DOI: 10.15849/IJASCA.250730.01

Int. J. Advance Soft Compu. Appl, Vol. 17, No. 2, July 2025
Print ISSN: 2710-1274, Online ISSN: 2074-8523

Bermuda Triangle Optimizer (BTO): A Novel Metaheuristic Method for Global Optimization

Hisham A. Shehadeh

Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Al-Huson, Jordan

e-mail: h.shehadeh@bau.edu.jo

Abstract

Metaheuristic algorithms have garnered a lot of attention in the optimization field. Humans have been using metaheuristic algorithms to solve problems for last decade. Using concepts from natural selection, evolution, and problem-solving techniques, the application of these techniques to combinatorial optimization issues has quickly expanded as a field of study. In this paper, a novel meta-heuristic optimization method is proposed, called "Bermuda Triangle Optimizer (BTO)". The underlying concepts of the farmer one is inspired by the force of attraction of Bermuda, which many aircraft and ships are mysteriously disappeared in this area. The area of attractive force has a form of triangle roughly bounded by Florida, Bermuda, and Puerto Rico in which attracts any object to the center of triangle. Based on this theory, this point can be considered as the best solution (optimal value). BTO is tested in solving test bed suites of "Congress on Evolutionary Computation (CEC 2017)". The BTO is compared against well-known physical based optimization algorithms, which are "Chernobyl Disaster Optimizer (CDO)", and "Gravitational Search Algorithm (GSA)". The outcomes of this study prove the performance of BTO in which can be considered as viable alternative.

Keywords: ***Physical-based algorithms, Metaheuristic algorithms, Optimization, Randomness, Algorithms***

1 Introduction

Optimization is the process of generating a large number of possible candidate solutions to find the best one that will yield the minimal or maximum value for the specified problem [1]. Gradient descent is a well-known technique for tackling derivative optimization problems. Although derivatives can yield precise optimal solutions, their exponential processing cost can make NP-hard problems unsolvable in a short amount of time. As a result, researchers are currently looking at other methods that provide almost perfect outcomes in a reasonable polynomial amount of time.

In the past few decades, algorithmic and artificial intelligence research has focused heavily on metaheuristics. They offer a potent alternative to conventional gradient-based

Received 4 December 2024; Accepted 14 April 2025

mathematical methods for solving difficult optimization problems. One advantage of these approaches is that they can produce almost perfect answers in a manageable length of time. The simplicity, scalability, and flexibility of metaheuristics make them superior to previous methods. Their versatility, broad applicability, and diversity have also sparked research into the development and enhancement of a number of optimization problem-solving techniques. The field of optimization has evolved to address a variety of challenging, high-dimensional issues. This is particularly true in situations where typical gradient-based approaches are insufficient due to issues like non-linearity, discontinuity, or non-convexity in the objective function [1].

Recent, nature-inspired optimization algorithms have a vast range of applications in various domains such as biology, economics, and engineering [1]. These algorithms are developed to solve complex problems efficiently and quickly [2]. Four main categories of these algorithms can be bifurcated to generate an optimal solution to different kinds of problems. These categories can be listed as follows:

- Swarm-based algorithms are inspired by the behavior of swarms in collaborative reproduction and survive. An example of this kind of algorithms is “Sperm Swarm Optimization” [3].
- Physical-based algorithms are inspired by physical principles and theories of the universe. Examples of this kind of algorithms are “Chernobyl Disaster Optimizer (CDO)” [4], and “Gravitational Search Algorithm (GSA)” [5].
- Evolutionary-based algorithms is inspired by Darwin’s theory of evolution. An example of this algorithms is “Genetic Algorithm (GA)” [6].
- Human - based optimization algorithms is inspired by the human behavior, and life style [7]. Example of this category is Harmony Search Algorithm (HAS).

The exploration and exploitation principles should be applied by the aforementioned categories of algorithms in an effort to reach at the global optimal solution. The capacity of an algorithm to identify every aspect of a problem's dimension is known as the exploration principle. Conversely, exploitation describes an algorithm's capacity to arrive at the best possible answer to a problem. Hence, these algorithms strive for equilibrium among the previously listed principles.

Before it can be reviewed about the aforementioned main categories of optimization algorithms CDO, and GSA proses can be stated as follows [4, 8]:

- GSA and CDO are very easy to use and understand.
- GSA and CDO are capable of exploration.
- GSA and CDO may be used to solve parametrical, non-differential, non-continuous, nonparametric, and even multi-dimensional problems.

According to a different perspective, CDO, and GSA have several odds, some of which are as follows [4, 8].:

- The delayed convergence of the GSA and CDO limits their potential for exploitation.
- They face problem of tripping in a local minima of wide search space.

These weaknesses of the aforementioned algorithms are motivated us to propose new optimization algorithm. In this paper, we propose an algorithm, namely “Bermuda Triangle Optimizer (BTO)”, which is inspired by force of attraction of Bermuda. There are many aircraft and ships are mysteriously disappeared in this triangle of Bermuda. The proposed algorithm will be capable to reach the global optima in excellent performance and speed.

Hence, in this article, a test bed problem of "Congress on Evolutionary Computation (CEC 2017)" is used for this purpose, which are used to evaluate the efficiency of the proposed method with the classical CDO and GSA.

The rest of the paper is structured as follows. Background on "Gravitational Search Algorithm (GSA)", and "Chernobyl Disaster Optimizer (CDO)", are discussed in Sec.2. "Bermuda Triangle" and "Bermuda Triangle Optimizer (BTO)" are presented in Sec.3. Experimental and outcomes of the study are presented in Sec.4 We conclude the outcomes in Sec.5.

