWOIC trainer is assessed against many other meta-heuristics trainers. The results show that the proposed algorithm eventually provides very competitive outcomes, implying that the WGWOIC algorithm offers a better exploitation, explores the search space and effectively solves several different classification problems.

KEYWORDS

Meta-heuristic algorithms, Evolutionary algorithms, Nature-inspired algorithms, Swarm-based algorithms, Grey wolf optimizer, Neural network, Multi-layer perceptron.

1. INTRODUCTION

Real-world problems have an unknown search space with their unknown solution. Besides, only limited resources and limited time are available to tackle these problems. Therefore, an optimum solution should exist to resolve the above issues and overcome the limitations. Consequently, there should be the existence of such algorithms provided likely to the optimum solution. The optimization algorithms are able to fulfill the above requirements and overcome the above limitations. The optimization algorithms' particular category is called meta-heuristic algorithms, becoming more popular in the last two decades due to their simplicity and flexibility. Its processing begins with random solution(s) and ends at the optimum solution(s), making it more robust. In other words, the meta-heuristic algorithms' initial solutions are known as random solutions; evolve these random solutions evolve through the applied well-known algorithms and obtain final solutions known as optimum solutions. To sum up, the researchers' main aim is to design various new meta-heuristic optimization algorithms and enhance the existing algorithms to obtain global optimum solutions. Surprisingly, the No Free Lunch (NFL) theorem [1] states that a specific optimization algorithm cannot extensively tackle all types of problems; however, not all

A. Kumar, Lekhraj and A. Kumar are with the Department of Computer Science and Engineering, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, UP, India, Emails: [1]alokkumar@mnnit.ac.in,[2]lekhraj@mnnit.ac.in and [3]anojk@mnnit.ac.in

kinds of optimization algorithms can solve a single problem. Consequently, this encourages designing various new meta-heuristic optimization algorithms and improving existing algorithms.

Five types of optimization algorithms come under meta-heuristic algorithms: evolutionary algorithms, bio-stimulated algorithms, physics-based algorithms, nature-inspired algorithms and swarm intelligence-based algorithms. The hierarchical diagram of various meta-heuristic algorithms is depicted in Figure 1. Holland proposed the Genetic Algorithm (GA) [2] in 1992 and it is the most famous algorithm of evolutionary class. The GA algorithm is typically justified by the Darwin's evolution theory. The first real-world application of GA was control system optimization using genetic algorithms, which was proposed by Krishnakumar and Goldberg [3]. Subsequently, the various other evolutionary algorithms such as Differential Evolution (DE) [4] algorithm, Biogeography-Based Optimizer (BBO) [5], Evolutionary Programming (EP) [6], Genetic Programming (GP) [7], …etc., came into the picture. In addition to the above evolutionary algorithms, Covariance Matrix Adaptation (CMA-ES) [8] and Fast Evolutionary Programming (FEP) [9] are other evolutionary meta-heuristic algorithms. In the second class, the Artificial Immune System [10], Bacterial Foraging Optimization (BFO) [11], …etc., come under bio-stimulated algorithms.

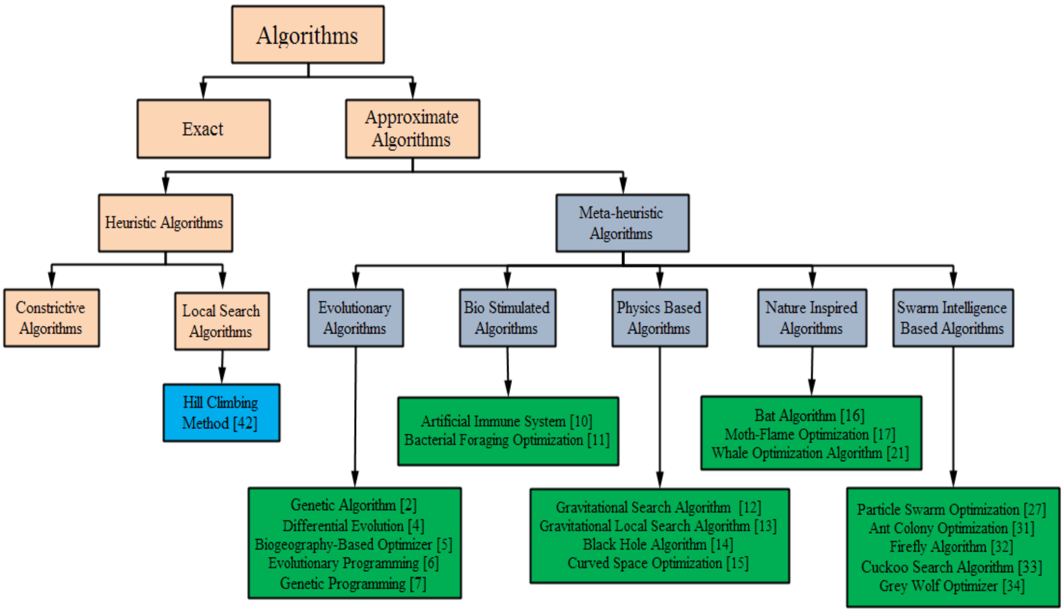


Figure 1. Hierarchical representation of meta-heuristic optimization algorithms.

The third class of meta-heuristic algorithms is referred to as the physics-based algorithms that mimic the physics rules. The Gravitational Search Algorithm (GSA) [12], Gravitational Local Search Algorithm (GLSA) [13], Black Hole (BH) [14] algorithm, Curved Space Optimization (CSO) [15], …etc., come under these algorithms. Subsequently, the Bat Algorithm (BA) [16], Moth-Flame Optimization (MFO) [17], Whale Optimization Algorithm (WOA) [21], …etc., are listed with nature-inspired algorithms that are the fourth category of meta-heuristic algorithms. In addition, the above algorithms are used to tackle real-world applications; for instance, Abed-alguni [22] introduced a novel Q-learning approach using the bat algorithm that finds optimal Q-values and validated the performance on the shortest path problem and the taxi problem. Besides the algorithms mentioned earlier, the State of Mater Search (SMS) [23], Flower Pollination Algorithm (FPA) [24], …etc., are called population-based algorithms that are inspired by a different source.

The last but not most minor class is swarm intelligence-based algorithms, inspired by the intelligent swarm behavior. A large group of homogeneous living species is called a swarm, such as bird flocking and fish schooling. The Particle Swarm Optimization (PSO) [27] is the most famous swarm-based algorithm that mimics the social behavior of birds. Kennedy and Eberhart proposed this algorithm in 1995. In addition to the PSO, Ant Colony Optimization (ACO) [31], Firefly Algorithm (FA) [32], Cuckoo Search Algorithm (CSA) [33], Grey Wolf Optimizer (GWO) [34], …etc., are likely grouped into swarm-based algorithms.

The GWO is a novel and emerging swarm-intelligent algorithm inspired by the grey wolves' social dominance hierarchy and hunting strategy. The GWO has been successfully tailored to tackle various discrete and continuous optimization problems. The main drawback of the GWO is that it may be stuck in sub-optimal solutions (stagnation in local optima) due to its less exploration in the early stages. Nevertheless, the exploration and exploitation should be adequately balanced to extensively investigate the search space for achieving the most optimal solutions. In order to overcome the above limitations, Mittal et al. [35] proposed an advanced variant of GWO; namely, Modified Grey Wolf Optimizer (mGWO). This variant offers a pertinent equilibrium among exploration and exploitation for the search space; however, the modification is attempted in only controlling parameter $\vec{a}$; hence, further improvement may be possible. Furthermore, Singh [36] enhanced the biological order of the social hierarchy of grey wolves in order to propose another variant of GWO. However, this research work improved only the biological structure regarding the social hierarchy of grey wolves, but not the significant enhancement in the mathematical model accordingly. In addition, Kumar et al. [38] proposed another variant of GWO, named WMGWO. This research work allocated the static weight instead of dynamic weights to the alpha, beta and delta search agents. Hence, the above limitations and drawbacks motivate the researcher, to propose another novel variant of GWO.

In brief, our main contributions are as follows:

- The position update equation has been modified to enhance the hunting behavior of grey wolves in order to propose a novel algorithm of GWO to overcome the above limitations and shortcomings of the basic GWO and its very recant algorithms.
- In addition, the exploitation equation is adopted [35] to enhance the encircling and attacking mechanisms.
- This research work considers 33 mathematical benchmark functions from two different datasets to examine the effectiveness and justify the robustness of the proposed algorithm.
- The results are compared against various state-of-the-art meta-heuristics optimization algorithms: MFO [17], WOA [21], PSO [27], GWO [34], mGWO [35], MVGWO [36] and WMGWO [38].
- Furthermore, the proposed algorithm has been applied to optimize the weights and biases to train the MLP solving real-world classification problems considering five datasets and the results were compared with those of several well-known meta-heuristic trainers that eventually offer salutary influence against unknown search space.

1.1 Roadmap

The remaining sections of this paper are organized in the following way. The MFO, WOA, PSO and GWO and their many recent variants are discussed in Section 2 that addresses the limitations and shortcomings associated with grey wolf optimizers and their very recant variants, which eventually motivates to propose another enhanced algorithm of GWO. Section 3 describes the key functionality of basic GWO and a comprehensive discussion about the suggested novel algorithm. Section 4 discusses the performance results and experimental analysis of the proposed algorithm against state-of-the-art meta-heuristic algorithms. The proposed algorithm employed with a multi-layer perceptron is presented in Section 5. Finally, Section 6 concludes this research work and provides a future research direction.

2. Literature Review

This section elaborates a comprehensive discussion regarding various state-of-the-art meta-heuristic optimization algorithms. The MFO, WOA, PSO, GWO and very recent algorithms of GWO will be the key focus in this study; however, the applied potential applications also will be extensively discussed.

The Moth-Flame Optimization (MFO) [17] is the nature-inspired optimization algorithm that was proposed by Mirjalili in 2015. The transverse orientation is the inspiration to introduce this algorithm. The computational cost of MFO is O(t*n2 + t*n*d), in which 't' refers to the maximum number of iterations, 'n' indicates the number of moths and 'd' is the number of variables. This algorithm is tested on 29 benchmark functions and seven real-engineering problems. The effectiveness of the proposed algorithm is validated using the comparison of results against PSO, GSA, BA, FPA, SMS, FA and GA. Consequently, this algorithm was able to outperform the comparative algorithms against the majority of test functions. The other very recent variants of MFO are proposed in [18]-[20].

The Whale Optimization Algorithm (WOA) [21] is the most recent meta-heuristic optimization algorithm. Mirjalili and Lewis proposed this algorithm in 2016. This proposed algorithm is another nature-inspired algorithm and is motivated by the bubble-net hunting strategy of humpback whales. The whale is a mammal, implying that whales provide milk for their children and are recognized as giant animals on earth. To validate the performance of this algorithm, it is benchmarked on 29 optimization test functions and six engineering-design problems. The welded beam design, tension/compression spring design, pressure vessel design and other three bar truss design problems are considered to be solved using this algorithm. In bar truss design, 52-bar truss design, 25-bar truss design and 15-bar truss design are taken into account. The comparative analysis of results has validated the effectiveness of WOA against PSO, GSA, DE and CMA-ES / FEP. The Binary WOA [25] and another very recently enhanced algorithm of WOA with the map-reduced application are proposed in [26].

The Particle Swarm Optimization (PSO) [27] is a Swarm Intelligence (SI) meta-heuristic optimization algorithm. It was the first social behavior-based algorithm proposed by Kennedy and Eberhart in 1995. The algorithm mimics the social intelligence of bird flocking and fish schooling. The practical execution of the PSO algorithm starts with random solutions (called initial population), which are optimized over the course of iterations using PBEST (Personal best) and GBEST (Global best) parameters. The velocity and position vectors are the mathematical parameters, whereas inertia weight, the cognitive component and social components are the tuning parameters of this algorithm. The several other latest improved algorithms of PSO named UPSO [28] and population size in PSO are referenced in [29]. In addition, Alshdaifat and Bataineh [30] enhanced the PSO, named improved PSO and further employed it with Chebyshev distribution (which defines the search space for IPSO) for optimizing and thinning of the planar array.

