Expert Systems With Applications 238 (2024) 122147

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Greylag Goose Optimization: Nature-inspired optimization algorithm

El-Sayed M. El-kenawy [a], Nima Khodadadi [b], Seyedali Mirjalili [c,d], Abdelaziz A. Abdelhamid [e], Marwa M. Eid [f], Abdelhameed Ibrahim [g,*]

[a] Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura, 35111, Egypt
[b] Department of Civil, Architectural and Environmental Engineering, University of Miami, 1251 Memorial Drive, Coral Gables, FL, USA
[c] Centre for Artificial Intelligence Research and Optimization, Torrens University Australia, Fortitude Valley, QLD 4006, 35516, Australia
[d] University Research and Innovation Center, Obuda University, 1034 Budapest, Hungary
[e] Department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, 11566, Egypt
[f] Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura, Egypt
[g] Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University, Mansoura, 35516, Egypt

ARTICLE INFO

Keywords:
Swarm-based algorithms
Meta-heuristic
Algorithm
Engineering optimization

ABSTRACT

Nature-inspired metaheuristic approaches draw their core idea from biological evolution in order to create new and powerful competing algorithms. Such algorithms can be divided into evolution-based and swarm-based algorithms. This paper proposed a new nature-inspired optimizer called the Greylag Goose Optimization (GGO) algorithm. The proposed algorithm (GGO) belongs to the class of swarm-based algorithms and is inspired by the Greylag Goose. Geese are excellent flyers and during their seasonal migrations, they fly in a group and can cover thousands of kilometers in a single flight. While flying, a group of geese forms themselves as a "V" configuration. In this way, the geese in the front can minimize the air resistance of the ones in the back. This allows the geese to fly around 70% farther as a group than they could individually. The GGO algorithm is first validated by being applied to nineteen datasets retrieved from the UCI Machine Learning Repository. Each dataset contains a varied amount of characteristics, instances, and classes that are used to choose features. After that, it is put to use in the process of solving a number of engineering benchmark functions and case studies. Several case studies are solved using the proposed algorithm too, including the pressure vessel design and the tension/compression spring design. The findings demonstrate that the GGO method outperforms numerous other comparative optimization algorithms and delivers superior accuracy compared to other algorithms. The results of the statistical analysis tests, such as Wilcoxon's rank-sum and one-way analysis of variance (ANOVA), demonstrate that the GGO algorithm achieves superior results.

1. Introduction

Optimization is the process of finding the best solution to a certain problem among all of the alternatives. Optimization algorithms can be roughly divided into two classes, i.e. deterministic algorithms and stochastic intelligent algorithms, taking into account their nature. When using deterministic algorithms, the solutions to the same problems are always the same because the initial starting values are always the same (Brownlee, 2011; Yang, 2008). Stochastic algorithms, in contrast to deterministic ones, use random steps to attain their goals. The process of optimization can never be redone in this way. Nonetheless, in most cases, both of them can arrive at the exact final ideal solutions. It is possible to further categorize stochastic algorithms as either heuristic or metaheuristic. A heuristic algorithm is a technique for discovering the optimal solutions by trial and error. On the other hand, with some prior knowledge of random search, metaheuristic algorithms solve optimization problems stochastically. In other words, it is an optimization method that begins with a random solution. In the next iteration, it then randomly explores and exploits the available search space with a defined probability (Yang, Gandomi, Talatahari, & Alavi, 2012; Yıldız et al., 2009). Fig. 1 shows that population-based metaheuristic algorithms can be put into four groups: physics-based, evolutionary-based, bio-inspired, and swarm-based.

Bio-inspired metaheuristic techniques are approaches that draw their core idea from biological evolution in order to create new and powerful competing algorithms. These techniques have grown in popularity over the last two decades. This is due to their powerful and efficient performance in dealing with high-dimensional nonlinear optimization problems (Gandomi & Alavi, 2011; Yang et al., 2012). These techniques have the capability of exploiting the relevant information

* Corresponding author.
E-mail addresses: skenawy@ieee.org (E.-S.M. El-kenawy), nima.khodadadi@miami.edu (N. Khodadadi), ali.mirjalili@torrens.edu.au (S. Mirjalili), abdelaziz@cis.asu.edu.eg (A.A. Abdelhamid), mmm@ieee.org (M.M. Eid), afai79@mans.edu.eg (A. Ibrahim).

https://doi.org/10.1016/j.eswa.2023.122147
Received 19 February 2023; Received in revised form 11 October 2023; Accepted 11 October 2023
Available online 18 October 2023

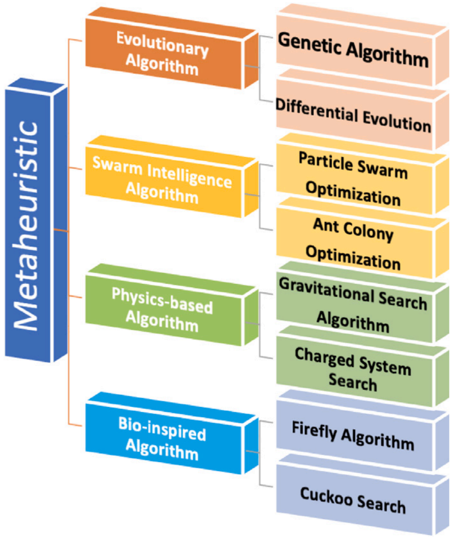

Fig. 1. Metaheuristic Algorithm Classification.

contained within the population in order to discover the most effective solution. Numerous scholars have introduced many studies on these algorithms to date, and several nature-inspired metaheuristic algorithms have been suggested, including the Bat Algorithm (BA) and Grey Wolf Optimization (GWO) algorithm. It is reported that there are about 150 different optimization algorithms for solving (getting the best results) optimization problems (Valdez, Castillo, & Melin, 2021). Nature-inspired algorithms can be broadly divided into two categories: biologically-based ones and natural-phenomena-based ones. The biologically-based algorithms can be further separated into evolution-based and swarm-based algorithms. Genetic algorithms, differential evolution, cultural evolution, and evolutionary techniques are examples of evolution-based algorithms. Examples of swarm-based algorithms include particle-swarming, ant colony, artificial bees, bats; birds; cats; and bacterial foraging. For the natural-phenomena-based algorithms, the gravitational search, simulated annealing, and the big bang big crunch are examples of algorithms of this type based on the physical laws (Valdez et al., 2021).

In addition to solving optimization problems, metaheuristic algorithms find widespread use in the domain of feature selection, which is concerned with the dimensions of the dataset in order to generate predictions. When data sets become more dimensional, the performance of classification algorithms degrades significantly (Bennasar, Hicks, & Setchi, 2015). Additionally, high-dimensional datasets suffer from several shortcomings, including a lengthy model creation time, poor performance, and redundant data, all of which make data analysis extremely challenging. To address this issue, feature selection is used as a significant preprocessing phase aimed at extracting a subset of features from a large data collection and improving the quality of the classification or clustering model, hence removing noisy, superfluous, and ambiguous data (El-kenawy, Ibrahim, Mirjalili, Eid, & Hussein, 2020). Using metaheuristic algorithms, various attempts have been made in the past to handle feature selection problems. These include Particle Swarm Optimization (PSO) (Gu, Cheng, & Jin, 2018), Genetic Algorithm (GA) (Kabir, Shahjahan, & Murase, 2011), Artificial Bee Colony (ABC) algorithm (Schiezaro & Pedrini, 2013) and Chicken Swarm Optimization (CSO) (Hafez, Zawbaa, Emary, Mahmoud, & Hassanien, 2015). Recently, other algorithms have also been used to discover the optimal set of features and these include binary bat algorithm (Qasim & Algamal, 2020), chaotic binary black hole algorithm (Qasim, Al-Thanoon, & Algamal, 2020), Butterfly Optimization Algorithm (Awad et al., 2020), Moth–Flame Optimization (Al-Tashi et al., 2022) and whale optimization (Nematzadeh, Enayatifar, Mahmud, & Akbari, 2019).

There are other meta-heuristic optimization algorithms with different versions; standard version such as Mountain Gazelle Optimizer (MGO) (Abdollahzadeh, Gharehchopogh, Khodadadi, & Mirjalili, 2022), Artificial Hummingbird Algorithm (AHA) (Zhao, Wang, & Mirjalili, 2022), Stochastic Paint Optimizer (SPO) (Kaveh, Talatahari, & Khodadadi, 2020b), Ebola Optimization Search Algorithm (EOSA) (Oyelade, Ezugwu, Mohamed, & Abualigah, 2022), and Al-Biruni Earth Radius (BER) (El-kenawy et al., 2023), Improved version such as Advanced Charged System Search (ACSS) (Kaveh, Khodadadi, Azar, & Talatahari, 2021), Improved Sine Cosine Algorithm (ISCA) (Abd Elaziz, Oliva, & Xiong, 2017), Dynamic Arithmetic Optimization Algorithm (DAOA) (Khodadadi, Snasel, & Mirjalili, 2022), Chaotic Hunger Games Search Optimization, (CHGSO) (Onay & Aydemır, 2022), Chaotic Harris Hawks Optimization (CHHO) (Ibrahim, Ali, Eid, & El-kenawy, 2020) and Dynamic Water Strider Algorithm (DWSA) (Kaveh, Eslamlou, & Khodadadi, 2020). Hybrid versions such as Hybrid Invasive Weed Optimization-Shuffled Frog-Leaping algorithm (IWO-SFLA) (Kaveh, Talatahari, & Khodadadi, 2020a), Hybrid Grey Wolf Optimization-Particle Swarm Optimization (GWO-PSO) (El-Kenawy & Eid, 2020) and Hybrid genetic-Grasshopper Algorithm (GA-GOA) (Arrif et al., 2022). Multi-objective version such as Multi-objective Thermal Exchange Optimization algorithm (MOTEO) (Khodadadi, Talatahari, & Dadras Eslamlou, 2022), Multi-objective Lichtenberg Algorithm (MOLA) (Pereira, Oliver, Francisco, Cunha, & Gomes, 2022), Multi-objective Cuckoo Search Algorithm (MOCSA) (Chandrasekaran & Simon, 2012), Multi-objective Artificial Hummingbird Algorithm (MOAHA) (Zhao, Zhang, et al., 2022), and Multi-objective artificial vultures optimization (MOAVOA) (Khodadadi, Soleimanian Gharehchopogh, & Mirjalili, 2022), that we have not yet covered. This work discussed a few of the most important ones. In addition, some other methods were applied in different fields (Al-Tashi et al., 2022). These meta-heuristics, which are based on a variety of local and global search strategies, provide researchers with a greater selection of algorithms from which they can choose to address optimization issues in a variety of fields.

The Greylag Goose Optimization (GGO) algorithm is the name given to the optimization strategy that is being proposed in this paper. The GGO algorithm kicks off with the generation of numerous individuals through random chance. Each person offers a potential solution that can be considered for inclusion in the pool of solutions to the issue. The GGO algorithm's dynamic group behavior places each member into one of two groups: an exploration group and an exploitation group. With each iteration, the number of possible solutions in each group is dynamically adjusted in accordance with the solution that was determined to be the best. The goose explorer will conduct a hunt for interesting new places to investigate in the area immediately surrounding its current position. The responsibility of bettering the solutions that are currently in place falls on the shoulders of the exploitation team. At the end of each cycle, the GGO determines which participants have achieved the highest level of fitness and awards them appropriately.

The contribution of this paper can be summarized as follows.

- A novel Greylag Goose Optimization (GGO) algorithm, a new nature-inspired optimizer
- A novel binary GGO (bGGO) algorithm for feature selection.
- The GGO algorithm is first validated by being applied to nineteen datasets retrieved from the UCI Machine Learning Repository.
- It is put to use in the process of solving a number of engineering benchmark functions and case studies.
- In order to solve problems involving constrained optimization, the GGO algorithm is utilized. The pressure vessel design and the tension/compression spring design are the two designs that have been validated through testing.
- The Wilcoxon rank-sum test and the ANOVA test are used to evaluate the statistical significance of the GGO algorithm.
- The findings highlight that the GGO method outperforms numerous other comparative optimization algorithms and delivers superior accuracy compared to other algorithms than other algorithms.

The remaining parts of this work are structured as described in the following paragraphs. In Section 2, we talk about the work that is been done in connection with the most recent and innovative algorithms. Section 3 discusses both the recommended GGO algorithm as well as the binary GGO method. Additionally, an analysis of the complexity of both of these algorithms is included. In the following section, " Section 4, " we will analyze the outcomes of the tests. In the last section, we draw a close to this endeavor and offer some recommendations for the years to come.

2. Related work

The optimal solution can be found with the help of a meta-heuristic algorithm, which is a form of a random algorithm. The two most significant issues that are associated with meta-heuristic algorithms are premature convergence and getting stuck in local optimal points. Meta-heuristic algorithms have been developed as a means of addressing such problems. Meta-heuristic algorithms, a type of approximation optimization algorithm, can be used to solve a wide variety of problems by delivering answers for how to get out of local minima. In recent decades, many different kinds of meta-heuristic algorithms have been proposed and presented. Meta-heuristic algorithms can be categorized using a variety of different criteria. The most well-known of this category is population-based. As the proposed algorithm "Greylag Goose Optimization (GGO)" belongs to the population-based algorithms, we will focus our literature review about these algorithms which include Particle Swarm Optimization (PSO) (Eberhart & Kennedy, 1995), Grey Wolf Optimization (GWO) (Mirjalili, Mirjalili, & Lewis, 2014), Artificial Bee Colony (ABC) (Karaboga, 2010) and Ant Colony Optimization (ACO) (Dorigo, Birattari, & Stutzle, 2006).

Particle Swarm Optimization (PSO) is among the most well-known swarm optimization algorithms. It was introduced by Eberhart and Kennedy (1995). PSO is inspired by the collective behavior of birds and fish. It has been employed in several studies to optimize the final selection of feature subsets. In Xue, Zhang, and Browne (2014), the authors proposed a new initialization method of POS and integrated it with the best strategies of particle updating for enhancing the feature selection process. The suggested PSO-feature selection method was based on the wrapper approach and used signal objectives for numerical applications. The reported results showed that this method has improved the computational complexity by selecting a subset of significant and non-redundant features. By combining filter and wrapper techniques, Moradi and Gholampour (2016) suggested offering an efficient PSO-based technique for the feature selection problem. This technique proposed a novel local search technique that helped to identify the best subset consisting of non-redundant and significant features. This PSO-based technique used a single objective to address numerical and medical applications. Additionally, Rostami, Forouzandeh, Berahmand, and Soltani (2020) combined the graph theory and PSO to address the feature selection problem in medical applications to improve diagnosis accuracy. They used the node centrality criterion to propose a novel way of initializing the particles of the PSO algorithm. Using a multi-objective fitness function, they managed to classify the features into two sets: one contains the least similar features to one another, and the second contains the most relevant features to the target class. Using these two classes of features, a disease can be diagnosed.

GWO (Mirjalili et al., 2014) is one of the recent optimization algorithms inspired by grey wolves. It is modeled based on grey wolves' natural leadership structure and hunting mechanism. The literature shows several GWO-based methods aiming to select the best features for machine learning applications. Emary, Zawbaa, and Hassanien (2016) designed a binary variant of the GWO and used it in a new feature selection method for binary classification applications (e.g., medical applications). This feature selection method was based on the wrapper approach and used a single fitness function depending on classification accuracy and the number of picked features. Tu, Chen, and Liu (2019) developed a multi-strategy ensemble GWO for feature selection in order to improve the performance of prior GWO-based feature selection approaches. Tu's method was a wrapper-based feature selection approach, used GWO Single objective, and was designed for numerical and medical applications. Furthermore, Abdel-Basset, El-Shahat, El-henawy, de Albuquerque, and Mirjalili (2020) suggested a GWO-based wrapper feature selection technique in which they employed a mutation operator with the GWO for data classification of numerical and medical applications. The mutation operator indicated in this study attempts to remove features that are redundant or unrelated to the classification problem.

Another recent optimization algorithm based on swarm intelligence is the Artificial Bee Colony (ABC) (Karaboga, 2010) which is inspired by the behavior of honey bees. In the ABC algorithm, the colony of artificial bees consists of three groups of bees: employed bees, onlookers, and scouts. The employed artificial bees make up the first half of the colony, and the onlookers make up the second. There is just one employed bee per food source. In other words, the number of bees actively working in the hive equals the number of nearby food sources. When the other bees quit their food source, the employed bee acts as a scout. The ABC is a well-known swarm intelligence system that performs good optimization; however, it has several drawbacks. A novel ABC with an effective search method based on random neighborhood structure, known as RNSABC, was proposed by Ye et al. (2022) in order to improve the performance of ABC. A new random neighborhood structure (RNS) was built in RNSABC. The size of each neighborhood is independent and arbitrary. Based on RNS, a better search approach has been developed. Additionally, a depth-first search approach was used to strengthen the onlooker bee phase's role. Cui, Hu, and Rahmani (2022) suggested an upgraded ABC with dynamic composition (ABCDC) in order to enhance its performance. Due to the fact that the original ABC and most of its variations have a fixed ratio between employed and onlooker bees, there are not enough onlooker bees to fully utilize the seeking space in a finite amount of time. Therefore, they suggested a mechanism to more effectively identify the global optimum by adjusting the number of employed bees and bystander bees. In order to increase the diversity of the initial population, a symmetric Latin hypercube design was used. Additionally, in both the employed bee phase and the spectator bee phase, two differential search equations with self-adaptive parameters were applied.

Taking cues from the cooperative nature of insects and other animals, swarm intelligence is a cutting-edge method for tackling complex problems. Specifically, ant colony optimization, a general-purpose optimization methodology, is one of many methodologies and techniques that have been inspired by ants. The concept of Ant Colony Optimization (ACO) (Dorigo et al., 2006) is based on the strategies employed by certain species of ants when searching for food. For the purpose of signaling a good route to the rest of the colony, these ants use pheromones they leave behind. Over the years since the initial ant colony optimization algorithm was introduced, ACO has gained the interest of a growing number of researchers and found widespread use in a variety of practical settings. An improved ant colony optimization (ICMPACO) algorithm based on the multi-population strategy, co-evolution mechanism, pheromone updating strategy, and pheromone diffusion mechanism was developed by Deng, Xu, and Zhao (2019). To enhance the convergence rate and prevent a trap into the local optimum value, the proposed ICMPACO method breaks the optimization problem into many sub-problems and classifies the ants in the population as either elite ants or common ants. The optimization capacity was enhanced by using the pheromone updating approach. Using the pheromone diffusion process, ants can generate a pheromone that has an effect on a wide area from where it was initially released. When it comes to implementing information sharing, the co-evolution mechanism was employed to facilitate communication between various groups within a population. The experimental findings demonstrated the effectiveness of the proposed ICMPACO algorithm in solving the traveling salesman problem (TSP). Kaveh and Talatahari (2010) described an enhanced ant colony optimization (IACO) method for solving engineering design problems under severe constraints. Due to its sub-optimization technique, IACO could deal with both continuous and discrete issues (SOM). By utilizing the finite element method as a search-space update mechanism, SOM was built on a foundation of efficiency and accuracy. In addition to minimizing the need for as many iterations, SOM could also reduce pheromone matrices and decision vectors. Although IACO shortened the time spent optimizing and performing pheromone update operations, it did not improve the likelihood of an optimal solution being found.

Swarm Intelligence is a multi-agent, current AI technique that has shown great promise for tackling practical optimization issues using meta-heuristic approaches. This method has some merits and demerits. Advantages are such as: despite the failure of several agents, the overall task is nevertheless done successfully (Robust), the colony operates independently of any higher authority (Modularity), and rapid spread of network modifications is possible (Speed). On the other hand, it is hard to anticipate behavior just on the rules themselves (Behavior); without understanding how an agent operates, it is impossible to comprehend how a colony works (Knowledge) and any deviation from these elementary norms alters the collective behavior (Sensitivity) are some disadvantages.

3. Proposed Greylag Goose Optimization

The inspiration and mathematical algorithm of the proposed Greylag Goose Optimization are explained in this section.

3.1. GGO inspiration

The social behavior as well as the dynamic activity of geese served as the basis for the inspiration for the GGO algorithm.

3.1.1. Geese: Social behavior

Geese are known for their loyalty. They form lifelong bonds with their spouses and are very protective of their children. Often, they prefer to keep close to an ill or injured partner or chick. They keep doing so while the winter season is approaching and the rest of the flock departs for warmer climes. When a goose's partner dies, the goose goes into isolation, and some geese may spend the remainder of their lives like a single goose, refusing to marry again for the rest of their lives (Horton, 2008).

