Journal of Scientific & Industrial Research
Vol. 82, November 2023, pp. 1193-1207
DOI: 10.56042/jsir.v82i11.5322

Improved Chaotic Grey Wolf Optimization for Training Neural Networks

B V Ramana[1], Nibedan Panda[2]*, S Teja[1], Hitesh Mohapatra[2], A K Dalai[3] & S K Majhi[4]

[1]Department of Information Technology, Aditya Institute of Technology and Management, Tekkali 532 201, Andhra Pradesh, India
[2]School of Computer Engineering, KIIT Deemed to be University, Bhubaneswar 751 024, Odisha, India
[3]School of CSE, VIT, Amaravati 522 037, Andhra Pradesh, India
[4]Department of CSE, Veer Surendra Sai University of Technology, Burla 768 018, Odisha, India

Received 03 October 2021; revised 20 April 2023; accepted 21 September 2023

This paper introduces one improved version of the Grey Wolf Optimization algorithm (GWO), one of the newly established nature-inspired metaheuristic algorithms, and the suggested approach is termed Chaotic Grey Wolf Optimization (CGWO). The newly suggested approach CGWO is premeditated by the integration of the chaos technique with the GWO algorithm, aiming to resolve global optimization problems by maintaining a proper balance between exploration and exploitation. In the proposed approach, CGWO is assessed over the classic 23 benchmark functions. The proficiency of the freshly suggested approach, CGWO is verified by comparing it with contemporary methods as well as examined through statistical analysis also. Further, the same CGWO is utilized to train neural networks (MLP) by considering benchmark datasets, for data classification and establishing a better classifier algorithm.

Keywords: ANN, Chaos technique, GWO, Metaheuristic optimization, Swarm intelligence

Introduction

Nature gives a portion of the productive approaches to take care of some complex problems or issues. Algorithms that are stimulated by nature are called nature-inspired algorithms. Nature-motivated calculations are novel, encompass critical thinking techniques, and have been dragging significant consideration for their great performance. Representative instances of nature-inspired algorithms include swarm intelligence, evolutionary computing, neural computation, cognitive computing, and artificial immune systems. These nature-inspiring algorithms have a major role in unravelling many real-world intricate difficulties by using optimization techniques and models. Optimization is defined as the assortment of the finest solutions.[1] Fundamental results and numerical optimization methods can be used to find the ideal choice among many possible alternatives. Optimization techniques follow deterministic and random intelligence approaches. Deterministic methodologies produce indistinguishable arrangements if the initiation values are equal when taking care of a similar issue. Unlike deterministic methodologies, gradient-free stochastic approaches are mainly centred on random walks.

*Author for Correspondence
E-mail: nibedan.pandafcs@kiit.ac.in

Swarm Intelligence represents a sort of critical thinking capacity that rises in the collaborations of basic data handling units. The idea of a swarm suggests variety, stochasticity, randomness, and chaos and the concept of intelligence refers to the critical thinking technique in some way.[2] The data handling units that create a swarm can be quickened, machine-driven, computational, and scientific. These data handling units may be birds, insects, or human beings and those may be array elements, robots, or anything else.[3] Generally, optimization problems can be solved by utilizing two different ways, those are classical methods and metaheuristic methods. Some of the methods like gradient descent are examples of classical methods and these are easier to implement.[4] However, the drawback of these traditional approaches is time-consuming and accuracy also depends on the type of variables, conditions, and objective function of the solved problem. To avoid the drawbacks of traditional methods, metaheuristic algorithms are turning into a significant piece of current advancement.[5] The biological and physical behaviour in nature is emulated by these algorithms. There is a wide scope of development of metaheuristic algorithms for global search. The search process can be guided by these metaheuristic algorithms, which use an intelligent learning mechanism. Prominent nature-inspired metaheuristic procedures are Particle Swarm Optimization (PSO), Differential Evolution (DE), Artificial Bee Colony (ABC), Bat Algorithm (BA), Ant Lion Optimizer (ALO), Ant Colony Optimization (ACO), Genetic Algorithm (GA), Harmony Search algorithm (HS), Cuckoo Search algorithm (CS), Fruit fly Optimization Algorithm (FOA), Moth-Flame Optimization (MFO) and Grey Wolf Optimizer (GWO).

In this study, we have considered GWO as a meta-heuristic optimizing algorithm. This optimizer is centred on the control hierarchy and chasing mechanisms of grey wolves. The three main steps of hunting include searching for the quarry, encircling the quarry, and attacking the quarry are used to accomplish the optimization. The main goal of this approach is to update the present points of wolves in the discrete searching space so that we can get the finest feature for enhanced classification results. It is proved that the GWO gives improved results in most of the trials when it is equated with other meta-heuristic optimizing techniques. In some of the situations, it also shows deprived performance and trap in local minima, as it depends on heuristics. So to deal with such kind of situations hybrid algorithms are the most suitable with their high range of exploration and exploitation phases. A hybrid algorithm is a set of rules that integrates two or further different algorithms that can find the solution for a similar problem, both picking one or moving among them over the direction of a set of rules.[6,7] The main goal of such hybrid techniques are to mitigate the respective flaw of specific algorithms, possibly integrate exceptional algorithms collectively, or beautify an algorithm with one-of-a-kind techniques.[8,9] The goal of this research is to design the hybrid approach CGWO, in which diverse chaotic systems are cast to interchange the crucial constraints of GWO, which aids in switching GWO's local and global searching abilities, and to propose a mathematical model of grey wolf leadership hierarchy and chasing mechanisms in nature. Further, the suggested hybrid CGWO approach is used to train Multi-Layer Perceptron (MLP) to get the minimum training error and maximize the accuracy of prediction and classification rate.

Literature Review

This section comprehends the detailed investigation of GWO and its various variants using chaotic functions. GWO is a population-based meta-heuristic approach stimulated by natural wolf packs proposed by Mirjalili *et al.*[10] in 2014. The author explained the algorithm by going through the three main stages of the hunt: searching, encircling, and attacking the prey. Also, they have mitigated classical engineering problems using the proposed method. In 2015, Seyedali Mirjalili[11] used the GWO algorithm for training Multi-layer perceptrons. He utilized standard datasets including function approximation and classification to benchmark the analysis of the performance of his work and he compared these results with some other evolutionary training algorithms. Mirjalili *et al*.[12] suggested Multi-objective GWO in 2015, which is a unique algorithm for multi-criterion optimization. This paper illustrated that to optimize complications with numerous objectives, the Multi-Objective GWO (MOGWO) can be used. In 2015, An Automatic generation control of a multi-area ST-thermal power System using GWO based on classical controllers was proposed by Sharma & Saikia where he proved that GWO optimized PID controller's achievement is coming in a superior way than other optimizing techniques in terms of settling time.[13] In 2017, Rodriguez *et al*. presented a fuzzy hierarchical operator in GWO. This work deals with the performance of GWO while a modern hierarchical operator is incorporated into the GWO algorithm. The introduced variable is a hierarchical transformation that is stimulated by the social hierarchy of wolves pack. It has been proved that the results taking the largest impression in GWO are based on the usage of fuzzy logic.[14] Song *et al*. published a paper on GWO for constraint assessment in surface waves in the year of 2015. The research was unique and commanding surface wave dispersion curve inversion scheme so-called GWO. In this work, it is substantiated that this strategy is benchmarked on different data such as noise-free, field, and noisy data. These results are compared with other contemporary algorithms.[15] Kohli *et al*. introduced a chaotic GWO algorithm for constrained optimization glitches. He has assimilated chaotic theory with GWO to get better results in global convergence. In this, the CGWO is equated with standard GWO and also with contemporary approaches. Here, the result was validated by using some constrained engineering problems.[16] In 2018, Saxena *et al*. applied β chaotic map with GWO. In this publication, the balanced bridging technique depends on exploiting β-chain chaos to improve GWO. The control vector of this GWO is combined with the chaotic β chain to improve exploration and exploitation efficiency and the applicability of the proposed variable is equated with two realistic complications.[17] Mitic *et al*. in 2016 applied this CGWO for learning and imitation of robot movement paths. In this, he explained the novel approach that combines knowledge from demonstration procedures and muddled optimization to replicate the desired motion paths. In this, four various chaotic techniques were designed CBA, CFA, CAPSO, and CGWO. This GWO algorithm was tested on benchmark problems with the help of using ten well-defined chaotic maps.[18] In 2018, Heidari & Abbaspour used this CGWO to solve realistic complications. In this, a fresh chaotic calculation was introduced in GWO to make the agents move a chaotic sequence towards a randomly specified wolf. The performance of enhanced GWO was equated with other six optimizers which are from CEC-2011.[19] In the year 2020, Panda & Majhi attempted to resolve complicated real-world problems by training various neural networks starting from MLP to higher-order neural networks (HONN) in terms of contemporary hybrid metaheuristic approaches.[20,21] In the process they have cast off recent evolutionary techniques like STS and OBL with swarm-based approaches such as SSA and SHO to develop an improved hybrid version of the same to deal with ongoing problems.[22,23] GWO is also utilized with CNN to deal with image data effectively, recently Mohakud & Dash proposed a GWO-based CNN model to categorize skin cancer from specified input images. From the experimentation, the author claimed that the GWO-CNN model has the potential to deal with image data.[24] In modern days, due to the increase in problem complexity, the hybridized algorithms perform better in terms of better convergence rate, avoidance of local optima trap and suitable for a superior trainer. Hence the recognition of such kinds of algorithms is evolving day by day.