Mirjalili et al. [34] developed, theoretically defined and programmatically implemented the Grey Wolf Optimizer (GWO) in 2014. The GWO is a genuinely emerging meta-heuristic optimization algorithm in the literature. The grey wolves are found in Eurasia and North America called *Canis lupus*. The grey wolves' social structure and hunting mechanism were cited as influences for this approach. Figure 2 depicts the dominant social hierarchy. According to Figure 2, the pack's all grey wolves are categorized into four categories corresponding to their specific dominancy and pursuit role. The figure illustrates that the alpha wolf is at the top of the dominant social hierarchy. This wolf is the pack's leader and is referred to as the manager of the pack. The beta wolf is the pack discipliner and the alpha's counselor who endures the next step down. On the social hierarchy's third level, the delta wolf is located that is sentinel, hunter, advisor to beta and caretaker to the pack. As the group's helpers and babysitters, the omega wolves are left. The specific types' hunting method incorporates three pivotal steps that refer to additional motivation rather than the social hierarchy. As a result, the primary phases are to seek the prey and annoy the prey until it gives up or stops and then attack the target in the end. The mathematical model regarding the above lemma has been formulated to introduce the GWO algorithm. In addition, on 29 test functions and four engineering-design real-world's problems, performance has been tested and certified. In addition, the PSO, GSA, DE and FEP algorithms were compared to the GWO in order to verify the findings of this work. Consequently, the experimental results determine the effectiveness of this algorithm that produces very competitive results. However, this algorithm may be stuck in local optima that refer to its main drawback. In addition, the poor solution accuracy and sluggish convergence rate address it more challenging for further improvement.

Mittal et al. [35] proposed an advanced variant of GWO; namely, modified Grey Wolf Optimizer (mGWO), in order to maintain pertinent equilibrium among exploration and exploitation of the search space. In order to accomplish the focus on objective, they employed the exponential decay function instead of the linear function pertaining to the constant vector $\vec{a}$ in the enforcement of the standard algorithm of GWO. The exponential function devotes seventy and thirty percent iterations to exploration and exploitation, respectively. In comparison, in order to accomplish the linear function, half of the iteration; i.e., the first fifty percent is dedicated to the exploration and the remaining fifty percent is committed to exploitation. Note that multimodal, unimodal, composite and fixed-dimension multimodal benchmark functions are imposed to illuminate the proposed variant's performance, considering that standard deviation and average are statistical parameters of appraisal with 3000 iterations and 30 number of population. In addition, the selection of cluster heads in wireless sensor networks is often regarded as a relatively well-known real-world application. To sum up, the mGWO outperforms occasionally or has very competitive outcomes compared to other meta-heuristic algorithms and original GWO. However, the modification is attempted in only controlling parameter $\vec{a}$, hence further improvement may be possible.

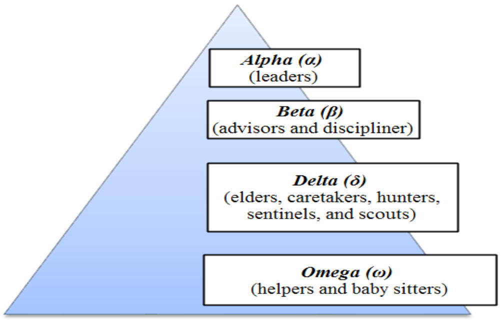


Figure 2. Social dominant hierarchy of grey wolves (dominance decreases from top down) [34].

The MVGWO is another variant of GWO proposed by Singh [36] in 2018. There was a biological improvement in the social hierarchy, in which five groups are formed for the total population of wolves. Thus, the proposed algorithm extends the social order up to five levels after including gamma wolves at the third level from the top. According to biological theory, the top four levels' wolves (i.e., delta, gamma, beta and alpha) participated in the hunting and finding of prey. In order to carry out mathematical implementation, encircling behavior and position update equation has been modified in terms of basic GWO to improve the results. The average of the best four solutions is utilized to update the remaining solutions over the course of iteration and find the most optimal solution at the end of the last iteration. The obtained results show that this algorithm provides very competitive results concerning PSO, GWO and modified mean GWO [37]. In addition, the newly modified algorithm performs considerably better to tackle the cantilever beam design problem and sine dataset. This research work improved only the biological structure of the grey wolves, but it has not led to significant enhancement in the mathematical model accordingly.

Kumar et al. [38] proposed another variant of GWO, named WMGWO, in 2019. There are four levels of social hierarchy. The proposed variant employed a weighted mean factor instead of uniform distribution in order to update the omegas. It suggested 54, 30 and 16 percent weightage to alpha, beta and delta wolves (i.e., search agents or solution), respectively. The performance of the proposed variant is validated after comparing the results with these of GWO, mGWO and MVGWO. The outcomes of this algorithm are very competitive against comparative algorithms. In addition, this algorithm performs very well on the function approximation and classification datasets. This research work utilized static weight instead of dynamic weights to the alpha, beta and delta search agents, which pointed to the limitation of this work. To improve the diversity of GWO, Abed-alguni and Barhoush [39] introduced a distributed approach of GWO by organizing its population using the island model. Furthermore, the proposed algorithm was tested on thirty CEC 2014 functions and fifteen standard test functions that provide competitive performance against other tested algorithms.

3. Proposed Work

The basic Grey Wolf Optimizer (GWO) and the novel proposed algorithm (Weighted Grey Wolf Optimizer with Improved Convergence Rate (WGWOIC)) will be discussed in this section.

3.1 Grey Wolf Optimizer

As we discussed in the preceding section, the grey wolf optimizer [34] mimics the social dominant hierarchy of grey wolves and their social hunting mechanism inspires this algorithm. According to the biological theory of grey wolves, hunting is attempted exclusively *via* the top three levels' wolves (i.e., alpha, beta and delta). The alpha wolf is the dominant wolf, then beta and then delta. All other wolves dominate the omega wolves in the pack. The social dominant hierarchy of grey wolves has been depicted in Figure 2. In order to solve any optimization problem using GWO, the process commences with the random population (also designated as random search agents or random solutions). Subsequently, this algorithm's workflow would originate. For the mathematical model of GWO, the best three solutions obtained so far are saved and remaining solutions (including the omegas) are adapted based on the above three best search agents according to Equation 1.

$$\vec{X}(\text{Curr\_iter}+1)=\frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3} \tag{1}$$

According to Equation 1, Curr_iter refers to the current iteration value that linearly increases to Max_iter (maximum number of iterations). The vectors $\overrightarrow{X_1}, \overrightarrow{X_2}\ and\ \overrightarrow{X_3}$ indicate the updated best positions (solutions) of alpha, beta and delta wolves, respectively. These three wolves update their positions according to the prey's position that is formulated by Equation 2. However, there is no idea regarding the location of the prey (optimum) in an abstract search space. Therefore, alpha, beta and delta determine the prey's probable location by taking the mean of their positions. The mathematical result of the mean is represented by $\vec{X}(Cur\_iter+1)$, on which basis the remaining wolves (omegas) update their positions. In order to determine the preceding best three positions, $\overrightarrow{D_\alpha}, \overrightarrow{D_\beta}\ and\ \overrightarrow{D_\delta}$ vectors have to be figured out using Equation 2, whereas the vector $\vec{D}$ indicates the distance from wolf to prey.

$$\left.\begin{array}{l}\overrightarrow{D_\alpha}=\left|\overrightarrow{C_1}.\overrightarrow{X_\alpha}-\vec{X}\right|, \overrightarrow{X_1}=\overrightarrow{X_\alpha}-\overrightarrow{A_1}.(\overrightarrow{D_\alpha})\\ \overrightarrow{D_\beta}=\left|\overrightarrow{C_2}.\overrightarrow{X_\beta}-\vec{X}\right|, \overrightarrow{X_2}=\overrightarrow{X_\beta}-\overrightarrow{A_2}.(\overrightarrow{D_\beta})\\ \overrightarrow{D_\delta}=\left|\overrightarrow{C_3}.\overrightarrow{X_\delta}-\vec{X}\right|, \overrightarrow{X_3}=\overrightarrow{X_\delta}-\overrightarrow{A_3}.(\overrightarrow{D_\delta})\end{array}\right\} \tag{2}$$

The vectors $\vec{A}$ and $\vec{C}$ are called controlling parameters that provide equilibrium among exploration and exploitation for the abstract search space. The value of these controlling parameters is calculated by using Equation 3.

$$\vec{A}=2*\vec{a}.\overrightarrow{r_1}-\vec{a}, \vec{C}=2.\overrightarrow{r_2} \tag{3}$$

The vectors $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ are known as random vectors between [0, 1]; i.e., the computational value of $\vec{C}$ would be found between [0, 2] and the value of $\vec{A}$ between [-2, 2]. The vector $\vec{C}$ is deliberately used to provide a random value throughout the algorithm, which offers randomness to the GWO algorithm. In contrast, if vector $\vec{A}$'s mathematical weight is found in the interval [-1, 1], it supports the algorithm to converge toward the most optimal solution; otherwise, it diverges from the current solution in order to find the optimum. In addition to the above vectors, another vector,$\vec{a}$, is also called a controlling parameter which is calculated by Equation 4. The value of this controlling parameter is decreased linearly to 0 over the course of iterations. Therefore, the initial fifty percent values oblige exploration and the remaining fifty percent is devoted to exploitation.

$$\vec{a}=2*\left(1-\frac{\text{Curr\_iter}}{\text{Max\_iter}}\right) \tag{4}$$

The GWO algorithm was tested on twenty-nine benchmark functions and the obtained results were analyzed to check the performance. In order to investigate the performance, the comparative analysis against other well-known algorithms asserts that the GWO algorithm encounters some limitations and challenges, such as stagnation in local optima, low solving accuracy and slow convergence rate. Hence, these limitations and challenges encourage proposing another algorithm of GWO. Therefore, we have introduced another algorithm of GWO titled Weighted Grey Wolf Optimizer with Improved Convergence Rate (WGWOIC), as discussed in the coming sub-section.

3.2 Proposed WGWOIC Algorithm

As we have discussed earlier, the basic GWO and its very recently developed algorithms have some limitations and shortcomings. Therefore, we have proposed another algorithm of GWO to overcome these limitations and resolve the deficiencies. The proposed algorithm is designated as Weighted Grey Wolf Optimizer with Improved Convergence Rate (WGWOIC). Therefore, we have modified the hunting (position update equation) and attacking (exploitation equation) behavior of grey wolves to introduce this algorithm that indicates this research work's novelty. In addition to the above contribution, we have practiced two different benchmark datasets to determine the effectiveness of performance in terms of strength and robustness of the proposed algorithm against other comparative state-of-the-art optimization algorithms.

According to the advancement in the biological theory of grey wolves to introduce this novel algorithm, the wolves hunt the prey and attack the target with a different mechanism. For the mathematical model, the position update equation (hunting) and exploitation equation (attacking the prey) are modified to assist the enhancement of the studied algorithm in the literature. Therefore, Equation 1 obliges hunting the prey; i.e., it is known as the hunting equation or position update equation. In contrast, Equation 4 obliges attacking the target; i.e., an attacking equation or exploitation equation. In order to accomplish our objective, we will apply Equation 5 instead of Equation 1 and Equation 7 instead of Equation 4.

$$\vec{X}(\mathrm{Curr\_iter}+1) = (\vec{w_1}+\vec{r}).\vec{X_1} + (\vec{w_2}.\vec{X_2} - \vec{w_3}.\vec{X_3}) \tag{5}$$

The vector $\vec{r}$ is a random weight granted to the alpha solution computed as half of the random value over every course of iteration. This random weight obliges the computed solution to be slightly tilted toward the alpha, because the alpha is likely to be considered the best solution to the problem. It is brightly founded from the comprehensive literature that alpha, beta and delta wolves contain an excellent knowledge of prey (solution) against remaining wolves. Simultaneously, the alpha wolf comprises the most optimal solution among the population, then beta and then delta. We have utilized this lemma and accordingly updated the hunting and attacking mechanisms of the wolves. Therefore, the alpha wolf is not delivering decisions alone for hunting, staying and other pack activities. Thus, the alpha wolf has also taken the advice of beta and delta. For the mathematical model, these are the three best solutions among all solutions obtained so far. In this research work, we are giving extra weight to the alpha solution to update the solutions; alongside, we have also added a weighted difference of beta and delta solutions. Consequently, the updated solutions are slightly tilted toward the alpha solution and the weighted difference provides diversity to the proposed algorithm. In Equation 5, the vectors $\vec{w_1}$, $\vec{w_2}$ and $\vec{w_3}$ are considered as influence factors that provide the influence weighted to alpha, beta and delta solutions, respectively, during every course of iteration. These factors are mathematically formulated by Equation 6.

$$\vec{w_1} = \frac{\vec{r_1}}{\vec{r_1}+\vec{r_2}+\vec{r_3}}, \vec{w_2} = \frac{\vec{r_2}}{\vec{r_1}+\vec{r_2}+\vec{r_3}}, \vec{w_3} = \frac{\vec{r_3}}{\vec{r_1}+\vec{r_2}+\vec{r_3}} \tag{6}$$

It is clear that the total sum value of these influence factors is '1' ($\vec{w_1}+\vec{w_2}+\vec{w_3}=1$). In order to calculate the above influence vectors, the vectors $\vec{r_1}$, $\vec{r_2}$ and $\vec{r_3}$ are utilized with the random weights in [0, 1] to provide the randomness to the proposed algorithm; not only the initial iteration, even till the final iteration. The influence factors' values will assist to find the most optimal location of the prey after putting these values in Equation 5.

