Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

A new meta-heuristic algorithm of giant trevally for solving engineering problems

Marjan Aliyari *[1]

[1]Department of Mathematics of Ayatollah Borujerdi University, Borujerd, Iran

ABSTRACT

As science and technology is progressing in engineering problems are also getting much more complex. So, solving these problems is of pivotal concern. Besides, the optimal solution among the solutions is of great value. Among them, innovative algorithms inspired by artificial intelligence or the hunting behavior of animals in nature have a special place. In this article, a new algorithm named Giant Trevally Optimizer (GTO) is presented, by simulating the hunting strategy of this type of fish, a novel algorithm with the same title is introduced, which has been examined, and subjected to various tests and criteria. In the performance studies of the GTO algorithm with several efficient meta-heuristic algorithms to find the global optimal solution, fifteen criterion functions having various features along with two hard problems in engineering design were used. The performance of the GTO algorithm has been better than other algorithms.

Keyword: Swarm intelligence algorithm, Exploration, Exploitation, engineering problems.

ARTICLE INFO

Article history:
Research paper
Received 27, February 2023
Received in revised form 12, April 2023
Accepted 16 May 2023
Available online 01, June 2023

AMS subject Classification: 11Y16, 65k10, 68Q25.

*Corresponding author: M. Aliyari. Email: m.aliyari@abru.ac.ir

1 Introduction

With the progress of science and technology, not only solving applied science and engineering problems has become more complicated, but also finding optimal solutions has become one of the serious problems of these problems. Meanwhile, classical optimization methods based on mathematics do not have the limitations of today's problems. One of the most significant features and challenges of these problems is that the solution space is discrete or unknown, in which case mathematical optimization methods cannot be used for solving them. In recent years, the use of meta-heuristic algorithms to solve practical problems has been proposed and welcomed. These algorithms give acceptable answers according to the conditions of each problem, But concerning that these algorithms are sensitive to parameters setting ways applied by users, there is no guarantee for finding globally optimal solutions. Because of their random nature there is no guarantee of finding globally optimal solutions. This is even though that classical optimization methods based on mathematics do not respond to the difficult and complex conditions of today's problems, including the limitations of these methods. Now the question is which one of these algorithms is superior to the other? Is there any meta-heuristic algorithm that is superior to the other? The case of the NFL[15] is an answer to such questions. No Free Lunch Theorem, often abbreviated NFL or NFLT, is a theoretical finding that shows all optimization algorithms perform equally well when their performances are considered over all possible objective functions. The name refers to the saying "There is no such thing as a free lunch", meaning there are no easy shortcuts to success.

We know that one of the important concerns in the optimization process is the possibility of randomness of the search space, which may not always produce a sufficiently optimal solution. As a result, many MAs have been developed by researchers to provide acceptable optimal solutions or at least as optimal as possible. The author of this study were inspired to propose a new optimization method that can provide satisfactory results for a wide range of optimization tasks. The novelty and contribution of this research is in the design of a new MA called Giant Trevally Optimizer (GTO), which is based on the behavior and strategies of giant trevally when hunting seabirds. These novel hunting strategies of foraging moving patterns, choosing the appropriate area in terms of quantity of food, and jumping out of water to attack and catch the prey were the main inspiration in the design of the GTO[10].

In this direction, the GTO algorithm is presented and simulated, and then it is tested to ensure its effectiveness in solving optimization problems in various criteria functions, including unimodal and multimodal with different characteristics. In the following, we use this algorithm to solve several difficult and complex real engineering optimization problems and compare the obtained answers with several strong and efficient meta-heuristic algorithms. This manuscript is organized as follows: Section 2 introduces and briefly describes important types of meta-heuristic algorithms. Section 3 analyzes the behavior of giant trevally and mathematically simulate it. In this section, we also present the flowchart of this algorithm. In section 4, we use benchmark functions and other validation methods for this algorithm, solve several real problems, and compare the performance of this algorithm with several others algorithms, and finally, in section 5, provides some concluding remarks.

2 Metaheuristics Algorithms

Meta-heuristic algorithms (MA) can be categorized as follows:

2.1 EVOLUTIONARY ALGORITHMS (EA)

This type of algorithm is based on the theory of the evolution of different animal species. The most important category of this group is Genetic Algorithm (GA)[6] and Differential Evolution Algorithm (DE)[12], which mainly differ in the process of selecting the next generation.

2.2 SWARM INTELLIGENCE ALGORITHMS (SIA)

This group of algorithms is based on the collective behavior of different animal species. The most significant category of this group is the Particle Swarm Optimization Algorithm (PSO)[5] and the new and useful algorithm called the Reptile Search Algorithm (RSA)[4]. In this group of algorithms, the hunting behavior (prey search) of group animals in nature is simulated. In the other words, finding the prey is the same random solution and the prey itself is the exploitation phase.

2.3 HUMAN-BASED-ALGORITHMS (HA)

Evolutionary computing, Human Based Genetic Algorithm (HBGA) is the same as Genetic Algorithm except that it allows humans to make suggestions to help the evolutionary process. Therefore, an HBGA has human relations for initial values, mutation, and recombination crossover. On the other hand, it can use interfaces to evaluate its price selection. In fact acctually, an HBGA out sources a conventional genetic algorithm to humans. An example of this class of algorithms is the algorithm based on knowledge-sharing acquisition (GSK) [9].

