

# Bombardier Beetle Optimizer (BBO) for Optimizing Engineering Design Problems

Hisham A. Shehadeh<sup>1,\*</sup>, and Widi Aribowo<sup>2</sup>

<sup>1</sup> Department of Computer Sciences, Faculty of Information Technology and Computer Science, Yarmouk University, Irbid 21163, Jordan.

<sup>2</sup> Department of Electrical Engineering, Faculty of Vocational, Universitas Negeri Surabaya, Surabaya, Indonesia.

\* Corresponding author: h.shehadeh@yu.edu.jo

Received: 3 May 2026 Accepted: 29 August 2026 Published: 12 October 2026

DOI: 10.15849/ijasca.168 License: Open access under CC BY 4.0. Authors retain copyright.

Publisher: Al-Zaytoonah University of Jordan (ZUJ), Amman, Jordan.

Cite this article: H. A. Shehadeh and W. Aribowo, "Bombardier beetle optimizer (BBO) for optimizing engineering design problems," *Int. J. Advance Soft Compu. Appl.*, vol. 18, no. 3, pp. 221–246, 2026, doi: <https://doi.org/10.15849/ijasca.168>.

## ABSTRACT

This paper provides a comprehensive comparative study of various metaheuristic optimization algorithms with the intention of emphasizing the most adequate method for optimizing well-established design problems of engineering. The algorithms chosen for this study include "Bombardier Beetle Optimizer (BBO)", "Particle Swarm Optimization (PSO)", and "Grey Wolf Optimizer (GWO)". The experimental results of this paper are extracted in two phases. In the first phase, these algorithms are utilized to optimize the "Congress on Evolutionary Computation (CEC) 2017" test bed suite. While in the second phase, they are applied to solve the well-known problems of engineering design, such as Speed Reducer Design, Pressure Vessel Design, Cantilever Beam Design, and Robot Gripper optimization problems. Moreover, two techniques for evaluating the proposed algorithms are selected in this paper, which are quantitative and qualitative analyses. For the former one, a set of key performance indicators are used to measure solution quality, which are mean, best fitness, and standard deviation. While for the later one, the convergence curves are drawn to estimate the efficiency of algorithms over iterations. This helps for a robust estimation of each algorithm's ability to poise exploration and exploitation in the domain of any problem. These case studies provide more evidence of the algorithms' versatility and applicability in managing challenging, multifaceted, and limited optimization jobs. The outcomes show that BBO has the merit in optimizing these problems.

**Keywords:** Randomness; Optimization; Bombardier Beetle; CEC test bed suites; Engineering design problems.

## 1 Introduction

The process of creating a list of potential candidate solutions and selecting the one that either minimizes or maximizes the desired value for a specific problem is called optimization [1]. Over the past years, metaheuristics have become a hot topic in artificial intelligence researches. They serve as a powerful alternative to mathematical methods of traditional gradient-based selection for solving complex optimization problems. One of their key advantages is their capacity to produce nearly perfect solutions in a manageable computational time [2].

Compared to conventional approaches, metaheuristic principles stand out for their simplicity, scalability, and flexibility. Their versatility and broad applicability have encouraged extensive research into developing and enhancing optimization techniques for various problem domains. As a result, optimization methods become the key for solving various hard and complex problems. These methods can be listed into four groups, such as manmade-based, swarm-based, physical-based, and evolutionary-based optimization algorithms [3]. These groups along with examples on them are listed in table 1.

Table 1. Acronyms and citations for metaheuristic algorithms

| Category      | Algorithm                  | Acronym | Citation                               |
|---------------|----------------------------|---------|----------------------------------------|
| Physics-based | Bermuda Triangle Optimizer | BTO     | (Shehadeh <i>et al.</i> , 2025)<br>[4] |

|                      |                                     |      |                                          |
|----------------------|-------------------------------------|------|------------------------------------------|
| Swarm-based          | Atom Search Optimization            | ASO  | (Zhao <i>et al.</i> , 2025) [5]          |
|                      | Chernobyl Disaster Optimizer        | CDO  | (Shehadeh <i>et al.</i> , 2023) [6]      |
|                      | Sand Cat Swarm Optimization         | SCSO | (Seyyedabbasi <i>et al.</i> , 2023) [7]  |
|                      | Sperm Swarm Optimization            | SSO  | (Shehadeh <i>et al.</i> , 2018) [8]      |
|                      | Greylag Goose Optimization          | GGO  | (El-Kenawy <i>et al.</i> , 2024) [9]     |
| Evolutionary based   | Elk herd optimizer                  | EHO  | (Al-Betar <i>et al.</i> , 2024) [10]     |
|                      | Synergistic fibroblast optimization | SFO  | (Dhivyaprabha <i>et al.</i> , 2017) [11] |
|                      | Differential evolution              | DE   | (Storn & Price, 1997) [12]               |
| Manmade optimization | Fireworks Algorithm                 | FA   | (Tan <i>et al.</i> , 2010) [13]          |
|                      | Harmony Search                      | HS   | (Geem <i>et al.</i> , 2001) [14]         |

In this paper, the GWO [15], PSO [16], and a recently proposed bio-inspired approach known as the BBO [17] are utilized to optimize a range of engineering optimization problems due to their notable advantages, which are listed as follows:

1. BBO is one of the latest bio-inspired optimizers, which employs random tuning parameters in its procedure representing exact defense and escape mechanisms. This helps it to escape from the local minima easily.
  2. PSO are simple to apply and understand.
  3. GWO is a powerful and efficient method for solving complex optimization problems.
- In addition to that, the authors in [17] proved the ability of BBO in solving various types of mathematical problems, however, it is not utilized in solving real life problems such as engineering design problems, based on that we are motivated to use it in this research. The rest of the paper is organized as follows: Sec.2 shows the review on the selected optimization algorithms. Sec.3. presents the experimental and results. We wrap up the results in section 4.

## 2 Review on Bombardier Beetle Optimizer (BBO), Particle Swarm Optimization (PSO), and Grey Wolf Optimizer (GWO)

In this paper, we choose BBO, PSO, and GWO to review on them and to utilize them in order to optimize a few well-known engineering designing problems.

### 2.1 Bombardier Beetle Optimizer (BBO)

This method is proposed by Shehadeh *et al.* [17] in 2026. BBO is a novel bio-inspired metaheuristic algorithm, which is developed for global optimization. The principles of BBO is drawn around the defense and escape mechanisms of the Bombardier Beetle when it feels threatened. In the defense mechanism of the Bombardier Beetle, a toxic chemical spray can be triggered from its back nozzle within its belly in a dedicated reaction chamber. The well-controlled enzymatic mechanism of this reaction consists of oxygen, p-benzoquinones (irritating substances), and hot water vapor. The bombardier beetle is seen in figure 1 (a) as it activates its chemical defense mechanism. The Bombardier Beetle's abdomen and specialized response chambers are seen in figure 1 (b). The algorithm is structured around three main phases, which are listed in the following points:

## A. Initialization Phase:

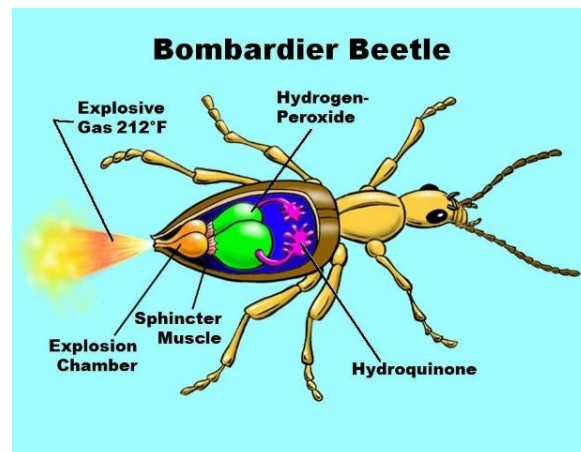
The method starts its procedure by setting a population of  $N$  Bombardier Beetles, which each one of them represents a vector of potential solution in search space of  $m$ -dimension. Each dimension  $d$  of a beetle's position  $x_{i,d}$  is initialized randomly within the problem's lower ( $lb_d$ ) and upper ( $ub_d$ ) bounds using the following formula [17]:

$$x_{i,d} = lb_d + r \cdot (ub_d - lb_d) \quad (1)$$

where  $r$  is a value, which is set between 0 and 1 randomly.