2 Related Work

In this section, a description of physical based algorithms namely, GSA, and CDO are presented, in which their metaphor, structure, and even their mathematical formulations are elaborated comprehensively.

2.1 "Standard Gravitational Search Algorithm (GSA)"

This approach was created by E. Rashedi et al. as a physical technique that is inspired by Newton's theory and rule [5]. According to Newton's hypothesis, "everything in the universe attracts everything else with a force that is inversely proportional to the square of their distance apart and directly proportional to the product of their masses" [5]. This theory, presupposes a collection of agents in the search space domain, which is simulated by the GSA technique. Every agent in that area creates a gravitational pull on other agent, which the heavier mass pulls the softer ones.

The agents are randomly assigned to the search space domain at the start of the iterations in the GSA method. Depending on that, the following formula is used to determine the gravitational forces between possible solutions (agents) *i* and *j*:

$$F_{ij}^{d}(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon}\left(x_j^d(t) - x_i^d(t)\right), \qquad (1)$$

where,

- M_{aj} – is active mass of gravity of potential solution *j;*
- M_{pi} – is the passive mass of gravity of potential solution *i;*
- G(t) – is the constant of gravitational at time;
- R_{ij} – is the Euclidian distance among two potential solutions *j* and *i.*
- ε – is a constant factor.

The following formula is used to calculate G(t) [5]:

$$G(t) = G_0 \times \exp(-a \times iter/\max iter), \qquad (2)$$

where,

- G_0 – is the potential value that is created initially;
- *iter* – is the iteration of current;
- a – coefficient of descending;
- *maxiter* – is value of final iteration.

The total force that affects a possible solution in dimension *d* of the problem's search space can be computed as follows [5]:

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t), \qquad (3)$$

where,

- $rand_j$ is a random factor between (0, 1).

The acceleration of every possible solution should be calculated using equation 4 [5], which states that "the acceleration of potential solution is proportional to the force and the inverse of its mass" and depends on the motion low.

$$ac_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)}, \qquad (4)$$

where,

- t – is a necessary amount of time;
- M_{ii} – is the prospective solution's inertia mass, i.

The following formula can be used to get the potential solutions' location and velocity:

$$vel_i^d(t+1) = rand_i \times vel_i^d(t) + ac_i^d(t), \qquad (5)$$

$$x_i^d(t+1) = x_i^d(t) + vel_i^d(t+1), \qquad (6)$$

where,

- $rand_i$ is a random value between (0, 1).

2.2 "Standard Chernobyl Disaster Optimizer (CDO)"

The "Chernobyl Disaster Optimizer (CDO)" is inspired by Chernobyl nuclear accident that is happened in Chernobyl in 1986. In this algorithm, Shehadeh [4] simulates the propagation and effects of radiation particles that are omitted from nuclei, which attack humans in order to address optimization challenges. The kinds of particles of radiation that the optimizer takes into account are alpha (α), beta (β), and gamma (γ). These particles and explosion zone are depicted in the Figure 1 [4].

In CDO, Shehadeh assumes that the present locations of the gamma, beta, and alpha particles are $X_\gamma(t)$, $X_\beta(t)$, and $X_\alpha(t)$ respectively. The following models, in that order, provide the gamma, beta, and alpha particle propagation [4]:

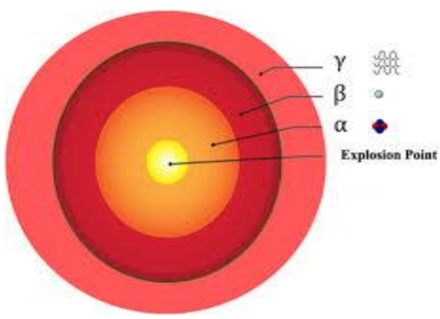


Figure 1: The particles of radiation and explosion point

$$\rho_\gamma = \frac{x_h}{S_\gamma} - (\mathrm{WS}_h \cdot \mathrm{rand}()) \tag{7}$$

$$\rho_\beta = \frac{x_h}{0.5 \cdot S_\beta} - (\mathrm{WS}_h \cdot \mathrm{rand}()) \tag{8}$$

$$\rho_\alpha = \frac{x_h}{0.25 \cdot S_\alpha} - (\mathrm{WS}_h \cdot \mathrm{rand}()) \tag{9}$$

where,

- x_h is the area of human walking within a circle with a random radius between 0 and 1 [4];

$$x_h = r^2 \cdot \pi \tag{10}$$

- S_γ, S_β, and S_α are the normalized random speeds of the gamma, beta and alpha particles respectively [4]:

$$S_\gamma = \log(\mathrm{rand}(1{:}\,300{,}000)) \tag{11}$$

$$S_\beta = \log(\mathrm{rand}(1{:}\,270{,}000)) \tag{12}$$

$$S_\alpha = \log(\mathrm{rand}(1{:}\,160{,}000)) \tag{13}$$

- WS_h is walking speed of human, which is decreased linearly from 3km to 0, defined as Eq (14) [4]:

$$\mathrm{WS}_h = 3 - 1 * ((3) / \text{Maximum\_Iteration}) \tag{14}$$

The difference between the gamma, beta and alpha particles positions and total position can be calculated using the following formulas, respectively [4]:

$$\Delta_\gamma = \left|A_\gamma \cdot X_\gamma(t) - X_T(t)\right| \tag{15}$$

$$\Delta_\beta = \left|A_\beta \cdot X_\beta(t) - X_T(t)\right| \tag{16}$$

$$\Delta_\alpha = |A_\alpha \cdot X_\alpha(t) - X_T(t)|$$

where:

- A_γ, A_β, and A_α are the propagation areas of the gamma, beta, and alpha particles, respectively, represented as the area of a circle with a random radius between 0 and 1as depicted in Figure 1 [4].

$$A_\gamma = A_\beta = A_\alpha = r^2 \cdot \pi \tag{17}$$

- X_T is the average speeds of all particles [4]:

$$X_T = \frac{v_\gamma + v_\beta + v_\alpha}{3} \quad (18)$$

where,

$\boldsymbol{v_\gamma}$, $\boldsymbol{v_\beta}$, and $\boldsymbol{v_\alpha}$ are the Gradient Descent Factors of gamma, beta and alpha particles respectively, which are calculated using the following formulas to find the optimal solution [4]**:**

$$v_\gamma = \left(X_\gamma(t) - \rho_\gamma \cdot \Delta_\gamma\right) \quad (19)$$

$$v_\beta = 0.5 \,.\left(X_\beta(t) - \rho_\beta \cdot \Delta_\beta\right) \quad (20)$$

$$v_\alpha = 0.25 \,.(X_\alpha(t) - \rho_\alpha \cdot \Delta_\alpha) \quad (21)$$

3 Review on Bermuda Triangle

The Triangle of Bermuda [9] is one of our time's most persistent and fascinating mysteries. A number of ships and aircraft are reported to have inexplicably vanished from the Bermuda Triangle, often called the Devil's Triangle, which is located in the western portion of the “North Atlantic Ocean (NAO)”. The Bermuda Triangle's boundaries are generally located by the points of San Juan, Puerto Rico; Miami, Florida; and Bermuda [10]. The Bermuda Triangle is depicted in Figure.2.

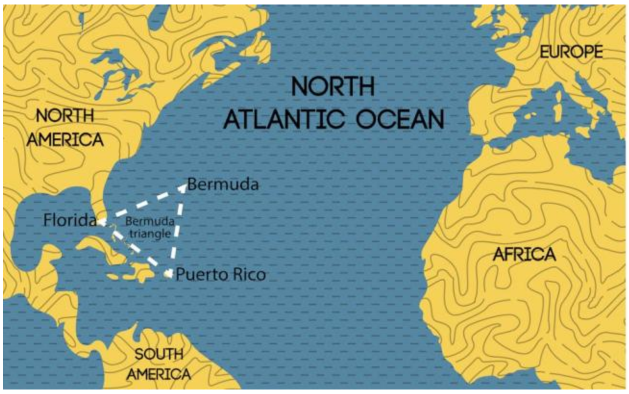


Figure 2: Bermuda Triangle

The most well-known catastrophic events connected to the Bermuda Triangle are [10, 11]:

1. Flight 19: Five U.S. Navy TBM Avenger torpedo bombers were on a training mission when they vanished over the Bermuda Triangle on December 5, 1945.
2. USS Cyclops: A collier, or coal ship, of the U.S. Navy vanished in 1918 while passing through the Bermuda Triangle.
3. Star Tiger: A British South American Airways DC-3 aircraft went missing on January 30, 1948, while flying from Puerto Rico to Miami.

Numerous hypotheses have been put out to explain the enigmatic events in the Bermuda Triangle, including: 1. Magnetic Anomalies [12]: according to some experts, the Bermuda triangle has odd magnetic anomalies that can skew compass readings and interfere with navigational aids. 2. Methane Gas [12]: According to some scientists, ships and aircraft may sink as a result of methane gas bubbles rising from the seafloor. 3. Rogue Waves [13]: According to another idea, the Bermuda triangle frequently experiences rogue waves, sometimes referred to as freak waves, which can be extremely strong and high. 4. Human Error: According to a number of experts, human error such as poor navigation or malfunctioning equipment is to blame for the incidents that have occurred in the Bermuda Triangle.

3.1 Bermuda Triangle Optimizer (BTO)

Based on the prior information, we can notice that The Bermuda Triangle has a massive force that pulls any object near to its area. Based on Einstein's gravitational theory of black holes or any objects that have an attractive force, the Eq(22) can be used as a model of Bermuda Triangle of gravity force [14].

$$G_{force} = \frac{CUG \cdot M_1 \cdot M_2}{r^2} \tag{22}$$

where,

- The CUG – is the constant of universal gravitation, which is 6,67x10^{-11}Nm^{2}Kg^{-2} [15];
- M_1 – is the mass of center of Bermuda that generates the gravitational field, which is a random number;
- M_2 – is that mass that affect by M_1, which is a random number;
- r – is the distance between M_1 *and* M_2, which is a random number.

Based on scientists, the zone of Bermuda force is approximately of the total area range between 500,000 and 1,510,000 square miles [16]. This zone is coefficient calculated by the following equation:

$$Zone_{BF} = area_{\min} + Iter_i * \left(\frac{area_{\max} - area_{\min}}{Iter_T} \right) \tag{23}$$

where,

- $area_{\min}$ – is the minimal value of area of Bermuda force, which is 500000 square miles. We take the logarithm for this value to normalize it;
- $Iter_i$ – is the counter value at the i$_{th}$ iteration;
- $area_{\max}$ – is the maximum value of area of Bermuda force, which is 1,510,000 square miles. We take the logarithm for this value to normalize it;
- $Iter_T$ – is the maximum number of iterations.