2.4 SCIENCE-BASED-ALGORITHMS (SCA)

Another new category of meta-heuristic algorithms, which has recently been proposed and investigated, is the modeling of physical phenomena or chemical laws, the focus of science-based algorithms (for example, gravity, ion motion, etc.). Gas Brownian Motion Optimization (GBMO) [2], and Charge System Search (CSS) [7] are considered the most significant SCAs.

3 GIANT TREVALLY OPTIMIZATION (GTO)

This section describes a proposed MA inspired by nature and is called Giant Trevally Optimizer (GTO) in its roaming range [8]. The giant trevally was first described by Swedish naturalist Peter Forskell in 1775 based on specimens taken from the Red Sea near Yemen and Saudi Arabia, one of which was designated as the holotype[13]. This giant grows in the warmer months and its peaks vary by region. This fish grows relatively fast and reaches sexual maturity at the age of three years and a length of about 60 cm. The giant trevally is an apex predator in most of its habitats and, it uses intelligent hunting methods, known for hunting individually and in schools (groups).
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. Juveniles can migrate up to 70 kilometers from their home atolls and reefs [3].
It is known that the most major member in the movement of a group is the leader. The strategy of the giant trevally in its hunting is that it first looks for the prey (food) and after choosing it, jumps out of the water and hits the prey (seabirds) with a long jump. Hunting process can be divided into four general parts as follows. Searching for prey, which is mainly seabirds, choosing the correct area for hunting, chasing the prey and, finally attacking the prey. The GTO algorithm is derived from these movements and simulation steps.

3.1 Simulation

Similar to all population-based algorithms, in this algorithm, the optimization process starts with the creation of random initial solutions. At this stage, each giant trevally is a candidate for solving the optimization problem. every member of this community forms a d-dimensional vector, and finally, the entire members form a matrix. Population of this algorithm can be described as a matrix (3.1).

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots x_{1,d} \\ \vdots & & \vdots & \vdots \\ x_{i,1} & \cdots & x_{i,j} & \cdots x_{i,d} \\ \vdots & & \vdots & \vdots \\ x_{N,1} & \cdots & x_{N,j} & \cdots x_{N,d} \end{bmatrix} \tag{3.1}$$

Here X is a solution for GTO, d is the number of decision variables and 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. At first, it is necessary to randomly assign positions in the solution spaces to each trevally, we obtain these positions using (3.2) as follows:

$$X_{(i,j)} = L_j + (U_j - L_j) \times R \tag{3.2}$$

Where $j = 1, \ldots, d$, and $i = 1, \ldots, N$, R is a random number in the interval $[0, 1]$. U_j, L_j represents restriction the highest and the lowest value that a population member can have. According to (3.2), a set of these values is stored in the F vector as follows:

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix} = \begin{bmatrix} f(X_1) \\ \vdots \\ f(X_i) \\ \vdots \\ f(X_N) \end{bmatrix} \tag{3.3}$$

$F_i = f(X_i)$ is the value of the i^{th} member in the objective function and F as the set of these values is called the vector of the objective function. As previously stated, the GTO algorithm can be divided into four general parts. In the first stage of prey search (predation) using *Levy* flights, which are a special class of generalized random walks in which the length of the steps during the walk is described by a ”heavy-tailed” probability distribution, then selection of the hunting area, Chasing and jumping out of the water to catch prey. Fig 1 shows trevally jumping out of the water and catching prey.

Figure 1: Giant trevally fish hunting

Considering that giant trevally can travel long distances to find food, their movement can be simulated using the following:

$$X(t+1) = Best_P \times R + [(U - L) \times R + L] \times Levy(d) \tag{3.4}$$

Where $X(t+1)$ is the next iteration giant trevally position vector, $Best_P$ It shows the best position determined in the during their last search, and R is a random number in$[0, 1]$. $Levy(d)$ is a Levy fight, a particular class of non-Gaussian random process whose step size is determined by a Levy distribution [16]. It is important to note that Levy's flight behavior has been shown by various animals as birds and marine wild animals [17], [11] and is calculated by (3.5):

$$Levy(d) = 0.01 \times \frac{u \times \sigma}{\sqrt[\beta]{\mid v \mid}} \tag{3.5}$$

Where, $\beta \in (0, 2)$ is the index of the Levy flight distribution function, which is assigned a value of 1.5 in this study, $u, v \in (0, 1)$ are random numbers. σ is calculated using the following:

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\beta\pi}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}}\right) \tag{3.6}$$