Framework of GWO

Inspiration

Mirjalili *et al.* proposed this algorithm in the year 2014.[(10)] This algorithm is centred on representing the guidance hierarchy and hunting process of grey wolves in the wild. In the wild, these wolves are habitual to live in a pack. The average wolf pack size is 5 to 12 animals. These wolves can be of four types based on their strength in every pack. The social hierarchy of the grey wolves can be organized into four ways. The alpha wolves (α): Forerunners are male or female, entitled alphas. Alpha is mainly responsible for choices regarding hunting, resting places, waking times, etc. These alphas will acknowledge the remaining wolves in the clan. The alpha wolf is also known as the leading wolf because its orders must be obeyed by the clan. He is the best and strongest member of the group. Beta wolf (β): These are the second level of the pyramid. These betas are the helpers who support Alpha in policymaking or any other happenings in the clan. He will obey the alpha's orders and act as an advisor to the alpha and a disciplinarian for the clan. Deltas wolves (δ): These are the third level of hierarchy in the clan. If a wolf does not belong to alpha, beta, or omega, then these are named deltas. Deltas must obey alphas and betas and simultaneously guide the omega wolves. Omega wolves (ω): These are at the lowest hierarchical level within the clan. These wolves act as scapegoats. They must always report to all other wolves.

Mathematical Model and Algorithm

To perform optimization and design the GWO procedure, the chasing method and the societal pyramid of wolves are mathematically modelled. The proposed scientific representations of the societal pyramid, tracking, encircling, and attacking prey are as per the following. In the GWO algorithm, the optimization is guided by alpha (α), beta (β), and delta (δ). The omega (ω) wolves will be followed by these three wolves in the hierarchy. The prey during hunting can be encircled by these grey wolves. Mathematically to model the behaviour of encircling, we use the following equations:

$$\vec{D} = \left|\overrightarrow{C} \cdot \vec{X}_p(t) - \vec{X}(t)\right| \quad ...(1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \quad ...(2)$$

where, t specifies the present reiteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X}_p$ is the position vector of the prey, and $\vec{X}$ designates the position vector of a grey wolf.

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 \cdot \vec{a} \quad ...(3)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad ...(4)$$

where, constituents of $\vec{a}$ are linearly declined from 2 to 0 during iterations and r_1, r_2 are random vectors within$[0,1]$.

Hunting

To mathematically model the hunting behaviour of wolves, we use the first 3 finest results which are gotten so far and constrain other wolves to renew their locations according to the location of the finest wolf. Here we use Eq. 5 & 6 to update the positions of search agents.

$$\vec{D}_{\infty} = \left|\vec{C}_1 \cdot \vec{X}_{\infty} - \vec{X}\right|, \vec{D}_{\beta} = \left|\vec{C}_2 \cdot \vec{X}_{\beta} - \vec{X}\right|, \vec{D}_{\delta} = \left|\vec{C}_3 \cdot \vec{X}_{\delta} - \vec{X}\right| \quad ... (5)$$

$$\vec{X}_1 = \vec{X}_{\infty}\vec{A}_1 \cdot \left(\vec{D}_{\infty}\right), \vec{X}_2 = \vec{X}_{\beta}\vec{A}_2 \cdot \left(\vec{D}_{\beta}\right), \quad \vec{X}_3 = \vec{X}_{\delta}\vec{A}_3 \cdot \left(\vec{D}_{\delta}\right) \quad ... (6)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad ... (7)$$

Using these above equations, all the search agents will alter their locations according to the locations of alpha, beta, and delta in the search region. Here alpha, beta, and delta wolves will assess the location of the target, and the remaining agents can alter their locations arbitrarily around the prey. Here the hunting of prey can be completed by attacking the prey. Approaching the prey can be modelled mathematically by decreasing the value of $\vec{a}$, where a decreases from 2 to 0 over the iterations. Here $\vec{A}$ is also decreased by $\vec{a}$ and it is having random values in between the range of $[-2a, 2a]$. Grey wolves mostly search as per the location of alpha, beta, and delta. They wander from one another to look for prey and combine to assault prey. Here C vector, which is a component of GWO contains random values in $[0, 2]$. The random behaviour throughout optimization can be shown in GWO because of this vector C. The pseudo-code for GWO is stated in Table 1.

Table 1 — Pseudocode representation of GWO algorithm

Algorithm 1 Pseudocode illustration of GWO

Configure the grey-wolf populace Yi (cnt = 1, 2, ..., n)
Set a, A, and C
Compute the fitness of individual search-agent
X_{α} =finest search-agent
X_{β} =second finest search-agent
X_{δ} =third finest search-agent
while (iterc < Max count of recapitulations)
  for each independent search-agent
Alter the location of the present search-agent using the previously mentioned equations
  end for
  Alter a, A, and C
  Compute the fitness of entire search-agents
  Alter X_{α}, X_{β}, and X_{δ}
  iterc=iterc+1
end while
return X_{α}

Chaos Technique

Chaos theory is a part of science concentrating on the investigation of chaos states of dynamical systems. Chaos can be demarcated as a phenomenon where any small change in its preliminary condition leads to a non-linear change in the forthcoming behaviour. Edward Lorenz was the first pioneer of this concept. The deterministic laws control the random states of disorder and inconsistencies in dynamical systems. These deterministic laws have a major role in chaos techniques, which are highly delicate to basic conditions. Little changes in introductory conditions, for example, those because of measurements in estimation or because of adjusting mistakes in numerical calculation can return broadly wandering results for such dynamical systems, knowing longstanding predictions of their behaviour is generally not necessary. It may happen with dynamic systems which are naturally deterministic and cannot be anticipated. The future properties of these dynamical systems will always follow a particular expansion, and these are determined by their input conditions, without complex stochastic factors. This character is called simply chaos or deterministic chaos. The meaning of a deterministic system is one in which no stochastic nature is taking part in the development of further states of the system. According to the summary of chaotic behaviour which is explained by Edward Lorenz[25], the nature of chaos is present in a lot of usual systems, comprising fluid flow, heartbeat abnormalities, weather, and climate. In some systems which are having artificial mechanisms, such as the stock market and highway traffic, this chaotic behaviour occurs spontaneously.[26–29] The nature of these chaotic systems is initially predictable for some time and after that, it appears to become random. There are three factors based on which the total time that the behaviour of a chaotic system can be effectually predicted. Those things are a time scale depending on the dynamics of the system which is called the Lyapunov time, how much uncertainty can be endured in the calculation, and how precisely its present state can be estimated. At the point when important predictions can’t be made, the system seems arbitrary.

