



A competitive learning-based Grey wolf Optimizer for engineering problems and its application to multi-layer perceptron training

Vamsi Krishna Reddy Aala Kalananda¹ · Venkata Lakshmi Narayana Komanapalli¹

Received: 22 December 2021 / Revised: 2 September 2022 / Accepted: 13 March 2023 /
Published online: 22 March 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

This article presents a competitive learning-based Grey Wolf Optimizer (Clb-GWO) formulated through the introduction of competitive learning strategies to achieve a better trade-off between exploration and exploitation while promoting population diversity through the design of difference vectors. The proposed method integrates population sub-division into majority groups and minority groups with a dual search system arranged in a selective complementary manner. The proposed Clb-GWO is tested and validated through the recent CEC2020 and CEC2019 benchmarking suites followed by the optimal training of multi-layer perceptron's (MLPs) with five classification datasets and three function approximation datasets. Clb-GWO is compared against the standard version of GWO, five of its latest variants and two modern meta-heuristics. The benchmarking results and the MLP training results demonstrate the robustness of Clb-GWO. The proposed method performed competitively compared to all its competitors with statistically significant performance for the benchmarking tests. The performance of Clb-GWO the classification datasets and the function approximation datasets was excellent with lower error rates and least standard deviation rates.

Keywords Competitive learning-based Grey wolf Optimizer (Clb-GWO) · Grey wolf Optimizer (GWO) · CEC2020 and CEC2019 · Multi-layer perceptron training

1 Introduction

1.1 Introduction to meta-heuristics

Optimization through meta-heuristics has emerged as a prominent trend for problem-solving and systematic resource management in multi-disciplinary research and real-world scenarios with

✉ Venkata Lakshmi Narayana Komanapalli
vlnarayana.k@vit.ac.in

Vamsi Krishna Reddy Aala Kalananda
vamsikrishna.ak2022@gmail.com

¹ School of Electrical Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India

numerous applications. Optimization has been adopted and encouraged by researchers and experts apprehensive of its simplicity and efficacy in solving complex problems with a greater degree of success [1, 10, 11, 13, 36, 45]. The development of Genetic algorithm (GA) [38], Particle swarm optimization (PSO) [74] marks a watershed in the history of optimization with a myriad of other techniques to follow soon. The optimization techniques iterate sequentially to determine the best optimal solutions concerning the objective function as it explores a complex labyrinth of peaks and valleys known as the “*search-landscape/search space*”. With a fraction of the knowledge required to determine the solutions while considering the ambivalent state of the problems, optimization is a promising aberration and has been at the forefront of the many-sided research avenues that continue to thrive toward the perfection of the existing and forthcoming systems [45].

1.2 A review of meta-heuristics in multimedia applications and artificial intelligence

Literature in recent times [11, 23, 71] provides an outlook on the widespread applications of meta-heuristic-based stochastic optimizers in multimedia tools and artificial intelligence (AI). The adaption of meta-heuristic solvers coupled with IoT [58] in big-data data analytics [52], blockchain [11], video games [10], artificial intelligence [24], feature selection [15], machine learning [14, 53], and deep learning has gained immense popularity on account of its simplicity, accuracy in training and testing and robustness to entrapment [7, 58]. Figure 1 provides a classification of various areas within the realm of multimedia and AI adopting meta-heuristic algorithms.

Moreover, multiple other domains including medicine and health sectors have been increasingly relying on the utilization of AI and multimedia tools to improve their accuracy and efficiency while working with large datasets. A few examples from the literature lately include the development of a multi-feature fusion convolution neural network (CNN) framework to represent the complex morphology and gene expression patterns [68], fall prediction based on key points of human bones through bone map dataset and CNN to prevent the damage to the elderly [70] etc. Computing systems for image and vision integrate AI to enhance the detection capabilities and these include a neural network-based edge-oriented framework for saliency detection enhancement for complex images [69], Deep Convolutional Generative Adversarial Networks (DCGAN) with TensorFlow deep learning framework for virtual face generation [31] etc. The financial engineering domain with AI-based prediction has been of increasing concern as they help to track and predict the trends of various markets and effectively design strategic products and services to maximize profits. For example, the research at [67] developed a novel mobile personalized recommended method based on the money flow model for the stock

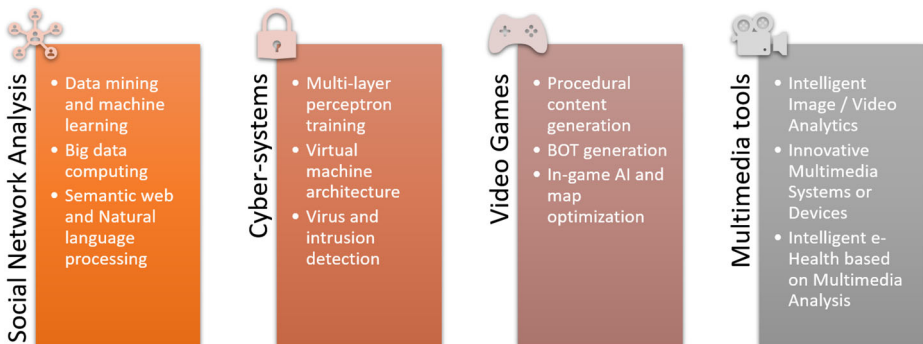


Fig. 1 Application of meta-heuristic optimization to various domains in multimedia and AI

exchange to provide investors with reliable practical investment guidance and receive more returns. Machine learning strategies based on two-dimensional numerical models in financial engineering [66] to reduce the prediction error and improve forecasting precision for major U.S. stock market index fall under the same category.

The efficacy of meta-heuristics in multimedia tools and AI has been well-researched and documented in the literature. Compared to the traditional solvers such as the gradient-descent method or back-propagation method where the initial solutions play a crucial role in the outcome of optimization, the stochasticity of meta-heuristics forms its major strength in exploring the various possibilities of solution combinations with very little to no dependence on the initial guess. Examples include (i) a continuation approach for training Artificial Neural Networks (ANNs) with meta-heuristics by J. R-Delado et al. in [55] including Particle Swarm Optimization (PSO), Firefly Algorithm (FA) and Cuckoo Search. The execution times were lowered by about 5–30% without statistically significant loss of accuracy for the public benchmark datasets and this was achieved by the accelerated convergence of the meta-heuristics. (ii) Automated fine-tuning compiler heuristics through meta-optimization and machine learning to reduce compiler design complexity and tedium of heuristic tuning is implemented in [61]. The resulting framework improved the average speedup of the heuristic compilation by 23% with an average performance improvement of 25% on the training set and 9% on the test set. (iii) In other works, ANN training integrating a hybrid meta-heuristic combining the exploration and exploitation capabilities of Invasive Weed Optimization and Differential Evolution (DE) was realized in [46]. Benchmarked against the 5 and 10-layered multi-layer perceptron training, the hybrid algorithm lowered the training and testing errors by about 5 to 10% for the three different datasets with a faster rate of convergence. (iv) In a similar development, S. Benabderrahmane in 2017 [5], combined machine learning and swarm intelligence for real-time object detection and tracking to accelerate time processing and enhance the extraction efficiency of the classifier. Experimenting with genetic algorithms (GA), particle swarm optimization (PSO), random walk and a novel hybrid combination of these methods, significant improvements were observed in computation time, efficiency and accuracy. (v) Interactive software design incorporating meta-heuristic algorithms for search engines with user-provided evaluation and rating systems to develop Interactive Evolutionary Algorithms (IEAs) is implemented in [59]. A comparative analysis between greedy local search, an evolutionary algorithm and ant colony optimization (ACO) showed that ACO-based interactive search outperformed the latter for the software design problem.

Examples of integration of meta-heuristics for the optimization of multimedia tools include (i) An optimized network flow wavelet-based image coding for multipath selection to maximize the received multiple description coding (MDCs) in a lossy network model in [28]. The multi-objective optimization problem was tackled using GA and PSO-based simulations for various random network topologies. PSO delivered the most optimal routings with reduced packet loss and increased throughput. (ii) In other works, a video watermarking scheme for anti-piracy protection incorporates Squirrel Search Algorithm (SSA) with constraints on video quality and other thresholds [6]. The embedded watermarking scheme utilized the frame selection method on five different videos and the proposed SSA-based framework recorded a Peak Signal to Noise ratio (PSNR) of 71.06 dB outperforming eight methods from the literature. (iii) In similar developments, phishing website detection integrating a support vector machine and an Improved spotted hyena optimization (ISHO) algorithm is proposed to select proper features for classifying phishing websites [56]. Compared to PSO, FA and bat

algorithm-based SVM classifiers, the ISHO-based SVM achieved higher classification accuracy compared to others.

1.3 Challenges associated with meta-heuristic algorithms

Although the efficacy of the swarm-based nature-inspired optimization algorithms is imputed to its multifarious search mechanisms, the same mechanisms are prone to a myriad of problems and complications that require addressal for these algorithms to maximize their potential. The search mechanisms devised to mimic the processes in nature, which account for the core of any nature-inspired meta-heuristic, may not be competent at optimizing every class of optimization problem. Even though they are meticulously crafted, the crucial and conflicting case of exploration and exploitation has been eluding researchers and motivating them towards the realization of a near-perfect optimization strategy. Grounding on this, lately, there has been a gold rush for the development of better/improved variants of nature-inspired algorithms best suited to the search landscape of the problem being dealt with. Simultaneously, there is an upsurge in the publications relating to the development of better/improved variants tackling the following aspects.

- The deterioration in the performance (stagnation of fitness or sluggish convergence characteristics) of an optimization technique with the increasing number of problem dimensions is more often than not ascribed as “*the curse of dimensionality*”, coined by Richard E. Bellman. The manifold reason is that there could be several possibilities of every decision variable for each combination of values and the fitness of all such possibilities are to be computed within a present number of function evaluations resulting in solutions very far from the global optimum.
- The swarm intelligent optimization algorithms are inherent to the conflicting case of the balancing of exploration (global search) and exploitation (local search). An ideal trade-off between exploration (diversification) and exploitation (intensification) is needed such that the algorithm is capable of understanding the condition to explore further or improve the existing solutions [2, 23].
- Another issue managed through the improvement of meta-heuristics is the near-perfect coordination of the tuning criterion (otherwise called “*algorithm-specific parameters*”) and, notably, the requirement to tune several of such parameters to extract the best possible performance is often a tedious and time-consuming one and improper or inappropriate tuning has often been the major reason for the algorithms’ failure.
- Algorithms excelling at unconstrained cases may not perform equally well for a constrained problem and similarly, algorithms with quick convergent characteristics may not deliver the best optimality compared to others. Furthermore, algorithms designed to explore efficiently over complex landscapes may not be efficient at local search and vice-versa. This is commonly alluded to as the “*No free lunch theory*” [64], which expresses that the perfect optimization algorithm is not practically realizable and no meta-heuristics can deliver the best performance for every optimization task.

To address the aforementioned issues and extract the best performance for the chosen optimization problem, several improved/upgraded meta-heuristics have been proposed and have gained significance owing to their superior performance in terms of optimality, consistency and robustness [8]. The improvement of any meta-heuristic algorithm for the specific application is predominantly done through the introduction and empirical establishment of special techniques/operators that advance the exploration to newer areas within the search

space while at the same time balancing the exploitation/local search. This procedure of altering a meta-heuristic to upgrade its existing abilities for better and more robust performance is known as “improving” or “enhancing” or “modifying” [9]. Researchers frequently turn towards improvisation and improving the existing meta-heuristics to guarantee that a near-perfect compromise between the exploration and exploitation is achieved to a good extent such that the need for the tedious tuning process is reduced through adaptive techniques that are capable of dynamically adapting to the search landscape and also work towards restructuring them to eliminate the algorithms’ weaknesses in dealing with complex optimization tasks. This is represented in Fig. 2.

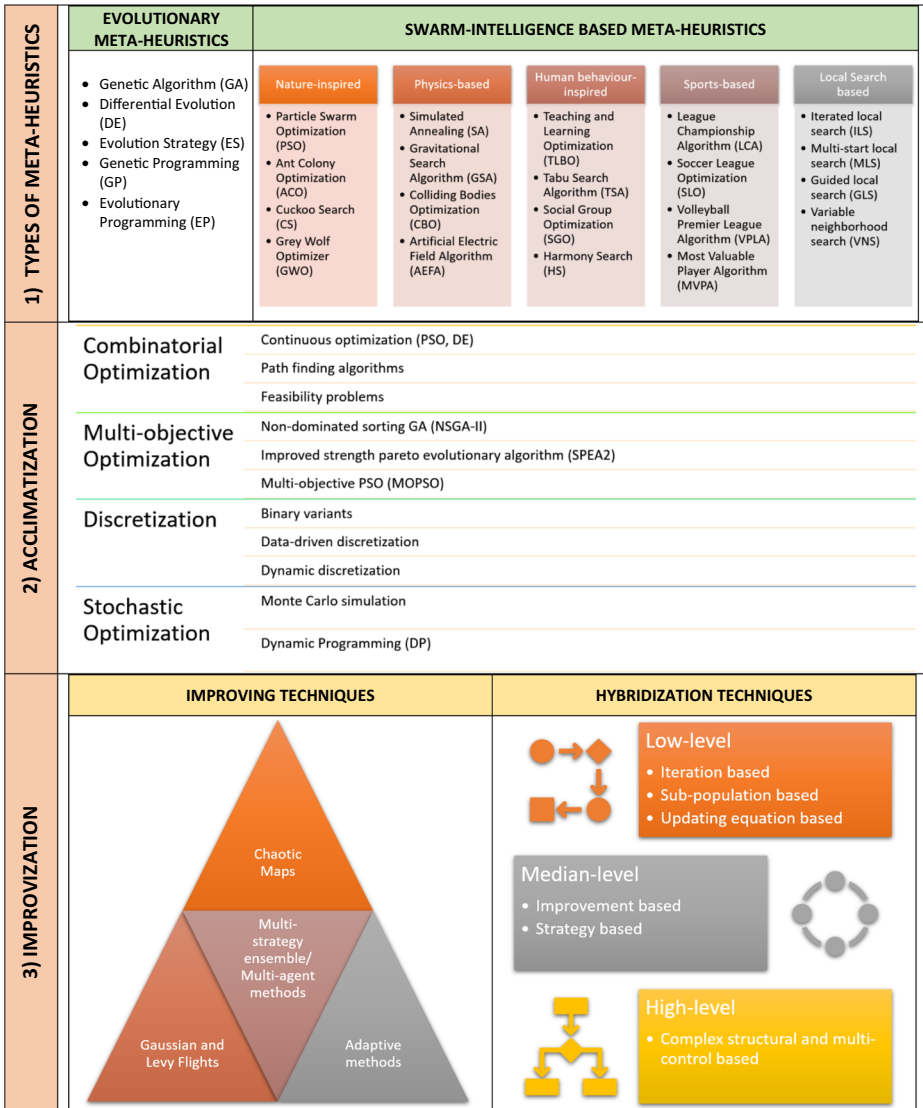


Fig. 2 Infographic depicting the classification, acclimatization and improvisation in meta-heuristic optimization algorithms

1.4 Contributions of the proposed work

In this work, Grey Wolf Optimizer (GWO) has been studied extensively with its merits and demerits have been analysed and an improved version is realised to overcome the various shortcomings associated with it. The proposed algorithm has been named Competitive-learning GWO (Clb-GWO) and it employs four major modifications.

1. Dual search mechanisms with new techniques for global and local search have been devised and arranged in a selective complementary fashion.
2. Population sub-grouping into major and minor groups is considered to dedicate sections of the population to learn and adapt with respect to the problem landscape.
3. Novel difference vectors are designed to promote population diversity and prevent population stagnation.
4. Non-linear hunting and competitive learning strategies are formulated and integrated systemically with adaptive mechanisms to achieve equilibrium between exploration and exploitation.

The underlying reasons for the choice of GWO over the other optimization paradigms are as follows. (1) GWO is one of the most successful state-of-the-art optimization techniques with impeccable performance in multi-disciplinary applications and stands unabated with incredible competence outperforming other paradigms as outlined in the literature survey. (2) The simple structure of GWO is easier to be realized in any programming language of choice and can be deployed to various optimization problems in accordance with the researchers' interests. (3) There exists a plethora of publications wherein the performance of GWO has been greatly improved through either application-specific enhancements or hybridization indicating a greater scope of its re-usability for a potentially robust variant of GWO for the many-sided research avenues. (4) The tuning of GWO has been experimented with quite often to improve the accuracy, population diversity, lower its susceptibility to "*the curse of dimensionality*", overthrow local entrapment etc. There is always room for improvement considering the applicability of the variant aimed at, e.g., complex constrained optimization problems with a higher dimensional count could require additional modifications to the algorithmic structure and dynamic tuning resulting in a greater search gradation. (5) The selection and population updating strategies and their coherence to the performance of the algorithms, and the outcome of the optimization have been reviewed and analysed leading to a multitude of variants that exploit various population selection and updating techniques to leverage the algorithm's full potential.

1.4.1 Highlights of the current work

The highlights of the proposed work are outlined as follows.

- A novel version of GWO immune to *the curse of dimensionality* and premature convergence is designed through multi-population and adaptive learning mechanisms
- Extensive testing through the latest benchmarking (2020 and 2019) suites is carried out to determine the suitability and effectiveness of the proposed algorithm while proving an updated overview of GWO's performance for the latest benchmarking standards.
- Comprehensive comparisons are made with the recent and advanced variants of GWO with have been evaluated for older benchmarking (2005) suites so far. This study

compares multiple aspects of the variants of GWO to establish their performance standards for the latest in benchmarking.

- The validation of the proposed method's performance for complex real-world problems in multimedia tools and artificial intelligence is established through MLP training (5 classification datasets and 3 function approximation datasets).

The remainder of this article is organized as follows. Section 2 focuses on the working of GWO followed by a discussion of its merits and demerits. Section 3 discusses the formulation of the competitive learning-based GWO technique with a detailed description of its various attributes. The performance of Clb-GWO with ten different meta-heuristics (including five variants of GWO, two modern meta-heuristics and two state-of-the-art advanced meta-heuristics) is analysed in Section 4 through CEC2020 and CEC2019 benchmarking suites. Additionally, Section 4 analyses the sensitivity of the various tuning parameters on the outcome of optimization and the effect of population size, and number of iterations on the exploration and exploitation. Section 5 analyses the performance of the proposed method with real-world complex optimization tasks (MLP training for five classification datasets and three function approximation datasets). The conclusion, merits and demerits of Clb-GWO potential applications and the future scope of the current work are given in Section 5.

2 Grey wolf Optimizer

Grey Wolf Optimizer, referred to as GWO, is a swarm-based, nature-inspired meta-heuristic optimization algorithm based on the leadership hierarchy and hunting mechanism of grey wolves (*Canis lupus*). Developed in 2014 by Seyedali Mirjalili, Seyed Mohammad Mirjalili and Andrew Lewis, GWO has risen to become one of the prominent state-of-the-art optimizers. [42]. GWO is unique with its excellently crafted social hierarchical system as it groups the grey wolves into *alpha*, *beta*, *delta*, and *omega* and explores and exploits the search space. The tuning requisites of GWO constitute the basic specification of the population size and iteration count and an optional control vector. The balance of the exploration and exploitation is achieved through a linearly decreasing nature of the control vector which is set to decrement from 2 to 0 over the course of iterations. The simplicity of its algorithmic structure and its outstanding performance towards optimization of both unconstrained and constrained with good convergence properties has attracted researchers and practitioners from various fields to opt for it. Computer Science, Machine Learning and Artificial Intelligence, Engineering, Mathematics, Energy, Materials Science, Physics and Astronomy, etc., are some of the applications of GWO across various disciplines.

2.1 Working of GWO

To understand the working of GWO, it is essential to gain insight into how the social hierarchy of wolves is considered in mathematical modelling. GWO considers the alpha wolves (male/female) as the leader (the dominant wolves) as they dictate the functioning of the group and are predominantly responsible for decision-making and managing the group. The second-order consists of the beta wolves which are the subordinates and the advisors and also command the other lower order of wolves. The third in the line-up is the omega wolves which form the lowest ranking group and often assume the roles of a scapegoat or a babysitter.

Additionally, the delta wolves which don't identify themselves as alpha, beta or omega are the Scouts, sentinels, elders, hunters, and caretakers in the group. The delta wolves dominate the omegas but obey the betas and alphas forming an intermediate between the beta wolves and delta wolves. The collective foraging activity based on the social hierarchy forms the core of GWO. Figure 3 depicts the social dominance based hierarchical system of the grey wolves.

In GWO, the best solution is considered as the alpha, the second-best solution is beta and the third-best solution is delta respectively. The latter of the population is considered the omegas. A comprehensive description of the various aspects of the mathematical modelling of GWO is as follows.

Encircling the prey The first phase of GWO is aimed at determining the position of the prey. Initially assumed to be unknown, the algorithm explores the search space considering that the prey's position is located near the optimal solution. Once, the location of the prey is found, they encircle it as a part of the hunting process. To locate a better solution, grey wolves explore the area around the location of prey.

Eq. (1) and Eq. (2) constitute the mathematical model for the encircling of the prey in GWO.

$$\vec{P}_{gw}(t + 1) = \vec{P}_p(t) - \vec{A} \cdot \vec{d} \tag{1}$$

$$\vec{d} = \left| \vec{E} \cdot \vec{P}_p(t) - \vec{P}_{gw}(t) \right| \tag{2}$$

where, \vec{P}_{gw} is the position of the grey wolf, \vec{A} and \vec{E} are coefficient vectors, t is the present iteration, $\vec{P}_p(t)$ is the position of the prey, $\|$ is the modulus operator to determine the absolute value and \cdot represents multiplication in an element-to-element manner.

Eq. (3) and Eq. (4) describe the mathematical formulation of the co-efficient vectors \vec{A} and \vec{E} .

$$\vec{A} = 2 \cdot \vec{a} \cdot \overline{rand}_1 - \vec{a} \tag{3}$$

$$\vec{E} = 2 \cdot \overline{rand}_2 \tag{4}$$

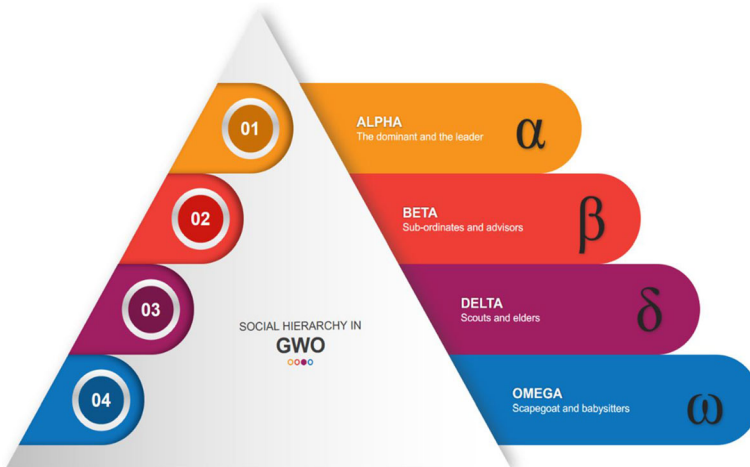


Fig. 3 Social dominant hierarchy of the grey wolves

where, \vec{a} is the control vector whose value tends to linearly decrease from an initial value of 2 to a final value of 0 over the course of iterations and \vec{rand} denotes a random vector in [0, 1].