Now in the stage of choosing the right area for hunting, the giant troll must determine the best area for its prey and food. This behavior is simulated as follows:

$$X(t+1) = Best_P \times R \times A + Mean_{Info} - X_i(t) \times R \tag{3.7}$$

Where $X(t + 1)$ is the position vector of giant trevallies in the next iteration t and $A \in [0.3, 0.4]$ is a position change controlling parameter. $X_i(t)$ is the location of the giant trevally i, at time t. Also, $Mean_{Info}$ is the average of the previous information, which indicates that the giant trevally has kept all the information from the previous stages, and of course, it can be calculated as follows [10]:

$$Mean_{Info} = \frac{1}{N} \sum_{i=1}^{N} X_i(t) \tag{3.8}$$

After the trevally is in the correct position for hunting, now it is time for it to come out of the water in this position and attack the bird (prey) with a jump. This is the exploitation (intensification)phose. But the remarkable and significant point is that to simulate this movement in mathematical, we must understand that the trevally in the water is affected by visual waves, which is also due to the refraction of light. It is necessary to remember that when light waves enter the sea from the air, refraction of light occurs. Conforming to Snell's law [1], both the incident ray and the refracted ray form an angle with the ordinary surface at the point of refraction. According to this law, by using refraction coefficients, which are constant values, if we know the radiation angle, we can predict what the refraction angle will be and vice versa. More details can be seen in [10]. The end of this section, the GTO flowchart is shown in Fig2:

4 Algorithm performance in various criteria

In checking the validity of the GTO algorithm performance, two test samples are performed, and the results obtained by GTO are evaluated, and compared with the results

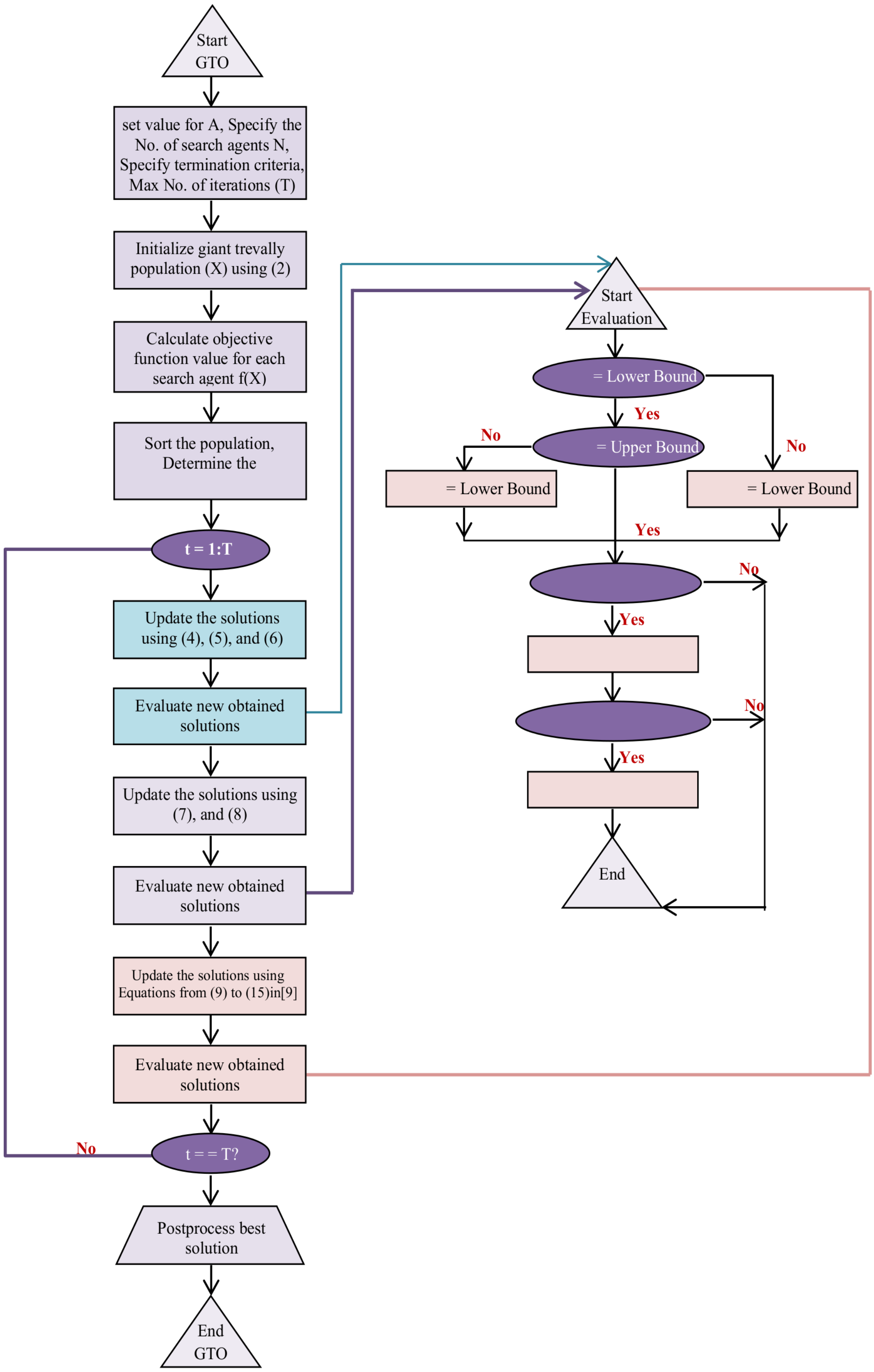

Figure 2: The flowchart of the GTO algorithm.

