

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

# Giant Trevally Optimizer (GTO): A Novel Metaheuristic Algorithm for Global Optimization and Challenging Engineering Problems

Haval Tariq Sadeeq<sup>1</sup>, and Adnan Mohsin Abdulazeez<sup>2</sup>

<sup>1</sup>Information Technology Department, Technical College of Informatics-Akre, Duhok Polytechnic University, Duhok, Iraq

<sup>2</sup>Energy Department, Technical College of Engineering-Duhok, Duhok Polytechnic University, Duhok, Iraq

Corresponding author: Haval Tariq Sadeeq (e-mail: haval.tariq@dpu.edu.krd).

**ABSTRACT** Metaheuristic algorithms are becoming powerful methods for solving continuous global optimization and engineering problems due to their flexible implementation on the given problem. Most of these algorithms draw their inspiration from the collective intelligence and hunting behavior of animals in nature. This paper proposes a novel metaheuristic algorithm called the Giant Trevally Optimizer (GTO). In nature, giant trevally feeds on many animals, including fish, cephalopods, and seabirds (sooty terns). In this work, the unique strategies of giant trevally when hunting seabirds are mathematically modeled and are divided into three main steps. In the first step, the foraging movement patterns of giant trevallies are simulated. In the second step, the giant trevallies choose the appropriate area in terms of food where they can hunt for prey. In the last step, the trevally starts to chase the seabird (prey). When the prey is close enough to the trevally, the trevally jumps out of the water and attacks the prey in the air or even snatches the prey from the water surface. The performance of GTO is compared against state-of-the-art metaheuristics for global optimization on a set of forty benchmark functions with different characteristics and five complex engineering problems. The comparative study, scalability analysis, statistical analysis based on the Wilcoxon rank sum test, and the findings suggest that the proposed GTO is an efficient optimizer for global optimization. Note that the MATLAB source codes for GTO will be publicly available after the acceptance of the paper.

**INDEX TERMS** Giant Trevally Optimizer (GTO), optimization, metaheuristics, exploration, exploitation, benchmark functions.

## I. INTRODUCTION

The aim of optimization is to define the best possible solution for the system parameters so that the design can be completed at the lowest possible cost. Real-world optimization tasks tend to be discrete or unrestrained by any particular constraints [1][2]. Consequently, it is difficult to obtain optimal solutions using conventional mathematical-based programming methods. Hence, optimization techniques have been developed in recent years, which can be found in almost all scientific domains to promote the performance of various systems and minimize their computational costs [3]. There are some drawbacks and limits to conventional optimization approaches, such as unknown solution space and a potential to become stuck at a local optimum. And indeed, they also have only a single solution [4].

There have been numerous novel Metaheuristic Algorithms (MAs) proposed in recent years to resolve these

concerns. These algorithms have been used in a number of contexts, which is because of how simple they are and how easy it is to use them [5][6]. Also, the main parts of these methods do not depend on gradient information or the mathematical properties of the objective landscape [7]. However, the problem with most MAs, though, is that they are often very sensitive to the way that user-defined parameters are tuned. Another drawback is that there is no guarantee of finding the global best solution due to its stochastic nature [8].

For the most part, MAs fall into one of two categories: single-solution based, and population-based. In the first category, an MA may employ a single solution or agent to perform the optimization process. This is referred to as trajectory methods. Tabu Search [9] and Simulated Annealing [10] are two examples of these methods. On the other hand, the majority of modern MAs are population-based or multi-agents, as they traverse search space using a set of points or

individuals. The Firefly algorithm [11], cuttlefish optimization algorithm [12], and Lion Optimization Algorithm [13] are examples of this approach. This strategy is appropriate for global searching since it allows for both global exploration and local exploitation until stopping criteria are met.

No matter how different MAs are, they all have one common trait: the searching steps have two phases: exploration (diversification) and exploitation (intensification) [14]. During the exploration phase, the algorithm should maximize and promote the use of its randomized operators in order to exhaustively investigate various regions of the search space. As a result, during the initial stages of the searching process, the exploratory behaviors of a well-designed algorithm should have an enriched-enough random character to effectively distribute more randomly-generated solutions to diverse areas of the problem topography [15]. The exploitation phase typically follows the exploration phase. The algorithm makes an effort to concentrate on the neighborhood of higher-quality solutions inside the feature space during this phase. An efficient algorithm ought to be able to make a decent balance between exploration and exploitation. Consequently, the chance of becoming locked in local optima and the disadvantages of immature convergence rises [16].

As so many MAs have already been developed, the main question is whether or not there is still a need for more. The answer to this critical question is the No Free Lunch (NFL) theorem [17]. When it comes to solving optimization problems, some algorithms are more effective than others. The NFL theorem explains this fact because each real-world problem has its own unique characteristics and mathematical model. Therefore, there is no guarantee that a certain MA will solve all optimization problems in an efficient manner.

Another important concern in the procedure for optimality-seeking is the randomness facility of the search space, which may not always produce adequate optimal solution. As a result, many MAs have been developed by researchers to provide acceptable optimal solutions, or at least as optimal as possible. Based on the foregoing, the authors of this study were inspired to propose a novel optimization method that can yield satisfactory results for a wide variety of optimization tasks.

The novelty and contribution of this research is in the design of a new MA called the Giant Trevally Optimizer (GTO), which is based on the behavior and strategies of giant trevallies during hunting seabirds. Each step of the proposed GTO is outlined, and a mathematical model is provided. Forty objective functions of unimodal and multimodal types with different characteristics have been utilized to evaluate the effectiveness of the proposed GTO in optimization. Furthermore, GTO is applied to five complex engineering problems. Finally, the GTO's performance is compared with other well-known optimization algorithms: Differential Evolution (DE) [18], Gravitational Search Algorithm (GSA) [19], Gray Wolf Optimization (GWO) [20], Moth Flame Optimization (MFO) [21], Particle Swarm Optimization

(PSO) [22], and Whale Optimization Algorithm (WOA) [23].

The paper is organized as follows: section 2 describes the related works; section 3 explains the behavior of giant trevally with the proposed algorithm; section 4 presents the proposed flow chart and the pseudo code; and section 5 presents the results. Finally, section 6 discusses the main conclusions and findings. [24].

## II. RELATED WORKS

In general, metaheuristics can be classified into four different categories:

### A. EVOLUTIONARY ALGORITHMS (EA)

EAs are based on the principles of species evolution theory. The Genetic Algorithm (GA) is one of numerous EAs that fall within this group [24]. The concept of GA stems from Darwin's idea of natural selection. The main components of this algorithm are: selection, crossover, and mutation, which are used to produce new generations. Differential Evolution (DE) [18] is another method inspired by natural evolution. The DE algorithm consists of four basic steps: random initialization of the population, mutation, recombination, and finally selection. The main difference between the GA and the DE algorithm is in the selection process for generating the next generation.

Inspired by the geographical dispersal of species, including patterns of movement and extinction, based on this occurrence, [25] came up with the Biogeography-based optimizer (BBO) algorithm, which is a population-based metaheuristic for global optimization. The Black Widow Optimization Algorithm (BWO) [26] is another evolutionary algorithm inspired by the evolution process of a spider population. Cannibalism is an essential part of this approach. Convergence occurs early in this stage because species with poor fitness are excluded.

### B. SWARM INTELLIGENCE (SI) ALGORITHMS

SI refers to developing algorithms that are inspired by the collective behavior of diverse animal species. The most well-known of these algorithms is Particle Swarm Optimization (PSO) [22]. The individual search agent is called a particle. Each particle has a velocity and a position vector allocated to it based on its social and individual experience.

The foraging activity of real ants acts as the inspiration for Ant Colony Optimization (ACO) [27]. Ants conduct a random search for food in the direct proximity of their nest. Ants carry some of their food back to the nest as soon as they identify a food source that meets their needs. The ant leaves behind a chemical pheromone trail as it makes its way back. Depending on the amount and quality of food, the amount of pheromone deposited will lead other ants to the food source.

ACO's inspiration comes from real ant foraging activity. Ants begin their food search by randomly scanning the area around their nest. As soon as an ant finds a food source, it examines the quantity and quality of the food and takes some

of it back to the nest. A chemical pheromone trail is left behind by the ant as it makes its way back. Depending on the amount and quality of food, the amount of pheromone deposited will guide other ants to the meal. The GTO that is proposed in this paper falls into this category, which mimics the hunting strategies of giant trevally marine fish.

### C. HUMAN-BASED-ALGORITHMS (HA)

This category includes algorithms that are based on human behavior. Walking, talking, and others, as well as mental processes, are all incorporated into the algorithms. An example of this class of algorithms is the Gaining Sharing Knowledge based algorithm (GSK) [28], which mimics the process of gaining and sharing knowledge during the human life span. To accomplish optimization, GSK uses two mathematical models: a junior gaining and sharing phase and a senior gaining and sharing phase. Another important algorithm for this category is Teaching–Learning–Based Optimization (TLBO) [29]. TLBO takes its inspiration from the natural teaching–learning phenomenon of a classroom and is divided into two parts. The first part consists of the ‘Teacher Phase’ and the second part consists of the ‘Learner Phase’.

### D. SCIENCE-BASED-ALGORITHMS (SCA)

Modeling physical occurrences or chemical rules is the focus of science-based algorithms (e.g., ion motion, gravity, electrical charges, etc.). Simulated annealing (SA) [10], Charged System Search (CSS) [30], gravitational search algorithm (GSA) [19], Galaxy-based Search Algorithm (GbSA) [31], heat transfer search [32], Curved Space Optimization [33], Gases Brownian motion optimization [34], and Central Force Optimization (CFO) [35] are regarded as the most popular SCAs. It's worth noting here, that Table 1 summarizes several recent MAs.

## III. GIANT TREVALLY OPTIMIZER (GTO)

This section provides a description of the proposed MA, which derives its inspiration from nature and is called the Giant Trevally Optimizer (GTO).

### A. INSPIRATION AND BEHAVIOR OF GIANT TREVALLY DURING HUNTING:

The giant trevally (*Caranx ignobilis*) is a large marine predator in the jack family. It is also called the giant kingfish. They are abundant in the Indian and Pacific oceans, such as areas around Australia and New Zealand. They are also found off the East Africa and around the Hawaiian Islands [36].

Giant trevally is usually silver with some dark spots. It can be recognized by its sharp head, strong tail scutes, and numerous additional anatomical details. Their height can reach up to 170 cm and 80 kg of weight. Their daily diet consists of fish, cephalopods, crustaceans, and birds [37].

Literature investigated the movement of giant trevallies within their ecosystems and between habitats as the search space expands. Some data suggests that adult giant trevallies make daily and seasonal movements of up to 9 kilometers

within their roaming range [38]. Juveniles can migrate up to 70 kilometers from their home atolls and reefs [39].

In most of its habitats, the giant trevally is a top predator and uses intelligent ways to hunt. The Giant trevally is known to hunt alone and in groups (schools). According to [40], grouped (schooled) predators are most effective at capturing schooled prey. The most effective member of a group or school at capturing prey is the leader, or first predator.

During the dry season, over half a million terns crowd onto one of the remote atolls in the Indian Ocean. It was reported that about fifty giant trevallies come from neighboring reefs, attracted by this abundance of potential prey where the juvenile terns start learning to fly. After specifying the hunting area, the giant trevally starts to stalk (chase) its prey, then jumps out of the water and catches the prey (seabird).

These novel hunting strategies of foraging moving patterns, choosing the appropriate area in term of quantity of food and jumping out of water to attack and catch the prey where the main inspiration in the design of the GTO.

### B. INITIALIZATION

Similar to other population-based MAs, GTO starts the optimization process by randomly generating initialization solutions called giant trevallies. In GTO, each giant trevally is a feasible or a candidate solution to the optimization problem. From a mathematical perspective, each member of the population is a vector, and these vectors constitute the population matrix of the algorithm. The GTO population members are modeled according to (1):

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N \times Dim} = \begin{bmatrix} x_{1,1} & \dots & x_{1,j} & \dots & x_{1,Dim} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \dots & x_{i,j} & \dots & x_{i,Dim} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \dots & x_{N,j} & \dots & x_{N,Dim} \end{bmatrix}_{N \times Dim} \quad (1)$$

Where,  $X$  is the candidate solution of GTO,  $X_i$  is the  $i^{th}$  candidate solution of GTO,  $Dim$  is the number of decision variables of given problem,  $N$  is the number of GTO members,  $x_{i,j}$  is the value of the  $j^{th}$  variable specified by the  $i^{th}$  candidate solution.

Once the number of population and number of dimensions have been selected, they will remain the same for the duration of the experiment. It is necessary to randomly assign positions to each trevally in the problem's solution space before they can begin to function. This random assignment must cover all feasible regions in the  $N \times Dim$  search space, as shown in the following equation:

$$X_{i,j} = Minimum_j + (Maximum_j - Minimum_j) \times R \quad (2)$$

Where  $i = 1, 2, \dots, N$  and  $j = 1, 2, \dots, Dim$ ,  $R$  is a random number in the interval  $[0, 1]$ .  $Maximum_j$ ,  $Minimum_j$  represent the restrictions on the defined problem for the  $j^{th}$  dimension i.e., the maximum and minimum value that a population member can have.

As previously stated, each population member in the suggested GTO is a candidate solution to the presented

problem. As a result, the objective function of the given problem can be assessed using each of the candidate solutions. According to (3), a vector is used to represent the set of these values:

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N \times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N \times 1} \quad (3)$$

Where  $F_i$  denotes the  $i^{th}$  member's value of the objective function, and  $F$  indicates the collection of these values as the objective function vector.

### C. MATHEMATICAL MODEL OF THE PROPOSED GTO

The proposed GTO algorithm mimics the behavior of giant trevallies during hunting seabirds. Consequently, the optimization procedures of the proposed GTO algorithm are represented in three steps: extensive search using Levy flight, choosing area step to determine the hunting area, and chasing and attacking the prey by jumping out of the water. Hence, the exploration phase of the GTO is represented in the first two steps, and the third one represents the exploitation phase of the GTO. The giant trevally when hunting in the nature is shown in Fig.1.



FIGURE 1. Giant trevally jumping out of the water and attacking the sooty tern.

#### 1) STEP 1: EXTENSIVE SEARCH

If we consider the nature of giant trevallies, and as mentioned earlier, giant trevallies can travel long distances for their daily diet. Hence, in this step, foraging movement patterns of giant trevallies are simulated using (4):

$$X(t+1) = Best_p \times R + ((Maximum - Minimum) \times R + Minimum) \times Levy(Dim) \quad (4)$$

where  $X(t+1)$  is the next-iteration giant trevally position vector,  $Best_p$  indicates the current search space chosen by giant trevallies based on the best position determined during their last search,  $R$  is a random number that takes a value

between 0 and 1.  $Levy(Dim)$  is the Levy flight, which is a special class of non-Gaussian stochastic process whose step sizes are determined by the so-called Levy distribution [41]. The ability of the algorithm to perform a global search is aided by the occasional large steps it takes. Furthermore, the main advantages of using Levy flight are the avoidance of local optima and the improvement of the convergence rate [42].