Hunting As soon as the location of the prey has been recognized, the hunting proves commences guided by the alpha. Supported by the beta, delta and on rare occasions by the omega, the positions of the omegas are updated in conjuncture with the mean position of the alpha, beta and delta. The best three solutions obtained are saved as described in the hierarchical dominance of the wolves to further estimate the location of prey and guide the omegas to update their positions around it in the subsequent iterations.

The distances between the current grey wolf and the three dominant wolves are given in Eq. (5) and the positions formulated based on the distances are given in Eq. (6).

$$\begin{aligned} \vec{d}_\alpha &= \left| \vec{E}_1 \cdot \vec{P}_\alpha - \vec{P}_{gw} \right| \\ \vec{d}_\beta &= \left| \vec{E}_2 \cdot \vec{P}_\beta - \vec{P}_{gw} \right| \\ \vec{d}_\delta &= \left| \vec{E}_2 \cdot \vec{P}_\delta - \vec{P}_{gw} \right| \end{aligned} \tag{5}$$

$$\begin{aligned} \vec{P}_1 &= \vec{P}_\alpha - A_1 \cdot (\vec{d}_\alpha) \\ \vec{P}_2 &= \vec{P}_\beta - A_2 \cdot (\vec{d}_\beta) \\ \vec{P}_3 &= \vec{P}_\delta - A_3 \cdot (\vec{d}_\delta) \end{aligned} \tag{6}$$

Finally, the position of the grey wolf is given by Eq. (7)

$$\vec{P}_{gw}(t+1) = \left[\frac{\vec{P}_1 + \vec{P}_2 + \vec{P}_3}{3} \right] \tag{7}$$

where, \vec{P}_{gw} is the position of the grey wolf, \vec{P}_α , \vec{P}_β and \vec{P}_δ represent the positions of the alpha, beta and delta wolves, \vec{A} and \vec{E} are the co-efficient vectors.

2.2 Demerits of the canonical GWO

Although efficient in several applications, the shortcomings of GWO include a lack of population diversity, local entrapment, premature convergence, lack of a stronger exploitation system etc. to name a few. Several review articles [14, 21, 22, 26, 44, 49] have outlined the limitations of GWO and there has been a greater focus towards the improvement of GWO to achieve a reliable and robust variant.

A summarization of the critical limitations of GWO from various review and work articles has been listed below

- GWO has been susceptible to *the curse of dimensionality* in several benchmark and real-world applications. The performance has been deteriorating in problems with multiple constrained and higher independent decision variables owing to the selection and the population updating strategy. The algorithm’s inability to manage multiple dimensions as

it may not reposition all its search agents appropriately has been researched extensively to realise a better variant immune to such drawbacks.

- The convergence speeds are slower compared to other algorithms depending on sorting techniques for benchmarking and real-world scenarios.
- The system of splitting iterations aimed at an explorative search for the first half and exploitative intensification for the next half does not necessarily guarantee that the majority of the search space has been covered and the conflicting aspects of exploration versus exploitations have not been perfectly balanced despite its good performance compared to the classical paradigms.
- Complex and multi-modal search landscapes have been a challenging aspect as the algorithm is more likely to fall prey to local entrapment leading to premature convergence.
- The exploration system being robust initially, narrows down to the location of the three dominant wolves with the progression of iterations and the wolves may not move far away from each other beyond the initial stages resulting in premature convergence.
- The higher dependence on the three dominant wolves in the wolf pack localizes the population towards the end of iterations causing local entrapment inevitable. If entrapment occurs at an earlier stage, there are no adaptive techniques to escape it.

3 Proposed method: Competitive learning-based Grey wolf Optimizer

In this work, a competitive learning GWO is proposed after a comprehensive analysis of GWO, its other state-of-the-art variants, review articles, and publications related to GWO and its applications. The improved algorithm, named *Competitive-learning based GWO* is devised to address the various limitations of GWO as mentioned previously to improve its immunity to *the curse of dimensionality* and local entrapment with an accelerated convergence towards the global optimum and to achieve the desired equilibrium between the global (exploration) and local search (exploitation) with an enhanced population diversity.

3.1 Analysis and deductions from the previous publications aimed at improving GWO

Although very successful at multi-disciplinary optimization, GWO has its fair share of criticism and controversies surrounding its algorithmic structure for having a one-sided search system known to favour the geometric centre of the search landscapes. There have been several publications demonstrating its demerits that have pointed out its weaknesses, including the lack of a strong exploration system for multi-modal and complex landscapes. The population progression system in GWO favours diversity in the initial stages and quickly converges to the surroundings of the dominant wolves leading to stagnation and loss of population diversity. The analysis at [50] demonstrated this tendency of GWO to slide to the geometric centre (Functions with 0 as the location of the global optimum in the publication) and proposed a verification method through a set of nine modified test functions with varying degrees of shifted global optimum positions. The study concluded that the performance deterioration of GWO was proportional to the degree of shift in the global optimum from 0 and deduced the linearly decreasing nature of the control variable as one of the possible reasons. Multiple works have focussed on improving the population diversity with the intent of lowering the algorithm's dependence on the three dominant wolves in the wolf pack [4, 34, 73]. This has also been referred to as the reason for the algorithm's premature convergence which happens a result of local entrapment [19, 76]. The algorithm's lack of

immunity towards *the curse of dimensionality* has also been the centre of focus as well, with studies indicating the excessive dependence on the dominant wolves and the lack of elitism among the population as the reasons for a poor exploratory system [21, 22, 26, 44]. Slower convergence has been reported in several cases as the algorithm accepts all the population members to replace the older population (*Mu, Lambda* (μ, λ) selection) despite their inferior fitness [63].

On the other hand, several publications have credited the linear control strategy as it provided a basic foundation that can be further improved to ensure robustness in the performance of GWO for other complex landscapes. Reference [47] adopted the standard GWO foraging techniques with an additional dimensional learning strategy based on Euclidean distance and greedy selection to improve the performance of GWO for complex landscapes with an improved GWO algorithm. The performance of IGWO has been verified against the CEC2017 benchmarking suite where none of the benchmarking functions had their global optimum at '0'. In [16], one of the more popular variants of GWO, a random walk GWO with greedy selection is proposed and tested against the CEC2014 suite (also with functions having no optima located at '0') and demonstrated its robustness with complex landscapes. Additionally, selective opposition-based GWO in [12] incorporating the Spearman coefficient in an opposition-based learning scheme to improve the fitness of omegas with respect to the difference between the alpha and omegas is proposed. In [75], hybridization of GWO with Biogeography-Based Optimization (BBO) algorithm to enhance the population diversity of GWO and accelerate the convergence speeds has been proposed. It compared the hybrid GWO with EPSDE, SHADE and SinDE for the CEC2014 benchmarking suite where it outperformed them by a large margin for the thirty benchmark functions.

In other developments, non-linear control strategies have been very popular to establish a solid balance between exploration and exploitation for several multi-disciplinary applications. W.Long et al. in [32] proposed an exploration-enhanced GWO by experimenting with multiple non-linear modulation indices for the control parameter ' α ' and deduced that an initial value of 1 or higher nearer to 1.5 is promising for multi-modal landscapes. The article at [73] proposed an improved GWO with exponential control vectors based on the current and final iterations to enhance the exploration quality for the truss optimization. Research has also been directed at balancing the exploration and the exploitation system of the GWO through the introduction of chaotic strategies to mitigate local entrapment [3, 18, 27, 33, 51, 57]. Hybridization/combinatorial variants of GWO with the existing swarm and evolutionary algorithms have been an ongoing trend since the publication of GWO. The combinatorial variants operate in synergy combining the best aspects of both their parent algorithms with robust and consistent performance across all standards and have considerable performance improvement for the conflicting cases of exploration versus exploitation [60, 63, 77].

3.2 Motivation

In addition to the aforementioned aspects and a myriad of publications of GWO, the motivation for the current work is as follows:

1. Although GWO is a relatively old meta-heuristics, its search process can be efficiently improved to enhance its robustness and consistency for multi-modal and complex landscapes.

2. To combat the demerits associated with the linear exploration method, other non-linear schemes can be experimented with and incorporated into the search mechanism to promote the diversity of the population.
3. A balance between exploration and exploitation can be promoted through the design of suitable difference vectors which can be incorporated strategically and systematically at different stages in the search process.
4. The segregation of the population into two groups has not been experimented with GWO so far to dedicate smaller sections of the population to achieve a specific purpose. The previous multi-strategy ensemble variants have aimed at modifying the structure of GWO completely and have not experimented with population sub-division as of lately.
5. The inclusion of a second position updation strategy and greedy selection has been proven to be beneficial for most complex search spaces and grounding on this the competitive learning phase with distinct strategies is laid out.
6. Success and failure-based strategy adaption, the most popular with the state-of-the-variants of DE have been experimented with to allow the algorithm to learn and adapt to any possible scenario.

Hence, based on the above developments, the current study proposes an improved GWO with an ensemble of strategies to improve population diversity and exploration quality. The following aspects have been considered for the development of the proposed method:

1. The proposed method is built on the merits of the standard GWO linear hunting scheme and a second learning phase with greedy selection follows it as has been the most successful for composite and complex landscapes as seen with the various improved variants.
2. The proposed method follows the population sub-division and is implemented in two phases with each phase complimenting the other in terms of the search strategy and selection mechanisms.
3. The enhancement of population diversity has been the core of the current method.
4. A non-linear control strategy and competitive learning strategy have been added to the standard GWO to improve its robustness and immunity against *the curse of dimensionality*
5. Benchmarking is done through the recent CEC2020 suite and CEC2019 suites, neither of which have been considered for benchmarking so far and none of the benchmarking functions in them has their global optimum located at '0'.
6. Comparisons are not only made against the advanced variants of GWO but also with the state-of-the-art advanced meta-heuristics from the literature and these algorithms are kept consistent throughout the entire benchmarking and real-world testing.

3.3 Implementation

Clib-GWO is implemented in two phases and in both phases, the population is divided into two subgroups. The algorithm's phases and population groups are structured in a selective complementary fashion with the aim of promoting population diversity over convergence. The population is divided into a majority group with 90 % of the wolves and a minority group with 10 % of the remaining wolves. The majority group is mainly responsible for the large-scale exploration and exploitation while the minority groups are reserved to promote divergence and convergence as per their formulation. The advantages of population sub-division

have been highlighted in several of the state-of-the-art advanced meta-heuristics such as EPSO [35], EPSDE [37], MPEDE [65] etc.

3.3.1 Modified GWO phase

i) Majority Group 1 / Hunting group:

As discussed earlier, the linear hunting strategy from the standard GWO has its own set of merits and demerits and grounding on this, the first phase considers a linear hunting scheme complemented by a non-linear hunting scheme. The hunting scheme, either linear hunting or non-linear hunting is selected with a random probability such that both the schemes contribute to the generation of a new population as represented by Eq. (8).

$$\vec{P}_{hunt}(t + 1) = \begin{cases} \text{Linear GWO hunting } \left(\vec{P}_{gw}^{lin} \right) & pr_1 > 0.5 \\ \text{Non-linear GWO hunting } \left(\vec{P}_{gw}^{nl} \right) & \text{otherwise} \end{cases} \tag{8}$$

where, $\vec{P}_{hunt}(t + 1)$ is the updated position of the grey wolf through the various hunting schemes, pr_1 is a random number in 0 and 1 generated through uniform distribution.

Linear hunt The linear hunting scheme is the same as the encircling and hunting of prey technique from the standard GWO algorithm. The distance and position vectors are described in Eq. (9) and Eq. (10) respectively.

$$\begin{aligned} \vec{d}_\alpha &= \left| \vec{E}_1 \cdot \vec{P}_\alpha - \vec{P}_{gw} \right| \\ \vec{d}_\beta &= \left| \vec{E}_2 \cdot \vec{P}_\beta - \vec{P}_{gw} \right| \\ \vec{d}_\delta &= \left| \vec{E}_2 \cdot \vec{P}_\delta - \vec{P}_{gw} \right| \end{aligned} \tag{9}$$

$$\begin{aligned} \vec{P}_1 &= \vec{P}_\alpha - A_1 \cdot \left(\vec{d}_\alpha \right) \\ \vec{P}_2 &= \vec{P}_\beta - A_2 \cdot \left(\vec{d}_\beta \right) \\ \vec{P}_3 &= \vec{P}_\delta - A_3 \cdot \left(\vec{d}_\delta \right) \end{aligned} \tag{10}$$

The coefficient vectors are described by Eq. (11) and Eq. (12) respectively.

$$\vec{A} = 2 \vec{a} \cdot \vec{r}_1 - \vec{a} \tag{11}$$

$$\vec{E} = 2 \cdot \vec{r}_2 \tag{12}$$

The positions of grey wolves are updated as per Eq. (13)

$$\vec{P}_{gw}^{lin}(t + 1) = \left[\frac{\vec{P}_1 + \vec{P}_2 + \vec{P}_3}{3} \right] \tag{13}$$

where, \vec{P}_{gw}^{lin} is the updated position of the grey wolf through the linear hunting scheme, \vec{P}_{gw} is the position of the grey wolf, \vec{P}_α , \vec{P}_β and \vec{P}_δ represent the positions of the alpha, beta and delta wolves, \vec{A} and \vec{E} are the co-efficient vectors, \vec{a} is the control vector whose value tends to linearly decrease from an initial value of 2 to a final value of 0 over the course of iterations ‘ t ’ and \vec{r} denotes a random vector in [0, 1].

Non-linear hunt The non-linear hunting scheme is considered to improve the population diversity based on an exponentially decreasing vector through the course of iterations. This strategy includes the worst solution to form the differential vector while also considering the selection of randomized omega wolves to prevent local stagnation which is often the result of the dominant wolves converging quickly to a single point in the search space. The non-linearity associated with the control vector prevents the solutions from sliding towards the geometric centre of the search space which has been known to severely impact its performance for shifted and rotated benchmark functions. The distance and position vectors are described in Eq. (14) and Eq. (15) respectively.

$$\begin{aligned} \vec{d}_\alpha &= \left| \left[\begin{matrix} \vec{P}_\alpha - \vec{P}_W \\ \vec{P}_\beta - \vec{P}_W \\ \vec{P}_\delta - \vec{P}_W \end{matrix} \right] - \vec{P}_{gw} \right| \\ \vec{d}_\beta &= \left| \left[\begin{matrix} \vec{P}_\alpha - \vec{P}_W \\ \vec{P}_\beta - \vec{P}_W \\ \vec{P}_\delta - \vec{P}_W \end{matrix} \right] - \vec{P}_{gw} \right| \\ \vec{d}_\delta &= \left| \left[\begin{matrix} \vec{P}_\alpha - \vec{P}_W \\ \vec{P}_\beta - \vec{P}_W \\ \vec{P}_\delta - \vec{P}_W \end{matrix} \right] - \vec{P}_{gw} \right| \end{aligned} \tag{14}$$

$$\begin{aligned} \vec{P}_1 &= \vec{\phi} \cdot \vec{P}_{\omega(r1)} - \vec{r}_3 \cdot \left(\vec{d}_\alpha \right) \\ \vec{P}_2 &= \vec{\phi} \cdot \vec{P}_{\omega(r2)} - \vec{r}_4 \cdot \left(\vec{d}_\beta \right) \\ \vec{P}_3 &= \vec{\phi} \cdot \vec{P}_{\omega(r3)} - \vec{r}_5 \cdot \left(\vec{d}_\delta \right) \end{aligned} \tag{15}$$

where, \vec{P}_W position of the grey wolf with the worst fitness value, $\vec{\phi}$ is the control vector whose value tends to exponentially decrease from an initial value of 1 to a final value of 0 over the course of iterations ‘ t ’.

The exponential control vector decreases from 1 to 0 over the course of iterations as described in Eq. (16).

$$\vec{\phi} = e^{(-0.05 \times t)} \tag{16}$$

The final position of the grey wolf is the average of the three positions described by Eq. (17).

$$\vec{P}_{gw}^{nl}(t + 1) = \left[\frac{\vec{P}_1 + \vec{P}_2 + \vec{P}_3}{3} \right] \tag{17}$$

where, \vec{P}_{gw}^{nl} is the updated position of the grey wolf through the non-linear hunting scheme.

ii) **Minority Group 1 / Diverging group:**

The first minority group comprising the remaining 10 % of wolves is retained for re-initialization and random repositioning to diverge the wolves and prevent local entrapment. The divergence is achieved through the described equations in Eq. (18) chosen at random. The divergence vector \vec{P}_Ω is formulated to push the wolves far away from each other.

$$\vec{P}_{hunt}(t + 1) = \begin{cases} \vec{P}_{gw} + \Delta \cdot [\vec{P}_{\Omega}] & pr_2 > 0.5 \\ lb + \vec{r}_6 \cdot [ub - lb] & otherwise \end{cases} \tag{18}$$

where

$$\vec{P}_{\Omega} = \vec{P}_{\omega(r_a)} - \vec{P}_{\omega(r_b)}$$

where, is a random vector in [1, 13], \vec{P}_{Ω} denotes the difference vector of any two randomly chosen omega wolves $\vec{P}_{\omega(r_a)}$ and $\vec{P}_{\omega(r_b)}$, lb and ub denote the lower and upper bounds for the decision variables.

The diversity preserving Mu, Lambda (μ, λ) selection follows the modified GWO phase for the population updation wherein every new solution is accepted to replace its parent solution despite its improved or deteriorated fitness value. However, the memory of the three dominant wolves is updated when a wolf with better fitness than their respective fitness is found.

3.3.2 Competitive learning phase

The competitive learning process follows the standard GWO procedure to further improve the quality of solutions, expand the solution space and ensure a better balance of exploration and exploitation. The search processes are synchronized to allow for the exploration of the search space in both the GWO and the competitive learning phases with a higher emphasis on exploration through the majority competitive learning group followed by a greedy selection process to ensure those fitter solutions replace the older ones.

Similar to the first phase, the competitive learning phase also divides the population into two sub-groups i.e., the Minority group 2 and majority group 2.

i) Minority Group 2 /Converging group:

The second minority group considered the first 10 % of the wolves to improve the local search and accelerate the convergence in a controlled manner. Here, fitness-based repositioning is implemented to guide the wolves to the promising areas of the search space. A single-dimensional update strategy is followed to generate one random number for all the problem dimensions as it ensures accelerated convergence for multi-modal and separable functions.

Every wolf in the second minority group is compared with a random wolf other than the three dominant wolves and repositioned closer to the alpha wolf with respect to its fitness. Fitter wolves are allowed to migrate slowly while the non-fitter wolves are given a higher degree of freedom to reposition themselves much closer to the alpha. Local search around the current position and exploitation of the best solutions is facilitated through the \vec{P}_{Ω}^{gw} and $\vec{P}_{\Omega}^{\alpha}$ vectors respectively as described below in Eq. (19).

$$\vec{P}_{learn}(t + 1) = \begin{cases} \vec{P}_{\omega(r)_{hunt}} + \vec{r}_7 \cdot [\vec{P}_{\Omega}^{gw}] + \vec{r}_8 \cdot [\vec{P}_{\Omega}^{\alpha}] & Fit(i) < Fit(r) \\ \vec{P}_{gw} - \vec{r}_9 \cdot [\vec{P}_{\Omega}^{\alpha}] & Fit(i) > Fit(r) \end{cases} \tag{19}$$

$$\vec{P}_{\Omega}^{gw} = \vec{P}_{gw} - \vec{P}_{\omega(r)}$$

$$\vec{P}_{\Omega}^{\alpha} = \vec{P}_{\alpha} - \vec{P}_{\omega(r)}$$

where, $\vec{P}_{learn}(t + 1)$ is the updated position of the grey wolf through the various learning schemes, $\vec{P}_{\Omega}^{\alpha}$ denotes the difference vector of the alpha wolf \vec{P}_{α} and a randomly chosen omega wolf $\vec{P}_{\omega(r)}$ and \vec{P}_{Ω}^{gw} denotes the difference vector of the current wolf \vec{P}_{gw} and a randomly chosen omega wolf $\vec{P}_{\omega(r)}$.

The inclusion of at least one omega wolf whose position has not been modified from the previous hunting phase is made sure to prevent the loss of diversity during the repositioning process.

ii) Majority group 2 / Learning group:

This learning phase is selected based on the success and failure rates that serve as the moderators to switch between the linear learning and adaptive learning techniques that have been described below. Multi-dimensional update strategy which has been proven to be excellent for non-separable functions has been applied to the learning group, wherein the random numbers are unique for each dimension such that it expands the search space around them for a stronger global search emphasis. Initially, the selection is made probabilistically, and a learning parameter named ‘competitive rate’ controls the selection of the schemes best suited to ensure that the exploration goes on in a smooth and undisturbed manner as per Eq. (20). The competitive rate is the sum of the number of consecutive failures (fr) and success (sr) corresponding to each of the strategies. A detailed description of the competitive rate and its impact on the learning outcomes are discussed in the upcoming sub-sections.

$$\vec{P}_{learn}(t + 1) = \begin{cases} \text{Linear GWO learning} \left(\vec{P}_{clb}^{lin} \right) & \text{If } fr > 10 \text{ or If } sr > 5 \\ \text{Adaptive GWO learning} \left(\vec{P}_{clb}^{adapt} \right) & \text{otherwise} \end{cases} \quad (20)$$

$Comp = fr + sr$

where, $Comp$ denotes the competitive rate fr stands for the failure rate and sr stands for the success rate respectively.