of other algorithms. Firstly case: The first group test is based on the performance of algorithms based on fifteen benchmark test functions with a variety of features. Secondly case: The GTO algorithm performance is evaluated by using two complicated engineering problems that are important, and widely used. Case 1: Testing Benchmark Functions. Recently several classes of benchmark functions with different characteristics are used. for checking to the performance of algorithms in practice, instead of using them in individual problems. In examining this type of criterion functions, if the performance of the algorithm is acceptable, one can be optimistic about its results in solving real problems. At this stage, to check and compare the GTO algorithm with other algorithms, we use unimodal and multimodal functions, each of which has different characteristics, such as separable and inseparable, and can test the algorithm in different dimensions. In this study, the performance of GTO is compared with seven different meta-heuristic algorithms DE, GSA, WCA(Water Cycle Algorithm), MFO, PSO, and RSA. The calculations of this work have been done in MATLAB 2020 software, and each algorithm is executed 50 times for each function, and the population size, and the number of repetitions are set to 50 and 1000, respectively.
In this test, we compare all the investigated algorithms based on two criteria, the mean "*Mean*" and the standard deviation "*Std*" of the best solution:

$$Mean = \frac{1}{Run} \sum_{i=1}^{R} Best_G \tag{4.1}$$

$$Std = \sqrt{\frac{1}{Run}(Best_G - Mean)^2} \tag{4.2}$$

Where $Best_G$ is the global solution, *Mean* is the average solution obtained in the i^{th} , and *Run* is the number of independent runs. The results obtained from these two criteria for fifteen different benchmark functions in features can be seen in Table 1:
As you can see in Table 1, the GTO algorithm is the most efficient optimization algorithm compared to other competitors and produces good results in terms of average objective functions and standard deviation compared to others. From a statistical point of view, GTO was the most effective for nine functions out of fifteen benchmark functions (1,3, 4, 6, 7, 9, 10, 13, and 14), and in three functions (2, 8, and 12) it was the best result but shared with at least one. For the three remaining benchmark functions, GTO ranked second with the least difference from the best-competing algorithms. Note that, the benchmark functions used are listed in Table 2 of Appendix 1 at the end of this manuscript.
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 functions reported in Table 1. The proposed algorithm consistently ranks first or second in the vast majority of test problems. This is a result of integrated exploration mechanisms in the proposed

N	*In*	*GTO*	*DE*	*GSA*	*WCA*	*MFO*	*PSO*	*RSA*
1	M.	**0**	8.25E-10	1.11E-16	1.57E-5	133.25	1.11E-17	1.10E-16
	Std.	**0**	1.51e-10	2.03e-17	2.94e-60	235.25	2.05e-18	1.96e-17
2	M.	**0**	**0**	**0**.1742	**0**	3352.2310	158.2356	**0**
	Std.	**0**	**0**	**0**.312	**0**	615.23601	28.2312	**0**
3	M.	**4.27E-06**	8.26E-02	0.06521	4.36E-04	1.9001	1.62E-01	3.24E-05
	Std.	**9.2134e-07**	0.01624	0.0101	7.8521e-05	0.3562	0.02541	7.38e-06
4	M.	**9.95E-10**	1.1521E+00	1.52325	1.752E+00	1.6525E+00	9.36E-05	2.69E-02
	Std.	**1.8625e-10**	0.3252	0.28654	0.32536	0.37562	1.5932e-05	6.05e-02
5	M.	-209.952	-202.965	-209.968	-209.895	-17.021	**-209.999**	-208.625
	Std.	0.0086	1.3625	0.0045	7.2531e-04	6.8536	**1.8259e-04**	1.364e-03
6	M.	**2.9652E-08**	27.7021	32.7652	2.72E+01	265833	21.5398	6.025E-01
	Std.	**5.6521e-09**	5.652	5.8625	4.6589	4.3265e+05	3.8958	2.169e-02
7	M.	**0.21754**	1.1251	0.6853	0.6667	51712.21	0.06667	0.6665
	Std.	**0.3981**	0.2054	0.1303	0.1216	9.4526e+03	0.1217	0.1216
8	M.	**0.9979**	**0.9979**	3.9374	2.9521	2.4525	1.6562	1.0231
	Std.	**0**	**0**	0.5395	0.3721	0.2593	0.1203	0.0521
9	M.	**-1114.81**	-11121.6	-2608.56	-6289.32	-8686.21	-3589.23	-1120.31
	Std.	**260.9856**	264.3256	1.8564e+03	1.8186e+03	709.1532	1.6582e+03	262.6321
10	M.	**-10.1526**	-10.1495	-6.5821	-9.3201	-6.04562	-5.3077	-10.1596
	Std.	**5.4852e-05**	6.3501e-04	0.6525	0.1523	0.7496	0.8850	7.226e-05
11	M.	-10.5362	-10.5361	**-10.5364**	-10.2653	-7.9425	-3.7521	-9.8952
	Std.	3.6525e-05	5.4725e-05	0	**0**.0495	0.4725	1.2352	0.0158
12	M.	**-3.8628**	**-3.8628**	**-3.8628**	-3.8616	**-3.8628**	**-3.8628**	**-3.8628**
	Std.	**0**	**0**	**0**	2.2274e-04	**0**	**0**	**0**
13	M.	**-3.32189**	-3.322	-3.3219	-3.2739	-3.2333	-3.2903	-3.2832
	Std.	**3.5785e-04**	3.6524e-04	3.5602e-04	0.0085	0.0158	0.0054	0.0075
14	M.	**0**	0.0002	7.7782	0.0372	9.03195	0.0067	0.0002
	Std.	**0**	3.1038e-05	1.4205	6.7821e-04	1.6490	0.0012	2.014e-05
15	M.	-1.08093	**-1.08094**	-1.05262	**-1.08094**	**-1.08094**	**-1.08094**	-1.0790
	Std.	0	**0**	0.0050	**0**	**0**	0.0848	0