(a)



(b)

Figure 1. The bombardier beetle is seen in (a) as it activates its chemical defense mechanism. The Bombardier Beetle's abdomen and specialized response chambers are seen in (b). [17]

## B. Defense mechanism (exploration phase)

This stage models the beetle's toxic chemical spray as a defensive mechanism of Bombardier Beetles, which the reaction takes place within a specialized chamber in the beetle's abdomen and involves a precisely controlled enzymatic process. This process generates a mixture of hot water vapor, oxygen, and irritating chemical components such as p-benzoquinones. This toxic chemical spray will be triggers directly to the predator when Bombardier Beetles feels threatened, which there are two cases of three of activating this mechanism. These cases are listed as follows [17]:

1. The area of Bombardier Beetles does not intersect the area of predator, which this mechanism is inhibited;
2. Its defensive zone intersects the area of predator, which this mechanism is activated;

3. Its defensive zone overlaps the area of predator, which this mechanism is activated.

From a mathematical perspective, the following formula is utilized to settle the area of intersection between couple of circles, which represents the region where the spray of toxic chemical is dispersed [17].

$$\begin{aligned}
 A &= A(R, d_1) + A(r, d_2) \\
 &= r^2 \cos^{-1} \left( \frac{d^2 + r^2 - R^2}{2dr} \right) \\
 &\quad + R^2 \cos^{-1} \left( \frac{d^2 + R^2 - r^2}{2dR} \right) \\
 &\quad - \frac{1}{2} \sqrt{(-d + r + R) \times (d + r - R) \times (d - r + R) \times (d + r + R)}.
 \end{aligned} \quad (2)$$

where:

1. R denotes the radius of the first circle, representing the Bombardier beetle's defensive area;
2. r denotes the predator's area, which is the second circle radius;
3. d denotes the distance among the centroids of the two circles.

In order to activate its defense system, the Bombardier Beetle locates the predator. Equation (3) computes a new position for each individual of the population based on the Bombardier Beetle's location update rule and the region where the beetle and the predator intersect. According to Eq. (4) [17], the new location takes the place of the old one if its objective function value is superior.

$$x_{i,d}^{PBB} = \frac{(x_{i,d} + (SelectePredator_{i,d} \times CIA \times CR \cdot x_{i,d}))}{Spray} \quad (3)$$

$$X_i = \begin{cases} X_i^{P1}, & F_i^{Obj} < F_i \\ X_i, & \text{else} \end{cases} \quad (4)$$

where,

1.  $x_{i,d}^{PBB}$  represents the newly possible position of the  $i$ -th Bombardier Beetle in the  $d$ -th dimension;
2.  $SelectePredator_{i,d}$  represents the position of the predator in the  $d$ -th dimension;
3.  $CIA$  is the intersection area of couple of circles between the predator's position and the Bombardier Beetle's position. This value is calculated using Eq. (2).
4.  $CR$  stands for the poisonous spray's chemical reaction, which is made up of the following elements:
  1. The boiling point of water is represented by a value of 100 for hot water vapor;
  2. oxygen is a value in the interval [0, 1] that is taken randomly;
  3. and irritating compounds like p-benzoquinone are represented by a random number in the interval [0, 1].
5. Iteration is the value of current iteration;
6. Max iteration is the procedure's maximum iterations.

7.  $N$  is population amount (Bombardier Beetles);
8.  $m$  is of decision variables mount;
9.  $F_i^{obj}$  is the value of objective function.
10. Spray represents the exponential function describing the spread of the toxic chemical spray, which is calculated using Eq. (5).

$$Spray = \left( chaos \times 2.7^{\left( 100 \left( \frac{iteration}{Max\_iterations} \right) \right)} \right) \quad (5)$$

A variety of models, of chaos maps to mimics the real chaos of deploy the hazardous spray over domain of any problem. The sine map is mostly utilized in BBO's process [17].

### C. Escape mechanism (exploitation phase)

Newton's Theory of Insect Lift is applied in this algorithm's escape phase:

This theory of insect lift is created based on the laws of motion of Newton, which are the second and third laws:

- Newton's second law is:

1. “Force is equal to mass multiplied by acceleration”.

- Newton's third law is:

2. “Action is equal to reaction”.

When an insect flaps its wings, the air is accelerated downward, which creates an upward reactive force, called lift. The lift is depicted in Fig. 2, where the bombardier beetle's wings produce lift to flee and fly away from the predator. The algorithm's exploitation phase is derived from the equation of empirical lift for flapping flight, which is formulized in formula (6). The aerodynamics of an airplane are comparable to this, although they are altered for erratic flapping action. Algorithm 1 lists the Bombardier Beetle Optimizer's (BBO) whole pseudocode [17].

---


$$L = LC * 0.5 * \rho * V^2 * A \quad (6)$$


---

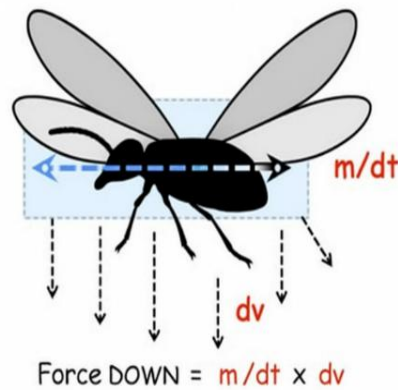


Figure 2. Flapping flight, an upward reactive force of the bombardier Beetle [17]

where,

- LC is the coefficient of lift, which is found experimentally for the motion of insect wings. It takes a random value between 0 and 1;
- $\rho$  is the density of air, which is a random value between 0 and 1;
- V is relative velocity of the wing, which is a random value between 0 and 1;
- A is the area of effective wing, which is a random value between 0 and 1.

---

#### Algorithm 1. Pseudocode of BBO.

##### Start BBO.

1. input the problem's variables, constraints, and objective function.
  2. Determine the population size (N) and BBO maximum iterations number (Max\_Iter).
  3. Using Equation (1), generate the matrix of initial population
  4. Assess the objective function
  5. For  $t = 1$  to Max\_Iter:
    6. For  $i = 1$  to N:
      7. Section # 1: exploration phase of BBO, which is the defense mechanism.
    8. Evaluate the area of intersection among predator and Bombardier Beetle positions using formula (2).
    9. Based on Equation (5), perform defense mechanism by spraying the toxic chemical spray.
    10. Based on Equation (3), calculate the population's new positions of and update the solution.
    11. Section #2: exploitation phase of the algorithm, which is escape (insect lift and escape (fly away)).
    12. Based on Equation (6), calculate the lift formula of Bombardier Beetle.
    13. Based on the following Equation, calculate the new position of Bombardier Beetle.
 
$$\frac{L \times (ub_d - lb_d)}{t}$$
    14. Based on Equation (3), update the population's new position and compute the solution.
-

- 
15. Based on formula (4), compute fitness value.
  16. The solution of best candidate found so far is recorded at the end.
  17. Print the optimal value of the objective function.

**End BBO.**

---

## 2.2 Particle Swarm Optimization (PSO)

PSO is a swarm-based metaheuristic optimization method, which is inspired by how birds interact with one another when looking for food. It was initially introduced by Kennedy and Eberhart in 1995 [16]. As in Equation (7), each particle's position on the search space domain will be modified depending on three steps: the particle's starting velocity, its best velocity, and its global best velocity. The bird swarm is led by the global one. The velocity rule of this method has three settings, which are cognitive factor( $c_1$ ), inertia weight ( $w$ ), and social factor ( $c_2$ ). The parameters  $c_1$  and  $c_2$  are mainly configured to 2 [16].

$$fV_{i,m}^{(t+1)} = w * v_{i,m}^{(t)} + c_1 * rand_1() * (pbest_{i,m} - x_{i,m}^{(t)}) + c_2 * rand_2() * (gbest_m - x_{i,m}^{(t)}) \quad (7)$$

## 2.3 Grey Wolf Optimizer

In 2014, Mirjalili et al. presented GWO [15], which is a nature-inspired method that mimics the hunting and social structure habits of grey wolves. These wolves were separated into the following sections by Mirjalili et al [15]:

1. Alpha ( $\alpha$ ) is a leader, which mimics the best value discovered so far;
2. Beta ( $\beta$ ) wolves, which are at the second level of the wolves' social structure, which are the second-best solution discovered so far;
3. Delta ( $\delta$ ) wolves, which occupy the third rung of the wolves' social hierarchy, which are the third-best answer discovered so far;

The algorithm's exploration phase of searching for prey is one of the tactics used by these social hierarchy components. Examples of the algorithm's exploitation phase include finding and attacking prey. The GWO applies the final position of the components of these social hierarchy and divides them into three to obtain the total position determined thus far, as seen in the following equation [15].

$$total\ Position = \frac{Position_{\alpha} \times Position_{\beta} \times Position_{\delta}}{3} \quad (8)$$

## 3 Experimental and Results

This section uses "MATLAB R2023a," which runs on an i5 CPU Intel core with RAM of 8 GB and Windows 11, to code the BBO, PSO, and GWO. Table 2 shows the parameters of the algorithms. These

algorithms are first utilized to optimize “the Congress on Evolutionary Computation (CEC) 2017 suite” in order to test them [18] and determine the best optimizer between them. After that, these algorithms are applied to optimize the will known engineering design problems. We organize these tests as follows:

Table 2. Parameters of BBO, PSO, and GWO

| <b>BBO</b>                                                         |             |
|--------------------------------------------------------------------|-------------|
| (d) is the distance between the predator and the Bombardier Beetle | Rand (0, 1) |
| Radiuses of circle 1 and circle 2, which are r1 and r2             | Rand (0, 1) |
| Typical insect wing lift coefficient (CL)                          | Rand (0, 1) |
| Density of Air ( $\rho$ )                                          | Rand (0, 1) |
| Area of wing (A)                                                   | Rand (0, 1) |
| Velocity of wing (V)                                               | Rand (0, 1) |
| Vapor of hot water of toxic spray                                  | 100         |
| Value of oxygen                                                    | Rand (0, 1) |
| Population size                                                    | 30          |
| Number of Iterations                                               | 50          |
| <b>PSO</b>                                                         |             |
| $C_1$ and $C_2$                                                    | 2           |
| Population size                                                    | 30          |
| Number of Iterations                                               | 50          |
| <b>GWO</b>                                                         |             |
| Number of Iterations                                               | 50          |
| Populations size                                                   | 30          |

### 3.1 Optimizing the well-known CEC2017 test bed suites

The well-known “(CEC) 2017” [18] benchmark functions, which consist of 23 mathematical models with unimodal, multimodal, and benchmark functions of hybrid modal, are used to test the aforementioned algorithms. Tables from 3 through 5 present the outcomes of various algorithms. The mean, standard deviation, and best fitness in which the qualitative assessments are used in this comparison. The majority of functions, including function from 1 to 4, functions from 6 to 11, functions from 13 to 16, and functions from 21 to 23 are best solved by the BBO method, which has an advantage in solving the unimodal benchmark functions, including functions from F1 to F4, according to ranking rows in the following tables. Moreover, BBO has the merit in optimizing the noisy and ragged models, such as F5 benchmark function, along with multimodal functions from F6 to F10 benchmark functions. GWO is ranked second, and it effectively solves the hybrid modal functions F12, F17, F18, and F20. The PSO method typically is ranked last.

Table 3. Results of algorithms in optimizing benchmark functions from 1 to 7 of CEC2017

|    |       | <b>PSO</b> | <b>GWO</b> | <b>BBO</b> |
|----|-------|------------|------------|------------|
| F1 | Best  | 249.3578   | 1.385141   | 1.92E-29   |
|    | Mean  | 1.05E+04   | 7.71E+03   | 1.49E+03   |
|    | Worst | 7.86E+04   | 7.89E+04   | 5.63E+04   |

|           |       |             |             |             |
|-----------|-------|-------------|-------------|-------------|
|           | Std   | 19083.73643 | 15227.00221 | 8144.389006 |
|           | Rank  | 3           | 2           | 1           |
| F2        | Best  | 90.42198    | 0.118071    | 1.98E-16    |
|           | Mean  | 1.38E+11    | 1.76E+11    | 3.74E+10    |
|           | Worst | 6.90E+12    | 8.82E+12    | 1.87E+12    |
|           | Std   | 9.75984E+11 | 1.24753E+12 | 2.6441E+11  |
|           | Rank  | 3           | 2           | 1           |
|           | Best  | 4730.745    | 7020.243    | 7.00E-30    |
| F3        | Mean  | 2.01E+04    | 3.19E+04    | 4.10E+03    |
|           | Worst | 8.60E+04    | 1.76E+05    | 1.55E+05    |
|           | Std   | 21300.0867  | 32236.9215  | 22582.45079 |
|           | Rank  | 2           | 3           | 1           |
| F4        | Best  | 7.527749    | 3.418805    | 2.59E-14    |
|           | Mean  | 2.42E+01    | 2.81E+01    | 3.32E+00    |
|           | Worst | 8.92E+01    | 8.68E+01    | 9.00E+01    |
|           | Std   | 23.4883253  | 22.68484655 | 14.41958093 |
|           | Rank  | 3           | 2           | 1           |
| F5        | Best  | 175940.9    | 68.31131    | 0000.0      |
|           | Mean  | 9.40E+06    | 1.33E+07    | 6.96E+06    |
|           | Worst | 2.41E+08    | 2.61E+08    | 3.22E+08    |
|           | Std   | 38696124.52 | 46018424.14 | 45652603.1  |
|           | Rank  | 3           | 2           | 1           |
| F6        | Best  | 235.5879    | 2.934269    | 0000        |
|           | Mean  | 1.09E+04    | 6.38E+03    | 1.95E+03    |
|           | Worst | 7.45E+04    | 6.43E+04    | 7.07E+04    |
|           | Std   | 19934.96623 | 13600.7537  | 10369.36252 |
|           | Rank  | 3           | 2           | 1           |
| F7        | Best  | 125.3795    | 0.03885     | 0.000187    |
|           | Mean  | 1.25E+02    | 7.79E+00    | 2.22E+00    |
|           | Worst | 1.26E+02    | 1.16E+02    | 110.5244627 |
|           | Std   | 0.145336252 | 23.62630518 | 15.62895645 |
|           | Rank  | 3           | 2           | 1           |
| Sum Rank  |       | 20          | 15          | 7           |
| Mean Rank |       | 2.8572      | 2.14285     | 1           |

Table 4. Results of algorithms in optimizing the benchmark functions from 8 to 13 of CEC2017

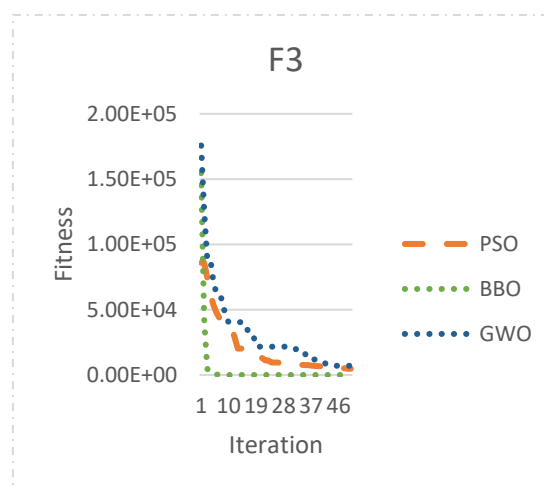
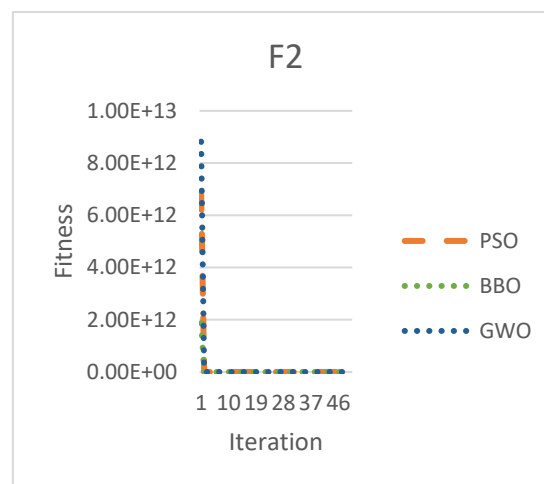
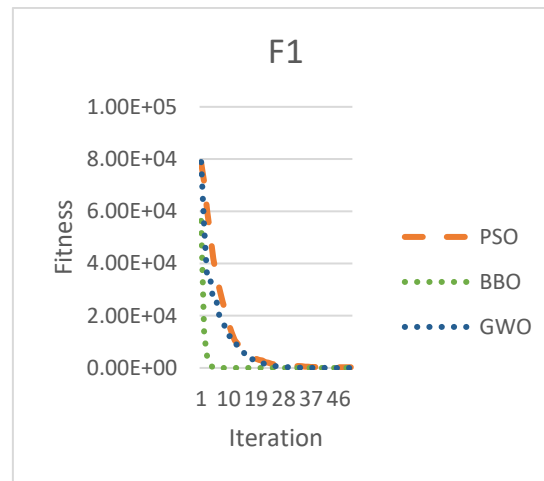
|    |       | <b>PSO</b> | <b>GWO</b>  | <b>BBO</b>  |
|----|-------|------------|-------------|-------------|
| F8 | Best  | -2751.25   | -2824.58    | -2943.26    |
|    | Mean  | -2.58E+03  | -2.64E+03   | -2.87E+03   |
|    | Worst | -1.85E+03  | -2.09E+03   | -1.66E+03   |
|    | Std   | 266.740582 | 149.8753732 | 307.4372613 |