Linear learning The linear GWO learning scheme adopts the linearly decreasing control vector from the standard GWO phase to search for new solutions around the most promising areas in the search space and has a good global search ability. The second technique comprising random omega wolves from the current and hunting population is simply added to prevent the one-sided search progression associated with the linear control strategy and hence has been given a lower priority. The linear search process is prone to drive the population to the geometric centre and to avoid this the difference vectors \vec{P}_{Ω} and \vec{P}_{Ω}^{hunt} are designed. Linear hunting is described by Eq. (21).

$$\vec{P}_{clb}^{lin}(t + 1) = \begin{cases} \vec{P}_{\alpha} + \vec{a} \cdot [\vec{P}_{\Omega}] & pr_3 < 0.75 \\ \vec{P}_{\alpha} + r_{10} \cdot [\vec{P}_{\Omega}] + r_{11} \cdot [\vec{P}_{\Omega}^{hunt}] & \text{otherwise} \end{cases} \quad (21)$$

where

$$\vec{P}_{\Omega}^{hunt} = \vec{P}_{\omega(r)_{hunt}} - \vec{P}_{\omega(r)}$$

where, $\overrightarrow{P_{clb}^{lin}}$ is the updated position of the grey wolf through the linear learning scheme, \overrightarrow{a} is the control vector whose value tends to linearly decrease from an initial value of 2 to a final value of 0 over the course of iterations ‘ t ’, $\overrightarrow{P_{\omega(r)_{hunt}}}$ denotes a randomly chosen omega wolf from the previous hunting schemes and $\overrightarrow{P_{\Omega}^{hunt}}$ denotes the difference vector of a randomized omega wolf $\overrightarrow{P_{\omega(r)_{hunt}}}$ from the hunting schemes and a randomly chosen omega wolf $\overrightarrow{P_{\omega(r)}}$.

Adaptive learning The adaptive hunting scheme comprises an adaptive cooperative learning technique (the first technique) with the alpha, beta and delta wolves to form the solution vector while the second technique involves the selection of only randomised omega wolves from the current and the previous hunting populations. The first technique is prioritized over the second as the knowledge of the alpha, beta and delta can be exploited efficiently in guiding the omegas to more promising areas. The second strategy serves the purpose of diversity enhancement and prevents excessive dependence on the dominant wolves at all times in the search process through the divergence vectors $\overrightarrow{P_{\Omega_1}}$ and $\overrightarrow{P_{\Omega_2}}$ respectively and hence its priority is set to be lower for its selection. Adaptive learning is achieved through the vectors $\overrightarrow{P_{\Omega}^{\alpha}}$ and $\overrightarrow{P_{gw}^{\beta,\delta}}$ wherein the information from the three dominant wolves is used to reposition the wolves from the previous phases. Adaptive hunting is described by Eq. (22).

$$\overrightarrow{P_{clb}^{adapt}}(t + 1) = \begin{cases} \overrightarrow{P_{gw}} + \overrightarrow{R_1} \cdot \left[\overrightarrow{P_{\Omega}^{\alpha}} \right] - \overrightarrow{R_2} \cdot \left[\overrightarrow{P_{gw}^{\beta,\delta}} \right] & pr_4 < 0.75 \\ \overrightarrow{P_{\omega(r)_{hunt}}} + r_{12} \cdot \left[\overrightarrow{P_{\Omega_1}} \right] + r_{13} \cdot \left[\overrightarrow{P_{\Omega_2}} \right] & otherwise \end{cases} \quad (22)$$

$$\overrightarrow{R} = rand(1, D)$$

$$\overrightarrow{P_{\Omega}^{\alpha}} = \overrightarrow{P_{\omega(r)_{hunt}}} - \overrightarrow{P_{\alpha}}$$

$$\overrightarrow{P_{gw}^{\beta,\delta}} = \overrightarrow{P_{gw}} + \left(\overrightarrow{P_{\beta}} + \overrightarrow{P_{\delta}} \right)$$

where, $\overrightarrow{P_{clb}^{adapt}}$ is the updated position of the grey wolf through the adaptive learning scheme, \overrightarrow{R} is a random vector comprising random numbers in [0,1] of the size of 1 by D , with D representing the problem dimensions.

The final step is the fitness evaluations of all the newer population members. The greedy selection technique follows the competitive learning phase to update the population pool with superior solutions from the competitive learning phase. The greedy selection allows for the population members from the competitive learning strategies with better fitness compared to the one from the modified GWO process. The survival of the fittest strategy is followed to select the fitter population members and discard the rest. In the case of inferior solutions, the positions from the modified GWO procedure are retained as given by Eq. (23).

$$\vec{P}_{gw}(t+1) = \begin{cases} \vec{P}_{hunt}(t+1) & \text{if } f(\vec{P}_{hunt}) < f(\vec{P}_{learn}) \\ \vec{P}_{learn}(t+1) & \text{otherwise} \end{cases} \quad (23)$$

where, $f(\vec{P}_{learn})$ is the fitness score of the decision variables obtained by the competitive learning strategy and $f(\vec{P}_{hunt})$ fitness score of the decision variables obtained by the modified GWO procedure.

The overall algorithmic structure of Clb-GWO is presented in Fig. 4.

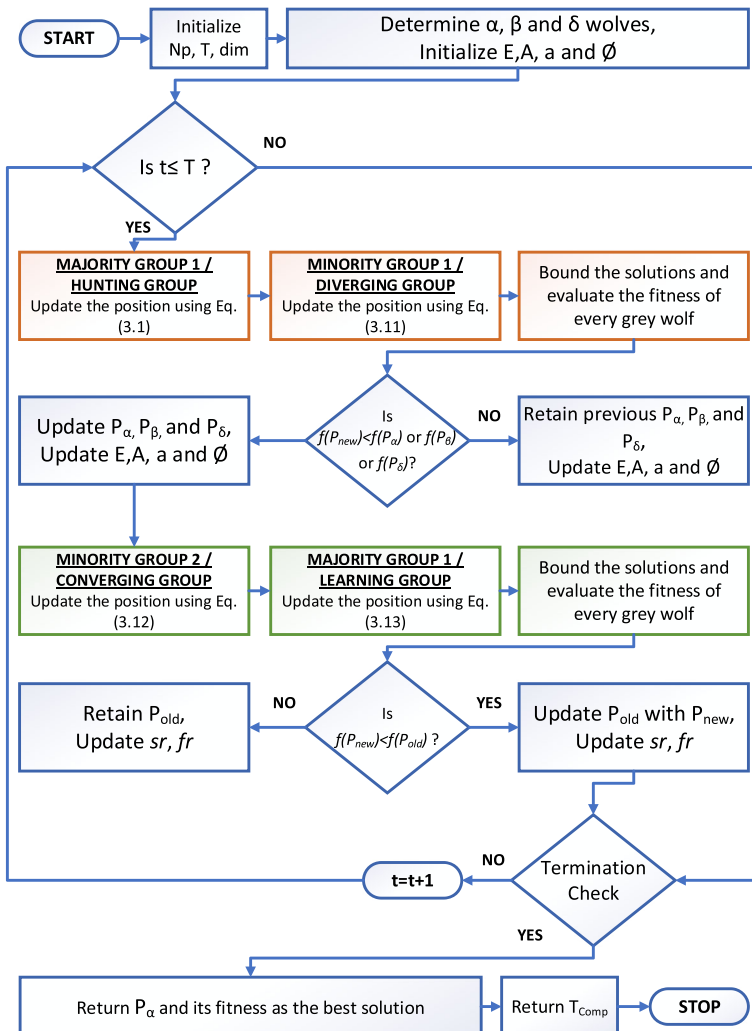


Fig. 4 Flowchart of Clb-GWO

3.3.3 Pseudocode of Clb-GWO

Algorithm 2 Clb-GWO

1. **Initialize** the positions of the grey wolves through randomization
2. **Evaluate** the fitness of every wolf
Assign the position of the grey with the best solution to \vec{P}_α
3. **Assign** the position of the grey with the second-best solution to \vec{P}_β
Assign the position of the grey with the third-best solution to \vec{P}_δ
4. **Initialize** the values of E , A , \vec{a} , $\vec{\phi}$ and \vec{P}_{gw}
5. **Increment** iterations until $t < \text{maximum iterations}$
6. **loop** for the 90 percent of wolves in the population
7. **Select** the hunting mechanism as per Eq. (3.1)
8. **Determine** \vec{d}_α , \vec{d}_β and \vec{d}_δ
9. **Calculate** the positions \vec{P}_1 , \vec{P}_2 and \vec{P}_3
10. **Update** the position of the current wolf using $\vec{P}_{gw}(t+1) = \left\lfloor \frac{\vec{P}_1 + \vec{P}_2 + \vec{P}_3}{3} \right\rfloor$
11. **Update** the values of E , A , \vec{a} , $\vec{\phi}$
12. **loop** for the remaining 10 percent of wolves in the population
13. **Select** the divergence mechanism as per Eq. (3.11)
14. **Evaluate** the fitness of all the wolves
15. **Update** \vec{P}_α , \vec{P}_β and \vec{P}_δ
16. **loop** for the 10 percent wolves in the population
17. **Select** the convergence mechanism as per Eq. (3.12)
18. **loop** for the remaining 90 percent of wolves in the population
19. **Select** the competitive learning mechanism as per Eq. (3.13)
20. **Update** fr , sr
21. **Evaluate** the fitness of all the wolves
22. **Update** the population of all the wolves
23. **Termination**

3.3.4 Analysis of the difference vectors

The difference vectors lie at the core of the proposed method and have been designed after a meticulous study and analysis of the various possible combinations used in previous advanced meta-heuristics. The primary function of the various vectors is to lower the dependence of the algorithms at all times on the three dominant wolves and eventuate to increased diversity in the population. Most of the difference vectors comprise a ransom omega wolf from the population pool which has been deliberately planned to eliminate the clustering of the wolves at any given time and extend the course of exploration over a greater interval of time. Although this can result in slower convergence, the implementation of the search mechanism with them is eliminating the one-side search system in GWO that has received a lot of criticism. The evolution of the wolfpack can be directed in the right direction to explore and exploit

systematically and without being susceptible to entrapment. A tabulation of the various difference vectors designed for the proposed method is tabulated in Table 1.

3.3.5 Exploration and exploitation

The exploration in Clb-GWO is largely contributed by the two majority search groups in the two phases simultaneously, i.e., the encircling mechanism from the linear GWO process where exploitation is favoured for $\vec{a} > 1$ or the diversity promoting the non-linear exponential scheme and the competitive learning strategies wherein the higher distance between the three dominant wolves (*alpha, beta and delta*) puts it to explore around them.

The exploration mechanism in Clb-GWO aims at enhancing the distribution of omega wolves around the potentially promising areas obtained from the standard GWO process considering the difference vectors based on randomised omega wolves. This makes up for two simultaneous exploration processes that run in synergy following each other targeting the various areas of the search spaces with all the new positions of the wolves entering the population pool in the standard GWO process followed by a re-exploration in the Clb-GWO process but with a priority at selectin only the fitter positions involving other random positions to avoid the stagnation of the population members. The existing population is positioned around the *alpha, beta and delta* from the social hierarchy-based hunt and the competitive learning system through the inclusion of a random population from the population pool allows re-distribution of the wolves far away from each other and converge steadily at the best position towards the end of iterations improving the evasion of local entrapment and faster convergence. A mechanism like this provides ample time for the search processes to effectively explore covering a majority of the search space and dedicating the remaining iterations to effectively exploit them to improve the accuracy of the solutions. This technique is different from the conventional approaches of dedicating two separate phases for either exploration or exploitation only. In Clb-GWO, the linear and adaptive competitive learning strategies ensure that a smooth transition from exploration to exploitation is achieved and the elitism promoting

Table 1 Tabulation of the various difference vectors implemented in Clb-GWO

S. No.	Name of the vector	Formulation	Purpose
01.	Distance/Dominant-Worst-Current	$\vec{d}_{\alpha/\beta/\gamma} = \left \left[\vec{P}_{\alpha/\beta/\gamma} - \vec{P}_W \right] - \frac{\vec{P}_{gw}}{P_1} \right $	Distance vector to prevent local entrapment
02.	Position/Exp-omega rand-distance	$\vec{P}_1 = \vec{\phi} \cdot \vec{P}_{\omega(r)} - \vec{r} \cdot \left(\vec{d}_{\alpha/\beta/\gamma} \right)$	Non-linear position update mechanism to promote population diversity
03.	Position/omega rand-2	$\vec{P}_\Omega = \vec{P}_{\omega(r_a)} - \vec{P}_{\omega(r_b)}$	Divergence of population and population diversity enhancement
04.	Position/current-omega rand	$\vec{P}_\Omega^{gw} = \vec{P}_{gw} - \vec{P}_{\omega(r)}$	Local search around the current position
05.	Position/alpha-omega rand	$\vec{P}_\Omega^\alpha = \vec{P}_\alpha - \vec{P}_{\omega(r)}$	Exploitation around the best solution
06.	Position/ hunt omega rand -omega rand	$\vec{P}_\Omega^{hunt} = \vec{P}_{\omega(r)_{hunt}} - \vec{P}_{\omega(r)}$	Enhancement of population diversity
07.	Position/ hunt omega rand -alpha	$\vec{P}_\Omega^{\alpha^?} = \vec{P}_{\omega(r)_{hunt}} - \vec{P}_\alpha$	Adaptive Exploration and exploitation
08.	Position/ current -beta delta	$\vec{P}_{gw}^{\beta,\delta} = \vec{P}_{gw} + \left(\vec{P}_\beta + \vec{P}_\delta \right)$	Adaptive Exploration and exploitation

population selection prevents inferior solutions from influencing the other omega wolves and enables the system to gain a better insight into the current state of the population members and their knowledge of the search space. Several of the recent meta-heuristics (WOA [41], SSA [43], ChoA [25], HHO [20], SMA [29], ALO [39] etc.) have adopted a similar strategy of exploration-exploitation split allowing the population to diverge from each other for the first half of the iterations and converge for the latter. These meta-heuristics have shown good performance with better population diversity and decent convergence behaviour. In a theoretical sense, there could be several other advantages with such an exploration-exploitation split system but in a practical sense there are many limitations as well and the research towards a perfect exploration and exploitation system is still provides a lot of room for development.

The minority population groups complement each other with the top 10 % of the population being placed closer to the dominant wolves and the bottom 10 % spreading around the entire search space promoting better diversity in the population. The interpreting aspect is that the population from one group can influence the other at finding new potential areas to explore or prevent local entrapment around a single point at various points during the search.

Hence to encourage better exploration and good population diversity, the two processes handle the population member differently with different selection mechanisms resulting in a robust ensemble of strategies that aid each other and allowing the population of omegas to learn and adapt to the complex search spaces to explore and exploit optimally.

3.3.6 Time complexity and computational complexity

The position update system in Clb-GWO occurs twice i.e., the standard GWO procedure assigns the positions to all the wolves in the wolf pack after the evaluation of the fitness of the wolves from its previous iteration to determine the alpha, beta and delta wolves. This is followed by the second position update following any of the competitive learning strategies. The greedy selection follows the competitive learning strategy wherein the new position of all the wolves updated from the competitive learning strategy are assessed to decide on preserving the fitter solutions or discarding the inferior ones. The fitness evaluation and the position updates are performed for all the members in the population pool twice in an iteration. Hence, it is obvious that Clb-GWO performs double fitness evaluations (DFEs) per iteration. For an iterative count of T iterations with a population size of N each having a D number of decision variables/dimensions, the following are the computational complexities of individual phases. In addition to the total computational complexity of the standard GWO process, which is $O(N \times (D + T + (T \times D)))$, the competitive learning procedure has an additional computational complexity of $O(T \times (N \times D))$ for the competitive learning phase followed by fitness evaluation of all the new position for the greedy selection with $O(N \times T)$. Summing up, the total computational complexity is $O(N \times (D + 2 \times (T + (T \times D))))$.

In the same manner, the time complexity of Clb-GWO is measured considering its total run time i.e., ' t_{total} ' for one independent run. It is as shown in Eq. (24).

$$t_{total} = t_1 \times O_1 + t_2 \times O_2 + \dots t_N \times O_N \quad (24)$$

where, t_1, t_2, \dots, t_N are the computational times needed by GWO to complete the various operations O_1, O_2, \dots, O_N for N number of wolves. The various operations and the time requirements are presented in Table 2.

Therefore, based on analysis from Table 2, the time complexity of Clb-GWO is $O(N)$.

Table 2 The time complexity of Clb-GWO algorithm

Phase	Time	Total Time required for N number of wolves	Time Complexity
Random Initialization of wolves	t_1	$t_1 \times N$	$O(N)$
Fitness evaluation of the initialized population	t_2	$t_2 \times N$	$O(N)$
Linear/non-linear hunting (Modified GWO phase)	t_3	$t_3 \times N$	$O(N)$
Fitness evaluation for the $Mu, Lambda$ (μ, λ) selection	t_4	$t_4 \times N$	$O(N)$
Competitive learning phase	t_5	$t_5 \times N$	$O(N)$
Fitness evaluation for the greedy selection	t_6	$t_6 \times N$	$O(N)$

4 Results and discussion

Extensive testing and analysis employing standard benchmark functions, standard engineering problems and real-world complex optimization problems are chosen to evaluate the performance potential of the proposed algorithm. The benchmarking tests include the CEC2020 benchmark functions (dimensions set to 5, 10, 15 and 20) to assess the algorithm's susceptibility to *the curse of dimensionality*, assess the performance improvement with respect to the higher number of function evaluations and CEC2019 benchmark functions to evaluate the algorithm's capability to avoid local entrapment and premature convergence. This is followed by the application of Clb-GWO and the other competitor algorithms towards the multi-layer perceptron training for five classification datasets and three function approximation datasets.

All the exterminations considered for the current work are performed on an Ultrabook running the operating system of Microsoft Windows 10® Pro (Version 20H2 - OS Build 19,042.867) with 16 Gigabytes of DDR3 RAM powered by an Intel(R) Core (TM) i7-4700MQ quad-core CPU @ 2.40GHz. MATLAB R2020a is chosen to code all the algorithms for all the considered exterminations in the comparative analysis.

4.1 Description of benchmark functions and performance evaluation criteria

The performance evaluation criteria for all the fifteen algorithms including Clb-GWO for the different benchmarking scenarios (CEC2020, CEC2019 benchmarking suites) are as follows. The purpose of the two benchmarking suites and their importance in the validation of meta-heuristics is specified in Table 3.

- The average (mean) and the standard deviation values are obtained based on 30 independent runs for all the algorithms in comparison. In addition to them the best and the worst fitness values are provided in certain cases (CEC2019 benchmarking suite, standard engineering problems, the power flow optimization problems).
- The NFEs are modified for each problem type and the population size is set based on the NFEs. The details of these are provided prior to the results of every test case. The computational times for the CEC2020 are computed as per the documentation at [72] and the average computational times are provided for the other tests.
- No additional tuning modifications to the algorithm-specific parameters have been made for the entire benchmarking and real-world complex optimization problems.

Table 3 Significance of the CEC2020 and CEC2019 test suites in the validation of meta-heuristics

Benchmarking Suite	Dimensionality	Computational Budget	Importance	Features validated
CEC2020 (10 Functions)	Fixed at 5, 10, 15 and 20	Limited	This test suite helps analyse the performance variation with respect to increased dimensionality while allocating higher computational budgets	<ul style="list-style-type: none"> • Immunity to the curse of dimensionality • Computational budget management • Entrapments evasion • Balance of exploration and exploitation
CEC2019 (10 Functions)	Fixed at 10	Unlimited	This test suite evaluates the accuracy of optimization for complex landscapes with no restrictions on the computational budget.	<ul style="list-style-type: none"> • Accuracy and precision of optimization • Entrapments evasion • Exploitation avoiding stagnation • Extended exploration

- d. The first statistical test, i.e., Wilcoxon's rank-sum test at a 0.05 significance level is performed for Clb-GWO concerning the other algorithms. For better performance of the other algorithms with respect to Clb-GWO "+" symbol is used, for the similar performance of the other algorithms with respect to Clb-GWO "≈" symbol is used and for the inferior performance of the other algorithms concerning Clb-GWO "-" symbol is used.
- e. The second statistical test, i.e., a ranking test through a non-parametric Friedman's test is performed to rank the best-performing algorithms.

4.1.1 Algorithms in the benchmarking framework

- The performance of Clb-GWO is compared and validated against the standard GWO algorithm from 2014 and five of its latest state-of-the-art variants whose description is provided in Table 4.

Table 4 Description of the state-of-the-art meta-heuristics used in the comparative analysis

Categorization	Name of the variant	Authors	Year	Reference
Latest advanced variants of GWO	IGWO (Improved Grey Wolf Optimizer)	M.H. Nadimi-Shahraki et al.	2020	[47]
	MEGWO (Multi-strategy Ensemble Grey Wolf Optimizer)	Q.Tu et al.	2018	[62]
	CGWO (Chaotic Grey Wolf Optimization algorithm)	M. Kohli et al.	2017	[27]
	IGWODE (Improved Grey Wolf Optimizer based on Differential Evolution and Elimination Mechanism)	J. S. Wang et al.	2019	[63]
SOGWO (Selective Opposition based Grey Wolf Optimization)	S. Dhargupta et al.	2020	[12]	
Modern meta-heuristics	WOA (Whale Optimization Algorithm)	S. Mirjalili, A. Lewis	2016	[41]
	ChOA (Chimp Optimization Algorithm)	M.Khishe and M.R.Mosavi	2020	[25]
State-of-the-art advanced meta-heuristics	CLPSO (Comprehensive Learning Particle Swarm Optimizer)	Liang et al.	2006	[30]
	GABC (<i>g</i> best guided Artificial bee colony)	G. Zhu et al.	2010	[78]

- Additionally, two state-of-the-art advanced meta-heuristics within the swarm intelligence namely, CLPSO and GABC, have been employed to assess the performance of the proposed method. A brief description of the two state-of-the-art advanced meta-heuristics is provided in Table 4.
- In addition to the aforementioned variants of GWO, two of the modern meta-heuristics are selected for the testing and validation process. A brief description of the two modern meta-heuristic is provided in Table 4.
- To assess the performance of the proposed methods and rank them for each benchmarking suite, the statistical results of winners/top-performing algorithms are also added in their sub-sections to provide a comprehensive analysis of the current standings of the proposed method.

4.1.2 Tuning settings of the algorithms

To ensure that a fair comparison is achieved, it is required to set/tune the algorithm-specific parameters (tuning parameters) appropriately to extract the best performance. Hence, after a meticulous review of the various algorithms' performances, the following tuning settings have been finalized to ensure that the chosen algorithms deliver their best performance to the fullest of their potential. Please note that the values of the tuning parameters provided in Table 30 (Appendix) remain the same for the entire benchmarking process and real-world problems tackled in the remainder of the manuscript.

The basic parametric tuning for all the algorithms is shown in Table 5.

4.2 CEC2020 benchmarking suite

The first set of benchmarking tests is performed using the CEC2020 benchmarking suite with 5, 10, 15 and 20 dimensions as per the competition rules for the functions described in Table 6. The benchmarking allows for the exponential growth of the computational resources (Number of function evaluations-NFEs) for the increase in the dimensionality of the test functions and their complexity. It comprises 10 scalable benchmark problems within the search range $[-100, 100]^D$ with the global optimum shifted and rotated based on a rotation matrix generated from standard normally distributed entries by Gram-Schmidt ortho-normalization.

The termination criteria for the CEC2020 test functions as defined by the documentation are described in Table 7.