The probability ratio of Bermuda force can be calculated by the following equation. In math, the p-value is the probability that the null hypothesis is true. (1 – the p-value) is the probability that the alternative hypothesis is true, as in Eq.(24).

$$PoF = 1 - \left(\frac{Iter_i^{\frac{1}{G_{Force}}}}{Iter_T^{\frac{1}{G_{Force}}}} \right) \quad (24)$$

where,

- $Iter_i$ – is the counter value at the i_{th} iteration;
- $Iter_T$ – is the maximum number of iterations.
- G_{force} – is Eq. (22), which is Bermuda triangle force.

The probability of prescience of any object inside Bermuda triangle or outside the triangle has two values more than 0.5 or less than 0.5 randomly. If the random value is greater than 0.5, the theory of this algorithm will apply the subtraction operation, which means the massive attraction force and can be formulated based on area of Bermuda triangle as depicted in Figure 3. The object inside Bermuda, it already has massive attraction force. In Eq.(24), PoF is (1 – the p-value), which is the probability that the alternative hypothesis is true, that is why we use subtraction operation in Eq.(28) in section 1.On the other hand, if the random value of prescience of any object is less than 0.5, the theory of this algorithm will apply the addition operation, which means the less attraction force. This can be calculated by the subtraction between the Bermuda triangle area and the surrounded area (the yellow circle area) as depicted in Figure 3. Based on that, the object will move to the optimal solution based on Eq.(28). The full procedure of Bermuda Triangle Optimizer (BTO) is presented in Algorithm 1.

Figure 3: Massive attraction force and less attraction force

where,

- LB and UB – are the lower and upper bound of a problem search space;
- $Zone_{BF}$ – the zone of Bermuda force, which can coefficient calculated using Eq.(23)
- *PoF* – is the probability ratio of probability ratio of Bermuda force, which can be calculated based on Eq. (24);
- Best(x_j) – is the best achieved value;
- A_{cc} – is accelerator function that can be calculated using the following formula, which is used to speed up the ocean flow.

$$A_{cc} = r \times e^{(-20 \times (Iter_i / Iter_T))} \quad (25)$$

where,

- r – is a random value;
- e – is exponential value;
- $Iter_i$ – is the counter value at the i_{th} iteration;
- $Iter_T$ – is the maximum number of iterations.

- $Triangle_{area}$ – is the Bermuda area, which has a massive attraction force. This area can be calculated by Eq. (26). This area is depicted in Figure 3 as a black dashed triangle.

$$Triangle_{area} = 0.5 \times r_1 \times r_2 \tag{26}$$

where,

- r_1 – is a random value, which is the base of triangle;
- r_2 – is a random value, which is the height of triangle;

- $Circle_{area}$ – is the area that surrounded by Bermuda, which has less attraction force. This are can be calculated by Eq.(27). This area is depicted in Figure 3 as a yellow circle.

$$Circle_{area} = \pi \times r^2 - Triangle_{area} \tag{27}$$

where,

- r – is a random value, which the radius of circle;
- $Triangle_{area}$ – is the Bermuda area, which is Eq(26).

$$X_{i,j}(Iter_i + 1) = \begin{Bmatrix} \text{Choas} \times \text{Tringle}_{\text{area}} \times \text{Acc} \times best(x_j) - PoF \times ((UB - LB) \times Zone_{BF} + LB), random > 0.5 \\ \text{Choas} \times Circle_{area} \times \text{Acc} \times best(x_j) + PoF \times ((UB - LB) \times Zone_{BF} + LB), \text{otherwise} \end{Bmatrix} \tag{28}$$

BTO uses the levy and chaos methods to simulate the exact movement of attracted objects to the central of Bermuda in which these objects mainly are forced to move in an irregular path by ocean tide and Bermuda force. The levy method is modeled in the following equation, which is used to make a wide variety values in the initialization step. For the chaos methods, we have used a set of methods such as Chebyshev map, Circle map, Gauss/mouse map, Iterative map, Singer map, Sinusoidal map, and Tent map to increase the speed of algorithm and to explore and exploit all the portions of search space domain.

(29)

$$Levy(\alpha) = 0.05 \times \frac{x}{|y|^{1/\alpha}}$$

$$x = Normal(0, \sigma_x^2), \; y = Normal(0, \sigma_x^2)$$

$$\sigma_x = \left[\frac{\Gamma(1+\alpha)\sin\left(\frac{\alpha\pi}{2}\right)}{\Gamma\left(\frac{1+\alpha}{2}\right)\alpha 2^{\frac{(\alpha-1)}{2}}} \right](1/\alpha) and (\sigma_x = 1) dan (\alpha = 1.5)$$

Algorithm 1 *Bermuda Triangle Optimizer (BTO)*
Begin
Step 1: ***1: Initialize the attracted objects in Bermuda.***
2: Initialize the solutions' positions randomly. (Solutions: i=1, ..., N.)
Step 2: while (*the end iteration is not achieved*) **do**
Step 3: *evaluate the Fitness Function for the given solutions*
    Generate the best solution
    Calculate the zone of Bermuda force value using Eq. (23).
    Update the probability ratio of Bermuda force value using Eq. (24).
    for (i=1 to Solutions) do
    for (j=1 to Solutions) do
    Generate a random value between [0, 1], which is the probability of presence the object in Bermuda
    if random >0.5 **then**
    the object is inside Bermuda, which apply Subtraction operator ("− ").
    Update the positions of solutions using the first theory in Eq. (28).
    Else
    the object is outside Bermuda, which apply Addition operator
    (need more attraction force to pull the objects inside Bermuda triangle) (" + ").
    Update the positions of solutions using the second theory in Eq. (28).
    EndIf
Step 4: *update the iteration counter*
End.