In this regard, it's worth mentioning here that many literatures have shown that the behavior of Levy flight is exhibited by a wide variety of animals, including marine predators [43] [44].  $Levy(Dim)$  be calculated using (5):

$$Levy(Dim) = step \times \frac{u \times \sigma}{|v|^{1/\beta}} \quad (5)$$

Where  $step$  is the step size and fixed to 0.01,  $\beta$  is the index of the Levy flight distribution function that can take values from 0 to 2 and has been set to 1.5 in this paper,  $u$  and  $v$  are random numbers normally distributed in the range (0,1).  $\sigma$  is calculated by using (6):

$$\sigma = \left( \frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{\frac{\beta-1}{2}}} \right) \quad (6)$$

#### 2) STEP 2: CHOOSING AREA

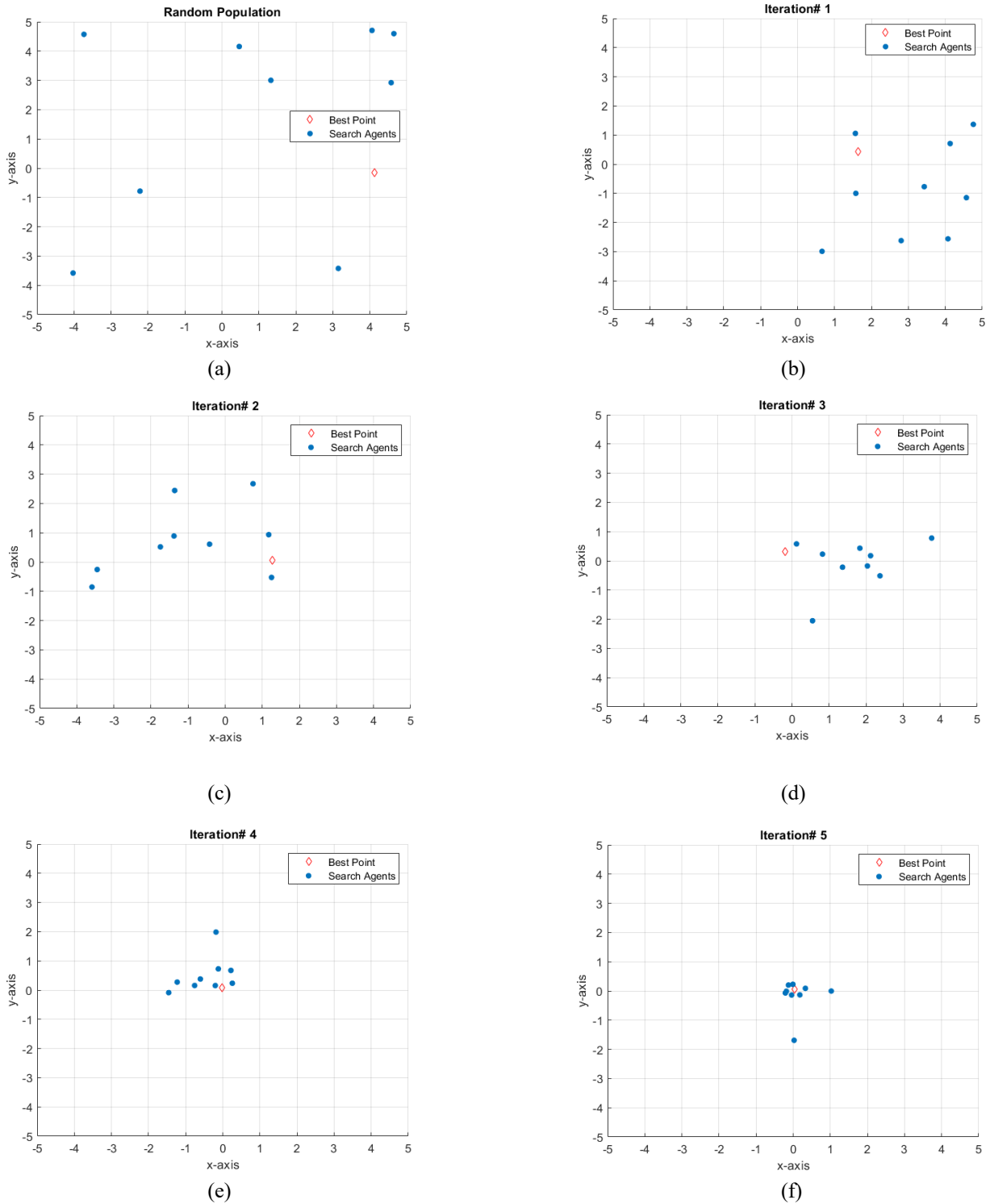
In the choosing area step, giant trevallies identify and select the best area in terms of the amount of food (seabirds) within the selected search space where they can hunt for prey. Equation (7) simulates this behavior mathematically.

$$X(t+1) = Best_p \times \mathcal{A} \times R + Mean\_Info - Xi(t) \times R \quad (7)$$

where  $X(t+1)$  is the position vector of giant trevallies in the next iteration  $t$ ,  $\mathcal{A}$  is a position-change-controlling parameter with a range from 0.3 to 0.4.  $Xi(t)$  is the location of the giant trevally  $i$ , at time  $t$  (current iteration). Meanwhile,  $Mean\_Info$  which refers to the mean, indicates that these giant trevallies have used up all the available information from the previous points and can be calculated using (8).

$$Mean\_Info = \frac{1}{N} \sum_{i=1}^N Xi(t) \quad (8)$$

The effectiveness of the choosing area step, i.e., (7), has been evaluated using the Sphere function with 10 solutions (search agents) and five iterations. Fig. 2 illustrates that using the best points and the mean as a basis for the choosing area step enhances the quality of all solutions. Fig. 2f shows that all the search agents are located near the best point.



**FIGURE 2.** a) Random population of 10 solutions. b) Choosing area after one iteration. c) Choosing area after two iterations. d) Choosing area after three iterations. e) Choosing area after four iterations. f) Choosing area after five iterations.



### 3) STEP 3: ATTACKING

In the previous step and after specifying the best area for hunting. In this step, which represents the exploitation (intensification) phase of the GTO, the trevally starts to chase the bird (prey). Here, and finally, the trevally attacks the bird when it gets close enough to the bird by making an acrobatic jump out of the water and catching the bird.

In order to simulate the behavior of giant trevally during chasing and attacking the prey, it was assumed in GTO that trevallies are affected by visual distortion, which is mainly caused by the refraction of light. Refraction of light is the deflection of the trajectory of a light wave as it traverses the interface between two media, such as water, glass, and air. As shown in Fig. 3, light from point A in the 1<sup>st</sup> medium enters the 2<sup>nd</sup> medium through the intersection point S, hence the refraction occurs and arrives point B at last. It should be mentioned here that when light moves from a rarer medium like air to a denser medium like water, it bends toward the normal as it enters the denser medium of water. According to Snell's law [45], both the incident ray and the refracted ray must form an angle with the normal to the surface at the point of refraction. The medium that the light rays are traveling through also plays a significant role. Snell's Law makes this connection clear with the use of refractive indices, which are fixed values for certain media.

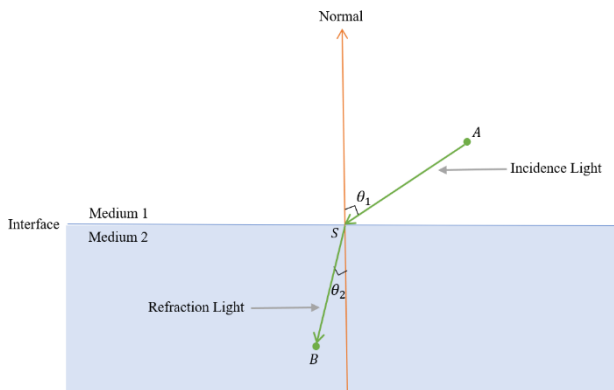


FIGURE 3. Principle of light refraction.

In GTO, as indicated in fig. 4, the bird is behaving as an object and the giant trevally is acting as an observer. The visual distortion is represented by the dashed line in Fig. 4, which indicates the apparent (false) height of the bird, which is always seen to be higher than it is actual height due to the refraction of the light.

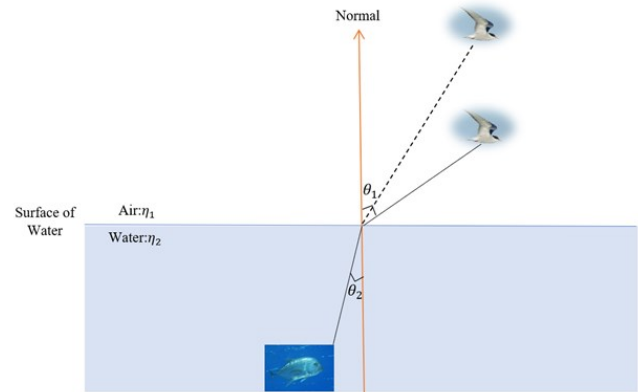


FIGURE 4. Visual Distortion in GTO.

Here, if we know the angle of incidence, it is possible to predict what the angle of refraction will be, and likewise, if we know the angle of refraction, it is possible to predict the angle of incidence. The Snell's law is demonstrated below in (9).

$$\eta_1 \sin \theta_1 = \eta_2 \sin \theta_2 \quad (9)$$

Where  $\eta_1 = 1.00029$  and  $\eta_2 = 1.33$  represents the absolute refractive index of air and water, respectively, whereas  $\theta_1$  and  $\theta_2$  represents the angle of incidence and angle of refraction respectively.  $\theta_2$  is a random number in the interval  $[0, 360]$ . From (9),  $\theta_1$  can be calculated using below (10):

$$\sin \theta_1 = \frac{\eta_2}{\eta_1} \sin \theta_2 \quad (10)$$

Then, the visual distortion  $\mathcal{V}$  can be calculated using (11):

$$\mathcal{V} = \sin(\theta_1^\circ) \times \mathcal{D} \quad (11)$$

Where  $\sin$  is the sine of variable in degrees,  $\mathcal{D}$  is the distance between the prey and the attacker, and can be calculated using (12):

$$\mathcal{D} = |(\text{Best}_p - Xi(t))| \quad (12)$$

Where  $\text{Best}_p$  is the best-obtained solution so far; it represents the location of the prey.

Then the behavior of giant trevally when chasing and jumping out of the water is mathematically simulated using (13).

$$X(t+1) = \mathcal{L} + \mathcal{V} + \mathcal{H} \quad (13)$$

Where  $X(t+1)$  is the solution of the next iteration of  $t$ , which is generated by the attacking step,  $\mathcal{L}$  represents the launch speed to simulate chasing the bird and can be calculated using (14):

$$\mathcal{L} = Xi(t) \times \sin(\theta_2^\circ) \times F_{obj}(Xi(t)) \quad (14)$$

Where  $F_{obj}(Xi(t))$  refer to the fitness value of  $X$  at the current iteration  $t$ .

The last term  $\mathcal{H}$  in (13) specifies the jumping slope function that enables the algorithm to adaptively perform an appropriate transition from the exploration phase to the exploitation phase and can be calculated using (15):

$$\mathcal{H} = R \times (2 - t \times \frac{2}{T}) \quad (15)$$

Where  $t$  and  $T$  refer to the current iteration and the maximum number of iterations respectively,  $R$  is a random number and here refer to different motion sense of the giant trevally during exploitation step.

#### IV. PROPOSED FLOW CHART AND PSEUDO-CODE OF GTO

In this section, the flowchart of GTO algorithm is illustrated in Fig. 5, Moreover, the pseudo-code of GTO is demonstrated in Algorithm 1.

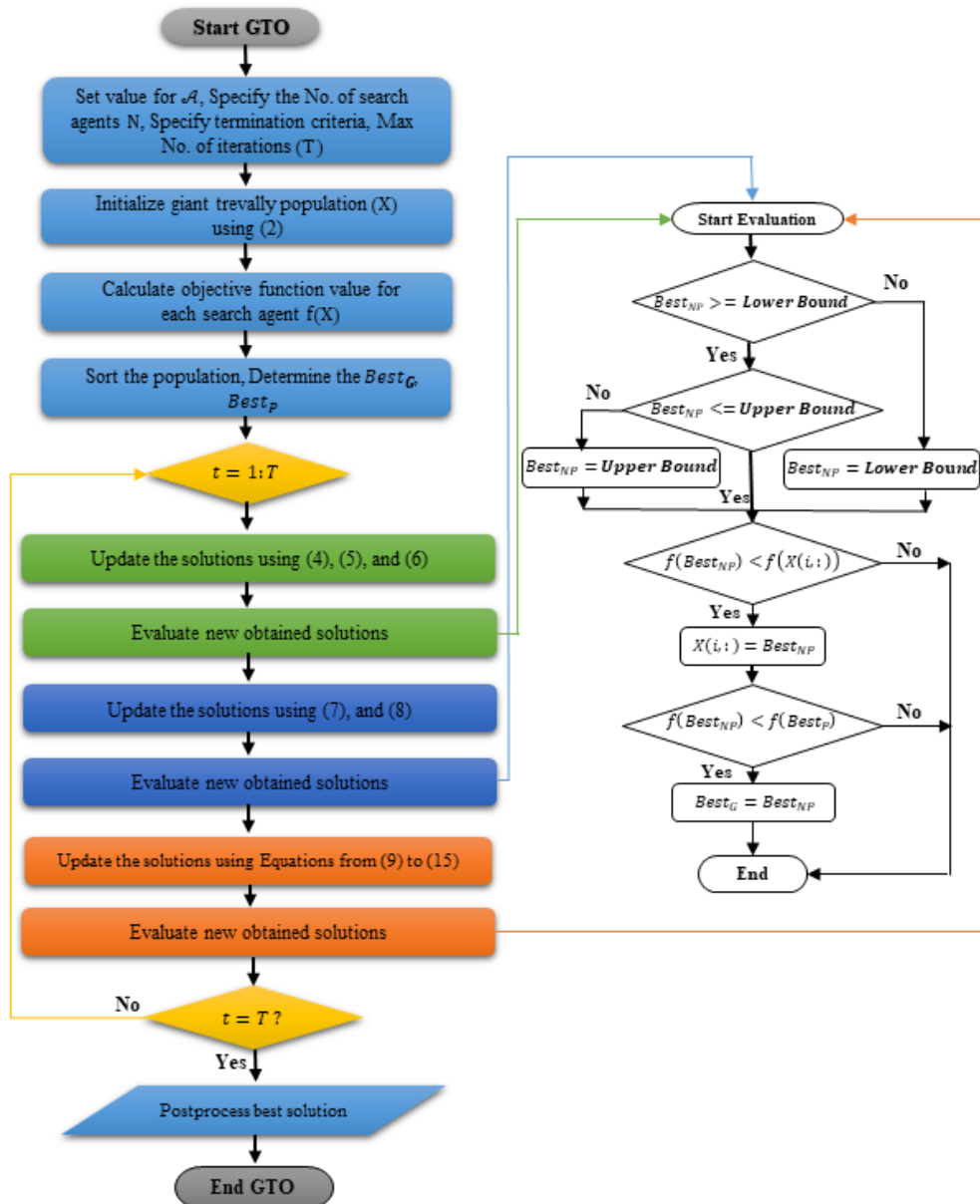


FIGURE 5. The flowchart of GTO algorithm.

### Algorithm 1: Giant Trevally Optimizer

1. *Begin*
2. Set value for  $\mathcal{A}$  parameter
3. Specify the No. of giant trevallies:  $N$
4. Specify termination criteria, Max No. of iterations ( $T$ )
5. Randomly generate population of giant trevally ( $X$ ) using (2)
6. *for*  $t = 1:T$
7. Calculate objective function for each search agent  $f(X)$
8. Sort the population
9. Determine the global best solution ( $Best_G$ )
10. Determine  $Best_P$  as the location of prey (best location)
11. *for*  $i = 1:N$
12. **Extensive Search Step:**
13. Calculate Levy flight distribution function  $Levy$  using (5) and (6)
14. Calculate new best position  $Best_{NP}$  using (4)
15. *if*  $f(Best_{NP}) < f(X(i,:))$
16.  $X(i,:) = Best_{NP}$
17. *if*  $f(Best_{NP}) < f(Best_P)$
18.  $Best_G = Best_{NP}$
19. *End if*
20. *End if*
21. **Choosing area step:**
22. Calculate mean of  $X$  using (8)
23. Calculate  $Best_{NP}$  using (7)
24. Repeat steps 15 to 20
25. **Attacking step:**
26. Calculate visual distortion  $\mathcal{V}$  using (11)
27. Calculate launch speed  $\mathcal{L}$  using (14)
28. Calculate  $Best_{NP}$  using (13)
29. Switch from exploration to exploitation using (15)
30. Repeat steps 15 to 20
31. *End for*
32. Postprocess best solution and visualization
33. *End*

## V. EXPERIMENTAL ANALYSIS AND RESULTS

To appropriately validate the performance of the GTO algorithm, two sets of experiments are conducted, and the experimental results provided by GTO are assessed and compared to those provided by other algorithms.

Case 1: The first experiment evaluates the performance of algorithms from multiple perspectives using forty benchmark test functions with various types of characteristics.

Case 2: The effectiveness of the GTO algorithm is evaluated in the second experiment using five challenging engineering design optimization problems.