Geese take pleasure in posturing their feathers while hunting for food in the grass and gathering leaves and sticks to improve their nests. In the spring of each year, hatch eggs. The male geese protect the eggs hidden in their homes while the females nurse them for thirty days. Some geese prefer to put eggs in the same nest year after year while possible (Indiana DNR, 2022).

3.1.2. Geese: Dynamic behavior

A collection of geese families band together to form a bigger group known as a gaggle, in which the birds take care of one another, as shown in Fig. 2. In this group, there are typically one or two guards, "sentries", who keep an eye out for predators while the rest of the group is busy eating. The gaggle members alternate the role of the sentry, much like sailors on a ship keeping looking out for enemies or pirates. Healthy geese are known to look after injured colleagues and wounded geese will band together to protect one another from predators and to aid one another in finding food (Birdfact, 2022). Ducks are outgoing and social animals that are at their most comfortable when traveling in big groups, which they refer to as "paddlings" when they are on the water. They spend their days foraging for food in the grass or in the shallow water, and at night, they sleep with the other individuals with whom they paddle.

Ganders are capable of amazing feats of flight. They are able to travel thousands of kilometers in a single flight while on their annual migrations because they travel in large flocks, as shown in Fig. 2. While flying, a group of geese (flocks) forms themselves as a "V" configuration. In this way, the geese in the front can minimize the air resistance of the ones in the back. This allows the geese to fly around 70% farther as a group than they could individually. When the geese at the front got fatigued, they swapped to the back, allowing the ones in the back to be their encouragement to the leaders. Geese enjoy long memories helping them to reach their targets. They navigate during their annual migrations by relying on familiar landmarks and the stars for guidance (Green, 2004).

3.2. Greylag Goose Optimization (GGO) algorithm

The proposed optimization technique in this work is called the Greylag Goose Optimization (GGO) algorithm. The GGO algorithm starts with randomly generating several individuals, as shown in Algorithm 1. Each individual indicates a solution that can be a candidate solution to the problem. The GGO population is considered as $X_i (i = 1, 2, \dots, n)$ with size n representing a gaggle. An objective function, F_n, is selected to evaluate the individuals in the group. After calculating the objective function for each individual (agent) X_i, the best solution (leader) is selected and is indicated as P.

The Dynamic Groups behavior of the GGO algorithm divides all the individuals into an exploration group (n_1) and an exploitation group (n_2). The number of solutions in each group is managed dynamically with each iteration according to the best solution. The exploration group processes, with n_1 agents, and the exploitation group, with n_2 agents, are shown in Fig. 3. GGO initiates the groups with 50% exploration and 50% exploitation. Then, the number of agents in the exploration group (n_1) is decreased, and the number of agents in the exploitation group (n_2) is increased. However, if the best solution's objective function value did not change for three continuous iterations, the algorithm starts to increase the number of agents in the exploration group (n_1) to get another best solution and avoid local optima hopefully.

3.2.1. Exploration operation

As discussed in more detail below, exploration is responsible for both interesting locating areas of the search space and preventing local optimum stagnation by moving towards the ideal answer.

Moving towards the best solution: By employing this tactic, the geese explorer will search for promising new locations to explore near its current location. This is accomplished by repeatedly comparing the numerous potential nearby options to find the best one based on fitness. The GGO algorithm uses the following equations to achieve this for the $\mathbf{A}$ and $\mathbf{C}$ vectors updated as $\mathbf{A} = 2\mathbf{a}.r_1 - \mathbf{a}$ and $\mathbf{C} = 2.r_2$ during iterations with the $\mathbf{a}$ parameter changed linearly from 2 to 0:

$$\mathbf{X}(t+1) = \mathbf{X}^*(t) - \mathbf{A}.|\mathbf{C}.\mathbf{X}^*(t) - \mathbf{X}(t)| \tag{1}$$

where $\mathbf{X}(t)$ is an agent at an iteration t. The $\mathbf{X}^*(t)$ represents the best solution (leader) position. The $\mathbf{X}(t+1)$ is the updated position of the agent. The r_1 and r_2 values are changing within $[0, 1]$, randomly.

The following equation will be used based on choosing three random search agents (paddlings), named $\mathbf{X}_{Paddle1}$, $\mathbf{X}_{Paddle2}$, and $\mathbf{X}_{Paddle3}$, in order to force agents not to be affected by one leader position to get greater exploration. The position of the current search agent will be updated as follows for $|\mathbf{A}| \geq 1$.

$$\begin{aligned}\mathbf{X}(t+1) = & w_1 * \mathbf{X}_{Paddle1} + \mathbf{z} * w_2 * (\mathbf{X}_{Paddle2} - \mathbf{X}_{Paddle3}) + \\ & (1 - \mathbf{z}) * w_3 * (\mathbf{X} - \mathbf{X}_{Paddle1})\end{aligned} \tag{2}$$

where the value of w_1, w_2, and w_3 are updating in $[0, 2]$. The parameter $\mathbf{z}$ is decreasing exponentially and is calculated as in the following equation.

$$\mathbf{z} = 1 - \left(\frac{t}{t_{max}}\right)^2 \tag{3}$$

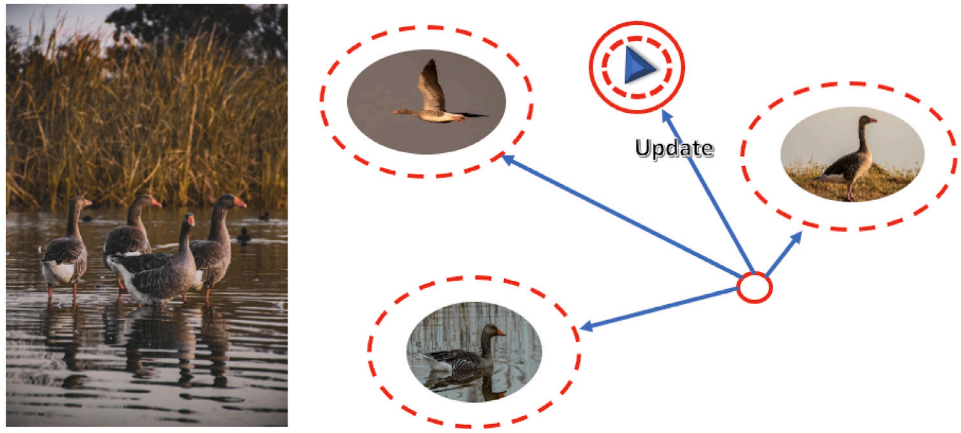

Fig. 2. Greylag Goose moving towards the prey (best solution) and searching the area around the best solution.

Fig. 3. Proposed Greylag Goose Optimization exploration, exploitation, and dynamic groups.

where iteration number is represented as t, and t_{max} represents the maximum iterations number.

The second updating process, where the $\mathbf{a}$ and $\mathbf{A}$ vector values are decreased, is as follows for $r_3 \geq 0.5$.

$$\mathbf{X}(t+1) = w_4 * |\mathbf{X}^*(t) - \mathbf{X}(t)|.e^{bl}.cos(2\pi l) + [2w_1(r_4 + r_5)] * \mathbf{X}^*(t), \tag{4}$$

where b parameter is a constant, l is a random value in $[-1, 1]$. The w_4 parameter is updating in $[0, 2]$, while r_4 and r_5 are updating in $[0, 1]$.

3.2.2. Exploitation operation

The exploitation team is in charge of improving the solutions that are already in place. The GGO identifies who has the highest fitness at the conclusion of each cycle and awards them accordingly. The GGO employs two different strategies, which are detailed below, to achieve its exploitation objective.

Moving towards the best solution: The following equation is used to proceed in the direction of the best solution.

The three solutions (sentries), $\mathbf{X}_{Sentry1}$, $\mathbf{X}_{Sentry2}$, and $\mathbf{X}_{Sentry3}$, guide other individuals ($\mathbf{X}_{NonSentry}$) to change their positions towards the estimated position of the prey. The following equations show the process of position updating.

$$\begin{aligned}
\mathbf{X}_1 &= \mathbf{X}_{Sentry1} - \mathbf{A}_1.|\mathbf{C}_1.\mathbf{X}_{Sentry1} - \mathbf{X}|, \\
\mathbf{X}_2 &= \mathbf{X}_{Sentry2} - \mathbf{A}_2.|\mathbf{C}_2.\mathbf{X}_{Sentry2} - \mathbf{X}|, \\
\mathbf{X}_3 &= \mathbf{X}_{Sentry3} - \mathbf{A}_3.|\mathbf{C}_1.\mathbf{X}_{Sentry3} - \mathbf{X}|,
\end{aligned} \tag{5}$$

where $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$ are calculated as $\mathbf{A} = 2\mathbf{a}.r_1 - \mathbf{a}$ and $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$ are calculated as $\mathbf{C} = 2r_2$. The updated positions for the population, $\mathbf{X}(t+1)$, can be expressed as an average of the three solutions of $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$ as follows

$$\mathbf{X}(t+1) = \overline{\mathbf{X}_i}\Big|_0^3 \tag{6}$$

Searching the area around the best solution The most promising choice is located close to the best response (leader) while flying. This prompts some individuals to search for enhancements by investigating regions close to the ideal response, named $\mathbf{X}_{Flock1}$. The GGO performs the aforementioned process using the following equation.

$$\mathbf{X}(t+1) = \mathbf{X}(t) + \mathbf{D}(1+\mathbf{z}) * w * (\mathbf{X} - \mathbf{X}_{Flock1}) \tag{7}$$

3.2.3. Selection of the best solution

By utilizing a mutation technique and scanning members of the exploration group, the GGO offers exceptional exploration capabilities. The GGO can postpone convergence thanks to its strong exploring capabilities. Algorithm 1 contains the GGO pseudo-code, which is viewable. We start by providing the GGO with data, including the population size, mutation rate, and iterations. The participants are subsequently split into two groups by the GGO: those who perform exploratory work and those who perform exploitative labor. The GGO method dynamically modifies the size of each group throughout the iterative process of discovering the best answer. Each team employs two techniques to do its work. To provide variation and in-depth research, the GGO randomly rearranges the answers between iterations. In one iteration, a solution component from the exploration group might migrate to the exploitation group in the following. The GGO's elitist strategy makes sure that the leader is kept in position throughout the procedure.

The steps of the GGO algorithm, as shown in Fig. 3, are applied for updating positions of the exploration group (n_1) and the exploitation group (n_2). The parameter r_1 is updated during iterations as $r_1 = c\left(1 - \frac{t}{t_{max}}\right)$, where t is current iteration, c is a constant, and t_{max} is the number of iterations. At the end of each iteration, GGO updates the agents in the search space, and the agents' order is randomly changed to exchange the agents' roles in the exploration and exploitation groups. At the final step, GGO returns the best solution.

Algorithm 1 :Proposed GGO Algorithm

Initialize GGO population $\mathbf{X}_i (i = 1, 2, \ldots, n)$, size n, iterations t_{max}, objective function F_n.
Initialize GGO parameters $\mathbf{a}$, $\mathbf{A}$, $\mathbf{C}$, b, l, c, r_1, r_2, r_3, r_4, r_5, w, w_1, w_2, w_3, w_4, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$, $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$, $t = 1$
Calculate objective function F_n for each agents $\mathbf{X}_i$
Set $\mathbf{P}$ = best agent position
Update Solutions in exploration group (n_1) and exploitation group (n_2)
while $t \le t_{max}$ **do**
  for ($i = 1 : i < n_1 + 1$) **do**
    if ($t\%2 == 0$) **then**
      if ($r_3 < 0.5$) **then**
        if ($|A| < 1$) **then**
          Update position of current search agent as $\mathbf{X}(t+1) = \mathbf{X}^*(t) - \mathbf{A}.|\mathbf{C}.\mathbf{X}^*(t) - \mathbf{X}(t)|$
        else
          Select three random search agents $\mathbf{X}_{Paddle1}$, $\mathbf{X}_{Paddle2}$, and $\mathbf{X}_{Paddle3}$
          Update ($\mathbf{z}$) by the exponential form of $\mathbf{z} = 1 - \left(\frac{t}{t_{max}}\right)^2$
          Update position of current search agent as
          $\mathbf{X}(t+1) = w_1 * \mathbf{X}_{Paddle1} + \mathbf{z} * w_2 * (\mathbf{X}_{Paddle2} - \mathbf{X}_{Paddle3}) + (1 - \mathbf{z}) * w_3 * (\mathbf{X} - \mathbf{X}_{Paddle1})$
        end if
      else
        Update position of current search agent as
        $\mathbf{X}(t+1) = w_4 * |\mathbf{X}^*(t) - \mathbf{X}(t)|.e^{bl}.cos(2\pi l) + [2w_1(r_4 + r_5)] * \mathbf{X}^*(t)$
      end if
    else
      Update individual positions as
      $\mathbf{X}(t+1) = \mathbf{X}(t) + \mathbf{D}(1 + \mathbf{z}) * w * (\mathbf{X} - \mathbf{X}_{Flock1})$
    end if
  end for
  for ($i = 1 : i < n_2 + 1$) **do**
    if ($t\%2 == 0$) **then**
      Calculate $\mathbf{X}_1 = \mathbf{X}_{Sentry1} - \mathbf{A}_1.|\mathbf{C}_1.\mathbf{X}_{Sentry1} - \mathbf{X}|$, $\mathbf{X}_2 = \mathbf{X}_{Sentry2} - \mathbf{A}_2.|\mathbf{C}_2.\mathbf{X}_{Sentry2} - \mathbf{X}|$,
      $\mathbf{X}_3 = \mathbf{X}_{Sentry3} - \mathbf{A}_3.|\mathbf{C}_3.\mathbf{X}_{Sentry3} - \mathbf{X}|$
      Update individual positions as $\mathbf{X}(t+1) = \overline{\mathbf{X}_i}\Big|_0^3$
    else
      Update position of current search agent as
      $\mathbf{X}(t+1) = \mathbf{X}(t) + \mathbf{D}(1 + \mathbf{z}) * w * (\mathbf{X} - \mathbf{X}_{Flock1})$
    end if
  end for
  Calculate objective function F_n for each $\mathbf{X}_i$
  Update parameters
  Set $t = t + 1$
  Adjust beyond the search space solutions
  if (Best F_n is same as previous two iterations) **then**
    Increase solutions of exploration group (n_1)
    Decrease solutions of exploitation group (n_2)
  end if
end while
Return best agent $\mathbf{P}$

3.2.4. Complexity analysis

The GGO algorithm's complexity analysis is presented in this section based on Algorithm 1. Using population number indicated as n iterations number as t_{max}, the complexity can be defined for each part of the algorithm as

- Initializing of GGO population: O (1).
- Initializing of GGO parameters $\mathbf{a}$, $\mathbf{A}$, $\mathbf{C}$, b, l, c, r_1, r_2, r_3, r_4, r_5, $\mathbf{w}$, w_1, w_2, w_3, w_4, $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$, $\mathbf{C}_1$, $\mathbf{C}_2$, $\mathbf{C}_3$, $t = 1$: O (1).
- Evaluating objective function F_n for each agents $\mathbf{X}_i$: O (n).
- Getting best individual $\mathbf{P}$: O (n).
- Update solutions in exploration group (n_1) and exploitation group (n_2): O (n).
- Update position of search agent in exploration group: O ($t_{max} \times n$).
- Select three random search agents $\mathbf{X}_{Paddle1}$, $\mathbf{X}_{Paddle2}$, and $\mathbf{X}_{Paddle3}$: O ($t_{max} \times n$).
- Update $\mathbf{z}$ by an exponential form: O ($t_{max} \times n$).
- Update position of search agent based on $\mathbf{z}$ and random agents: O ($t_{max} \times n$).
- Update position of search agent in exploration group if $r_3 \ge 0.5$: O ($t_{max} \times n$).
- Update position of search agent in exploration group if $t\%2 \ne 0$: O ($t_{max} \times n$).
- Calculate $\mathbf{D}_{Sentry1}$, $\mathbf{D}_{Sentry2}$, and $\mathbf{D}_{Sentry3}$: O ($t_{max} \times n$).
- Calculate $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$: O ($t_{max} \times n$).
- Update position of search agent in exploitation group: O ($t_{max} \times n$).
- Update position of search agent in exploitation group if $t\%2 \ne 0$: O ($t_{max} \times n$).
- Evaluating agents' objective function: O ($t_{max} \times n$).
- Updating parameters: O (t_{max}).
- Increasing iteration counter: O (t_{max}).

Table 1
Datasets from UCI Repository.

No.	Dataset	# Attributes	# Instances	# Classes
1	Hepatitis	19	155	2
2	Ionosphere	34	351	2
3	Vertebral	6	310	2
4	Seeds	7	210	3
5	Parkinsons	23	197	2
6	Australian	14	690	2
7	Blood	5	748	2
8	Breast_Cancer	10	699	2
9	Diabetes	8	768	2
10	Lymphography	18	148	4
11	Zoo	17	101	7
12	Ring	20	7400	2
13	Titanic	3	2201	2
14	Towonorm	20	7400	2
15	Waveform	21	5000	3
16	Tic-Tac-Toe	9	949	2
17	Mofn	10	1324	2
18	HAR (Smartphones)	561	10 299	6
19	ISOLET	617	7797	26

Table 2
Binary GGO algorithm configuration.

Parameter	Value
# Agents	10
# Iterations	100
Dimension	# Features in dataset
Domain	[0, 1]
# Runs	20
Inertia factor of SC	0.1
r_1, r_2, r_3, r_4, r_5	[0, 1]
w_1, w_2, w_3, w_4	[0, 2]
α of F_n	0.99
β of F_n	0.01

Table 3
Configuration of compared algorithms with 100 iterations and 10 agents for each one.

Algorithm	Parameter (s)	Value (s)
GWO	a	2 to 0
PSO	Inertia W_{max}, W_{min}	[0.9,0.6]
	Acceleration constants C_1, C_2	[2, 2]
SFS	Maximum diffusion level	1
WOA	a	2 to 0
	r	[0, 1]
MVO	Wormhole existence probability	[0.2, 1]
SBO	Step size	0.94
	Mutation probability	0.05
	Upper and lower limit difference	0.02
FA	# fireflies	10
GA	Mutation ratio	0.1
	Crossover	0.9
	Selection mechanism	Roulette wheel
SCA	Parameters r_2, r_3, r_4,	[0, 1]

- Adjust beyond the search space solutions: $O\ (t_{max})$.
- Test if best F_n is same as previous two iterations: $O\ (t_{max})$.

The GGO algorithm has a $O\ (t_{max} \times n)$ level of complexity. The algorithm complexity for issues with d variables can be calculated as $O\ (t_{max} \times n \times d)$.

3.3. Proposed binary GGO algorithm

Recently, feature selection has become one of the most prepossessing processes in the process of data analysis. This is because feature selection tries to reduce the high dimensional of the data by deleting features that are irrelevant or redundant. As a result, they have been used in a variety of domains finding relevant features that minimize classification errors is the primary objective of this feature selection optimization strategy. Feature selection can be expressed mathematically as a minimized optimization problem. In the event of problems with feature selection, the GGO algorithm's solutions will be strictly binary, with values of 0 or 1. To make it easier to choose features from the dataset, the continuous values of the proposed GGO method will be converted to binary values [0, 1] as illustrated in the steps of Algorithm 2. The following equation, which is based on the $Sigmoid$ function, is used in this study (El-kenawy et al., 2022).

Algorithm 2 Proposed Binary GGO Algorithm.

Initialize GGO population, objective function, and GGO parameters
Convert solution to binary [0 or 1]
Calculate objective function for each agent and get the best agent position
Update Solutions in exploration group and exploitation group
while $t \le t_{max}$ **do**
  for $(i = 1 : i < n_1 + 1)$ **do**
    if $(t\%2 == 0)$ **then**
      if $(r_3 < 0.5)$ **then**
        if $(|A| < 1)$ **then**
          Update position of current search agent in exploration group
        else
          Update position of current search agent based on three random search agents
        end if
      else
        Update position of current search agent
      end if
    else
      Update individual positions
    end if
  end for
  for $(i = 1 : i < n_2 + 1)$ **do**
    if $(t\%2 == 0)$ **then**
      Update position of current search agent in exploitation group
    else
      Update position of current search agent
    end if
  end for
  Convert updated solution to binary
  Calculate objective function
  Update parameters
  Adjust beyond the search space solutions
  Update Solutions in exploration group and exploitation group
end while
Return best agent

$$
x_d^{t+1} = \begin{cases} 1 & \text{if } Sigmoid(m) \ge 0.5 \\ 0 & otherwise, \end{cases} \qquad (8)
$$
$$
Sigmoid(m) = \frac{1}{1 + e^{-10(m-0.5)}},
$$

where the binary solution at iteration t and dimension d is indicated by x_d^{t+1}. The output solutions can be scaled by the $Sigmoid$ function to be binary ones. The value will change to 1 if $Sigmoid(m) \ge 0.5$, otherwise it will remain at 0. The algorithm's chosen features are reflected in the m parameter. In Algorithm 2, the binary GGO algorithm is thoroughly explained. The computation complexity is determined by studying the GGO algorithm to be $O\ (t_{max} \times n)$ and will be $O\ (t_{max} \times n \times d)$ for the d dimension.