4 Experimental and Result

The performance of the proposed "Bermuda Triangle Optimizer (BTO) are estimated using test bed suites of 23 mathematical problems, which are taken from the well-known "Congress on Evolutionary Computation (CEC 2017)" [4]. GSA, and CDO and the proposed BTO are coded in "MATLAB R2023a" and run on Intel core i5 CPU, 8 GB RAM utilizing Windows 11.

The efficiency and quality of results of BTO are compared against CDO and GSA algorithms in terms of best fitness (optimal value), standard deviation (σ), and mean (μ) value. The best fitness values of methods for each benchmark function are recorded in the last iteration of the procedure. The proposed method, CDO and GSA have various parameters in which are initialized at the beginning of the procedures. These parameters are stated in Table 1.

Table 1: Parameters of GSA, CDO and BTO

Parameters	***Value***
	GSA
a – coefficient of descending;	20
G_0 – is the potential value that is created initially	1
Size of population	30
Numbers of iterations/generations	1000
Size of population (swarm size)	30
Numbers of iterations	1000
	CDO
S_γ – is the speed of gamma	Rand (1, 300,000) **km/s**

S_β – is the speed of beta	Rand (1, 270,000) **km/s**
S_α – is the speed of alpha	Rand (1, 16,000) **km/s**
r – is the radius of radiations propagation	Rand (0, 1)
Size of population	30
Numbers of iterations	1000
BTO	
CUG	6,67x10^{-11}Nm^{2}Kg^{-2}
$area_{\min}$	log(500000)
$area_{\max}$	log(1,510,000)
Size of population	30
Numbers of generations	1000

Table 2: GSA, CDO and BTO numerical results of benchmark functions.

Problem Number	GSA	CDO	BTO
	Best fitness	**Best fitness**	**Best fitness**
(1)	1.01e-16	2.29e-262	0
(2)	7.53e-08	2.79e-135	3.13e-310
(3)	5.79e+02	1.83e-226	0
(4)	2.470583	1.52e-126	2.39E-303
(5)	26.83491	27.23930087	8.91678572
(6)	297.666	7.5	0.46577979
(7)	0.072594	3.19e-05	6.06E-05
(8)	-3415.7	-3720.669834	-2056.1316

Table 3: GSA, CDO and BTO statistical results of benchmark functions.

Problem Number	GSA		CDO		BTO	
	μ	σ	μ	σ	μ	σ
(1)	2.11e+2	2725.425	2.81e+03	9218.187043	6.237578e+0	43.35072
(2)	1.13e+7	3.58e+08	2.14e+10	6.1833e+11	402896e+1	2.084922
(3)	1.03e+3	4559.733	8.58e+04	100057.5951	307.9632	1536.95
(4)	3.72e+00	5.20553	1.10e+01	23.72986324	7.365007	18.39352
(5)	2.85e+05	7657691	7.46e+06	26860036.51	38370.17	658704.6
(6)	4.96e+02	3336.817	3.59e+03	11301.18413	408.026	1583.102
(7)	6.82e+00	22.82138	4.43e+00	16.00126141	0.463912	1.488222
(8)	- 3.41e+03	61.87957	- 3.66e+03	109.3767252	-2029.17	101.5579

Table 4: GSA, CDO and BTO numerical results of benchmark functions.

Problem Number	GSA	CDO	BTO
	Best fitness	**Best fitness**	**Best fitness**
(9)	42.8	0.00000	0.00000
(10)	7.59e-09	4.44e-15	4.44E-16
(11)	6.333	0.00000	0.00000
(12)	0.750	1.106102428	1.42830782
(13)	0.0110	0.294755937	0.37245674
(14)	1.995	2.982107311	1.05809603
(15)	2.3e-3	3.1e-4	4.7342e-4
(16)	-1.03	-1.03	-1.03

Table 5: GSA, CDO and BTO statistical results of benchmark functions.

Problem Number	GSA		CDO		BTO	
	μ	σ	μ	σ	μ	σ
(9)	6.89E+01	63.1026	1.82E+02	132.4407183	0.161237	2.128815
(10)	6.31E-01	1.850071	2.79E+00	6.369129417	3.454814	6.567251
(11)	1.44E+01	49.52972	3.60E+01	110.5403913	10.58792	31.13853
(12)	7.87E+05	20086708	1.33E+07	53182172.98	1.02E+08	2.76E+08
(13)	1.64E+06	40544592	2.13E+07	86557208.63	1791279	6579388
(14)	2.43E+00	13.57987	5.42E+00	0.234957768	1.277868	4.675419
(15)	6.21E-03	0.011477	1.27E-03	0.025089272	0.008319	0.019413
(16)	-1.02E+00	0.088165	-9.93E-01	0.102033521	-9.346E-1	0.336537

Table 6: GSA, CDO and BTO numerical results of benchmark functions.

Problem Number	GSA	CDO	BTO
	Best fitness	Best fitness	Best fitness
(17)	0.39	0.39	0.39
(18)	3	3	3
(19)	-3.8	-3.8	-3.8
(20)	-3.32	-3.32	-2.9818014
(21)	-10.15	-9.336480824	-3.784962
(22)	-10.40	-7.940198741	-4.5279703
(23)	-10.53	-8.657742972	-4.0144968

Table 7: GSA, CDO and BTO statistical results of benchmark functions.