## A. BENCHMARK TEST FUNCTIONS

Forty benchmark functions, which are presented in Appendix A and are fully described in [46] [47], are used in this experiment. The two main classes of functions are represented in this sizeable test suite: unimodal functions with separable and non-separable characteristics; and multimodal functions also with separable and non-separable characteristics. Since unimodal functions only have a single global optimum, they are well-suited for evaluating the exploitative (intensification) capabilities of algorithms, whereas multimodal functions, which can have many different solutions, can be used to test the algorithms' abilities to explore (diversification) and avoid local optimums.

The separable property demonstrates that the variables can be decomposed into a product of functions of each variable, whereas the non-separable property does not allow for this decomposition due to the interdependence of the variables. The non-separable property makes it more challenging to identify the global optimum. It is worth noting here that of the 40 functions used in the case 1 experiment, 16 functions are unimodal, 4 of them are separable, the rest are non-separable, and 24 functions are multimodal, 6 functions are separable, and 18 of them are non-separable.

The performance of GTO is compared with those of six different meta-heuristic algorithms, including DE, GSA, GWO, MFO, PSO, and WOA. Each optimizer is run 30 times for each function, and the population size and number of iterations are each set to 30 and 1000, respectively. Table 2 lists the parameters used by each algorithm.

TABLE 2: Parameter settings for each algorithm.

Algorithm	Parameter	Value
DE	Scale factor, crossover rate	0.5, 0
PSO	Social and cognitive parameters, inertia weight	2, 2, decrease from 0.9 to 0.4
GSA	$G_0$ , Alpha coefficient	100, 20
GWO	Convergence parameter $a$	$2 \rightarrow 0$
WOA	Convergence parameter $a$ , constant variable $b$	$2 \rightarrow 0, 1$
MFO	Convergence parameter $a$ , logarithmic factor $b$	$-1 \rightarrow -2, 1$
GTO	$\mathcal{A}, \mathcal{H}$	0.4, $2 \rightarrow 0$

In this experiment, we compare all of the candidate algorithms based on two criteria, the mean "Mean" and the standard deviations "Std" of the best solutions:

$$Mean = \frac{1}{Run} \sum_{i=1}^R Best_G \quad (16)$$

$$Std = \sqrt{\frac{1}{Run} (Best_G - Mean)^2} \quad (17)$$

Where  $Best_G$  is the global solution,  $Mean$  is the average solution obtained in the  $i^{th}$  independent run and  $Run$  is the number of independent runs. It's evident that the algorithm



can come up with more reliable and stable solutions when the values of the two evaluation criteria are smaller.

Based on the data in Table 3, it is clear that the GTO algorithm is the most effective optimizer and produces the best results in terms of mean of objective functions and standard deviation. It was the most efficient optimizer for 8 out of 16 benchmark functions (4, 8, 11, 12, 13, 14, 15, and 16) and provides the best results with at least one of the competitive algorithms in five functions (1, 2, 3, 6, and 7). For the remaining three functions, GTO came in second place, with slightly different results. Consequently, the proposed GTO algorithm is sufficient to produce excellent exploitation. Note that throughout all the comparison tables in this paper, the bold type indicates the best results.

When evaluating the exploration capability of an optimization algorithm, multimodal functions prove to be extremely helpful. Optimization of these types of functions (i.e., separable and non-separable multimodal functions) is extremely difficult because local optima can only be avoided through an adequate balance between diversification and intensification.

GTO has a very good exploration capability, according to the results for functions 17–40 reported in Table 4. In fact, the proposed algorithm consistently ranks first or second in the vast majority of test problems. This is as a result of integrated exploration mechanisms in the proposed GTO that guide this algorithm in the direction of the optimum global.

Fig. 6 displays the comparison of convergence rate changes on several benchmark functions, which demonstrates that GTO was able to find the optimal solution faster than the other algorithms in the early stages of the course of iteration. To explain this, thanks to the second step of GTO, which guides the search agents to the near global solutions as was demonstrated earlier, also the adaptive parameter in the third step of the algorithm make it possible for the search agents to exploit in an efficient manner.

### 1) SCALABILITY ANALYSIS

As the number of dimensions used in an algorithm increase, the algorithm's performance is subject to fluctuations, making scalability an essential criterion to observe. Previous section experimental results show that GTO converges well to low-dimensional benchmark functions. Unfortunately, many algorithms struggle to deal with the complex high-dimensional optimization problems that are common in real-world applications. The GTO is then used to solve 16

benchmark functions F1-F16 in dimensions (100, 500, and 1000) to further validate the efficacy of the proposed method for high dimensional optimization. Tables 5, 6, and 7 detail the outcomes of each of the seven algorithms, with the same parameter settings as the previous experiments.

The results of GTO are considerably better than those of the other six algorithms in dealing with high-dimensional functions. For functions: 1, 2, 3, 6, 7, 9, 10, 11, and 12, it was noticed that GTO always produces the global optima regardless of the number of dimensions. These findings demonstrate that GTO is not affected by the so-called "curse of dimensionality."

The main reasons behind this stable performance and these outstanding results are the proper balance between exploration and exploitation. Additionally, the extensive search step, plays an important role in these kinds of problems and ensures that new feasible points are found in order to prevent stagnation in local optima.

### 2) STATISTICAL ANALYSIS

Reporting optimization results of objective functions with mean and standard deviation indices allows for meaningful comparison and evaluation of optimization algorithms. However, it's still possible for one algorithm to be randomly superior to several others, even after several separate executions. Hence, a Wilcoxon sum rank test [48] is presented in this section to statistically demonstrate the GTO's superiority over six competing algorithms. Two samples can be compared for their similarity using the Wilcoxon sum rank test, a non-parametric statistical test. This test establishes whether or not the difference between two samples is statistically significant.

This analysis uses a metric known as  $p$ -value to determine if the corresponding algorithm is significantly better than the other. The results of the simulation test comparing the proposed GTO to all other competing algorithms are shown in Table 8. If the  $p$ -value  $< 0.05$ , the proposed GTO outperforms the competing algorithm for that set of objectives. As it is clear from Table 8, GTO outperforms all other algorithms according to the obtained pairwise  $p$ -value.

**TABLE 8.** *p*-values obtained from Wilcoxon sum rank test on table 3, and 4 benchmark functions.

Compared Algorithms	Table 3	Table 4
GTO vs DE	6.103516E-05	1.056959E-02
GTO vs GSA	1.508582E-02	3.285842E-03
GTO vs GWO	6.713867E-03	5.980582E-04
GTO vs MFO	6.103516E-04	1.174976E-03
GTO vs PSO	6.133369E-03	5.991194E-04
GTO vs WOA	4.199219E-02	9.350911E-04

The *p*-value obtained when comparing GTO to other state-of-the-art algorithms with various dimensionality scales, as shown in Table 9, is another confirmation of the significant superiority of GTO.

**TABLE 9.** *p*-values obtained from Wilcoxon sum rank test on scalability analysis with dimensions 100, 500, and 1000.

Compared Algorithms	Dimensions		
	100	500	1000
GTO vs DE	4.377772E-04	4.381111E-04	4.387772E-04
GTO vs GSA	6.430412E-04	6.441412E-04	6.430414E-04
GTO vs GWO	8.544922E-04	1.708984E-03	6.103516E-05
GTO vs MFO	6.103516E-04	6.113716E-04	6.104517E-04
GTO vs PSO	9.725524E-03	3.204586E-03	8.360654E-03
GTO vs WOA	1.708984E-03	4.638672E-03	2.148438E-02

### 3) QUALITATIVE AND QUANTITATIVE ASSESSMENT

We have so far addressed the performance and the results in terms of exploration (diversification) and exploitation (intensification). Even though these results demonstrate inferentially that the GTO algorithm converges to a point in a problem space and enhances initial solutions, we investigate the convergence of the proposed optimizer in more detail in the following sections. Hence, four metrics are calculated and discussed to confirm the convergence of the GTO algorithm:

- Search space history
- Trajectory of the first giant trevally in its first dimension
- Average fitness of all giant trevallies
- Convergence rate

The tests are repeated using 10 giant trevallies over 100 iterations on some of the benchmark functions. Fig. 7 presents the findings.

The first criterion is a qualitative indicator of change over time in the sampled points. In Fig. 7. The black dots represent the optimization samples. The giant trevallies appear to follow a similar pattern across all test functions, probing promising areas of the search space and exploiting with high precision

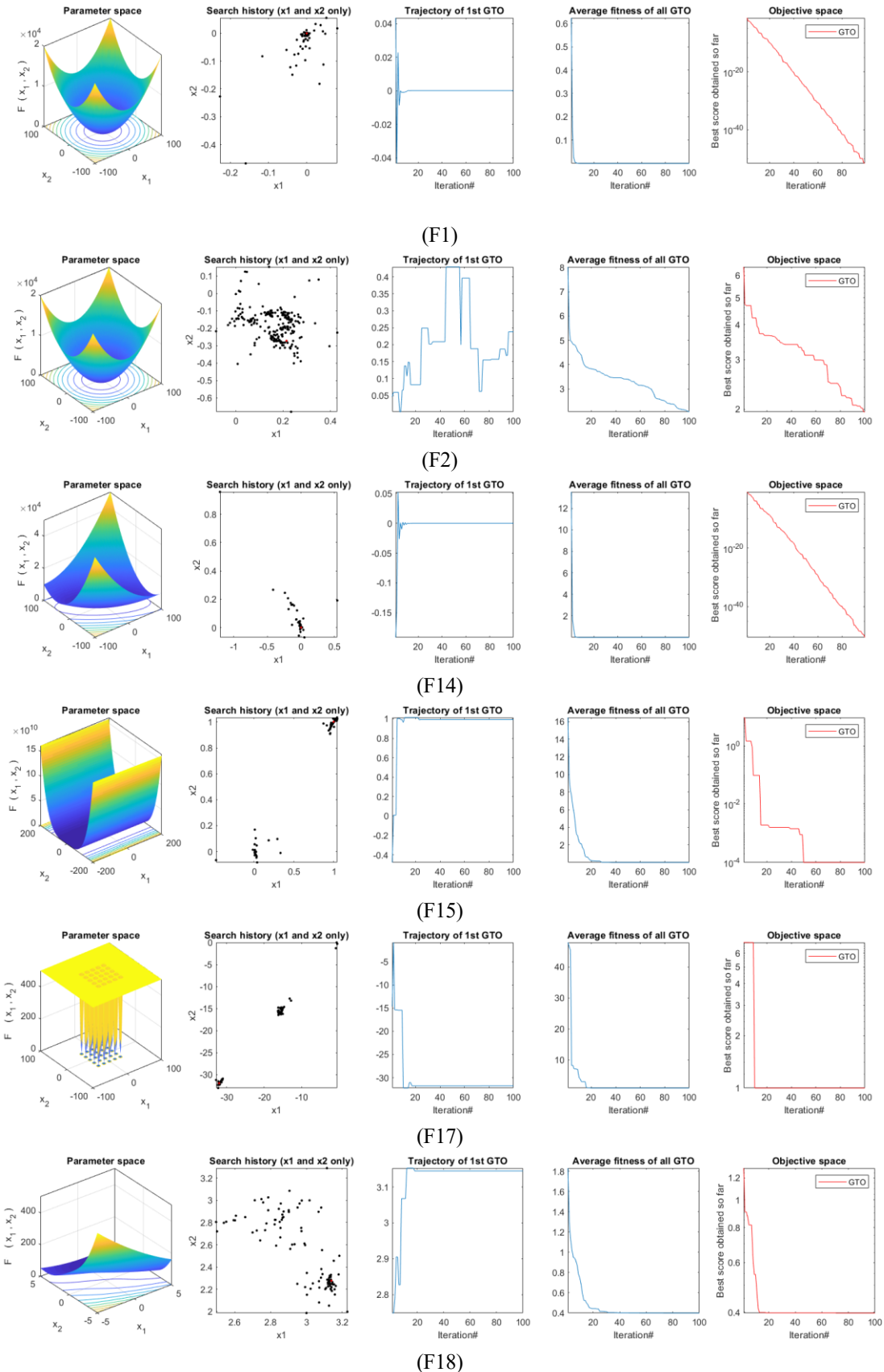
close to global optimums. These outcomes the effectiveness of the GTO algorithm in estimating global optimums of optimization problems.

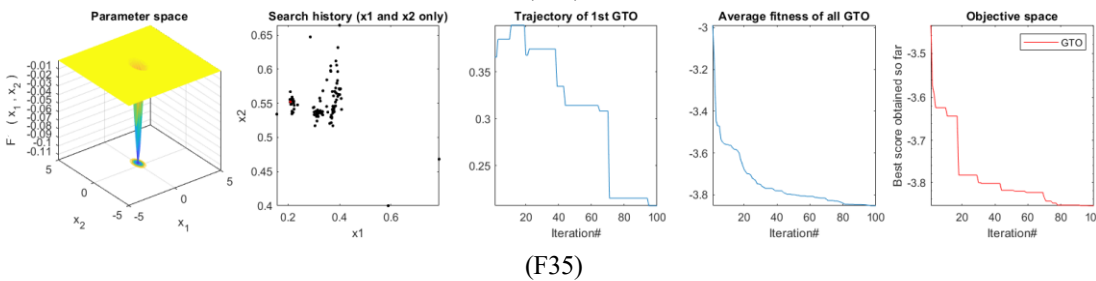
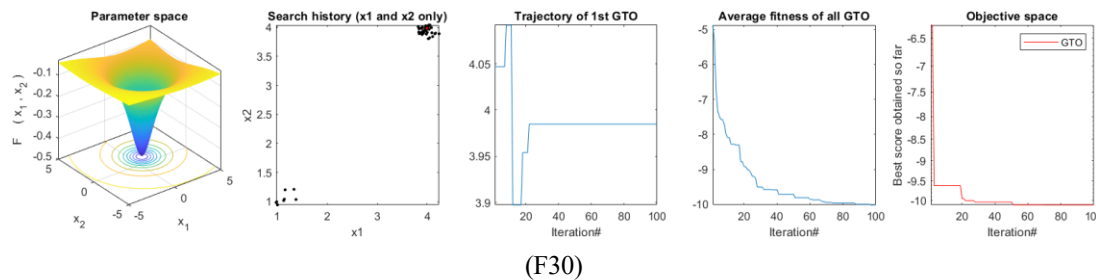
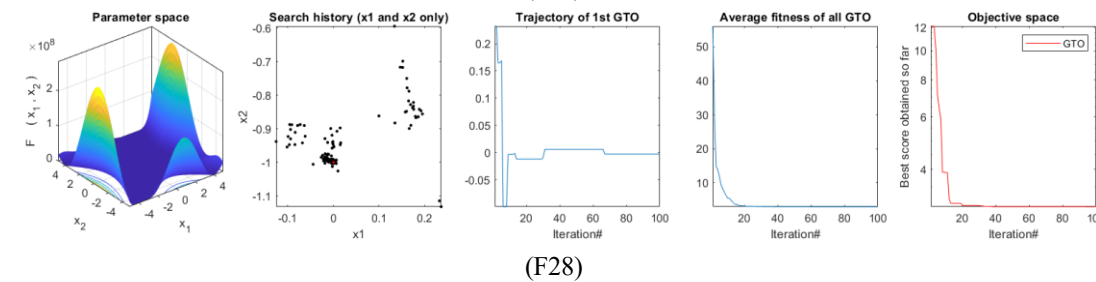
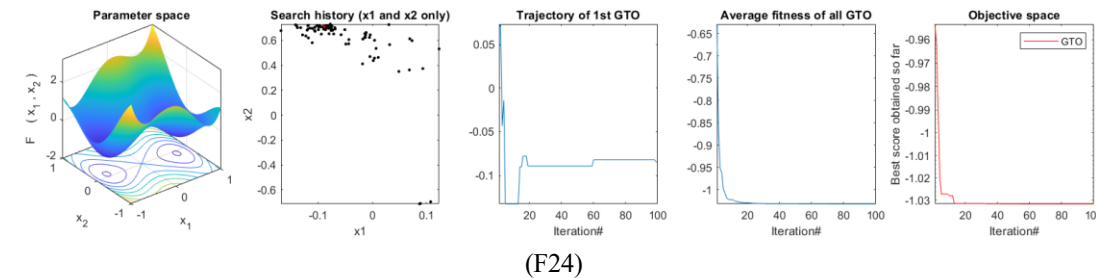
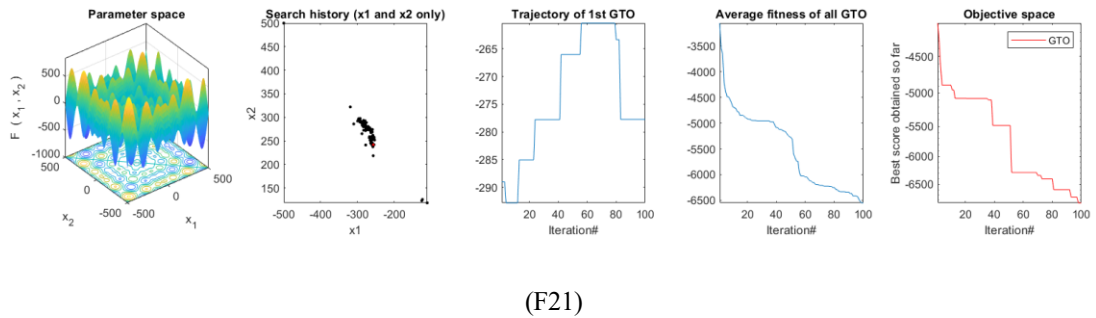
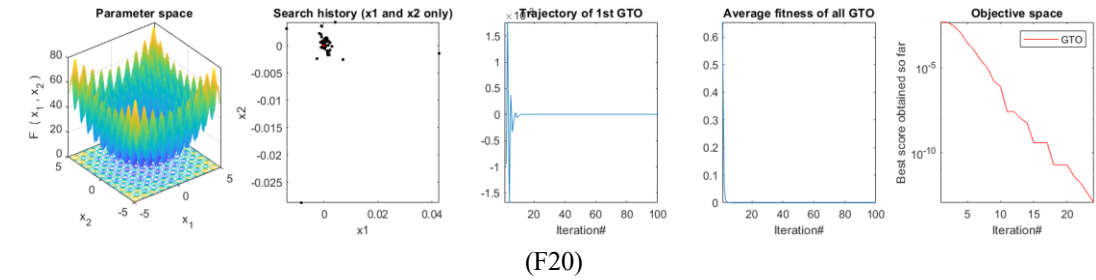
The second metric displays the evolution of the initial giant trevally's first dimension over the course of iterations; it is also a qualitative metric. With the help of this metric, we can see if the first giant trevally (as a stand-in for all giant trevallies) undergoes unpredictable changes in the early iterations and smoother changes in the later iterations. In addition, the fluctuations are seen to decrease over the course of iteration, a behavior that ensures a smooth transition between diversification and intensification.