Proposed CGWO Algorithm

Chaos is a mathematical technique that has received enormous attention among researchers due to its underlying associated randomness and irregularities. However such states are controlled by defined patterns and deterministic regulations which are extremely inclined towards initial constraints. A chaotic technique is frequently used with optimization engineering, as it uses various chaotic parameters to approximate. Due to the advantages of non-recurrence and haphazardness, the entire search process can be carried out in a superior manner within the search region in contrast to other techniques which also depend on possibilities. To enhance the ability of GWO to find global optimum along with improving the convergence rate, we have incorporated the technique of chaos with GWO. Due to the underlying dynamic nature of chaos, it boosts the capability of any chosen metaheuristic approach for achieving optimum within the search region. Out of 12 available chaotic map functions we have chosen the Logistic map function and applied it further for realistic applications. A variety of chaotic maps with various formulas can be used to bring chaos into optimization methods. Since a decade ago, chaotic maps have been gaining popularity in the optimization field due to their dynamic behaviour, which aids optimization algorithms in examining the search space more thoroughly. Of the available chaotic maps, many of them have been utilized the same to mitigate a wide range of real-world complex applications. To enhance the efficacy of the GWO approach, we have integrated the chaos technique with GWO, and the hybrid version of the proposed method is considered a Chaotic Grey Wolf Optimization (CGWO) algorithm. The initial value can be taken from the range of [0,1] in any chaotic map. Here we have taken the initial value as 0.6. In this work, the model is centred on the common S-curve logistic function that displays how a populace advances gradually, at the point quickly, before tightening as it arrives at its conveying limit. Here we used the logistic function which is one of the chaotic functions. It uses a differential equation that treats time as regular. This condition characterizes the guidelines or rules of our framework: here n signifies the populace at some random time x, and p signifies the growth or development rate. Here, we have taken the initial value of n_x as 0.6 as it should be in the range of [0,1]. The development rate p-value as 0.5 is taken here as the range of the growth rate can be in the range of 0 to 4. The pseudo-code for GWO is stated in Table 2.

$$n_{x+1} = pn_x(1- n_x) \quad \text{... (8)}$$

Evaluation of CGWO over Standard Constrained Functions

The outcomes obtained by CGWO over 23 standard benchmark functions are presented in Table 3. The diverse performance measures achieved from these benchmark functions set have been introduced as max, min, median, mean, and Standard Deviation (SD). The benchmark dimensions utilized are minimization methods and can be separated into four kinds. Those are unimodal, multi-modal, and fixed-measurement multi-modal. In those 23 benchmark functions, F1 to F7 are the category of unimodal, F8 to F13 are under the category of

Table 2 — Pseudocode representation of suggested Chaotic-GWO (CGWO) algorithm

Algorithm 2 Pseudocode illustration of CGWO

```
Initialize the variable iterc and arbitrarily configure the populace of grey wolves where (cnt=1, 2..., n)
Configure the value of the chaotic map x0arbitrarily
Configure constraints a, A and C
Compute the fitness of each Wolf
Xα = finest wolf
Xβ = second finest wolf
Xδ =third finest wolf
while (iterc< Max_recapitulations )
 Sort the populace of grey wolves rendering their fitness
Configure the chaotic number using the chaotic map equation (Eq. 8)
    for each independent search-agent
            Alter the position of the present wolf using Eq. 5
    end for
Configure constraints a, A, C
Compute fitness of entire wolves.
 AlterXα, Xβ, Xδ
Substitute the worst fit wolf with the finest fit wolf iterc=iterc+1
 End while
 returnXα
```

Table 3 — Unimodal, Multimodal and Fixed Dimension Multimodal Benchmark Functions

Function	Dimension	Range	fmin
Unimodal Benchmark Functions			
$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100,100]	0
$f_2(x) = \sum_{i=1}^{n} \lvert x \rvert + \prod_{i-1}^{n} \lvert x_i \rvert$	30	[−10,10]	0
$f_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	30	[−100,100]	0
$f_4(x) = max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	30	[−100,100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_1^2)^2 + (x_i - 1)^2]$	30	[−30,30]	0
$f_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[−100,100]	0
$f_7(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1]$	30	[−1.28,1.28]	0
Multimodal Benchmark Functions			
$f_8(x) = \sum_{i-1}^{n} x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[−500, 500]	418.9829X5
$f_9(x) = \sum_{i=1}^{n} [x_1^2 - 10\cos(2\pi x_i) + 10]$	30	[−5.12,5.12]	0
$f_{10}(x) = -20\, exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_1^2}\right) - exp\left(\sum_{i=1}^{n} cos(2\pi x_i)\right) + 20 + e$	30	[−32,32]	0
$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_1^2 - \prod_{i=1}^{n} cos\,(\frac{xi}{\sqrt{i}}) + 1$	30	[−600, 600]	0
$f_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_i) + \sum_{i-1}^{n-1} (y_{1-} 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i-1}^{n} u(x_i, 10, 100, 4)$; $y_i = 1 + \frac{xi - 1}{4}$; $u(x_i, a, k, m) = \begin{cases} k(x_i - a)m\ x_i > a \\ 0\ -a < x_i < a \\ k(-x_i - a)m\ x_i < -a \end{cases}$	30	[−50, 50]	0
$f_{13}(x) = 0.1\{\sin^2(3\pi y_i) + \sum_{i=1}^{n} (x_{1-} 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + sin^2 (2\pi x_n)]\} + \sum_{i-1}^{n} u(x_i, 5, 100, 4)$	30	[−50, 50]	0
$f_{14}(x) = -\sum_{i=1}^{n} \sin(x_i) . \left(\sin(\frac{i. x_i^2}{\pi})\right)^{2m}, m = 10$	30	$[0, \pi]$	−4.687
$f_{15}(x) = \left[e - \sum_{i=1}^{n} \left(\frac{x_i}{\beta}\right)^{2m} - 2e - \sum_{i=1}^{n} x_i^2\right] \cdot \prod_{i=1}^{n} cos^2 x_i, m = 5$	30	[−20,20]	−1

(*Contd*.)

Table 3 — Unimodal, Multimodal and Fixed Dimension Multimodal Benchmark Functions (*Contd.*)

Multimodal Benchmark Functions			
$f_{16}(x) = \left\{\left[\sum_{i=1}^{n} \sin^2(x_i)\right] - exp - (\sum_{i=1}^{n} x_i^2] - exp - \left(\sum_{i=1}^{n} x_i^2\right)\right\} \cdot exp\left[-\sum_{i=1}^{n} sin^2 \sqrt{\lvert x_1 \rvert}\right]$	30	[−10,10]	−1
Fixed Dimension Multimodal Benchmark Functions			
$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	[−5,5]	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * [30 + (2x_1 - 3x_2)^2 * (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[−2,2]	3
$f_{19}(x) = \sum_{i=1}^{4}[c_i \exp(-\sum_{i=1}^{3} a_{ij}\ (x_j - p_{ij})^2)$	3	[1,3]	−3.86
$f_{20}(x) = \sum_{i=1}^{4}[c_i \exp(-\sum_{i=1}^{6} a_{ij}\ (x_j - p_{ij})^2)$	6	[0,1]	−3.32
$f_{21}(x) = \sum_{i=1}^{5}[(X - a_i)(X - a_i)^{T+} c_i]^{-1}$	4	[0,10]	−10.1532
$f_{22}(x) = \sum_{i=1}^{7}[(X - a_i)(X - a_i)^{T+} c_i]^{-1}$	4	[0,10]	−10.4028
$f_{23}(x) = \sum_{i=1}^{10}[(X - a_i)(X - a_i)^{T+} c_i]^{-1}$	4	[0,10]	−10.5363

multi-modal functions, and the remaining is the type of fixed-dimension multi-modal functions.[17] Improved outcome by suggested CGWO method over GWO, across all types of considered benchmark problems signifies the supremacy of said suggested method in terms of exploitation and exploration capability, which leads to smoother attainment of a global peak. In the conveyed consequence, the proposed CGWO method reveals enhanced magnitudes in comparison to GWO, setting up its better competency for acceptance globally. The consequences attained are illustrated in Table 4. The convergence curves of the CGWO approach over the GWO approach utilizing benchmark functions are presented in Fig. 1. The y-axis of the curve shows the results obtained for the standard functions, and the x-axis shows the number of possible repetitions of the experiment. CGWO's convergence curve is smoother than the GWO curve, indicating that it is more effective in reaching the optimal peak.

The extensive acceptance of any algorithm is possible only when it requires the least amount of time for its execution. Hence we have equated the worst-case time complexity of CGWO with GWO. The initialization of the populace takes $O(n * d)$, where n indicates populace size and d signifies the dimension of the problem. Fitness computation requires time as $O(n * d)$ time. So, worst case time complexity of CGWO is $O(n * d * Maxrecapitulations$, where Max_recapitulations denotes the maximum count of reiterations.