In the binary GGO algorithm, a solution's quality is assessed using the objective equation F_n. The following equation uses F_n to represent a classifier's error rate, Err, a set of chosen features, s, and a set of missing features, S.

$$F_n = \alpha Err + \beta \frac{|s|}{|S|} \tag{9}$$

where $\beta = 1 - \alpha$ and $\alpha \in [0, 1]$ denote the population significance of the provided trait. K-nearest neighbor (k-NN) is a popular, simple classification strategy. The approach is successful if it can give a subset of features that can result in a low classification error rate. This method makes use of the k-NN as a classifier to guarantee the validity of the features selected. The shortest distance between the query instance and the training instances is the sole criterion for classifier selection; this experiment does not make use of a K-nearest neighbor model.

4. Experimental results

The section on the experimental outcomes is broken up into two different scenarios. The first scenario is intended to evaluate the efficacy of the GGO algorithm in solving feature selection problems by using nineteen distinct datasets from the public machine learning repository at the University of California, Irvine, as test subjects. The following scenario tests the capability of the provided algorithm to solve twenty-three benchmark functions, which are separated into unimodal and multimodal function categories.

4.1. First scenario: Feature selection

In this work, a total of nineteen datasets were taken from the UCI repository and used to test and evaluate the effectiveness of the proposed approach for feature selection issues. The nineteen datasets that are presented in Table 1 have been constructed with a varied amount of features/attributes, instances, and classes so that the methods can be tested on their performance while dealing with a variety of issues. The presented algorithm of binary GGO (bGGO) is evaluated alongside the original version, in this case, including bGWO (Al-Tashi, Abdul Kadir, Rais, Mirjalili, & Alhussian, 2019), bPSO (Bello, Gomez, Nowe, & Garcia, 2007), bSFS (Salimi, 2015), bWOA (Mirjalili & Lewis, 2016), bMVO (Mirjalili, Mirjalili, & Hatamlou, 2016), bSBO (Samareh Moosavi & Khatibi Bardsiri, 2017), bFA (Fister, Yang, Fister, & Brest, 2012), bGA (Kabir et al., 2011) algorithms, Modified GWO (bMGWO) (El-Kenawy, Eid, Saber, & Ibrahim, 2020), hybrid of PSO and GWO (bGWO-PSO) (Şenel, Gokçe, Yuksel, & Yigit, 2019), hybrid of GA and GWO (bGWO-GA) (El-Kenawy et al., 2020), and hybrid of SCA and PSO (bSCA-PSO) (Fouad, El-Desouky, Al-Hajj, & El-Kenawy, 2020) in which b denotes the binary variant of the algorithm. Tables 2 and 3 contain discussion on the configuration of the presented GGO and comparative algorithms while they were running the experiments. Each algorithm started with 100 iterations, and 10 agents were activated at the beginning of each iteration.

4.1.1. First scenario: Performance metrics

Experiments use the following metrics in order to evaluate the proposed algorithm's ability to choose features for analysis. If M is the number of repeats, g_* is the optimal solution, and N is the total number of points, then the ideal solution is g_*. The following equation can be used to determine the Average Error for the value L representing a class for a point, C representing the output of the classifier for that point, and $Match$ representing the degree to which the two inputs are matched.

$$\text{AvgError} = 1 - \frac{1}{M} \sum_{j=1}^{M} \frac{1}{N} \sum_{i=1}^{N} Match(C_i, L_i) \tag{10}$$

The Average Fitness can be computed, for $size(g_j^*)$ as the vector g_j^* size, and D represents the size of the dataset, as follows.

$$\text{AvgSelectSize} = \frac{1}{M} \sum_{j=1}^{M} \frac{size(g_j^*)}{D} \tag{11}$$

The Best Fitness and the Worst Fitness are computed as in the following equations.

$$\text{Best Fitness} = Min_{j=1}^{M} g_j^* \tag{12}$$

$$\text{Worst Fitness} = Max_{j=1}^{M} g_j^* \tag{13}$$

The Mean and the Standard Deviation (SD) are represented in the following equations.

$$\text{SD} = \sqrt{\frac{1}{M-1} \sum (g_j^* - Mean)^2} \tag{14}$$

$$\text{Mean} = \frac{1}{M} \sum_{j=1}^{M} g_j^* \tag{15}$$

4.1.2. First scenario: Results and discussion

In Tables 4–10, the comparative findings between the GGO and other meta-heuristic approaches are presented. In Tables 4–6, respectively, the average error, average selected size, and average fitness of categorization accuracy are provided. These tables show that the GGO has excellent accuracy in most datasets. According to Table 7, the average of best fitness for the GGO is second to none. The average of worst fitness for GGO in Table 8 is the lowest. This table shows that the performance of GGO is better than other methods. Additionally, it can be seen that the GGO algorithm, which has the lowest standard deviation among all datasets, is more stable than others by analyzing the stability of the algorithms as shown in Table 9.

Table 4 displays the results of the provided and contrasted methods based on the average error for each of the nineteen datasets. The results that are based on selected sizes on average are shown in Table 5. The results of the evaluation based on the participants' average fitness, their greatest fitness, and their worst fitness are presented in Tables 6, 7, and 8, respectively. Table 9 displays the standard deviation fitness results obtained from the various algorithms that were examined. Table 11 displays the p-values of the proposed and other tested algorithms for the nineteen datasets. This reflects the performance of the suggested algorithm, which had a p-value that was lower than 0.005 for each and every dataset. The performance of the binary GGO algorithm for solving the problem of feature selection is validated by the results that are provided in the tables ranging from Table 4 to Table 9.

From Tables 10 and 11, according to Wilcoxon's rank sum test and p-value, it is clear that the GGO is significantly superior to other methods. These results suggest that GGO will behave in a way that finds a balance between exploration and exploitation, ultimately improving both the convergence rate and the output quality.

4.2. Second scenario: Benchmark functions

This scenario puts the proposed algorithm through its paces by testing its ability to find the optimal solution for the benchmark functions. In this subsection, a total of twenty-three functions are utilized. These functions are comprised of seven unimodal, six multimodal, and ten multimodal-based fixed-dimension benchmark functions. Three tables of benchmark functions are presented. Table 12 shows the description of the unimodal benchmark functions. Table 13 describes the multimodal benchmark functions. Table 14 describes the multimodal-based fixed-dimension benchmark functions. In this scenario, the GGO algorithm is compared to the original GWO (Al-Tashi et al., 2019), PSO (Bello et al., 2007), WOA (Mirjalili & Lewis, 2016), Feedforward Error Propagation (FEP) algorithm (Mendil & Benmahammed, 1999), Gravitational Search Algorithm (GSA) (Rashedi, Nezamabadi-pour, & Saryazdi, 2009), and GA (Kabir et al., 2011) algorithms.

Table 4
Presented bGGO and compared algorithms' average error.

Dataset	bGGO	bGWO	bGWO_PSO	bPSO	bSFS	bWOA	bMGWO	bMVO	bSBO	bGWO_GA	bFA	bGA
Hepatitis	0.1823	0.2308	0.1998	0.2345	0.2108	0.1986	0.2100	0.1997	0.2436	0.2100	0.2049	0.1986
Ionosphere	0.1213	0.1565	0.1500	0.1812	0.1615	0.1593	0.1399	0.1677	0.1626	0.1779	0.1664	0.1671
Vertebral	0.1998	0.2164	0.2225	0.2237	0.2243	0.2177	0.2321	0.2155	0.2366	0.2229	0.2372	0.2080
Seeds	0.2405	0.2988	0.2445	0.2819	0.2601	0.2881	0.2588	0.2897	0.2600	0.2613	0.2987	0.2815
Parkinsons	0.1282	0.1522	0.1628	0.1567	0.1555	0.1467	0.1528	0.1590	0.1611	0.1677	0.1513	0.1444
Australian	0.1492	0.1677	0.1588	0.1630	0.1636	0.1589	0.1885	0.1757	0.1983	0.1579	0.1676	0.1587
Blood	0.2296	0.2512	0.2548	0.2599	0.2678	0.2466	0.2514	0.2488	0.2519	0.2396	0.2612	0.2588
Breast_Cancer	0.0386	0.0475	0.0488	0.0433	0.0489	0.0413	0.0497	0.0490	0.0474	0.0469	0.0474	0.0461
Diabetes	0.2688	0.2700	0.2502	0.2714	0.3444	0.2991	0.2747	0.2789	0.2565	0.2602	0.2790	0.2899
Lymphography	3.3356	3.4996	3.3774	3.4909	3.4491	3.5890	3.3186	3.7782	3.5664	3.4338	3.5808	3.0891
Zoo	0.1214	0.1420	0.1378	0.1452	0.1613	0.1677	0.1500	0.1535	0.1607	0.1882	0.1503	0.1544
Ring	0.1482	0.1618	0.1644	0.1690	0.1679	0.1663	0.1632	0.1730	0.1678	0.1672	0.1622	0.1658
Titanic	0.2312	0.2411	0.2399	0.2334	0.2260	0.2309	0.2311	0.2236	0.2110	0.2231	0.2400	0.2271
Towonorm	0.0301	0.0489	0.0433	0.0617	0.0634	0.0356	0.0355	0.0473	0.0517	0.0700	0.0578	0.0689
WaveformEW	0.3711	0.4332	0.4199	0.4354	0.3998	0.3978	0.4406	0.4230	0.4134	0.4375	0.4244	0.4468
Tic-Tac-Toe	0.2567	0.2733	0.2766	0.2689	0.2690	0.2496	0.2678	0.2611	0.3170	0.2777	0.2778	0.2810
Mofn	0.0577	0.1213	0.1388	0.1333	0.1366	0.1299	0.0987	0.1290	0.1104	0.1412	0.1435	0.1424
HAR Using Smartphones	0.3791	0.9667	0.8914	0.8331	0.9668	1.6337	0.6449	1.8835	1.8660	0.7882	1.4090	0.8668
ISOLET	0.5118	0.8992	0.7710	0.9512	0.9800	0.9787	0.7660	0.9713	0.9819	0.8365	0.9744	0.6833

Table 5
Average select size of the presented bGGO and compared algorithms.

Dataset	bGGO	bGWO	bGWO_PSO	bPSO	bSFS	bWOA	bMGWO	bMVO	bSBO	bGWO_GA	bFA	bGA
Hepatitis	0.3511	0.4066	0.4016	0.5371	0.3613	0.6325	0.4530	0.5070	0.5221	0.4819	0.5422	0.5020
Ionosphere	0.1356	0.2677	0.2921	0.4929	0.3153	0.3331	0.2551	0.4731	0.4320	0.4259	0.4898	0.3925
Vertebral	0.4101	0.5020	0.5020	0.5104	0.5051	0.5104	0.4907	0.5187	0.4347	0.7028	0.5104	0.5104
Seeds	0.5286	0.5071	0.5442	0.7028	0.5161	0.6885	0.5634	0.5307	0.6024	0.5161	0.5881	0.5522
Parkinsons	0.2442	0.4130	0.3925	0.4701	0.2930	0.4860	0.3102	0.5385	0.4920	0.5020	0.4723	0.4495
Australian	0.2631	0.4299	0.4585	0.5337	0.4872	0.6806	0.3725	0.5158	0.5731	0.5302	0.5266	0.5087
Blood	0.6115	0.7028	0.6024	0.6526	0.6131	0.7781	0.5458	0.7405	0.6526	0.6526	0.7656	0.7781
Breast_Cancer	0.4697	0.5261	0.5010	0.5949	0.5261	0.6388	0.5224	0.5949	0.6513	0.6513	0.6388	0.5824
Diabetes	0.3213	0.5146	0.5271	0.6087	0.4770	0.6401	0.4267	0.5899	0.5271	0.5271	0.5773	0.5648
Lymphography	0.2423	0.3761	0.2786	0.4848	0.5349	0.5099	0.5227	0.4792	0.5349	0.5349	0.4597	0.4319
Zoo	0.2498	0.3628	0.3833	0.4952	0.4564	0.5020	0.3925	0.4769	0.4929	0.4837	0.4929	0.4609
Ring	0.3110	0.3203	0.3403	0.3604	0.3324	0.3253	0.3214	0.3353	0.3403	0.3203	0.3554	0.3353
Titanic	0.8000	0.8000	0.8701	0.8199	0.8601	0.8534	0.8345	0.8869	0.8173	0.8000	0.8869	0.8534
Towonorm	0.6211	0.8542	0.8517	0.6914	0.8116	0.9795	0.7315	0.8517	0.7916	0.6713	0.7490	0.8717
WaveformEW	0.4332	0.5294	0.5151	0.5818	0.5203	0.8966	0.6009	0.6414	0.6009	0.6867	0.6081	0.6534
Tic-Tac-Toe	0.4581	0.5346	0.4757	0.6126	0.4688	0.7518	0.4745	0.6126	0.4714	0.5911	0.6348	0.6126
Mofn	0.1767	0.6122	0.2017	0.6786	0.2432	0.8775	0.2113	0.6582	0.4163	0.4412	0.6888	0.6990
HAR Using Smartphones	0.5431	0.8876	0.7779	0.8825	0.9044	0.9067	0.7012	0.9197	0.9584	0.7146	0.9595	0.7763
ISOLET	0.6687	0.8818	0.7827	0.8185	1.0006	0.9704	0.7755	0.9407	0.9867	0.7907	0.9382	0.7995

Table 6
Average fitness of the presented bGGO and compared algorithms.

Dataset	bGGO	bGWO	bGWO_PSO	bPSO	bSFS	bWOA	bMGWO	bMVO	bSBO	bGWO_GA	bFA	bGA
Hepatitis	0.2118	0.2617	0.2440	0.2481	0.2620	0.2491	0.2274	0.2491	0.2560	0.2199	0.2559	0.2491
Ionosphere	0.1375	0.1821	0.1486	0.2098	0.1549	0.1949	0.1788	0.2038	0.1556	0.1948	0.2004	0.1898
Vertebral	0.2113	0.3921	0.2179	0.3974	0.2453	0.3892	0.2239	0.4051	0.3394	0.3263	0.4037	0.3931
Seeds	0.3553	0.5071	0.4558	0.3794	0.4528	0.3908	0.3893	0.3780	0.4297	0.4498	0.3886	0.3787
Parkinsons	0.1504	0.1635	0.1536	0.1803	0.1640	0.1711	0.1720	0.1826	0.1898	0.1637	0.1703	0.1757
Australian	0.3096	0.3288	0.1535	0.3206	0.1410	0.3230	0.1823	0.3243	0.1648	0.1579	0.3312	0.3189
Blood	0.8448	0.8752	0.2556	0.8824	0.2687	0.8756	0.2504	0.8746	0.2500	0.2395	0.8672	0.8812
Breast_Cancer	0.3011	0.3399	0.3505	0.3361	0.3473	0.3350	0.3475	0.3357	0.5483	0.3496	0.3378	0.3382
Diabetes	0.5433	0.5880	0.5886	0.5952	0.5901	0.5946	0.5881	0.5907	0.5958	0.5883	0.5940	0.6026
Lymphography	3.0116	3.4323	4.4009	3.5002	4.6916	3.6005	3.5003	3.7637	4.7530	5.2647	3.4941	3.1963
Zoo	0.1441	0.1749	0.1792	0.1757	0.1846	0.1956	0.1668	0.1742	0.1776	0.1699	0.1864	0.1811
Ring	1.3222	1.3924	0.1637	1.3980	0.1677	1.3966	0.1404	1.3938	0.1659	0.1675	1.3963	1.3984
Titanic	2.5467	2.6898	2.7282	2.6870	2.8013	2.6845	2.7301	2.6795	2.9591	0.8000	2.6866	2.6811
Towonorm	1.1338	1.2848	1.4457	1.3018	1.6133	1.2701	1.3510	1.2816	1.5555	1.6729	1.2924	1.3044
WaveformEW	1.1665	1.1948	1.4307	1.2176	0.4359	1.1876	1.3256	1.2074	1.4169	0.4371	1.2145	1.2341
Tic-Tac-Toe	0.6122	0.6268	0.7919	0.6238	0.2684	0.6109	0.6453	0.6145	0.6218	0.6913	0.6299	0.6274
Mofn	0.4946	0.5685	0.5285	0.5853	0.5471	0.5786	0.5320	0.5703	0.5912	0.5433	0.5877	0.5941
HAR Using Smartphones	0.5338	0.8597	0.7919	0.8586	0.9144	0.9167	0.7681	0.9381	0.9779	0.7922	0.9258	0.7790
ISOLET	0.5942	0.8489	0.7798	0.8860	0.9371	0.9507	0.7807	0.9362	0.9909	0.8044	0.9197	0.8055

Table 7
Presented bGGO and compared algorithms' best fitness.

Dataset	bGGO	bGWO	bGWO_PSO	bPSO	bSFS	bWOA	bMGWO	bMVO	bSBO	bGWO_GA	bFA	bGA
Hepatitis	0.1190	0.1497	0.2276	0.1497	0.2111	0.1886	0.1780	0.2081	0.2081	0.2081	0.1886	0.1497
Ionosphere	0.0811	0.1121	0.1375	0.1460	0.1018	0.1206	0.1453	0.1036	0.1290	0.1800	0.0951	0.1290
Vertebral	0.2998	0.3284	0.3767	0.3188	0.3670	0.3188	0.3176	0.3381	0.3767	0.3477	0.3381	0.3188
Seeds	0.1000	0.5071	0.2424	0.1572	0.2364	0.1572	0.1674	0.2282	0.2282	0.2708	0.2424	0.1856
Parkinsons	0.0570	0.0755	0.1061	0.1214	0.0755	0.0908	0.1673	0.0908	0.0908	0.1520	0.0755	0.0908
Australian	0.2442	0.2900	0.2986	0.2857	0.3019	0.2813	0.2943	0.2986	0.3072	0.3072	0.2943	0.2900
Blood	0.8228	0.8351	0.8391	0.8431	0.8711	0.8351	0.8601	0.8431	0.8411	0.8311	0.8311	0.8311
Breast_Cancer	0.3032	0.3131	0.3301	0.3174	0.3179	0.3089	0.3156	0.3131	0.3259	0.3301	0.3046	0.3174
Diabetes	0.5118	0.5504	0.5659	0.5581	0.5581	0.5387	0.5564	0.5581	0.5504	0.5542	0.5465	0.5465
Lymphography	1.9554	1.9125	2.8649	1.9125	3.1314	1.4869	1.4261	1.6693	3.8376	3.8984	1.9733	1.7504
Zoo	0.0875	0.0908	0.1214	0.0908	0.1520	0.1061	0.1347	0.0908	0.1061	0.1520	0.0908	0.1061
Ring	1.2279	1.3718	1.3862	1.3854	1.3931	1.3834	1.3862	1.3818	1.3858	1.3923	1.3718	1.3870
Titanic	2.6439	2.6439	2.6463	2.6544	2.6626	2.6544	2.6471	2.6476	2.6563	0.8000	2.6544	2.6544
Towonorm	1.1612	1.2734	1.2698	1.2758	1.2875	1.2641	1.2142	1.2665	1.2830	1.2838	1.2810	1.2862
WaveformEW	1.1017	1.1562	1.2033	1.1503	1.1736	1.1295	1.2097	1.1693	1.1658	1.1908	1.1426	1.1753
Tic-Tac-Toe	0.5447	0.5648	0.5835	0.5679	0.5911	0.5804	0.5901	0.5648	0.5866	0.5835	0.5742	0.5773
Mofn	0.4320	0.4945	0.4877	0.5083	0.5270	0.5335	0.4775	0.4945	0.4716	0.5747	0.5495	0.5243
HAR Using Smartphones	0.4970	0.8361	0.7650	0.8172	0.9043	0.9136	0.7091	0.8838	0.9039	0.7080	0.9147	0.7667
ISOLET	0.5987	0.8088	0.7149	0.8502	0.8944	0.9172	0.7374	0.9176	0.8039	0.7821	0.8941	0.7938

Table 8
Presented bGGO and compared algorithms' worst fitness.