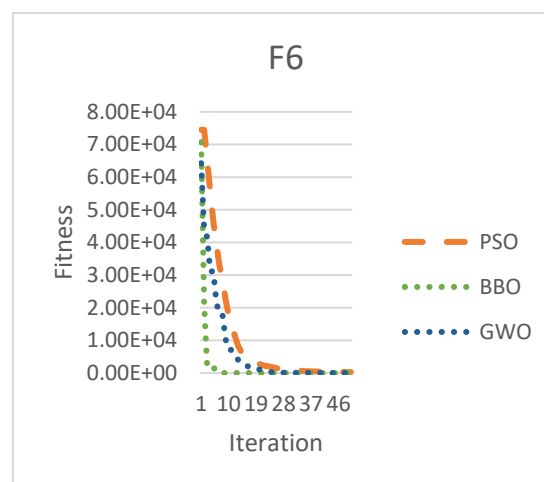
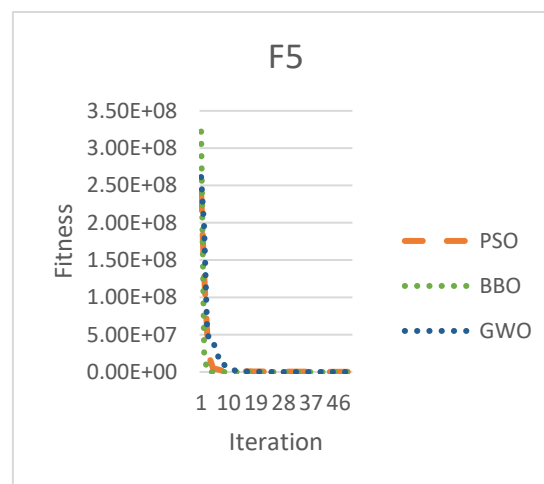
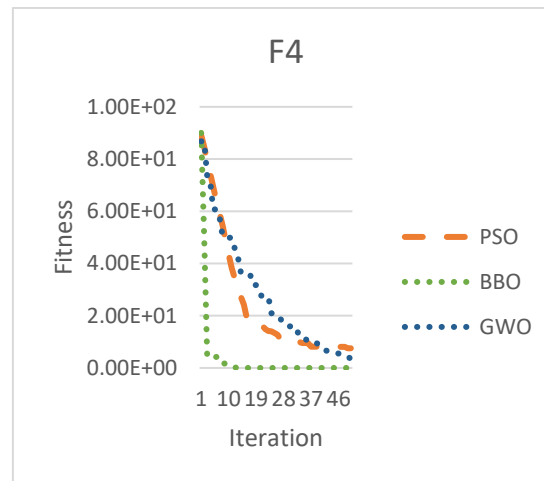
|           | Rank  | 3           | 2           | 1           |
|-----------|-------|-------------|-------------|-------------|
| F9        | Best  | 371.7325    | 101.5001    | 0.0000      |
|           | Mean  | 3.74E+02    | 2.42E+02    | 2.08E+01    |
|           | Worst | 4.75E+02    | 4.57E+02    | 3.97E+02    |
|           | Std   | 14.55593506 | 91.66533742 | 80.11291212 |
|           | Rank  | 3           | 2           | 1           |
| F10       | Best  | 8.868476    | 0.860056    | 4.44E-16    |
|           | Mean  | 1.17E+01    | 8.35E+00    | 1.28E+00    |
|           | Worst | 2.07E+01    | 2.04E+01    | 2.09E+01    |
|           | Std   | 3.034778817 | 6.399238153 | 4.48700696  |
|           | Rank  | 3           | 2           | 1           |
| F11       | Best  | 195.6512    | 0.553461    | 0.0000      |
|           | Mean  | 4.05E+02    | 6.86E+01    | 1.98E+01    |
|           | Worst | 6.46E+02    | 6.76E+02    | 4.93E+02    |
|           | Std   | 130.2777777 | 155.2211788 | 76.93681038 |
|           | Rank  | 3           | 2           | 1           |
| F12       | Best  | 9.79442     | 0.58412     | 0.745981    |
|           | Mean  | 2.56E+07    | 3.99E+07    | 1.27E+07    |
|           | Worst | 6.51E+08    | 4.94E+08    | 6.25E+08    |
|           | Std   | 106260816.6 | 116478149.6 | 88368702.09 |
|           | Rank  | 3           | 1           | 2           |
| F13       | Best  | 34.60654    | 2.040789    | 0.00486     |
|           | Mean  | 3.95E+07    | 9.48E+07    | 2.48E+07    |
|           | Worst | 7.83E+08    | 1.46E+09    | 1.24E+09    |
|           | Std   | 152062757.3 | 312965914.8 | 175654096.7 |
|           | Rank  | 3           | 2           | 1           |
| Sum Rank  |       | 18          | 11          | 7           |
| Mean Rank |       | 3           | 1.833       | 1.1666      |

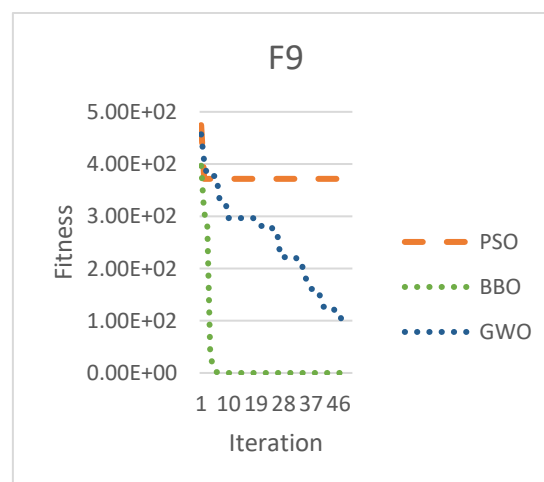
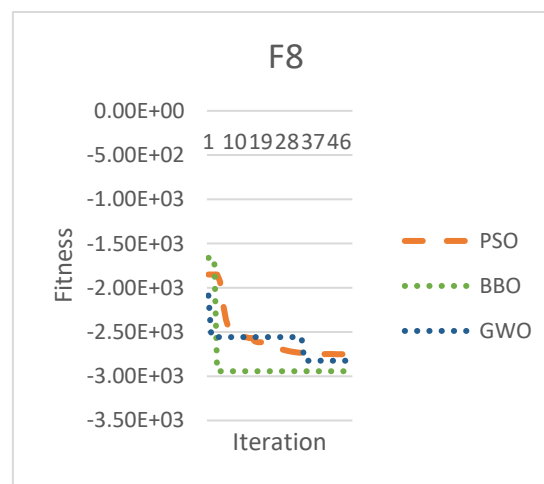
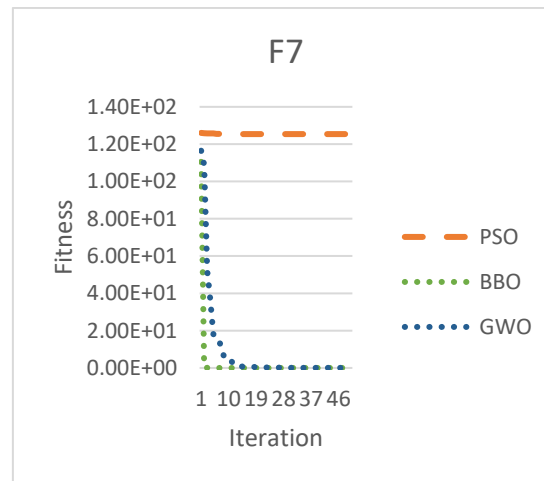
Table 5. Results of algorithms in optimizing the benchmark functions from 14 to 23 of CEC2017