Comparison among Contemporary Metaheuristic Algorithms

There is a significant difference among the various metaheuristic algorithms in terms of design and the amount of control boundaries. Every algorithm has its controlling parameters and flow of control in terms of structural depiction. Contrasting the proposed CGWO and other metaheuristic techniques, there might be no impact but while finding the solutions for complex problems and assessing their achievements, there will be some unique nature will take place. These assessments will depend on some classic benchmark functions which are unimodal, composite, hybrid, and multimodal sets. With the assistance of these classic benchmark functions and by taking six techniques that are MFO[31], SCA[32], ALO[33], SSA[30], GWO[10,] and proposed CGWO, the results have been fixed. The proposed method affirms its incomparability over the remaining compared techniques. Acquired outcomes are represented in Table 5.

Non-Parametric Analysis for Performance Estimation

Non-parametric analysis is required to estimate the noteworthy distinction among various metaheuristic algorithms. The total insights of population size and no particular presumptions are required by this non-parametric test. In this current investigation, Friedman and Holm's test is accomplished. The Friedman test is a class of non-parametric assessment in nature, which is part of insights that do not rely exclusively upon defined groups of likelihood dissemination. It was first created by Milton Friedman. Like the parametric

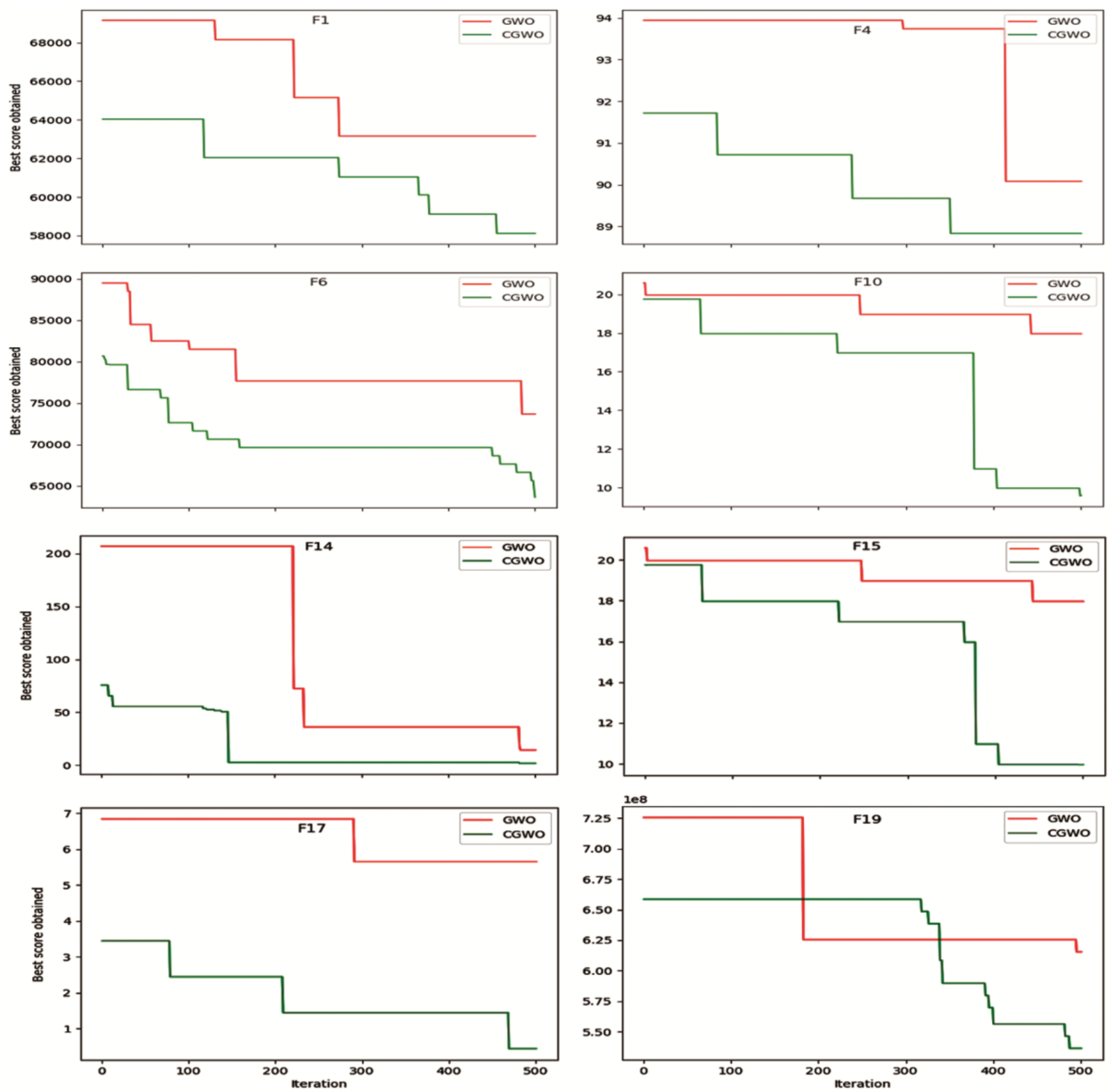

Fig. 1 — Convergence curve obtained from benchmark functions.

Table 4 — Mean, median, SD, and maximum error value, minimum error value acquired from benchmark functions. Improved consequences marked bold

Benchmark Functions	Algorithm	Mean	Median	Std. Dev.	Min Error	Max Error
1	GWO	7.71E+04	7.34E+04	6614.11	7.31E+04	8.47E+04
	CGWO	**6.90E+04**	6.80E+04	2754.02	6.69E+04	7.21E+04
2	GWO	5.85E+10	1.73E+10	772388	1.06E+10	1.48E+11
	CGWO	**4.01E+08**	4.39E+08	127022.1	2.59E+08	5.04E+08
3	GWO	1.87E+05	1.71E+05	43090.55	1.55E+05	2.36E+05
	CGWO	**1.41E+05**	1.43E+05	29324.29	1.11E+05	1.70E+05
4	GWO	9.32E+01	9.24E+01	2.35	9.14E+01	9.59E+01
	CGWO	**8.81E+01**	8.87E+01	2.54	8.53E+01	9.03E+01
5	GWO	2.82E+08	2.82E+08	335267.28	2.48E+08	3.16E+08
	CGWO	**2.41E+08**	2.24E+08	4612073.27	2.05E+08	2.93E+08
6	GWO	7.62E+04	7.94E+04	6529.21	6.87E+04	8.05E+04

(*Contd*.)

Table 4 — Mean, median, SD, and maximum error value, minimum error value acquired from benchmark functions. Improved consequences marked bold (*Contd.*)