Dataset	bGGO	bGWO	bGWO_PSO	bPSO	bSFS	bWOA	bMGWO	bMVO	bSBO	bGWO_GA	bFA	bGA
Hepatitis	0.3112	0.3446	0.3056	0.3640	0.2861	0.3640	0.3445	0.3251	0.3251	0.3251	0.3835	0.3251
Ionosphere	0.2466	0.2565	0.2055	0.2650	0.2888	0.2820	0.2650	0.2480	0.2310	0.2905	0.2990	0.3074
Vertebral	0.4577	0.4346	0.3960	0.4635	0.4253	0.4635	0.4732	0.5021	0.5021	0.4442	0.6083	0.5021
Seeds	0.5169	0.5071	0.4412	0.6116	0.4412	0.6116	0.4211	0.5406	0.4412	0.4412	0.5548	0.6258
Parkinsons	0.2532	0.2285	0.2437	0.2743	0.2590	0.2590	0.2590	0.2590	0.2437	0.2743	0.2438	0.2438
Australian	0.3399	0.3979	0.3418	0.3806	0.3202	0.3634	0.4368	0.3461	0.3375	0.3375	0.4972	0.3547
Blood	0.9441	0.9389	0.9229	0.9389	0.9668	0.9189	0.9229	0.9389	0.9229	0.9229	0.8989	0.9389
Breast_Cancer	0.3551	0.3770	0.3600	0.3557	0.3600	0.3514	0.3685	0.3557	0.3557	0.3472	0.3642	0.3685
Diabetes	0.6279	0.6591	0.5814	0.6241	0.6008	0.6435	0.6280	0.6474	0.5892	0.5969	0.6358	0.7095
Lymphography	5.2335	5.2562	5.0535	5.9452	6.4598	5.4791	5.6443	5.4993	6.6747	7.3232	5.1143	5.1346
Zoo	0.2278	0.2743	0.2285	0.2896	0.2437	0.3202	0.2691	0.2438	0.2590	0.2796	0.2438	0.2896
Ring	1.3917	1.4051	1.4120	1.4164	1.4172	1.4124	1.3893	1.4095	1.4144	1.4071	1.4176	1.4136
Titanic	2.5016	3.1085	2.6883	2.7439	2.7724	2.7914	2.6625	2.7290	2.6978	0.8000	2.8728	2.7426
Towonorm	1.0857	1.3011	1.2854	1.3430	1.3068	1.2903	1.2983	1.2983	1.2951	1.3156	1.3064	1.3430
WaveformEW	1.1776	1.2925	1.2521	1.2705	1.2788	1.2128	1.2467	1.2574	1.2616	1.2467	1.2788	1.2937
Tic-Tac-Toe	0.6410	0.6955	0.6862	0.7515	0.6613	0.6613	0.6893	0.6831	0.7079	0.6924	0.6831	0.6706
Mofn	0.5602	0.6411	0.5839	0.6526	0.6661	0.6182	0.6130	0.6411	0.5953	0.5930	0.6137	0.6411
HAR Using Smartphones	0.5686	0.8695	0.8709	0.8807	0.9348	0.9912	0.7909	0.9488	0.8486	0.8032	0.9357	0.8151
ISOLET	0.6879	0.8657	0.8177	0.8947	0.9518	0.9697	0.8190	0.9596	1.1301	0.8468	0.9422	0.8378

Table 9
Standard deviation fitness of the presented bGGO and compared algorithms.

Dataset	bGGO	bGWO	bGWO_PSO	bPSO	bSFS	bWOA	bMGWO	bMVO	bSBO	bGWO_GA	bFA	bGA
Hepatitis	0.0312	0.0521	0.0356	0.0598	0.0521	0.0464	0.0426	0.0399	0.0477	0.0444	0.0508	0.0505
Ionosphere	0.0311	0.0427	0.0385	0.0335	0.0749	0.0425	0.0379	0.0398	0.0377	0.0467	0.0482	0.0477
Vertebral	0.0245	0.0289	0.0287	0.0401	0.0487	0.0378	0.0377	0.0405	0.0513	0.0357	0.0643	0.0451
Seeds	0.0766	0.5071	0.0818	0.1042	0.0895	0.1076	0.0955	0.0898	0.0998	0.0862	0.0834	0.1061
Parkinsons	0.0307	0.0415	0.0519	0.0430	0.0768	0.0527	0.0356	0.0530	0.0578	0.0479	0.0445	0.0376
Australian	0.0151	0.0321	0.0189	0.0220	0.0269	0.0237	0.0342	0.0154	0.0168	0.0194	0.0457	0.0179
Blood	0.0196	0.0252	0.0339	0.0286	0.0448	0.0248	0.0227	0.0251	0.0404	0.0362	0.0209	0.0315
Breast_Cancer	0.0061	0.0142	0.0126	0.0093	0.0332	0.0111	0.0107	0.0115	0.0117	0.0065	0.0132	0.0133
Diabetes	0.0151	0.0303	0.0176	0.0170	0.0370	0.0225	0.0317	0.0268	0.0165	0.0191	0.0256	0.0352
Lymphography	0.8368	0.9041	0.8831	1.1398	1.0976	0.8498	1.7677	0.9750	1.1356	1.3710	0.9041	0.9408
Zoo	0.0387	0.0456	0.0446	0.0497	0.0591	0.0560	0.0448	0.0467	0.0634	0.0562	0.0456	0.0478
Ring	0.0057	0.0084	0.0099	0.0078	0.0102	0.0094	0.0067	0.0078	0.0108	0.0059	0.0117	0.0063
Titanic	0.0143	0.0997	0.0154	0.0271	0.0448	0.0322	0.0450	0.0194	0.0187	0.8000	0.0463	0.0216
Towonorm	0.0054	0.0087	0.0065	0.0163	0.0118	0.0067	0.0107	0.0083	0.0084	0.0126	0.0076	0.0147
WaveformEW	0.0216	0.0317	0.0206	0.0294	0.0419	0.0255	0.0314	0.0314	0.0426	0.0230	0.0335	0.0337
Tic-Tac-Toe	0.0196	0.0346	0.0397	0.0237	0.0447	0.0295	0.0222	0.0205	0.0435	0.0420	0.0297	0.0300
Mofn	0.0278	0.0391	0.0417	0.0472	0.0318	0.0342	0.0705	0.0215	0.0501	0.0070	0.0404	0.0169
HAR Using Smartphones	0.0155	0.0335	0.0300	0.0313	0.0514	0.0546	0.0223	0.0458	0.0503	0.3022	0.0447	0.0303
ISOLET	0.0210	0.0416	0.0380	0.0401	0.0604	0.0626	0.0303	0.0515	0.0526	0.3257	0.0526	0.0304

Table 10
The Wilcoxon Signed Rank test of the presented bGGO against compared algorithms.

	bGGO	bGWO	bGWO_PSO	bPSO	bSFS	bWAO	bMGWO	bMVO	bSBO	bGWO_GA	bFA	bGA
Theoretical median	0	0	0	0	0	0	0	0	0	0	0	0
Actual median	0.1824	0.2407	0.2133	0.2088	0.2162	0.1984	0.2172	0.1993	0.2233	0.2314	0.2159	0.1986
Number of values	20	20	20	20	20	20	20	20	20	20	20	20
Wilcoxon Signed Rank Test												
Sum of signed ranks (W)	210	210	210	210	210	210	210	210	210	210	210	210
Sum of positive ranks	210	210	210	210	210	210	210	210	210	210	210	210
Sum of negative ranks	0	0	0	0	0	0	0	0	0	0	0	0
P value (two-tailed)	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
Exact or estimate?	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact
P value summary	****	****	****	****	****	****	****	****	****	****	****	****
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?												
Discrepancy	0.1824	0.2407	0.2133	0.2088	0.2162	0.1984	0.2172	0.1993	0.2233	0.2314	0.2159	0.1986

Table 11
The p-values of the presented bGGO against compared algorithms for the nineteen datasets.

Dataset	bGWO	bGWO_PSO	bPSO	bSFS	bWOA	bMGWO	bMVO	bSBO	bGWO_GA	bFA	bGA
Hepatitis	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Ionosphere	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Vertebral	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0671	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Seeds	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Parkinsons	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Australian	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Blood	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0784	<0.0001	<0.0001
Breast_Cancer	<0.0001	<0.0001	<0.0001	<0.0001	0.0633	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Diabetes	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0663	<0.0001	<0.0001	<0.0001	<0.0001
Lymphography	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Zoo	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Ring	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Titanic	<0.0001	<0.0001	<0.0001	0.0579	<0.0001	<0.0001	<0.0001	<0.0001	0.0891	<0.0001	<0.0001
Towonorm	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0587	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
WaveformEW	<0.0001	<0.0001	<0.0001	0.0863	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Tic-Tac-Toe	<0.0001	<0.0001	<0.0001	<0.0001	0.0776	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Mofn	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
HAR Using Smartphones	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
ISOLET	<0.0001	0.0815	0.0761	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0711

Table 12
Description of the unimodal benchmark functions.

Function	D	Range
$f_1(w) = \sum_{i=1}^{n} w^2$	30	[−100, 100]
$f_2(w) = \sum_{i=1}^{n} \lvert w_i \rvert + \prod_{i=1}^{n} \lvert w_i \rvert$	30	[−10, 10]
$f_3(w) = \sum_{i=1}^{n} (\sum_{j=1}^{i} w_i)^2$	30	[−100, 100]
$f_4(w) = max_i\{\lvert w_i \rvert, 1 \leq i \leq D\}$	30	[−100, 100]
$f_5(w) = \sum_{i=1}^{D-1} [100(w_{i+1} - w_i^2)^2 - (w_i - 1)^2]$	30	[−30, 30]
$f_6(w) = \sum_{i=1}^{D} (w_i + 0.5)^2$	30	[−100, 100]
$f_7(w) = \sum_{i=1}^{D} i w_i^4 + rand[0, 1]$	30	[−1.28, 1.28]

4.2.1. Second scenario: Results and discussion

Table 15 shows the mean and standard deviation (StDev) of the suggested and compared algorithms over the benchmark functions from f_1 to f_{23}. This table demonstrates that the suggested GGO algorithm reached zero values in mean and StDev in certain circumstances, and in other cases, it generated results that were superior to those achieved by the compared single and hybrid methods. In Table 16, we present the average execution times (avg_time), standard deviation of execution times (std_time), and average function evaluations (avg_FEs) for both the suggested and compared algorithms across various benchmark functions. The proposed algorithm demonstrated superior performance in terms of lower execution times for the unimodal benchmark functions f_1, f_2, f_3, f_4, f_5, f_6, and f_7. Additionally, it outperformed the compared algorithms on the multimodal benchmark functions f_8, f_9, f_{10}, f_{11}, f_{12}, and f_{13}. Furthermore, in the case of multimodal-based fixed-dimension benchmark functions, the GGO algorithm exhibited reduced execution times for f_{19} and f_{23}. The convergence curve of the presented and compared algorithms for sample benchmark functions of f_1, f_2, f_3, f_6, f_9, f_{11}, f_{21} and f_{23} are shown in Fig. 4. Figs. 5, 6, and 7 show the box plot of the suggested and compared algorithms for benchmark function (f_1 to f_7), (f_8 to f_{16}), and (f_{17} to f_{23}), respectively.

4.2.2. Second scenario: ANOVA and Wilcoxon's rank-sum

The ANOVA test for benchmark functions from f_1 to f_8 are shown in Table 17. Tables 18 and 19 present the ANOVA test for benchmark functions from f_8 to f_{14} and f_{15} to f_{23}, respectively. Wilcoxon Signed Rank test for the benchmark functions samples of f_1, f_8, and f_{14} based on the suggested GGO algorithm against the compared algorithms are introduced in Tables 20, 21, and 22, respectively. The performance of the proposed continuous GGO algorithm for the benchmark functions is confirmed by the results of the algorithm when it is applied to this situation and compared to the algorithms that are considered state-of-the-art.

5. Engineering problems

In this part, the capability of the algorithm is tested to solve two constrained optimization problems involving the design of a tension/compression spring and a pressure vessel. We will validate the GGO algorithm by solving two restricted optimization examples. These examples will involve the design of tension/compression springs (Celik & Kutucu, 2018) and pressure vessels (Zou, Liu, Gao, & Li, 2011). The two engineering problems are described mathematically in this section. The GGO algorithm results are compared with GA (Kabir et al., 2011), GSA (He, Zhu, Wang, Yu, & Yao, 2019), GWO (Al-Tashi et al., 2019), and PSO (Bello et al., 2007) algorithms results to get the minimum cost.

Table 13
Description of the multimodal benchmark functions.

Function	D	Range	f_{min}
$f_{08}(w)=\sum_{i=1}^{D}-w_i\sin\left(\sqrt{\lvert w_i\rvert}\right)$	30	[−500, 500]	−12569.487
$f_{09}(w)=\sum_{i=1}^{D}[w_i^2-10\cos(2\pi w_i)+10]$	30	[−5.12, 5.12]	0
$f_{10}(w)=-20\exp(-0.2\sqrt{\sum_{i=1}^{D}w_i^2})-\exp(\frac{1}{d}\sum_{i=1}^{D}\cos(2\pi w_i))+20+\eta$	30	[−32, 32]	0
$f_{11}(w)=\frac{1}{4000}\sum_{i=1}^{D}w_i^2-\prod_{i=1}^{D}\cos(\frac{w_i}{\sqrt{i}})+1$	30	[−600, 600]	0
$f_{12}(w)=\frac{\pi}{D}\{10\sin^2(\pi y_i)+\sum_{i=1}^{D-1}(y_i-1)^2[1+10\sin^2(\pi y_i+1)+(yD-1)^2+\sum_{i=1}^{D}u(w_i,10,100,4)]\}$; $y_i=1+\frac{w_i+1}{4}$, $u(w_i,h,k,m)=\begin{cases}k(w_i-h)^m & w_i>h\\ 0 & -h<w_i<h\\ k(-w_i-h)^m & w_i<-h\end{cases}$	30	[−50, 50]	0
$f_{13}(w)=0.1\{10\sin^2(3\pi y_i)+\sum_{i=1}^{D-1}(w_i-1)^2[1+10\sin^2(3\pi y_i+1)]+(w_n-1)^2[1+\sin^2(2\pi w_n)]\}+\sum_{i=1}^{n}u(w_i,5,100,4)$	30	[−50, 50]	0

Table 14
Description of multimodal-based fixed-dimension benchmark functions.

Function	D	Range	f_{min}
$f_{14}(w)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(w_i-h_{ij})^6}\right)^{-1}$	2	[−65, 65]	1
$f_{15}(w)=\sum_{i=1}^{11}\left[h_i-\frac{w_1(b_i^2+b_iw_2)}{b_i^2+b_iw_3+w_4}\right]^2$	4	[−5, 5]	0.00030
$f_{16}(w)=4w_1^2-2.1w_1^4+\frac{1}{3}w_1^6+w_1w_2-4w_2^2+4w_2^4$	2	[−5, 5]	−1.0316
$f_{17}(w)=\left(w_2-\frac{5.1}{4\pi^2}w_1^2+\frac{5}{\pi}w_1+-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos w_1+10$	2	[−5, 5]	0.398
$f_{18}(w)=[1+(w_1+w_2+1)^2(19-14w_1+3w_1^2-14w_2+6w_1w_2+3w_2^2)]\times[30+(2w_1-3w_2)^2w(18-32w_1+12w_1^2+48w_2-36w_1w_2+27w_2^2)]$	2	[−2, 2]	3
$f_{19}(w)=-\sum_{i=1}^{4}b_i\exp(-\sum_{i=1}^{3}h_{ij}(w_j-p_{ij})^2)$	3	[1, 3]	−3.86
$f_{20}(w)=-\sum_{i=1}^{4}b_i\exp(-\sum_{i=1}^{6}h_{ij}(w_j-p_{ij})^2)$	6	[0, 1]	−3.32
$f_{21}(w)=-\sum_{i=1}^{5}\left[(w-h_i)(w-h_i)^T+b_i\right]^{-1}$	4	[0, 10]	−10.1532
$f_{22}(w)=-\sum_{i=1}^{7}\left[(w-h_i)(w-h_i)^T+b_i\right]^{-1}$	4	[0, 10]	−10.4028
$f_{23}(w)=-\sum_{i=1}^{10}\left[(w-h_i)(w-h_i)^T+b_i\right]^{-1}$	4	[0, 10]	−10.5363

5.0.1. Tension/compression spring design problem

Fig. 8 shows the schematic diagram of tension/compression spring design (TCSD) (Celik & Kutucu, 2018). TCSD is considered a continuously constrained problem. The algorithm aims to minimize the volume of a coil spring under a constant tension/compression load. The TCSD has three design variables which are the number of spring's active coils, L, the diameter of the winding, d, and the diameter of the wire, w. The mathematical formulation of the TCSD can be described as follows:

Minimize

$$f(w,d,L)=(L+2)w^2d \tag{16}$$

Subject to the following constraints

$$\begin{aligned} g_1&=1-\frac{d^3+L}{71785w^4}\le 0\\ g_2&=\frac{d(4d-w)}{w^3(12566d-w)}+\frac{1}{5108w^2}-1\le 0\\ g_3&=1-\frac{140.45w}{d^2L}\le 0\\ g_4&=\frac{2(w+d)}{3}-1\le 0\end{aligned} \tag{17}$$

where the three variables range are as follows:

$$\begin{aligned}&0.05\le w\le 2.0,\\ &0.25\le d\le 1.3,\\ &2.0\le L\le 15\end{aligned} \tag{18}$$

Table 23 compares the best parameter solutions for the tension/compression spring design problem. According to Table 24, GGO has a comparative result in terms of best results, and its solution ranked second. Statistical results show that GGO is more reliable than methods. Table 24 also displays the outcome and optimal variable values for many algorithms, including GD, PSO, GSA, and DA. The Wilcoxon test, a non-parametric statistical analysis, was employed to evaluate the results' significance. The results of this p-values test at 0.0001 are shown in Table 25, which shows how the GGO clearly outperforms other algorithms based on p-values less than 0.0001. Fig. 9 shows the Tension/Compression spring design problem QQ curves. Table 26 indicates the ordinary one-way ANOVA of Tension Compression.

5.1. Pressure vessel design problem

The problem of the cylindrical vessel (Zou et al., 2011) is that it is capped at both ends by hemispherical heads, as shown in Fig. 10. The problem objective is the minimization of the total cost, which includes the material, forming, and welding costs. Four variables in this design need to be optimized. The first parameter is the thickness of the shell (T_s), and the second is the thickness of the head (T_h). The third and fourth parameters are the inner radius, R, and the length of the cylindrical section, L, not including the head. The parameters of T_s and T_h are integer multiples of 0.0625-inch, the available thickness of steel plates, and R and L are continuous values. The mathematical formulation of the problem can be described as follows:

Minimize

$$\begin{aligned}f(T_s,T_h,R,L)=&0.6224T_sRL+1.7781T_hR^2+\\ &3.1661T_s^2L+19.84T_h^2L\end{aligned} \tag{19}$$

Subject to the following constraints

$$\begin{aligned} g_1&=-T_s+0.0193R\le 0\\ g_2&=-T_h+0.0095R\le 0\\ g_3&=-\pi R^2L-4/3\pi R^3+1,296,000\le 0\\ g_4&=L-240\le 0\end{aligned} \tag{20}$$

Table 15
Mean and standard deviation (StDev) the suggested and compared algorithms over the benchmark functions (f_1 to f_{23}).