Table 5 Description of the basic tuning parameters for all the algorithms used in the comparative analysis

Algorithm	N (Population Size)	T (Iterations)
GWO, WOA, ChOA, CLPSO, CGWO	$10 \times D$	Max NFEs/N
IGWO, MEGWO, GABC, IGWO-DE	$10 \times D$	Max NFEs/ $2 \times N$
Cib-GWO	$5 \times D$	Max NFEs/ $2 \times N$

Table 6 Description of the test functions from the CEC2020 benchmarking suite

Categorization	Function No.	Function	$F_i^* = Fi(x^*)$
Unimodal Function	F1	Shifted and Rotated Bent Cigar Function	100
Basic Functions	F2	Shifted and Rotated Schwefel's Function	1100
	F3	Shifted and Rotated Lunacek bi-Rastrigin Function	700
	F4	Expanded Rosenbrock's plus Griewangk's Function	1900
	F5	Hybrid Function 1 ($N=3$)	1700
Hybrid Functions	F6	Hybrid Function 2 ($N=4$)	1600
	F7*	Hybrid Function 3 ($N=5$)	2100
	F8	Composition Function 1 ($N=3$)	2200
Composition Functions	F9	Composition Function 2 ($N=4$)	2400
	F10	Composition Function 3 ($N=5$)	2500

*F7 is defined for 10, 15 and 20 Dimensions

4.2.1 Results of benchmarking (CEC2020 test suite)

In this sub-section, Clb-GWO is compared against the standard GWO, five of its latest and advanced variants and two of the modern meta-heuristics. The benchmarking results (mean and standard deviation) are shown in Table 8, the p -values of Wilcoxon's rank-sum test and the results are shown in Table 9 and the results of Friedman's non-parametrical test are shown in Table 10.

Analysis of results

- Clb-GWO outperformed the standard GWO and its three variants by a significant margin and ranked first in Friedman's test.
- However, the performance was poor for the functions with 5 dimensions for five out of the ten functions as it was slower to exploit within the given NFEs. This is on account of the algorithm's tendency to promote exploration and population diversity over exploitation leading to slower convergence. The converging group/Minority group 2 of Clb-GWO is solely responsible for its exploitation for most of the search and the minority groups tendency to exploit very late in the search process are also responsible for a slow convergence with limited computational resources.
- The performance improved greatly for the functions with a higher number of dimensions as the algorithm was able to benefit from the increased number of function evaluations enabling deeper exploitation towards the end of the search. This is evident with F1, wherein Clb-GWO reached the minimum error value of IE-08 for 10D, 15D and 20D while the others had their solutions too far away from the global optimum.

Table 7 Termination criteria for the CEC2020 benchmarking suite

Dimensions	Maximum NFEs	Minimum error value to reach
5	50,000	1.00E-08
10	1,000,000	1.00E-08
15	3,000,000	1.00E-08
20	10,000,000	1.00E-08

Table 8 The values of mean and standard deviation (std) of the CEC2020 benchmarking suite comparing Clb-GWO with the ten competitor algorithms

5D	WOA		ChOA		IGWO-DE		MEGWO		SOGWO		IGWO	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
F1	3.57E+03	4.14E+03	2.11E+06	1.21E+06	4.56E+03	3.59E+03	0.00E+00	0.00E+00	4.63E+03	4.16E+03	5.32E+02	5.73E+02
F2	2.95E+02	1.15E+02	3.49E+02	5.94E+01	9.47E+01	9.78E+01	9.19E+01	6.75E+01	1.64E+02	1.01E+02	9.60E+01	2.68E+01
F3	1.69E+01	6.87E+00	1.47E+01	2.16E+00	8.19E+00	2.15E+00	5.53E+00	9.39E-01	8.65E+00	2.45E+00	6.10E+00	6.62E-01
F4	1.38E+00	1.22E+00	4.29E+00	8.19E+00	5.04E-01	1.54E-01	7.67E-01	1.24E-01	3.66E-01	1.39E-01	3.48E-01	1.42E-01
F5	5.75E+02	5.33E+02	3.47E+02	1.38E+02	1.27E+02	1.13E+02	9.08E-05	1.29E-04	7.17E+02	1.63E+03	3.18E+00	3.08E+00
F6	1.04E+01	9.47E+00	7.07E+01	6.07E+01	5.28E+00	2.01E+01	1.13E-02	1.13E-02	1.01E+01	3.04E+01	4.51E-01	2.47E-01
F7	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
F8	5.44E+01	4.61E+01	9.44E+01	5.16E+01	6.99E+01	4.18E+01	2.69E+01	4.53E+01	5.67E+01	4.91E+01	5.07E+01	5.15E+01
F9	1.84E+02	1.10E+02	2.39E+02	5.63E+01	2.12E+02	9.81E+01	1.04E+02	6.18E+01	2.14E+02	1.01E+02	1.37E+02	7.66E+01
F10	3.48E+02	1.01E-01	3.50E+02	2.54E+01	3.47E+02	2.10E-02	3.47E+02	6.40E+01	3.47E+02	2.78E-02	3.47E+02	1.18E+01
R	W+(9), L-(0), T≈(0)	W+(8), L-(0), T≈(1)	W+(8), L-(0), T≈(1)	W+(8), L-(0), T≈(1)	W+(8), L-(0), T≈(1)	W+(8), L-(0), T≈(1)	W+(5), L-(2), T≈(2)	W+(9), L-(0), T≈(0)	W+(9), L-(0), T≈(0)	W+(9), L-(0), T≈(0)	W+(7), L-(0), T≈(2)	W+(9), L-(0), T≈(1)
10D												
F1	3.18E+03	2.26E+03	5.68E+08	2.95E+08	2.39E-01	1.52E-01	1.42E+01	1.64E+00	6.56E+03	4.62E+03	1.10E+04	4.13E+03
F2	8.15E+02	2.78E+02	1.08E+02	1.99E+02	2.43E+02	2.13E+02	2.36E+02	1.92E+02	2.12E+02	1.70E+02	1.82E+02	2.78E+02
F3	6.75E+01	1.98E+01	5.69E+01	1.10E+01	2.21E+01	6.58E+00	3.01E+01	2.93E+00	1.98E+01	5.55E+00	2.77E+01	6.21E+00
F4	3.09E+00	1.76E+00	2.67E+03	5.53E+03	1.41E+00	5.81E-01	1.16E+00	2.34E-01	8.63E-01	5.10E-01	1.38E+00	4.70E-01
F5	1.32E+04	2.05E+04	5.35E+03	1.63E+03	4.15E+03	2.78E+03	2.30E+02	4.93E+01	3.24E+03	3.24E+03	2.67E+02	1.41E+02
F6	2.63E-01	1.33E-01	4.99E-01	1.23E-01	2.81E-01	1.28E-01	8.50E-01	8.62E-02	2.23E-01	2.10E-01	5.39E-01	1.82E-02
F7	7.77E+03	8.74E+03	5.46E+03	2.26E+03	1.27E+03	2.03E+03	7.96E+01	7.96E+00	2.95E+03	2.34E+03	1.78E+02	1.05E+02
F8	1.14E+02	5.19E+00	2.29E+02	7.92E+01	1.25E+02	1.09E+02	1.00E+02	8.28E-03	1.05E+02	2.76E+01	1.04E+02	3.30E+01
F9	2.80E+02	1.25E+02	3.65E+02	8.24E+00	3.41E+02	1.15E+01	1.00E+02	2.14E-03	3.11E+02	7.42E+01	2.63E+02	1.13E+02
F10	4.44E+02	3.78E+01	4.37E+02	2.66E+01	4.25E+02	2.28E+01	4.01E+02	2.77E-01	4.22E+02	2.33E+01	3.99E+02	4.28E-01
R	W+(9), L-(0), T≈(1)	W+(9), L-(0), T≈(1)	W+(9), L-(0), T≈(1)	W+(9), L-(0), T≈(1)	W+(9), L-(0), T≈(1)	W+(9), L-(0), T≈(1)	W+(9), L-(0), T≈(1)	W+(9), L-(0), T≈(1)	W+(8), L-(0), T≈(2)	W+(8), L-(0), T≈(2)	W+(9), L-(0), T≈(1)	W+(9), L-(0), T≈(1)
15D												
F1	4.48E+02	3.83E+01	4.47E+02	2.62E+01	8.82E+03	7.71E+03	2.40E+01	3.60E-01	4.36E+02	1.60E+01	3.98E+02	3.60E-01
F2	7.03E+02	2.53E+02	1.04E+03	1.92E+02	3.62E+02	1.81E+02	3.72E+02	2.39E+02	2.93E+02	2.04E+02	2.07E+02	2.88E+02
F3	6.95E+01	2.31E+01	6.11E+01	8.75E+00	3.60E+01	2.04E+01	9.83E+01	3.83E+00	9.06E+01	6.95E+00	8.06E+01	7.16E+00
F4	4.54E+00	2.47E+00	4.32E+03	7.76E+03	3.04E+00	1.34E+00	1.96E+00	2.85E-01	3.35E+00	6.13E-01	2.45E+00	6.02E-01
F5	7.11E+04	1.14E+05	7.45E+03	2.21E+03	4.16E+04	2.80E+04	4.64E+02	3.93E+01	4.20E+03	2.88E+03	4.94E+02	1.83E+02

Table 8 (continued)

	WOA		ChOA		IGWO-DE		MEGWO		SOGWO		IGWO	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
5D												
F6	4.35E+01	2.39E+01	8.12E+03	2.47E+03	8.34E+01	5.36E+01	1.56E+03	7.77E+04	3.12E+01	1.84E+01	8.12E+03	2.47E+03
F7	1.16E+04	9.31E+03	6.52E+03	2.40E+03	3.18E+03	4.09E+03	3.06E+02	9.42E+00	4.27E+03	4.19E+03	3.73E+02	1.28E+02
F8	1.11E+02	4.83E+00	2.27E+02	6.47E+01	1.74E+02	1.87E+01	1.00E+02	1.02E+02	1.03E+02	2.35E+00	1.04E+02	3.30E+01
F9	4.85E+02	1.28E+02	9.70E+02	1.81E+02	3.88E+02	9.13E+00	4.14E+02	8.35E+01	4.39E+02	6.58E+00	4.29E+02	1.64E+01
F10	4.33E+02	2.96E+01	4.38E+02	8.46E+00	4.96E+02	7.63E+01	4.31E+02	9.37E+01	4.23E+02	2.09E+01	4.98E+02	3.80E+01
R	W+(9), L-(0), T≈(0)		W+(9), L-(0), T≈(1)		W+(8), L-(0), T≈(1)		W+(8), L-(0), T≈(1)		W+(8), L-(0), T≈(1)		W+(6), L-(3), T≈(1)	
20D												
F1	4.34E+03	6.75E+03	6.71E+03	2.78E+02	6.41E+03	6.94E+03	2.00E+02	1.44E+02	1.32E+03	3.36E+02	1.23E+03	3.74E+02
F2	8.41E+02	2.63E+02	1.33E+03	1.18E+02	1.01E+03	1.43E+02	4.01E+02	2.12E+02	4.21E+02	2.26E+02	5.35E+02	3.83E+02
F3	8.51E+01	2.51E+01	8.35E+01	1.50E+01	4.43E+01	2.31E+01	5.91E+01	3.82E+00	5.64E+01	8.37E+00	8.09E+01	6.64E+00
F4	6.09E+00	3.43E+00	1.78E+04	2.47E+04	5.98E+00	3.23E+00	2.87E+00	5.56E+01	2.33E+00	6.44E+01	2.07E+00	5.48E+01
F5	5.05E+04	6.79E+04	5.52E+04	1.55E+05	2.77E+04	3.00E+04	5.01E+02	7.46E+01	4.14E+03	2.76E+03	8.76E+02	4.04E+02
F6	1.48E+00	4.25E+00	6.69E+00	8.78E+00	1.86E+00	6.90E+01	2.52E+03	1.92E+03	1.50E+00	4.31E+00	3.24E+02	7.93E+02
F7	1.37E+04	1.30E+04	1.21E+04	4.96E+03	2.68E+03	2.92E+03	3.76E+02	1.47E+01	4.63E+03	4.62E+03	4.45E+02	1.29E+02
F8	1.07E+02	2.49E+01	3.75E+02	4.09E+02	2.01E+02	2.19E+02	1.43E+02	4.94E+01	1.05E+02	3.11E+00	1.06E+02	6.77E+01
F9	5.02E+02	1.14E+02	8.85E+02	9.37E+01	3.99E+02	1.57E+01	4.70E+02	1.14E+02	4.93E+02	1.14E+01	4.77E+02	6.19E+01
F10	4.50E+02	1.15E+01	4.64E+02	3.57E+01	4.40E+02	1.21E+02	4.59E+02	1.53E+02	4.36E+02	1.62E+01	4.98E+02	3.67E+01

Table 8 (continued)

	CGWO			CLPSO			GABC			GWO			Cib-GWO			
	Mean	Std.	Std.	Mean	Std.	Std.	Mean	Std.	Std.	Mean	Std.	Std.	Mean	Std.	Std.	
5D																
F1	4.14E+03	3.54E+03	1.64E+01	1.31E+01	4.58E+01	8.49E+01	3.75E+03	3.06E+03	2.00E+02	7.80E+01	7.80E+01	7.80E+01	2.00E+02	3.06E+03	2.00E+02	7.80E+01
F2	7.92E+01	3.01E+01	4.94E+00	4.93E+00	7.58E+00	7.66E+00	1.46E+02	1.09E+02	6.28E+01	6.33E+00	6.33E+00	6.33E+00	6.28E+01	1.09E+02	6.28E+01	6.33E+00
F3	9.65E+00	2.65E+00	3.13E-01	4.95E+00	4.95E+00	1.54E+00	9.08E+00	2.64E+00	9.08E+00	6.33E+00	6.33E+00	6.33E+00	9.08E+00	2.64E+00	6.33E+00	8.08E-01
F4	4.78E-01	1.94E-01	8.51E-02	1.44E-01	8.76E-02	8.87E-02	3.81E-01	1.92E-01	3.81E-01	1.92E-01	1.92E-01	1.92E-01	1.05E-02	1.92E-01	1.05E-02	9.41E-02
F5	2.86E+02	2.54E+02	2.99E-01	1.18E-01	3.34E-02	1.82E-01	8.40E+02	1.84E+03	8.40E+02	6.13E+00	6.13E+00	6.13E+00	6.13E+00	1.84E+03	6.13E+00	6.58E+00
F6	1.34E+00	6.32E-01	1.14E-01	1.14E-01	9.61E-02	5.02E-02	2.09E+01	4.01E+01	2.09E+01	7.75E-02	7.75E-02	7.75E-02	7.75E-02	4.01E+01	7.75E-02	1.42E-01
F7	NA	NA	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
F8	9.33E+00	7.01E+00	4.00E+00	4.00E+00	0.00E+00	0.00E+00	5.28E+01	5.01E+01	5.28E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.01E+01	0.00E+00	0.00E+00
F9	1.02E+02	8.94E+00	2.33E+01	1.08E+02	1.46E+02	4.62E+01	2.02E+02	1.03E+02	1.46E+02	2.02E+02	1.03E+02	1.03E+02	2.02E+02	1.03E+02	1.00E+02	0.00E+00
F10	3.48E+02	2.11E-02	6.85E+01	3.61E+02	3.98E+02	5.74E+01	3.47E+02	2.08E-02	3.47E+02	3.47E+02	2.08E-02	2.08E-02	3.47E+02	2.08E-02	3.47E+02	1.55E-02
R	W+(9), L-(0), T≈(0)		W+(4), L-(2), T≈(3)	W+(4), L-(2), T≈(3)	W+(4), L-(3), T≈(2)	W+(4), L-(3), T≈(2)	W+(8), L-(0), T≈(1)		W+(8), L-(0), T≈(1)							
10D																
F1	3.28E-01	1.86E-01	1.73E+02	3.63E+02	1.73E+02	5.12E+02	6.16E+03	4.50E+03	3.85E+02	5.12E+02	4.50E+03	4.50E+03	0.00E+00	4.50E+03	0.00E+00	0.00E+00
F2	5.23E+02	2.91E+02	4.61E+01	1.01E+02	1.46E+01	9.09E+00	1.35E+02	1.18E+02	1.46E+01	9.09E+00	1.18E+02	1.18E+02	2.80E+02	1.18E+02	2.80E+02	9.46E+01
F3	2.77E+01	9.86E+00	1.27E+00	2.61E+01	2.27E+01	6.07E-01	2.05E+01	6.36E+00	2.27E+01	6.07E-01	6.36E+00	6.36E+00	1.59E+01	6.36E+00	1.59E+01	2.54E+00
F4	1.29E+00	8.24E-01	2.21E-01	5.10E-01	5.04E-01	1.25E-01	1.45E+00	4.46E-01	5.04E-01	1.25E-01	4.46E-01	4.46E-01	5.83E-01	4.46E-01	5.83E-01	1.69E-01
F5	5.05E+03	2.82E+03	5.82E+02	1.56E+03	2.38E+04	1.66E+04	2.28E+03	1.97E+03	2.38E+04	1.66E+04	1.97E+03	1.97E+03	5.06E+01	1.97E+03	5.06E+01	9.79E+00
F6	3.23E-01	1.77E-01	4.92E-04	8.63E-04	5.45E-02	9.58E-02	3.23E-01	2.03E-01	5.45E-02	9.58E-02	2.03E-01	2.03E-01	6.25E-04	2.03E-01	6.25E-04	1.40E-03
F7	7.68E+02	3.07E+02	9.11E+01	1.73E+02	4.76E+02	8.24E+02	2.37E+03	3.66E+03	4.76E+02	8.24E+02	3.66E+03	3.66E+03	5.64E+00	3.66E+03	5.64E+00	1.11E+00
F8	9.04E+01	1.64E+01	1.46E+01	1.05E+02	1.07E+02	8.92E+00	1.21E+02	3.28E+01	1.07E+02	8.92E+00	3.28E+01	3.28E+01	8.06E+00	3.28E+01	8.06E+00	3.40E+00
F9	3.24E+02	6.11E+01	5.45E+00	1.27E+02	1.04E+02	2.20E+01	3.13E+02	7.49E+01	1.04E+02	5.45E+00	7.49E+01	7.49E+01	1.00E+00	7.49E+01	1.00E+00	0.00E+00
F10	4.15E+02	2.11E+01	4.93E+01	3.99E+02	3.99E+02	1.40E+02	4.28E+02	2.28E+01	3.99E+02	1.40E+02	2.28E+01	2.28E+01	3.99E+02	2.28E+01	3.99E+02	4.28E-01
R	W+(9), L-(0), T≈(1)		W+(7), L-(1), T≈(2)	W+(8), L-(1), T≈(1)	W+(8), L-(1), T≈(1)	W+(8), L-(1), T≈(1)	W+(9), L-(0), T≈(1)		W+(8), L-(1), T≈(1)							
15D																
F1	5.08E+03	5.84E+03	3.15E+02	4.15E+02	1.52E+03	2.02E+03	4.38E+02	1.42E+01	1.52E+03	2.02E+03	1.42E+01	1.42E+01	0.00E+00	1.42E+01	0.00E+00	0.00E+00
F2	7.41E+02	3.08E+02	4.27E+01	1.43E+02	2.90E+01	1.37E+01	2.83E+02	2.19E+02	2.90E+01	1.37E+01	2.19E+02	2.19E+02	5.95E+02	2.19E+02	5.95E+02	2.29E+02
F3	6.49E+01	1.78E+01	1.01E+01	6.66E+01	3.16E+01	3.25E+00	1.60E+02	6.11E+01	3.16E+01	3.25E+00	6.11E+01	6.11E+01	2.82E+01	6.11E+01	2.82E+01	3.66E+00

Table 8 (continued)

	CGWO		CLPSO		GABC		GWO		Cib-GWO	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
5D										
F4	2.98E+00	1.67E+00	1.72E+00	1.68E-01	3.05E+00	1.45E-01	2.17E+00	6.52E-01	1.05E+00	2.34E-01
F5	5.33E+04	3.00E+04	6.56E+02	1.89E+02	6.10E+02	1.70E+02	3.77E+03	2.86E+03	2.66E+02	2.49E+02
F6	9.95E-01	7.28E-01	9.67E-02	1.11E-02	9.63E-02	1.03E-01	3.37E-01	2.92E-01	4.80E-02	1.03E-02
F7	7.31E+03	5.57E+03	4.76E+02	2.36E+02	1.12E+03	8.55E+02	4.17E+03	3.33E+03	1.29E+00	1.27E+00
F8	2.06E+02	3.38E+01	1.07E+02	1.41E+01	1.03E+02	3.66E+01	1.04E+02	3.00E+00	1.00E+00	0.00E+00
F9	3.82E+02	1.20E+01	4.31E+02	7.77E+00	4.04E+02	8.05E+00	4.36E+02	5.46E+00	3.07E+02	4.00E+00
F10	4.64E+02	9.14E+01	4.54E+02	4.02E+01	4.47E+02	1.08E+02	4.34E+02	1.99E+01	4.00E+02	0.00E+00
R	W+(8), L-(0), T≈(2)		W+(8), L-(1), T≈(1)		W+(8), L-(1), T≈(1)		W+(8), L-(0), T≈(2)			
20D										
F1	4.10E+03	4.42E+03	3.80E+02	5.44E+01	3.78E+02	7.69E+01	3.86E+03	1.22E+02	0.00E+00	0.00E+00
F2	1.24E+03	2.48E+02	3.75E+02	8.08E+01	3.96E+02	4.54E+01	5.99E+02	3.23E+02	9.89E+02	1.95E+01
F3	8.40E+01	2.79E+01	4.81E+01	1.62E+00	4.89E+01	1.39E+00	1.02E+02	5.18E+00	4.62E+01	5.16E+00
F4	6.48E+00	5.50E+00	3.83E+00	3.20E-01	2.71E+00	1.94E-01	2.65E+00	7.01E-01	2.05E+00	1.16E+00
F5	7.13E+04	5.49E+04	9.57E+02	3.52E+02	9.87E+02	7.39E+02	3.35E+03	2.82E+03	6.90E+01	2.68E+01
F6	1.81E+00	5.20E-01	2.80E-02	4.61E-02	1.15E-01	1.13E-01	4.14E+00	7.76E+00	0.00E+00	0.00E+00
F7	5.50E+04	5.82E+04	7.60E+02	3.94E+02	4.04E+02	4.29E+01	3.99E+03	4.07E+03	3.57E+00	5.57E+00
F8	1.65E+02	2.56E+02	1.09E+02	2.03E+01	1.02E+02	2.11E+00	1.04E+02	3.09E+00	1.00E+02	0.00E+00
F9	4.26E+02	1.66E+01	5.02E+02	6.92E+00	4.51E+02	6.22E+01	5.41E+02	1.98E+01	3.96E+02	1.27E-01
F10	4.83E+02	4.73E+01	4.98E+02	1.07E+01	4.87E+02	4.15E+01	4.34E+02	2.13E+01	4.14E+02	1.07E-03
R	W+(9), L-(0), T≈(1)		W+(7), L-(1), T≈(2)		W+(8), L-(1), T≈(1)		W+(9), L-(1), T≈(0)			