Problem Number	GSA		CDO		BTO	
	μ	σ	μ	σ	μ	σ
(17)	4.08e-01	0.043303	4.05e-01	0.088137775	0.4222	0.097661
(18)	3.59e+00	2.782326	5.87e+00	2.496927259	3.835106	4.694067
(19)	-3.85e+00	0.075046	-3.85e+00	0.019301841	-3.7249	0.315734
(20)	-3.24e+00	0.207765	-3.26e+00	0.112827492	-2.9269	0.266177
(21)	-9.14e+00	2.654223	-8.43e+00	1.93810293	-3.19894	1.133138
(22)	-9.40e+00	2.65385	-7.39e+00	1.49386918	-4.0088	1.242694
(23)	-9.38e+00	2.793826	-7.60e+00	1.408646007	-3.46273	1.146176

Table 8: Ranking the approaches from best to worse obtained fitness value

Function	Based on the fitness values, we can rank the methods from best to worse as follows
(1)	BTO, CDO, followed by GSA
(2)	BTO, CDO, followed by GSA
(3)	BTO, CDO, followed by GSA

(4)	BTO, CDO, followed by GSA
(5)	BTO, CDO, followed by GSA
(6)	BTO, CDO, followed by GSA
(7)	BTO, CDO, followed by GSA
(8)	CDO, GSA, followed by BTO
(9)	BTO and CDO in the same rank, followed by GSA
(10)	BTO, CDO, followed by GSA
(11)	BTO and CDO in the same rank, followed by GSA
(12)	GSA, CDO, followed by BTO
(13)	GSA, CDO, followed by BTO
(14)	BTO, GSA, followed by CDO
(15)	GSA, CDO, followed by BTO
(16)	All in the same rank
(17)	All in the same rank
(18)	All in the same rank
(19)	All in the same rank
(20)	GSA, and CDO in the same rank, followed by BTO
(21)	GSA, CDO, followed by BTO
(22)	GSA, CDO, followed by BTO
(23)	GSA, CDO, followed by BTO

The experimental results can be summarized in the tables from two to seven, in which the best optimal values are summarized in the highlighted background. Ten iterations of the tests are conducted to guarantee that the results converge. Statically speaking, the BTO outperforms the other methods in generating the least fitness value of fifteen benchmark functions. GSA is the best on ten benchmark function, and CDO is the best on five benchmark functions. for the mean values, the BTO is the best on nine benchmark functions. GSA is the best on nine benchmark functions. CDO is the best on six benchmark functions. From table 8, we can notice, that BTO superior in solving f1, f2, f3, f4, f5, f6, f7, f14 functions, which are complex and ragged functions. In addition, BTO beats in the same rank with CDO in solving f9, and f10. On the other hand, for functions f16, f17, f18, and f19 all beats in the same rank. The results of convergence rates prove the efficiency and speed of BTO. This is clear in the figures of convergence of f1, f2, f3, f4, f5, f6, f7, f14, f9, and f11.

6 Conclusion

In this paper, we propose a novel metaheuristic optimization algorithm, namely “Bermuda Triangle Optimizer (BTO)”, which is a physical-based optimizer. In BTA, the objects around this area are affected by attraction force of Bermuda Triangle, which are pulled to the central of it. This algorithm is compared against other well-known physical-based optimizers, which are “Chernobyl Disaster Optimizer (CDO)”, and “Gravitational Search Algorithm (GSA)”. The thirty-two benchmark functions of CEC 2017 are utilized to evaluate the proposed algorithm, which contains unimodal, multimodal, and complex mathematical functions. The outcomes show that the BTO can be considered as viable alternative. In the future, we will enhance the BTO by hybridizing it with other physical based or swarm based algorithms to increase its performance and efficiency. In addition, we will apply it to optimize some problems of Unmanned Aerial Vehicle (UAV) and Wireless Sensor Network (WSN).

Ethical Approval & Consent for publication:

We give our consent for the publication of identifiable details, which can include photograph(s) and/or videos and/or case history and/or details within the text ("Material") to be published in the above Journal and Article. We confirm that we have seen and been given the opportunity to read both the Material and the Article (as attached) to be published by your journal. In Addition, a sample of data of this paper will be available upon request. The open source code of our algorithms, namely, CDO and BTO are available via the following links:

CDO: https://www.mathworks.com/matlabcentral/fileexchange/124351-chernobyl-disaster-optimizer-cdo

BTO: https://github.com/sh7adeh1990/BTO

Conflicts of Interest: The authors declare no conflicts of interest.

References

[1] Seyyedabbasi, A., Hu, G., Shehadeh, H. A., Wang, X., & Canatalay, P. J. (2025). V-shaped and S-shaped binary artificial protozoa optimizer (APO) algorithm for wrapper feature selection on biological data. *Cluster Computing*, *28*(3), pp. 163.

[2] Wang, X., Snášel, V., Mirjalili, S., Pan, J. S., Kong, L., & Shehadeh, H. A. (2024). Artificial Protozoa Optimizer (APO): A novel bio-inspired metaheuristic algorithm for engineering optimization. *Knowledge-based systems*, *295*, pp. 111737.

[3] Shehadeh, H. A., Ahmedy, I., & Idris, M. Y. I. (2018, February). Sperm swarm optimization algorithm for optimizing wireless sensor network challenges. In *Proceedings of the 6th international conference on communications and broadband networking* (pp. 53-59).