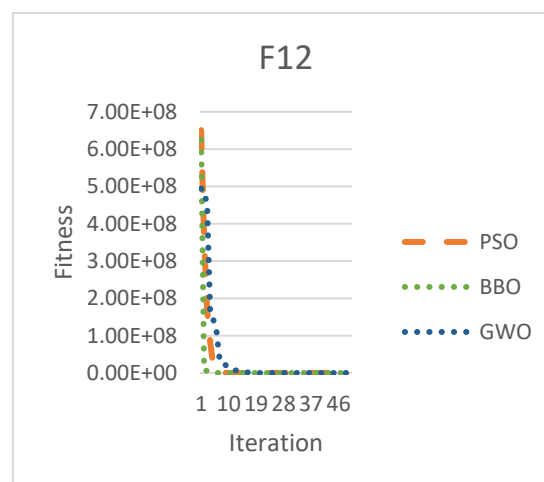
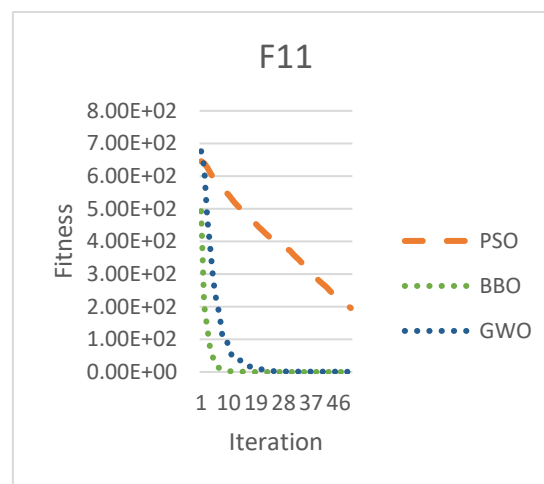
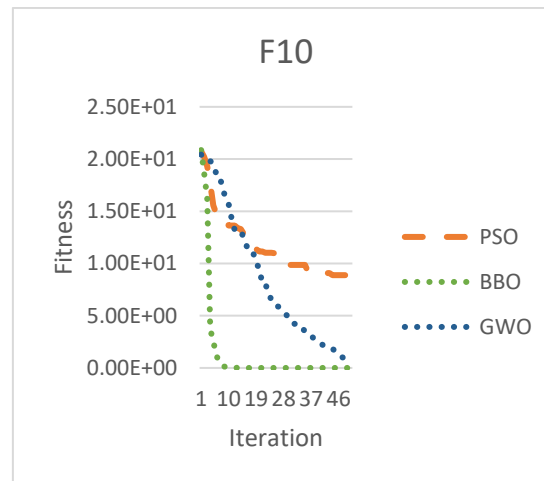
|     |       | PSO         | GWO         | BBO         |
|-----|-------|-------------|-------------|-------------|
| F14 | Best  | 1.992263    | 0.998032    | 0.998004    |
|     | Mean  | 1.18E+01    | 1.17E+00    | 1.58E+00    |
|     | Worst | 2.33E+02    | 1.27E+00    | 1.31E+01    |
|     | Std   | 33.28399592 | 0.119187461 | 2.375032241 |
|     | Rank  | 3           | 2           | 1           |
| F15 | Best  | 0.002662    | 0.000819    | 0.00033     |
|     | Mean  | 1.02E-02    | 3.90E-03    | 5.66E-03    |
|     | Worst | 1.15E-01    | 4.49E-02    | 1.71E-01    |
|     | Std   | 0.018947424 | 0.01054359  | 0.026057218 |
|     | Rank  | 3           | 2           | 1           |

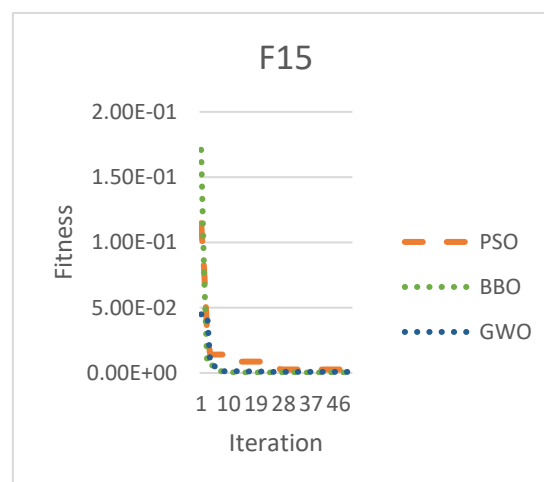
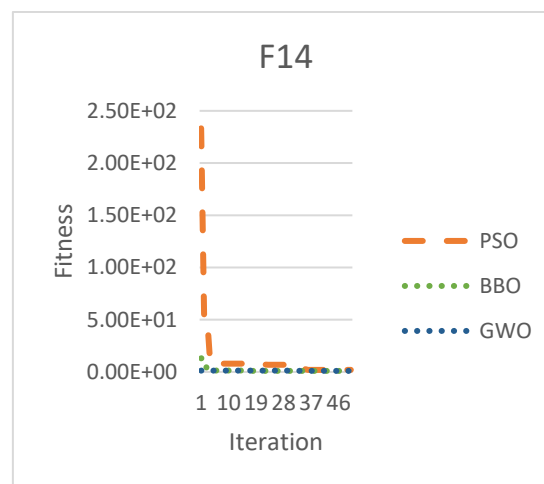
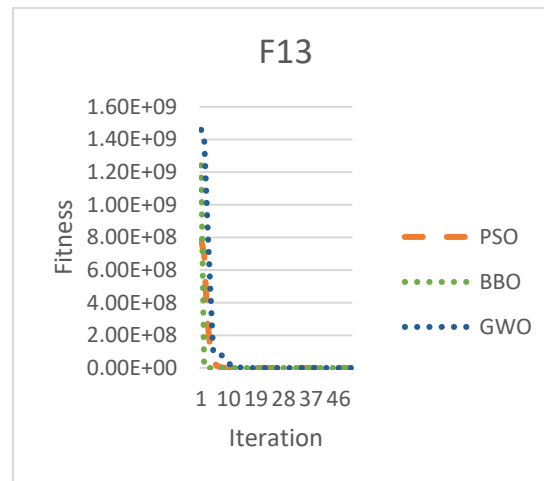
|           |       |             |             |             |
|-----------|-------|-------------|-------------|-------------|
| F16       | Best  | -1.02764    | -1.0315     | -1.0315     |
|           | Mean  | -1.01E+00   | -1.02E+00   | -9.62E-01   |
|           | Worst | -3.85E-01   | -8.43E-01   | 1.56E+00    |
|           | Std   | 0.092821772 | 0.045046996 | 0.369337452 |
|           | Rank  | 2           | 1           | 1           |
| F17       | Best  | 0.403181    | 0.401727    | 0.415829    |
|           | Mean  | 4.39E-01    | 4.12E-01    | 4.69E-01    |
|           | Worst | 9.55E-01    | 4.41E-01    | 1.35E+00    |
|           | Std   | 0.131854341 | 0.01598898  | 0.153189354 |
|           | Rank  | 2           | 1           | 3           |
| F18       | Best  | 3.505351    | 3.000322    | 3.563197    |
|           | Mean  | 1.95E+01    | 4.34E+00    | 6.58E+00    |
|           | Worst | 1.25E+02    | 4.04E+01    | 9.53E+01    |
|           | Std   | 37.68198916 | 6.104226742 | 13.09075811 |
|           | Rank  | 2           | 1           | 3           |
| F19       | Best  | -3.86005    | -3.84109    | -3.83138    |
|           | Mean  | -3.84E+00   | -3.75E+00   | -3.80E+00   |
|           | Worst | -3.46E+00   | -1.85E+00   | -3.03E+00   |
|           | Std   | 0.061006059 | 0.33151615  | 0.138700549 |
|           | Rank  | 1           | 2           | 3           |
| F20       | Best  | -2.51129    | -3.05116    | -2.99167    |
|           | Mean  | -2.44E+00   | -2.89E+00   | -2.79E+00   |
|           | Worst | -1.43E+00   | -1.46E+00   | -1.85E+00   |
|           | Std   | 0.191201484 | 0.294312457 | 0.261585465 |
|           | Rank  | 3           | 1           | 2           |
| F21       | Best  | -3.23193    | -4.50544    | -10.1517    |
|           | Mean  | -2.73E+00   | -3.86E+00   | -8.57E+00   |
|           | Worst | -7.50E-01   | -4.36E-01   | -3.68E-01   |
|           | Std   | 0.640097259 | 1.063311166 | 3.283441892 |
|           | Rank  | 3           | 2           | 1           |
| F22       | Best  | -2.51585    | -2.87848    | -10.3873    |
|           | Mean  | -2.13E+00   | -2.42E+00   | -7.88E+00   |
|           | Worst | -5.40E-01   | -4.15E-01   | -5.42E-01   |
|           | Std   | 0.425396207 | 0.587104768 | 2.785654648 |
|           | Rank  | 3           | 2           | 1           |
| F23       | Best  | -2.10679    | -4.42904    | -10.2494    |
|           | Mean  | -1.72E+00   | -2.80E+00   | -7.91E+00   |
|           | Worst | -4.97E-01   | -6.90E-01   | -5.47E-01   |
|           | Std   | 0.392643349 | 1.002714165 | 3.185094658 |
|           | Rank  | 3           | 2           | 1           |
| Sum Rank  |       | 25          | 16          | 17          |
| Mean Rank |       | 2.5         | 1.6         | 1.7         |