Benchmark Functions	Algorithm	Mean	Median	Std. Dev.	Min Error	Max Error
	CGWO	**6.90E+04**	7.23E+04	11926.97	5.58E+04	7.89E+04
7	GWO	1.78E+02	1.75E+02	12.66	1.66E+02	1.91E+02
	CGWO	**1.55E+02**	1.60E+02	12.60	1.41E+02	1.65E+02
8	GWO	−4.04E+03	−5.42E+03	2445.83	−5.49E+03	−1.22E+03
	CGWO	**−4.13E+03**	−5.45E+03	2356.72	−5.53E+03	−1.41E+03
9	GWO	4.38E+02	4.38E+02	8.19	4.30E+02	4.46E+02
	CGWO	**4.25E+02**	4.27E+02	5.59	4.18E+02	4.28E+02
10	GWO	2.06E+01	2.06E+01	0.55	2.00E+01	2.11E+01
	CGWO	**2.05E+01**	2.05E+01	0.53	2.00E+01	2.10E+01
11	GWO	5.20E+02	5.20E+02	19.97	5.00E+02	5.40E+02
	CGWO	**5.13E+02**	5.17E+02	17.56	4.94E+02	5.28E+02
12	GWO	6.15E+08	5.95E+08	812238.67	5.45E+08	7.04E+08
	CGWO	**5.42E+08**	5.08E+08	1153578.9	4.48E+08	6.71E+08
13	GWO	1.54E+09	1.50E+09	2427020	1.32E+09	1.80E+09
	CGWO	**8.95E+08**	1.14E+09	7025109	1.03E+08	1.44E+09
14	GWO	1.53E+01	1.40E+01	5.25	1.08E+01	2.11E+01
	CGWO	**5.73E+00**	7.21E+00	3.76	1.45E+00	8.55E+00
15	GWO	6.14E-01	5.62E-01	0.20	4.40E-01	8.41E-01
	CGWO	**1.73E-01**	2.12E-01	0.07	8.23E-02	2.24E-01
16	GWO	2.50E+00	2.62E+00	1.49	9.50E-01	3.94E+00
	CGWO	**3.67E-01**	8.95E-01	0.91	−6.90E-01	8.97E-01
17	GWO	4.16E+00	3.28E+00	2.42	2.29E+00	6.90E+00
	CGWO	**1.68E+00**	1.16E+00	1.09	9.52E-01	2.94E+00
18	GWO	7.71E+01	1.01E+02	45.76	2.43E+01	1.05E+02
	CGWO	**6.95E+01**	9.00E+01	41.45	2.18E+01	9.68E+01
19	GWO	−6.39E-01	−3.00E-01	0.85	−1.62E+00	−1.01E-03
	CGWO	**−7.62E-01**	−3.13E-01	1.06	−1.97E+00	−7.09E-05
20	GWO	−3.30E-01	−3.15E-01	0.09	−4.35E-01	−2.39E-01
	CGWO	**−1.08E+00**	−1.05E+00	0.27	−1.37E+00	−8.24E-01
21	GWO	−4.07E-01	−4.78E-01	0.12	−4.80E-01	−2.64E-01
	CGWO	**−5.82E-01**	−6.42E-01	0.10	−6.48E-01	−4.55E-01
22	GWO	−5.27E-01	−5.37E-01	0.16	−6.87E-01	−3.57E-01
	CGWO	**−6.61E-01**	−7.10E-01	0.20	−8.35E-01	−4.39E-01
23	GWO	−6.30E-01	−6.55E-01	0.12	−7.35E-01	−4.99E-01
	CGWO	**−8.97E+02**	−1.21E+00	1551.31	−2.69E+03	−8.64E-01

Table 5 — Result acquired from proposed CGWO equated with contemporary metaheuristic techniques over considered 23 benchmark functions. Improved consequences marked bold

Benchmark Functions	MFO	SCA	ALO	GWO	CGWO
1	8.48E+4	8.27E+4	9.09E+4	7.71E+04	**6.90E+04**
2	6.18+10	5.99E+10	6.09E+10	5.85E+10	**4.01E+08**
3	2.12E+5	1.90E+5	1.64E+6	1.87E+05	**1.41E+05**
4	2.09E+2	1.69E+2	2.7E+2	9.32E+01	**8.81E+01**
5	2.03E+10	2.66E+9	1.99E+9	2.82E+08	**2.41E+08**
6	1.29E+5	8.61E+4	6.66E+5	7.62E+04	**6.90E+04**
7	**−2.01E+2**	**−2.01E+2**	−6.19E+0	1.78E+02	1.55E+02
8	−3.97E+3	−4.01E+3	−3.69E+3	−4.04E+03	**−4.13E+03**
9	4.69E+2	4.7E+3	3.6E+3	4.38E+02	**4.25E+02**
10	4.79E+1	4.03E+1	2.10E+1	2.06E+01	**2.05E+01**

(*Contd.*)

Table 5 — Result acquired from proposed CGWO equated with contemporary metaheuristic techniques over considered 23 benchmark functions. Improved consequences marked bold

Benchmark Functions	MFO	SCA	ALO	GWO	CGWO
11	5.98E+2	6.1E+2	5.19E+2	5.20E+02	**5.13E+02**
12	7.25E+8	6.22E+8	2.02E+9	6.15E+08	**5.42E+08**
13	1.7E+9	1.62E+9	9.05E+8	1.54E+09	**8.95E+08**
14	2.19E+1	2.24E+1	2.01E+1	1.53E+01	**5.73E+00**
15	1.1E-1	1.06E-1	1.13E-1	**6.14E-01**	1.73E-01
16	1.09E+2	3.25E+0	3.19E+0	2.50E+00	**3.67E-01**
17	6.11E+0	5.7E+0	4.2E+0	4.16E+00	**1.68E+0**
18	1.06E+2	1.01E+2	9.1E+1	7.71E+01	**6.95E+01**
19	−3.88E-1	−3.42E-1	−3.91E-1	−6.39E-01	**−7.62E-01**
20	−1.03E+0	−1.06E+0	−1.01E+0	−3.30E-01	**−1.08E+00**
21	−1.88E-1	−2.63E-1	−1.31E-1	−4.07E-01	**−5.82E-01**
22	−3.55E-1	−4.68E-1	−3.22E-1	−5.27E-01	**−6.61E-01**
23	−5.16E-1	−5.02E-1	−6.11E-1	−6.30E-01	**−8.97E+02**

rehashed measures such as ANOVA, it is utilized to identify the contracts in treatments over various test endeavours. This test is used to examine different results that came from various metaheuristic algorithms. The assumed null hypothesis is expressed as H_0 stated as, H_0 can be evaluated by every metaheuristic algorithm and there is no significant difference among all those optimizing techniques. The significance level $\propto$ is taken as 0.05 to evaluate the hypothesis. In this mathematical analysis, each metaheuristic approach is assigned a performance level that can be limited between the spectrums of 1 to T. Based on the outcome attained from the classic objective method, this ranking is distributed. The average performance level Avg_r can be calculated by using Eq. (9).

$$Avg_r = \frac{Total\ addition\ of\ all\ ranks\ attained\ from\ considered\ classifiers}{Total\ number\ of\ datasets} \quad \dots (9)$$

Friedman statistics can be calculated by using Eq. (10)

$$FD_s = \frac{(T-1)X_F^2}{T(CC-1)-(X_F^2)} \quad \dots (10)$$

where, $X_F^2 = \frac{12T}{CC*(CC+1)}\left[\sum_r (Avg_r)^2 - \frac{CC*(CC+1)^2}{4}\right]$

Here, T represents the number of benchmarked test functions and CC represents the number of metaheuristic optimization classifiers. The statistics of Friedman FD_s is propagated by consulting the F-distribution with $(CC-1)$ and $(CC-1)*(T-1)$ degree of freedom. In this, 23 classic benchmark functions and five optimization techniques have been engaged to calculate the degree of freedom which will be under the limit of 4 to 88. The witnessed value of $F(4, 88)$ with $\propto = 0.05$ is 5.6581.[34] If and only if the outcome of FD_s is lesser than the examined critical significance, then the assumed null hypothesis can be accepted. Otherwise, the null hypothesis will be rejected. If the null hypothesis is acknowledged, it represents that there exists no clear divergence among all the castoff techniques. The dismissal of the hypothesis specifies that some dissimilarity exists amongst all the chosen techniques. After implementing the Friedman test, if the rejection of the null hypothesis has ensued, some other test will be conducted which is termed as Holms strategy. The main objective of the Holms Test is to prove that the results of the implemented method are preferable in comparison with other optimizing techniques. Here, H_0 is used to express the null hypothesis. The combination of techniques that are compared executes uniformly. In this Holms Test, we need Y value. This Y value can be calculated using Eq. (11).

$$Y = \frac{AR_p - AR_c}{CV} \quad \dots (11)$$

where, $CV = \sqrt{\frac{CC(CC+1)}{6*T}}$

where, AR_p and AR_c signifies the average rank of proposed as well as compared classifiers and CV indicates computed value. Depending on the tabular form of the normal distribution that is depending on the calculated Y value the value of probability P_{val} is calculated.[35] Based on the outcome of the comparison between the P_{val} and $\frac{\propto}{(CC-i)}$ values, it will then be further decided whether the null hypothesis is recommended or excluded. If all the attained P_{val} will be smaller than $\frac{\propto}{(CC-i)}$values, then it can be described as the rejection of the hypothesis which means the proposed approach leads a guiding technique relating to competence. The average rank obtained from considered techniques and results observed from Holm's test are presented in Table 6 & 7.