The third metric is a quantitative average of all giant trevallies' fitness over the class of the iterative process. Certainly, the average fitness should enhance as the number of iterations progresses if the algorithm is successful in improving its candidate solutions. Based on the average fitness curves depicted in Fig. 7, it appears that the GTO algorithm has decreased fitness across the board for the test functions. The search agents get better and better over time, as evidenced by the decreasing average fitness curves, which is another fact worth mentioning here. Since the GTO algorithm adaptively switches between exploration and exploitation, the giant trevallies tend to converge with an increasing number of iterations. Also, this behavior is enabled by the powerful mechanism in the choosing area step of the proposed GTO.

The convergence rate of the GTO algorithm is the final quantitative comparison criterion presented here. After each iteration, we record the fitness of the leading giant trevally and plot their convergence curves in Fig. 7. Consistently decreasing fitness indicates that the GTO algorithm is convergent. It is also important to note that the accelerated degradation can also be observed in convergence curves, due to the previously mentioned reason.

In conclusion, this section provided experimental proof that the GTO algorithm achieves competitive results, and in most cases, even better performance, compared to other metaheuristic algorithms. Furthermore, two qualitative and two quantitative indicators were used to experimentally demonstrate the GTO algorithm's convergence. Therefore, it can be stated that the suggested GTO method will be effective in tackling real-world problems.





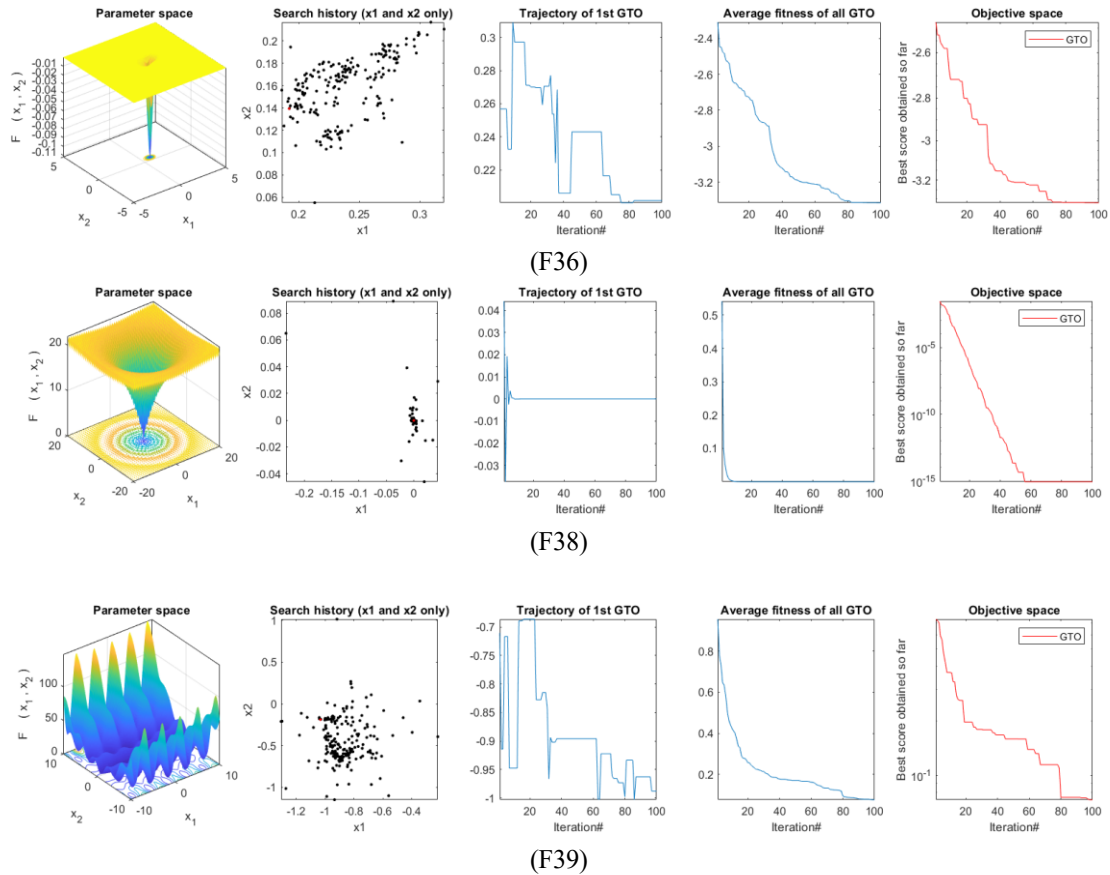


FIGURE 7. Search history, trajectory of 1st GTO, mean fitness of all GTO, convergence analysis.

### B. CASE 2: ENGINEERING DESIGN OPTIMIZATION PROBLEMS (EDOP)

To further investigate the applicability of GTO, five engineering design optimization problems (EDOP), which employ a wide variety of challenges, are implemented, and the findings are discussed here. Metaheuristic algorithms are not designed to solve constraint optimization problems directly [49], so this paper uses the straightforward death penalty technique to transform the original problems from their constrained to their unconstrained form.

It's worth mentioning here that the number of population sizes is set to 30 and the maximum number of iterations is set to 3000. All algorithms are executed for 30 independent runs for all EDOP. All DEOP are described mathematically in Appendix B.

#### 1) CANTILEVER BEAM

This challenging problem is an illustration of the optimization of the mass of a cantilever beam with a square cross section, and it arises in the field of structural engineering [50]. As can be seen in Fig. 8, the beam is stably supported at one end, and a vertical force is exerted at the

cantilever's free node. The beam is made up of five cubes with a fixed thickness (2/3) in this case. Thus, the objective of this design is to minimize the weight of the beam.

Table 10 presents the best solutions to this problem, as determined by the GTO and other meta-heuristic algorithms. We can see that the GTO yields a superior solution compared to the alternatives. In addition, Table 11 compares the statistical results of the GTO algorithm with those of other methods, demonstrating that the GTO yields a more precise result based on the best, mean, and the standard deviation indicators.

TABLE 10. Comparison of the best results of the cantilever beam design.

	GTO	DE	GSA	GWO	MFO	PSO	WOA
$f_x$	<b>1.33654</b>	1.34335	1.33996	1.33996	1.34002	1.33995	1.34271
<b>I</b>	<b>6</b>	6	6	2	5	6	0
$x_1$	5.99672	5.99614	5.97155	6.01540	6.04598	6.01642	6.03893
	4	9	2	8	5	2	7
$x_2$	5.32276	5.26084	5.37482	5.31406	5.29555	5.30972	5.28223
	0	2	4	1	5	1	5
$x_3$	4.49241	4.53196	4.48290	4.49103	4.45908	4.49462	4.29083
	2	1	5	5	7	2	4
$x_4$	3.51229	3.53565	3.50296	3.50886	3.51824	3.50049	3.70927
	7	2	1	8	1	6	9
$x_5$	2.14980	2.15139	2.14380	2.14723	2.15588	2.15239	2.19109
	1		7	1	8	9	4



**TABLE 11. Comparison of statistical results of the cantilever beam design.**

	GTO	DE	GSA	GW O	MFO	PSO	WOA
<b>Best</b>	<b>1.336</b> <b>541</b>	1.343 356	1.339 966	1.339 962	1.340 025	1.339 956	1.342 710
<b>Mean</b>	<b>1.336</b> <b>675</b>	1.359 551	1.340 012	1.339 985	1.340 549	1.339 958	1.393 314
<b>Worst</b>	1.336 857	1.378 276	1.340 152	1.340 039	1.342 187	1.339 975	1.515 978
<b>Std.</b>	<b>2.446</b> <b>5e-05</b>	0.004 2	6.337 1e-04	6.287 9e-04	7.317 6e-04	6.238 6e-04	0.010 4

## 2) THREE-BAR TRUSS

Minimizing the weight of a statically loaded three-bar truss is the goal of this practical example. The area of bars 1 and 3 and the area of bar 2 are the two parameters of interest as shown in Fig. 9. In addition, there are multiple constraints placed on this design problem by deflection, stresses, and buckling [51].

Table 12 displays GTO's best performance in comparison to other algorithms. Table 6 shows that when compared to other methods, GTO produces results that are very competitive. Also, Table 13 shows the statistical findings obtained using these methods. It's clear that the GTO offers slightly better results than competing optimizers. The findings demonstrate that the GTO can perform well in a constrained environment.

**TABLE 12. Comparison of the best results of the three-bar truss design.**

	GTO	DE	GSA	GW O	MFO	PSO	WOA
<b><math>f_x</math></b>	<b>263.8</b> <b>9584</b>	263.8 9630	263.8 9594	263.8 9597	263.8 9599	263.8 9587	263.8 9596
$x_1$	0.788 673	0.788 666	0.776 897	0.788 928	0.788 444	0.713 045	0.789 227
$x_2$	0.408 253	0.408 249	0.442 625	0.407 716	0.408 903	0.442 56	0.407 158

**TABLE 13. Comparison of statistical results of the three-bar truss design.**

	GTO	DE	GSA	GW O	MFO	PSO	WOA
<b>Best</b>	<b>263.8</b> <b>9584</b>	263.8 9630	263.8 9594	263.8 9597	263.8 9599	263.8 9587	263.8 9596
<b>Mean</b>	<b>263.8</b> <b>9592</b>	263.9 1707	263.9 2366	263.8 9765	263.9 5871	265.0 601	263.9 2441
<b>Worst</b>	263.8 9607	263.9 8920	264.0 5357	263.9 0476	264.5 4097	268.0 0685	264.0 224
<b>Std.</b>	<b>2.738</b> <b>6e-05</b>	0.003 8	0.005 0	2.884 7e-04	0.011 4	0.212 5	0.005 2

## 3) GEAR TRAIN DESIGN

The goal of this engineering design is to minimize the ratio cost of the gear train [52] depicted in Fig. 10. The design variables are the numbers of teeth on the gears, specifically  $nA(=x_1)$ ,  $nB(=x_2)$ ,  $nC(=x_3)$ , and  $nD(=x_4)$ .

Table 14 shows that the proposed GTO finds a new optimal design cost for this problem. From Table 15, GTO obtains the best results in terms of best, mean, std, and even the worst result obtained by GTO is better than the best results obtained by all other optimizers. This proves that GTO can be effective in solving discrete problems as well.

**TABLE 14. Comparison of the best results of the gear train design.**

	GTO	DE	GSA	GW O	MFO	PSO	WOA
<b><math>f_x</math></b>	<b>2.42E</b> <b>-18</b>	9.75E -10	2.70E -12	2.70E -12	8.89E -10	2.31E -11	2.70E -12
$x_1$	34.65 788	31.06 397	52.08 308	39.96 404	53.97 429	51.18 068	42.65 045
$x_2$	12	12	18.17 396	19.84 126	12	26.46 333	15.93 899
$x_3$	12	12.01 193	21.96 011	14.04 941	37.11 827	15.45 01	18.69 951
$x_4$	28.79 761	32.16 129	52.70 814	48.76 423	57.27 61	53.26 366	49.38 557

**TABLE 15. Comparison of statistical results of the gear train design.**

	GTO	DE	GSA	GW O	MFO	PSO	WO A
<b>Best</b>	<b>2.42E</b> <b>-18</b>	9.75E -10	2.70E -12	2.70E -12	8.89E -10	2.31E -11	2.70E -12
<b>Mean</b>	<b>6.89E</b> <b>-15</b>	8.47E -09	2.54E -10	8.06E -11	1.03E -08	0.004 429	8.88E -10
<b>Worst</b>	2.94E -14	5.52E -08	1.18E -09	9.92E -10	2.73E -08	0.021 025	3.30E -09
<b>Std.</b>	<b>5.367</b> <b>2e-15</b>	1.007 8e-08	2.154 4e-10	1.811 1e-10	4.984 3e-09	0.003 8	6.024 9e-10

## 4) PRESSURE VESSEL DESIGN

The purpose of this problem is to minimize the manufacturing costs, including material, forming, and welding of the cylindrical pressure vessel, whose schematic is shown in Fig. 11. The vessel has caps on both ends, and the head is hemispherical in shape. This design problem has four constraints and four variables, including the thickness of the shell  $T_s(=x_1)$ , the thickness of the head  $T_h(=x_2)$ , the

inner radius  $R(=x_3)$ , and the length of the cylindrical section, not including the head  $L(=x_4)$  [53].

Table 16 displays the results of the competitive optimizers in terms of optimal values and optimal variables. According to the findings, GTO discovers a remarkably different structure than those found by other methods, which can lead to the lowest possible fabrication cost. Table 17 verifies the robustness of the proposed algorithm, showing that the best statistical indicators are provided by the GTO.

**TABLE 16. Comparison of the best results of the pressure vessel design.**

	GTO	DE	GSA	GWO	MFO	PSO	WOA
$f_x$	<b>5889.</b>	5921.	6156.	6060.	6059.	5984.	6069.
$s$	<b>5</b>	126	492	177	714	972	587
$x_1$	0.778	0.807	1.056	12.85	13.39	0.832	19.88
	834	383	348	594	113	568	446
$x_2$	0.385	0.399	0.522	7.401	7.205	0.411	0.510
	442	34	153	851	682	539	008
$x_3$	40.34	41.81	54.73	42.09	42.09	43.13	65.22
	289	368	306	885	845	826	63
$x_4$	199.6	180.2	64.72	176.6	176.6	164.1	10.00
	848	459	945	336	366	641	016

**TABLE 17. Comparison of statistical results of the pressure vessel design.**

	GTO	DE	GSA	GW O	MFO	PSO	WOA
<b>Best</b>	<b>5889.</b>	5921.	6156.	6060.	6059.	5984.	6069.
	<b>5</b>	126	492	177	714	972	587
<b>Mean</b>	<b>5967.</b>	6196.	6502.	6343.	6553.	6316.	7042.
	<b>494</b>	557	55	283	519	036	849
<b>Worst</b>	6175.	6971.	7419.	7581.	7412.	6750.	8443.
	568	786	475	062	96	757	797
<b>Std.</b>	<b>14.23</b>	56.06	111.9	82.84	121.2	77.87	210.5
	<b>97</b>	07	271	91	327	45	718

### 5) PISTON LEVER DESIGN

The basic purpose of piston lever design is to specify the location of the piston elements:  $H(=x_1)$ ,  $B(=x_2)$ ,  $D(=x_3)$ , and  $X(=x_4)$  by setting the volume of oil to a minimum while the piston lever is raised from  $0^\circ$  to  $45^\circ$  as depicted in Fig. 12 [54].