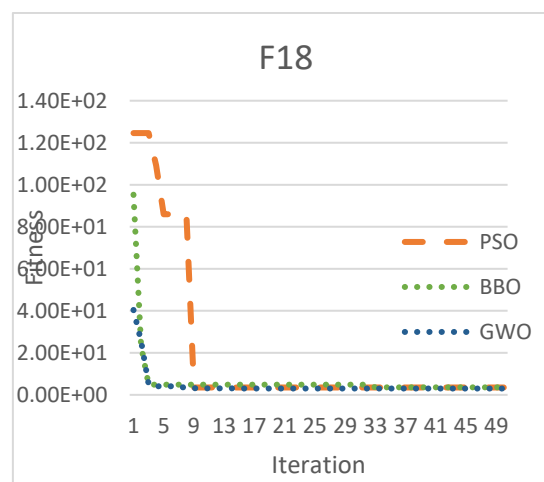
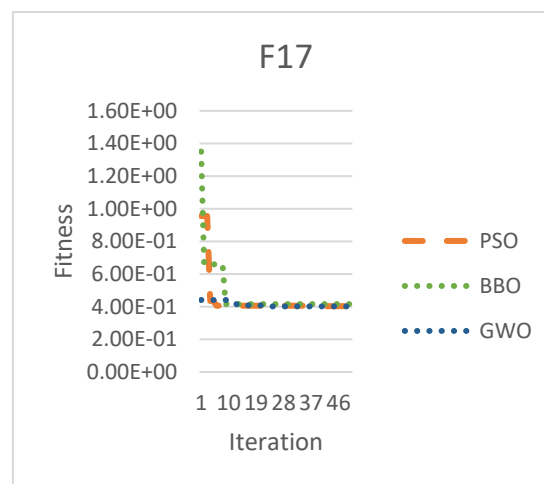
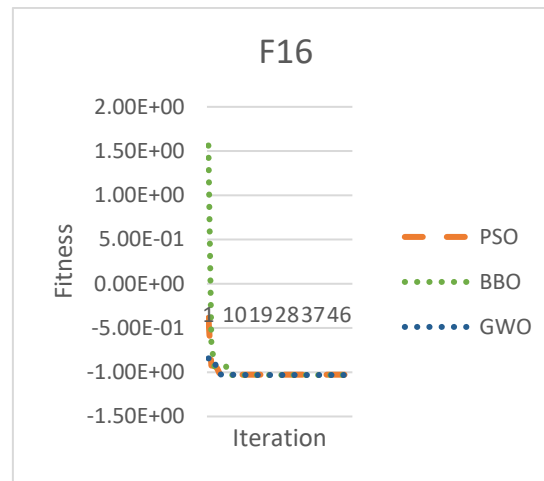


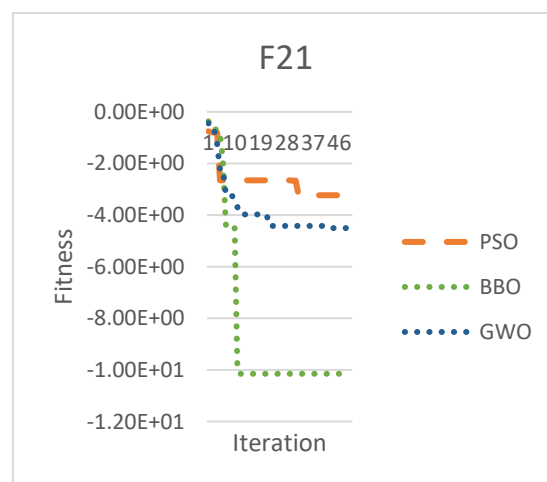
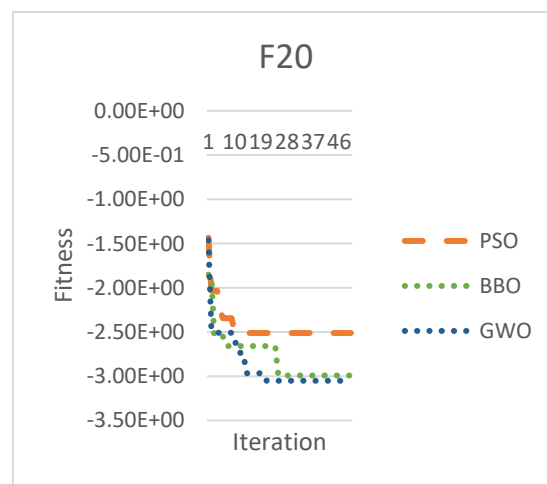
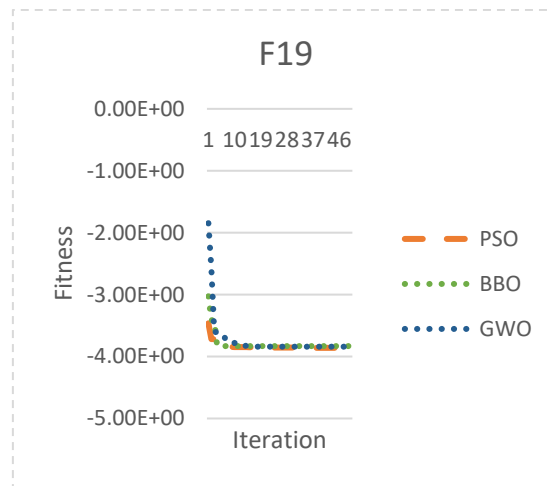