Table 1: The results obtained from seven efficient meta-heuristic algorithms compared to GTO algorithm for different criteria functions.(*In.* Indicator, M. Mean, Std. Standard deviation)

GTO that guide this algorithm in the direction of the optimum global. Fig3 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.
We know that the results of the optimization of the objective functions based on the average and standard deviation indicators provide the possibility of meaningful comparison and evaluation of the optimization algorithms. But, even after several separate runs, it is possible that one algorithm randomly outperforms several algorithms. Hence, a Wilcoxon rank sum test [14] is presented in this section to statistically demonstrate the superiority of GTO over seven competing algorithms. In this test, it is possible to compare two samples in terms of similarity using the Wilcoxon rank sum test, which is a non-parametric statistical test. It can also be determined whether the difference between the two samples is statistically significant or not. In this analysis, a measure called p-value is used to determine whether the corresponding algorithm is significantly better than another algorithm. As you can see in Table 2, when comparing the GTO with other algorithms, the p-value obtained with different dimensional scales shows the superiority of the GTO algorithm over other important and efficient algorithms.

CASE 2: Optimization of engineering design
1. Cantilever beam
One of the important and practical problems in the field of mechanical engineering is the problem of cantilever beams (Fig 4). The goal of solving this problem is to minimize the

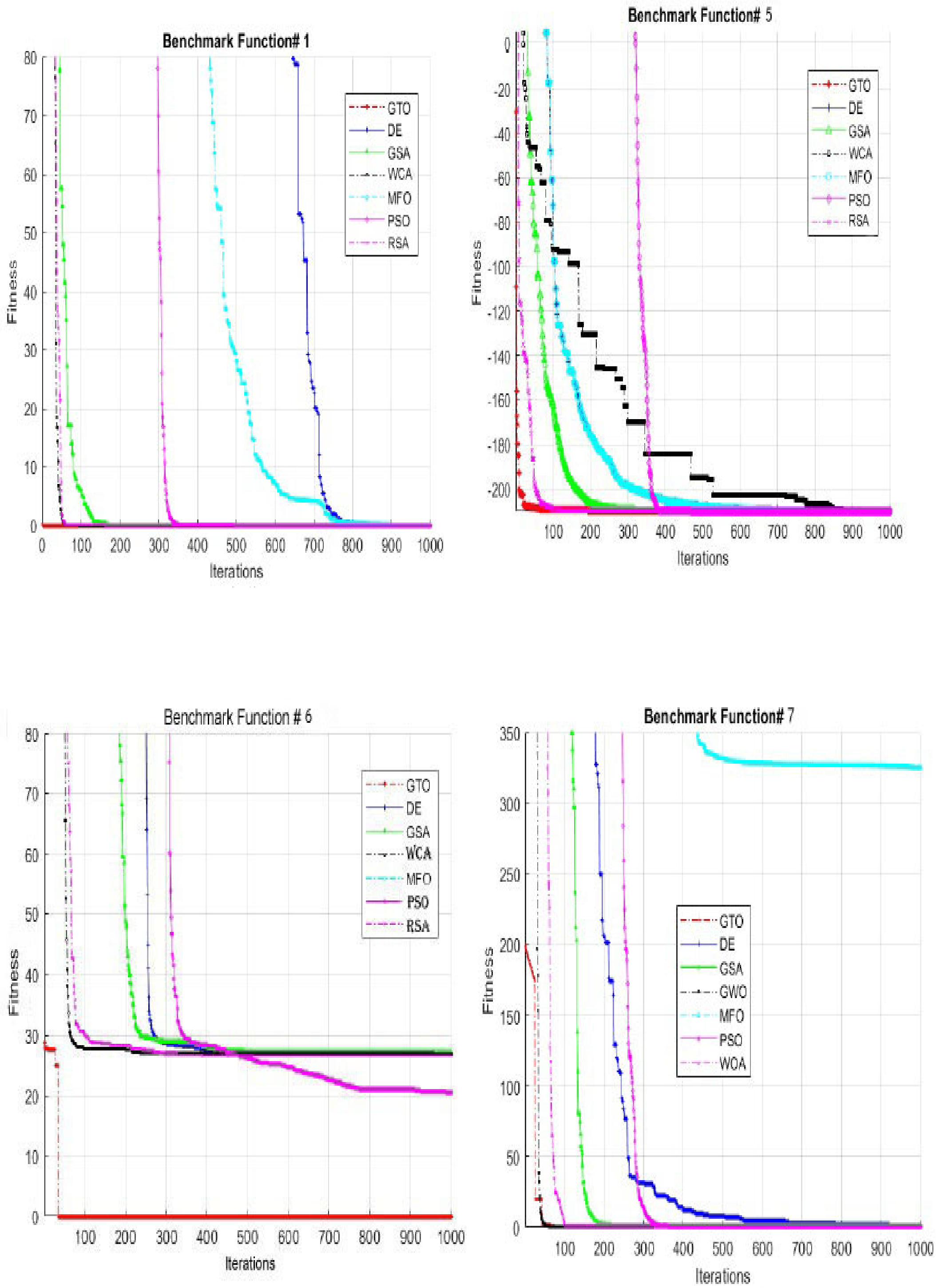

Figure 3: Convergence curve change rate of GTO with other algorithms in a number of benchmark test functions.

GTO	*Dimension*		
vs to	150	700	1200
DE	4.3778E-04	4.3811E-04	4.3878E-04