In addition to the above contribution, we have adopted Equation 7 from reference [35] in order to compute the value of vector $\vec{a}$. Now, we will use Equation 7 instead of Equation 4 to enhance the exploitation of the recommended algorithm.

$$\vec{a} = 2 * \left(1 - \frac{\mathrm{Curr\_iter}^2}{\mathrm{Max\_iter}^2}\right) \tag{7}$$

From the comprehensive literature, the grey wolves accomplish their hunt by attacking the prey when it stops moving. In a mathematical model, the vector $\vec{a}$ performs this task. The value of this vector is decreased exponentially from '2' to '0' over the course of iterations. The initial seventy percent values of this vector that decrease slowly oblige extensive exploration of the search space, whereas the remaining thirty percent components that decrease quickly oblige fast exploitation toward the solution. This vector eventually maintains good equilibrium among exploration and exploitation of the abstract search space; i.e., it provides fast convergence and more diversity. This vector is also utilized to decrease randomness; implying that the WGWOIC algorithm would be converging toward the final solution. The pseudo-code of the proposed algorithm is depicted in Figure 3.

4. RESULTS AND DISCUSSION

In this section, we have benchmarked the WGWOIC algorithm on 33 fairly well-known numerical benchmark test functions. The first twenty-three classical functions are included from

Initialize the population of grey wolves (Search Agents) $\overrightarrow{X_k}$ (k = 1, 2, 3... n)
Initialize controlling parameters $\vec{a}$, $\vec{A}$, and $\vec{C}$
Compute the fitness of each search agent
$\overrightarrow{X_\alpha}$ = the most fittest solution from all search agents
$\overrightarrow{X_\beta}$ = the second fittest solution from all search agents
$\overrightarrow{X_\delta}$ = the third fittest solution from all search agents
while (Curr_iter <= Max_iter) **do**
  for each search agent ($\overrightarrow{X_k}$) **do**
    Update the position of current $\overrightarrow{X_k}$ using equation **(5)**
  end for
  Update the value of controlling parameters $\vec{a}$ (using equation **7**), $\vec{A}$, and $\vec{C}$
  Compute the fitness of all search agents
  Update the value of $\overrightarrow{X_\alpha}$, $\overrightarrow{X_\beta}$, and $\overrightarrow{X_\delta}$
  Curr_iter = Curr_iter + 1
end while
return $\overrightarrow{X_\alpha}$

Figure 3. Pseudo-code of the WGWOIC algorithm.

CEC 2005, which many researchers utilized in their work. These benchmarked functions are minimization functions and categorized into three groups: unimodal (first seven functions), multimodal (following six functions) and fixed-dimension multimodal (last ten functions) benchmark functions. However, these benchmark test functions have different dimensions and boundary ranges that indicate the main challenges for the proposed algorithm in order to optimize the above test functions.

Subsequently, the remaining ten test functions are modern single objective minimization functions (CEC01 to CEC10) included from CEC-C06 2019. These test functions are scalable and known as the 100-Digit Challenge. The functions from CEC04 to CEC10 are rotated and shifted, whereas the functions from CEC01 to CEC03 are not. The dimensionality of CEC 01, CEC 02 and CEC 03 test functions is 9, 16 and 18, whereas the boundary range is [-8192, 8192], [-16384, 16384] and [-4, 4], respectively. In contrast, the dimensionality of the remaining functions (from CEC04 to CEC10) is the same, each with 10-dimensional in [−100, 100] boundary range. The detailed discussion about the first twenty-three test functions is in reference [34] and that of the remaining ten functions of CEC-C06 2019 are in [40]. At the same time, the programming implementation of CEC-C06 2019 functions is performed in reference [41]. Interestingly, this research work considers the above two different benchmark functions in order to evaluate the performance with the effect of the proposed algorithm that validates the strength and confirms the robustness of the WGWOIC algorithm.

For the mathematical implementations, the population size of the proposed algorithm and other state-of-the-art comparative algorithms is 30. All algorithms are iteratively repeated 500 times over the course of iterations to obtain the most optimal solution in one independent run. Subsequently, all algorithms are repeated 30 separate runs on each benchmark function. The average (mean) value of these 30 independent runs eventually indicates the outcome (optimum global value) to the corresponding benchmark function. The other statistical variables (Best, Worst and Std.) are also utilized to validate the effectiveness of the proposed algorithm's outcomes against the studied comparative well-known algorithms. The best statistical variable shows the minimum value throughout the 30 independent runs, whereas the worst refers to the maximum value. On the other hand, Std. stands for standard deviation, estimated through 30 separate runs. Therefore, the lowest values represent the optimum values of each statistical variable regarding all algorithms concerning individual functions. The performance of the proposed algorithm is validated against many swarm intelligence-based algorithms, such as Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Modified GWO (mGWO), Modified Variant of GWO (MVGWO) and Weighted Mean GWO (WMGWO). In addition, the proposed algorithm is also compared with two nature-inspired algorithms: Moth-Flame Optimization (MFO) and Whale Optimization Algorithm (WOA). However, the GWO has already been reached with PSO as the swarm intelligence-based algorithm, GSA as the physics-based algorithm and DE, FEP and CMA-ES as the evolutionary algorithms. Table 1 lists the simulation hardware and software environment on which the practical implementation of this work has been conducted.

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 03, September 2021.

Table 1. Experimental environment.

Parameter	Hardware and Software Configuration
Implementation Tool	MATLAB R2017a
Processor	Intel(R) Core(TM) i7-4770 CPU@ 3.40GHz
RAM	12.0 GB
Operating System	64-bit Operating System

Table 2 shows the computational results of unimodal test functions of the WGWOIC algorithm and other comparative state-of-the-art meta-heuristic algorithms. It must be noted that there are seven unimodal functions (F1-F7) which are having single optima. These benchmark functions validate the effectiveness of the exploitation performance of the proposed algorithm. This algorithm highly outperforms on F1, F3, F4 and F7 functions against well-known meta-heuristic comparative algorithms and provides very competitive results on the remaining unimodal functions.

Table 2. Results of unimodal benchmark functions.

Functions	Criteria	PSO	MFO	WOA	GWO	mGWO	MVGWO	WMGWO	WGWOIC
F1	Best	1.63886E-05	0.158221405	2.79122E-88	1.70234E-29	4.48066E-39	9.38977E-22	2.12829E-39	1.78217E-85
	Worst	0.000772625	20002.75312	3.06783E-69	3.2754E-27	3.39035E-34	5.58183E-20	2.9993E-36	5.13144E-81
	Mean	0.00015484	2670.895162	1.10514E-70	8.15964E-28	1.4291E-35	7.52036E-21	1.83367E-37	3.54905E-82
	Std	0.000155845	6395.635248	5.59769E-70	8.6975E-28	6.17581E-35	1.09375E-20	5.53603E-37	9.94418E-82
F2	Best	0.005416498	0.15259518	1.87787E-58	4.31376E-17	5.76217E-23	1.08145E-13	1.64269E-23	1.41081E-46
	Worst	0.122725197	60.04019074	2.18917E-50	4.50204E-16	7.94649E-21	2.57471E-12	1.02317E-21	5.6575E-44
	Mean	0.034715846	35.09009906	1.03354E-51	1.18525E-16	1.02612E-21	9.93847E-13	1.19183E-22	9.82965E-45
	Std	0.028536391	18.67345722	4.0533E-51	8.06393E-17	1.58156E-21	5.24122E-13	1.90205E-22	1.16457E-44
F3	Best	32.15428241	2669.898671	17647.46009	4.91195E-09	2.03961E-10	1.72282E-06	6.41861E-13	1.91335E-61
	Worst	184.6971148	43943.76011	72344.95817	0.001286365	5.57145E-06	0.008222075	2.37055E-06	2.20352E-55
	Mean	95.15931899	18590.53167	45553.67157	4.96083E-05	2.44554E-07	0.000462987	1.46535E-07	1.04907E-56
	Std	33.71188464	10843.57994	12483.87229	0.000234343	1.0163E-06	0.001481328	4.72723E-07	4.21886E-56
F4	Best	0.641915093	54.8705224	4.669865609	5.87068E-08	8.74413E-11	4.03672E-06	6.91339E-11	2.03365E-35
	Worst	1.787857145	81.0563014	89.38407148	2.28106E-06	7.62111E-09	0.000176699	6.02427E-09	1.21729E-31
	Mean	1.156085088	69.21414183	45.72716132	6.322E-07	1.66461E-09	3.74597E-05	9.19472E-10	8.08294E-33
	Std	0.319797238	6.824054473	27.6934708	5.65885E-07	1.91586E-09	3.84676E-05	1.25221E-09	2.54366E-32
F5	Best	27.20201917	309.3363674	27.24769259	25.74622807	26.05555255	25.9950154	26.04231519	27.24591773
	Worst	265.766625	80033040.29	28.76765556	28.55897177	28.72223185	28.80989169	28.73810387	28.90556925
	Mean	76.73511711	2686931.035	28.07688863	27.05204892	26.94454281	27.31357229	26.94364717	28.02425226
	Std	51.71636381	14608392.47	0.445358351	0.753119474	0.656530374	0.883426624	0.670585955	0.512886109
F6	Best	7.68491E-06	0.542083492	0.112093851	0.23186088	0.243501503	0.45882678	0.248906786	3.707888307
	Worst	0.00657646	10106.08889	0.959981164	1.755332071	1.255444247	3.264675357	1.507850263	4.764877793
	Mean	0.000365663	1013.161376	0.407078881	0.778026799	0.592550125	1.342234409	0.832794353	4.3288351
	Std	0.001186222	3081.722471	0.240597427	0.398519267	0.256924289	0.572752401	0.314861247	0.332820474
F7	Best	0.064203293	0.085215057	0.000116117	0.00043649	0.000379606	0.001117032	0.000518688	2.96069E-05
	Worst	0.302894078	29.62285404	0.012456871	0.006311453	0.003197678	0.005411433	0.003334457	0.001077777
	Mean	0.19398366	3.463092413	0.002654718	0.002235133	0.001484414	0.002787913	0.001564218	0.000380318
	Std.	0.064673982	6.895509972	0.00282493	0.001183375	0.000822758	0.001058371	0.000753789	0.000301542

Table 3. Results of multimodal benchmark functions.