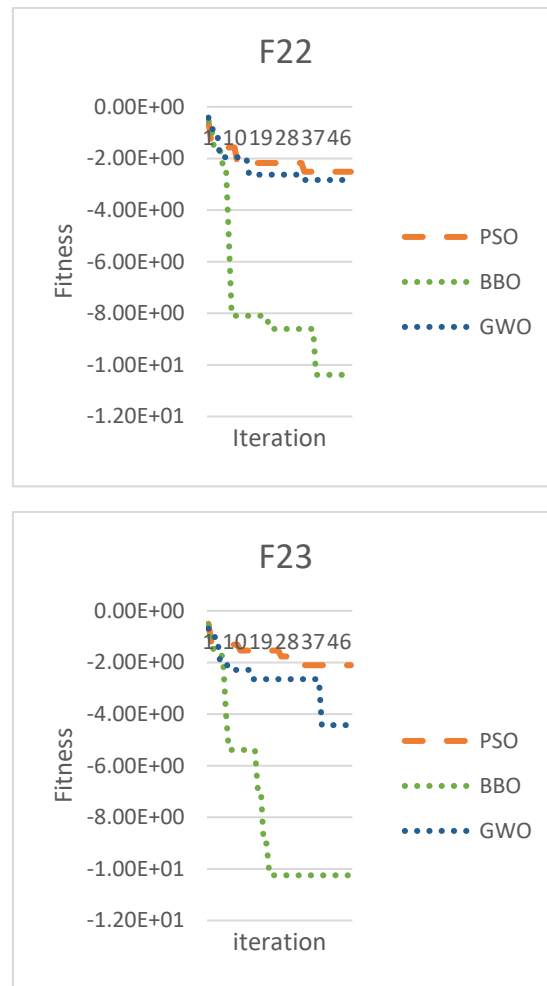


Figure 3. Results of convergence of optimizing CEC2017 Benchmark test suites

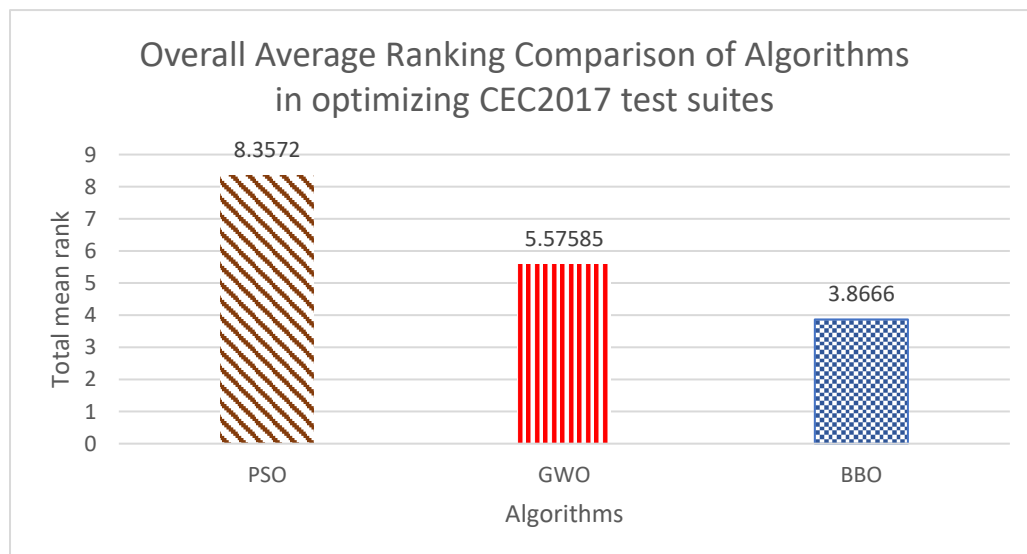


Figure 4. Overall average ranking comparison of algorithms in optimizing CEC2017 test suites

As seen in Figure 4, BBO obviously dominates the ranking scores and performs better overall than the other algorithm, which has the lowest total mean rank. Second place goes to GWO, which has a competitive performance. PSO is ranked last, which has the greatest value of total mean rank. The algorithm convergence rate for each algorithm utilized to solve each benchmark function of the CEC 2017 test bed suites, which is determined for the qualitative test. BBO converges faster than GWO, as seen in Figure 3. The PSO, which has the lowest rank of convergence, came after them.

### 3.2 Optimizing the well-known engineering design problems.

In this section, an empirical study of BBO, GWO, and PSO is applied to optimize a various of problems of engineering design, such as “Pressure Vessel Design”, “Speed Reducer Design”, “Robot Gripper Problem”, and “Cantilever Beam Design” [19]. These problems are listed in the following points:

#### 1. Speed Reducer Design Problem

The mathematical formula of the “Speed Reducer Design problem” is listed below. It consists of seven decision variables, which its geometric structure is depicted in Figure 5. Table 6 shows a comparison of performance of the BBO, PSO, and GWO methods for solving this optimization problem. As shown in the table, all seven variables have an influence on the final solution. The findings indicate that BBO achieved the minimum weight value of 2994.4711 and ranked first among all methods. GWO achieved the second-best result with a weight of 3004.1406, while PSO reveal the weakest performance, yielding a weight of 3242.5112.

Consider:  $y = [y_1, y_2, y_3, y_4, y_5, y_6, y_7] = [b, m, p, l_1, l_2, d_1, d_2]$

Minimize:  $f(y) = 0.7854y_1y_2^2(3.3333y_3^2 + 14.9334y_3 - 43.0934) - 1.508y_1(y_6^2 + y_7^2) + 7.4777(y_6^3 + y_7^2) + 0.7854(y_4y_6^2 + y_5y_7^2)$ .

Subject to:

$$g_1(\vec{y}) = \frac{27}{y_1y_2^2y_3} - 1 \leq 0, g_2(\vec{y}) = \frac{397.0}{y_1y_2^2y_3^2} - 1 \leq 0$$

$$g_3(\vec{y}) = \frac{1.90y_4^4}{y_2y_6^4y_3} - 1 \leq 0, g_4(\vec{y}) = \frac{1.90y_5}{y_2y_7^4y_3} - 1 \leq 0,$$

$$g_5(\vec{y}) = \frac{\sqrt{\left(\frac{745y_5}{y_2y_3}\right)^2 + 16.9 \times 10^6}}{110y_6^3} - 1 \leq 0$$

$$g_6(\vec{y}) = \frac{\sqrt{\left(\frac{745y_5}{y_2y_3}\right)^2 + 157.5 \times 10^6}}{85y_7^3} - 1 \leq 0$$

$$g_7(\vec{y}) = \frac{y_2y_3}{40} - 1$$

$$g_8(\vec{y}) = \frac{5y_2}{y_1} - 1 \leq 0$$

$$, g_9(\vec{y}) = \frac{y_1}{12y_2} - 1 \leq 0$$

$$g_{10}(\vec{y}) = \frac{1.5y_6 + 1.9}{y_4} - 1 \leq 0$$

$$g_{11}(\vec{y}) = \frac{1.1y_7 + 1.9}{y_5} - 1 \leq 0$$

Where  $2.6 \leq y_1 \leq 3.6$ ,  $0.7 \leq y_2 \leq 0.8$ ,  $17 \leq y_3 \leq 28$ ,  $7.3 \leq y_4 \leq 8.3$ ,  $2.9 \leq y_6 \leq 3.9$ ,  $5.0 \leq y_7 \leq 5.5$

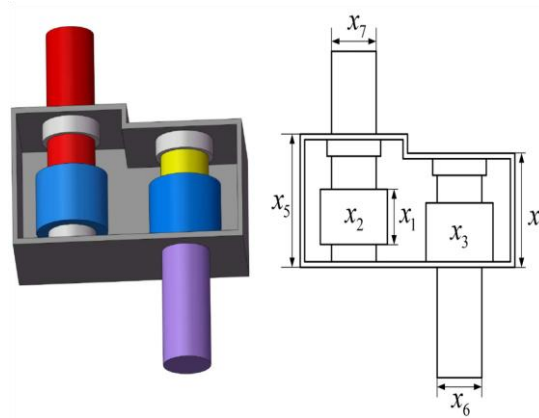


Figure 5. Speed Reducer Design Problem [19]

Table 6. Results of algorithms in optimizing the speed reducer design problem

| Algorithm  | $y_1$  | $y_2$  | $y_3$   | $y_4$  | $y_5$  | $y_6$  | $y_7$  | Minimum Weight   | Rank     |
|------------|--------|--------|---------|--------|--------|--------|--------|------------------|----------|
| <b>BBO</b> | 3.5000 | 0.7000 | 17.0000 | 7.3000 | 7.7153 | 3.3502 | 5.2867 | <b>2994.4711</b> | <b>1</b> |
| <b>GWO</b> | 3.5001 | 0.7000 | 17.0084 | 7.6345 | 7.8102 | 3.3594 | 5.2879 | 3004.1406        | 2        |
| <b>PSO</b> | 3.1594 | 0.8000 | 17.0000 | 7.9999 | 7.5015 | 3.2844 | 5.2239 | 3242.5112        | 3        |

## 2. Pressure Vessel Design Problem

Four choice variables are used to present the mathematical formulation of the Pressure Vessel Design issue from a different perspective. The geometric structure of this problem is seen in Figure 6. The outcomes of the suggested methods in resolving the Pressure Vessel Design optimization problem are shown in Table 7. As shown in Table 7, all four variables contribute to the final solution. The results indicate that BBO achieved the lowest cost value of 5901.31, ranking first among the compared methods. GWO achieved the second-best performance with a cost of 5942.8412, whereas PSO obtained the highest cost, exceeding 7800.