The best results for competitive algorithms are shown in Table 18, which indicates a very close optimal solution has been provided by all the methods. Looking at Table 19, again, we can see that GTO is able to provide the best

average result and superior results compared to other methods.

**TABLE 18. Comparison of the best results of the piston lever design.**

	GTO	DE	GSA	GWO	MFO	PSO	WOA
$f_x$	8.412	8.412	8.453	8.415	<b>8.412</b>	<b>8.412</b>	8.449
	70	72	428	263	<b>698</b>	<b>698</b>	975
$x_1$	0.05	0.05	230.6	0.050	0.05	0.05	0.051
			226	251			874
$x_2$	2.052	2.229	277.9	2.042	2.041	2.041	2.045
	859	121	328	033	514	514	915
$x_3$	4.089	4.455	73.35	4.083	4.083	4.083	4.085
	713	794	033	365	027	027	849
$x_4$	119.6	100.2	0.895	119.9	120	120	119.9
	391	813	566	851			579

**TABLE 19. Comparison of statistical results of the piston lever design.**

	GTO	DE	GSA	GW O	MFO	PSO	WOA
<b>Best</b>	8.412	8.412	8.453	8.415	<b>8.412</b>	<b>8.412</b>	8.449
	70	72	428	263	<b>698</b>	<b>698</b>	975
<b>Mean</b>	<b>9.616</b>	150.9	80.93	57.76	33.65	151.5	37.25
	<b>84</b>	143	964	063	24	667	455
<b>Worst</b>	12.92	1036.	635.9	167.5	201.5	167.4	184.2
	728	329	648	857	324	727	112
<b>Std.</b>	<b>0.219</b>	26.01	13.24	9.009	4.608	26.13	5.265
	<b>8</b>	71	15	7	1	62	8

### VI. CONCLUSIONS AND FUTURE WORK

This paper presented a novel swarm-based metaheuristic algorithm inspired by the hunting behavior of giant trevallies. The proposed algorithm (named as GTO, Giant Trevally Optimizer) included three steps to simulate the behavior of giant trevallies. The steps are: extensive search (exploration), choosing area (exploration), chasing and attacking the prey (exploitation).

To investigate the exploration and exploitation capabilities of the proposed algorithm, two different sets of experiments were employed. The first experiment consisted of forty benchmark functions with a wide variety of characteristics, such as unimodal, multimodal, separable, and non-separable. The obtained results are compared with some other well-known MAs, and it was observed that the proposed GTO provides better results according to the mean, the standard deviation values, and the Wilcoxon sum rank test, which has been made to ensure that the results are not gained by chance. Furthermore, qualitative and quantitative assessment of the results using extra indicators has been

presented in this paper to check and confirm the convergence of the proposed optimizer in more detail.

The second experiment consists of five challenging engineering design optimization problems, to check the validity of the GTO to be applied to real-world problems. The problems were cantilever beam design, three-bar truss design, gear train design, pressure vessel design, and piston lever design. It is worth mentioning here that GTO showed very powerful and reliable performance when compared to other well-known MAs.

Finally, several research directions can be suggested for further. Firstly, a multi-objective version of GTO to deal

with NP-hard problems such as travelling salesman person. Secondly, GTO can be applied to tackle further challenging real-world problems and a diverse range of applications such as feature selection, image processing, and COVID-19 modeling. Last, but not least, the proposal for the binary version of the GTO.

**TABLE 1. Some Recent MAs.**

Category	Algorithm	Ref.	Inspiration	Objective
EA	Tree growth algorithm.	[55]	Inspired by the struggle of trees to obtain light and sustenance.	Global optimization, engineering optimization Problems.
	Invasive tumor growth optimization algorithm.	[56]	Inspired by invasive tumor growth.	Global optimization, support vector machine.
	Virus spread optimization.	[57]	Motivated by the contagious nature of viruses.	Global optimization, real world applications, classification problems.
SI	Bald eagle algorithm.	[58]	Bald eagle hunting strategies.	Global optimization.
	Dwarf mongoose optimization algorithm.	[59]	The dwarf mongoose's foraging style.	Global optimization, engineering optimization problems.
	Northern goshawk optimization.	[60]	Northern goshawk techniques when hunting prey.	Global optimization, engineering optimization problems.
	Flying foxes optimization.	[61]	Inspiration taken from flying fox heatwave survival techniques.	Global optimization, engineering optimization problems.
HA	Group teaching optimization algorithm.	[62]	Human group teaching techniques.	Global optimization, engineering optimization problems.
	Collective decision optimization algorithm.	[63]	Inspired by decision-making behavior of human.	Global optimization, training artificial neural networks.
	Queuing search Algorithm.	[64]	inspired from human activities in queuing process.	Global optimization, engineering optimization problems.
SCA	Atomic orbital search.	[65]	Inspired by quantum mechanics' fundamental concepts.	Global optimization, engineering optimization problems.
	Crystal structure algorithm.	[66]	Takes its motivation from the components of the crystals.	Global optimization, power electronics problems.
	Henry gas solubility optimization.	[67]	Imitates Henry's law's characteristics.	Global optimization, engineering optimization problems.
	Planet optimization algorithm.	[68]	Motivated by Newton's law of gravity.	Global optimization, engineering optimization problems.

**TABLE 3. Comparison of optimization results obtained for the unimodal, separable, and non-separable benchmark functions.**

N	In.	GTO	DE	GSA	GWO	MFO	PSO	WOA
1	M.	<b>0</b>	8.25E-10	1.11E-16	1.61E-59	1333.334	1.11E-17	<b>0</b>
	Std.	<b>0</b>	1.5062e-10	2.0266e-17	2.9394e-60	243.4324	2.0266e-18	<b>0</b>
2	M.	<b>0</b>	<b>0</b>	0.166667	<b>0</b>	3333.433	157.8333	<b>0</b>
	Std.	<b>0</b>	<b>0</b>	0.0304	<b>0</b>	608.5988	28.8163	<b>0</b>
3	M.	<b>0</b>	1.37E-10	1.13E-15	7.57E-60	803.3335	8.25E-17	<b>0</b>
	Std.	<b>0</b>	2.5013e-11	2.0631e-16	1.3821e-60	146.6680	1.5062e-17	<b>0</b>
4	M.	<b>4.95E-06</b>	8.28E-02	0.056662	4.29E-04	1.860137	1.57E-01	5.45E-04
	Std.	<b>9.0374e-07</b>	0.0151	0.0103	7.8324e-05	0.3396	0.0287	9.9503e-05
5	M.	9.62E-08	1.43E-03	<b>3.05E-29</b>	1.66E-07	5.92E-25	1.02E-01	2.92E-13
	Std.	1.7564e-08	2.6108e-04	<b>5.5685e-30</b>	3.0307e-08	1.0808e-25	0.0186	5.3312e-14
6	M.	<b>-1</b>	-9.93E-01	-0.90858	-0.9999	<b>-1</b>	-9.00E-01	-0.9999
	Std.	<b>0</b>	0.0013	0.0167	1.8257e-05	<b>0</b>	0.0183	1.8257e-05
7	M.	<b>0</b>	6.15E-05	1.57E-21	2.64E-226	1.51E-48	8.58E-118	<b>0</b>
	Std.	<b>0</b>	1.1228e-05	2.8664e-22	0	2.7569e-49	1.5665e-118	<b>0</b>
8	M.	<b>9.90E-10</b>	1.11E+00	1.521034	1.61E+00	1.34E+00	9.24E-05	4.52E-01
	Std.	<b>1.8075e-10</b>	0.2027	0.2777	0.2939	0.2446	1.6870e-05	0.0825
9	M.	-49.9999	-49.9866	<b>-50</b>	-49.9999	<b>-50</b>	<b>-50</b>	-49.9999
	Std.	1.8257e-05	0.0024	<b>0</b>	1.8257e-05	<b>0</b>	<b>0</b>	1.8257e-05
10	M.	-209.955	-202.969	-209.978	-209.996	-175.994	<b>-209.999</b>	<b>-209.999</b>
	Std.	0.0082	1.2837	0.0040	7.3030e-04	6.2086	<b>1.8257e-04</b>	<b>1.8257e-04</b>
11	M.	<b>0</b>	3.26E+01	8.47E-05	2.23E-75	12.33778	3.39E-28	2.85E-08
	Std.	<b>0</b>	5.9519	1.5464e-05	4.0714e-76	2.2526	6.1893e-29	5.2034e-09
12	M.	<b>0</b>	1.53273	0.010525	9.79E-07	459.6953	0.000183	8.91E-07
	Std.	<b>0</b>	0.2798	0.0019	1.7874e-07	83.9285	3.3411e-05	1.6267e-07
13	M.	<b>0</b>	9.85E-06	5.06E-08	7.98E-36	29.66679	0.003072	3.12e-316
	Std.	<b>0</b>	1.7984e-06	9.2383e-09	1.4569e-36	5.4164	5.6087e-04	<b>0</b>
14	M.	<b>0</b>	23380.67	436.2741	7.44E-18	17667.1	0.311367	2.84E+03
	Std.	<b>0</b>	4.2687e+03	79.6524	1.3584e-18	3.2256e+03	0.0568	518.5107
15	M.	<b>2.86E-08</b>	27.7071	32.79915	2.71E+01	2677502	21.85406	2.60E+01
	Std.	<b>5.2216e-09</b>	5.0586	5.9883	4.9478	4.8884e+05	3.9900	4.7469
16	M.	<b>0.21748</b>	1.12314	0.687494	0.666667	51702.47	0.666667	0.666688
	Std.	<b>0.0397</b>	0.2051	0.1255	0.1217	9.4395e+03	0.1217	0.1217

\* In: Indicator, M: Mean, Std: Standard deviation.

**TABLE 4. Comparison of optimization results obtained for the multimodal, separable, and non-separable benchmark functions.**

N	In.	GTO	DE	GSA	GWO	MFO	PSO	WOA
17	M.	<b>0.9980</b>	<b>0.9980</b>	3.937612	2.982105	2.412845	1.656108	1.328687
	Std.	<b>0</b>	<b>0</b>	0.5367	0.3622	0.2583	0.1202	0.0604
18	M.	0.397887	0.397887	0.397887	0.397887	0.397887	0.397887	0.397887
	Std.	2.06E-05	2.06E-05	2.06E-05	2.06E-05	2.06E-05	2.06E-05	2.06E-05
19	M.	0	0	0	0	0	0	0
	Std.	0	0	0	0	0	0	0
20	M.	<b>0</b>	9.12E-08	26.03474	0.445983	161.3768	64.38736	<b>0</b>
	Std.	<b>0</b>	1.6651e-08	4.7533	0.0814	29.4632	11.7555	<b>0</b>
21	M.	<b>-11140.1</b>	-11121.6	-2608.62	-6288.74	-8685.58	-3582.94	-11123.5
	Std.	<b>260.9715</b>	264.3492	1.8186e+03	1.1467e+03	709.1035	1.6407e+03	264.0023
22	M.	-1.8013	-1.8013	-1.8013	-1.8013	-1.8013	-1.8013	-1.8013
	Std.	0	0	0	0	0	0	0

23	M.	<b>0</b>	1.97E-05 3.5967e-	0.035509	<b>0</b>	3.07E-05	<b>0</b>	<b>0</b>
	Std.	<b>0</b>	06	0.0065	<b>0</b>	5.6050e-06	<b>0</b>	<b>0</b>
24	M.	<b>-1.03163</b>	-1.031628 3.6515e-	-1.031628	-1.031628	-1.031628	-1.031628	-1.031628
	Std.	<b>0</b>	07	3.6515e-07	3.6515e-07	3.6515e-07	3.6515e-07	3.6515e-07
25	M.	0	0	0	0	0	0	0
	Std.	0	0	0	0	0	0	0
26	M.	<b>0</b>	8.12E-06 1.4825e-	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
	Std.	<b>0</b>	06	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
27	M.	-186.7308	-186.7308 1.8257e-	-184.381	-186.7307	<b>-186.7309</b>	-186.7295	-186.7308
	Std.	1.8257e-05	05	0.4290	3.6515e-05	<b>0</b>	2.5560e-04	2.5560e-04
28	M.	<b>3</b>	3.001152 2.1033e-	2.9999	3.000008	2.9999	3.9	3.000001
	Std.	<b>0</b>	04	1.8257e-05	1.4606e-06	1.8257e-05	0.1643	1.8257e-07
29	M.	<b>0.00030</b>	0.001058 1.3839e-	0.0025	0.00030	0.001844	0.000513	0.000547
	Std.	<b>0</b>	04	4.0166e-04	0	2.8189e-04	3.8888e-05	4.5096e-05
30	M.	<b>-10.1529</b>	-10.1497 6.3901e-	-6.58316	-9.31073	-6.04891	-5.30667	-10.1528
	Std.	<b>5.4772e-05</b>	04	0.6518	0.1538	0.7493	0.8849	7.3030e-05
31	M.	-10.4026	-10.4027 3.6515e-	<b>-10.4029</b>	-10.4025	-7.27585	-4.76256	-9.97112
	Std.	5.4772e-05	05	<b>0</b>	7.3030e-05	0.5709	1.0298	0.0788
32	M.	-10.5362	-10.5361 5.4772e-	<b>-10.5364</b>	-10.2654	-7.94219	-3.7531	-10.35597
	Std.	3.6515e-05	05	<b>0</b>	0.0495	0.4736	1.2385	0.0329
33	M.	<b>0.203693</b>	0.434647	9.238981	0.887854	0.579337	1.339865	2.058237
	Std.	<b>0.0372</b>	0.0794	1.6868	0.1621	0.1058	0.2446	0.3758
34	M.	<b>0.00618</b>	0.069194	0.066936	0.258871	0.130371	0.003463	1.338754
	Std.	<b>0.0011</b>	0.0126	0.0122	0.0473	0.0238	6.3225e-04	0.2444
35	M.	<b>-3.86278</b>	<b>-3.86278</b>	<b>-3.86278</b>	-3.86156	<b>-3.86278</b>	<b>-3.86278</b>	-3.86172
	Std.	<b>0</b>	<b>0</b>	<b>0</b>	2.2274e-04	<b>0</b>	<b>0</b>	1.9353e-04
36	M.	<b>-3.32196</b>	-3.322 3.6515e-	-3.32195	-3.27394	-3.23331	-3.29029	-3.27807
	Std.	<b>3.5785e-04</b>	04	3.5602e-04	0.0084	0.0158	0.0054	0.0077
37	M.	<b>0</b>	0.00017 3.1038e-	7.778281	0.003715	9.031947	0.006728	0.000681
	Std.	<b>0</b>	05	1.4201	6.7826e-04	1.6490	0.0012	1.2433e-04
38	M.	<b>8.88E-16</b>	2.51E-05 4.5826e-	7.99E-09	1.62E-14	1.52E+01	5.95058	3.26E-15
	Std.	<b>1.6213e-16</b>	06	1.4588e-09	2.9577e-15	2.7751	1.0864	5.9519e-16
39	M.	4.51E-11	1.38E-10 2.5195e-	0.012564	4.89E-01	2.01E+00	8.277471	1.55E-02
	Std.	8.2341e-12	11	0.0023	0.0893	0.3670	1.5113	0.0028
40	M.	-1.08093	<b>-1.08094</b>	-1.05262	<b>-1.08094</b>	<b>-1.08094</b>	-0.61572	<b>-1.08094</b>
	Std.	<b>0</b>	<b>0</b>	0.0050	<b>0</b>	<b>0</b>	0.0848	<b>0</b>