Table 9 Results of the Wilcoxon’s rank sum test comparing Clb-GWO with the ten competitor meta-heuristics for the CEC2020 benchmarking suite

Clb-GWO vs.	5D			10D			15D			20D						
	R+	R-	p value	Result	R+	R-	p value	Result	R+	R-	p value	Result				
WOA	45	0	3.39E-02	+	55	0	3.42E-02	+	55	0	7.57E-02	+	54	1	1.04E-02	+
ChOA	45	0	4.20E-02	+	55	0	7.28E-03	+	54	1	2.57E-02	+	55	0	1.72E-02	+
IGWO-DE	44	0	1.98E-03	+	54	1	1.86E-02	+	54	1	1.00E-02	+	54	1	1.21E-02	+
MEGWO	35	6	5.81E-02	≈	52	1	4.72E-02	+	49	3	4.06E-02	+	52	3	4.20E-02	+
SOGWO	44	0	3.80E-02	+	54	1	5.39E-02	+	54	1	1.21E-02	+	54	1	1.40E-02	+
IGWO	39	3	5.62E-02	≈	52	1	1.30E-02	+	52	3	2.73E-02	+	54	1	2.12E-02	+
CGWO	45	0	2.22E-02	+	55	0	1.62E-02	+	55	0	7.06E-02	+	55	0	2.89E-02	+
CLPSO	35	10	3.14E-02	≈	49	3	1.30E-02	+	54	1	1.62E-02	+	54	1	3.45E-02	+
GABC	30	10	6.82E-02	≈	49	3	1.51E-02	+	54	1	1.40E-02	+	54	1	3.49E-02	+
GWO	44	1	3.80E-02	+	54	1	2.39E-02	+	54	1	1.04E-02	+	54	1	2.89E-02	+

Table 10 Ranking the algorithms based on Friedman’s rank for the CEC2020 benchmarking suite (Clb-GWO vs ten competitor meta-heuristics)

Rank	5D			10D			15D			20D		
	Algorithm	Friedman’s rank	Algorithm	Friedman’s rank	Algorithm	Friedman’s rank	Algorithm	Friedman’s rank	Algorithm	Friedman’s rank	Algorithm	Friedman’s rank
1.	Clb-GWO	3.03078	Clb-GWO	3.0431	Clb-GWO	2.4171	Clb-GWO	2.6094	Clb-GWO	2.4171	Clb-GWO	2.6094
2.	GABC	3.81871	CLPSO	4.5340	MEGWO	4.3060	MEGWO	4.4816	MEGWO	4.3060	MEGWO	4.4816
3.	MEGWO	4.29084	GABC	5.1787	CLPSO	5.1971	CLPSO	4.6456	GABC	5.1971	GABC	4.6456
4.	CLPSO	4.51272	IGWO	5.5357	IGWO	5.2769	IGWO	5.8898	CLPSO	5.2769	CLPSO	5.8898
5.	IGWO	4.93539	MEGWO	5.7595	GABC	5.7235	GABC	5.8952	IGWO	5.7235	IGWO	5.8952
6.	CGWO	6.97059	SOGWO	6.4557	SOGWO	6.4318	SOGWO	6.1655	SOGWO	6.4318	SOGWO	6.1655
7.	IGWO-DE	7.43577	CGWO	7.2491	GWO	7.1948	GWO	7.1387	IGWO-DE	7.1948	IGWO-DE	7.1387
8.	GWO	8.35584	IGWO-DE	7.6922	IGWO-DE	7.5712	GWO	7.2463	GWO	7.5712	GWO	7.2463
9.	WOA	8.65310	GWO	7.7577	CGWO	8.3462	CGWO	8.8502	WOA	8.3462	WOA	8.8502
10.	SOGWO	8.74267	WOA	9.3218	ChOA	8.6555	ChOA	8.8669	CGWO	9.3218	CGWO	8.8669
11.	ChOA	10.61183	ChOA	11.2157	WOA	9.0922	WOA	10.0344	ChOA	11.2157	ChOA	10.0344

Table 11 The time complexity (in ms) of the various algorithms in the comparative analysis for the CEC2020 benchmarking suite

Algorithm	Dim	T0	T1	T2	(T2-T1)/T0
GWO	5	5.22E+01	3.78E+02	1.22E+03	1.61E+04
	10	5.22E+01	4.04E+02	1.41E+03	1.92E+04
	15	5.22E+01	4.46E+02	1.70E+03	2.40E+04
Clb-GWO	20	5.22E+01	4.84E+02	2.02E+03	2.94E+04
	5	5.22E+01	3.78E+02	2.73E+03	4.50E+04
	10	5.22E+01	4.04E+02	3.42E+03	5.79E+04
IGWO	15	5.22E+01	4.46E+02	4.16E+03	7.12E+04
	20	5.22E+01	4.84E+02	4.97E+03	8.61E+04
	5	5.22E+01	3.78E+02	5.54E+03	9.89E+04
SOGWO	10	5.22E+01	4.04E+02	5.98E+03	1.07E+05
	15	5.22E+01	4.46E+02	5.89E+03	1.04E+05
	20	5.22E+01	4.84E+02	6.07E+03	1.07E+05
MEGWO	5	5.22E+01	3.78E+02	2.87E+03	4.78E+04
	10	5.22E+01	4.04E+02	3.79E+03	6.49E+04
	15	5.22E+01	4.46E+02	5.04E+03	8.81E+04
WOA	20	5.22E+01	4.84E+02	6.09E+03	1.07E+05
	5	5.22E+01	3.78E+02	8.05E+02	8.19E+03
	10	5.22E+01	4.04E+02	9.43E+02	1.03E+04
ChOA	15	5.22E+01	4.46E+02	1.04E+03	1.14E+04
	20	5.22E+01	4.84E+02	1.02E+03	1.03E+04
	5	5.22E+01	3.78E+02	8.24E+02	8.55E+03
GABC	10	5.22E+01	4.04E+02	8.67E+02	8.88E+03
	15	5.22E+01	4.46E+02	9.54E+02	9.74E+03
	20	5.22E+01	4.84E+02	9.63E+02	9.19E+03
CLPSO	5	5.22E+01	3.78E+02	4.75E+03	8.39E+04
	10	5.22E+01	4.04E+02	8.92E+03	1.63E+05
	15	5.22E+01	4.46E+02	1.28E+04	2.36E+05
IGWODE	20	5.22E+01	4.84E+02	1.66E+04	3.09E+05
	5	5.22E+01	3.78E+02	1.23E+03	1.63E+04
	10	5.22E+01	4.04E+02	1.22E+03	1.56E+04
CGWO	15	5.22E+01	4.46E+02	1.25E+03	1.54E+04
	20	5.22E+01	4.84E+02	1.29E+03	1.55E+04
	5	5.22E+01	3.78E+02	2.16E+03	3.41E+04
IGWODE	10	5.22E+01	4.04E+02	2.25E+03	3.53E+04
	15	5.22E+01	4.46E+02	2.36E+03	3.67E+04
	20	5.22E+01	4.84E+02	2.49E+03	3.84E+04
CGWO	5	5.22E+01	3.78E+02	1.16E+03	1.88E+04
	10	5.22E+01	4.04E+02	1.32E+03	2.64E+04
	15	5.22E+01	4.46E+02	2.14E+03	3.55E+04
CGWO	20	5.22E+01	4.84E+02	2.92E+03	4.32E+04
	5	5.22E+01	3.78E+02	1.32E+03	1.81E+04
	10	5.22E+01	4.04E+02	1.44E+03	2.34E+04
CGWO	15	5.22E+01	4.46E+02	2.65E+03	3.54E+04
	20	5.22E+01	4.84E+02	2.84E+03	3.64E+04

- Apart from F2, the performance of Clb-GWO has been good for the others indicative of the influence of the difference vectors in maintaining population diversity and promoting exploration in complex landscapes.
- The performance of the other meta-heuristics was not on par with Clb-GWO. The increase in the problem dimensions lead to entrapment of the population members and failed to escape it leading to poor diversity in the population. The sorting techniques adopted by them increased the computational times and failed to produce a notable improvement in

Table 12 Description of the 10 CEC2019 benchmark functions (composition functions) used to determine the algorithms' ability to avoid local entrapment

Function No.	Function	$F_i^* = F_i(X^*)$	Dimensions	Search Range
C1	Storn's Chebyshev Polynomial Fitting Problem	1	9	[-8192, 8192]
C2	Inverse Hilbert Matrix Problem	1	16	[-16,384, 16,384]
C3	Lennard-Jones Minimum Energy Cluster Problem	1	18	[-4,4]
C4	Shifted and Rotated Rastrigin's Function	1	10	[-100,100]
C5	Shifted and Rotated Griewank's Function	1	10	[-100,100]
C6	Shifted and Rotated Weierstrass Function	1	10	[-100,100]
C7	Shifted and Rotated Schwefel's Function	1	10	[-100,100]
C8	Shifted and Rotated Expanded Schaffer's F6 Function	1	10	[-100,100]
C9	Shifted and Rotated Happy Cat Function	1	10	[-100,100]
C10	Shifted and Rotated Ackley Function	1	10	[-100,100]

the performance. Another reason could be traced back to the presence of a large number of tuning parameters and no adaptive control strategy and out of the fourteen chosen algorithms, WOA, GWO, IGWO required no additional tuning parameter settings while MEGWO, SOGWO and ChOA required the tuning of special algorithm-specific parameters (2 to 4 parameters) whose values have been set based on their corresponding publications. Although the empirical setting favoured performance for problems with a lower number of dimensions as seen in [12, 18, 27, 48, 63], the same performance was not reflected for the larger dimensional problems. One particular reason for this is to do with the formulation of the solution set wherein every dimension/decision variable has not achieved the global best solution leading to an imbalance in the optimization and thereby producing highly non-optimal solutions.

- Clb-GWO's performance is consistent through the testing with the increase in the number of problem dimensions having very little effect on the efficiency of the algorithm as seen in Table 8. The other algorithms' performance dwindled over the increase in the dimensions with IGWO being the most hard-hit followed by SOGWO and MEGWO. Although IGWO features the greedy selection strategy, the neighbourhood construction strategy complicates the nearest neighbour search in high dimensional space. It is not possible to quickly reject candidates by using the difference in one coordinate as a lower bound for a distance based on all the dimensions. This system can lead to various phenomena that arise when analysing and organizing data in high-dimensional spaces that do not occur in low-dimensional settings. This is evident by the performance of IGWO being better at handling lower-dimensional problems as seen from the benchmarking.
- Furthermore, comparing Clb-GWO's performance with IGWO and MEGWO for the 5D test case, the results are in favour of the latter as their exploitation capabilities at lower dimensionalities aid their convergence, accelerating intensification. One more reason for the slow convergence of Clb-GWO for the 5D test case is the computational budget limitation to 50,000 NFEs which limits Clb-GWO from adequately transitioning from exploration to exploitation. Clb-GWO's structure is majorly inclined towards exploration over exploitation as premature exploitation tends to be a major contributor to stagnation. This is well established by the fact that Clb-GWO attained good convergence for the rest of the test cases i.e., 10, 15 and 20D cases as it outperformed all the competitor algorithms

Table 13 The values of best, worst, mean and the standard deviation of the eleven algorithms for the CEC2019 benchmark functions

	WOA	ChOA	IGWO-DE	MEGWO	SOGWO	IGWO	CGWO	CLPSO	GABC	GWO	Clb-GWO
C1	Best	9.3680	1.0000	I*	I*	I*	I*	1.7656	1.0018	I*	I*
	Worst	5280.9448	226.0117	2.03E+04	1.0001	651.8578	2.91E+05	74.0243	71.4033	163.3091	I*
	Mean	1385.5550	84.5880	1.49E+03	1	113.355	3.9717	19.7563	18.6294	35.8708	I*
	Std	1719.1255	82.1027	4.47E+03	3.58E-07	198.9775	9.3973	7.19E+04	22.0578	59.2792	0
C2	Best	4796.4153	15.8565	104.5533	12.2129	93.7646	220.5894	7.3107	970.7971	47.2376	I
	Worst	12.421.8426	2233.1361	334.9583	190.5386	442.8793	417.1708	1.35E+03	2120.9228	426.0417	4.8461
	Mean	7875.8187	537.2287	211.6980	105.9166	227.329	198.5653	561.5409	1570.3215	196.3516	2.8565
	Std	2622.5518	697.8926	75.2713	47.0831	130.6496	92.0493	357.5290	338.5399	114.7176	0.2389
C3	Best	1.4091	2.5858	1.0000	I*	1.0001	I*	9.9158	1.0323	I*	I*
	Worst	2.3003	5.0410	7.7112	2.4426	1.8553	1.4097	10.778	1.4992	4.6071	1.4091
	Mean	1.4983	3.5087	1.9578	1.5899	1.3024	1.204	10.6481	1.3573	1.6935	1.1169
	Std	0.2818	0.7330	2.0287	0.5209	0.285	0.2149	2.5897	0.258	0.1701	1.0321
C4	Best	18.9092	22.7814	3.9922	2.0057	2.112	1.0037	3.9887	2.99	2.9993	1.9846
	Worst	41.7932	41.4589	19.4294	8.8895	14.2117	18.6022	6.2702	5.0018	7.9666	6.9798
	Mean	29.9855	35.1974	11.2927	6.6138	7.0791	6.5989	10.6511	5.1481	3.7892	4.1889
	Std	8.1028	5.5949	5.3917	2.4497	4.1834	6.6404	3.2849	0.8172	0.6337	1.8357
C5	Best	1.2167	2.2299	1.1073	1.0166	1.1295	1.2327	1.0863	I*	1.1405	I*
	Worst	2.2078	13.3613	1.6991	1.1022	1.5799	1.4738	1.0454	1.0124	1.5455	1.0073
	Mean	1.6753	8.9882	1.3654	1.0682	1.3551	1.3922	1.1945	1.0263	1.0034	1.0025
	Std	0.3582	5.4932	0.1950	0.0255	0.1655	0.0736	0.0877	0.0115	0.0045	0.0036
C6	Best	3.9246	4.8112	1.1675	1.1002	1.1159	1.0783	2.5825	1.2698	1.0026	I*
	Worst	10.1845	8.5899	3.3434	3.9911	3.8106	1.1495	6.3434	1.3837	1.4831	1.0003
	Mean	6.4257	6.3724	2.1025	1.9856	1.5606	1.1006	4.6654	1.3282	1.1531	1.0003
	Std	1.9677	1.4100	0.9126	1.2914	0.8158	0.021	1.3986	0.0352	0.2264	0.1048
C7	Best	5.44.4020	855.3027	36.5808	159.5207	134.2897	1.4864	101.3917	1.1268	24.8104	1.2498
	Worst	1189.6895	1376.3450	449.3146	682.3775	682.3166	900.7683	494.7004	289.1054	691.0171	438.1265
	Mean	878.9262	1194.665	233.2815	444.0077	323.9188	303.3859	360.6525	171.8881	355.1107	140.0919
	Std	231.8833	171.9296	129.0133	154.5254	191.259	300.246	132.4887	51.1906	108.6524	70.8483
C8	Best	3.4332	3.7118	1.4323	1.213	1.5802	1.2184	2.4731	1.5879	1.535	1.5224
	Worst	4.4420	4.8955	3.1773	3.8515	3.5694	2.3314	3.1357	2.4992	3.4378	2.884
	Mean	3.9866	4.308	2.4062	2.7362	2.8672	1.7853	3.5771	1.9823	2.5668	2.023
	Std	0.3453	0.3697	0.6469	0.759	0.6533	0.4394	0.5258	0.1883	0.3928	0.4406
C9	Best	1.1135	1.0950	1.0379	1.0487	1.0509	1.0406	1.1088	1.0588	1.0371	1.065

Table 13 (continued)

	WOA	ChOA	IGWO-DE	MEGWO	SOGWO	IGWO	CGWO	CLPSO	GABC	GWO	Clb-GWO
Worst	1.6330	1.4384	1.1043	1.1059	1.0718	1.0908	1.2642	1.1451	1.1238	1.1219	1.1156
Mean	1.3429	1.2885	1.0699	1.0754	1.0613	1.0614	1.1718	1.1302	1.0922	1.0716	1.0895
Std	0.1664	0.1173	0.0208	0.0215	0.0074	0.0168	0.0654	0.0122	0.0225	0.0265	0.0151
C10	20.9989	21.1632	21.1975	21.1106	21.1302	1.0803	2.2033	8.3417	1.0406	21.1499	1*
Worst	21.2272	21.3325	21.2953	21.2588	21.316	21.3587	21.2712	21.0854	21.0137	21.3042	20.9999
Mean	21.0323	21.2704	21.2394	21.1961	21.2149	19.2181	19.3005	14.8246	15.6083	21.2548	14.102
Std	0.0736	0.0516	0.0293	0.0441	0.0624	6.3734	6.0075	3.4325	8.1395	0.0421	7.5719

Table 14 The p-values obtained from the Wilcoxon's rank sum test comparing Clb-GWO with the ten algorithms for the CEC2019 benchmark functions

	WOA	ChOA	IGWO	SO-GWO	MEGWO	GWO	GABC	CLPSO	CGWO	IGWODE
C1	1.04E-12 (-)	5.48E-12 (-)	1.18E-11 (-)	5.48E-12 (-)	3.25E-01 (+)	8.41E-12 (-)	1.47E-07 (-)	8.81E-08 (-)	6.25E-11 (-)	2.67E-12 (-)
C2	2.37E-12 (-)	4.92E-12 (-)	6.21E-13 (-)	4.92E-12 (-)	4.34E-12 (-)	4.37E-11 (-)	2.79E-12 (-)	3.20E-09 (-)	3.25E-12 (-)	4.15E-11 (-)
C3	1.03E-06 (-)	4.41E-08 (-)	1.19E-04 (-)	4.41E-08 (-)	1.72E-05 (-)	6.25E-05 (-)	1.06E-04 (-)	2.10E-11 (-)	4.64E-04 (-)	2.15E-06 (-)
C4	2.57E-06 (-)	2.97E-08 (-)	2.15E-04 (-)	2.97E-08 (-)	4.58E-03 (-)	8.99E-03 (-)	1.25E-04 (-)	1.74E-03 (-)	6.68E-04 (-)	3.94E-03 (-)
C5	5.57E-02 (≈)	2.35E-12 (-)	2.29E-05 (-)	2.35E-12 (-)	6.37E-02 (≈)	3.56E-05 (-)	6.82E-01 (+)	3.20E-03 (-)	2.65E-05 (-)	1.36E-05 (-)
C6	2.71E-05 (-)	2.75E-08 (-)	2.99E-02 (+)	2.75E-08 (-)	6.90E-04 (-)	1.30E-08 (-)	9.26E-03 (-)	1.06E-03 (-)	9.66E-09 (-)	9.95E-09 (-)
C7	1.68E-08 (-)	2.85E-13 (-)	2.32E-06 (-)	2.85E-13 (-)	1.24E-07 (-)	1.76E-07 (-)	2.23E-01 (+)	1.06E-02 (+)	1.31E-07 (-)	1.40E-07 (-)
C8	4.71E-03 (-)	1.21E-04 (-)	5.36E-01 (+)	1.21E-04 (-)	7.62E-03 (-)	3.56E-03 (-)	1.35E-01 (+)	2.10E-02 (+)	2.65E-03 (-)	6.65E-04 (-)
C9	3.96E-03 (-)	1.91E-03 (-)	2.01E-03 (-)	1.91E-03 (-)	1.28E-03 (-)	3.56E-02 (≈)	3.80E-01 (+)	2.80E-03 (-)	2.65E-02 (≈)	1.74E-02 (≈)
C10	5.26E-12 (-)	4.44E-13 (-)	1.35E-12 (-)	4.44E-13 (-)	1.07E-12 (-)	3.23E-12 (-)	2.56E-02 (+)	6.53E-03 (-)	2.40E-12 (-)	1.44E-12 (-)
L (-)	9	10	8	10	8	9	5	8	9	9
W (+)	0	0	2	0	1	0	5	2	0	0
T (≈)	1	0	0	0	1	1	0	0	1	1

Table 15 Ranking the eleven algorithms based on the Friedman's for the CEC2019 benchmark functions

Algorithms	Friedman's rank	Generalized rank
Clb-GWO	3.8472	1
IGWO	5.2828	2
CLPSO	5.7109	3
MEGWO	5.7719	4
GWO	6.2291	5
SOGWO	6.9104	6
GABC	7.1222	7
IGWODE	7.5165	8
CGWO	7.7684	9
ChOA	8.7428	10
WOA	10.621	11

avoiding entrapment to a greater degree. Considering that most real-world scenarios have higher dimensionality ($D > 10$ up to 1000), diversification through enhanced exploration is the key to achieving the best optimality.

The time complexity of all algorithms in the comparative analysis has been calculated as per the CEC2020 documentation and is given in Table 11.

The computational times for Clb-GWO are on the higher side as it employs multiple strategies and adaptive measures that work in coordination with each other. Clb-GWO's computational times are slightly lower compared to that of SOGWO and IGWO which require individual distance evaluations of all population members with respect to each other. The higher computational times can be overcome using parallel computational capabilities with modern computing systems as the multi-population-based algorithmic structure of Clb-GWO can be modified to allow both the population pools to operate independent of each other. The same modification cannot be made in the canonical GWO nor its variants as they follow a linear structure that requires the computations to take place in a top to bottom approach as their algorithmic structure dictates. The parallel compute capabilities of Clb-GWO can be extended to CPU's with virtualization technologies to further boost the computational speeds and achieve lower computational times which is a constraint in the standard GWO and its variants.