Application of CGWO for Training MLP

In the space of artificial intelligence and computational insight, neural networks are the most significant method. In the year 1943, neural networks were first reported. Neural systems are an arrangement of calculations that refers to the neurons of a human cerebrum that are intended to predict unknown patterns. There are various types of Neural Networks in writing, for example, Learning Vector Quantization (LVQ), Spiking Neural Networks (SNN), Radial Basis Function (RBF) neural networks, Feed Forward Neural Networks (FFNN), and Higher Order Neural Networks (HONN). In this study Multilayer Perceptron (MLP) is introduced which is a type of feed-forward artificial neural network. MLP is an example of a supervised learning technique and is utilized for the grouping and regression of a few sorts of N-dimensional problems. In MLP, there should be at least 3 layers of nodes and those are the input layer, a hidden layer, and an output layer. The hidden layer is the transitional layer, which is available in the middle of the input layer and yield layer. MLP's major task is to use an activation function to process the input that was handled by the previous layer. The variable (ω) is a wolf has a higher tendency toward (α), (β), and (δ), respectively. This means that the next position of (ω) is a wolf is close to α and β. That is, the contribution α-wolf in redefining weights and

Table 6 — Rank acquired from proposed CGWO and contemporary metaheuristic techniques over considered 23 benchmark functions. Improved consequences marked bold

Castoff Function	MFO	SCA	ALO	GWO	CGWO
1	8.48E+04(4)	8.27E+04(3)	9.09E+04(5)	7.71E+04(2)	**6.90E+04**(1)
2	6.18+10(5)	5.99E+10(3)	6.09E+10(4)	5.85E+10(2)	**4.01E+08**(1)
3	2.12E+05(4)	1.90E+05(3)	1.64E+06(5)	1.87E+05(2)	**1.41E+05**(1)
4	2.09E+02(4)	1.69E+02(3)	2.7E+02(5)	9.32E+01(2)	**8.81E+01**(1)
5	2.03E+10(5)	2.66E+09(3)	1.99E+09(4)	2.82E+08(2)	**2.41E+08**(1)
6	1.29E+05(4)	8.61E+04(3)	6.66E+05(5)	7.62E+04(2)	**6.90E+04**(1)
7	**−2.01E+02**(1.5)	**−2.01E+02**(1.5)	−6.19E+00(3)	1.78E+02(5)	1.55E+02(4)
8	−3.7E+03(5)	−4.01E+03(3)	−3.9E+03(4)	−4.04E+03(2)	**−4.13E+03**(1)
9	4.69E+02(3)	4.7E+03(5)	3.6E+03(4)	4.38E+02(2)	**4.25E+02**(1)
10	4.79E+01(5)	2.10E+01(3.5)	2.10E+01(3.5)	2.06E+01(2)	**2.05E+01**(1)
11	5.98E+02(4)	6.1E+02(5)	5.19E+02(2)	5.20E+02(3)	**5.13E+02**(1)
12	7.25E+08(4)	6.22E+08(3)	2.02E+09(5)	6.15E+08(2)	**5.42E+08**(1)
13	1.7E+09(5)	1.62E+09(4)	9.05E+08(2)	1.54E+09(3)	**8.95E+08**(1)
14	2.19E+01(4)	2.24E+01(5)	2.01E+01(3)	1.53E+01(2)	**5.73E+00**(1)
15	1.1E-01(4)	1.06E-01(5)	1.13E-01(3)	**6.14E-01**(1)	1.73E-01(2)
16	1.09E+02(5)	3.25E+00(4)	3.19E+00(3)	2.50E+00(2)	**3.67E-01** (1)
17	6.11E+00(5)	5.7E+00(4)	4.2E+00(3)	4.16E+00(2)	**1.68E+00**(1)
18	1.06E+2(5)	1.01E+2(4)	9.1E+01(3)	7.71E+01(2)	**6.95E+01**(1)
19	−3.88E-01(4)	−3.42E-01(3)	−3.91E-01(5)	−6.39E-01(2)	**−7.62E-01**(1)
20	−0.03E+00(4)	−1.06E+00(5)	−1.01E+00(3)	−3.30E-01(2)	**−1.08E+00**(1)
21	−1.88E-01(4)	−2.63E-01(3)	−1.31E-01(5)	−4.07E-01(2)	**−5.82E-01**(1)
22	−3.55E-01(4)	−4.68E-01(3)	−3.22E-01(5)	−5.27E-01(2)	**−6.61E-01**(1)
23	−5.16E-01(4)	−5.02E-01(5)	−6.11E-01(3)	−6.30E-01(2)	**−8.97E+00**(1)
$\boldsymbol{Avg_r}$	4.19	3.65	3.80	2.17	**1.17**

Table 7 — Outcomes generated from Holm's Test

Rank	Classifiers	$Y - value$	P_{val}	$\frac{\propto}{(CC-i)}$	Hypothesis
1	MFO	−6.47	0.00004	0.0125	Disallowed
2	SCA	−5.31	0.00004	0.0166	Disallowed
3	ALO	−5.64	0.00004	0.025	Disallowed
4	GWO	−2.14	0.0157	0.05	Disallowed

biases ω Wolves is the best. Hence the Weights and biases are updated and approach optimality leading to enhanced MLP in each iteration. The concept of concern is used to fix the optimum quantity of neurons and hidden layers. In this, the learning task maps an input layer to an output layer based on example input-output pairs. MLP assesses the training data and produces a derived function that is used to map upcoming examples. The goal of MLP is to correctly determine the class labels for unseen data. The weights of individual connections in MLP can be modified during training. The main effort of MLP trainers is to train the MLPs by looking for the ideal loads and to get the maximum accuracy for incomprehensible examples of given inputs. To gain exactness, we have to fine-tune the parameters of the MLP after a specified number of reiterations. After the training process, this model is very useful for predicting unknown patterns. The architecture of MLP is depicted in Fig. 2.

The output can be measured by employing Eq. (12).

$$a_k = \sum_{i=1}^{d} W_{ki} * X_i + e_k, k = 1,2,\ldots m \quad \text{... (12)}$$

where, the value of m infers the direct amalgamations of constraints that fit the 1st layer, d connotes d amount of magnitudes as input and a_k refers to the consequence from i^{th} hidden neuron, W_{ki} refers to associated weights, X_i refers i^{th} amount of inputs and e_k refers to biases. The diverse initiations transform by a single activation function which is considered to be sigmoid and depicted in Eqs. (13–15).

$$Sd_i = \frac{1}{(1+exp(-a_k)} \quad \text{... (13)}$$

Ultimately the output can be calculated as,

$$a_h = \sum_{i=1}^{h}(W_{ih} * a_i) + e_h, h = 1,2,\ldots,m \quad \text{... (14)}$$

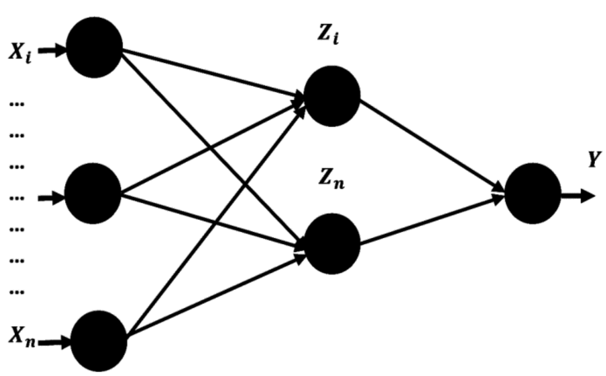

Fig. 2 — Elementary Arrangement of MLP

where, W_{ih} refers to weights from hidden nodes to h^{th} yield node.

$$O_n = \frac{1}{(1+exp(-a_h)} \quad \text{... (15)}$$

The fundamental concern with gradient and other traditional formulating algorithms is that they can become stuck in confined optima and have a sluggish convergence rate, prompting the creation of metaheuristic-based neural network formulating algorithms. The MLP is trained using the suggested CGWO algorithm. The weights and biases are initially selected randomly which are required for the parameters for MLP. To get superlative accuracy in terms of classification, we have to pass a set of initial values which are preferred arbitrarily. These weights and biases can be identified using Eq. (16-17).

$$\vec{W} = W_{1,1}, W_{1,2}, W_{1,3}, \ldots, W_{i,j} \quad \text{... (16)}$$

$$\vec{B} = b_1, b_2, b_3, \ldots, b_n \quad \text{... (17)}$$

where, $W_{i,j}$ signifies weights and b_n signifies biases. Afterward providing the preliminary constraints, we must develop the set of arbitrarily produced constraint lists numerous times to calculate the efficacy of MLP. After measuring MLP's efficacy, the Root-Mean-Square-Error (RMSE) can be computed. The gap between the actual value acquired from the model, i.e. MLP, and the goal value after completing the required number of iterations can be calculated as RMSE. Aside from RMSE, features such as average RMSE and standard deviation may be computed from RMSE values, and these can be obtained by iterating the parameters repeatedly throughout the MLP. The organization of MLP for dissimilar datasets is presented in Table 8.