**TABLE 5. Results of benchmark functions (F1–F16), with 100 dimensions.**

N	In.	GTO	DE	GSA	GWO	MFO	PSO	WOA
1	M.	<b>0</b>	47.877578	921.12704	1.76E-29	29727.183	0.2337119	2.49E-150 4.5461e-
	Std.	<b>0</b>	8.7412	168.1740	3.2133e-30	5.4274e+03	0.0427	151
2	M.	<b>0</b>	59.1	1365.667	<b>0</b>	0.000123	1190.6	<b>0</b>
	Std.	<b>0</b>	10.7901	249.3355	<b>0</b>	2.2457e-05	217.3728	<b>0</b>
3	M.	<b>0</b>	1.1385858	4.03E-06	1.44E-60	140.00001	4.78E-19	9.14E-149 1.6687e-
	Std.	<b>0</b>	0.2079	7.3577e-07	2.6291e-61	25.5604	8.7270e-20	149
4	M.	<b>5.19E-06</b> <b>9.4756e-</b>	1.717125	1.977263	0.002855	194.1186	1.193912	0.000613
	Std.	<b>07</b>	0.3135	0.3610	5.2125e-04	35.4410	0.2180	1.1192e-04
5	M.	1.33E-07 2.4282e-	0.026144	<b>0</b>	5.46E-08	1.23E-32	<b>0</b>	3.64E-11
	Std.	08	0.0048	<b>0</b>	9.9686e-09	2.2457e-33	<b>0</b>	6.6457e-12
6	M.	<b>-1</b>	-0.06846	-0.9999	-0.9999	<b>-1</b>	-0.80002	-0.9999
	Std.	<b>0</b>	0.1701	1.8257e-05	1.8257e-05	<b>0</b>	0.0365	0.0077
7	M.	<b>0</b>	0.0188633	5.61E-95	2.49E-211	7.51E-91	8.74E-113	<b>0</b>
	Std.	<b>0</b>	0.0034	1.0242e-95	0	1.3711e-91	1.5957e-113	<b>0</b>
8	M.	<b>3.50E-10</b> <b>6.3901e-</b>	18.07028	2.196793	0.974126	1.430611	2.00E-05	0.540568
	Std.	<b>11</b>	3.2992	0.4011	0.1779	0.2612	3.6515e-06	0.0987
9	M.	-49.9999 1.8257e-	-37.1471	-49.9543	-49.9343	<b>-50</b>	<b>-50</b>	-49.9999
	Std.	05	2.3466	0.0083	0.0120	<b>0</b>	<b>0</b>	1.8257e-05
10	M.	-209.949	87.31515	-132.634	-189.664	-209.914	<b>-209.999</b>	-209.914
	Std.	0.0092	54.2821	14.1250	3.7128	0.0157	<b>1.8257e-04</b>	0.0157
11	M.	<b>0</b>	56.00411	0.213742	3.35E-71	17.29012	8.53E-31	0.436312
	Std.	<b>0</b>	10.2249	0.0390	6.1162e-72	3.1567	1.5574e-31	0.0797
12	M.	<b>0</b>	31.00918	2.822843	3.26E-06	137.5193	0.00019	8.53E-06
	Std.	<b>0</b>	5.6615	0.5154	5.9519e-07	25.1075	3.4689e-05	1.5574e-06
13	M.	<b>0</b>	4.544403	6.656125	5.29E-18	161.8037	2.818065	3.23E-103 5.8971e-
	Std.	<b>0</b>	0.8297	1.2152	9.6582e-19	29.5412	0.5145	104
14	M.	<b>0</b>	261278.1	9253.6796	4.794336	210644.9	2073.485	758529.4
	Std.	<b>0</b>	4.7703e+04	1.6895e+03	0.8753	3.8458e+04	378.5648	1.3849e+05
15	M.	<b>2.53E-08</b> <b>4.6191e-</b>	9528.716	7434.74301	97.77792	1.14E+08	134.66370	97.95426
	Std.	<b>09</b>	1.7397e+03	1.3574e+03	17.8517	2.0813e+07	24.5861	17.8839
16	M.	<b>0.249904</b>	574.4743	1983.282043	0.666677	3255158	3.9581657	0.6667362
	Std.	<b>0.0456</b>	104.8842	362.0961	0.1217	5.9431e+05	0.7227	0.1217

**TABLE 6. Results of benchmark functions (F1–F16), with 500 dimensions.**

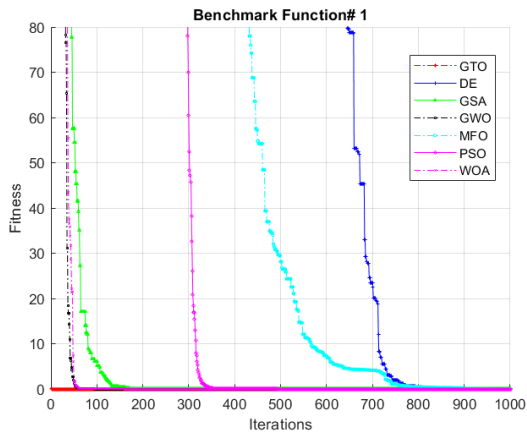
N	In.	GTO	DE	GSA	GWO	MFO	PSO	WOA
1	M.	<b>0</b>	482710.07	42097.5342	2.02E-12	958395.19	180.59105	1.45E-147
	Std.	<b>0</b>	8.8130e+04	7.6859e+03	3.6880e-13	1.7498e+05	32.9713	2.6473e-148
2	M.	<b>0</b>	489993.3	8103.11375	<b>0</b>	4000.002	21051.4	<b>0</b>
	Std.	<b>0</b>	8.9460e+04	1.4794e+03	<b>0</b>	730.2971	3.8434e+03	<b>0</b>
3	M.	<b>0</b>	2088.8798	6.173474	1.30E-60	840.00002	3.15E-18	3.82E-153
	Std.	<b>0</b>	381.3755	1.1271	2.3735e-61	153.3623	5.7511e-19	6.9743e-154
4	M.	<b>7.53E-06</b>	11633.41	669.12571	0.014366	30538.66	9.403064	0.001533

	Std.	<b>1.3748e-06</b>	2.1240e+03	122.1651	0.0026	5.5756e+03	1.7168	2.7989e-04
5	M.	1.17E-07	0.470109	<b>0</b>	2.54E-08	5.16E-31	0.152414	0.152414
	Std.	2.1361e-08	0.0858	<b>0</b>	4.6374e-09	9.4208e-32	0.0278	0.0278
6	M.	<b>-1</b>	-0.00026	-0.9999	-0.9999	<b>-1</b>	-0.80002	-0.9999
	Std.	<b>0</b>	0.1825	1.8257e-05	1.8257e-05	<b>0</b>	0.0365	1.8257e-05
7	M.	<b>0</b>	0.0719741	6.00E-99	4.54E-213	5.33E-61	4.39E-121	<b>0</b>
	Std.	<b>0</b>	0.0131	1.0954e-99	0	9.7312e-62	8.0150e-122	<b>0</b>
8	M.	<b>3.46E-10</b>	286.2456	2.146697	0.947493	1.599825	0.000113	1.86436
	Std.	<b>6.3171e-11</b>	52.2611	0.3919	0.1730	0.2921	2.0631e-05	0.3404
9	M.	-49.9999	97.08827	-46.3648	-49.9999	<b>-50</b>	<b>-50</b>	-49.9999
	Std.	1.8257e-05	26.8545	0.6637	1.8257e-05	<b>0</b>	<b>0</b>	1.8257e-05
10	M.	-209.970	2547.579	-65.637	-150.296	-209.969	-209.999	<b>-209.973</b>
	Std.	0.0055	503.4627	26.3570	10.9004	0.0057	1.8257e-04	<b>0.0049</b>
11	M.	<b>0</b>	70.83805	7.94025	1.27E-71	17.37926	1.44E-32	0.096259
	Std.	<b>0</b>	12.9332	1.4497	2.3187e-72	3.1730	2.6291e-33	0.0176
12	M.	<b>0</b>	1305.176	1.547747	6.08E-07	102.8377	0.000197	3.87E-07
	Std.	<b>0</b>	238.2914	0.2826	1.1101e-07	18.7755	3.5967e-05	7.0656e-08
13	M.	<b>0</b>	1.40E+62	236.1038	6.13E-08	2260.626	62.85609	1.28E-101
	Std.	<b>0</b>	2.5560e+61	43.1065	1.1192e-08	412.7320	11.4759	2.3369e-102
14	M.	<b>0</b>	5386943	250912.02	129940.9	3209064	213039.34	27713499
	Std.	<b>0</b>	9.8352e+05	4.5810e+04	2.3724e+04	5.8589e+05	3.8895e+04	5.0598e+06
15	M.	<b>5.10E-07</b>	1476309373	5987205.58	497.695488	3936013373	5298.087351	495.5053716
	Std.	<b>9.3113e-08</b>	2.6954e+08	1.0931e+06	90.8663	7.1861e+08	967.2940	90.4665
16	M.	<b>0.249998</b>	172114794	1237432.27	0.66675	444413426	1573.061095	0.666849347
	Std.	<b>0.0456</b>	3.1424e+07	2.2592e+05	0.1217	8.1138e+07	287.2003	0.1217

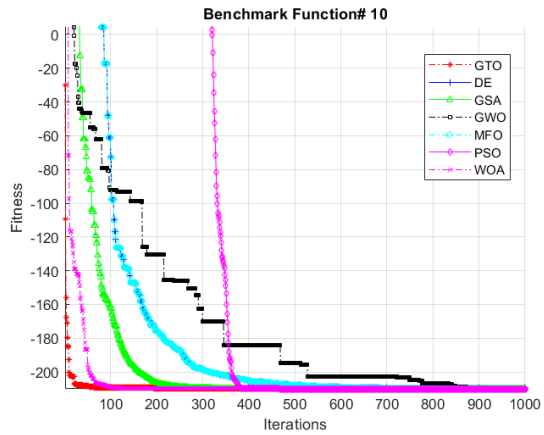
**Table 7. Results of benchmark functions (F1–F16), with 1000 dimensions.**

N	In.	GTO	DE	GSA	GWO	MFO	PSO	WOA
1	M.	<b>0</b>	1855193	94802.0596	8.70E-09	2453702.649	2245.940245	1.94E-147
	Std.	<b>0</b>	3.3871e+05	1.7308e+04	1.5884e-09	4.4798e+05	410.0507	3.5419e-148
2	M.	<b>0</b>	1866844.8	17932.9470	0.1	2000.000996	40148.8	<b>0</b>
	Std.	<b>0</b>	3.4084e+05	3.2741e+03	0.0183	365.1486	7.3301e+03	<b>0</b>
3	M.	<b>0</b>	5097.232	15.829149	1.76E-60	740.0000394	2.13E-20	4.59E-153
	Std.	<b>0</b>	930.6230	2.8900	3.2133e-61	135.1049	3.8888e-21	8.3802e-154
4	M.	<b>3.12E-05</b>	119525.1	5438.69527	0.020116	171528.0089	25.18264557	0.000957551
	Std.	<b>5.6963e-06</b>	2.1822e+04	992.9654	0.0037	3.1317e+04	4.5977	1.7482e-04
5	M.	1.96E-08	1.016611	<b>0</b>	5.56E-08	1.11E-32	<b>0</b>	5.71E-11
	Std.	3.5785e-09	0.1856	<b>0</b>	1.0151e-08	2.0266e-33	<b>0</b>	1.0425e-11
6	M.	<b>-1</b>	-0.01269	-0.98878	-0.9999	<b>-1</b>	-0.80002	-0.9999
	Std.	<b>0</b>	0.1803	0.0020	1.8257e-05	<b>0</b>	0.0365	1.8257e-05
7	M.	<b>0</b>	0.1078396	7.08E-98	1.92E-214	8.71E-55	7.84E-119	<b>0</b>
	Std.	<b>0</b>	0.0197	1.2926e-98	<b>0</b>	1.5902e-55	1.4314e-119	<b>0</b>
8	M.	<b>9.38E-11</b>	750.2222	3.077624	1.2967011	0.7695730	0.0015519	0.6506095
	Std.	<b>1.7125e-11</b>	136.9712	0.5619	0.2367	0.1405	2.8335e-04	0.1188
9	M.	-49.9999	223.6079	-47.1642	-49.9999	<b>-50</b>	<b>-50</b>	-49.9999
	Std.	1.8257e-05	49.9537	0.5177	1.8257e-05	<b>0</b>	<b>0</b>	1.8257e-05
10	M.	-209.9419	2547.579	-10.537	-198.91304	-209.903	<b>-209.999</b>	-209.9508
	Std.	0.0106	503.4627	36.4168	2.0242	0.0177	<b>1.8257e-04</b>	0.0090
11	M.	<b>0</b>	70.83805	3.416503	1.44E-71	22.349076	3.83E-33	0.1842702
	Std.	<b>0</b>	12.9332	0.6238	2.6291e-72	4.0804	6.9926e-34	0.0336
12	M.	<b>0</b>	1305.176	4.545906	3.78E-06	141.4858937	0.000148	1.75E-08
	Std.	<b>0</b>	238.2914	0.8300	6.9013e-07	25.8317	2.7021e-05	3.1950e-09
13	M.	<b>0</b>	1.66E+62	4.16E+27	0.000152	2371.727	1799.069614	3.36E-103
	Std.	<b>0</b>	3.0307e+61	7.5951e+26	2.7751e-05	433.0161	328.4637	6.1345e-104
14	M.	<b>0</b>	6972141.9	2521757.4	826882.784	14707687.53	992500.3948	112746398.5
	Std.	<b>0</b>	1.2729e+06	4.6041e+05	1.5097e+05	2.6852e+06	1.8120e+05	2.0585e+07

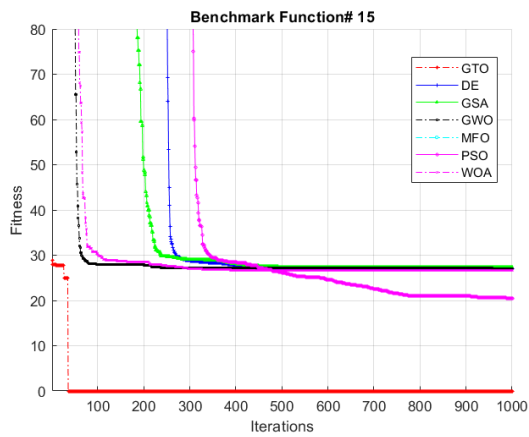
						1100951463		
15	M.	<b>3.57E-06</b>	2077998246	15790196.57	997.144	6	51377.57135	992.136551
	Std.	<b>6.5179e-07</b>	3.7939e+08	2.8829e+06	182.0528		2.0101e+09	9.3802e+03
16	M.	<b>0.250001</b>	335370699	5869118.073	0.866822		2643605992	22897.79199
	Std.	<b>0.0456</b>	6.1230e+07	1.0715e+06	0.1583		4.8265e+08	4.1805e+03



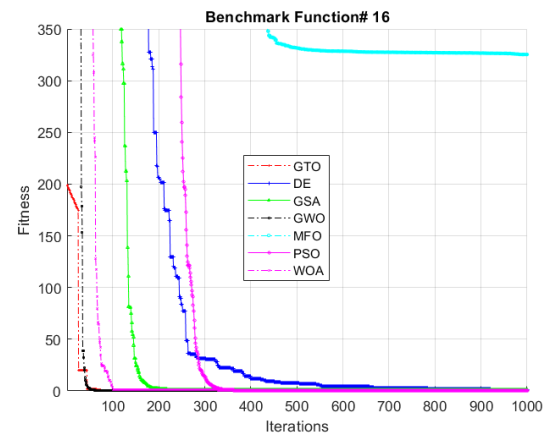
(F1)



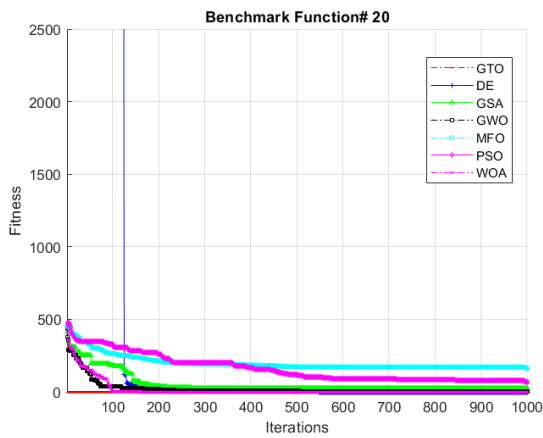
(F10)