GSA	6.4304E-04	6.4441E-04	6.4304E-04
WCA	8.5450E-04	1.7090E-03	6.1036E-05
MFO	6.1036E-04	6.1138E-03	6.1045E-04
PSO	9.7255E-03	3.2045E-03	8.3607E-03
RSA	5.0144E-04	5.1102E-04	3.5231E-03

Table 2: p-values obtained from Wilcoxon sum rank test on Table 1 benchmark functions.

weight of the beam as much as possible. The dummy values in this problem are the length of the five bearings seen in Fig 4 [17]. Table 3 shows the average of the best solutions obtained from these seven meta-heuristic algorithms, although the difference between the unknown parameters and the optimal value of the objective function is not large, still the solutions obtained from GTO are better compared to the others.
In addition, Table 4 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.

$$\text{Mimimize: } f(X) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5),$$
$$\text{Subject to: } g(X) = \frac{61}{x_1^3} + \frac{27}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3)} - 10$$
$$0.01 \leq (x_1, x_2, x_3, x_4, x_5) \leq 100$$

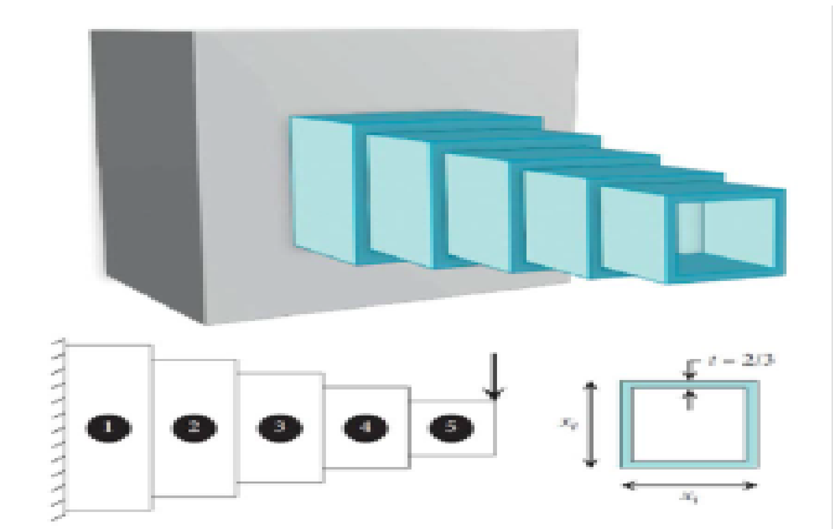

Figure 4: Cantilever beam design problem.

	GTO	*DE*	*GSA*	*WCA*	*MFO*	*PSO*	*RSA*
f_x	**1.3369**	1.3613	1.3399	1.3401	1.3401	1.3400	1.3400
x_1	5.9667	5.9961	5.9716	6.0154	6.0460	6.0164	6.011
x_2	5.3056	5.3608	5.3747	5.3140	5.2960	5.3097	5.349
x_3	4.5020	4.6720	4.4830	4.4910	4.4591	4.4946	4.476
x_4	3.5019	3.5357	3.5030	3.5089	3.5182	3.5005	3.491
x_5	2.1491	2.2513	2.1439	2.1472	2.1559	2.1534	2.145

Table 3: The results of solving the cantilever beam problem.

	GTO	*DE*	*GSA*	*WCA*	*MFO*	*PSO*	*RSA*
Best	**1.3369**	1.3613	1.3399	1.3401	1.3401	1.3400	1.3400
Mean	**1.3367**	1.3595	5.9716	6.0154	6.0460	6.0164	6.011
Worst	1.3368	1.3378	5.3747	5.3140	5.2960	5.3097	5.349
Std.	**2.457e-05**	0.0042	4.6.33e-04	6.2880e-04	7.317e-04	6.239e-04	0.0095

Table 4: Comparison of statistical results of the cantilever beam design.