Function/Algorithm		GGO	PSO	DE	WOA	GWO	FEP	GSA	GA
f_1	Mean	0	0.000136	8.2E−14	1.41E−30	6.59E−28	0.00057	2.53E−16	4.6E−172
	StDev	0	0.000202	5.9E−14	4.91E−30	6.34E−05	0.00013	9.67E−17	0
f_2	Mean	0	0.042144	1.5E−09	1.06E−21	7.18E−17	0.0081	0.055655	3.44E−90
	StDev	0	0.045421	9.9E−10	2.39E−21	0.029014	0.00077	0.194074	6.13E−90
f_3	Mean	0	70.12562	6.8E−11	5.39E−07	3.29E−06	0.016	896.5347	1.7E−127
	StDev	0	22.11924	7.4E−11	2.93E−06	79.14958	0.014	318.9559	8.6E−127
f_4	Mean	0	1.086481	0	0.072581	5.61E−07	0.3	7.35487	1.15E−75
	StDev	0	0.317039	0	0.39747	1.315088	0.5	1.741452	2.45E−75
f_5	Mean	0	96.71832	0	27.86558	26.81258	5.06	67.54309	28.37287
	StDev	0	60.11559	0	0.763626	69.90499	5.87	62.22534	0.582802
f_6	Mean	0	0.000102	0	3.116266	0.816579	0	2.5E−16	3.932626
	StDev	0	8.28E−05	0	0.532429	0.000126	0	1.74E−16	0.431755
f_7	Mean	0	0.122854	0.00463	0 .001425	0.002213	0.1415	0.089441	0.022992
	StDev	0	0.044957	0.0012	0.001149	0.100286	0.3522	0.04339	0.021966
f_8	Mean	−8219.34578	−4841.29	−11080.1	−5080.76	−6123.1	−12554.5	−2821.07	−4080.18
	StDev	822.514	1152.814	574.7	695.7968	−4087.44	52.6	493.0375	551.6504
f_9	Mean	0	46.70423	69.2	0	0.310521	0.046	25.96841	0
	StDev	0	11.62938	38.8	0	47.35612	0.012	7.470068	0
f_{10}	Mean	3.94E−16	0.276015	9.7E−08	7.4043	1.06E−13	0.018	0.062087	7.99E−16
	StDev	0	0.50901	4.2E−08	9.897572	0.077835	0.0021	0.23628	1.07E−15
f_{11}	Mean	0	0.009215	0	0.000289	0.004485	0.016	27.70154	0
	StDev	0	0.007724	0	0.000289	0.006659	0.022	5.040343	0
f_{12}	Mean	9.98878E−05	0.006917	7.9E−15	0.339676	0.053438	9.2E−06	1.799617	0.556173
	StDev	7.33541E−05	0.026301	8E−15	0.214864	0.020734	3.6E−06	0.95114	0.063582
f_{13}	Mean	6.68974E−05	0.006675	5.1E−14	1.889015	0.654464	0.00016	8.899084	2.132497
	StDev	3.1286E−05	0.008907	4.8E−14	0.266088	0.004474	0.000073	7.126241	0.174792
f_{14}	Mean	0.98900212	3.627168	0.998004	2.111973	4.042493	1.22	5.859838	0.998004
	StDev	8.17E−13	2.560828	3.3E−16	2.498594	4.252799	0.56	3.831299	1.37E−09
f_{15}	Mean	0.000344557	0.000577	4.5E−14	0.000572	0.000337	0.0005	0.003673	0.002318
	StDev	5.1869E−05	0.000222	0.00033	0.000324	0.000625	0.00032	0.001647	0.010072
f_{16}	Mean	−1.03101234	−1.03163	−1.03163	−1.03163	−1.03163	−1.03	−1.03163	−1.03163
	StDev	1.31203E−06	6.25E−16	3.1E−13	4.2E−07	−1.03163	4.9E−07	4.88E−16	4.44E−06
f_{17}	Mean	0.397653247	0.397887	0.397887	0.397914	0.397889	0.398	0.397887	0.398223
	StDev	1.48888E−05	0	9.9E−09	2.7E−05	0.397887	1.5E−07	0	0.001395
f_{18}	Mean	3.452813511	3	3	3	3.000028	3.02	3	3.000029
	StDev	1.82541E−05	1.33E−15	2E−15	4.22E−15	3	0.11	4.17E−15	4.22E−05
f_{19}	Mean	−3.8634587	−3.86278	N/A	−3.85616	−3.86263	−3.86	−3.86278	−3.86272
	StDev	8.39875E−05	2.58E−15	N/A	0.002706	−3.86278	0.000014	2.29E−15	9.02E−05
f_{20}	Mean	−3.49789131	−3.26634	N/A	−2.98105	−3.28654	−3.27	−3.31778	−3.25066
	StDev	0.057996651	0.060516	N/A	0.376653	−3.25056	0.059	0.023081	0.081811
f_{21}	Mean	−11.1227535	−6.8651	−10.1532	−7.04918	−10.1514	−5.52	−5.95512	−6.03721
	StDev	0.009978511	3.019644	2.5E−06	3.629551	−9.14015	1.59	3.737079	1.998973
f_{22}	Mean	−11.0224287	−8.45653	−10.4029	−8.18178	−10.4015	−5.53	−9.68447	−6.76809
	StDev	0.007133788	3.087094	3.9E−07	3.829202	−8.58441	2.12	2.014088	2.628446
f_{23}	Mean	−9.33542997	−9.95291	−10.5364	−9.34238	−10.5343	−6.57	−10.5364	−5.79459
	StDev	0.008833409	1.782786	1.9E−07	2.414737	−8.55899	3.14	2.6E−15	2.643454

where the four variables range are as follows:

$$0 \leq T_s \leq 99, 0 \leq T_h \leq 99, \\ 10 \leq R \leq 200, 10 \leq L \leq 200 \quad (21)$$

A conclusion that can be drawn from the findings presented in Table 28 is that the GGO has the lowest cost, with a difference of about 109 between it and the GWO, which came in second place. Table 28 compares the GGO with some optimization approaches used to tackle the challenge when determining the design variables. Since the GGO obtained the smallest weight, it is clear from the findings in Table 27 that it offers the best option. Table 29 displays the outcomes of this p-values test at 0.0001, which clearly demonstrates how the GGO performs better than other algorithms based on p-values less than 0.0001. Fig. 11 shows the Tension/Compression spring design problem QQ curves. Table 30 indicates the ordinary one-way ANOVA of Tension Compression.

6. CEC 2017 benchmarks

The suggested GGO algorithm is evaluated with ten dimensions for the CEC 2017 benchmarks (Jianhua & Zhiheng, 2021). The CEC 2017 benchmark consists of 29 problems spread over ten dimensions, each with a relevance level of 5%. The statistical results of the proposed GGO algorithm on the CEC 2017 benchmarks with ten dimensions are outlined in Table 34, which provides a table summary. The statistical results of benchmark comparisons with ten dimensions, best and standard deviations of error, are presented in Table 31 below. These results come from the optimal solution of GGO and other state-of-the-art algorithms, and they were compiled from 51 separate runs for each of the 29 benchmark functions. The performance of the suggested method is further demonstrated by the findings based on CEC 2017, which were used. Table 32 shows the p-values of the presented GGO against compared algorithms for the 29 test functions, while Table 33 presents the ANOVA test for GWO, PSO, SFS, BBO, ACO, AMO, GA, and GGO for the 29 test functions. The statistical analysis shows the superiority of the proposed GGO algorithm for the 29 CEC 2017 benchmark functions.

7. Sensitivity analysis of the GGO parameters

This section discusses the parameter sensitivity analysis performed by the GGO. In order to do this study of GGO, we will be using the following parameters: r_1, r_2, r_3, r_4, r_5, w_1, w_2, w_3, and w_4. When applying the method to solve the evaluated optimization problem in this work, these settings determine the algorithm's performance. Every alteration to a parameter has the potential to affect the optimization method.

Table 16
The avg_time, std_time, and avg_FEs of the suggested and compared algorithms over some benchmark functions.

Function/Algorithm		GGO	PSO	GA	GWO	WOA	DE
f_1	avg_time	0.403	1.596	1.555	2.173	0.957	1.869
	std_time	0.024	0.037	0.066	0.082	0.038	0.038
	avg_FEs	7815.000	15 000.000	15 000.000	15 000.000	15 000.000	1530.000
f_2	avg_time	0.981	1.689	2.315	2.280	1.046	2.086
	std_time	0.071	0.021	1.308	0.041	0.013	0.125
	avg_FEs	14 612.000	15 000.000	15 000.000	15 000.000	15 000.000	1530.000
f_3	avg_time	1.998	4.034	4.043	4.559	3.334	5.759
	std_time	0.201	0.041	0.086	0.098	0.044	0.486
	avg_FEs	6312.000	15 000.000	15 000.000	15 000.000	15 000.000	1530.000
f_8	avg_time	0.914	1.706	1.387	2.168	1.003	2.228
	std_time	0.018	0.010	0.011	0.011	0.046	0.253
	avg_FEs	15 000.000	15 000.000	15 000.000	15 000.000	15 000.000	1530.000
f_9	avg_time	0.934	1.735	1.625	2.255	1.034	2.230
	std_time	0.007	0.088	0.116	0.027	0.012	0.213
	avg_FEs	412.000	15 000.000	1605.000	15 000.000	5768.400	1530.000
f_{10}	avg_time	1.145	1.935	1.841	2.434	1.227	2.275
	std_time	0.041	0.024	0.015	0.034	0.011	0.078
	avg_FEs	15 000.000	15 000.000	15 000.000	15 000.000	15 000.000	1530.000
f_{11}	avg_time	0.051	2.001	0.228	1.385	1.290	2.968
	std_time	0.009	0.069	0.017	0.718	0.060	0.706
	avg_FEs	401.000	15 000.000	1839.000	15 000.000	15 000.000	1530.000
f_{12}	avg_time	1.780	2.431	2.394	3.005	1.766	3.396
	std_time	0.044	0.016	0.013	0.097	0.079	0.460
	avg_FEs	15 000.000	15 000.000	15 000.000	15 000.000	15 000.000	1530.000
f_{13}	avg_time	1.623	2.346	2.322	2.848	1.698	3.176
	std_time	0.033	0.009	0.016	0.042	0.063	0.398
	avg_FEs	15 000.000	15 000.000	15 000.000	15 000.000	15 000.000	1530.000
f_{17}	avg_time	0.466	0.276	0.777	0.316	0.272	0.961
	std_time	0.018	0.008	0.011	0.011	0.049	0.026
	avg_FEs	15 000.000	15 000.000	15 000.000	15 000.000	15 000.000	1530.000
f_{21}	avg_time	3.444	3.305	3.603	3.299	3.360	5.107
	std_time	0.044	0.039	0.020	0.036	0.247	0.094
	avg_FEs	15 000.000	15 000.000	15 000.000	15 000.000	15 000.000	1530.000
f_{23}	avg_time	6.108	6.839	7.112	6.711	6.997	9.812
	std_time	0.133	0.214	0.023	0.067	0.421	0.091
	avg_FEs	15 000.000	15 000.000	15 000.000	15 000.000	15 000.000	1530.000

Table 17
ANOVA test for benchmark functions from f_1 to f_4.

f_1					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	2.42E−07	4	6.05E−08	F (4, 145) = 12.04	P < 0.0001
Residual (within columns)	7.29E−07	145	5.03E−09	–	–
Total	9.71E−07	149	–	–	–
f_2					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	269.7	4	67.43	F (4, 145) = 7.730	P < 0.0001
Residual (within columns)	1265	145	8.723	–	–
Total	1535	149	–	–	–
f_3					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	4.25E+10	4	1.06E+10	F (4, 145) = 186.8	P < 0.0001
Residual (within columns)	8.24E+09	145	56 830 905	–	–
Total	5.07E+10	149	–	–	–
f_4					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	61 550	4	15 387	F (4, 145) = 92.40	P < 0.0001
Residual (within columns)	24 147	145	166.5		
Total	85 697	149			

7.1. One-at-a-time sensitivity analysis

The One-at-a-Time (OAT) sensitivity measure has been utilized in the process of examining sensitivity (Confalonieri, Bellocchi, Bregaglio, Donatelli, & Acutis, 2010). OAT is widely acknowledged as one of the most straightforward approaches to sensitivity analysis. OAT evaluates the efficacy of an algorithm by changing only one of its parameters while leaving the others unaltered. The observed changes in GGO's time and fitness values for F22 are detailed in Tables 35 and 36, respectively, when the parameter values are tweaked. After adding 5% to the length of each parameter of r_1, r_2, r_3, r_4, and r_5 while adding 10% to the length of each parameter of w_1, w_2, w_3, and w_4 in order to obtain a new evaluation value, we chose 20 unique values from within the interval of each parameter and displayed them in Tables 35 and 36.

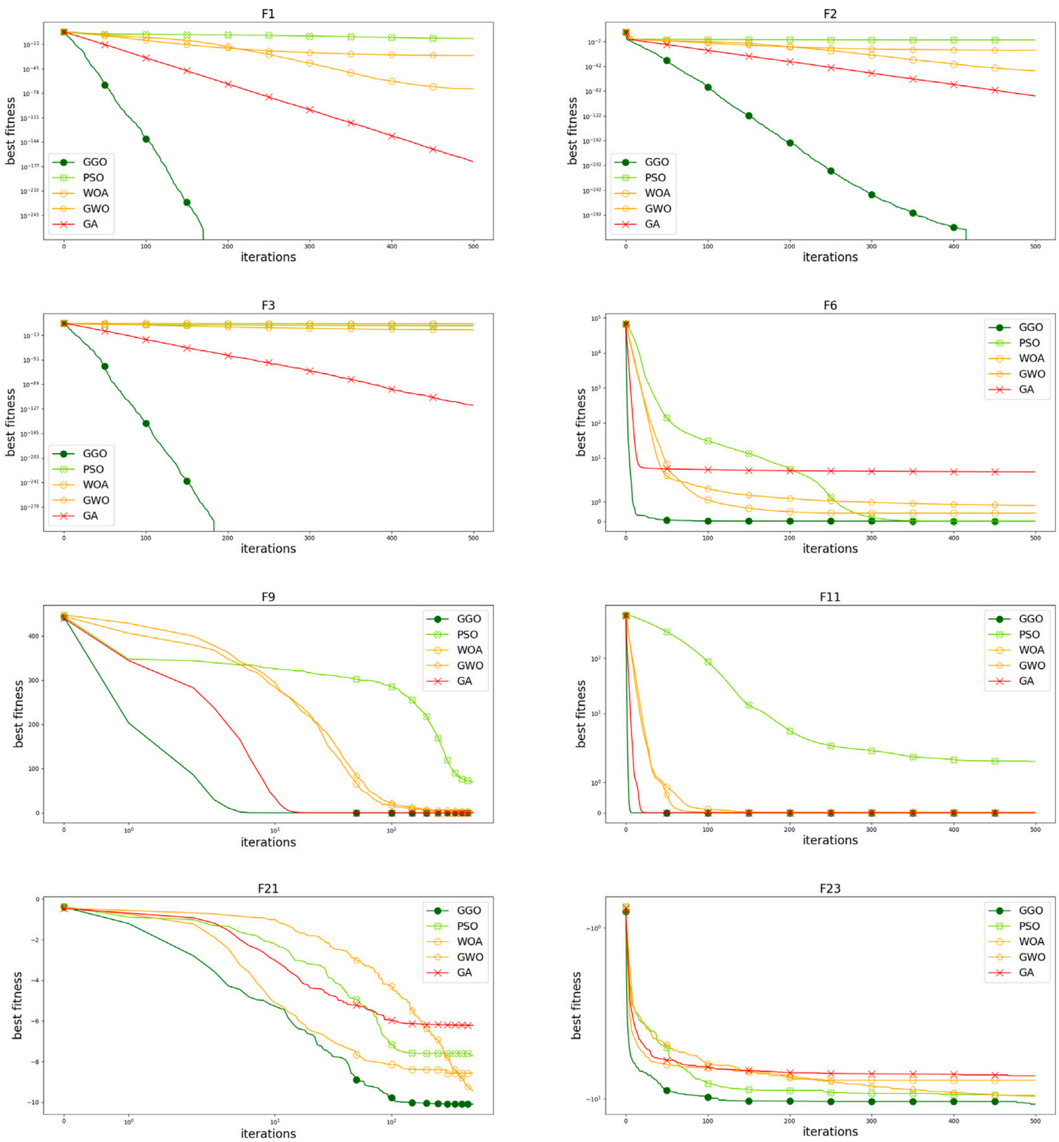

Fig. 4. Convergence curve of the presented and compared algorithms for the functions of f_1, f_2, f_3, f_6, f_9, f_{11}, f_{21} and f_{23}.

The algorithm was run ten times on each of these variables, and the table below shows the averages for the amount of time and how well the variables performed. The analysis of the convergence time of F22 is depicted in Fig. 12. In addition, the analysis of the convergence fitness of F22 is illustrated in Fig. 13.

7.2. Statistical significance

An analysis of variance (ANOVA) was performed on the convergence time and fitness values while altering the GGO's parameters in order to assess whether or not there is a statistically significant difference between the means of the data found in Tables 35 and 36. Tables 37 and 38 present the findings of the ANOVA test that was conducted by the GGO. In Tables 37 and 38, every one of the p-values is lower than 0.05. Adjusting the parameter values results in a difference in mean values that is statistically significant between the nine groups of convergence time and the nine groups of minimal fitness. This difference may be seen when looking at the data. The level of significance used in this test was 0.05, and the T-test that was done had one tail. The results of the T-test for each observed pair of convergence time and minimal fitness parameters for GGO are provided in Tables 39 and 40 respectively. In the tables, a statistically significant difference between the groups is shown by p-values with a value of less than 0.05.

8. Conclusion

In this research, a new optimizer inspired by nature called the Greylag Goose Optimization (GGO) method was developed. The Greylag Goose served as the model for the algorithm that will be developed, which will be denoted as GGO and will be of the swarm-based variety. The flying ability of geese is second to none. They are able to travel thousands of kilometers in a single flight while on their annual migrations because they travel in large groups and fly together. While in the air, a flock of geese will often create the letter "V" with their bodies. The geese in the front can help reduce the amount of air resistance experienced by the geese at the back in this way. Because of this, the geese can go approximately 70 percent further as a flock than they could fly on their own. In the first step of the validation process, the GGO algorithm is applied to nineteen datasets that have been retrieved from the UCI Machine Learning Repository. Each dataset has a different number of characteristics, instances, and classes, all of which are utilized in the process of selecting features. After that, it is put to use in the process of resolving a variety of engineering benchmark functions and case studies. Afterward, The GGO algorithm is used whenever there is a need to find solutions to situations that include constrained optimization. The two designs that have been proven through testing are

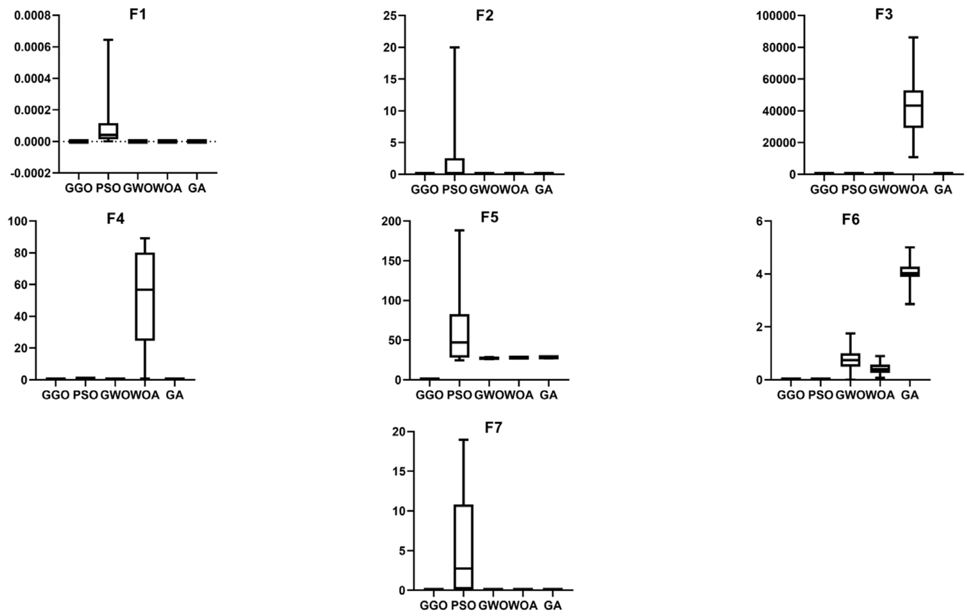

Fig. 5. Box plot of the suggested and compared algorithms for benchmark function (f_1 to f_7).