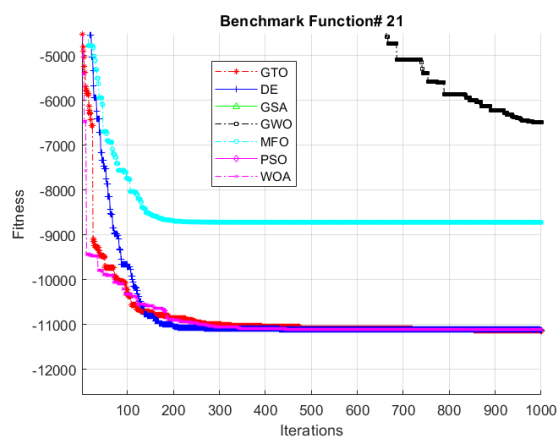
(F15)



(F16)



(F20)



(F21)

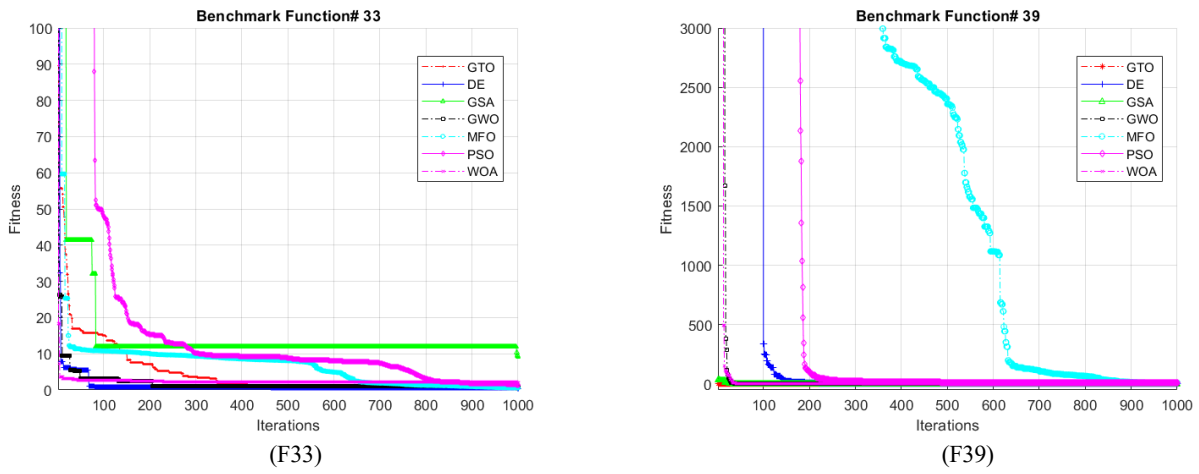


FIGURE 6. Convergence curve change rate of GTO with other algorithms in a number of benchmark test functions.

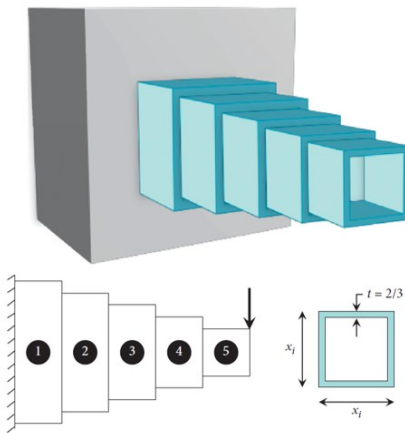


FIGURE 8. Cantilever beam.

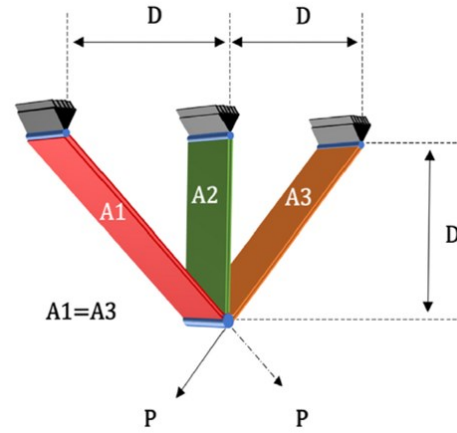


FIGURE 9. Three-bar truss.

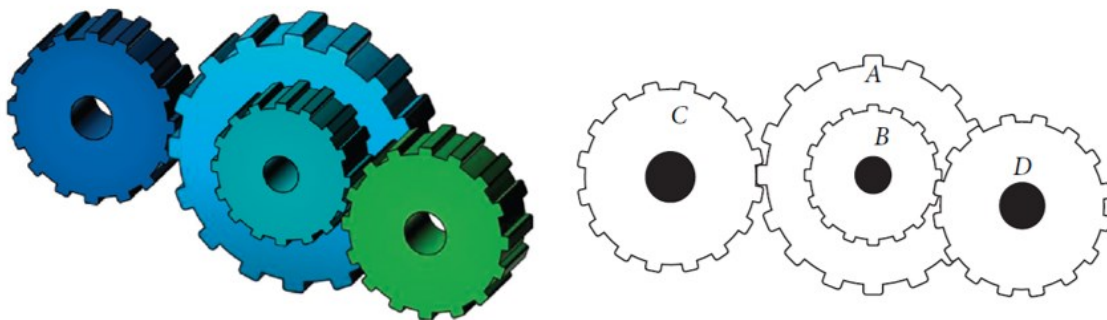


FIGURE 10. Gear train design.

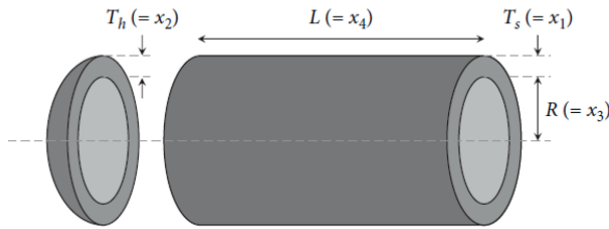


FIGURE 11. pressure vessel design.

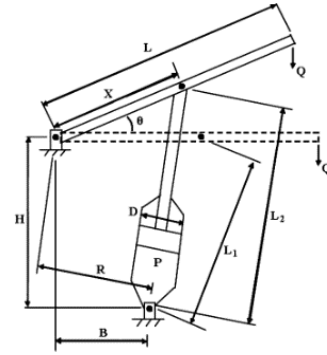


FIGURE 12. piston lever design.

## APPENDIX A

### 1. Benchmark functions used in tables 3, 5, 6, and 7.

NO.	Name	Function	Opt.	Range	Ch.	Dim
1	Sphere	$f_2(X) = \sum_{i=1}^n x_i^2$	0	[-100,100]	US	30
2	Step	$f_1(X) = \sum_{i=1}^n ([x_i + 0.5])^2$	0	[-100,100]	US	30
3	SumSquare s	$f_3(X) = \sum_{i=1}^n ix_i^2$	0	[-10,10]	US	30
4	Quartic	$f_4(X) = \sum_{i=1}^n x_i^4 + random[0,1)$	0	[-1.28,1.28]	US	30
5	Beale	$f_5(X) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	0	[-4.5,4.5]	UN	5
6	Easom	$f_6(X) = -\cos(x_1)\cos(x_2)e^{-(x_1-\pi)^2-(x_2-\pi)^2}$	-1	[-100,100]	UN	2
7	Matyas	$f_7(X) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	0	[-10,10]	UN	2
8	Collvile	$f_8(X) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2)$	0	[-10,10]	UN	4
9	Trid6	$f_9(X) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}$	-50	$[-Dim^2, Dim^2]$	UN	6
10	Trid10	$f_{10}(X) = \sum_{i=1}^n (x_i - 1)^2 - \sum_{i=2}^n x_i x_{i-1}$	-210	$[-Dim^2, Dim^2]$	UN	10
11	Zakharov	$f_{11}(X) = \sum_{i=1}^n x_i^2 + (\sum_{i=1}^n 0.5ix_i)^2 + (\sum_{i=1}^n 0.5ix_i)^4$	0	[-5,10]	UN	10
12	Powell	$f_{12}(X) = \sum_{i=1}^{n/k} (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4$	0	[-4,5]	UN	24
13	Schwefel 2.22	$f_{13}(X) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	0	[-10,10]	UN	30



14	Schwefel 1.2	$f_{14}(X) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2$	0	[-100,100]	UN	30
15	Rosenbrock	$f_{15}(X) = \sum_{i=1}^n [100(x_{i+1} + x_i^2) + (x_i - 1)^2]$	0	[-30,30]	UN	30
16	Dixon-Price	$f_{16}(X) = (x_i - 1)^2 + \sum_{i=2}^n i(2x_i^2 - x_{i-1})^2$	0	[-10,10]	UN	30

\* Opt: Optimal solution, Ch: Characteristics Dim: Dimensions, U: Unimodal, S: Separable, N: Non-Separable.

**2. Benchmark functions used in table 4.**

NO.	Name	Function	Opt.	Range	Ch.	Dim
17	Foxholes	$f_{17}(X) = \left[ \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{\sum_{i=1}^{25} (x_i - a_{ij})^6} \right]^{-1}$	0.998	[-65.536,65.536]	MS	2
18	Branin	$f_{18}(X) = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	0.398	[-5,10] × [0,15]	MS	2
19	Bohachevsky 1	$f_{19}(X) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) + 0.7$	0	[-100,100]	MS	2
20	Rastrigin	$f_{20}(X) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	0	[-5.12,5.12]	MS	30
21	Schwefel	$f_{21}(X) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	12569.5	[-500,500]	MS	30
22	Michalewicz 2	$f_{22}(X) = - \sum_{i=1}^n \sin(x_i) (\sin(ix_i^2/\pi))^{20}$	-1.8013	[0,π]	MS	2
23	Schaffer	$f_{25}(X) = 0.5 + \frac{\sin^2(\sqrt{x_1^2 + x_2^2}) - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	0	[-100,100]	MN	2
24	Six Hump Camel Back	$f_{26}(X) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	-1.03163	[-5,5]	MN	2
25	Bohachevsky 2	$f_{27}(X) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) (4\pi x_2) + 0.3$	0	[-100,100]	MN	2
26	Bohachevsky 3	$f_{28}(X) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1 + 4\pi x_2) + 0.3$	0	[-100,100]	MN	2
27	Shubert	$f_{29}(X) = (\sum_{i=1}^5 i \cos((i+1)x_1 + i)) (\sum_{i=1}^5 i \cos((i+1)x_2 + i))$	-186.7309	[-10,10]	MN	2
28	Goldstein-Price	$f_{30}(X) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	3	[-2,2]	MN	2
29	Kowalik	$f_{31}(X) = \sum_{i=1}^{11} \left( a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right)^2$	0.00031	[-5,5]	MN	4
30	Shekel5	$f_{32}(X) = - \sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	-10.1532	[0,10]	MN	4
31	Shekel7	$f_{33}(X) = - \sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	-10.4029	[0,10]	MN	4
32	Shekel10	$f_{34}(X) = - \sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	-10.5364	[0,10]	MN	4
33	Perm	$f_{35}(X) = \sum_{k=1}^n \left( \sum_{i=1}^n (i^k + \beta) ((x_1/i)^k - 1) \right)^2$	0	[-Dim, Dim]	MN	4
34	PowerSum	$f_{36}(X) = \sum_{k=1}^n \left( \left( \sum_{i=1}^n x_i^k \right) - b_k \right)^2$	0	[0, Dim]	MN	4
35	Hartman3	$f_{37}(X) = - \sum_{i=1}^4 c_i \exp \left( - \sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right)$	-3.86278	[0,1]	MN	3

36	Hartman6	$f_{38}(X) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2\right)$	-3.32 236	[0,1]	MN	6
37	Griewank	$f_{39}(X) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0	[-600,600]	MN	30
38	Ackley	$f_{40}(X) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	0	[-32,32]	MN	30
39	Penalized2	$f_{41}(X) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{29} (x_i - 1)^2 P[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_{30})]\} + \sum_{i=1}^{30} u(x_i, 5, 100, 4)$	0	[-50,50]	MN	30
40	Langerman2	$f_{42}(X) = -\sum_{i=1}^m c_i \left(\exp\left(-\frac{1}{\pi} \sum_{j=1}^n (x_j - a_{ij})^2\right) \cos(\pi \sum_{j=1}^n (x_j - a_{ij})^2)\right)$	-1.08	[0,10]	MN	2

\* Opt: Optimal solution, Ch: Characteristics Dim: Dimensions, M: Multimodal, S: Separable, N: Non-Separable.

## APPENDIX B

This appendix presents the formulation of all EDOPs used in this paper.

### 1. CANTILEVER BEAM

Minimize:  $f(X) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5)$ ,

Subject to:  $g(X) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0$ ,

Variable range:  $0.01 \leq (x_1, \dots, x_5) \leq 100$

### 2. Three-Bar Truss

Minimize:  $f(X) = 2\sqrt{2}x_1 + x_2 \times l$ ,

Subject to:

$$g_1(X) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0,$$

$$g_2(X) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0,$$

$$g_3(X) = \frac{x_2}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0,$$

Where

$$l = 100 \text{ cm},$$

$$P = 2kN/cm^3,$$

$$\sigma = 2kN/cm^3,$$

Variable range:  $0 \leq x_1 \leq 0, 1 \leq x_2 \leq 1$

### 3. Gear Train Design

Minimize:  $f(X) = \left(\frac{1}{6.931} - \frac{x_3x_2}{x_1x_4}\right)^2$

Subject to:  $12 \leq (x_1, \dots, x_4) \leq 60$ ,

### 4. Pressure Vessel Design

Minimize:  $f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$ ,

Subject to:

$$g_1(X) = -x_1 + 0.0193x_3 \leq 0,$$

$$g_2(X) = -x_2 + 0.00954x_3 \leq 0,$$

$$g_3(X) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0,$$

$$g_4(X) = x_4 - 240 \leq 0,$$

Variable range:

$$0.0625 \leq x_1, x_2 \leq 99,$$

$$10 \leq x_3, x_4 \leq 200$$

## 5. Piston Lever Design

Minimize:  $f(X) = \frac{1}{4}\pi x_3^2(L_2 - L_1),$

Subject to:

$$g_1(X) = QL\cos\theta - R \times F \leq 0,$$

$$g_2(X) = Q(L - x_4) - M_{max} \leq 0,$$

$$g_3(X) = 1.2(L_2 - L_1) - L_1 \leq 0,$$

$$g_4(X) = \frac{x_3}{3} - x_2 \leq 0,$$

Where

$$R = \frac{|-x_4(x_4\sin\theta + x_1) + x_1(x_2 - x_4\cos\theta)|}{\sqrt{(x_4 - x_2)^2 + x_1^2}},$$

$$F = \frac{\pi P x_3^2}{4},$$

$$L_1 = \sqrt{(x_4 - x_2)^2 + x_1^2},$$

$$L_2 = \sqrt{(x_4\sin\theta + x_1)^2 + (x_2 - x_4\cos\theta)^2},$$

$$\theta = 45^\circ,$$

$$Q = 10000 \text{ lbs},$$

$$L = 240 \text{ in},$$

$$M_{max} = 1.8 \times 10^6 \text{ lbs in},$$

$$P = 1500 \text{ psi},$$

Variable range:

$$0.05 \leq x_1, x_2, x_3 \leq 500,$$

$$0.05 \leq x_3 \leq 120$$

## REFERENCES

- [1] H. Gezici and H. Livatyali, "Chaotic Harris hawks optimization algorithm," *J. Comput. Des. Eng.*, vol. 9, no. 1, pp. 216–245, 2022, doi: 10.1093/jcde/qwab082.
- [2] H. Sadeeq and A. M. Abdulazeez, "Hardware Implementation of Firefly Optimization Algorithm Using FPGAS," *ICOASE 2018 - Int. Conf. Adv. Sci. Eng.*, pp. 30–35, 2018, doi: 10.1109/ICOASE.2018.8548822.
- [3] A. Sabry Eesa, A. Mohsin Abdulazeez, and Z. Orman, "Cuttlefish Algorithm – A Novel Bio-Inspired," *Int. J. Sci. Eng. Res.*, vol. 4, no. 9, pp. 1978–1986, 2013.
- [4] X.-S. S. Yang and M. Karamanoglu, *Nature-Inspired Metaheuristic Algorithms Second Edition*, vol. 4, no. C. 2013.
- [5] H. Sadeeq, A. Abdulazeez, N. Kako, and A. Abraham, "A novel hybrid bird mating optimizer with differential evolution for engineering design optimization problems," *Lect. Notes Data Eng. Commun. Technol.*, vol. 5, pp. 522–534, 2018, doi: 10.1007/978-3-319-59427-9\_55.
- [6] M. A. Rahman, R. Sockalingam, M. Othman, K. Biswas, L. Abdullah, and E. A. Kadir, "Nature-inspired metaheuristic techniques for combinatorial optimization problems: Overview and recent advances," *Mathematics*, vol. 9, no. 20, pp. 1–32, 2021, doi: 10.3390/math9202633.
- [7] M. Premkumar, P. Jangir, R. Sowmya, H. H. Alhelou, S. Mirjalili, and B. S. Kumar, "Multi-objective equilibrium optimizer: Framework and development for solving multi-objective optimization problems," *J. Comput. Des. Eng.*, vol. 9, no. 1, pp. 24–50, 2022, doi: 10.1093/jcde/qwab065.
- [8] A. Darwish, "Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications," *Futur*.