4.3 CEC2019 benchmarking suite

The CEC2019 test suite from Special Session and Competition on Single Objective Numerical Optimization in 2019 introduced 10 special functions to be minimized with limited control parameter “*tuning*” for each function [54]. The test functions were meticulously crafted with multiple local optima and one unique global optimal solution to ensure that the exploratory prowess and local minima avoidance characteristics are put to test. Similar to composition functions from the previous CEC sessions, the CEC 2019 benchmark suite presents challenging exploratory conditions with their landscape shifted and rotated to further complicate the search process of an algorithm. It is to be noted that these functions are extremely challenging for any global optimization algorithm to determine the global optimal solution as their formulation is such that they are intended to trap the algorithms at local best positions, especially for algorithms designed with a tendency to converge to the central point of the

Table 16 Comparison of the computational times (ms) of the eleven algorithms for the CEC2019 benchmark functions

	GWO	IGWO	ChOA	MEGWO	WOA	SOGWO	CLPSO	ABC	CGWO	IGWODE	Clb-GWO
C1	2.70E+04	1.03E+05	1.42E+05	2.23E+04	2.47E+04	1.59E+05	4.94E+04	3.02E+04	2.11E+04	3.93E+04	1.97E+04
C2	1.90E+04	9.68E+04	1.92E+05	1.17E+04	1.26E+04	2.02E+05	3.91E+04	1.60E+04	1.48E+04	2.77E+04	1.98E+04
C3	1.90E+04	9.51E+04	2.16E+05	1.13E+04	1.16E+04	2.02E+05	3.40E+04	1.53E+04	1.48E+04	2.77E+04	1.90E+04
C4	1.96E+04	1.07E+05	1.49E+05	1.28E+04	1.48E+04	1.64E+05	3.87E+04	1.86E+04	1.53E+04	2.85E+04	1.54E+04
C5	1.98E+04	1.09E+05	1.48E+05	1.37E+04	1.51E+04	1.64E+05	3.82E+04	1.95E+04	1.55E+04	2.88E+04	1.48E+04
C6	7.64E+04	1.34E+05	1.53E+05	7.52E+04	7.79E+04	1.61E+05	8.67E+04	7.16E+04	5.96E+04	1.11E+05	7.41E+04
C7	1.67E+04	9.01E+04	1.19E+05	1.13E+04	1.34E+04	1.29E+05	3.64E+04	1.71E+04	1.30E+04	2.43E+04	1.56E+04
C8	1.60E+04	9.05E+04	1.20E+05	1.02E+04	1.28E+04	1.29E+05	3.45E+04	1.57E+04	1.25E+04	2.33E+04	1.45E+04
C9	1.54E+04	9.08E+04	1.17E+05	9.83E+03	1.19E+04	1.29E+05	3.27E+04	1.56E+04	1.20E+04	2.24E+04	1.42E+04
C10	1.99E+04	1.05E+05	1.42E+05	1.38E+04	1.44E+04	1.60E+05	4.37E+04	2.00E+04	1.55E+04	2.90E+04	1.46E+04

Table 17 Comparison of mean and the standard deviation of Clb-GWO for the variations in the value of the non-linear control vector $(\vec{\phi})$ for the CEC2020 benchmarking suite

	Sinusoidal vector		Exponentially decreasing and oscillating vector		Linear decreasing vector		Chaotic map - Logistic		Chaotic map - Tent		Exponentially decreasing vector	
	Mean	std	Mean	std	Mean	std	Mean	std	Mean	std	Mean	std
F1	1.89E+01	2.27E+01	2.76E+01	9.93E+01	0.00E+00	0.00E+00	1.05E+01	1.40E+01	1.79E+01	5.14E+01	0.00E+00	0.00E+00
F2	2.37E+02	1.43E+02	3.02E+02	1.19E+02	3.33E+02	2.86E+02	2.91E+02	1.69E+02	2.65E+02	1.52E+02	2.52E+02	1.72E+02
F3	1.87E+01	3.26E+00	2.09E+01	4.86E+00	1.92E+01	3.96E+00	1.88E+01	3.94E+00	1.88E+01	3.35E+00	1.96E+01	4.19E+00
F4	8.82E-01	3.81E-01	7.33E-01	3.48E-01	9.18E-01	3.53E-01	8.23E-01	2.50E-01	8.28E-01	3.14E-01	8.21E-01	3.85E-01
F5	1.08E+02	8.21E+01	1.04E+02	7.84E+01	1.51E+02	1.48E+02	1.09E+02	8.39E+01	1.18E+02	1.37E+02	1.07E+02	8.98E+01
F6	9.30E-01	1.65E-01	6.45E-01	1.91E-01	4.42E-01	2.01E-01	5.79E-01	1.62E-01	5.73E-01	1.50E-01	4.59E-03	1.58E-03
F7	4.43E+01	7.04E+01	3.12E+01	5.56E+01	6.18E+01	1.05E+02	3.87E+01	6.14E+01	5.17E+01	7.34E+01	3.80E+01	4.50E+01
F8	5.20E+01	4.40E+01	4.69E+01	4.36E+01	5.66E+01	3.92E+01	5.62E+01	4.71E+01	6.12E+01	4.49E+01	7.13E+00	3.63E+00
F9	1.00E+02	1.85E-13	1.21E+02	4.20E+01	1.03E+02	1.86E+01	1.03E+02	1.86E+01	1.14E+02	3.57E+01	1.10E+02	3.11E+01
F10	2.72E+02	1.38E+02	2.07E+02	1.27E+02	2.79E+02	1.31E+02	2.71E+02	1.47E+02	2.86E+02	1.35E+02	2.94E+02	1.27E+02

Table 18 Comparison of mean and the standard deviation of Clb-GWO for the variations in the value of Competition rate (*comp*) for the CEC2020 benchmarking suite

	<i>fr10/sr5</i>		<i>fr20/sr10</i>		<i>fr30/sr15</i>	
	Mean	std	Mean	std	Mean	std
F1	2.88E+01	1.44E+01	0.00E+00	0.00E+00	2.28E+01	5.30E+01
F2	3.29E+02	1.80E+02	2.65E+02	1.11E+02	3.03E+02	1.31E+02
F3	1.94E+01	4.14E+00	1.61E+01	2.92E+00	1.91E+01	2.91E+00
F4	8.39E-01	2.84E-01	7.73E-01	2.81E-01	8.56E+00	3.17E-01
F5	7.33E+01	4.25E+01	7.31E+01	5.70E+01	9.47E+01	9.03E+01
F6	1.06E+00	1.61E-01	5.84E-04	1.51E-03	6.63E-01	1.88E-01
F7	8.43E+01	7.62E+01	4.37E+01	2.14E+01	9.24E+01	8.76E+01
F8	7.14E+01	4.05E+01	8.44E+00	3.54E+00	6.96E+01	4.37E+01
F9	1.00E+02	1.44E-13	1.10E+02	3.24E+01	1.20E+02	4.31E+01
F10	3.39E+02	1.26E+02	3.19E+02	1.31E+02	2.99E+02	1.41E+02

search landscape. Additionally, these problems have a large number of dimensions making the search process even harder and complex and only the algorithms with a higher exploratory tendency of the entire search space can determine the global optimal solution or generate solutions in close proximity to the global best.

The description of the CEC2019 benchmarking suite is shown in Table 12.

The benchmarking results (mean and standard deviation) are shown in Table 13, the *p*-values of Wilcoxon's rank-sum test and the results are shown in Table 14, the results of Friedman's non-parametrical test are shown in Table 15 followed by the average computational times in Table 16. A maximum of 1,000,000 function evaluations were allowed for all the algorithms with 30 independent runs.

Analysis of results

- Clb-GWO entrapment evasion capabilities helped it attain higher accuracy and precision for the CEC2019 test suite. Compared to its competitors, it recorded lower instances of stagnation and reached the global optima for most test functions. While its competitors,

Table 19 Comparison of mean and the standard deviation of Clb-GWO for the variations in the N:T ratio for the CEC2020 benchmarking suite

	N=710, T=710 and N:T=1:1		N=500, T=1000 and N:T=1:2		N=350, T=1425 and N:T=1:4		N=250, T=2000 and N:T=1:8	
	Mean	std	Mean	std	Mean	std	Mean	std
F1	6.38E+01	4.07E+01	4.56E+00	3.65E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	3.16E+02	1.45E+02	3.13E+02	2.04E+02	3.84E+02	1.98E+02	2.63E+02	1.81E+02
F3	2.07E+01	5.28E+00	2.07E+01	5.82E+00	1.98E+01	3.89E+00	2.08E+01	5.64E+00
F4	7.99E-01	2.99E-01	8.97E-01	4.23E-01	9.45E-01	3.38E-01	9.83E-01	4.12E-01
F5	2.35E+02	1.34E+02	2.22E+02	1.15E+02	1.34E+02	1.09E+02	1.04E+02	7.35E+01
F6	6.85E-01	2.12E-01	6.45E-01	1.58E-01	6.58E-01	1.79E-01	7.35E-03	1.95E-03
F7	4.10E+01	6.36E+01	5.66E+01	6.85E+01	6.55E+01	7.64E+01	4.08E+01	1.15E+02
F8	6.03E+01	4.37E+01	5.77E+01	4.67E+01	6.83E+01	4.29E+01	9.54E+00	2.17E+00
F9	1.32E+02	8.71E+01	1.21E+02	5.36E+01	1.03E+02	3.24E+01	1.18E+02	5.86E+01
F10	3.53E+02	1.16E+02	3.71E+02	9.52E+01	3.47E+02	1.19E+02	2.81E+02	1.38E+02

Table 20 Description of the five classification datasets and three function approximation datasets used for MLP training

Type	Name of the Dataset	Mathematical Formulation	MLP Structure	Number of Attributes/Features	Number of inputs	Number of outputs	Number of classes	Number of training samples	Number of test samples	Number of dimensions	Optimization Range
Classification Datasets	XOR (3-Bit)	-	3-7-1	3	3	1	2	8	8	36	[-10, 10]
	Balloon	-	4-9-1	4	4	1	2	16	16	55	[-10, 10]
	Iris	-	4-9-3	4	4	3	3	150	150	75	[-10, 10]
	Breast Cancer	-	9-19-1	9	9	1	2	599	100	209	[-10, 10]
	Heart	-	22-45-1	22	22	1	2	80	187	1081	[-10, 10]
Function-Approximation Datasets	Sigmoid	$y = \frac{1}{1+e^x}$	1-15-1	1	1	1	-	61: x in [-3.0;0.1;3.0]	121: x in [-3.0;0.05;3.0]	46	[-10, 10]
	Cosine	$y = \left(\cos\left(\frac{x\pi}{7}\right)\right)^7$	1-15-1	1	1	1	-	31: x in [1.25;0.05;2-.75]	38: x in [1.25;0.04;2-.75]	46	[-10, 10]
	Sine	$y = \sin(2x)$	1-15-1	1	1	1	-	126: x in [-2π;0.1;2π]	252: x in [-2π;0.05;2π]	46	[-10, 10]

Table 21 Statistical results and the classification rates (CR) of the 11 algorithms for the MLP training with the five classification datasets

Algorithms	XOR			Balloon			Iris			Cancer			Heart		
	Mean	Std.	CR (%)	Mean	Std.	CR (%)	Mean	Std.	CR (%)	Mean	Std.	CR (%)	Mean	Std.	CR (%)
WOA	2.77E-02	3.72E-02	100	4.99E-06	1.57E-05	100	6.20E-02	9.97E-02	89.33	1.19E-03	6.01E-05	98	9.26E-02	1.25E-02	42.5
ChOA	4.17E-02	5.05E-02	100	6.44E-03	1.59E-02	100	3.56E-01	7.84E-02	45.33	1.62E-03	2.53E-04	98	1.39E-01	1.70E-02	41.25
IGWO-DE	1.36E-02	2.07E-02	100	1.11E-06	3.52E-06	100	6.01E-02	4.53E-02	88.67	8.30E-03	3.76E-03	94	1.17E-01	3.68E-02	67.5
MEGWO	1.96E-04	2.74E-04	100	1.99E-08	5.94E-08	100	4.45E-02	2.78E-02	86.67	1.15E-03	4.72E-05	99	1.08E-01	1.48E-02	36.25
SOGWO	6.14E-02	4.46E-02	100	3.58E-05	2.47E-05	100	2.20E-01	1.01E-01	62.67	2.11E-03	1.55E-03	98	1.50E-01	4.03E-03	31.25
IGWO	1.55E-02	2.88E-02	100	3.76E-10	7.97E-10	100	5.87E-02	1.02E-01	88.67	1.18E-03	4.43E-05	98	9.46E-02	1.04E-02	85
CGWO	2.81E-04	3.76E-04	100	6.68E-13	1.25E-13	100	3.44E-02	5.18E-03	83.33	1.44E-02	3.60E-03	86	1.12E-01	1.53E-02	83.75
CLPSO	1.08E-01	4.46E-02	37.5	6.74E-03	2.11E-02	100	3.38E-01	1.34E-01	26.67	1.68E-03	2.26E-04	99	1.13E-01	1.14E-02	77.5
GABC	3.34E-03	4.20E-03	100	7.88E-07	2.17E-06	100	7.29E-02	1.96E-02	54.67	2.00E-02	5.15E-03	94	1.67E-01	1.41E-02	72.5
GWO	8.01E-04	1.07E-03	100	9.36E-12	1.41E-12	100	2.49E-02	2.02E-03	88.67	8.27E-03	4.33E-03	96	9.52E-02	8.60E-02	47.5
Clb-GWO	8.28E-06	1.75E-05	100	7.69E-16	4.12E-15	100	2.22E-02	1.65E-03	91.33	1.11E-03	2.03E-05	99	8.23E-02	2.54E-02	87.5

Table 22 Statistical results and the test errors (TE) of the 11 algorithms for the MLP training with the three function approximation datasets

Algorithms	Sigmoid			Cosine			Sine		
	Mean	Std.	TE (%)	Mean	Std.	TE (%)	Mean	Std.	TE (%)
WOA	2.46E-01	1.12E-04	17.5337	1.77E-01	5.73E-04	4.7975	4.33E-01	1.55E-02	135.9979
ChOA	2.48E-01	1.26E-03	17.6221	1.92E-01	1.67E-02	4.9064	4.46E-01	1.60E-02	136.9244
IGWO-DE	2.47E-01	5.22E-04	17.7071	1.79E-01	2.70E-03	4.7904	4.34E-01	1.61E-02	138.6926
MEGWO	2.47E-01	1.62E-04	17.7831	1.78E-01	1.21E-03	4.7555	4.39E-01	3.83E-03	140.9139
SOGWO	2.47E-01	1.02E-03	17.5883	1.82E-01	1.07E-02	4.9302	4.46E-01	4.37E-03	143.7239
IGWO	2.46E-01	3.21E-05	17.6034	1.76E-01	3.57E-04	4.6671	4.32E-01	1.91E-02	132.5466
CGWO	2.47E-01	5.56E-05	17.7032	1.77E-01	4.31E-04	4.7364	4.38E-01	8.14E-03	135.9174
CLPSO	2.47E-01	5.92E-05	17.7824	2.71E-01	9.55E-02	4.8607	4.56E-01	3.85E-03	147.0052
GABC	2.47E-01	8.48E-05	17.6730	1.78E-01	8.08E-04	4.7718	4.31E-01	5.68E-03	142.0812
GWO	2.47E-01	7.30E-05	17.7149	1.77E-01	4.64E-04	4.8715	4.22E-01	1.19E-02	133.9117
Clb-GWO	2.46E-01	3.96E-05	17.4875	1.76E-01	3.22E-04	4.6334	4.19E-01	4.65E-03	132.2601

could only exploit the unimodal cases, i.e., function C1, Clb-GWO was effective for both unimodal and multi-modal functions as well.

- No tuning modifications except the population size and iterations were made to obtain the results.
- It is interesting to note that the performance of Clb-GWO has been poor for C7 (Shifted and Rotated Schwefel’s Function) which is the same function as F2 in the CEC2020 benchmarking suite indicating its weakness in exploring its complex landscape.
- Compared to the variants of GWO, Clb-GWO had the best optimal fitness for C1, C4, C5, C7, C8, C9 and C10. It is indicative of the algorithm’s exploratory and local minima avoidance capabilities. The proposed method generated solutions closer to the global optimal solutions for C4, C5, C6, C7, C8 and C10. For the functions C2, C4 and C5, the proposed method outperformed the other variants of GWO and the modern meta-heuristics.
- The learning and adaptive learning schemes are key to ensure that a thorough exploration of the search space is achieved with the divergence and convergence groups helping Clb-GWO to prevent clustering at local optimal points and extend the duration of exploration for the most possible time in the search process.

Table 23 Computational times (seconds) of the 11 algorithms for the MLP training for all the datasets

Algorithm	XOR	Balloon	Iris	Cancer	Heart	Sigmoid	Cosine	Sine
GWO	4.642	78.485	278.327	446.169	215.674	115.582	106.834	117.868
WOA	3.631	83.016	257.956	445.080	200.824	107.525	101.193	114.947
CGWO	5.042	88.242	254.763	436.795	203.752	108.403	102.482	115.960
IGWO	6.679	89.123	254.760	428.304	201.027	109.602	104.303	118.164
IGWO-DE	4.083	82.041	253.026	418.558	196.434	106.365	102.575	113.588
SOGWO	6.028	84.999	255.145	419.367	226.428	105.835	104.119	113.457
MEGWO	4.006	82.006	250.228	421.606	187.560	103.204	100.736	110.059
ChOA	18.567	101.218	261.798	448.081	436.681	118.178	114.410	124.201
CLPSO	4.718	83.013	243.539	409.368	191.807	104.569	100.259	112.333
GABC	4.287	81.694	243.405	396.128	181.685	104.776	99.997	111.433
Clb-GWO	4.236	80.352	259.565	435.156	195.156	105.654	103.457	114.655

Table 24 Comparison of the Statistical results and the classification rates (CR) of the algorithms from the literature and Clb-GWO for the XOR classification dataset

Algorithms	Ref	XOR						
		Mean	Std.	CR (%)	NFEs	Pop Size	Iterations	Rank
GWO	[40]	9.41E-03	2.95E-04	100	12,500	50	250	3
PSO	[40]	8.41E-02	3.59E-02	37.5	12,500	50	250	5
GA	[40]	1.81E-04	4.13E-04	100	12,500	50	250	2
ACO	[40]	1.80E-01	2.53E-02	62.5	12,500	50	250	7
ES	[40]	1.19E-01	1.16E-02	62.5	12,500	50	250	6
PBIL	[40]	3.02E-02	3.97E-02	62.5	12,500	50	250	4
Clb-GWO	This work	8.28E-06	1.75E-05	100	25,000	50	250	1

- The non-linear hunting aids in sampling vast areas of the complex landscapes promoting diversity as the exploration switches to exploitation towards the end of the search process. The advantage of the non-linear hunting is evident for functions C2, C6 and C10 wherein the variants of GWO had fallen prey to local entrapment.
- C1 and C6 were the easiest followed by C10 and C2. Functions C4, C7, C8 and C9 were the most challenging at achieving the perfect score with little to no improvement in the accuracy of the digits within the maximum NFEs.
- As explained earlier for the CEC2020 test suite, the higher computational times of Clb-GWO on account of its multi-population approach and adaptive measures' can be minimized through the adoption of parallel computational methods and virtualization technologies that allow multiple CPU cores to share the computational burden. The standard GWO on account of its linear algorithmic approach requires less overall computational times but cannot be extended to parallel systems as it has a single population pool whose members are updated one after the other without any sub-divisions amongst them.

4.4 Sensitivity to tuning parameters

Clb-GWO incorporates two tuning /parameters namely, the non-linear control vector ($\vec{\phi}$) to balance exploration and exploitation phases and the Competition rate (*comp*) which is used to

Table 25 Comparison of the Statistical results and the classification rates (CR) of the algorithms from the literature and Clb-GWO for the Balloon classification dataset

Algorithms	Ref	Balloon						
		Mean	Std.	CR	NFEs	Pop Size	Iterations	Rank
GWO	[40]	9.38E-15	2.81E-14	100	12,500	50	250	3
PSO	[40]	5.85E-04	7.49E-04	100	12,500	50	250	5
GA	[40]	5.08E-24	1.06E-23	100	12,500	50	250	1
ACO	[40]	4.85E-03	7.76E-03	100	12,500	50	250	6
ES	[40]	1.91E-02	1.70E-01	100	12,500	50	250	7
PBIL	[40]	2.49E-05	5.27E-05	100	12,500	50	250	4
Clb-GWO	This work	7.69E-16	4.12E-15	100	25,000	50	250	2

Table 26 Comparison of the Statistical results and the classification rates (CR) of the algorithms from the literature and Clb-GWO for the Iris classification dataset

Algorithms	Ref	Iris						
		Mean	Std.	CR	NFEs	Pop Size	Iterations	Rank
GWO	[40]	2.29E-02	3.20E-03	91.33	50,000	200	250	2
PSO	[40]	2.29E-01	5.72E-02	37.33	50,000	200	250	5
GA	[40]	8.99E-02	1.24E-01	89.33	50,000	200	250	3
ACO	[40]	4.06E-01	5.38E-02	32.66	50,000	200	250	7
ES	[40]	3.14E-01	5.21E-02	46.66	50,000	200	250	6
PBIL	[40]	1.16E-01	3.64E-02	86.66	50,000	200	250	4
Clb-GWO	This work	2.22E-02	1.65E-03	91.33	25,000	50	250	1

adapt the competitive learning strategies over the problem landscapes. These values must be determined through empirical analysis for the best exploration and exploitation trade-off and to prevent local entrapment. Additionally, the ratio of the population count (N) and the number of iterations (T) has to be assessed to analyze and deduce the optimal population count and the number of iterations required by both algorithms to perform effectively. Hence an experimental setup through the CEC2020 benchmarking suite with 10 dimensions is chosen with 1,000,000 NFEs. Three experimentations to determine the perfect setting of the tuning parameters (non-linear control vector ($\vec{\phi}$), Competition rate (*comp*) and the N:T ratio) for the perfect trade-off of the exploration and exploitation are opted and the mean and standard deviations of Clb-GWO are recorded for 30 independent runs.

4.4.1 Influence of the non-linear control vector ($\vec{\phi}$)

The non-linear control vector ($\vec{\phi}$) balances the exploration and exploitation phases in the modified GWO procedure and ensures are population diversity and prevents the algorithm from drifting towards the geometric centre of the landscape It set to decrease exponentially from 1 to 0 over the course of iterations after a comparative analysis of other linear, non-linear and chaotic vectors as described below.