To ensure a fair distinction between the metaheuristic algorithms under consideration, we have established a search agent count of 20 and a maximum iteration count of 100. To achieve a minimum RMSE, we have taken into account 10 distinct outcomes. The average and standard deviation were computed by analysing the 10 RMSE values obtained. The efficiency of the proposed model has been assessed by using six customary datasets, chosen from the UCI storehouse to solve practical problems.[36] Those datasets are Liver, Cancer, Balloon, Iris, Diabetes, and Heart. Here, an improved CGWO is used to train MLP. The results have equated with some other metaheuristic algorithms those are DE, GA, GWO, PSO, and SSA.

Simultaneously, the outcome is affirmed with CGWO alongside other available algorithms, concerning diverse estimates like min, RMSE, average, and Std. Dev, error rate, sensitivity, specificity, accuracy, and prevalence. Improved accuracy is evidence about the supplemented ability of CGWO, regarding better prevention of local optima trap. Lower RMSE, STD, and higher average prove the better exploitable strength. The accomplished outcomes are given in Tables 9 & 10.

Table 8 — Depiction of Customary datasets and MLP construction

Castoff Dataset	Attributes Count	Training Samples Count	Test Samples Count	Count of classes	MLP structure
Iris	4	150	30	3	4-4-1
Balloon	4	20	16	2	9-9-1
Cancer	9	683	120	2	8–8-1
Heart	13	270	60	2	13-13–1
Liver	6	345	70	2	4-4-1
Diabetes	8	768	150	2	6-6-1

Table 9 — Performance Assessment of CGWO concerning DE, GWO, GA, PSO, and PSO

Dataset/Algorithm		CGWO	DE	GA	GWO	PSO	SSA
Liver	Optimum RMSE	**0.4875**	0.5001	0.4901	0.4977	0.5001	0.5011
	AVG	0.5558	0.2213	0.2200	0.2001	0.2123	0.3011
	ST. DEV	0.0568	0.0201	0.0100	0.0220	0.0212	0.0401
	SENSITIVITY	0.6	0.7001	0.6201	0.9102	0.8076	0.7021
	SPECIFICITY	0.7	0.4999	0.5311	0.5001	0.5001	0.5911
	PREVALENCE	45.58	34.11	31.01	39.99	42.22	32.99
Cancer	Optimum RMSE	0.3546	**0.2632**	0.2798	0.3102	0.3332	0.3812
	AVG	0.4239	0.2901	0.2902	0.4102	0.4021	0.3998
	ST. DEV	0.0412	0.0198	0.0040	0.0499	0.0301	0.0213
	SENSITIVITY	1	1	1	1	1	1
	SPECIFICITY	0.8	0.8999	0.9234	0.9234	0.9234	0.9234
	PREVALENCE	56.66	50	50	50	50	50
Heart	Optimum RMSE	**0.3642**	0.5012	0.5002	0.4001	0.4796	0.4901
	AVG	0.5202	0.4922	0.4734	0.3931	0.4818	0.4821
	ST. DEV	0.0129	0.0101	0.0040	0.0088	0.0034	0.0090
	SENSITIVITY	0.85	0.8030	0.8222	0.8622	0.8799	0.7865
	SPECIFICITY	0.7	0.75	0.81	0.76	0.67	0.831
	PREVALENCE	53.70	41.01	39.98	41.88	45.33	39.01
Diabetes	Optimum RMSE	0.4591	**0.4511**	0.4666	0.5001	0.4609	0.5101
	AVG	0.6364	0.4444	0.4611	0.4866	0.4698	0.4701
	ST. DEV	0.0747	0.0030	0.0050	0.0080	0.0078	0.0041
	SENSITIVITY	0.9	0.7984	0.7912	0.7935	0.7946	0.7988
	SPECIFICITY	0.4	0.6811	0.6199	0.6511	0.5803	0.6911
	PREVALENCE	71.42	41.22	43.23	42.13	44.01	42.13
Balloon	Optimum RMSE	**0.2915**	0.4311	0.5021	0.3923	0.4199	0.4999
	AVG	0.3213	0.2233	0.5011	0.4124	0.4498	0.5113
	ST. DEV	0.094	0.0202	0.0032	0.0298	0.0201	0.0099
	SENSITIVITY	0.8	1	1	1	1	1
	SPECIFICITY	1	1	1	1	1	1
	PREVALENCE	40	50	50	50	50	50
Iris	Optimum RMSE	**0.1755**	0.2001	0.1999	0.1818	0.2166	0.2323
	AVG	0.4793	0.2199	0.2122	0.1999	0.2298	0.2802
	ST. DEV	0.0112	0.0201	0.0049	0.0133	0.0171	0.0381

Table 10 — Accuracy consequences attained from customary datasets

Algorithm/Dataset		DE	GA	PSO	SSA	GWO	CGWO
Liver	Error-Rate (%)	40.2	41.4	37.3	38.3	37.3	27.9411
	Accuracy (%)	59.8	58.6	62.7	61.7	62.7	**72.0588**
Cancer	Error-Rate (%)	3.2	3.2	3.2	3.2	3.2	6.6666
	Accuracy (%)	96.8	96.8	96.8	96.8	**96.8**	93.3333
Heart	Error-Rate (%)	23.2	20.3	22.2	15.9	20.3	15.5
	Accuracy (%)	76.8	79.7	77.8	84.1	79.7	**84.5**
Diabetes	Error-Rate (%)	26.7	26.4	27.8	25.3	26.4	24
	Accuracy (%)	73.3	73.6	72.2	74.7	73.6	**76**
Balloon	Error-Rate (%)	0	0	0	0	0	10.00
	Accuracy (%)	100	100	100	100	**100**	90.00
Iris	Error-Rate (%)	24.7	14.7	10	3.3	3.3	3.3
	Accuracy (%)	75.3	85.3	90	96.7	96.7	**96.7**

Conclusions

The introduced CGWO is an extension of GWO that integrates the concepts of chaos theory into GWO. Its effectiveness is confirmed by using it to solve the constrained problems set. It also performs a comparative assessment against several recently introduced meta-heuristics, such as PSO, GE, GWO, and SSA, to examine overall performance levels. The findings of the experiments reveal that using deterministic chaotic signals instead of linearly declining values is a significant modification of the GWO method, which outperforms other cast-off procedures. Based on the experimental consequences of various benchmark functions, we find that the proposed method performs well in most cases and achieves an improved convergence rate and accuracy compared to SSA and other considered methods. Moreover, performing statistical analysis using a variety of statistical tests confirms significant differences between results obtained with CGWO and some state-of-the-art techniques. Further, the suggested CGWO is cast off with MLP as a suitable classifier, for benchmark data classification. Observed outcomes signify its effectiveness as a better trainer algorithm, in terms of reduced error rate and hence improved accuracy.

In the future, the suggested CGWO can be utilized to resolve discrete optimization problems, as well as to train other higher-order neural networks to serve as a superior trainer algorithm and to analyse image data with CNN.

References

1 Dhiman G, SSC: A hybrid nature-inspired meta-heuristic optimization algorithm for engineering applications, *Knowl-Based Syst*, **222** (2021) 1–17, https://doi.org/10.1016/j.knosys.2021.106926.

2 Uthayakumar J, Vengattaraman T & Dhavachelvan P, Swarm intelligence-based classification rule induction (CRI) framework for qualitative and quantitative approach: An application of bankruptcy prediction and credit risk analysis, *J King Saud Univ Comput*, **32(6) (**2020) 647–57, https://doi.org/10.1016/j.jksuci.2017.10.007.