2. Design of pressure vessels
Another problem in the field of mechanical engineering is related to a cylindrical pressure tank (Figure 5) whose two ends are spherical (such as tanks for transporting fuel and gas concentrates). The purpose of this problem is to minimize the weight of the tank. The variables of this problem are the thickness of the tank, the thickness of the spherical head, the inner radius of the tank, and the length of the cylindrical area, which we represent by x_1, x_2, x_3, and x_4. This problem is simulated as follows in the majority of optimization problems:

$$\begin{gathered}
\text{Minimize } f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\
\text{Subject to: } g_1(X) = -x_1 + 0.0193x_3 \leq 0 \\
g_2(X) = -x_2 + 0.00954x_3 \leq 0 \\
g_3(X) = -\pi x_3^2x_4 - \tfrac{4}{3}\pi x_3^3 + 1296000 \leq 0 \\
g_4(X) = x_4 - 240 \leq 0 \\
0.356 \leq x_1, x_2 \leq 99 \quad , \quad 10 \leq x_3, x_4 \leq 200
\end{gathered}$$

.

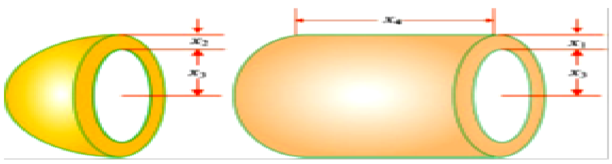

Figure 5: Pressure vessel design problem.

The results obtained from Table 5 show that although the answers obtained from the GTO algorithm are different from other competitors, they are still the best answer compared to them. This different but optimal performance of the GTO will be another confirmation of the superiority of this algorithm over other existing algorithms in this problem. Table 6 verifies the robustness of the proposed algorithm, showing that the best statistical indicators are provided by the GTO.

	GTO	DE	GSA	WCA	MFO	PSO	RSA
f_x	**5887.45**	5932.44	6494.91	6159.77	6370.50	5956.125	5966.76
x_1	0.7787	0.8061	1.0451	1.0451	0.7601	0.85265	0.798
x_2	0.3854	0.3993	0.5221	0.7401	7.2052	0.41153	0.393
x_3	40.3429	41.8136	54.7302	41.035	41.018	42.1382	41.255
x_4	199.64	180.2421	64.7273	176.6336	176.36	164.164	189.11

Table 5: The results obtained from efficient meta-heuristic algorithms compared to GTO algorithm for pressure tank design.

	GTO	DE	GSA	WCA	MFO	PSO	RSA
Best	**5887.45**	5932.44	6494.91	6159.77	6370.50	5956.125	5966.76
Mean	**5967.51**	6196.56	6502.62	6343.26	6601.23	6316.04	6101.26
Worst	6173.25	69.80.69	7432.01	7563.90	7413.01	6953.83	6493.69
Std.	**14.25**	56.06	112.01	83.02	121.05	77.64	47.08

Table 6: Comparison of statistical results of the pressure vessel design.

5 CONCLUSION

In the present, we used a new meta-heuristic algorithm of the swarm type, inspired by the hunting behavior of the giant trevally, while, they have been to optimize several major problems in engineering sciences. This algorithm has been generally divided into four main parts: general search, selection of the hunting area, chasing the prey, and attacking the prey. To check the capabilities of this algorithm in practice (exploration and exploitation), two important test categories have been used. The first test was related to fifteen criterion functions in different spectras (unimodal and multimodal, separable and inseparable). The results obtained from these functions have been compared with several other effective and valid meta-heuristic algorithms. Fortunately, the proposed algorithm has better results according to the mean values, standard deviation values, and the Wilcoxon collective rank test (which is made to ensure uncertainty). The second group of tests was related to hard problems in engineering design optimization, which was, of course, to check the performance of the algorithm for real problems. Two issues were related to the design of the console beams and that of the pressure tank. In these problems, the algorithm provided better answers than other MAs.

References

[1] Bennett, C. A., Principles of Physical Optics. Hoboken, NJ, USA: Wiley, (2022).

[2] Abdechiri, M., Meybodi, M. R., and Bahrami, H., Gases Brownian motion optimization: An algorithm for optimization (GBMO), Appl. Soft Comput, 13 (2013) 2932{2946.

[3] Abdussamad, E. M., Kasim, H. M., and Balasubramanian, T. S., Distribution, biology and behaviour of the giant trevally, Caranx ignobilis A candidate species for mariculture, Bengladesh J. Fisheries Res., 12 (2008) 89{94.

[4] Abualigah, L., Elaziz, M., Sumari, P., Geem, Z., Gandomi, A., REPTILE SEARCH ALGORITHM (RSA): A NATURE –INSPIRED METAHEURISTIC OPTIMIZER, Expert Systems with Applications, 191 (2021).

[5] Eberhart, R., Kennedy, J., New optimizer using particle swarm theory, in Proc. Int. Symp. Micro Mach. Hum. Sci., 3 (1995) 39{43.

[6] Holland, J. H., Genetic algorithms, Sci. Amer., 267 (1992) 66{72.

[7] Kaveh, A. and Talatahari, S., A novel heuristic optimization method: Charged system search, Acta Mech., 213 (2010) 267{289.

[8] Meyer, C. G., Ecol, M., Holland, K., Ser, P., and Papastamatiou, Y., Seasonal and diel movements of giant trevally Caranx ignobilis at remote Hawaiian atolls: Implications for the design of Marine Protected Areas, Mar. Ecol. Prog. Ser., 33 (2007) 13{25.