Firstly, its value is tested using the sinusoidal and exponentially decreasing and oscillating vectors in the range 0 and 1. Additionally, two chaotic Logistic map and Tent map (which have been the most opted chaotic maps in several chaotic variants of GWO) are chosen to

Table 27 Comparison of the Statistical results and the classification rates (CR) of the algorithms from the literature and Clb-GWO for the Cancer classification dataset

Algorithms	Ref	Cancer						
		Mean	Std.	CR	NFEs	Pop Size	Iterations	Rank
GWO	[40]	1.20E-03	7.45E-05	99.00	50,000	200	250	2
PSO	[40]	3.49E-02	2.47E-03	11.00	50,000	200	250	6
GA	[40]	3.03E-03	1.50E-03	98.00	50,000	200	250	4
ACO	[40]	1.35E-02	2.14E-03	40.00	50,000	200	250	3
ES	[40]	4.03E-02	2.47E-03	6.00	50,000	200	250	7
PBIL	[40]	3.20E-02	3.07E-03	7.00	50,000	200	250	5
Clb-GWO	This work	1.11E-03	2.03E-05	99	25,000	50	250	1

Table 28 Comparison of the Statistical results and the classification rates (CR) of the algorithms from the literature and Clb-GWO for the Heart classification dataset

Algorithms	Ref	Heart						
		Mean	Std.	CR	NFEs	Pop Size	Iterations	Rank
GWO	[40]	1.23E-01	7.70E-03	75.00	50,000	200	250	3
PSO	[40]	1.89E-01	8.94E-03	68.75	50,000	200	250	5
GA	[40]	9.30E-02	2.25E-02	58.75	50,000	200	250	2
ACO	[40]	2.28E-01	4.98E-03	0.00	50,000	200	250	7
ES	[40]	1.92E-01	1.52E-02	71.25	50,000	200	250	6
PBIL	[40]	1.54E-01	1.82E-02	45.00	50,000	200	250	4
Clb-GWO	This work	8.23E-02	2.54E-02	87.5	25,000	50	250	1

generate values between 0 and 1. The mean and standard deviations of the selected test functions for the two experimentations are provided in Table 17 and it can be observed that the exponentially decreasing vector in the range 1 to 0 provided a better balance of population diversity and intensification.

4.4.2 Influence of competition rate (*comp*)

The Competition rate (*comp*) is one of the key parameters to be tuned appropriately to ensure that the algorithm switches between linear and adaptive learning based on the complexity of the landscapes. The Competition rate is based on the failure rate or the success rate of the given learning strategy. The failure rate is a measure of the number of consecutive failures associated with the learning strategy and the success rate is vice-versa. The given learning strategy must fail a higher number of times before switching to the other learning strategy as the inferior solutions are ruled out from entering the population pool. At the same time, setting the success rate to a higher value than the failure rate deprives the population of diversity and lead to stagnation associated with the repetition of the same learning strategy. This setting is very crucial such that the linear learning strategy is capable of balancing the exploration and exploitation on its own based on the linear descent rate and is more likely to drive the population to the geometric centre. On the other hand, adaptive learning is dependent on the three dominant wolves and leads to entrapment if not adjusted. Hence the failure rate and

Table 29 Comparison of performance of various mechanisms for ANN training from the literature and Clb-GWO

Ref, Year	Datasets	Optimization Algorithm	Mechanism	Evaluation metric	Standard Deviation
2019	DS2OS traffic traces	NA	LR, SVM, DT, RF, ANN	Accuracy LR: 98.3% SVM: 98.2% DT: 99.4% RF: 99.4% ANN: 99.4%	LR: 0.0055 SVM: 0.0064 DT: 0.016 RF: 0.014 ANN: 0.021
Current work	XOR Balloon Iris Cancer Heart	Clb-GWO Clb-GWO Clb-GWO Clb-GWO Clb-GWO	MLP-FNN MLP-FNN MLP-FNN MLP-FNN MLP-FNN	CR=100% CR=100% CR=91.33% CR=99% CR=87.5%	1.75E-05 4.10E-15 1.65E-03 2.03E-05 2.54E-02

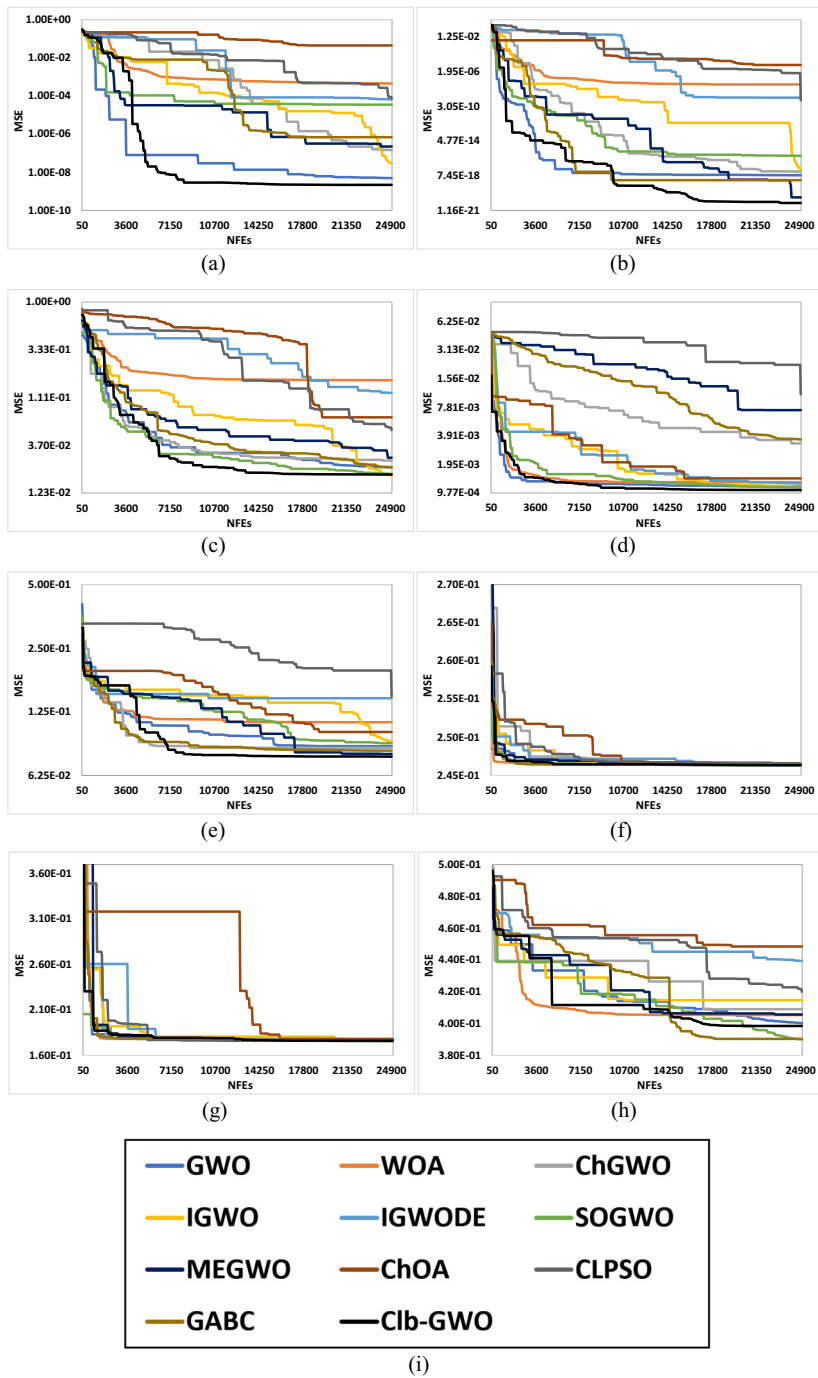


Fig. 5 Convergence characteristics of the 11 meta-heuristics for the MLP training for (a) XOR dataset (b) Balloon dataset (c) Iris (d) Cancer dataset (e) Heart dataset (f) Sigmoid dataset (g) Cosine dataset (h) Sine dataset (i) Legend

success rate follow the ratio $fr:sr = 2:1$ and their sum gives the competition rate which is always a multiple of 15. To assess the variation of the optimization outcomes, testing is performed with the competitive rate set to 15, 30 and 45 respectively and the results are shown in Table 18. It can be observed that the setting *comp* to 30 with '*fr*' at 20 and '*sr*' at 10 yields the best balance between the learning strategies.

4.4.3 Influence of the N:T ratio

The optimal ratio of the population size and iterations to match the NFEs can play an important role in determining how effectively these numbers translate to optimality. Clb-GWO is formulated with a dual search strategy and it relies on two function evaluations in every iteration and four ratios have been experimented with to determine the optimal ratio for the algorithm to effectively corresponds to optimal resource utilization. From Table 19, it has been observed that the ratios 1:4 and 1:8 have been the most successful at delivering a perfect balance between the global and local search within the set NFEs.

5 Multi-layer perceptron training

In order to demonstrate the effectiveness of the proposed algorithm towards handling of complex real-world with higher problem dimensions, five classification datasets and three function approximation datasets for the MLP training from the recent literature have been considered. The same algorithms are chosen with the previously set configurations for the algorithm tuning settings and a comprehensive comparative analysis is provided below.

5.1 Problem description

The MLP training is accomplished through Feedforward neural networks (FNNs) with input, hidden and output layers. The optimization algorithms are then integrated to determine the optimal combination of weights and biases within the given upper and lower bounds to achieve the highest classification/prediction accuracy. The optimal solution vector to be determined by the optimization algorithm is an array of weights and biases represented by Eq. (25).

$$\vec{S} = \{ \vec{W}, \vec{B} \} = \{ W_{1,1}, W_{1,2}, \dots, W_{n,n}, B_1, B_2, \dots, B_n \} \quad (25)$$

where, \vec{S} denotes the solution vector containing the weights and biases, \vec{W} is the sub-vector containing all the weights and \vec{B} is the sub-vector with all the biases, $W_{i,j}$ denotes the connection weight between the i^{th} and j^{th} nodes, n is the total number of input nodes, B_j denotes the bias (threshold) of the j^{th} hidden node and $j = 1, 2, \dots, h$ denotes the j^{th} hidden node and h denotes the total number of hidden nodes.

Following it, the weighted sums of inputs are calculated as per Eq. (26).

$$\omega_j = \sum_{i=1}^n (W_{i,j} I_i) - B_j \quad (26)$$

where, ω_j is the weighted sum of inputs for the j th hidden node, I_i is the i^{th} input.

Table 30 Description of the algorithm-specific tuning parameters for all the algorithms used in the comparative analysis

	Algorithm	Tuning / Algorithm-specific Parameters	Value
Standard variant	GWO	Control Vector (\vec{a}) to balance exploration and exploitation phases	Follows a linearly decrementing nature from an initial value of 2 to a final value of 0 over the progression of iterations.
Modern meta-heuristics	WOA	Control Vector (\vec{a}) to balance exploration and exploitation phases	linearly decreased from 2 to 0 over the course of iterations
	ChOA	Coefficient Vector (\vec{A})	Randomized in the interval $[-1, 1]$
		Chaotic Vector (m)	Tent chaotic map
Latest advanced variants of GWO	I-GWO	Control Vector (\vec{a}) to balance exploration and exploitation phases	Follows a linearly decrementing nature from an initial value of 2 to a final value of 0 over the progression of iterations.
		Radius (R_i)	Euclidean distance
	CGWO	Control Vector (\vec{a}) to balance exploration and exploitation phases	Follows a linearly decrementing nature from an initial value of 2 to a final value of 0 over the progression of iterations.
		Chaotic Map	Chebyshev map
	MEGWO	Scale factor (ρ)	Set between 0.5 and 0.1 ² based on $\rho \sim N(0.5, 0.1^2)$
		Global-best guidance rate (GR)	0.8
		Dispersion rate (DR)	DR _{max} =0.4, DR _{min} =0
	IGWODE	SR	SR _{max} =1, SR _{min} =0.6
		Control Vector (\vec{a}) to balance exploration and exploitation phases	Follows a linearly decrementing nature from an initial value of 2 to a final value of 0 over the progression of iterations.
		Control Vector (\vec{a}) to balance exploration and exploitation phases	Follows a linearly decrementing nature from an initial value of 2 to a final value of 0 over the progression of iterations.
Crossover Probability (CR)		0.8	
Dynamic Scaling Factor (F)		$F = f_{min} + (f_{max} - f_{min}) \times \left[\frac{T - (t-1)}{T} \right]$ where f_{max} and f_{min} are the maximum and minimum values of the scaling factor (2 and 0)	
Elimination Factor (R)		random integer between $N/(2 \times \varepsilon)$ and N/ε through $R = \left\lfloor \frac{N}{\varepsilon}, \frac{N}{0.75} \times \varepsilon \right\rfloor$ where ε is scale factor of wolves updating	
SOGWO	Spearman’s correlation coefficient (r_R)	Set between 0 and 1 through $r_R = 1 - \left[\frac{6 \times \sum (d_i)^2}{D \times (D^2 - 1)} \right]$	
	Control Vector (\vec{a}) to balance exploration and exploitation phases	Follows a linearly decrementing nature from an initial value of 2 to a final value of 0 over the progression of iterations.	
State-of-the-art advanced	CLPSO	Threshold variable Acceleration constants (cc_1 and cc_2) Inertia weight (w) Refreshing gap (m)	decreased linearly with iteration from \vec{a} Both set to 1.49445 $0.9 - [1 \text{ to } NFE_{max}] \times (0.7 / NFE_{max})$ Set to 7

Table 30 (continued)

Algorithm	Tuning / Algorithm-specific Parameters	Value
meta--heuristics	Learning Probability (P_c)	Particles from 1 to 30 have a P_c value ranging from 0.05 to 0.5 based on $P_{c_i} = 0.05 + 0.45 \times \frac{[\exp(\frac{10(i-1)}{ps-1})-1]}{(\exp(10)-1)}$
GABC	Strategy Limit	$ABC/best/1$ $0.6 \times SN \times D$ where, SN stands for population scale and D is the number of problem dimensions random number in the range $[-1,1]$ 300
Clb-GWO	Φ Chaotic iteration (K) with a sinusoidal iterator Control Vector ($\vec{\Phi}$) to promote population diversity	Follows an exponentially decrementing nature from an initial value of 1 to a final value of 0 over the progression of iterations.
	Competition rate ($comp$)	$fr + sr$ where $fr=20, sr=10$

The output of every individual hidden node is then computed as shown in Eq. (27).

$$H_j = Sigmoid(\omega_j) = \frac{1}{(1 + \exp(-\omega_j))} \tag{27}$$

where, H_j is the output of the j^{th} hidden node.

The weighted sum outputs of the hidden nodes are then calculated as per Eq. (28).

$$o_k = \sum_{j=1}^h (w_{j,k} \cdot H_j) - B_k' \tag{28}$$

Where, o_k is the weighted sum of inputs for the k^{th} output node, $w_{j,k}$ denotes the connection weight between the j^{th} and k^{th} nodes, B_k' denotes the bias (threshold) of the k^{th} output node.

The output of every output is then computed as per Eq. (29).

$$O_k = Sigmoid(o_k) = \frac{1}{(1 + \exp(-o_k))} \tag{29}$$

The objective function is simply formulated as Mean Square Error (MSE) where a given set of training samples is applied to the MLP and the difference between the desired output and the value that is obtained from the MLP. Finally, the performance of an MLP is evaluated based on the average of MSE over all the training samples as denoted by Eq. (30).

$$Minimize : F(\vec{S}) = \overline{MSE} \tag{30}$$

where,

$$\overline{MSE} = \sum_{t=1}^T \frac{\sum_{y=1}^Y (a_y^t - d_y^t)^2}{T}$$

where, $t = 1, 2, \dots, T$ denotes the current training sample and T denotes the total number of training samples, $y = 1, 2, \dots, Y$ denotes the current input and Y denotes the total number of outputs, a_y^t is the actual output of the y^{th} input unit when the t^{th} training sample appears in the input, d_y^t is the desired output of the y^{th} input unit when the t^{th} training sample appears in the input,

5.2 Experimental setup

A detailed description of the datasets is presented in Table 20 and the optimization is terminated upon reaching the maximum NFEs for 10 independent runs. In order to have a fair comparison, all the algorithms are given 25,000 NFEs with an initial population of 50. The statistical results of the optimization are given in Table 21 and Table 22 for the classification datasets and the function approximation datasets respectively. The computational times are given in Table 23.

It is evident from Table 21 that the classification training and testing efficiencies of Clb-GWO are excellent with similar computational times to its competitors. The performance of Clb-GWO was best for the XOR and balloon datasets and it managed to achieve the least MSE for both with 100% classification rates. Furthermore, the MSE obtained by Clb-GWO has been the least for all five of the classification datasets indicating its superiority in training neural networks. One important outcome of the optimization has been the effective handling of problem dimensions by Clb-GWO and this is clearly demonstrated in the Heart dataset where the highest classification rate of 87.5% was achieved by Clb-GWO with the least training MSE. Compared to the canonical GWO, the classification rate has more than doubled while the other variants of GWO excluding SOGWO and MEGWO fail to achieve higher classification rates. A major reason for their poor performance is the lack of strong exploration capabilities with the necessary adaptive frameworks to support the hunt of grey wolves to advance their hunt during the search process. Furthermore, the Heart dataset helps evaluate the proposed algorithm's immunity to *the curse of dimensionality* and highlights the excellent solution diversification system in Clb-GWO. In addition to these, the lower standard deviation rates for Clb-GWO help affirm its consistency in dealing with multi-modal large dimensional problems with enhanced exploration and exploitation.

The performance of Clb-GWO for the function approximation datasets (from Table 22) was more or less similar to its competitors and it can be observed the proposed methods present a slight advantage in training and testing with the least possible error rates. However, Clb-GWO remained the most consistent of them with the least possible deviation rates for all three cases. Given that the search landscapes for the function approximation datasets are composed of multiple peaks and valleys, most algorithms often converge to local optimal solutions as seen with the current testing. The convergence to the global optimal solution in these cases requires higher computational budgets well above the current considerations and while requiring additional training samples to improve the accuracy of testing. Nevertheless, the function-approximation datasets provide a good platform to validate the consistency of the meta-heuristic and it can be concluded that Clb-GWO is the most consistent of the testing group.

The computational times recorded by Clb-GWO are quite similar to the standard GWO despite its adaptive frameworks and multi-strategy approach. A closer inspection reveals that in all the cases except the balloon dataset, the computational times of Clb-GWO are

lower than GWO by about 2 to 5%. This is on account of the greedy selection integrated into the competitive learning strategies that avoid population updation in case of an inferior solution. The standard GWO on the other hand updates the population of all its members irrespective of their fitness levels requiring higher computational times for larger dimensional problems.

Tables 24, 25, 26, 27 and 28 compare the statistical results of Clb-GWO with several other algorithms including GWO, PSO, GA, Ant Colony Optimization (ACO), Evolution Strategy (ES) and Population-based Incremental Learning (PBIL) from [40].

On comparing Clb-GWO's performance with the other algorithms for the five classification datasets from the literature, it can be found that the overall efficiency of Clb-GWO is higher as it ranks on the top for all the datasets. Excluding the balloon dataset, Clb-GWO ranked first in terms of training and classification efficiency. Despite the fact that the competitor algorithms for XOR and Balloon datasets utilized half the computational budget, the performance to budget ratio of Clb-GWO is better as it recorded better mean and lower deviation rates accounting for its enhanced precision. For the latter, the MSEs and classification rates obtained by Clb-GWO are better than the rest as it utilized half the computational resources. The classification rate of Clb-GWO has never fallen below 85% indicating its superiority in training neural networks with larger datasets. The classification rate achieved by Clb-GWO at 87.5% for the Heart dataset has been the highest so far and it manages to reach the same with half the number of function evaluations compared to the methods from the literature.

A comparison of the performance of other methods such as Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF) and Artificial Neural Network (ANN) from [17] are provided in Table 29.

From Table 29, the performance of Clb-GWO for MLP-FNN training is on par with the other contemporary mechanisms as it achieved higher classification rates of 99% or above for three datasets. Furthermore, the lower standard deviation rates achieved for the MLP FNN training indicates the consistency and reliability of Clb-GWO for MLP training.

The convergence curves for the various datasets comparing all the algorithms are provided in Fig. 5.

Observing the nature of convergence, the proposed method has quicker convergence capabilities compared to its competitors. The convergence speeds for the five classification datasets have been dominated by Clb-GWO as it surpassed its competitors demonstrating its enhanced exploitation capabilities. It can also be observed that at times of stagnation, the adaptive triggers in Clb-GWO enable it to diversify the population such that stagnation is avoided. This is clearly seen in Fig. 5 (a), (b), (c) and (e) where the MSE drops down in sequences as the feedback system tunes the search direction to prevent entrapment as seen with ChOA and GWO. Similar convergence characteristics were obtained for all the algorithms with the function approximation models. However, Clb-GWO remained the fastest as it managed to converge within the least possible time span.

6 Conclusion

This article realizes an improved meta-heuristic optimization technique known as Clb-GWO with dual search strategies and population sub-division structured in a selective complimentary arrangement to improve population diversity through the development of difference vectors and learning strategies. The proposed method has a better performance in handling constrained and

unconstrained problems and has been effective at avoiding local entrapment in complex search landscapes. A better balance of exploration and exploitation with accelerated convergence has been witnessed across the various test cases with statistically significant performance.

Furthermore, the benchmarking analysis of the CEC2020 test functions proved that Clb-GWO was quite capable of effectively utilizing the increased computational resources and delivering optimal solutions for the increments in the problem dimensions. The results obtained were statistically significant (Wilcoxon's rank-sum test) and Clb-GWO ranked first in most test cases in Friedman's test. Clb-GWO stood immune to *the curse of dimensionality* and had little to no performance deterioration for the increased number of problem dimensions. Ten special benchmark functions from the CEC2019 benchmark suite to validate the exploratory skills and avoidance of local optima with challenging search landscapes comprising of several calculatedly placed local optimal solutions. The performance of Clb-GWO in comparison to GWO, its variants and the modern meta-heuristics was better having mean values closer to the best values and also having the least standard deviation. The proposed method avoided local entrapment with good exploratory capabilities in several cases compared to its competitors most of which fell prey to the local entrapment while recording lower computational times in most cases. The MLP training for different cases was dominated by Clb-GWO in terms of optimality, computational times and lower deviation. In terms of solution optimality and accuracy, Clb-GWO's performance was superior compared to the meta-heuristics from the literature with higher performance to computational cost ratio. This indicates that the proposed method can effectively handle problems with multiple constraints and a large number of dimensions.

6.1 Merits and demerits

The improvement and enhancement techniques in meta-heuristics have contributed to the betterment and have aided researchers to push the potential of optimization to new heights. These advancements although unique to one specific meta-heuristic can be experimented with the other algorithms and have opened up greater opportunities in the pursuit of the perfect optimization paradigm. Although quite successful and efficient in many cases, the improvement techniques have their fair share of criticism and complications. The "*No free lunch theorem*" summarizes that the perfect optimization algorithm for every optimization task is not practically feasible and an optimization technique that excels with one class of optimization problems may not perform adequately when deployed for other classes of optimization problems. Hence, to have a fair and unprejudiced view, the merits and the demerits associated with the proposed method are discussed below.

6.1.1 Merits

- The increasing problem complexity with the dimensions of the problem did not affect the performance of Clb-GWO for CEC2020 functions. The same performance could not be witnessed in the standard GWO algorithm nor its variants and the two modern meta-heuristics chosen from the comparison indicates that Clb-GWO is immune to *the curse of dimensionality*. This is attributed to the competitive learning phase and the greedy selection strategy promoting elitism and avoiding local entrapment while effectively repositioning the position of the wolves from the modified GWO process. The balance

of exploration and exploitation system by the various competitive leaving strategies is the other reason for its dominant performance in most benchmarking scenarios.