3 Panda N, Majhi S K & Pradhan R, A hybrid approach of spotted hyena optimization integrated with quadratic approximation for training wavelet neural network, *Arab J Sci Eng*, **47(8)** (2022) 10347–363, https://doi.org/10.1007/s13369-022-06564-4.

4 Teng A, Peng L, Xie Y, Zhang H & Chen Z, Gradient descent evolved imbalanced data gravitation classification with an application on internet video traffic identification, *Inf Sci*, **539** (2020) 447–60, https://doi.org/10.1016/j.ins.2020.05.141.

5 Karaoglan A D, Optimizing plastic extrusion process via grey wolf optimizer algorithm and regression analysis, *J Sci Ind Res*, **80(01)** (2021) 34–41, doi:10.56042/jsir.v80i01.35559.

6 Mahapatra A K, Panda N & Pattanayak B K, Hybrid PSO (SGPSO) with the Incorporation of discretization operator for training RBF neural network and optimal feature selection, *Arab J Sci Eng*, **48(8)** (2023) 9991–10019, https://doi.org/10.1007/s13369-022-07408-x.

7 Mahapatra A K, Panda N & Pattanayak B K, Quantized Salp Swarm Algorithm (QSSA) for optimal feature selection, *J Inf Technol*, **15(2)** (2023) 725–734, https://doi.org/10.1007/s41870-023-01161-6.

8 Apat S K, Mishra J, Raju K S & Padhy N, An artificial intelligence-based crop recommendation system using machine learning, *J Sci Ind Res*, **82(05)** (2023) 558–567, https://doi.org/10.56042/jsir.v82i05.1092.

9 Yadav V, Rahul M & Yadav R, A New Efficient Method for the detection of intrusion in 5 G and beyond Networks using Machine Learning, *J Sci Ind Res*, **80(01)** (2021) 60–50, doi:10.56042/jsir.v80i01.38894.

10 Mirjalili S, Mirjalili S M & Lewis A, Grey wolf optimizer, *Adv Eng Softw*, **69** (2014) 46–61, https://doi.org/10.1016/j.advengsoft.2013.12.007.

11 Mirjalili S, How effective is the Grey Wolf optimizer in training multi-layer perceptrons, *Appl Intell*, **43** (2015) 150–161, https://doi.org/10.1007/s10489-014-0645-7.

12 Mirjalili S, Saremi S, Mirjalili S M & Coelho L D, Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization, *Expert Syst Appl*, **47** (2016) 106–119, https://doi.org/10.1016/j.eswa.2015.10.039.

13 Sharma Y & Saikia L C, Automatic generation control of a multi-area ST–Thermal power system using grey wolf optimizer algorithm based classical controllers, *Int J Electr Power Energy Syst*, **73** (2015) 853–862, https:// doi.org/ 10.1016/j.ijepes.2015.06.005.

14 Rodríguez L, Castillo O, Soria J, Melin P, Valdez F, Gonzalez CI, Martinez G E & Soto J, A fuzzy hierarchical operator in the grey wolf optimizer algorithm, *Appl Soft Comput*, **57** (2017) 315–328, https://doi.org/10.1016/ j.asoc.2017.03.048.

15 Song X, Tang L, Zhao S, Zhang X, Li L, Huang J & Cai W, Grey wolf optimizer for parameter estimation in surface waves, *Soil Dyn Earthq Eng*, **75** (2015) 147–57, https://doi.org/10.1016/j.soildyn.2015.04.004.

16 Kohli M & Arora S, Chaotic grey wolf optimization algorithm for constrained optimization problems, *J Comput Des Eng*, **5(4)** (2018) 458–472, https://doi.org/10.1016/ j.jcde.2017.02.005.

17 Saxena A, Kumar R & Das S, β-chaotic map enabled grey wolf optimizer, *Appl Soft Comput*, **75** (2019) 84–105, https://doi.org/10.1016/j.asoc.2018.10.044.

18 Mitić M, Vuković N, Petrović M & Miljković Z, Chaotic metaheuristic algorithms for learning and reproduction of robot motion trajectories, *Neural Comput Appl*, **30** (2018) 1065–1083, https://doi.org/10.1007/s00521-016-2717-6.

19 Heidari A A & Abbaspour R A, Enhanced chaotic grey wolf optimizer for real-world optimization problems: A comparative study, *Handbook of Research on Emergent Applications of Optimization Algorithms, IGI Global*, (2018), 693–727, doi:10.4018/978-1-5225-2990-3.ch030.

20 Panda N & Majhi S K, Design and applications of improved metaheuristic algorithms for neural network training, *Intelligent Technologies: Concepts, Applications, and Future Directions*, (2022) 223–255, https://doi.org/10.1007/978-981-19-1021-0_10.

21 Chandra Sekhar C, Panda N, Ramana B V, Maneesha B & Vandana S, Effectiveness of backpropagation algorithm in healthcare data classification, *Green Technology for Smart City and Society: Proceedings of GTSCS 2020*, (2020) 289–298, https://doi.org/10.1007/978-981-15-8218-9_25.

22 Dalai A K, Jena A K, Ramana B V, Maneesha B & Panda N, Supervised machine learning approaches for medical data classification, *2022 2nd Int Conf Artif Intell Signal Process (AISP)*, (2022) 1–8, 10.1109/AISP53593.2022.9760688.

23 Jagadeeswara Rao G, Siva Prasad A, Sai Srinivas S, Sivaparvathi K & Panda N, Data classification by ensemble methods in machine learning, *Adv in Intell Comput Commun: Proc ICAC 2021*, (2022) 127–135, https://doi.org/10.1007/ 978-981-19-0825-5_13.

24 Mohakud R & Dash R, Designing a grey wolf optimization based hyper-parameter optimized convolutional neural network classifier for skin cancer detection, *J King Saud Univ Comput*, **34(8)** (2022) 6280–6291, https://doi.org/ 10.1016/j.jksuci.2021.05.012.

25 Mischaikow K & Mrozek M, Chaos in the Lorenz equations: a computer-assisted proof, *Bull Am Math Soc*, **32(1)** (1995) 66–72, https://doi.org/10.1090/S0273-0979-1995-00558-6.

26 Firth W J, Chaos--predicting the unpredictable, *BMJ-BRIT MED J*, **303** (1991) 1565–1568, doi:10.1136/bmj. 303.6817.1565.

27 Ditto W & Munakata T, Principles and applications of chaotic systems, *Commun ACM*, **38(11)** (1995) 96–102, https://doi.org/10.1145/219717.219797.

28 Selvam A M, Nonlinear dynamics and chaos: Applications in atmospheric sciences, *J Adv Math Appl*, **1(2)** (2012) 181–205, https://doi.org/10.1166/jama.2012.1014.

29 Safonov L A, Tomer E, Strygin V V, Ashkenazy Y & Havlin S, Multifractal chaotic attractors in a system of delay-differential equations modeling road traffic, *Chao*, **12(4)** (2002) 1006–1014, https://doi.org/10.1063/1.1507903.

30 Mirjalili S, Gandomi A H, Mirjalili S Z, Saremi S, Faris H & Mirjalili S M, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, *Adv Eng Softw*, **114** (2017) 163–91, https://doi.org/10.1016/j.advengsoft. 2017.07.002.

31 Mirjalili S, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowl-based Syst*, **89** (2015) 228–249, https://doi.org/10.1016/j.knosys. 2015.07.006.

32 Mirjalili S, SCA: a sine cosine algorithm for solving optimization problems, *Knowl-based Syst*, **96** (2016) 120–33, https://doi.org/10.1016/j.knosys.2015.12.022.

33 Mirjalili S, The ant lion optimizer, *Adv Eng Soft*, **83** (2015) 80–98, https://doi.org/10.1016/j.advengsoft.2015.01.010.

34 F Distribution Table, March 2018. Retrieved from http://www.socr.ucla.edu/applets.dir/f_table.html.

35 Normal Distribution Table. Retrieved from http:// math.arizona.edu/~rsims/ma464/standardnormaltable.pdf.

36 Bache K & Lichman M, UCI Machine Learning Repository [http://archive. ics. uci. edu/ml]. Irvine, CA: University of California, School of information and computer science, 28 (2013).