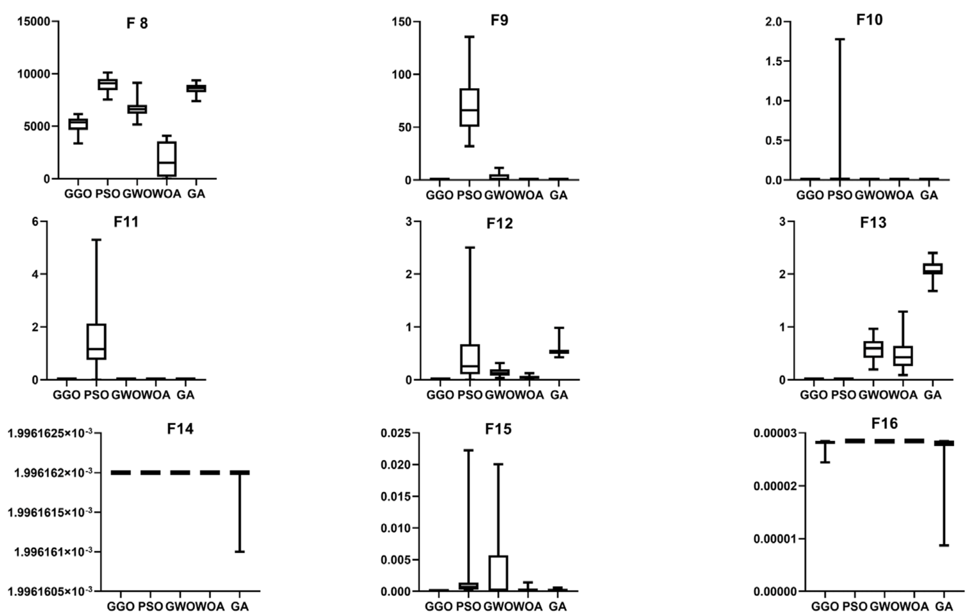

Fig. 6. Box plot of the suggested and compared algorithms for benchmark function (f_8 to f_{16}).

the pressure vessel design and the tension/compression spring design. Both of these designs involve springs. According to the data, the GGO approach outperforms a wide variety of other comparative optimization algorithms and provides greater accuracy in comparison to different algorithms than other algorithms. The findings of the statistical analysis tests, such as Wilcoxon's rank-sum and one-way analysis of variance (ANOVA), show that the GGO algorithm generates superior results. These findings are presented in the following sentence.

CRediT authorship contribution statement

El-Sayed M. El-kenawy: Conceptualization, Software, Data curation, Visualization, Project administration. **Nima Khodadadi:** Validation, Investigation, Resources, Writing – review & editing. **Seyedali Mirjalili:** Methodology, Validation, Writing – review & editing, Supervision. **Abdelaziz A. Abdelhamid:** Methodology, Formal analysis, Writing – original draft. **Marwa M. Eid:** Software, Formal analysis, Data curation, Visualization. **Abdelhameed Ibrahim:** Conceptualization, Investigation, Resources, Writing – original draft, Project administration.

Declaration of competing interest

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.

Data availability

Data will be made available on request.

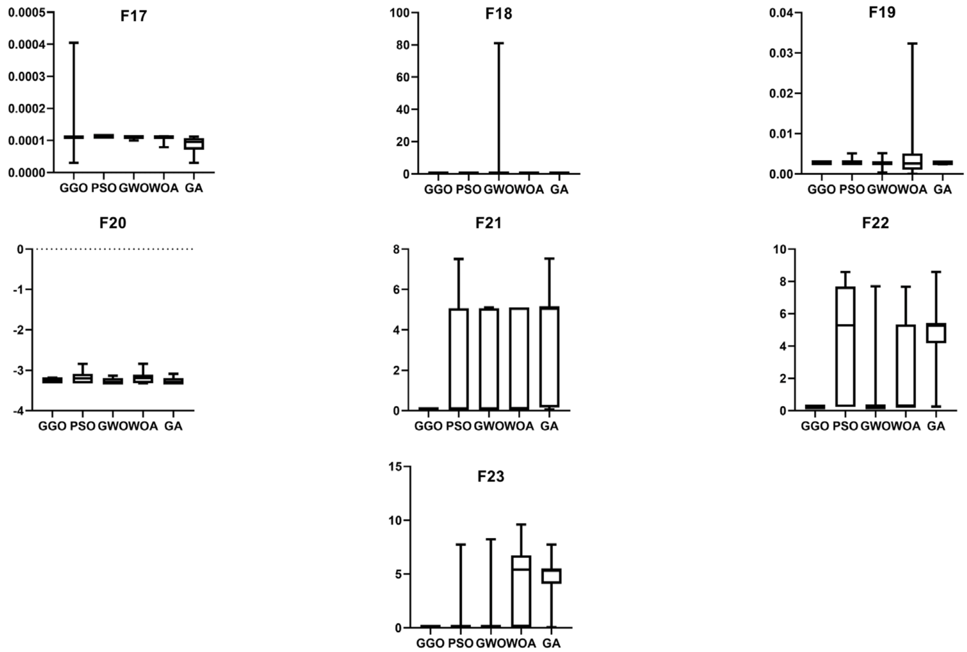

Fig. 7. Box plot of the suggested and compared algorithms for benchmark function (f_{17} to f_{23}).

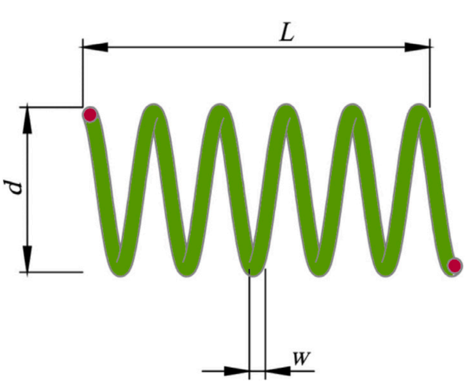

Fig. 8. Tension/Compression spring design problem (Celik & Kutucu, 2018).

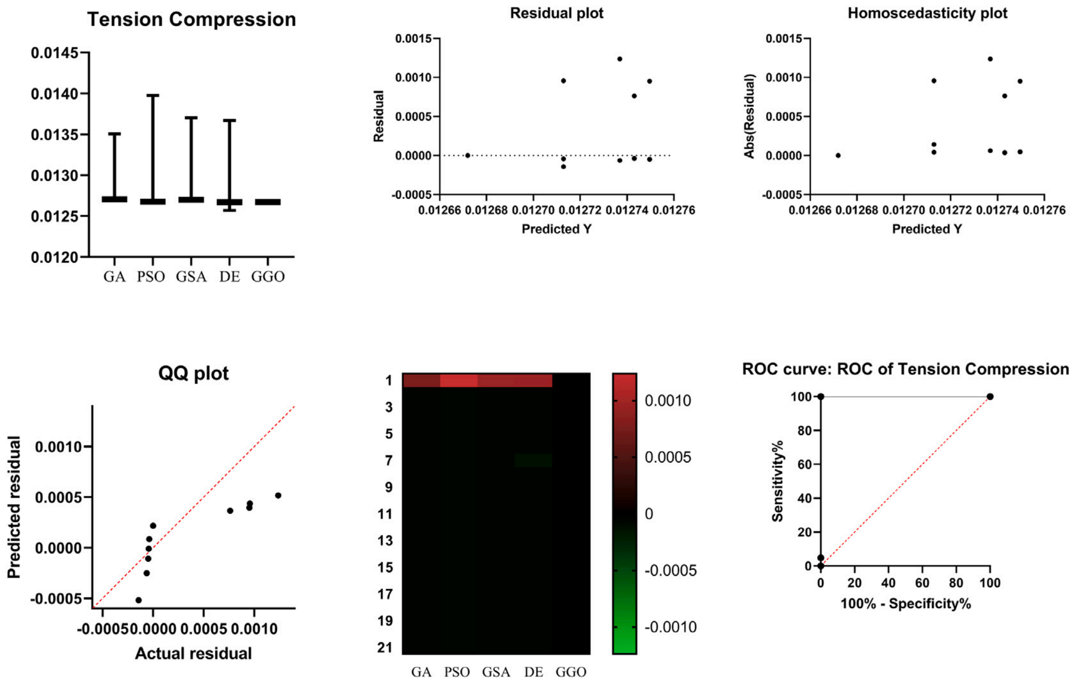

Fig. 9. Tension/Compression spring design problem QQ curves.

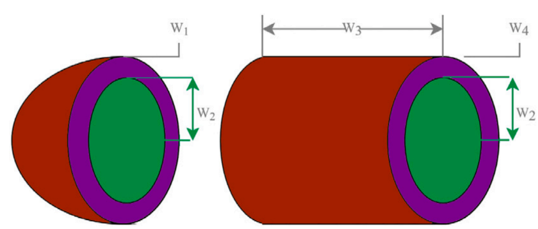

Fig. 10. Pressure Vessel design.

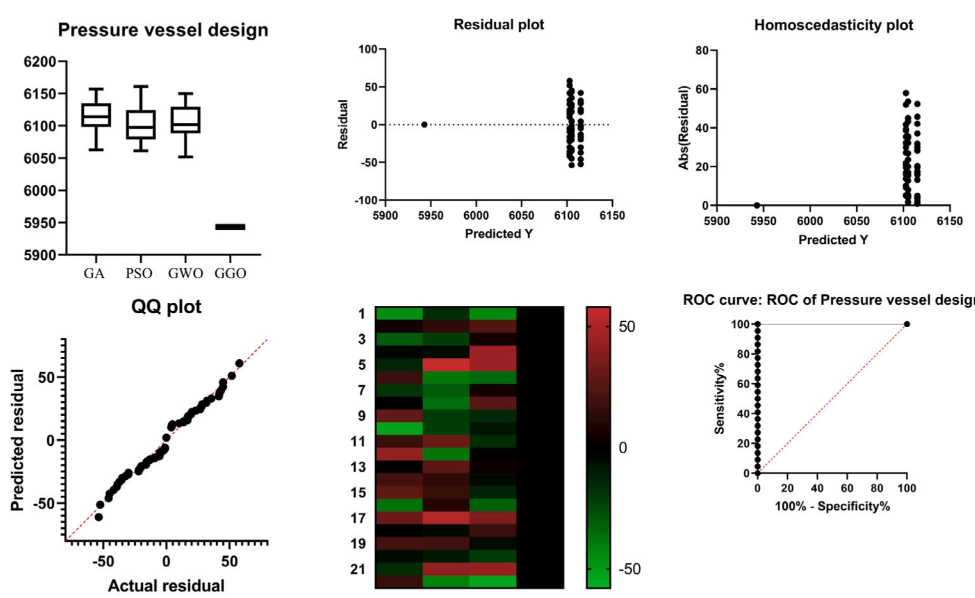

Fig. 11. Pressure Vessel design problem QQ curves.

Table 18
ANOVA test for benchmark functions from f_5 to $f_1$4.

	SS	DF	MS	F (DFn, DFd)	P value
f_5					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	59 376	4	14 844	F (4, 145) = 42.46	P < 0.0001
Residual (within columns)	50 691	145	349.6		
Total	110 067	149			
f_6					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	355.1	4	88.78	F (4, 145) = 1261	P < 0.0001
Residual (within columns)	10.21	145	0.07039		
Total	365.3	149			
f_7					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	566.5	4	141.6	F (4, 145) = 16.82	P < 0.0001
Residual (within columns)	1221	145	8.422		
Total	1788	149			
f_7					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	9.95E+08	4	2.49E+08	F (4, 145) = 271.6	P < 0.0001
Residual (within columns)	1.33E+08	145	915 567		
Total	1.13E+09	149			
f_9					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	117 171	4	29 293	F (4, 145) = 226.6	P<0.0001
Residual (within columns)	18 741	145	129.3		
Total	135 912	149			
f_{10}					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	0.2663	4	0.06657	F (4, 145) = 2.317	P=0.0600
Residual (within columns)	4.167	145	0.02874		
Total	4.433	149			
f_{11}					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	56.84	4	14.21	F (4, 145) = 46.72	P<0.0001
Residual (within columns)	44.11	145	0.3042		
Total	101	149			
f_{12}					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	7.514	4	1.878	F (4, 145) = 28.01	P<0.0001
Residual (within columns)	9.723	145	0.06706		
Total	17.24	149			
f_{14}					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	4.27E−19	4	1.07E−19	F (4, 145) = 4.462	P=0.0020
Residual (within columns)	3.47E−18	145	2.39E−20		
Total	3.89E−18	149			

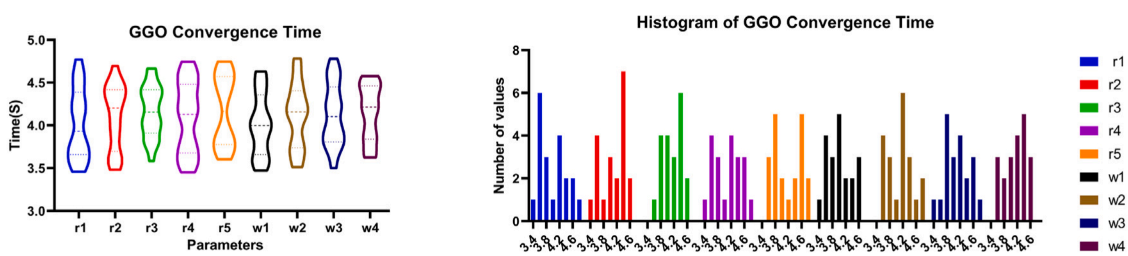

Fig. 12. Convergence time of f_{22}.

Table 19
ANOVA test for benchmark functions from f_{15} to f_{23}.

f_{15}					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	0.000608	4	0.000152	F (4, 145) = 6.239	P=0.0001
Residual (within columns)	0.00353	145	2.44E−05		
Total	0.004138	149			
f_{16}					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	4.27E−11	4	1.07E−11	F (4, 145) = 3.630	P=0.0075
Residual (within columns)	4.27E−10	145	2.94E−12		
Total	4.70E−10	149			
f_{17}					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	1.46E−08	4	3.64E−09	F (4, 145) = 4.633	P=0.0015
Residual (within columns)	1.14E−07	145	7.85E−10		
Total	1.28E−07	149			
f_{18}					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	175	4	43.74	F (4, 145) = 1.000	P=0.4097
Residual (within columns)	6342	145	43.74		
Total	6517	149			
f_{19}					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	0.000123	4	3.07E−05	F (4, 145) = 2.965	P=0.0217
Residual (within columns)	0.0015	145	1.03E−05		
Total	0.001623	149			
f_{20}					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	0.1186	4	0.02964	F (4, 145) = 2.915	P=0.0235
Residual (within columns)	1.474	145	0.01017		
Total	1.593	149			
f_{21}					
ANOVA table	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	225.5	4	56.38	F (4, 145) = 11.67	P<0.0001
Residual (within columns)	700.6	145	4.832		
Total	926.2	149			
f_{22}					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	352.5	4	88.11	F (4, 145) = 13.56	P<0.0001
Residual (within columns)	942.5	145	6.5		
Total	1295	149			
f_{23}					
	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	561.1	4	140.3	F (4, 145) = 28.82	P<0.0001
Residual (within columns)	705.7	145	4.867		
Total	1267	149			

Table 20
Wilcoxon Signed Rank Test for the benchmark function f_1 based on the suggested GGO algorithm against the compared algorithms.

	GGO	PSO	GWO	WOA	GA
Theoretical median	0	0	0	0	0
Actual median	0	4.13E−05	6.15E−28	4.03E−78	2.03E−174
Number of values	30	30	30	30	30
Wilcoxon Signed Rank Test	Samples with equal values				
Sum of signed ranks (W)		465	465	465	465
Sum of positive ranks		465	465	465	465
Sum of negative ranks		0	0	0	0
P value (two tailed)		<0.0001	<0.0001	<0.0001	<0.0001
Exact or estimate?		Exact	Exact	Exact	Exact
P value summary		****	****	****	****
Significant (alpha=0.05)?		Yes	Yes	Yes	Yes
How big is the discrepancy?					
Discrepancy		4.13E−05	6.15E−28	4.03E−78	2.03E−174

Table 21
Wilcoxon Signed Rank Test for the benchmark function f_8 based on the suggested GGO algorithm against the compared algorithms.

	GGO	PSO	GWO	WOA	GA
Theoretical median	0	0	0	0	0
Actual median	5362	9109	6614	1504	8636
Number of values	30	30	30	30	30
Wilcoxon Signed Rank Test					
Sum of signed ranks (W)	465	465	465	465	465
Sum of positive ranks	465	465	465	465	465
Sum of negative ranks	0	0	0	0	0
P value (two-tailed)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Exact or estimate?	Exact	Exact	Exact	Exact	Exact
P value summary	****	****	****	****	****
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?					
Discrepancy	5362	9109	6614	1504	8636

Table 22
Wilcoxon Signed Rank Test for the benchmark function f_{14} based on the suggested GGO algorithm against the compared algorithms.

	GGO	PSO	GWO	WOA	GA
Theoretical median	0	0	0	0	0
Actual median	0.001996	0.001996	0.001996	0.001996	0.001996
Number of values	30	30	30	30	30
Wilcoxon Signed Rank Test					
Sum of signed ranks (W)	465	465	465	465	465
Sum of positive ranks	465	465	465	465	465
Sum of negative ranks	0	0	0	0	0
P value (two-tailed)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Exact or estimate?	Exact	Exact	Exact	Exact	Exact
P value summary	****	****	****	****	****
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?					
Discrepancy	0.001996	0.001996	0.001996	0.001996	0.001996

Table 23
Comparison of the best solution for tension/compression spring design problem.

Parameters/Algorithm	GA	PSO	GSA	DE	GGO
w	0.051480	0.051728	0.050276	0.051609	0.0517805
d	0.351661	0.357644	0.323680	0.354714	0.35884616
L	11.632201	11.244543	13.525410	11.410831	11.17045364
g_1	−0.0033366	−8.25E−04	−4.9415155	−3.90E−05	−0.00022344
g_2	−0.0001097	−2.52E−05	3.7734181	−1.83E−04	−0.00016911
g_3	−4.0263180	−4.051306	−3.9831079	−4.048627	−4.05592442
g_4	−0.7312393	−0.727085	−0.750696	−0.729118	−0.72624889
f	0.0127048	0.0126747	0.0127022	0.0126702	0.0126718

Table 24
Descriptive statistics of tension compression.

	GA	PSO	GSA	DE	GGO
Number of values	21	21	21	21	21
Minimum	0.01271	0.01268	0.0127	0.01257	0.01267
Maximum	0.01351	0.01398	0.0137	0.01367	0.01267
Range	0.0008	0.0013	0.001	0.0011	0
Mean	0.01274	0.01274	0.01275	0.01271	0.01267
Std. Deviation	0.000175	0.000284	0.000218	0.00022	0
Std. Error of Mean	3.81E−05	6.19E−05	4.76E−05	4.81E−05	0

Table 25
One sample Wilcoxon test of tension compression.

	GA	PSO	GSA	DE	GGO
Theoretical median	0	0	0	0	0
Actual median	0.01271	0.01268	0.0127	0.01267	0.01267
Number of values	21	21	21	21	21
Wilcoxon Signed Rank Test					
Sum of signed ranks (W)	231	231	231	231	231
Sum of positive ranks	231	231	231	231	231
Sum of negative ranks	0	0	0	0	0
P value (two-tailed)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Exact or estimate?	Exact	Exact	Exact	Exact	Exact
P value summary	****	****	****	****	****
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?					
Discrepancy	0.01271	0.01268	0.0127	0.01267	0.01267

Table 26
Ordinary one-way ANOVA of tension compression.

	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	8.42E−08	4	2.11E−08	F (4, 100) = 0.5081	P=0.7299
Residual (within columns)	4.14E−06	100	4.14E−08		
Total	4.23E−06	104			

Table 27
Comparison of the best solution for pressure vessel design problem.

Parameters/Algorithm	GA	PSO	GWO	GGO
Ts	0.8125	0.8125	0.0812500	0.78382745
Th	0.4375	0.4375	0.434500	0.38873565
R	42.097398	42.0913	42.089181	40.49133483
L	176.65405	176.7465	176.758731	198.91194323
g_1	−0.000020	−1.37E−06	0.73107119	−0.00234468
g_2	−0.035891	−3.59E−04	−0.03296921	−0.00244831
g_3	−27.886075	−118.7687	−40.6168247	−6637.4479
g_4	−63.345953	−63.2535	−63.2412689	−41.088056
f	6059.9463	6061.0777	6051.5639	5943.04385697

Table 28
Descriptive statistics of pressure.

	GA	PSO	GWO	GGO
Number of values	22	22	22	22
Minimum	6063	6061	6052	5943
Maximum	6157	6161	6150	5943
Range	94.43	99.67	98.64	0
Mean	6115	6103	6105	5943
Std. Deviation	25.87	30.43	28.56	0
Std. Error of Mean	5.515	6.488	6.089	0

Table 29
One sample Wilcoxon test of pressure.

	GA	PSO	GWO	GGO
Theoretical median	0	0	0	0
Actual median	6114	6098	6102	5943
Number of values	22	22	22	22
Wilcoxon Signed Rank Test				
Sum of signed ranks (W)	253	253	253	253
Sum of positive ranks	253	253	253	253
Sum of negative ranks	0	0	0	0
P value (two-tailed)	<0.0001	<0.0001	<0.0001	<0.0001
Exact or estimate?	Exact	Exact	Exact	Exact
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes
How big is the discrepancy?				
Discrepancy	6114	6098	6102	5943

Table 30
Ordinary one-way ANOVA of pressure.