- The population sub-division into the majority and minority groups dedicated to a specific purpose following the dual search strategy has been the stronghold of the proposed method. Exploration and population diversity with the algorithm's dependency on the three dominant wolves has been balanced through this structuring of the algorithm.
- A good exploration and exploitation balance has been possible through the empirically set values of the non-linear control vector ($\vec{\phi}$) and competition rate (*comp*). The proposed method delivered solutions at the global optimum or closer to it in most complex benchmarking cases avoiding local entrapment and also had good convergence characteristics with the empirically tuned parametric settings.
- The system of double fitness evaluations (DFEs) has been a positive reinforcement allowing the modified GWO procedure and the competitive learning procedure to synergistically work with respect to each other and help reposition the omega wolves with increased population diversity encouraging better coverage of the search landscape.
- The linear and adaptive competitive learning strategies coupled with the greedy selection strategy promoted elitism by selecting the best wolves and the best solution is assigned as the alpha wolf and passed on to the standard GWO procedure in the next iteration to further refine and explore the solutions and help the algorithm to gain a better knowledge of the search landscape. An example of this can be seen in the CEC2019 benchmarking test and the standard engineering problems where Clb-GWO avoided local entrapment and delivered better optimal solutions while maintaining a minimal standard deviation of its results.
- The dependence of GWO on the alpha, beta and delta wolves to reposition every omega wolf is lowered in Clb-GWO. The formulation of unique difference vectors with the inclusion of randomised omega wolves allows for better information exchange between the different wolves and prevents local entrapment associated with a lower diversified population system.
- The greedy selectins strategy adopted in the competitive learning phase allows only the solutions with superior fitness to enter the population pool for the next generation. This system promotes elitism allowing the algorithm to concentrate its search on the potentially best areas within the search landscape. Global search is prioritised through the greedy selection and local search is prioritised through the initial Mu, Lambda selection the standard GWO procedure.

6.1.2 Limitations

- The incorporation of double fitness evaluations (DFEs) although beneficial to the performance of Clb-GWO, requires the lowering of the population size or the number of iterations for predefined NFES. This has been witnessed in all the benchmarking and other testing scenarios where the population size had to be lowered by half to match the required NFES while the iteration count was the same. Lower population size may result in reduced population diversity and the worst cases lead to local entrapment. Although Clb-GWO showed a greater capability at dodging the local entrapment, the choice to either drop the

population size or iteration count to match the other algorithms with single fitness evaluations (SFEs) has to be dealt with by the practitioner through meticulous planning.

- Despite the fact that the empirically set values for the non-linear control vector ($\vec{\phi}$) and competition rate (*comp*) resulted in good optimization outcomes, but the scope for additional tuning and modification persists. The performance of Clb-GWO for the CEC2019 was better only for seven out of the ten functions and for the others, the solutions were in close proximity to the global optimal solution. This indicates that through additional tuning management, a better performance suited for the problems' search landscape is possible.
- Although Clb-GWO benefits from a higher iterative count, its performance was comparatively better for the MLP training with only 250 iterations. The benchmarking for the CEC2020 with 5D test case was the only limiting condition as the lower number of functions evaluations prevented it from fully converging.
- The dependence of Clb-GWO on the random omegas (at least 7 different omega wolves) mandates the population size to be set above seven at all times. The algorithm may fail to run with a population size lower than seven.

6.2 Future scope

Clb-GWO can be deployed to a wide spectrum of problems falling under artificial intelligence, power systems, machine learning etc. Practitioners are free to modify the proposed method as per their requirements and hence to encourage such an extendibility, simplicity has been embraced in the design of Clb-GWO. The proposed method can be applied to various other optimization areas in power systems such as electric vehicle (EV) optimization, power electronics, smart grid integration, distribution systems, power dispatch problems, control systems, power quality enhancement etc. In computer science, the proposed method can be deployed toward neural network (NN) training (convolution NNs). Image classification, data classification, pattern recognition etc. can be optimized through the proposed methods. A plan to deploy the current method for the infection detection of COVID-19 from the X-ray images via a support vector classifier is in its roots. Feature selection is a potential area of application of the proposed methods through the formulation of a binary version of Clb-GWO. The realization of a multi-objective variant is a possibility for tackling problems requiring a Pareto-optimal front. The idea to develop a multi-objective variant for the optimization of energy management in EVs has been planned as a project. In electrical engineering, Clb-GWO can be adopted and modified for parameter estimation in photovoltaic cells and battery management systems. While penalty function has been considered for most of the benchmarking with the current method, other constraint handling techniques can be integrated and experimented with for several of the existing problems. The proposed competitive learning strategies can be extended to other meta-heuristics for experimental analysis towards its improvement.

Furthermore, Clb-GWO can be deployed to tackle problems at strategic, tactical and operational levels for ride-sharing systems like Uber, Ola etc. Optimal carpooling is another possibility that can be realized through Clb-GWO as it can effectively handle problems with higher dimensionality. Real-time optimization in several domains including automated sensing, sensor fusion, and electric vehicles can benefit from the exploratory capabilities of Clb-GWO and artificial intelligence techniques.

Appendix

Availability of source codes:

- The MATLAB code of Clb-GWO is available at the GitHub repository: <https://github.com/VAMSIKRISHNA-Optimization/A-Competitive-learning-based-Grey-Wolf-Optimizer-for-Engineering-Problems-and-its-application-to-Pow.git>
- The MATLAB code of CEC2020 test suite is available at the GitHub repository: <https://github.com/P-N-Suganthan/2020-Bound-Constrained-Opt-Benchmark>
- The MATLAB code of CEC2019 test suite is available at the GitHub repository: <https://github.com/P-N-Suganthan/CEC2019>
- The MATLAB code of MLP training for all the eight datasets is available at: https://in.mathworks.com/matlabcentral/fileexchange/52273-grey-wolf-optimizer-for-training-multi-layer-perceptrons-all-classification-and-function-datasets?s_tid=ta_fx_results

CRedit taxonomy Aala Kalananda Vamsi Krishna Reddy: Conceptualization, Methodology, Software Aala Kalananda Vamsi Krishna Reddy: Data curation, Writing- Original draft preparation. **Komanapalli Venkata Lakshmi Narayana and Aala Kalananda Vamsi Krishna Reddy:** Visualization, Investigation. **Komanapalli Venkata Lakshmi Narayana:** Supervision.: **Komanapalli Venkata Lakshmi Narayana:** Software, Validation.: **Komanapalli Venkata Lakshmi Narayana:** Writing- Reviewing and Editing.

Funding The authors declare that no funding has been received for the current manuscript.

Declarations

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Abdel-Basset M, Abdel-Fatah L, Sangaiah AK (2018) Metaheuristic algorithms: a comprehensive review. Elsevier Inc.
2. Alba E, Luque G, Nasmachnow S (2013) Parallel metaheuristics: recent advances and new trends. *Int Trans Oper Res* 20:1–48. <https://doi.org/10.1111/j.1475-3995.2012.00862.x>
3. Ali R, Abd M, Lu S (2018) Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert Syst Appl* 108:1–27. <https://doi.org/10.1016/j.eswa.2018.04.028>
4. Amirsadri S (2018) A levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training. *Neural Comput Applic* 30:3707–3720. <https://doi.org/10.1007/s00521-017-2952-5>
5. Benabderrahmane S (2017) Combining boosting machine learning and swarm intelligence for real time object detection and tracking: towards new meta-heuristics boosting classifiers. *Int J Intell Robot Appl* 1: 410–428. <https://doi.org/10.1007/s41315-017-0037-3>
6. Bhaskar A, Sharma C, Mohiuddin K et al (2022) A robust video watermarking scheme with squirrel search algorithm. *Comput Mater Contin* 71:3069–3089. <https://doi.org/10.32604/cmc.2022.019866>
7. Binh NTM, Binh HTT, Van Linh N, Yu S (2020) Efficient meta-heuristic approaches in solving minimal exposure path problem for heterogeneous wireless multimedia sensor networks in internet of things. *Appl Intell* 50:1889–1907. <https://doi.org/10.1007/s10489-019-01628-9>

8. Blum C, Puchinger J, Raidl GR, Roli A (2011) Hybrid metaheuristics in combinatorial optimization: a survey. *Appl Soft Comput J* 11:4135–4151. <https://doi.org/10.1016/j.asoc.2011.02.032>
9. Chakhlevitch K, Cowling P (2008) Hyperheuristics: recent developments. *Stud Comput Intell* 136:3–29. https://doi.org/10.1007/978-3-540-79438-7_1
10. Darwish A (2018) Bio-inspired computing: algorithms review, deep analysis, and the scope of applications. *Futur Comput Informatics J* 3:231–246. <https://doi.org/10.1016/j.fcij.2018.06.001>
11. Del Ser J, Osaba E, Molina D et al (2019) Bio-inspired computation: where we stand and what's next. *Swarm Evol Comput* 48:220–250. <https://doi.org/10.1016/j.swevo.2019.04.008>
12. Dhargupta S, Ghosh M, Mirjalili S, Sarkar R (2020) Selective opposition based Grey wolf optimization. *Expert Syst Appl* 151:113389. <https://doi.org/10.1016/j.eswa.2020.113389>
13. Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A (2019) A survey on new generation metaheuristic algorithms. *Comput Ind Eng* 137:106040. <https://doi.org/10.1016/j.cie.2019.106040>
14. Faris H, Aljarah I, Al-Betar MA, Mirjalili S (2018) Grey wolf optimizer: a review of recent variants and applications. *Neural Comput Appl* 30:413–435. <https://doi.org/10.1007/s00521-017-3272-5>
15. Gharehchopogh FS, Gholizadeh H (2019) A comprehensive survey: whale optimization algorithm and its applications. *Swarm Evol Comput* 48:1–24. <https://doi.org/10.1016/j.swevo.2019.03.004>
16. Gupta S, Deep K (2019) A novel random walk Grey wolf Optimizer. *Swarm Evol Comput* 44:101–112. <https://doi.org/10.1016/j.swevo.2018.01.001>
17. Hasan M, Islam MM, Zarif MII, Hashem MMA (2019) Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things (Netherlands)* 7:100059. <https://doi.org/10.1016/j.iot.2019.100059>
18. He M, Hong L, Yang ZY, Yang TB, Zeng J (2018) Bioactive assay and hyphenated chromatography detection for complex supercritical CO2 extract from Chaihu Shugan san using an experimental design approach. *Microchem J* 142:394–402. <https://doi.org/10.1016/j.microc.2018.07.016>
19. Heidari AA, Pahlavani P (2017) An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. *Appl Soft Comput J* 60:115–134. <https://doi.org/10.1016/j.asoc.2017.06.044>
20. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. *Futur Gener Comput Syst* 97:849–872. <https://doi.org/10.1016/j.future.2019.02.028>
21. Igiri CP, Singh Y, Poonia RC (2019) A review study of modified swarm intelligence: particle swarm optimization, firefly, bat and grey wolf Optimizer algorithms. *Recent Adv Comput Sci Commun* 13:5–12. <https://doi.org/10.2174/2213275912666190101120202>
22. Johnvictor AC (2020) Critical review of bio-inspired optimization techniques. *WIREs Comput Stat* 14(1). <https://doi.org/10.1002/wics.1528>
23. Kar AK (2016) Bio inspired computing - a review of algorithms and scope of applications. *Expert Syst Appl* 59:20–32. <https://doi.org/10.1016/j.eswa.2016.04.018>
24. Kaur J, Kalra A, Sharma D (2018) Comparative survey of swarm intelligence optimization approaches for ANN optimization. *Adv Intell Syst Comput* 624:305–314. https://doi.org/10.1007/978-981-10-5903-2_32
25. Khishe M, Mosavi MR (2020) Chimp optimization algorithm. *Expert Syst Appl* 149:113338. <https://doi.org/10.1016/j.eswa.2020.113338>
26. Kishor A, Singh PK (2016) Empirical Study of Grey Wolf Optimizer. 1037–1049. <https://doi.org/10.1007/978-981-10-0448-3>
27. Kohli M, Arora S (2018) Chaotic grey wolf optimization algorithm for constrained optimization problems. *J Comput Des Eng* 5:458–472. <https://doi.org/10.1016/j.jcde.2017.02.005>
28. Kuestogullari H, Leeson MS, Kole B, Hines EL (2014) Meta-heuristic algorithms for optimized network flow wavelet-based image coding. *Appl Soft Comput J* 14:536–553. <https://doi.org/10.1016/j.asoc.2013.09.001>
29. Li S, Chen H, Wang M, Heidari AA, Mirjalili S (2020) Slime mould algorithm: a new method for stochastic optimization. *Futur Gener Comput Syst* 111:300–323. <https://doi.org/10.1016/j.future.2020.03.055>
30. Liang JJ, Qin AK, Suganthan PN, Baskar S (2006) Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput* 10:281–295. <https://doi.org/10.1109/TEVC.2005.857610>
31. Liu S, Yu M, Li M, Xu Q (2019) The research of virtual face based on Deep convolutional generative adversarial networks using TensorFlow. *Phys A Stat Mech its Appl* 521:667–680. <https://doi.org/10.1016/j.physa.2019.01.036>
32. Long W, Jiao J, Liang X, Tang M (2018) Engineering applications of artificial intelligence an exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. *Eng Appl Artif Intell* 68: 63–80. <https://doi.org/10.1016/j.engappai.2017.10.024>
33. Lu C, Gao L, Li X, Hu C, Yan X, Gong W (2020) Chaotic-based grey wolf optimizer for numerical and engineering optimization problems. *Memetic Comput* 12:371–398. <https://doi.org/10.1007/s12293-020-00313-6>

34. Luo K (2019) Enhanced grey wolf optimizer with a model for dynamically estimating the location of the prey. *Appl Soft Comput J* 77:225–235. <https://doi.org/10.1016/j.asoc.2019.01.025>
35. Lynn N, Suganthan PN (2017) Ensemble particle swarm optimizer. *Appl Soft Comput J* 55:533–548. <https://doi.org/10.1016/j.asoc.2017.02.007>
36. Maier HR, Razavi S, Kapelan Z, Matott LS, Kasprzyk J, Tolson BA (2019) Introductory overview: optimization using evolutionary algorithms and other metaheuristics. *Environ Model Softw* 114:195–213. <https://doi.org/10.1016/j.envsoft.2018.11.018>
37. Mallipeddi R, Suganthan PN, Pan QK, Tasgetiren MF (2011) Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl Soft Comput J* 11:1679–1696. <https://doi.org/10.1016/j.asoc.2010.04.024>
38. Man KF, Tang KS, Kwong S (1996) Genetic algorithms: concepts and applications. *IEEE Trans Ind Electron* 43:519–534. <https://doi.org/10.1109/41.538609>
39. Mirjalili S (2015) The ant lion optimizer. *Adv Eng Softw* 83:80–98. <https://doi.org/10.1016/j.advengsoft.2015.01.010>
40. Mirjalili S (2015) How effective is the Grey wolf optimizer in training multi-layer perceptrons. *Appl Intell* 43:150–161. <https://doi.org/10.1007/s10489-014-0645-7>
41. Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
42. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf Optimizer. *Adv Eng Softw* 69:46–61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
43. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM (2017) Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191. <https://doi.org/10.1016/j.advengsoft.2017.07.002>
44. Mirjalili S, Aljarah I, Mafarja M et al Grey Wolf Optimizer : Theory, Literature Review, and Application in Computational Fluid Dynamics Problems. Springer International Publishing
45. Molina D (2018) An Insight into Bio-inspired and Evolutionary Algorithms for Global Optimization : Review, Analysis, and Lessons Learnt over a Decade of Competitions. 517–544
46. Movassagh AA, Alzubi JA, Gheisari M, Rahimi M, Mohan S, Abbasi AA, Nabipour N (2021) Artificial neural networks training algorithm integrating invasive weed optimization with differential evolutionary model. *J Ambient Intell Humaniz Comput*. <https://doi.org/10.1007/s12652-020-02623-6>
47. Nadimi-shahraki MH, Taghian S, Mirjalili S (2021) An improved grey wolf optimizer for solving engineering problems. *Expert Syst Appl* 166:113917. <https://doi.org/10.1016/j.eswa.2020.113917>
48. Nasrabadi MS, Sharafi Y, Tayari M (2016) A parallel grey wolf optimizer combined with opposition based learning. *1st Conf Swarm Intell Evol Comput CSIEC 2016 - proc* 18–23. <https://doi.org/10.1109/CSIEC.2016.7482116>
49. Negi G, Kumar A, Pant S, Ram M (2021) GWO : a review and applications. *Int J Syst Assur Eng Manag* 12:1–8. <https://doi.org/10.1007/s13198-020-00995-8>
50. Niu P, Niu S, Liu N, Chang L (2019) The defect of the Grey wolf optimization algorithm and its verification method. *Knowledge-Based Syst* 171:37–43. <https://doi.org/10.1016/j.knsys.2019.01.018>
51. Optimizer W (2020) Task assignment strategy for multi - robot based on improved Grey. *J Ambient Intell Humaniz Comput* 11:6319–6335. <https://doi.org/10.1007/s12652-020-02224-3>
52. Pan J (2017) *Advances-in-intelligent-information-hiding-and-multimedia-Signal-2018*. Pdf
53. Poli R, Kennedy J, Blackwell T (2007) Particle swarm optimization: an overview. *Swarm Intell* 1:33–57. <https://doi.org/10.1007/s11721-007-0002-0>
54. Price KV, Awad NH, Ali MZ, Suganthan PN (2018) Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization. 22
55. Rojas-Delgado J, Trujillo-Rasúa R, Bello R (2019) A continuation approach for training artificial neural networks with meta-heuristics. *Pattern Recogn Lett* 125:373–380. <https://doi.org/10.1016/j.patrec.2019.05.017>
56. Sabahno M, Safara F (2021) ISHO: improved spotted hyena optimization algorithm for phishing website detection. *Multimed Tools Appl* 81:34677–34696. <https://doi.org/10.1007/s11042-021-10678-6>
57. Saxena A, Kumar R, Das S (2019) β -chaotic map enabled Grey wolf Optimizer. *Appl Soft Comput J* 75: 84–105. <https://doi.org/10.1016/j.asoc.2018.10.044>
58. Sharma V, Tripathi AK (2022) A systematic review of meta-heuristic algorithms in IoT based application. *Array* 14:100164. <https://doi.org/10.1016/j.array.2022.100164>
59. Simons CL, Smith JE (2013) A comparison of meta-heuristic search for interactive software design. *Soft Comput* 17:2147–2162. <https://doi.org/10.1007/s00500-013-1039-1>
60. Singh N, Singh SB (2017) A novel hybrid GWO-SCA approach for optimization problems. *Eng Sci Technol an Int J* 20:1586–1601. <https://doi.org/10.1016/j.jestech.2017.11.001>
61. Stephenson M, Amarasinghe S, Martin M, O'Reilly UM (2003) Meta optimization: improving compiler heuristics with machine learning. *ACM SIGPLAN Not* 38:77–90. <https://doi.org/10.1145/780822.781141>

62. Tu Q, Chen X, Liu X (2019) Multi-strategy ensemble grey wolf optimizer and its application to feature selection. *Appl Soft Comput J* 76:16–30. <https://doi.org/10.1016/j.asoc.2018.11.047>
63. Wang J, Li S (2019) An improved Grey wolf Optimizer based on differential evolution and elimination mechanism. *Sci Rep* 9:7181. <https://doi.org/10.1038/s41598-019-43546-3>
64. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1: 67–82. <https://doi.org/10.1109/4235.585893>
65. Wu G, Mallipeddi R, Suganthan PN, Wang R, Chen H (2016) Differential evolution with multi-population based ensemble of mutation strategies. *Inf Sci (NY)* 329:329–345. <https://doi.org/10.1016/j.ins.2015.09.009>
66. Xu Q (2013) A novel machine learning strategy based on two-dimensional numerical models in financial engineering. *Math Probl Eng* 2013:1–6. <https://doi.org/10.1155/2013/659809>
67. Xu Q, Wu J, Chen Q (2014) A novel mobile personalized recommended method based on money flow model for stock exchange. *Math Probl Eng* 2014:1–9. <https://doi.org/10.1155/2014/353910>
68. Xu Q, Wang Z, Wang F, Gong Y (2019) Multi-feature fusion CNNs for Drosophila embryo of interest detection. *Phys A Stat Mech its Appl* 531:121808. <https://doi.org/10.1016/j.physa.2019.121808>
69. Xu Q, Wang F, Gong Y, Wang Z, Zeng K, Li Q, Luo X (2019) A novel edge-oriented framework for saliency detection enhancement. *Image Vis Comput* 87:1–12. <https://doi.org/10.1016/j.imavis.2019.04.002>
70. Xu Q, Huang G, Yu M, Guo Y (2020) Fall prediction based on key points of human bones. *Phys A Stat Mech its Appl* 540:123205. <https://doi.org/10.1016/j.physa.2019.123205>
71. Yang X-S (2015) Recent Advances in Swarm Intelligence and Evolutionary Computation, p 303. <https://doi.org/10.1007/978-3-319-13826-8>
72. Yue CT, Price KV, Suganthan PN et al (2019) Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization. Zhengzhou Univ Zhengzhou China Nanyang Technol Univ Singapore
73. Zakian AKP (2018) Improved GWO algorithm for optimal design of truss structures. *Eng Comput* 34:685–707. <https://doi.org/10.1007/s00366-017-0567-1>
74. Zhang Y, Wang S, Ji G (2015) A comprehensive survey on particle swarm optimization algorithm and its applications. *Math Probl Eng* 2015:1–38. <https://doi.org/10.1155/2015/931256>
75. Zhang X, Kang Q, Cheng J, Wang X (2018) A novel hybrid algorithm based on biogeography-based optimization and Grey wolf Optimizer. *Appl Soft Comput J* 67:197–214. <https://doi.org/10.1016/j.asoc.2018.02.049>
76. Zhang X, Wang X, Chen H, Wang D, Fu Z (2019) Improved GWO for large-scale function optimization and MLP optimization in cancer identification. *Neural Comput Applic* 4:1305–1325. <https://doi.org/10.1007/s00521-019-04483-4>
77. Zhang X, Lin Q, Mao W, Liu S, Dou Z, Liu G (2021) Hybrid particle swarm and Grey wolf Optimizer and its application to clustering optimization. *Appl Soft Comput J* 101:107061. <https://doi.org/10.1016/j.asoc.2020.107061>
78. Zhu G, Kwong S (2010) Gbest-guided artificial bee colony algorithm for numerical function optimization. *Appl Math Comput* 217:3166–3173. <https://doi.org/10.1016/j.amc.2010.08.049>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.