- Comput. Informatics J.*, vol. 3, no. 2, pp. 231–246, 2018, doi: 10.1016/j.fcij.2018.06.001.
- [9] F. Glover, “Future paths for integer programming and links to artificial intelligence,” *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, 1986, doi: 10.1016/0305-0548(86)90048-1.
- [10] S. Kirkpatrick, C. Gelatt, and M. Vecchi, “Optimization by Simulated Annealing,” *Science*, vol. 220, pp. 671–680, Jun. 1983, doi: 10.1126/science.220.4598.671.
- [11] X. S. Yang, “Firefly algorithm, Lévy flights and global optimization,” *Res. Dev. Intell. Syst. XXVI Inc. Appl. Innov. Intell. Syst. XVII*, pp. 209–218, 2010, doi: 10.1007/978-1-84882-983-1\_15.
- [12] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, “A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems,” *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2670–2679, 2015, doi: 10.1016/j.eswa.2014.11.009.
- [13] M. Yazdani and F. Jolai, “Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm,” *J. Comput. Des. Eng.*, vol. 3, no. 1, pp. 24–36, 2016, doi: 10.1016/j.jcde.2015.06.003.
- [14] J. Xu and J. Zhang, “Exploration-exploitation tradeoffs in metaheuristics: Survey and analysis,” *Proc. 33rd Chinese Control Conf. CCC 2014*, pp. 8633–8638, 2014, doi: 10.1109/ChiCC.2014.6896450.
- [15] K. Hussain, M. N. M. Salleh, S. Cheng, and Y. Shi, “On the exploration and exploitation in popular swarm-based metaheuristic algorithms,” *Neural Comput. Appl.*, vol. 31, no. 11, pp. 7665–7683, 2019, doi: 10.1007/s00521-018-3592-0.
- [16] H. M. Pandey, A. Chaudhary, and D. Mehrotra, “A comparative review of approaches to prevent premature convergence in GA,” *Appl. Soft Comput. J.*, vol. 24, pp. 1047–1077, 2014, doi: 10.1016/j.asoc.2014.08.025.
- [17] D. Wolpert and W. Macready, “No Free Lunch Theorems for Optimization,” *Evol. Comput. IEEE Trans.*, vol. 1, pp. 67–82, Jan. 1997.
- [18] R. Storn and K. Price, “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces,” *J. Glob. Optim.*, vol. 11, pp. 341–359, Jan. 1997, doi: 10.1023/A:1008202821328.
- [19] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, “GSA: A Gravitational Search Algorithm,” *Inf. Sci. (Ny)*, vol. 179, no. 13, pp. 2232–2248, 2009, doi: 10.1016/j.ins.2009.03.004.
- [20] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.
- [21] S. Mirjalili, “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,” *Knowledge-Based Syst.*, vol. 89, pp. 228–249, 2015, doi: 10.1016/j.knsys.2015.07.006.
- [22] R. Eberhart and J. Kennedy, “New optimizer using particle swarm theory,” *Proc. Int. Symp. Micro Mach. Hum. Sci.*, pp. 39–43, 1995, doi: 10.1109/mhs.1995.494215.
- [23] S. Mirjalili and A. Lewis, “The Whale Optimization Algorithm,” *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016, doi: 10.1016/j.advengsoft.2016.01.008.
- [24] J. H. Holland, “Genetic algorithms,” *Sci. Am.*, vol. 267, no. 1, pp. 66–72, 1992, doi: 10.1038/scientificamerican0792-66.
- [25] D. Simon, “Biogeography-Based Optimization,” *Evol. Comput. IEEE Trans.*, vol. 12, pp. 702–713, Jan. 2009, doi: 10.1109/TEVC.2008.919004.
- [26] V. Hayyolalam and A. A. Pourhaji Kazem, “Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems,” *Eng. Appl. Artif. Intell.*, vol. 87, no. July 2019, p. 103249, 2020, doi: 10.1016/j.engappai.2019.103249.
- [27] M. Dorigo, V. Maniezzo, and A. Colomi, “Ant system: Optimization by a colony of cooperating agents,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 26, no. 1, pp. 29–41, 1996, doi: 10.1109/3477.484436.
- [28] A. W. Mohamed, A. A. Hadi, and A. K. Mohamed, “Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm,” *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 7, pp. 1501–1529, 2020, doi: 10.1007/s13042-019-01053-x.
- [29] R. V. Rao, V. J. Savsani, and D. P. Vakharia, “Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems,” *CAD Comput. Aided Des.*, vol. 43, no. 3, pp. 303–315, 2011, doi: 10.1016/j.cad.2010.12.015.
- [30] A. Kaveh and S. Talatahari, “A novel heuristic optimization method: Charged system search,” *Acta Mech.*, vol. 213, no. 3–4, pp. 267–289, 2010, doi: 10.1007/s00707-009-0270-4.
- [31] H. S. Hosseini, “Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation,” *Int. J. Comput. Sci. Eng.*, vol. 6, no. 1/2, p. 132, 2011, doi: 10.1504/ijcse.2011.041221.
- [32] V. K. Patel and V. J. Savsani, “Heat transfer search (HTS): A novel optimization algorithm,” *Inf. Sci. (Ny)*, vol. 324, pp. 217–246, 2015, doi: 10.1016/j.ins.2015.06.044.
- [33] F. F. Moghaddam, R. F. Moghaddam, and M. Cheriet, “Curved Space Optimization: A Random Search based on General Relativity Theory,” pp. 1–16, 2012, [Online]. Available: <http://arxiv.org/abs/1208.2214>.

- [34] M. Abdechiri, M. R. Meybodi, and H. Bahrami, "Gases brownian motion optimization: An algorithm for optimization (GBMO)," *Appl. Soft Comput. J.*, vol. 13, no. 5, pp. 2932–2946, 2013, doi: 10.1016/j.asoc.2012.03.068.
- [35] R. Formato, "Central force optimization algorithm," *Intell. Syst. Ref. Libr.*, vol. 62, pp. 333–337, 2014, doi: 10.1007/978-3-319-03404-1\_19.
- [36] B. D. Pickett, J. R. Glass, P. G. Ridge, and J. S. K. Kauwe, "Genome of a Giant ( Trevally ): *Caranx ignobilis*," 2021.
- [37] E. M. Abdussamad, H. M. Kasim, and T. S. Balasubramanian, "Distribution, biology and behaviour of the giant trevally, *Caranx ignobilis* - a candidate species for mariculture," *Bengladesh J. Fish. Res.*, vol. 12, no. 1, pp. 89–94, 2008, [Online]. Available: [http://eprints.cmfri.org.in/7554/1/585-BANGLADESH\\_J.FISH\\_RES.\\_2008.pdf](http://eprints.cmfri.org.in/7554/1/585-BANGLADESH_J.FISH_RES._2008.pdf).
- [38] C. G. Meyer, M. Ecol, K. Holland, P. Ser, and Y. Papastamatiou, "*Caranx ignobilis*," *Mar. Ecol. Prog. Ser.*, vol. 333, pp. 13–25, 2007.
- [39] B. M. Wetherbee, K. N. Holland, C. G. Meyer, and C. G. Lowe, "Use of a marine reserve in Kaneohe Bay, Hawaii by the giant trevally, *Caranx ignobilis*," *Fish. Res.*, vol. 67, no. 3, pp. 253–263, 2004, doi: 10.1016/j.fishres.2003.11.004.
- [40] P. F. Major, "Predator-prey interactions in two schooling fishes, *Caranx ignobilis* and *Stolephorus purpureus*," *Anim. Behav.*, vol. 26, no. PART 3, pp. 760–777, 1978, doi: 10.1016/0003-3472(78)90142-2.
- [41] X. S. Yang, T. O. Ting, and M. Karamanoglu, "Random walks, Lévy flights, Markov chains and metaheuristic optimization," *Lect. Notes Electr. Eng.*, vol. 235 LNEE, no. January 2014, pp. 1055–1064, 2013, doi: 10.1007/978-94-007-6516-0\_116.
- [42] M. Chawla and M. Duhan, "Levy Flights in Metaheuristics Optimization Algorithms—A Review," *Appl. Artif. Intell.*, vol. 32, no. 9–10, pp. 802–821, 2018, doi: 10.1080/08839514.2018.1508807.
- [43] N. E. Humphries *et al.*, "Environmental context explains Lévy and Brownian movement patterns of marine predators," *Nature*, vol. 465, no. 7301, pp. 1066–1069, 2010, doi: 10.1038/nature09116.
- [44] D. W. Sims *et al.*, "Scaling laws of marine predator search behaviour," *Nature*, vol. 451, no. 7182, pp. 1098–1102, 2008, doi: 10.1038/nature06518.
- [45] C. A. Bennett, *Principles of physical optics*. John Wiley & Sons, 2022.
- [46] M. Jamil and X. S. Yang, "A literature survey of benchmark functions for global optimisation problems," *Int. J. Math. Model. Numer. Optim.*, vol. 4, no. 2, pp. 150–194, 2013, doi: 10.1504/IJMMNO.2013.055204.
- [47] D. Karaboga and B. Akay, "A comparative study of Artificial Bee Colony algorithm," *Appl. Math. Comput.*, vol. 214, no. 1, pp. 108–132, 2009, doi: 10.1016/j.amc.2009.03.090.
- [48] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in statistics*, Springer, 1992, pp. 196–202.
- [49] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art," *Comput. Methods Appl. Mech. Eng.*, vol. 191, no. 11–12, pp. 1245–1287, 2002, doi: 10.1016/S0045-7825(01)00323-1.
- [50] W. Zhao, L. Wang, and S. Mirjalili, "Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications," *Comput. Methods Appl. Mech. Eng.*, vol. 388, p. 114194, 2022, doi: 10.1016/j.cma.2021.114194.
- [51] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Futur. Gener. Comput. Syst.*, vol. 97, pp. 849–872, 2019.
- [52] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, and W. Zhao, "Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 114, no. June, p. 105082, 2022, doi: 10.1016/j.engappai.2022.105082.
- [53] E. Trojovska, M. Dehghani, and P. Trojovsky, "Zebra Optimization Algorithm: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm," *IEEE Access*, vol. 10, pp. 49445–49473, 2022, doi: 10.1109/ACCESS.2022.3172789.
- [54] M. Azizi, S. Talatahari, and A. Giaralis, "Optimization of Engineering Design Problems Using Atomic Orbital Search Algorithm," *IEEE Access*, vol. 9, pp. 102497–102519, 2021, doi: 10.1109/ACCESS.2021.3096726.
- [55] A. Cheraghali, M. Hajiaghayi-Keshteli, and M. M. Paydar, "Tree Growth Algorithm (TGA): A novel approach for solving optimization problems," *Eng. Appl. Artif. Intell.*, vol. 72, no. April, pp. 393–414, 2018, doi: 10.1016/j.engappai.2018.04.021.
- [56] D. Tang, S. Dong, Y. Jiang, H. Li, and Y. Huang, "ITGO: Invasive tumor growth optimization algorithm," *Appl. Soft Comput. J.*, vol. 36, pp. 670–698, 2015, doi: 10.1016/j.asoc.2015.07.045.
- [57] Z. Li and V. Tam, "A Novel Meta-Heuristic Optimization Algorithm Inspired by the Spread of Viruses," no. Zhixi Li, 2020, [Online]. Available: <http://arxiv.org/abs/2006.06282>.
- [58] H. A. Alsattar, A. A. Zaidan, and B. B. Zaidan, "Novel meta-heuristic bald eagle search optimisation algorithm," *Artif. Intell. Rev.*, vol. 53, no. 3, pp. 2237–2264, 2020, doi:

10.1007/s10462-019-09732-5.

- [59] J. O. Agushaka, A. E. Ezugwu, and L. Abualigah, "Dwarf Mongoose Optimization Algorithm," *Comput. Methods Appl. Mech. Eng.*, vol. 391, p. 114570, 2022, doi: 10.1016/j.cma.2022.114570.
- [60] M. Dehghani, S. Hubalovsky, and P. Trojovský, "Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems," *IEEE Access*, vol. 9, pp. 162059–162080, 2021, doi: 10.1109/ACCESS.2021.3133286.
- [61] K. Zervoudakis and S. Tsafarakis, *A global optimizer inspired from the survival strategies of flying foxes*. 2022.
- [62] Y. Zhang and Z. Jin, "Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems," *Expert Syst. Appl.*, vol. 148, p. 113246, 2020, doi: 10.1016/j.eswa.2020.113246.
- [63] Q. Zhang, R. Wang, J. Yang, K. Ding, Y. Li, and J. Hu, "Collective decision optimization algorithm: A novel heuristic optimization method," *Neurocomputing*, vol. 221, pp. 123–137, 2017, doi: 10.1016/j.neucom.2016.09.068.
- [64] J. Zhang, M. Xiao, L. Gao, and Q. Pan, "Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems," *Appl. Math. Model.*, vol. 63, pp. 464–490, 2018, doi: 10.1016/j.apm.2018.06.036.
- [65] M. Azizi, "Atomic orbital search: A novel metaheuristic algorithm," *Appl. Math. Model.*, vol. 93, pp. 657–683, 2021, doi: 10.1016/j.apm.2020.12.021.
- [66] S. Talatahari, M. Azizi, M. Tolouei, B. Talatahari, and P. Sareh, "Crystal Structure Algorithm (CryStAl): A Metaheuristic Optimization Method," *IEEE Access*, vol. 9, pp. 71244–71261, 2021, doi: 10.1109/ACCESS.2021.3079161.
- [67] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Futur. Gener. Comput. Syst.*, vol. 101, pp. 646–667, 2019, doi: 10.1016/j.future.2019.07.015.

- [68] T. Sang-To, M. Hoang-Le, M. A. Wahab, and T. Cuong-Le, "An efficient Planet Optimization Algorithm for solving engineering problems," *Sci. Rep.*, vol. 12, no. 1, pp. 1–18, 2022, doi: 10.1038/s41598-022-12030-w.



**HAVAL TARIQ SADEEQ** received the B.Sc. degree in computer sciences from the University of Duhok, Iraq, in 2009, the M.Sc. degree in computer sciences, from the faculty of science, University of Zakho, Iraq, in 2014. He is currently pursuing the Ph.D. degree with the Technical College of Informatics-Akre, Duhok Polytechnic University. His current research interests include soft computing, metaheuristics, swarm intelligence algorithms, stochastic optimization and engineering optimization.



**Adnan Mohsin Abdulazeez**, former president of Duhok Polytechnic University. Professor of Computer Engineering and Science. He received his B.Sc. in Electrical and Electronic Engineering, from University of Technology, Baghdad, in 1993, the M.Sc. in Computer and Control Engineering, from University of Technology, Baghdad, in 1997, and the Ph.D. in Computer Engineering from University of Mosul in 2007. He has been awarded the title of Professor since 2013. Prof. Abdulazeez supervised a lot of number of Ph.D. and M.Sc. students in national and international universities in Iraq and KRG. He has published more than 140 articles in local and international scientific journals. His research interest areas include intelligence system, soft computing, multimedia, network security and coding with FPGA implementation. Also, Prof. Abdulazeez is a reviewer for several accredited international journals.