	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	449 096	3	149 699	F (3, 84) = 248.4	P<0.0001
Residual (within columns)	50 625	84	602.7	–	–
Total	499 721	87	–	–	–

Table 31
The results of experiments using GWO, PSO, SFS, BBO, ACO, AMO, GA, and GGO with a total of 51 separate runs, 29 test functions, 10 variables, and 100,000 FES.

	GWO	PSO	SFS	BBO	ACO	AMO	GA	GGO
CEC-1	1.50E+08 ± 6.50E+07	2.30E+03 ± 3.05E+03	5.60E+03 ± 3.26E+03	4.20E+06 ± 1.87E+05	8.95E+10 ± 2.59E+10	5.34E+00 ± 7.57E+00	6.06E+06 ± 1.62E+06	0.00E+00 ± 0.00E+00
CEC-2	5.28E+02 ± 7.98E+02	0.00E+00 ± 0.00E+00	2.05E−02 ± 1.17E−02	4.39E+04 ± 2.58E+04	2.08E+07 ± 1.23E+08	4.56E+03 ± 1.48E+03	6.81E+04 ± 2.02E+04	0.00E+00 ± 0.00E+00
CEC-3	1.84E+01 ± 1.62E+01	2.82E+00 ± 1.21E+00	9.37E−01 ± 6.74E−01	9.99E+01 ± 2.27E+01	9.61E+03 ± 1.72E+03	4.60E+00 ± 1.07E+01	1.51E+02 ± 4.09E+01	0.00E+00 ± 0.00E+00
CEC-4	3.01E+01 ± 4.85E+00	1.59E+01 ± 7.08E+00	8.68E+00 ± 3.30E+00	4.21E+01 ± 1.06E+01	4.11E+02 ± 2.27E+01	5.41E+01 ± 6.68E+00	2.26E+02 ± 2.93E+01	0.00E+00 ± 0.00E+00
CEC-5	7.81E+00 ± 1.09E+00	8.27E−02 ± 3.36E−01	5.35E−03 ± 1.59E−03	8.98E−01 ± 4.07E−02	8.31E+01 ± 6.11E+00	0.00E+00 ± 0.00E+00	3.86E+01 ± 9.95E+00	0.00E+00 ± 0.00E+00
CEC-6	4.35E+01 ± 5.14E+00	1.72E+01 ± 4.46E+00	2.33E+01 ± 3.60E+00	1.14E+02 ± 1.46E+01	9.44E+02 ± 9.83E+01	9.15E+01 ± 6.90E+00	3.24E+02 ± 5.43E+01	0.00E+00 ± 0.00E+00
CEC-7	2.38E+01 ± 4.44E+00	1.23E+01 ± 5.33E+00	7.42E+00 ± 2.76E+00	4.31E+01 ± 1.00E+01	3.63E+02 ± 1.97E+01	5.42E+01 ± 5.76E+00	2.43E+02 ± 3.07E+01	0.00E+00 ± 0.00E+00
CEC-8	1.10E+01 ± 3.79E+00	0.00E+00 ± 0.00E+00	6.68E−06 ± 4.34E−06	1.09E+02 ± 8.42E+01	1.39E+04 ± 1.46E+03	1.05E−01 ± 2.07E−01	1.12E+03 ± 1.42E+03	0.00E+00 ± 0.00E+00
CEC-9	8.53E+02 ± 2.50E+02	6.30E+02 ± 2.64E+02	3.63E+02 ± 1.91E+02	2.29E+03 ± 4.48E+02	7.38E+03 ± 2.24E+02	1.05E−01 ± 2.07E−01	4.48E+03 ± 8.44E+02	0.00E+00 ± 0.00E+00
CEC-10	4.03E+01 ± 9.91E+00	1.10E+01 ± 7.27E+00	4.61E+00 ± 1.25E+00	1.43E+03 ± 1.40E+03	5.54E+03 ± 1.53E+03	4.91E+01 ± 2.47E+01	1.33E+03 ± 1.01E+03	0.00E+00 ± 0.00E+00
CEC-11	2.58E+06 ± 3.10E+06	1.33E+04 ± 1.24E+04	5.04E+03 ± 2.11E+03	3.52E+06 ± 2.19E+06	1.89E+10 ± 4.59E+09	6.07E+04 ± 6.88E+04	4.29E+06 ± 3.34E+06	0.00E+00 ± 0.00E+00
CEC-12	1.16E+04 ± 8.11E+03	6.45E+03 ± 5.72E+03	4.55E+01 ± 9.83E+00	1.51E+06 ± 4.77E+05	1.23E+10 ± 7.17E+09	6.78E+03 ± 3.23E+03	2.54E+06 ± 2.22E+06	0.00E+00 ± 0.00E+00
CEC-13	5.91E+02 ± 1.21E+03	4.57E+01 ± 1.93E+01	2.20E+01 ± 3.82E+00	8.23E+05 ± 8.54E+05	9.44E+06 ± 2.25E+07	2.22E+03 ± 1.39E+03	9.85E+05 ± 9.57E+05	0.00E+00 ± 0.00E+00
CEC-14	8.04E+02 ± 1.12E+03	5.62E+01 ± 5.89E+01	1.00E+01 ± 2.14E+00	7.01E+05 ± 3.26E+05	3.56E+09 ± 2.14E+09	4.22E+02 ± 5.61E+02	7.21E+05 ± 2.53E+05	0.00E+00 ± 0.00E+00
CEC-15	8.52E+01 ± 9.41E+01	2.05E+02 ± 1.19E+02	4.17E+00 ± 3.16E+00	8.95E+02 ± 3.11E+02	2.90E+03 ± 2.10E+02	5.52E+02 ± 1.19E+02	1.17E+03 ± 3.00E+02	3.67E+00 ± 3.75E+00
CEC-16	5.42E+01 ± 8.45E+00	4.56E+01 ± 2.30E+01	2.34E+01 ± 5.66E+00	3.81E+02 ± 2.16E+02	1.28E+03 ± 1.66E+02	8.49E+01 ± 1.77E+01	6.43E+02 ± 2.04E+02	1.17E+01 ± 6.18E+00
CEC-17	3.68E+04 ± 2.11E+04	5.10E+03 ± 5.76E+03	5.22E+01 ± 1.05E+01	1.95E+06 ± 1.96E+06	1.78E+08 ± 1.13E+08	1.43E+05 ± 5.21E+04	3.77E+06 ± 4.84E+06	3.34E−01 ± 2.2E−01
CEC-18	1.75E+03 ± 3.81E+03	9.57E+01 ± 2.95E+02	5.84E+00 ± 8.94E−01	8.37E+05 ± 3.40E+05	3.91E+09 ± 2.60E+09	1.32E+03 ± 1.53E+03	8.32E+05 ± 3.31E+05	0.00E+00 ± 0.00E+00
CEC-19	7.76E+01 ± 3.83E+01	5.51E+01 ± 5.42E+01	1.21E+01 ± 3.31E+00	4.33E+02 ± 1.85E+02	8.37E+02 ± 1.02E+02	1.42E+02 ± 4.76E+01	4.10E+02 ± 1.11E+02	0.00E+00 ± 0.00E+00
CEC-20	2.03E+02 ± 4.93E+01	1.79E+02 ± 5.65E+01	1.00E+02 ± 5.06E−02	2.49E+02 ± 1.04E+01	5.90E+02 ± 1.72E+01	2.53E+02 ± 6.95E+00	4.57E+02 ± 3.28E+01	0.00E+00 ± 0.00E+00
CEC-21	1.25E+02 ± 6.03E+00	9.38E+01 ± 2.55E+01	9.24E+01 ± 3.00E+01	1.58E+03 ± 1.49E+03	5.74E+03 ± 4.29E+02	1.00E+02 ± 0.00E+00	5.16E+03 ± 1.60E+03	1.00E+02 ± 4.17E−01
CEC-22	3.33E+02 ± 3.86E+00	3.28E+02 ± 1.24E+01	3.03E+02 ± 4.35E+01	4.02E+02 ± 1.24E+01	9.40E+02 ± 4.40E+01	3.98E+02 ± 9.02E+00	6.77E+02 ± 3.70E+01	2.11E+02 ± 2.32E+00
CEC-23	3.63E+02 ± 4.56E+00	3.24E+02 ± 8.33E+01	2.18E+02 ± 1.18E+02	4.67E+02 ± 1.33E+01	1.06E+03 ± 5.58E+01	4.64E+02 ± 8.44E+00	8.14E+02 ± 6.40E+01	3.04E+02 ± 1.06E+01
CEC-24	4.42E+02 ± 1.56E+01	4.25E+02 ± 2.29E+01	4.21E+02 ± 2.30E+01	3.93E+02 ± 9.81E+00	3.49E+03 ± 6.51E+02	3.87E+02 ± 1.07E+00	5.61E+02 ± 9.50E+01	4.66E+02 ± 2.43E+01
CEC-25	4.09E+02 ± 1.47E+02	2.74E+02 ± 7.63E+01	2.92E+02 ± 4.40E+01	1.63E+03 ± 1.56E+02	7.10E+03 ± 4.27E+02	1.49E+03 ± 1.95E+02	3.61E+03 ± 8.15E+02	2.02E+02 ± 0.07E+00
CEC-26	3.96E+02 ± 1.15E+00	4.03E+02 ± 1.97E+01	3.92E+02 ± 1.85E+00	5.26E+02 ± 7.66E+00	1.14E+03 ± 7.37E+01	5.16E+02 ± 4.81E+00	6.13E+02 ± 4.04E+01	3.62E+02 ± 1.12E−01
CEC-27	5.39E+02 ± 9.99E+01	4.54E+02 ± 1.57E+02	3.06E+02 ± 3.97E+01	4.33E+02 ± 2.47E+01	3.51E+03 ± 4.77E+02	3.14E+02 ± 3.61E+01	5.66E+02 ± 4.52E+01	3.09E+02 ± 4.13E+01
CEC-28	2.94E+02 ± 3.04E+01	3.05E+02 ± 4.50E+01	2.59E+02 ± 1.18E+01	7.18E+02 ± 1.45E+02	2.59E+03 ± 2.20E+02	5.33E+02 ± 2.46E+01	9.07E+02 ± 1.78E+02	2.91E+02 ± 4.96E+00
CEC-29	4.84E+05 ± 7.31E+05	2.00E+05 ± 3.79E+05	2.03E+03 ± 1.63E+03	3.31E+05 ± 1.23E+05	3.16E+09 ± 9.12E+08	4.71E+03 ± 7.99E+02	3.45E+05 ± 1.43E+05	4.82E+02 ± 3.92E+01

Table 32
The p-values of the presented GGO against compared algorithms for the 29 test functions.

	GWO	PSO	SFS	BBO	ACO	AMO	GA
CEC-1	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-2	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-3	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-4	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-5	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-6	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-7	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-8	<0.0001	<0.0001	<0.0001	<0.0001	0.0633	<0.0001	<0.0001
CEC-9	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0631
CEC-10	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-11	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-12	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-13	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-14	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	0.0573	<0.0001
CEC-15	<0.0001	<0.0001	<0.0001	0.0863	<0.0001	<0.0001	<0.0001
CEC-16	<0.0001	<0.0001	<0.0001	<0.0001	0.0756	<0.0001	<0.0001
CEC-17	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-18	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-19	<0.0001	0.0863	0.0613	<0.0001	<0.0001	<0.0001	<0.0001
CEC-20	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-21	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-22	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-23	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-24	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-25	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-26	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-27	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-28	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
CEC-29	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001

Table 33
ANOVA test for GWO, PSO, SFS, BBO, ACO, AMO, GA, and GGO for the 29 test functions.

	SS	DF	MS	F (DFn, DFd)	P value
Row Factor	145.3	29	5.011	F (29, 116) = 1.037	P=0.4274
Column Factor	561.1	4	140.3	F (4, 116) = 29.04	P<0.0001
Residual	560.4	116	4.831	–	–

Table 34
Results of GGO in 10D for the 29 test functions.

	Best	Worst	Median	Mean	Std
CEC-1	0	0	0	0	0
CEC-2	0	0	0	0	0
CEC-3	0	0	0	0	0
CEC-4	0	0	0	0	0
CEC-5	0	0	0	0	0
CEC-6	0	0	0	0	0
CEC-7	0	0	0	0	0
CEC-8	0	0	0	0	0
CEC-9	0	0	0	0	0
CEC-10	0	0	0	0	0
CEC-11	0	0	0	0	0
CEC-12	0	0	0	0	0
CEC-13	0	0	0	0	0
CEC-14	0	0	0	0	0
CEC-15	0.102847	115.7061	0	0	2.846899
CEC-16	1.001688	19.19553	6.196043	4.366698	7.836576
CEC-17	0.00216	8.87959	0.349836	0	5.891857
CEC-18	0	0	0	0	0
CEC-19	0	7.429979	4.444986	−2.39046	6.355141
CEC-20	52.90816	140.0878	136.4649	131.464	2.234339
CEC-21	0	84.445	84.445	76.96873	10.42859
CEC-22	51.23911	244.685	239.0028	239.3027	13.98651
CEC-23	180	322.8577	317.5574	314.6461	3.472153
CEC-24	256.8147	376.9422	331.574	334.2624	4.782208
CEC-25	179.12	280.7144	233.565	234.4895	8.95277
CEC-26	138.1255	326.2499	323.083	323.0971	14.76841
CEC-27	149.12	545.3868	304.7532	331.8775	49.00764
CEC-28	64.58027	211.9132	168.9655	172.1798	5.89391
CEC-29	143.6208	816 061.6	−1073.31	30976.78	158 672.9

Table 35
Results of the convergent time experimented with at various levels of the GGO’s parameters.

r_1		r_2		r_3		r_4		r_5		w_1		w_2		w_3		w_4	
Values	Time	Values	Time	Values	Time	Values	Time	Values	Time	Values	Time	Values	Time	Values	Time	Values	Time
0.05	4.137	0.05	3.586	0.05	4.267	0.05	3.534	0.05	4.298	0.1	3.999	0.1	4.438	0.1	3.971	0.1	3.664
0.1	4.435	0.1	4.479	0.1	4.313	0.1	3.740	0.1	4.401	0.2	3.526	0.2	4.458	0.2	3.500	0.2	3.816
0.15	4.630	0.15	3.923	0.15	4.472	0.15	3.825	0.15	3.920	0.3	4.392	0.3	3.690	0.3	3.652	0.3	4.226
0.2	4.210	0.2	4.408	0.2	4.450	0.2	4.245	0.2	3.851	0.4	4.521	0.4	4.113	0.4	3.820	0.4	3.806
0.25	3.813	0.25	4.422	0.25	3.812	0.25	4.130	0.25	3.773	0.5	4.139	0.5	4.214	0.5	3.803	0.5	4.219
0.3	4.243	0.3	3.607	0.3	3.995	0.3	4.484	0.3	4.545	0.6	3.996	0.6	4.299	0.6	4.653	0.6	4.427
0.35	3.460	0.35	4.308	0.35	3.585	0.35	4.745	0.35	3.848	0.7	4.025	0.7	4.318	0.7	4.259	0.7	4.478
0.4	3.574	0.4	3.658	0.4	4.112	0.4	3.657	0.4	3.679	0.8	3.707	0.8	4.571	0.8	3.776	0.8	4.477
0.45	3.832	0.45	3.814	0.45	4.002	0.45	4.667	0.45	4.603	0.9	3.579	0.9	4.782	0.9	4.182	0.9	4.532
0.5	4.573	0.5	4.695	0.5	4.396	0.5	3.451	0.5	4.412	1	4.631	1	3.724	1	4.227	1	3.922
0.55	4.500	0.55	3.482	0.55	4.426	0.55	4.130	0.55	4.530	1.1	4.055	1.1	3.786	1.1	3.953	1.1	3.634
0.6	3.536	0.6	4.520	0.6	4.072	0.6	4.567	0.6	3.740	1.2	3.862	1.2	3.517	1.2	4.414	1.2	4.213
0.65	4.251	0.65	4.487	0.65	4.009	0.65	4.371	0.65	3.604	1.3	4.422	1.3	4.170	1.3	4.242	1.3	4.144
0.7	3.694	0.7	4.221	0.7	3.783	0.7	4.096	0.7	4.579	1.4	3.473	1.4	4.153	1.4	4.027	1.4	4.322
0.75	3.773	0.75	4.059	0.75	4.200	0.75	3.600	0.75	3.790	1.5	4.252	1.5	3.661	1.5	3.890	1.5	3.914
0.8	4.773	0.8	4.186	0.8	4.666	0.8	3.830	0.8	4.656	1.6	4.617	1.6	4.042	1.6	3.783	1.6	4.412
0.85	4.031	0.85	3.547	0.85	4.627	0.85	4.614	0.85	4.031	1.7	3.574	1.7	4.731	1.7	4.780	1.7	4.578
0.9	3.535	0.9	3.993	0.9	3.787	0.9	4.474	0.9	3.634	1.8	3.644	1.8	3.513	1.8	4.620	1.8	3.625
0.95	3.650	0.95	4.325	0.95	3.881	0.95	4.253	0.95	4.748	1.9	3.994	1.9	3.874	1.9	4.594	1.9	4.574
1	3.688	1	4.375	1	4.367	1	3.562	1	4.727	2	3.837	2	4.165	2	4.465	2	4.052

Table 36
Minimum Fitness results for different values of the GGO's parameters.

r_1		r_2		r_3		r_4		r_5		w_1		w_2		w_3		w_4	
Values	Fitness	Values	Fitness	Values	Fitness	Values	Fitness	Values	Fitness	Values	Fitness	Values	Fitness	Values	Fitness	Values	Fitness
0.05	−11.2816	0.05	−10.777353	0.05	−10.2691	0.05	−9.9363	0.05	−10.2177	0.1	−10.3488	0.1	−10.5925	0.1	−9.59121	0.1	−9.68047
0.1	−11.816	0.1	−10.5296889	0.1	−9.73106	0.1	−10.1375	0.1	−9.97621	0.2	−10.9746	0.2	−10.7414	0.2	−11.0389	0.2	−10.5553
0.15	−11.2816	0.15	−10.4610467	0.15	−10.4932	0.15	−11.2277	0.15	−11.078	0.3	−9.55364	0.3	−9.61239	0.3	−11.0654	0.3	−10.8075
0.2	−10.2816	0.2	−9.63836813	0.2	−10.96	0.2	−10.465	0.2	−11.2499	0.4	−9.5732	0.4	−9.71374	0.4	−9.93474	0.4	−10.7043
0.25	−11.2816	0.25	−9.71200394	0.25	−10.1248	0.25	−9.80583	0.25	−10.1043	0.5	−10.5297	0.5	−10.2183	0.5	−9.85003	0.5	−10.0198
0.3	−11.2816	0.3	−10.54075	0.3	−9.62081	0.3	−10.8017	0.3	−11.1068	0.6	−9.90372	0.6	−9.48884	0.6	−10.1777	0.6	−10.7317
0.35	−11.2816	0.35	−9.85847991	0.35	−9.79173	0.35	−11.208	0.35	−11.2655	0.7	−11.0825	0.7	−10.398	0.7	−10.2295	0.7	−10.5532
0.4	−13.2816	0.4	−9.53153329	0.4	−10.0622	0.4	−9.99414	0.4	−9.49076	0.8	−10.362	0.8	−9.53441	0.8	−11.0516	0.8	−9.49902
0.45	−9.2816	0.45	−9.5246045	0.45	−10.0081	0.45	−9.44827	0.45	−9.53598	0.9	−10.6153	0.9	−10.2023	0.9	−11.2192	0.9	−10.5475
0.5	−11.2816	0.5	−10.8860063	0.5	−9.78342	0.5	−10.5485	0.5	−10.1314	1	−11.1659	1	−10.5912	1	−10.2532	1	−9.68924
0.55	−11.2816	0.55	−10.7439292	0.55	−9.82111	0.55	−10.8308	0.55	−10.4533	1.1	−10.4402	1.1	−10.0528	1.1	−11.037	1.1	−9.56651
0.6	−11.2816	0.6	−10.2426688	0.6	−11.2109	0.6	−10.3023	0.6	−10.3761	1.2	−9.54502	1.2	−9.86765	1.2	−9.95795	1.2	−9.7716
0.65	−11.2816	0.65	−10.483995	0.65	−10.7463	0.65	−10.6116	0.65	−10.6268	1.3	−9.84044	1.3	−10.9242	1.3	−11.02	1.3	−9.8236
0.7	−11.2816	0.7	−11.2024667	0.7	−9.73631	0.7	−10.989	0.7	−10.8216	1.4	−10.5202	1.4	−10.3774	1.4	−9.53366	1.4	−10.2886
0.75	−12.2816	0.75	−11.1699033	0.75	−10.4729	0.75	−10.4892	0.75	−10.5284	1.5	−9.66029	1.5	−10.7521	1.5	−10.6392	1.5	−10.624
0.8	−11.2816	0.8	−10.9313424	0.8	−10.7352	0.8	−10.8432	0.8	−10.6266	1.6	−10.4613	1.6	−10.7056	1.6	−9.88858	1.6	−9.82879
0.85	−13.6	0.85	−10.4883524	0.85	−9.74182	0.85	−10.3025	0.85	−9.45778	1.7	−10.0161	1.7	−10.5631	1.7	−9.96181	1.7	−10.8901
0.9	−13.816	0.9	−10.5306551	0.9	−10.7456	0.9	−10.9677	0.9	−9.72845	1.8	−9.80939	1.8	−11.058	1.8	−10.6786	1.8	−9.63846
0.95	−14.116	0.95	−9.91673761	0.95	−10.8103	0.95	−10.5016	0.95	−9.63262	1.9	−10.5553	1.9	−10.0753	1.9	−10.346	1.9	−10.7493
1	−14.8816	1	−11.2099791	1	−11.2327	1	−9.89203	1	−10.8742	2	−10.9657	2	−10.4337	2	−11.1176	2	−10.0681

Table 37
The findings of the ANOVA test for the parameters of the GGO's convergence time.