Functions	Criteria	PSO	MFO	WOA	GWO	mGWO	MVGWO	WMGWO	WGWOIC
F8	Best	-6740.805369	-9602.385548	-12569.45849	-7442.734249	-7291.978467	-7600.083886	-7344.16222	-4427.955563
	Worst	-2871.327832	-6870.808247	-7093.47263	-3480.901369	-2868.810357	-4831.305124	-3473.562341	-3093.946616
	Mean	-4541.466559	-8394.321875	-10354.0438	-5966.223462	-5595.560325	-5733.209196	-6029.573094	-3631.225745
	Std	1141.174277	676.4760095	1819.704826	835.1905512	1210.17144	721.447048	954.272808	343.1655987
F9	Best	36.32221174	100.7700387	0	5.68434E-14	0	9.01537E-11	0	0
	Worst	76.62989803	282.9948352	5.68434E-14	14.94139474	11.79543628	18.40187825	6.68404661	0
	Mean	54.06197808	162.7394495	3.78956E-15	2.626204088	0.546080577	9.184836132	0.39661479	0
	Std	9.810297529	43.58301536	1.44216E-14	3.749712597	2.28357397	4.675340967	1.52165200	0
F10	Best	0.002695217	0.708412743	8.88178E-16	7.54952E-14	1.15463E-14	5.21627E-12	7.99361E-	4.44089E-15
	Worst	1.360625922	19.96001708	7.99361E-15	1.46549E-13	3.64153E-14	3.58549E-11	2.22045E-	7.99361E-15
	Mean	0.185936458	14.7958024	4.79618E-15	9.64562E-14	2.19676E-14	1.64674E-11	1.5928E-14	4.79616E-15
	Std	0.445780516	7.348608771	2.696E-15	1.5843E-14	5.89582E-15	8.36597E-12	3.5751E-15	1.08403E-15
F11	Best	4.39798E-07	0.691226662	0	0	0	0	0	0
	Worst	0.039404049	90.93765307	0.216365287	0.012173631	0.039658157	0.032276084	0.01695686	0
	Mean	0.007400127	9.970233113	0.016906773	0.001148008	0.003918697	0.01048135	0.00056522	0
	Std	0.009604463	27.38383546	0.05315222	0.00351309	0.010120281	0.011535548	0.00309588	0
F12	Best	1.604334573	13619103.21	0.104111454	0.962482184	0.21421963	1.688354746	0.17418128	0.380581756
	Worst	10.35348117	325818040.9	7.096752496	5.888719656	2.945625079	12.85348765	2.38358817	1.169842467
	Mean	4.422974266	118292981.9	0.506732063	3.309574426	1.071433625	4.514060426	0.91067705	0.721041352
	Std	2.343229343	83682241.01	1.27166015	1.191318431	0.726975915	2.312196846	0.52318395	0.163739768
F13	Best	4.830503959	79928426.13	0.525800914	5.383776068	2.32031093	9.493814301	1.79363076	2.545021523
	Worst	41.92960075	585026133.6	2.676185434	26.42349085	11.53322581	37.16601697	7.33130944	3.55702194
	Mean	17.38999252	237951829.1	1.400475334	10.32643742	4.515938171	19.2946504	3.81086210	2.887597749
	Std.	9.929256026	113092504.8	0.465939596	4.177855191	1.866567399	6.642529041	1.43062099	0.193337137

In contrast to the unimodal functions, there are six multimodal functions (F8-F13) which are having a massive number of local optima. These functions are used to validate the effectiveness of the exploration performance of the WGWOIC algorithm against comparative algorithms. Table 3 lists the computational results of multimodal benchmark functions. The proposed algorithm is considerably better than all comparative algorithms on F9, F10 and F11 functions and is very competitive an the remaining multimodal functions.

Table 4. Results of fixed-dimension multimodal benchmark functions.

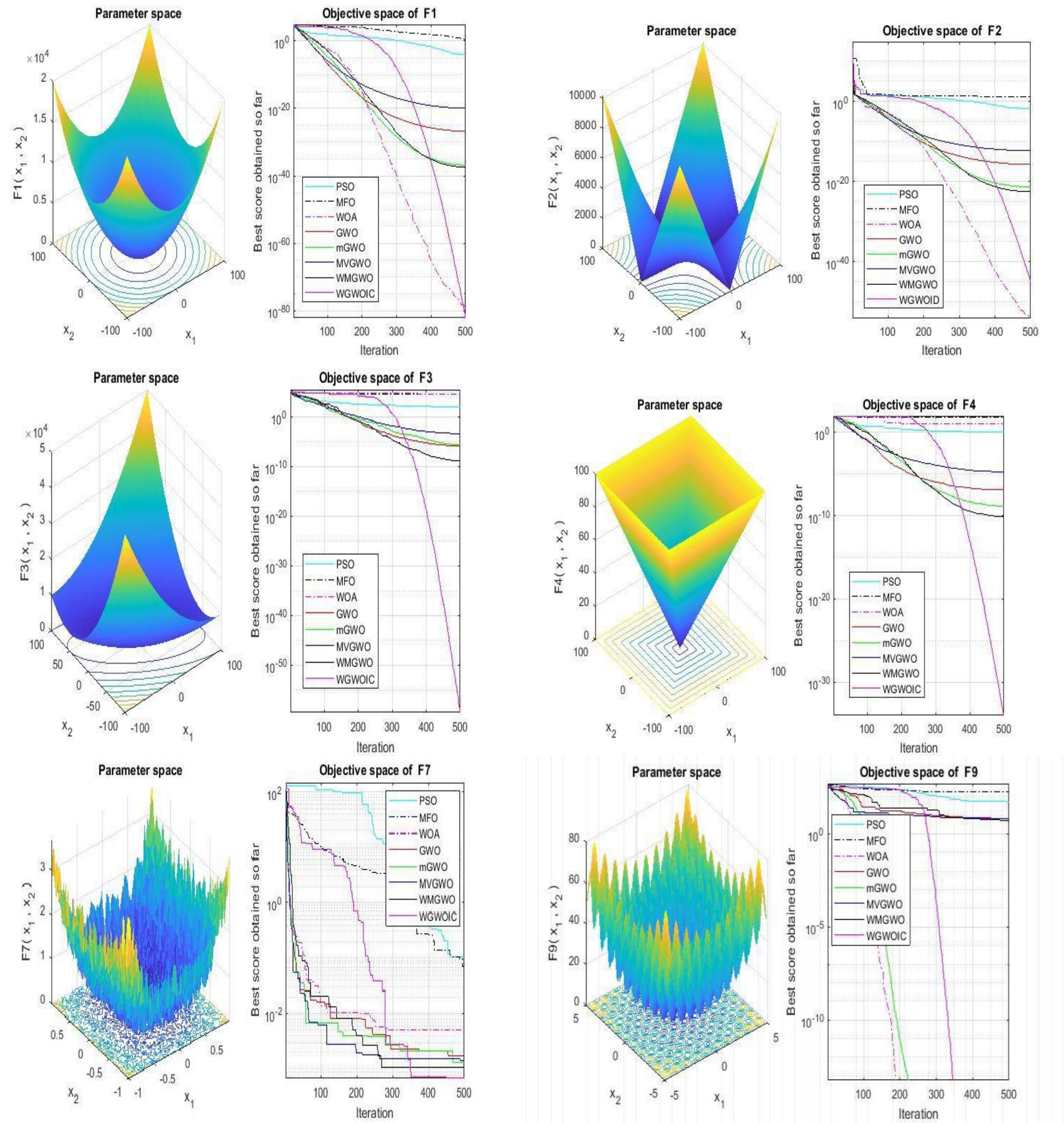
Functions	Criteria	PSO	MFO	WOA	GWO	mGWO	MVGWO	WMGWO	WGWOIC
F14	Best	0.998003838	0.998003838	0.998003838	0.998003838	0.998003838	0.998003838	0.998003838	0.998004058
	Worst	10.76318067	8.840835963	10.76318067	12.67050581	10.76318067	16.44090731	10.76318067	10.76318067
	Mean	3.55705999	2.413759985	2.408871047	4.230081157	3.093536966	6.076285786	2.442656154	2.04960213
	Std	2.963055118	1.994557984	2.53814438	4.11037804	3.210295799	4.921621581	2.458950183	2.499411434
F15	Best	0.000626681	0.000443089	0.000308353	0.000307496	0.000307649	0.000307834	0.000307991	0.000320747
	Worst	0.001594171	0.020363339	0.001606437	0.020363361	0.020363369	0.020363377	0.020363351	0.001331427
	Mean	0.000898827	0.001628901	0.000784627	0.004401317	0.005094642	0.003063687	0.001827055	0.0005669
	Std	0.000174434	0.003564441	0.000391673	0.008119157	0.008570346	0.006901842	0.005048324	0.000300483
F16	Best	-1.031628445	-1.031628453	-1.031628452	-1.031626947	-1.031628208	-1.031628131	-1.031626365	-1.031601951
	Worst	-1.02451432	-0.215463311	-0.906007779	-0.999980519	-0.999025583	-0.988939132	-0.9988901	-1.011447444
	Mean	-1.031280141	-1.004375325	-1.022052627	-1.029637815	-1.027771295	-1.028035569	-1.030414104	-1.028589411
	Std	0.001347638	0.149001867	0.028279174	0.007355832	0.009660761	0.009712765	0.005957786	0.004563169
F17	Best	0.397887358	0.397887358	0.397889389	0.3978963	0.397901342	0.397896281	0.39789264	0.398513531
	Worst	0.397903535	1.943140663	3.145500095	3.491223558	0.406921043	0.400650828	4.97391737	0.729053842
	Mean	0.397887912	0.505199511	0.691432704	0.502478431	0.400023301	0.39844679	0.557064981	0.442742179
	Std	2.95103E-06	0.391417638	0.557575662	0.564515144	0.00261759	0.00073983	0.83432747	0.065486544
F18	Best	3	3	3.000000147	3.000000205	3.000000115	3.00000016	3.000000194	3.00000061
	Worst	3	3	3.002647476	84.00001234	84.00012503	3.000177318	3.00007862	3.000237848
	Mean	3	3	3.000272395	5.700037282	5.700021785	3.000055953	3.000019692	3.000034633
	Std	1.96537E-15	2.15201E-15	0.000568143	14.78850434	14.78852855	5.11566E-05	2.03176E-05	4.77781E-05
F19	Best	-3.862782147	-3.862782148	-3.862104082	-3.862771483	-3.862606236	-3.862651908	-3.862475174	-3.862328044
	Worst	-3.862781313	-3.089764162	-2.773780759	-1.000795465	-3.810140407	-1.000783963	-3.814269526	-3.714218991
	Mean	-3.862782021	-3.824806378	-3.698618476	-3.761584254	-3.855949456	-3.761104497	-3.852313926	-3.809338982
	Std	2.39265E-07	0.152632921	0.277569519	0.521471286	0.011160136	0.521408327	0.012798558	0.03926783
F20	Best	-3.314418487	-3.06809545	-3.039630101	-3.301659308	-3.304775986	-3.288118188	-3.317169523	-3.007106385
	Worst	-0.909982661	-0.373612111	-0.43072469	-1.746400274	-1.780786493	-1.065166152	-1.769534999	-0.956158259
	Mean	-2.771059131	-2.042624029	-1.538709099	-2.862052074	-2.970472608	-2.770256553	-2.905743952	-2.132615795
	Std	0.60116875	0.867003605	0.751171691	0.424377537	0.325042763	0.546032058	0.449549744	0.667529105
F21	Best	-10.15319968	-10.15319968	-10.1508081	-10.15276796	-10.15130887	-10.15295394	-10.15211863	-4.953161739
	Worst	-2.630471668	-2.630471668	-2.627163857	-5.055188892	-2.6809166	-2.630342057	-5.054989528	-0.878075038
	Mean	-7.594555698	-6.545794	-8.674089067	-9.644703023	-9.221455628	-8.981010629	-9.130220118	-4.15674959
	Std	3.197162146	3.139790866	2.682803809	1.545931042	2.137959972	2.693260569	2.063434258	1.339198363
F22	Best	-10.40294057	-10.40294057	-10.40111333	-10.40280511	-10.40123363	-10.40284313	-10.40234302	-7.154492635
	Worst	-2.751933564	-2.751933564	-1.836472833	-1.837523021	-10.38840976	-10.39856825	-2.765762625	-2.599856113
	Mean	-9.317729361	-8.186183479	-7.787194844	-10.11578943	-10.39609553	-10.40120076	-10.14113624	-4.635777438
	Std	2.507392867	3.238901603	2.913737763	1.563515286	0.003127188	0.001018144	1.392993271	0.715724367
F23	Best	-10.53640982	-10.53640982	-10.53595185	-10.53549351	-10.53507373	-10.53612511	-10.53434444	-7.527868551
	Worst	-2.421734027	-2.421734027	-2.41789726	-2.421664232	-10.52035255	-2.421726384	-5.128106219	-0.942178182
	Mean	-9.109575858	-7.255347769	-6.655315468	-9.54281558	-10.52853524	-10.26427221	-10.16837401	-4.564613794
	Std.	2.803435329	3.659985998	3.355458636	2.607522456	0.004334215	1.481220641	1.370062816	0.991545922

On the other hand, the last ten functions (F14-F23) of the first benchmark dataset are known as fixed-dimension multimodal functions that validate the effectiveness of the exploration performance and the avoidance of local optima. Hence, the key focus is on global optima along with their exploitation performance for their convergence rate. Table 4 lists the computational results of fixed-dimension multimodal functions. The proposed algorithm outperforms all comparative well-known meta-heuristic algorithms an F14 and F15 functions and provides very competitive developments an remaining fixed-dimension multimodal functions. Hence, the proposed algorithm is validated and justified for global optimum and good convergence rate.