[4] Shehadeh, H. A. (2023). Chernobyl disaster optimizer (CDO): a novel meta-heuristic method for global optimization. *Neural Computing and Applications*, *35*(15), pp. 10733-10749

[5] E. Rashedi, H. Nezamabadi-Pour, & S. Saryazdi, GSA. (2009). A Gravitational Search Algorithm. *Information sciences*, *179*(13), pp. 2232-2248.

[6] Holland, J.H. (1992) Genetic algorithms. Sci. Am. 267(1), pp. 66–73 .

[7] R. Devaraj, S. K. Mahalingam, B. Esakki, A. Astarita, S. Mirjalili. (2022). A Hybrid GA-ANFIS and F-Race Tuned Harmony Search Algorithm for Multi-response Optimization of Non-Traditional Machining Process. *Expert Systems with Applications*, *199(2022), pp. 116965.*

[8] H. A. Shehadeh. (2021). A Hybrid Sperm Swarm Optimization and Gravitational Search Algorithm (HSSOGSA) for Global Optimization. Neural Comput & Applic, 33(18), 11739-11752.

[9] Group, D. (1984). *The Evidence for the Bermuda Triangle.* Wellingborough, Northamptonshire, UK: Aquarian Press. ISBN: 0-85030-413-X.

[10] Berlitz, C., & Valentine, J. M. (1974). The Bermuda Triangle. New York: Doubleday.

[11] Stewart, G. B. (2009). *The Bermuda Triangle*. Capstone.

[12] Malik, A. A., Siddique, F., Asad, M., & Azeem, W. (2021). Dominant factors having adverse effects on Bermuda Triangle region. Research in Marine Sciences, *6*(3), pp. 1014-1032.

[13] Everhart, C. (2024). *Vortex: The Bermuda Triangle Mystery*. Joy Kitchen University Press.

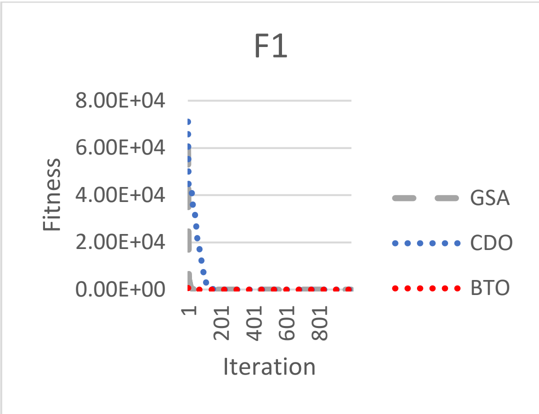
[14] Malek, a. (2017). The Conceptual Defect of The Law of Universal Gravitation or ‘free fall’: a Dialectical Reassessment of Kepler’s Laws. journal of Advances in Physics, 13(5), pp. 4876-4880.

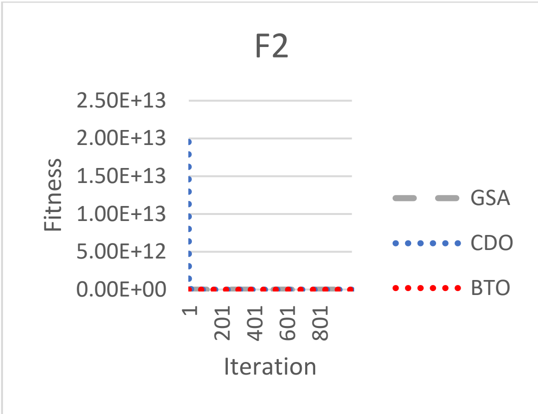
[15] Jaulin, L. (2002, October). Consistency techniques for the localization of a satellite. In *International Workshop on Global Optimization and Constraint Satisfaction* (pp. 157-170). Berlin, Heidelberg: Springer Berlin Heidelberg.

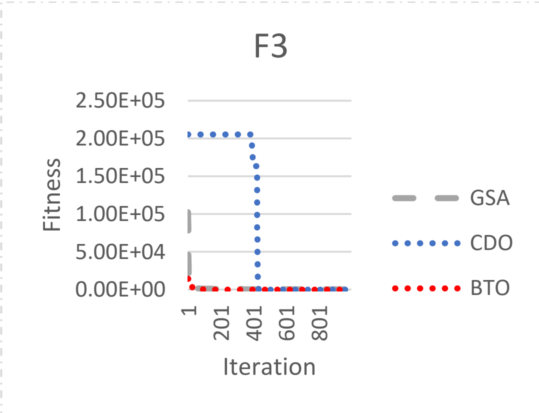
[16] Malik, A. A., Siddique, F., Asad, M., & Azeem, W. (2021). Dominant Factors Having Adverse Effects on Bermuda Triangle Region. *Research in Marine Sciences*, *6*(3), pp. 1014-1032.