	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	0.5288	8	0.0661	F (8, 171) = 0.4635	P=0.008804
Residual (within columns)	24.39	171	0.1426	–	–
Total	24.91	179	–	–	–

Table 38
The outcomes of the ANOVA tests pertaining to the GGO's Minimum Fitness parameters.

	SS	DF	MS	F (DFn, DFd)	P value
Treatment (between columns)	41.43	8	5.179	F (8, 171) = 11.18	P<0.0001
Residual (within columns)	79.2	171	0.4631	–	–
Total	120.6	179	–	–	–

Table 39
At a significance threshold of 0.05, a T-test with one tail was carried out to compare the various values of the GGO's convergence time parameters.

	r_1	r_2	r_3	r_4	r_5	w_1	w_2	w_3	w_4
Theoretical mean	0	0	0	0	0	0	0	0	0
Actual mean	4.017	4.105	4.161	4.099	4.168	4.012	4.111	4.131	4.152
Number of values	20	20	20	20	20	20	20	20	20
One sample t-test									
t, df	t=43.27, df=19	t=48.43, df=19	t=61.19, df=19	t=43.45, df=19	t=45.03, df=19	t=48.65, df=19	t=48.09, df=19	t=50.08, df=19	t=56.69, df=19
P value (two-tailed)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
P value summary	****	****	****	****	****	****	****	****	****
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?									
Discrepancy	4.017	4.105	4.161	4.099	4.168	4.012	4.111	4.131	4.152
SD of discrepancy	0.4152	0.3791	0.3041	0.4219	0.414	0.3688	0.3823	0.3688	0.3275
SEM of discrepancy	0.09283	0.08476	0.068	0.09434	0.09258	0.08248	0.08549	0.08247	0.07324
95% confidence interval	3.823 to 4.211	3.927 to 4.282	4.019 to 4.303	3.901 to 4.296	3.975 to 4.362	3.840 to 4.185	3.932 to 4.290	3.958 to 4.303	3.998 to 4.305
R squared (partial eta squared)	0.99	0.992	0.995	0.99	0.9907	0.992	0.9919	0.9925	0.9941

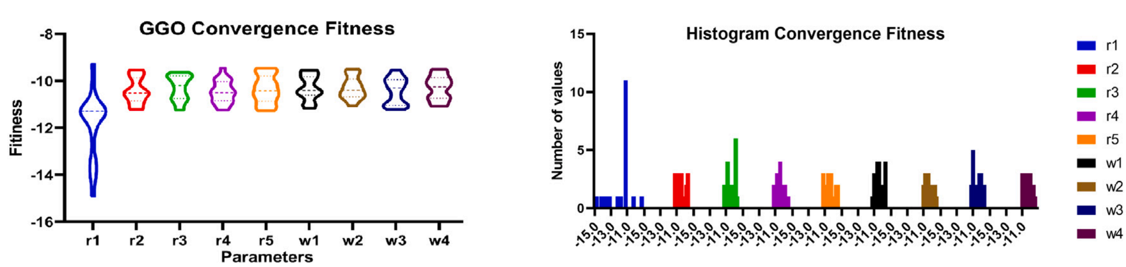

Fig. 13. Convergence fitness of f_{22}.

Table 40
At a significance threshold of 0.05, a T-test with one tail was carried out to compare the various values of the GGO's Minimum Fitness parameters.

	r_1	r_2	r_3	r_4	r_5	w_1	w_2	w_3	w_4
Theoretical mean	0	0	0	0	0	0	0	0	0
Actual mean	−11.87	−10.42	−10.3	−10.47	−10.36	−10.3	−10.3	−10.43	−10.3
Number of values	20	20	20	20	20	20	20	20	20
One sample t-test									
t, df	t=38.53, df=19	t=83.88, df=19	t=86.51, df=19	t=95.75, df=19	t=77.46, df=19	t=86.92, df=19	t=98.21, df=19	t=82.69, df=19	t=90.14, df=19
P value (two-tailed)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
P value summary	****	****	****	****	****	****	****	****	****
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?									
Discrepancy	−11.87	−10.42	−10.3	−10.47	−10.36	−10.3	−10.3	−10.43	−10.3
SD of discrepancy	1.378	0.5555	0.5327	0.4888	0.5984	0.5297	0.4688	0.564	0.5109
SEM of discrepancy	0.3082	0.1242	0.1191	0.1093	0.1338	0.1185	0.1048	0.1261	0.1142
95% confidence interval	−12.52 to −11.23	−10.68 to −10.16	−10.55 to −10.06	−10.69 to −10.24	−10.64 to −10.08	−10.54 to −10.05	−10.51 to −10.08	−10.69 to −10.17	−10.54 to −10.06
R squared (partial eta squared)	0.9874	0.9973	0.9975	0.9979	0.9968	0.9975	0.998	0.9972	0.9977

References

Abd Elaziz, M., Oliva, D., & Xiong, S. (2017). An improved opposition-based sine cosine algorithm for global optimization. *Expert Systems with Applications*, *90*, 484–500.

Abdel-Basset, M., El-Shahat, D., El-henawy, I., de Albuquerque, V. H. C., & Mirjalili, S. (2020). A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Systems with Applications*, *139*, Article 112824.

Abdollahzadeh, B., Gharehchopogh, F. S., Khodadadi, N., & Mirjalili, S. (2022). Mountain gazelle optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *Advances in Engineering Software*, *174*, Article 103282.

Al-Tashi, Q., Abdul Kadir, S. J., Rais, H. M., Mirjalili, S., & Alhussian, H. (2019). Binary optimization using hybrid grey wolf optimization for feature selection. *IEEE Access*, *7*, 39496–39508. http://dx.doi.org/10.1109/ACCESS.2019.2906757.

Al-Tashi, Q., Mirjalili, S., Wu, J., Abdulkadir, S. J., Shami, T. M., Khodadadi, N., et al. (2022). Moth-flame optimization algorithm for feature selection: A review and future trends. *Handbook of Moth-Flame Optimization Algorithm*, 11–34.

Arrif, T., Hassani, S., Guermoui, M., Sánchez-González, A., Taylor, R. A., & Belaid, A. (2022). GA-GOA hybrid algorithm and comparative study of different metaheuristic population-based algorithms for solar tower heliostat field design. *Renewable Energy*, *192*, 745–758.

Awad, A. A., Ali, A. F., Gaber, T., et al. (2020). Feature selection method based on chaotic maps and butterfly optimization algorithm. In *AICV* (pp. 159–169).

Bello, R., Gomez, Y., Nowe, A., & Garcia, M. M. (2007). Two-step particle swarm optimization to solve the feature selection problem. In *Seventh international conference on intelligent systems design and applications* (pp. 691–696). http://dx.doi.org/10.1109/ISDA.2007.101.

Bennasar, M., Hicks, Y., & Setchi, R. (2015). Feature selection using joint mutual information maximisation. *Expert Systems with Applications*, *42*(22), 8520–8532.

Birdfact (2022). What is a group of geese called? https://birdfact.com/articles/what-is-a-group-of-geese-called. (Accessed 26 July 2022).

Brownlee, J. (2011). *Clever algorithms: Nature-inspired programming recipes*. Jason Brownlee.

Celik, Y., & Kutucu, H. (2018). Solving the tension/compression spring design problem by an improved firefly algorithm. In *IDDM*.

Chandrasekaran, K., & Simon, S. P. (2012). Multi-objective scheduling problem: hybrid approach using fuzzy assisted cuckoo search algorithm. *Swarm and Evolutionary Computation*, *5*, 1–16.

Confalonieri, R., Bellocchi, G., Bregaglio, S., Donatelli, M., & Acutis, M. (2010). Comparison of sensitivity analysis techniques: A case study with the rice model WARM. *Ecological Modelling*, *221*(16), 1897–1906. http://dx.doi.org/10.1016/j.ecolmodel.2010.04.021, URL: https://www.sciencedirect.com/science/article/pii/S0304380010002371.

Cui, Y., Hu, W., & Rahmani, A. (2022). Improved artificial bee colony algorithm with dynamic population composition for optimization problems. *Nonlinear Dynamics*, *107*(1), 743–760.

Deng, W., Xu, J., & Zhao, H. (2019). An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. *IEEE Access*, *7*, 20281–20292.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, *1*(4), 28–39.

Eberhart, R., & Kennedy, J. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks, vol. 4* (pp. 1942–1948).

El-kenawy, E.-S. M., Abdelhamid, A. A., Ibrahim, A., Mirjalili, S., Khodadad, N., duailij, M. A. A., et al. (2023). Al-biruni earth radius (BER) metaheuristic search optimization algorithm. *Computer Systems Science and Engineering*, *45*(2), 1917–1934. http://dx.doi.org/10.32604/csse.2023.032497.

El-kenawy, E.-S. M., Albalawi, F., Ward, S. A., Ghoneim, S. S. M., Eid, M. M., Abdelhamid, A. A., et al. (2022). Feature selection and classification of transformer faults based on novel meta-heuristic algorithm. *Mathematics*, *10*(17), 3144. http://dx.doi.org/10.3390/math10173144.

El-Kenawy, E.-S., & Eid, M. (2020). Hybrid gray wolf and particle swarm optimization for feature selection. *International Journal Innovation Information Control*, *16*(3), 831–844.

El-Kenawy, E.-S. M., Eid, M. M., Saber, M., & Ibrahim, A. (2020). MbGWO-SFS: Modified binary grey wolf optimizer based on stochastic fractal search for feature selection. *IEEE Access*, *8*, 107635–107649. http://dx.doi.org/10.1109/access.2020.3001151.

El-kenawy, E.-S. M., Ibrahim, A., Mirjalili, S., Eid, M. M., & Hussein, S. E. (2020). Novel feature selection and voting classifier algorithms for COVID-19 classification in CT images. *IEEE Access*, *8*, 179317–179335. http://dx.doi.org/10.1109/access.2020.3028012.

Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, *172*, 371–381.

Fister, I., Yang, X.-S., Fister, I., & Brest, J. (2012). Memetic firefly algorithm for combinatorial optimization. *Technical report*, arXiv:1204.5165 URL: https://cds.cern.ch/record/1443422, Comments: 14 pages; Bioinspired Optimization Methods and their Applications (BIOMA 2012).

Fouad, M. M., El-Desouky, A. I., Al-Hajj, R., & El-Kenawy, E.-S. M. (2020). Dynamic group-based cooperative optimization algorithm. *IEEE Access*, *8*, 148378–148403. http://dx.doi.org/10.1109/access.2020.3015892.

Gandomi, A. H., & Alavi, A. H. (2011). Multi-stage genetic programming: a new strategy to nonlinear system modeling. *Information Sciences*, *181*(23), 5227–5239.

Green, M. (2004). Flying with the wind - Spring migration of arctic-breeding waders and geese over South Sweden. *Ardea*, *92*(2), 145–159.

Gu, S., Cheng, R., & Jin, Y. (2018). Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Computing*, *22*(3), 811–822.

Hafez, A. I., Zawbaa, H. M., Emary, E., Mahmoud, H. A., & Hassanien, A. E. (2015). An innovative approach for feature selection based on chicken swarm optimization. In *2015 7th international conference of soft computing and pattern recognition* (pp. 19–24). IEEE.

He, S., Zhu, L., Wang, L., Yu, L., & Yao, C. (2019). A modified gravitational search algorithm for function optimization. *IEEE Access*, *7*, 5984–5993. http://dx.doi.org/10.1109/access.2018.2889854.

Horton, J. (2008). Why do geese mate for life? https://animals.howstuffworks.com/birds/geese-mate-for-life.htm. (Accessed 26 July 2022).

Ibrahim, A., Ali, H. A., Eid, M. M., & El-kenawy, E.-S. M. (2020). Chaotic harris hawks optimization for unconstrained function optimization. In *2020 16th international computer engineering conference* (pp. 153–158). IEEE.

Indiana DNR (2022). Canada geese behavior & biology. https://www.in.gov/dnr/fish-and-wildlife/hunting-and-trapping/canada-geese-management/behavior-and-biology/. (Accessed 26 July 2022).

Jianhua, L., & Zhiheng, W. (2021). A hybrid sparrow search algorithm based on constructing similarity. *IEEE Access*, *9*, 117581–117595. http://dx.doi.org/10.1109/access.2021.3106269.

Kabir, M. M., Shahjahan, M., & Murase, K. (2011). A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing*, *74*(17), 2914–2928. http://dx.doi.org/10.1016/j.neucom.2011.03.034, URL: http://www.sciencedirect.com/science/article/pii/S0925231211002748.

Karaboga, D. (2010). Artificial bee colony algorithm. *Scholarpedia*, *5*(3), 6915.

Kaveh, A., Eslamlou, A. D., & Khodadadi, N. (2020). Dynamic water strider algorithm for optimal design of skeletal structures. *Periodica Polytechnica Civil Engineering*, *64*(3), 904–916.

Kaveh, A., Khodadadi, N., Azar, B. F., & Talatahari, S. (2021). Optimal design of large-scale frames with an advanced charged system search algorithm using box-shaped sections. *Engineering with Computers*, *37*(4), 2521–2541.

Kaveh, A., & Talatahari, S. (2010). An improved ant colony optimization for constrained engineering design problems. *Engineering Computations*.

Kaveh, A., Talatahari, S., & Khodadadi, N. (2020a). Hybrid invasive weed optimization-shuffled frog-leaping algorithm for optimal design of truss structures. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, *44*(2), 405–420.

Kaveh, A., Talatahari, S., & Khodadadi, N. (2020b). Stochastic paint optimizer: theory and application in civil engineering. *Engineering with Computers*, 1–32.

Khodadadi, N., Snasel, V., & Mirjalili, S. (2022). Dynamic arithmetic optimization algorithm for truss optimization under natural frequency constraints. *IEEE Access*, *10*, 16188–16208.

Khodadadi, N., Soleimanian Gharehchopogh, F., & Mirjalili, S. (2022). MOAVOA: a new multi-objective artificial vultures optimization algorithm. *Neural Computing and Applications*, 1–39.

Khodadadi, N., Talatahari, S., & Dadras Eslamlou, A. (2022). MOTEO: a novel multi-objective thermal exchange optimization algorithm for engineering problems. *Soft Computing*, 1–26.

Mendil, B., & Benmahammed, K. (1999). FEP learning algorithm: application to direct self-learning control. In *Proceedings of the 1999 IEEE international conference on control applications (Cat. no.99CH36328)*. IEEE, http://dx.doi.org/10.1109/cca.1999.806674.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51–67. http://dx.doi.org/10.1016/j.advengsoft.2016.01.008, URL: http://www.sciencedirect.com/science/article/pii/S0965997816300163.

Mirjalili, S., Mirjalili, S. M., & Hatamlou, A. (2016). Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Computing and Applications*, *27*(2), 495–513. http://dx.doi.org/10.1007/s00521-015-1870-7.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, *69*, 46–61.

Moradi, P., & Gholampour, M. (2016). A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Applied Soft Computing*, *43*, 117–130.

Nematzadeh, H., Enayatifar, R., Mahmud, M., & Akbari, E. (2019). Frequency based feature selection method using whale algorithm. *Genomics*, *111*(6), 1946–1955.

Onay, F. K., & Aydemır, S. B. (2022). Chaotic hunger games search optimization algorithm for global optimization and engineering problems. *Mathematics and Computers in Simulation*, *192*, 514–536.

Oyelade, O. N., Ezugwu, A. E.-S., Mohamed, T. I., & Abualigah, L. (2022). Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm. *IEEE Access*, *10*, 16150–16177.

Pereira, J. L. J., Oliver, G. A., Francisco, M. B., Cunha, S. S., Jr., & Gomes, G. F. (2022). Multi-objective lichtenberg algorithm: A hybrid physics-based meta-heuristic for solving engineering problems. *Expert Systems with Applications*, *187*, Article 115939.

Qasim, O. S., Al-Thanoon, N. A., & Algamal, Z. Y. (2020). Feature selection based on chaotic binary black hole algorithm for data classification. *Chemometrics and Intelligent Laboratory Systems*, *204*, Article 104104.

Qasim, O. S., & Algamal, Z. Y. (2020). Feature selection using different transfer functions for binary bat algorithm. *International Journal of Mathematical, Engineering and Management Sciences*, *5*(4), 697–706.

Rashedi, E., Nezamabadi-pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248. http://dx.doi.org/10.1016/j.ins.2009.03.004.

Rostami, M., Forouzandeh, S., Berahmand, K., & Soltani, M. (2020). Integration of multi-objective PSO based feature selection and node centrality for medical datasets. *Genomics*, *112*(6), 4370–4384.

Salimi, H. (2015). Stochastic fractal search: A powerful metaheuristic algorithm. *Knowledge-Based Systems*, *75*, 1–18. http://dx.doi.org/10.1016/j.knosys.2014.07.025, URL: http://www.sciencedirect.com/science/article/pii/S0950705114002822.

Samareh Moosavi, S. H., & Khatibi Bardsiri, V. (2017). Satin bowerbird optimizer. *Engineering Applications of Artificial Intelligence*, *60*(C), 1–15. http://dx.doi.org/10.1016/j.engappai.2017.01.006.

Schiezaro, M., & Pedrini, H. (2013). Data feature selection based on artificial bee colony algorithm. *EURASIP Journal on Image and Video Processing*, *2013*(1), 1–8.

Şenel, F. A., Gokçe, F., Yuksel, A. S., & Yigit, T. (2019). A novel hybrid PSO–GWO algorithm for optimization problems. *Engineering with Computers*, *35*(4), 1359–1373. http://dx.doi.org/10.1007/s00366-018-0668-5.

Tu, Q., Chen, X., & Liu, X. (2019). Multi-strategy ensemble grey wolf optimizer and its application to feature selection. *Applied Soft Computing*, *76*, 16–30.

Valdez, F., Castillo, O., & Melin, P. (2021). Bio-inspired algorithms and its applications for optimization in fuzzy clustering. *Algorithms*, *14*(4), 122.

Xue, B., Zhang, M., & Browne, W. N. (2014). Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing*, *18*, 261–276.

Yang, X. (2008). Introduction to mathematical optimization. In *From linear programming to metaheuristics*. Cambridge international science publishing.

Yang, X.-S., Gandomi, A. H., Talatahari, S., & Alavi, A. H. (2012). *Metaheuristics in water, geotechnical and transport engineering*. Newnes.

Ye, T., Wang, W., Wang, H., Cui, Z., Wang, Y., Zhao, J., et al. (2022). Artificial bee colony algorithm with efficient search strategy based on random neighborhood structure. *Knowledge-Based Systems*, *241*, Article 108306.

Yıldız, A. R., et al. (2009). *An effective hybrid immune-hill climbing optimization approach for solving design and manufacturing optimization problems in industry*. Elsevier Science.

Zhao, W., Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, *388*, Article 114194.

Zhao, W., Zhang, Z., Mirjalili, S., Wang, L., Khodadadi, N., & Mirjalili, S. M. (2022). An effective multi-objective artificial hummingbird algorithm with dynamic elimination-based crowding distance for solving engineering design problems. *Computer Methods in Applied Mechanics and Engineering*, *398*, Article 115223.

Zou, D., Liu, H., Gao, L., & Li, S. (2011). A novel modified differential evolution algorithm for constrained optimization problems. *Computers & Mathematics with Applications*, *61*(6), 1608–1623. http://dx.doi.org/10.1016/j.camwa.2011.01.029, URL: http://www.sciencedirect.com/science/article/pii/S0898122111000460.