Table 5. Results of CEC-C06 2019 benchmark test functions.

Functions	Criteria	PSO	MFO	WOA	GWO	mGWO	MVGWO	WMGWO	WGWOIC
CEC 01	Best	84016047950	264438535.5	10875873.0	63212.9355	517904.745	678610.799	45647.22929	41835.65672
	Worst	5.74709E+12	1.13967E+11	1.34126E+1	206372812	452193546	659991656.	7947145755	40370564.89
	Mean	2.04372E+12	18693542941	3082400885	197236612	467463912.	101803359.	512481102.5	1397974.063
	Std	1.26038E+12	29133114337	3328126714	464739296.	929119650.	158517407.	1455917657	7360752.576
CEC 02	Best	8997.812591	18.99160195	17.4052621	17.3462161	17.3541055	17.3471554	17.35019871	17.45346937
	Worst	24672.5164	165.4226319	18.9502206	17.7073142	17.6953701	18.7855508	17.37192198	17.94370798
	Mean	15385.68872	52.60855145	18.0426850	17.3885923	17.3974980	17.5968274	17.3617251	17.67726882
	Std	3959.693873	30.69601113	0.41002520	0.09915725	0.09145850	0.37644568	0.005624935	0.111560162
CEC 03	Best	12.70240436	12.70240431	12.7024071	12.7024043	12.7024044	12.7024042	12.70240438	12.70242326
	Worst	12.704906	12.70254569	12.7046566	12.7047524	12.7042670	12.7059078	12.70490424	12.70310611
	Mean	12.70254677	12.70243193	12.7025459	12.7025361	12.7025086	12.7026334	12.70257894	12.7025193
	Std	0.000525338	3.65032E-05	0.00040738	0.00047321	0.00037781	0.00072414	0.000631479	0.000159984

CEC 04	Best	4.974795285	13.92941682	91.0796608	25.4822642	31.7801790	16.6507263	32.82018789	807.4234065
	Worst	45.95817019	601.633464	916.872185	1013.16275	2409.03329	2474.53275	2351.156413	4595.461272
	Mean	15.6159303	177.0484372	328.230249	95.6389720	148.620689	245.427925	152.5301426	1926.940239
	Std	8.919899821	202.1005042	175.219281	174.713382	427.382512	616.514640	416.0275936	740.7697138
CEC 05	Best	1.858024186	2.344005101	1.95590679	1.62002275	1.79651098	1.65520471	1.722212455	2.471959211
	Worst	4.812449362	4.161472658	4.12150312	2.14344097	2.10298904	2.375428153	2.325748056	3.885164162
	Mean	3.42182944	3.190601392	3.16522614	1.91598588	1.95687935	1.988312207	1.980947909	2.995941288
	Std	0.701206358	0.441984574	0.46085167	0.10467692	0.08432479	0.166134256	0.122111473	0.306795588
CEC 06	Best	5.298933027	1.165535574	7.60266370	8.98336563	9.05185916	10.02866692	6.086274462	9.75098118
	Worst	11.14441722	10.79436376	10.7467440	12.1446098	12.2400230	12.43691074	12.33565412	12.03766536
	Mean	9.055851559	5.735387803	9.43556745	11.0579200	10.8917927	11.24706021	10.93271595	10.80254285
	Std	1.375294442	2.517629806	0.96060985	0.74307568	0.79118878	0.657555238	1.215991338	0.627645061
CEC 07	Best	-74.10859072	-126.335496	208.750080	11.2883697	122.093624	-26.62196701	158.3134872	494.7379115
	Worst	374.2502338	1252.64433	1168.45006	1076.17473	1095.05153	960.270531	1074.608334	1056.631639
	Mean	150.2608467	452.6023701	660.953618	521.059954	561.461254	379.797949	558.1062293	825.942669
	Std	113.9916614	260.2282114	240.432068	290.169065	296.427841	269.355960	257.1206831	133.0243803
CEC 08	Best	3.743690545	4.33845849	4.90500812	2.98562530	2.66140208	2.62848205	3.247684807	4.818489682
	Worst	6.20863408	6.900537982	6.68769272	6.61444688	6.34279806	6.54948398	6.493956809	6.726253322
	Mean	5.147764502	5.710775289	5.81845279	4.59610904	4.52684407	4.56915741	4.852959507	5.683344627
	Std	0.56888959	0.642495043	0.48673659	0.94864667	0.96882259	0.89918493	0.998069408	0.455802888
CEC 09	Best	2.339280498	2.46217298	2.92218413	3.25728240	2.63124987	2.78707052	3.108344345	6.187666562
	Worst	2.355534157	1121.584371	6.79360902	5.71864927	5.98819892	370.750220	6.115638485	82.13447988
	Mean	2.346186889	39.94266085	4.24822211	4.33404679	4.37809308	16.4262217	4.489645472	27.96418284
	Std	0.004419399	204.2895831	0.82564146	0.83304097	0.90941748	66.9255875	0.785456542	16.20840294
CEC 10	Best	20.04350652	19.99983504	20.0762604	20.2698906	20.3293946	20.3073990	20.2193504	15.99976137
	Worst	20.64711606	20.39759247	20.6018988	20.6921739	20.6703872	20.6638055	20.62843663	20.69321035
	Mean	20.27204742	20.15907072	20.2779601	20.5085575	20.498879	20.4792108	20.50050223	20.23494306
	Std.	0.15193306	0.117342779	0.11900441	0.09854305	0.09567490	0.08456730	0.088448808	0.902719447



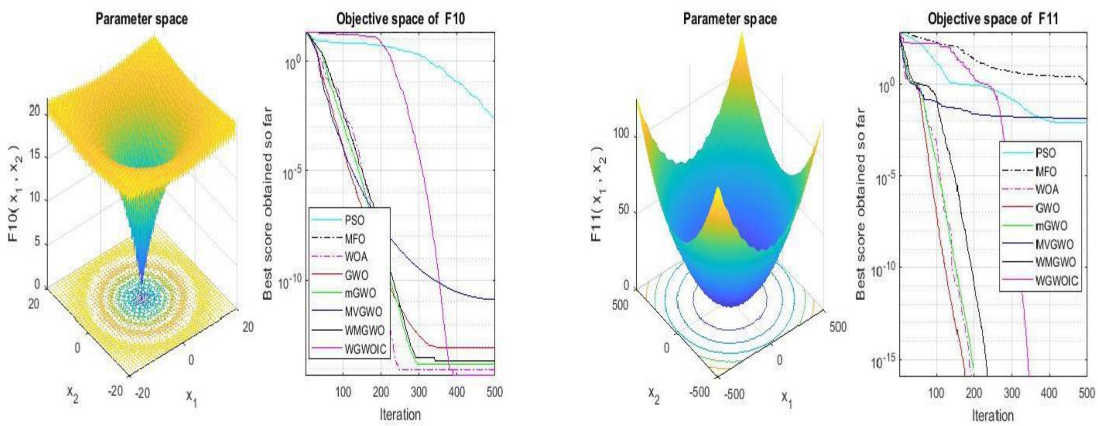


Figure 4. Convergence curve of WGWOIC algorithm and other comparative algorithms.

Subsequently, the last ten modern single objective functions (CEC01 to CEC10) have been included from CEC-C06 2019 to validate the scalability and effectiveness of the rotated and shifted ability of the proposed algorithm. Table 5 lists the computational results of these benchmark test functions. The proposed algorithm obtained very competitive effects on these functions against comparative well-known meta-heuristic algorithms. Hence, the proposed algorithm eventually offers very prominent scalability and the ability of rotated and shifted.

In addition to the above mathematical results of the experiment, we have obtained some graphical results, as depicted in Figure 4. These graphical results demonstrate the convergence rate of the proposed algorithm against other literature algorithms. The cyan color line indicates the PSO's curve graph (convergence rate graph). Similarly, the curve graphs of MFO, WOA, GWO, mGWO, MVGWO and WMGWO are shown by black color dash-dot line, magenta color dash-dot line, red color line, green color line, blue color line and black color line, respectively. In contrast, the curvegraph of WGWOIC is indicated by a magenta color line. Consequently, these graphical results justify the excellent convergence rate of the proposed algorithm compared with all comparative meta- heuristic algorithms.

5. WGWOIC IN TRAINING MULTI-LAYER PERCEPTRON

Neural Networks (NNs) [43] represents the most prominent and emerging invention in the soft computing field, proposed by McCulloch and Pitts in 1943. These networks impersonate the biological neurons of the human brain; hence they contain the ability to learn from experience. The learning methods are classified into two categories: supervised and unsupervised. As the name implies, supervised learning is provided by the supervisor or external sources (feedback). In contrast to supervised learning, unsupervised learning is conferred by merely inputs, but not accompanied by any supervisor or external sources (feedback). Neural network learning method is known as a trainer that is responsible concerning the networks' performance. Hence, the trainer is the most vital component of NNs. The set of input samples to the neural networks are notified as training samples and test samples are utilized in order to substantiate the effectiveness of their performance. However, the trainer has to accommodate the structural parameters of NNs to improve the performance in each training step. Consequently, the trainer extinctions after the training phase and the neural network are now ready to practice.

There are several types of neural networks studied in the literature, such as Recurrent Neural Networks (RNNs) [44], Feedforward Neural Networks (FNNs) [45] and so forth. The two-direction information flow between the neurons is implied in the RNNs, whereas FNNs are the simplest, most widely employed and share the information in exclusively one direction. Furthermore, FNNs are classified into two categories: Single-Layer Perceptrons (SLPs) [46] and Multi-Layer Perceptrons (MLPs) [47]. The SLPs comprise only one perceptron that constitutes it suitable to solve linear problems. In contrast to SLPs, MLPs consist of more than one perceptron at several layers, making them ideal for solving non-linear problems. There are proposed numerous applications of MLPs in the literature, such as function approximation, pattern classification and so forth. The pattern classification [48] is a supervised learning approach and it classifies the input data in to preconcert labeled classes, whereas

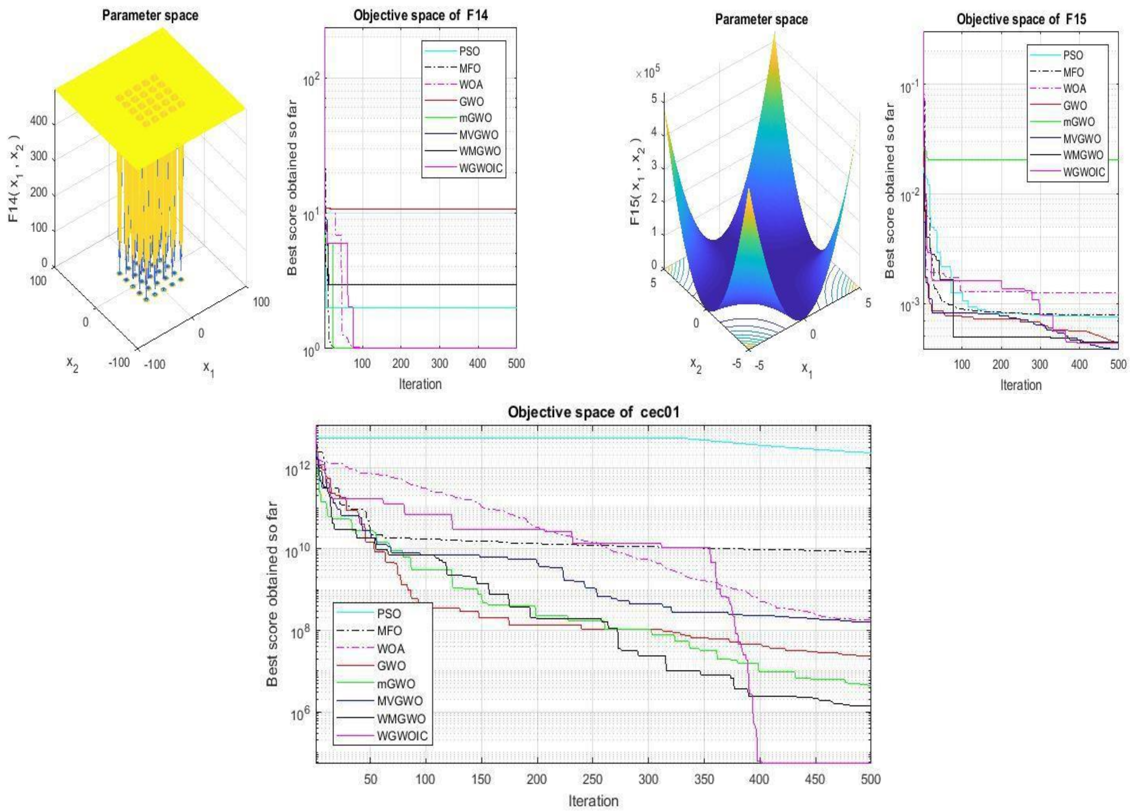


Figure 4. (*continued*).

function approximation [49] concerns the undertaking of modeling relationships amid input variables. The hierarchical classification diagram of neural networks is depicted in Figure 5.