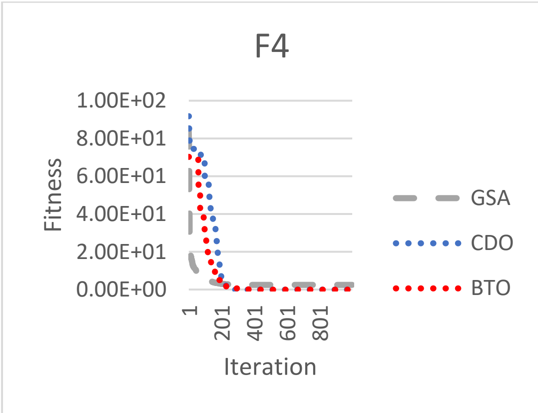
AUTHORS’ INFORMATION

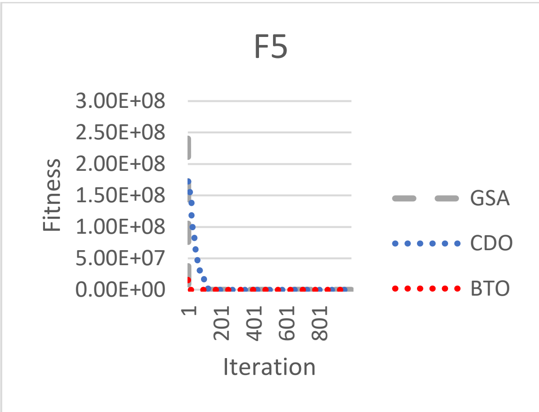
Hisham A. Shehadeh *received the B.S. degree in computer science from Al-Balqa` Applied University, Jordan, in 2012, the M.S. degree in computer science from the Jordan University of Science and Technology, Irbid, Jordan, in 2014, and the Ph.D. degree from the Department of Computer System and Technology, University of Malaya (UM), Kuala Lumpur, Malaysia, in 2018. He was an Assistant professor of CS/AI in Amman Arab University from 2020 to 2024. He was a research assistant at UM from 2017 to2018. He was a Teaching Assistant and a lecturer with CS Department, College of Computer and Information Technology, Jordan University of Science and Technology from 2013 to 2014 and from 2014 to 2016 respectively. Currently, He is an Assistant Professor of Computer Science at the Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Al-Huson, Jordan. His current research interests are, intelligent computing, metaheuristic algorithms and algorithmic engineering applications of WSN.*

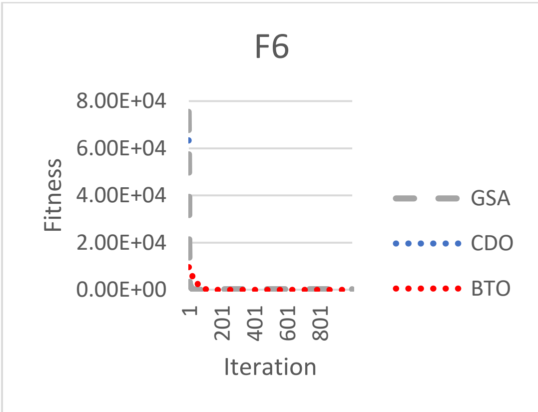

F1
8.00E+04
6.00E+04
4.00E+04
2.00E+04
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO

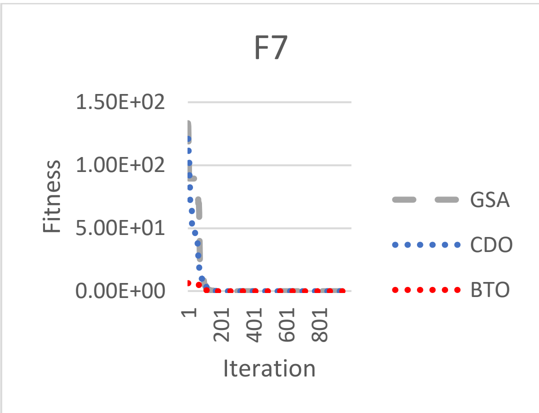

F2
2.50E+13
2.00E+13
1.50E+13
1.00E+13
5.00E+12
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO

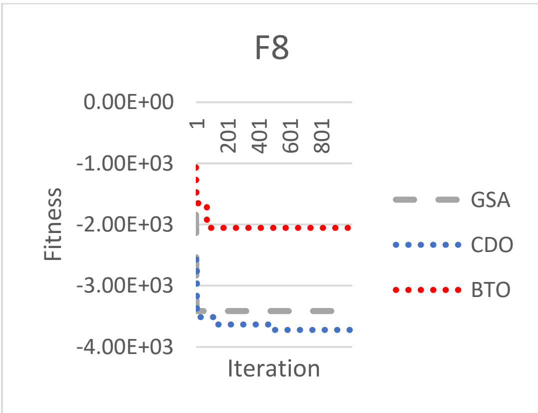

F3
2.50E+05
2.00E+05
1.50E+05
1.00E+05
5.00E+04
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO


F4
1.00E+02
8.00E+01
6.00E+01
4.00E+01
2.00E+01
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO


F5
3.00E+08
2.50E+08
2.00E+08
1.50E+08
1.00E+08
5.00E+07
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO


F6
8.00E+04
6.00E+04
4.00E+04
2.00E+04
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO


F7
1.50E+02
1.00E+02
5.00E+01
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO


F8
0.00E+00
-1.00E+03
-2.00E+03
-3.00E+03
-4.00E+03
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO


F9
5.00E+02
4.00E+02
3.00E+02
2.00E+02
1.00E+02
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO
F10
2.50E+01
2.00E+01
1.50E+01
1.00E+01
5.00E+00
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO
F11
8.00E+02
6.00E+02
4.00E+02
2.00E+02
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO
F12
1.00E+09
8.00E+08
6.00E+08
4.00E+08
2.00E+08
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO
F13
1.50E+09
1.00E+09
5.00E+08
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO
F14
5.00E+02
4.00E+02
3.00E+02
2.00E+02
1.00E+02
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO
F15
1.00E+00
8.00E-01
6.00E-01
4.00E-01
2.00E-01
0.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO
F16
3.00E+00
2.00E+00
1.00E+00
0.00E+00
-1.00E+00
-2.00E+00
Fitness
1
201
401
601
801
Iteration
GSA
CDO
BTO

Figure 4: Convergence rate of GSA, CDO and BTO in solving benchmark functions from 1 to 23.