[9] Mohamed, A. W., Hadi, A. A., and Mohamed, A. K., Gaining-sharing knowledge-based algorithm for solving optimization problems: A novel nature-inspired algorithm, Int. J. Mach. Learn. Cybern., 11 (2020) 1501{1529.

[10] Sadeeq, H. T., Abdulazeez, A. M., Giant Trevally Optimizer (GTO): A Novel Metaheuristic Algorithm for Global Optimization and Challenging Engineering Problems, IEEE Access, 10 (2022) 121615{121640.

[11] Sims, D.W., Southall, E. J.,Humphries, N. E., Hays,G. C., Bradshaw,C. J. A., Pitchford, J. W., James, A., Ahmed, M. Z., Brierley,A. S., Hindell, M. A., Morritt,D., Musyl, M. K., Righton, D., Shepard, E. L. C., Wearmouth,V. J., Wilson, R. P., Witt, M. J., and Metcalfe, J. D. Scaling laws of marine predator search behaviour, Nature, 451 (2008) 1098{11024.

[12] Storn, R., Price, K., Differential evolution:A simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim., 11 (1997) 341{359.

[13] Wetherbee, B.M., Holland, K.N., MeyerC.G., and Lowe,C.G., Use of a marine reserve in Kaneohe bay, Hawaii by the giant trevally, Caranx ignobilis, Fisheries Res., 67 (2004) 253{263.

[14] Wilcoxon, F., Individual comparisons by ranking methods, in Break throughs in Statistics, New York, NY, USA: Springer, 1 (1992) 196{202.

[15] Wolper, D.H., Macready, W.G., No free lunch theorems for optimization, IEEE Trans. Evol. Comput., 1 (1997) 67{82.

[16] Yang, X.S., Ting, T.O., and Karamanoglu,M., Random walks, Lévy flights, Markov chains and metaheuristic optimization, in Future Information Communication Technology and Applications (Lecture Notes in Electrical Engineering), 253 (2014) 1055{1064.

[17] Zhao, W., Wang, L., and Mirjalili,S., Articial hummingbird algorithm: A new bio inspired optimizer with its engineering applications, Comput. Methods Appl. Mech. Eng., 388(2022).

6 APPENDIX

NO.	Name	Function	Opt.	Range	Ch.	Dim
1	Sphere	$f(X)=\sum_{i=1}^{n} x_i^2$	0	[-100,100]	US	30
2	Step	$f(X)=\sum_{i=1}^{n}(\lfloor x_i+0.5\rfloor)^2$	0	[-100,100]	US	30
3	Quartic	$f(X)=\sum_{i=1}^{n} i x_i^4 + random[0,1)$	0	[1.28,1.28]	US	30
4	Colville	$f(X)=100(x_1^2-x_2)^2+(x_1-1)^2+(x_3-1)^2+90(x_3^2-x_4)^2+10.1((x_2-1)^2+(x_4-1)^2)+19.8(x_2-1)(x_4-1)$	0	[-10,10]	UN	2
5	Tried 10	$f(X)=\sum_{i=1}^{Dim}(x_i-1)^2-\sum_{i=2}^{Dim} x_i x_{i-1}$	-210	$[-Dim^2, Dim^2]$	UN	10
6	Rosenbrock	$f(X)=\sum_{i=1}^{Dim-1}[100(x_{i+1}+x_i)^2+(x_i-1)^2]$	0	[-30,30]	UN	30
7	Dixon-Price	$f(X)=(x_1-1)^2+\sum_{i=2}^{n} i(2x_i^2-x_{i-1})^2$	0	[-10,10]	UN	30
8	Foxholes	$f(X)=\left[\frac{1}{500}\sum_{i=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right]^{-1}$	0.998	[-65.536,65.536]	MS	2
9	Schwefel	$f(X)=\sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	12569	[-500,500]	MS	30
10	Bohachevsky 2	$f(X)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)\times 0.4\cos(4\pi x_2)+0.3$	0	[-100,100]	MN	2
11	Shekel5	$f(X)=-\sum_{i=1}^{5}[(x-a_i)(x-a_i)^T+c_i]^{-1}$	-10.1532	[0,10]	MN	4
12	Shekel10	$f(X)=-\sum_{i=1}^{10}[(x-a_i)(x-a_i)^T+c_i]^{-1}$	-10.5364	[0,10]	MN	4
13	Hartman3	$f(X)=-\sum_{i=1}^{4} c_i exp\left[-\sum_{j=1}^{3} a_{ij}(x_j-p_{ij})^2\right]$	-3.8627	[0,1]	MN	3
14	Griewank	$f(X)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	0	[-600,600]	MN	30
15	Langerman2	$f(X)=-\sum_{i=1}^{m} c_i\left(exp\left(\frac{-1}{\pi}\sum_{j=1}^{n}(x_j-a_{ij})^2\right)cos\left(\pi\sum_{j=1}^{n}(x_j-a_{ij})^2\right)\right)$	-1.08	[0,10]	MN	2

*Opt: Optimal solution, Ch: Characteristics, Dim: Dimension, U: Unimodel, M: Multimodel, S: Separable, N: Non-Separable

TABLE 5: Benchmark functions used in Tables 1.