Yu et al. [50] introduced an algorithm in order to train Support Vector Machines (SVMs) to alleviate computational complexities. The proposed algorithm utilized Fisher projection for bound vectors set to tackle linear and non-liner separates problems, for which linear and kernel Fisher discriminants were used to compute projection line. Yu et al. [51] proposed a two-side (user-side and item-side) Cross Domain Collaborate Filtering (CDCF) algorithm in order to diminish the sparsity problem that occurred in the recommender systems. The recommendation problem is converted into a classification problem by using the proposed model, alongside the SVM model employed to tackle the resultant classification problem that eventually performs significantly better than comparative methods.

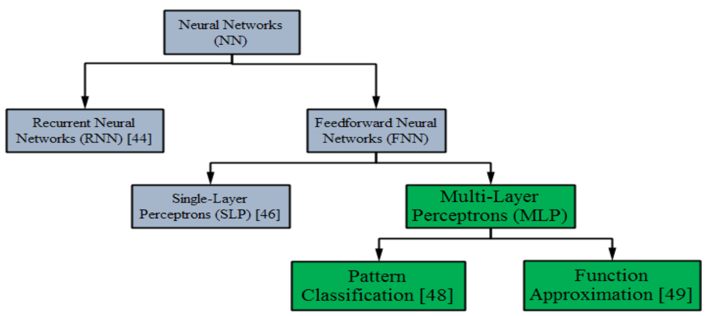


Figure 5. Hierarchical classification of neural networks.

Furthermore, Yu et al. [52] enhanced the CDCF algorithm and expanded user and item (two-dimensional) location as the feature vector using the Funk-SVD decomposition model. Further, the C4.5 decision tree algorithm has been utilized for training a classifier for predicting missing ratings.

Kolisetty and Rajput [53] contributed to intensive study regarding the significance of machine learning in analyzing big data's analysis (implications and challenges) in terms of data heterogeneity, classification imperfection and computational complexity. The analyzed data is utilized for predictive analysis and decision-making *via* data transformation and knowledge extraction. Furthermore, Ottom's critical focus [54] has emphasized the significance of big data in healthcare and its implementation tools and associated research challenges. The effectiveness of the Fuzzy-Rough Nearest Neighbour (FRNN) classifier is benchmarked by Liew et al. [55] for brainprint authentication over nine distinctive electroencephalogram channels' signals. Nahar et al. [56] employed a user-defined lexicon approach in order to determine the polarity (positive, negative) on Facebook posts and comments and achieved 98% accuracy. Apart from the above method, Naïve Bayes (NB), K-Nearest Neighbour (K-NN) and support vector machine classifiers have been utilized for polarity classification and produced 95.6, 96.8 and 97.8%, respectively. Al-Abdallah et al. [57] proposed a firefly algorithm classifier in order to tackle five binary classification problems. The effectiveness of the proposed classifier demonstrates competitive outcome against comparative classifiers. Alweshah et al. [58] proposed a hybrid approach (African buffalo optimization algorithm employed with the probabilistic neural network) to address the classification problem and applied it on 11 benchmark datasets in order to assess its accuracy. Furthermore, the water evaporation algorithm employed with the probabilistic neural network in order to tackle classification problems very effectively was developed by Alweshah et al. [59].

As we discussed earlier, the trainer is the most influential component of neural networks. There are recommended several trainers to NNs in the literature, such as Genetic Algorithms (GAs) [60], Particle Swarm Optimization (PSO) [61], Evolutionary Strategies (ESs) [62], Ant Colony Optimization (ACO) [63]-[64], Grey Wolf Optimizer (GWO) [65]-[66], Teaching-Learning Based Optimization (TLBO) [67], Moth-Flame Optimizer (MFO) [68]-[69], Population-based Incremental Learning (PBIL) [70] and so forth. In order to find a most optimal prediction for Dairy Product Demand (DPD) in Iran, Goli et al. [71] proposed a hybrid approach using GWO and cultural algorithm to improve MLPs. We are focusing on tackling the pattern classification problem using MLPs in this research work employed by the preceding proposed novel algorithm of GWO designated as WGWOIC trainer. The proposed trainer may assist in this field and provide considerably better outcomes than other comparative trainers.

5.1 Problem Formation

As discussed earlier, MLPs are specific types of Feedforward Neural Networks that contain one hidden layer with one input and one output layer. However, the output of MLPs depends on the inputs, weights and biases. In order to tackle the pattern classification datasets, these datasets already contain the inputs and outputs, while optimum weights and biases are also required for significant computational results. In this research work, we proposed an algorithm-based trainer called Weighted Grey Wolf Optimizer with Improved Convergence Rate-Multi-Layer Perceptron trainer (WGWOIC-MLP trainer) in order to optimize the values of weights and biases. The block diagram of this proposed work regarding problem formulation is depicted in Figure 6.

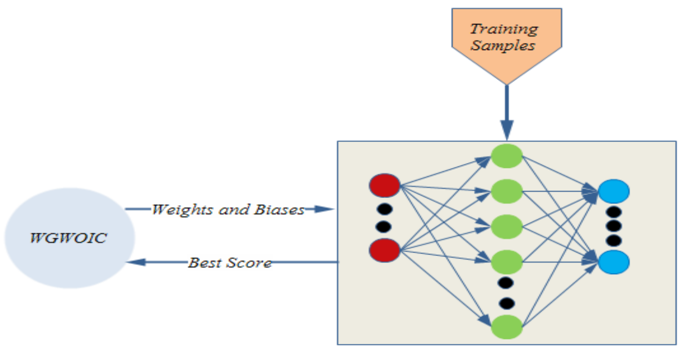


Figure 6. Block diagram of the proposed work.

The block diagram illustrates that the proposed algorithm trainer provides optimized weights and biases to the MLP that returns the best score for the testing samples. The performance is benchmarked on 3-bits XOR, balloon, iris, breast cancer and heart datasets that are well-known classification datasets in the literature. These datasets are of different difficulty levels that are considered from University of California at Irvine (UCI) Machine Learning Repository and the researchers may follow reference [61] for a detailed description of them. Therefore, the number of attributes is represented by

#Attributes for each dataset according to Table 6. Similarly, the number of training samples (#Training Samples) and test samples (#Testing Samples), number of classes (#Classes) and the corresponding MLP structure for each dataset are listed in Table 6.

Table 6. Details of pattern classification datasets.

Classification Datasets	#Attributes	#Training Samples	#Testing Samples	#Classes	MLP Structure
3-bits XOR	3	8	8	2	3-7-1
Balloon	4	16	16	2	4-9-1
Iris	4	150	150	3	4-9-3
Breast Cancer	9	599	100	2	9-19-1
Heart	22	80	187	2	22-45-1

The number of nodes in each MLP structure's hidden layer is two times the number of inputs plus one more (2*N+1, where N indicates the number of inputs in the particular dataset). The simulation results and discussion are described in the following sub-section. It may be noted that the training algorithms are represented by algorithm-MLP in this simulation.

5.2 Simulation Results and Discussion

The programming implementation for this research application has been done on the same platform as the preceding experiments. However, the settings of tuning parameters for the WGWOIC and other state-of-the-art meta-heuristic trainers are listed in Table 7. The population size (search agents) for each MLP training algorithm is 50 for the XOR and Balloon datasets, while it is 200 for the rest datasets. At the same time, the maximum number of generations is 250 for each training algorithm.

The classification rate and best score have been considered as performance-evaluating parameters of these training algorithms. However, the highest value of classification rate and lowest value of the best score indicate the most optimal solution. The best score is also called mean square error that is calculated by the difference between the actual value and the desired value of the individual sample.

According to Table 6, the XOR dataset contains three attributes, eight training/test samples and two classes. The MLP structure of this dataset is 3-7-1, implying that the multi-layer perceptron neural network contains three inputs nodes, seven hidden nodes and one output node and the trainer has 36 dimensions. The results of sub-experiments of all datasets are depicted in Figure 7 and listed in Table 8, which demonstrate that WGWOIC-MLP and GA-MLP trainers provide a 100 percent classification rate (accuracy) to classifying the XOR dataset. In contrast, the MFO-MLP trainer

Table 7. The initial parameters of training algorithms.

Training Algorithm	Parameter	Value
WGWOIC	$\vec{a}$	Exponentially decrease from 2 to 0
	Population	50 for the XOR and Balloon, 200 for rest
	#Generation	250
GA	Crossover	Single point (probability=1.0)
	Mutation	Uniform (Probability=0.01)
	Type	Real Coded
PSO	Topology	Fully Connected
	Social constant ($C2$)	1
	Cognitive constant ($C1$)	1
	Inertia constant (w)	0.3
ACO	Initial pheromone (τ)	1e-06
	Pheromone update constant (Q)	20
	Pheromone constant (q)	1
	Global pheromone decay rate (p_g)	0.9
	Local pheromone decay rate (p_t)	0.5
	Pheromone sensitivity (α)	1
	Visibility sensitivity (β)	5
ES	λ	10
	σ	1
MFO	b	1
	t	[-1, 1]
PBIL	Mutational probability	0.1
	Learning rate	0.05
	Elitism parameter	1

provides the highest best score against other well-known comparative trainers, whereas GA and WGWOIC-MLP also offer very significant results. In the case of the balloon dataset, it possesses four features, 16 training/test samples and two classes, according to Table 6. The MLP structure of the current dataset is 4-9-1 and the trainer has to optimize 55 variables. Surprisingly, all trainers' classification rates are 100 percent, whereas the GA-MLP achieves the best score while the WGWOIC-MLP trainer reached the second rank.

The iris is the most popular dataset in the literature and consists of four attributes, 150 training/test samples and three classes. The MLP structure of this current dataset is 4-9-3 and the trainer has 75 dimensions that have to be optimized. The experimental results clarify that the MFO-MLP trainer provides the highest classification rate, while WGWOIC-MLP trainer obtained the second-best accuracy. Moreover, the WGWOIC-MLP trainer offers the highest best score compared to the other trainers.

The breast cancer dataset is also a well-known dataset in the literature and contains nine features, 599 training samples, 100 test samples and two classes. The MLP structure of this dataset is 9-19-1 and the trainer has 209 variables. The experimental results show that both WGWOIC and MFO-MLP trainers provide 99 percent accuracy, while GA offers 98 percent classification rate. Moreover, the MFO-MLP trainer offers the highest best score among other meta-heuristic comparative trainers, while the WGWOIC-MLP trainer acquired the second rank.

The heart dataset is the most challenging dataset in the studied literature and consists of 22 attributes, 80 training samples, 187 test samples and two classes. The MLP structure of this current dataset is 22- 45-1 and the trainer has to optimize 1081 variables. The experimental results demonstrate that the WGWOIC-MLP trainer acquired the highest classification rate, while the MFO-MLP trainer provided the second-highest accuracy. Moreover, the GA and WGWOIC-MLP based trainer provides the highest best score compared other trainers.

To sum up, the experimental results justify that the proposed WGWOIC-MLP trainer in order to tackle the pattern classification problems provides significantly better outcomes than other well-known comparative trainers. In addition, the proposed algorithm extensively examines the search space in order to avoid the local optima and simultaneously offers an excellent equilibrium among exploration and exploitation to tackle the optimization problems; thus, its promising exploitation devotes considerably better convergence rate toward the most optimal solution to the proposed algorithm.

Table 8. Best score for the XOR, Balloon, Iris, Breast cancer and Heart datasets.

Training Algorithm	**Pattern Classification Datasets**				
	XOR (Best Score)	Balloon (Best Score)	Iris (Best Score)	Breast Cancer (Best Score)	Heart (Best Score)
WGWOIC	0.0057382	3.8195E-22	0.016928	0.0021115	0.129182
MFO	0.0000454	1.0876E-20	0.021996	0.0019964	0.178128
PSO	0.0750883	2.9306E-05	0.134318	0.0267692	0.159182
PBIL	0.0262799	5.2337E-06	0.059117	0.0243935	0.135457
GA	0.0002162	1.2192E-24	0.022447	0.0026742	0.075491
ES	0.1057665	0.0023197	0.299674	0.0401639	0.169544
ACO	0.1172278	0.0017509	0.327935	0.0114927	0.219699

In addition to the classification problem, the proposed method may be employed to tackle several potential applications of various crucial research domains of science and technology, such as machine learning applications (Training neural networks, feature selection, data clustering, optimizing SVMs), image processing applications (image thresholding, image classification), wireless sensor network applications (extending the network lifetime, network coverage problem, localization problem), engineering applications (robotics and path planning, power dispatch problems, designing and tuning controllers), Controller Placement Problem (CPP) in software-defined networking, software cost estimation and so forth.

The proposed method provides considerably better performance in terms of exploration and exploitation of the search space. The finding of this research work is that the outcomes of the WGWOIC algorithm are significantly better on high-dimensional functions, whereas it lacks in terms

Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 07, No. 03, September 2021.

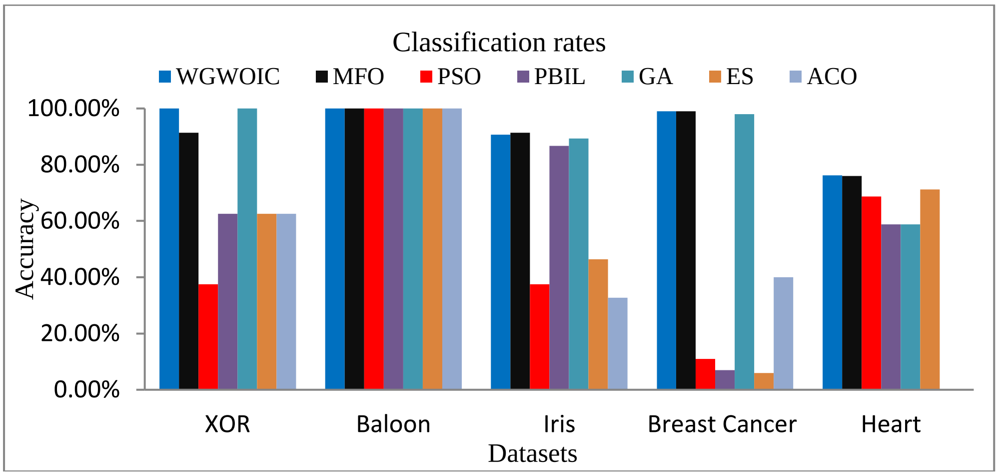


Figure 7. Classification rates for the XOR, Balloon, Iris, Breast cancer and Heart datasets.

of low-dimensional functions. However, the result analysis elaborates that the proposed method offers comparatively better results if it is employed with less population and fewer iterations to optimize low-dimensional problems. Moreover, there is scope for further research to overcome the above limitation adding some new operators or modifying the existing ones. On the other hand, the proposed method considerably tackles the classification problem of the real world.

6. Conclusion

This research work intends to introduce a novel algorithm of grey wolf optimizer to overcome its major impediment (stagnation in local optima) and the limitations of other algorithms. Further, the suggested algorithm is employed being a trainer of MLP neural networks to improve the accuracy of the classification problem. This paper introduced the novel algorithm nominated Weighted Grey Wolf Optimizer with Improved Convergence Rate (WGWOIC) for extensive exploration of the search space. Therefore, this research work has enhanced the hunting (position update equation) and the attacking (exploitation equation) mechanisms of basic GWO. In order to test the effectiveness of the WGWOIC algorithm's performance, it is benchmarked on 33 fairly popular numerical test functions that are considered from two different benchmark datasets. The experimental results of the benchmark datasets assist in justifying the strength and robustness of the proposed algorithm against the unknown search space of real-world applications. Surprisingly, the recommended algorithm outperforms on the majority of test functions against comparative studied meta-heuristic optimization algorithms, whereas it provides very competitive results on the remaining functions.

In addition, the WGWOIC algorithm was further employed as a trainer for multi-layer perceptron to classify five viral pattern classification datasets. Conclusively, it produces very competitive outcomes regarding classification rate and best score, demonstrating that the proposed algorithm is robust against challenging problems with unknown search spaces.

The experimental finding of the WGWOIC algorithm is that the proposed algorithm discovers better-quality solutions in terms of high exploration and exploitation abilities of abstract search space of numerical and real-world problems. The proposed algorithm may be further employed to tackle several potential applications of various crucial research domains; for instance, machine learning, image processing, wireless sensor network applications, controller placement problem in SDN and so forth. In addition, this algorithm may further improve after adopting the evolutionary mechanism, which is worth being the subject of further research works.

References

[1] D. Wolpert and W. Macready, "No Free Lunch Theorems for Optimization," IEEE Trans. on Evolutionary Computation, vol. 1, no. 1, pp. 67-82, April 1997.

[2] J. Holland, "Genetic Algorithms," Scientific American, vol. 267, no. 1, pp. 66-73, July 1992.

[3] K. Krishnakumar and D. E. Goldberg, "Control System Optimization Using Genetic Algorithms," Journal of Guidance, Control and Dynamics, vol. 15, no. 3, pp. 735-740, 1992.

[4] R. Storn and K. Price, "Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," J. of Global Optimization, vol. 11, no. 4, pp. 341-359, 1997.

[5] D. Simon, "Biogeography-based Optimization," IEEE Trans. on Evolutionary Computation, vol. 12, no. 6, pp. 702-713, March 2008.

[6] X. Yao, Y. Liu and G. Lin, "Evolutionary Programming Made Faster," IEEE Trans. on Evolutionary Computation, Vol. 3, no. 2, pp. 82-102, July 1999.

[7] J. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, vol. 1, MIT Press, 1992.

[8] N. Hansen, S. Müller and P. Koumoutsakos, "Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES)," Evolutionary Computation, Vol. 11, no. 1, pp. 1- 18, March 2003.

[9] X. Yao and Y. Liu, "Fast Evolutionary Programming," Evolutionary Programming, vol. 3, pp. 451-460, Feb. 1996.

[10] S. Hofmeyr and S. Forrest, "Architecture for an Artificial Immune System," Evolutionary Computation, vol. 8, no. 4, pp. 443-473, December 2000.

[11] K. Passino, "Bacterial Foraging Optimization," International Journal of Swarm Intelligence Research (IJSIR), vol. 1, no. 1, pp. 1-16, January 2010.

[12] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, "GSA: A Gravitational Search Algorithm," Information Sciences, vol. 179, no. 13, pp. 2232-2248, June 2009.

[13] B. Webster and P. Bernhard, "A Local Search Optimization Algorithm Based on Natural Principles of Gravitation," Proc. of the 2003 International Conf. on Information and Knowledge Engineering (IKE'03), pp. 255–261, Las Vegas, Nevada, USA, April 2003.

[14] A. Hatamlou, "Black Hole: A New Heuristic Optimization Approach for Data Clustering," Information Sciences, vol. 222, pp. 175-184, February 2013.

[15] F. Moghaddam, R. Moghaddam and M. Cheriet, "Curved Space Optimization: A Random Search Based on General Relativity Theory," arXiv Preprint arXiv: 1208.2214, August 2012.

[16] X. Yang, "A New Metaheuristic Bat-inspired Algorithm," Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Part of Studies in Computational Intelligence Book Series, vol. 284, pp. 65-74, Springer, Berlin, Heidelberg, 2010.

[17] S. Mirjalili, "Moth-flame Optimization Algorithm: A Novel Nature-inspired Heuristic Paradigm," Knowledge- based Systems, vol. 89, pp. 228-249, November 2015.

[18] B. Mohanty, "Performance Analysis of Moth Flame Optimization Algorithm for AGC System," International Journal of Modeling and Simulation, vol. 39, no. 2, pp. 73-87, April 2019.

[19] D. Pelusi, R. Mascella, L. Tallini, J. Nayak et al., "An Improved Moth-flame Optimization Algorithm with Hybrid Search Phase," Knowledge-based Systems, vol. 191, ID: 105277, 2020.

[20] P. Singh and SK. Bishnoi, "Modified Moth-flame Optimization for Strategic Integration of Fuel Cell in Renewable Active Distribution Network," Electric Power Systems Research, vol. 197, Article ID: 107323, 2021.

[21] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," Advances in Engineering Software, vol. 95, pp. 51-67, May 2016.

[22] B. H. Abed-alguni, "Bat Q-learning Algorithm," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 03, no. 01, pp. 52-71, DOI: 10.5455/jjcit.71-1480540385, April 2017.

[23] E. Cuevas, A. Echavarría and M. Ramírez-Ortegón, "An Optimization Algorithm Inspired by the States of Matter that Improves the Balance between Exploration and Exploitation," Applied Intelligence, vol. 40, no. 2, pp. 256-272, March 2014.

[24] X. Yang, "Flower Pollination Algorithm for Global Optimization," Proc. of International Conf. on Unconventional Computing and Natural Computation (UCNC 2012), Part of the Lecture Notes in Computer Science Book Series, vol. 7445, pp. 240-249, Springer, Berlin, Heidelberg, September 2012.

[25] A. G. Hussien, D. Oliva, E. Houssein, A. Juan and X. Yu, "Binary Whale Optimization Algorithm for Dimensionality Reduction," Mathematics, vol. 8, no. 10, 1821, October 2020.

[26] AK. Tripathi, H. Mittal, P. Saxena and S. Gupta, "A New Recommendation System Using Map-reduce-based Tournament Empowered Whale Optimization Algorithm," Complex & Intelligent Systems, vol. 7, no. 1, pp. 297-309, February 2021.

[27] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," Proceedings of the IEEE International Conference on Neural Networks (ICNN'95), vol. 4, pp. 1942-1948, November 1995.

[28] K. Parsopoulos and M. Vrahatis, "UPSO: A Unified Particle Swarm Optimization Scheme," Proc. of the International Conference of Computational Methods in Sciences and Engineering (ICCMSE 2004), CRC Press, pp. 868-873, April 2019.

[29] A. Piotrowski, J. Napiorkowski and A. E. Piotrowska, "Population Size in Particle Swarm Optimization," Swarm and Evolutionary Computation, vol. 58, Article ID: 100718, November 2020.

[30] N. S. Alshdaifat and M. H. Bataineh, "Optimizing and Thinning Planar Array Using Chebyshev Distribution and Improved Particle Swarm Optimization," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 01, no. 01, pp. 31-41, December 2015.

[31] S. Parsons, "Ant Colony Optimization by Marco Dorigo and Thomas Stützle, MIT Press, ISBN 0-262-04219-3," The Knowledge Engineering Review, vol. 20, no. 1, pp. 92, 2005.

[32] XS. Yang, "Firefly Algorithm, Stochastic Test Functions and Design Optimization," International Journal of Bio-inspired Computation, vol. 2, no. 2, pp. 78-84, January 2010.

[33] X. Yang and S. Deb, "Cuckoo Search *via* Lévy Flights," Proc. of IEEE 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 210-214, Coimbatore, India, December 2009.

[34] S. Mirjalili, S. Mirjalili S and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, pp. 46-61, March 2014.

[35] N. Mittal, U. Singh and B. Sohi, "Modified Grey Wolf Optimizer for Global Engineering Optimization," Applied Computational Intelligence and Soft Computing, vol. 2016, Article ID: 7950348, March 2016.

[36] N. Singh, "A Modified Variant of Grey Wolf Optimizer," International Journal of Science & Technology, Scientia Iranica, DOI: 10.24200/SCI.2018.50122.1523, 2018.

[37] N. Singh and S. Singh, "A Modified Mean Gray Wolf Optimization Approach for Benchmark and Biomedical Problems," Evolutionary Bioinformatics, vol. 13, DOI: 10.1177/1176934317729413, 2017.

[38] A. Kumar, A. Singh and A. Kumar, "Weighted Mean Variant with Exponential Decay Function of Grey Wolf Optimizer on Applications of Classification and Function Approximation Dataset," Proc. of the International Conference on Hybrid Intelligent Systems, Springer, Cham, pp. 277-290, December 2019.

[39] B. H. Abed-alguni and M. Barhoush, "Distributed Grey Wolf Optimizer for Numerical Optimization Problems," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 04, no. 03, pp. 1-20, DOI: 10.5455/jjcit.71-1532897697, December 2018.

[40] K. Price, N. Awad, M. Ali and P. Suganthan, "The 100-digit Challenge: Problem Definitions and Evaluation Criteria for the 100-digit Challenge Special Session and Competition on Single Objective Numerical Optimization," Technical Report, Nanyang Technological University, November 2018.

[41] M. Abdullah and T. Ahmed, "Fitness Dependent Optimizer Inspired by the Bee Swarming Reproductive Process," IEEE Access, vol. 7, pp. 43473-43486, March 2019.

[42] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 2nd Edn., Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2, pp. 111–114, 2003.

[43] W. McCulloch and W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," The Bulletin of Mathematical Biophysics, vol. 5, no. 4, pp. 115-133, December 1943.

[44] G. Dorffner, "Neural Networks for Time Series Processing," Neural Network World, vol. 6, pp.447-468, 1996.

[45] G. Bebis and M. Georgiopoulos, "Feed-forward Neural Networks," IEEE Potentials, vol. 13, no. 4, pp. 27- 31, October 1994.

[46] P. Auer, H. Burgsteiner and W. Maass, "A Learning Rule for Very Simple Universal Approximators Consisting of a Single Layer of Perceptrons," Neural Networks, vol. 21, no. 5, pp. 786-795, June 2008.

[47] P. Werbos, Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph.D. Dissertation, Harvard University, 1974.

[48] P. Melin, D. Sánchez and O. Castillo, "Genetic Optimization of Modular Neural Networks with Fuzzy Response Integration for Human Recognition," Information Sciences, vol. 197, pp. 1-19, August 2012.

[49] W. Gardner and S. Dorling, "Artificial Neural Networks (the Multilayer Perceptron): A Review of Applications in the Atmospheric Sciences," Atmospheric Environment, vol. 32, no. (14-15), pp. 2627-2636, August 1998.

[50] X. Yu, J. Yang and Z. Xie, "Training SVMs on a Bound Vectors Set Based on Fisher Projection," Frontiers of Computer Science, vol. 8, no. 5, pp. 793-806, October 2014.

[51] X. Yu, Y. Chu, F. Jiang, Y. Guo and D. Gong, "SVMs Classification Based Two-side Cross Domain Collaborative Filtering by Inferring Intrinsic User and Item Features," Knowledge-based Systems, vol. 141, pp. 80-91, February 2018.

[52] X. Yu, F. Jiang, J. Du and D. Gong, "A Cross-domain Collaborative Filtering Algorithm with Expanding User and Item Features *via* the Latent Factor Space of Auxiliary Domains," Pattern Recognition, vol. 94, pp. 96-109, October 2019.

[53] V. V. Kolisetty and D. S. Rajput, "A Review on the Significance of Machine Learning for Data Analysis in Big Data," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 01, pp. 155- 171, DOI: 10.5455/jjcit.71-1564729835, March 2020.

[54] M. A. Ottom, "Big Data in Healthcare: Review and Open Research Issues," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 03, no. 01, pp. 38-51, DOI: 10.5455/jjcit.71-1476816159, April 2017.

[55] S.-H. Liew, Y.-H. Choo and Y. F. Low, "Fuzzy-rough Classification for Brainprint Authentication," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 05, no. 02, pp. 52-71, DOI: 10.5455/jjcit.71-1556703387, August 2019.

[56] K. M.O. Nahar, A. Jaradat, M. S. Atoum and F. Ibrahim, "Sentiment Analysis and Classification of Arab Jordanian Facebook Comments for Jordanian Telecom Companies Using Lexicon-based Approach and Machine Learning," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 06, no. 03, pp. 52-71, DOI: 10.5455/jjcit.71-1586289399, Sep. 2020.

[57] R. Z. Al-Abdallah, A. S. Jaradat, I. Abu Doush and Y. A. Jaradat, "A Binary Classifier Based on Firefly Algorithm," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 03, no. 03, pp. 32- 46, DOI: 10.5455/jjcit.71-1501152301, December 2017.

[58] M. Alweshah, L. Rababa, M. H. Ryalat, A. Al Momani and M. F. Ababneh, "African Buffalo Algorithm: Training the Probabilistic Neural Network to Solve Classification Problems," Journal of King Saud University - Computer and Information Sciences, DOI: 10.1016/j.jksuci.2020.07.004, 2020.

[59] M. Alweshah, E. Ramadan, M. H. Ryalat, M. Almi'ani and A. I. Hammouri, "Water Evaporation Algorithm with Probabilistic Neural Network for Solving Classification Problems," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 6, no. 1, pp. 1-14, March 2020.

[60] S. Tang, M. Li, F. Wang, Y. He and W. Tao, "Fouling Potential Prediction and Multi-objective Optimization of a Flue Gas Heat Exchanger Using Neural Networks and Genetic Algorithms," International Journal of Heat and Mass Transfer, vol. 152, Article ID: 119488, May 2020.

[61] M. F. Ab Aziz, S. A. Mostafa, C. F. M. Foozy, M. A. Mohammed, M. Elhoseny and A. Z. Abualkishik, "Integrating Elman Recurrent Neural Network with Particle Swarm Optimization Algorithms for an Improved Hybrid Training of Multidisciplinary Datasets," Expert Systems with Applications, vol. 183, p. 115441, June 2021.

[62] F. E. Fernandes Jr and G. G. Yen, "Pruning of Generative Adversarial Neural Networks for Medical Imaging Diagnostics with Evolution Strategy," Information Sciences, vol. 558, pp. 91-102, May 2021.

[63] A. Zannou and A. Boulaalam, "Relevant Node Discovery and Selection Approach for the Internet of Things Based on Neural Networks and Ant Colony Optimization," Pervasive and Mobile Computing, vol. 70, Article ID: 101311, January 2021.

[64] H. Zhang, H. Nguyen, X. Bui et al., "Developing a Novel Artificial Intelligence Model to Estimate the Capital Cost of Mining Projects Using Deep Neural Network-based Ant Colony Optimization Algorithm," Resources Policy, vol. 66, Article ID: 101604, June 2020.

[65] S. Mirjalili, "How Effective Is the Grey Wolf Optimizer in Training Multi-layer Perceptrons?" Applied Intelligence, vol. 43, no. 1, pp. 150-161, July 2015.

[66] H. Faris, S. Mirjalili and I. Aljarah, "Automatic Selection of Hidden Neurons and Weights in Neural Networks Using Ggrey Wolf Optimizer Based on a Hybrid Encoding Scheme," International Journal of Machine Learning and Cybernetics, vol. 10, no. 10, pp. 2901-2920, October 2019.

[67] E. Uzlu, M. Kankal, A. Akpınar and T. Dede, "Estimates of Energy Consumption in Turkey Using Neural Networks with the Teaching–learning-based Optimization Algorithm," Energy, vol. 75, pp. 295-303, October 2014.

[68] W. Yamany, M. Fawzy, A. Tharwat and A. Hassanien, "Moth-flame Optimization for Training Multi-layer Perceptrons," Proc. of the 11th IEEE International Computer Engineering Conference (ICENCO), pp. 267-272, Cairo, Egypt, December 2015.

[69] R. Singh, S. Gangwar, D. Singh and V. Pathak, "A Novel Hybridization of Artificial Neural Network and Moth-flame Optimization (ANN–MFO) for Multi-objective Optimization in Magnetic Abrasive Finishing of Aluminium 6060," Journal of the Brazilian Society of Mechanical Sciences and Engineering, vol. 41, no. 6, pp. 1-19, June 2019.

[70] R. Vasco-Carofilis, M. Gutiérrez-Naranjo and M. Cárdenas-Montes, "PBIL for Optimizing Hyperparameters of Convolutional Neural Networks and STL Decomposition," Proc. of the International Conference on Hybrid Artificial Intelligence Systems, Springer, Cham, pp. 147-159, DOI: 10.1007/978-3-030-61705-9_13, November 2020.

[71] A. Goli, H. K. Zare, R. T. Moghaddam and A. Sadeghieh, "An Improved Artificial Intelligence Based on Gray Wolf Optimization and Cultural Algorithm to Predict Demand for Dairy Products: A Case Study," International Journal of Interactive Multimedia and Artificial Intelligence, vol. 5, no. 6, pp. 15-22, March 2019.

ملخص البحث:

تعمل هذه الورقة على تحسين آلية الصيد والهجوم من أجل تعديل معادلة الموقع المحدَّثة ومعادلة الاستكشاف –على الترتيب- لاقتراح خوارزمية مبتكرة تحمل اسم "نظام إيجاد الحلول المُثلى باستخدام طريقة الذّئاب الرّمادية الموزونة مع معدّل التقاء محسّن". وقد تمّ استقصاء فاعلية الخوارزمية المقترحة باختبارها على (33) وظيفة مرجعيّة مختلفة شائعة الى حدّ ما. وقد اختبرت تلك الوظائف المرجعيّة من مجموعتي بياناتٍ مختلفتين لتقييم قوّة الخوارزمية المقترحة ومتانتها فيما يتعلّق بحيّز البحث غير المعروف للمشكلة.

وبهدف تحليل الأداء، تمت مقارنة نتائج الخوارزمية المقترحة مع نتائج خوارزميات أخرى جرى التّطرُّق إليها في أدبيات الموضوع، مثل خوارزميات إيجاد الحلول المُثلى القائمة على استراتيجية كلّ من أسراب الدّقائق، وعثّة النّار، والحيتان، والذّئاب الرّمادية، وأحدث الخوارزميات المستندة الى استراتيجية الذّئاب الرّمادية. وتوضح نتائج المقارنة أنّ الخوارزمية المقترحة أبْلَتْ بلاءً حسناً مقارنةً بغيرها من الخوارزميات؛ فقد تفوّقت عليها في عددٍ من الوظائف وكانت منافسةً لها في وظائف أخرى.

من ناحية أخرى، تم تهجين الخوارزمية المقترحة بشبكة عصبيّة متعدّدة الطّبقات لتحسين الدّقة فيما يتعلّق بمشكلات التّصنيف؛ إذ يوفر "مدرّب" الخوارزمية المقترحة القيم المُثلى للوزن والانحيازللشبكة العصبيّة متعدّدة الطّبقات. كذلك تمّ فحص الأداء من حيث دقّة التّصنيف باستخدام خمس مجموعات بياناتٍ شائعة، وتقييم فاعليّة "مدرّب" النّظام المقترح بالمقارنة مع أنظمةٍ أخرى مشابهة. وأشارت النتائج الى أنّ الخوارزمية المقترحة كانت ذات تنافسيّة جيّدة من حيث استغلال واستكشاف حيّز البحث وحلّ العديد من مشكلات التّصنيف بفاعلية.