Consider:  $y = [y_1, y_2, y_3, y_4] = [T_s, T_h, R, L]$

$$\text{Minimize: } f(\vec{y}) = 0.6224y_1y_3y_4 + 1.7781y_2y_3^2 + 3.1661y_1^2y_4 + 19.84y_1^2y_3$$

Subject to:

$$\begin{aligned}
g_1(\vec{y}) &= -y_1 + 0.0193y_3 \leq 0, g_2(\vec{y}) = \\
&\quad -y_3 + 0.00954y_3 \\
&\quad \leq 0, \\
g_3(\vec{y}) &= -\pi y_3^2 y_4 - \frac{4}{3}\pi y_3^3 + 1296000 \leq \\
&\quad 0, g_4(\vec{y}) = y_4 \\
&\quad -240 \leq 0.
\end{aligned}$$

where  $0 \leq y_1, y_2 \leq 99, 10 \leq y_3, y_4 \leq 200$

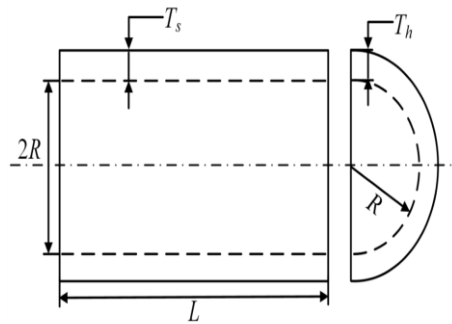


Figure 6. Pressure Vessel Design Problem [19]

Table 7. Results of algorithms in optimizing the pressure vessel design problem

| Algorithm  | $y_1$  | $y_2$  | $y_3$   | $y_4$    | Minimum Cost   | Rank     |
|------------|--------|--------|---------|----------|----------------|----------|
| <b>BBO</b> | 0.7793 | 0.3878 | 40.3671 | 199.5698 | <b>5901.31</b> | <b>1</b> |
| <b>GWO</b> | 0.7859 | 0.3849 | 40.3313 | 199.9405 | 5942.8412      | 2        |
| <b>PSO</b> | 0.1250 | 3.8125 | 25.1443 | 114.2507 | 7850.2362      | 3        |

### 3. Cantilever Beam Design Problem

Using five variables, the “Cantilever Beam Design Problem” is mathematically formulated by the following equation. The geometry of its structure is illustrated in Figure 7. The results are showed in Table 8, which lists the values of variables 1 through 5. Among the tested methods, the BBO algorithm ranks first, which obtained the lowest cost of 1.33996. GWO follows closely with a cost of 1.33998, while PSO yields a considerably higher cost for this optimization problem.

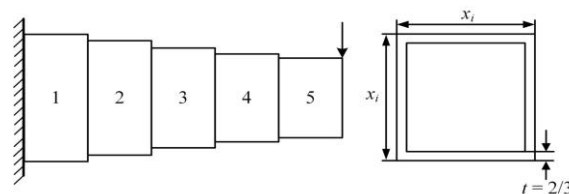


Figure 7. Cantilever Beam Design Problem [19]

Consider:  $y = [y_1, y_2, y_3, y_4, y_5]$

Minimize:

$$f(\bar{y}) = 0.6224(y_1 + y_2 + y_3 + y_4)$$

Subject to:

$$g(\bar{y}) = \frac{61}{y_1^3} + \frac{27}{y_0^3} + \frac{19}{y_3^2} + \frac{7}{y_1^3} + \frac{1}{y_5^3} - 1 \leq 0.$$

Where  $0.01 \leq y_1, y_2, y_3, y_4, y_5 \leq 100$ .

Table 8. Results of algorithms in optimizing the cantilever beam design problem

| Algorithm  | $y_1$   | $y_2$   | $y_3$   | $y_4$   | $y_5$   | Minimum weight | Rank     |
|------------|---------|---------|---------|---------|---------|----------------|----------|
| <b>BBO</b> | 6.01609 | 5.31142 | 4.48721 | 3.50227 | 2.15670 | <b>1.33996</b> | <b>1</b> |
| <b>GWO</b> | 6.01323 | 5.32102 | 4.49312 | 3.50675 | 2.13998 | 1.33998        | 2        |
| <b>PSO</b> | 6.1094  | 3.8265  | 16.6637 | 12.4117 | 4.4869  | 2.71429        | 3        |

#### 4. Robot Gripper Design Problem

The “Robot Gripper Design Problem” is optimized using the proposed algorithms, and their performance outcomes are summarized in Table 9. This problem is mathematically formulated using a seven-variable model. Figure 8 illustrates the geometric configuration of the system. Among the evaluated methods, BBO achieves the best performance with the lowest cost of 2.97112, which obtained the first rank. PSO attains the second position with a cost of 4.48748. Moreover, GWO records the weakest performance, which yielded a cost of 4.83446.

Consider:  $y = [y_1, y_2, y_3, y_4, y_5, y_6, y_7] = [a, b, c, e, f, l, \delta]$ .

$$\text{Minimize: } f(\bar{y}) = -\min_z F_k(\vec{y}, z) + \max_z F_k(\vec{y}, z)$$

Subject to:

$$g_1(\vec{y}) = -Y_{min} + h(\vec{y}, Z_{max}) \leq 0, g_2(\vec{y})$$

$$= -h(\vec{y}, Z_{max}) \leq 0,$$

$$g_1(\vec{y}) = -Y_{min} + h(\vec{y}, Z_{max}) \leq 0, g_2(\vec{y})$$

$$= -h(\vec{y}, Z_{max}) \leq 0,$$

$$g_5(\vec{y}) = l^2 + e^2 - (a + b)^2 \leq 0, g_6(\vec{y})$$

$$= b^2 - (a - e)^2 - (l - Z_{max})^2 \leq 0$$

$$g_7(\vec{y}) = Z_{max} - l \leq 0$$

where  $0 \leq e \leq 50, 100 \leq c \leq 200, 10 \leq f, a, b \leq 150, 1 \leq \delta \leq 3.14, 100 \leq l \leq 300$

Other parameter conditions are represented below:

$$\begin{aligned} F_k &= \frac{Pbsin(\alpha + \beta)}{2ccos(\alpha)} \\ \alpha &= \cos^{-1}\left(\frac{a^2 + g^2 - b^2}{2ag}\right) + \phi, \\ g &= \sqrt{e^2 + (z - l)^2} \\ \beta &= \cos^{-1}\left(\frac{b^2 + g^2 - a^2}{2ag}\right) - \phi, \\ \phi &= \tan^{-1}\left(\frac{e}{l - z}\right) \\ h(\vec{y}, z) &= 2(f + e + c\sin(\beta + \delta)) \\ Y_{min} &= 50, Y_{max} = 100, Y_G = 150 \\ , Z_{max} &= 100, P = 100. \end{aligned}$$

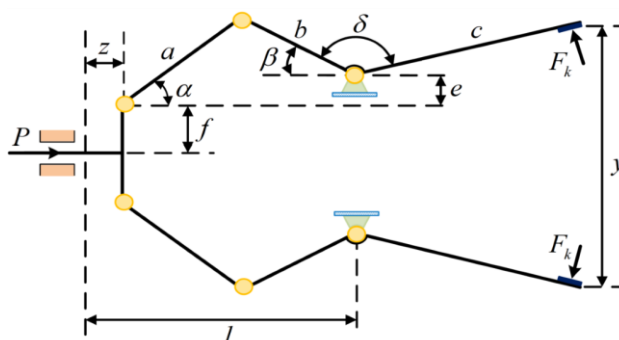


Figure 8. Robot Gripper Design Problem [19]

Table 9. Results of algorithms in optimizing the robot gripper design problem

| Algorithm  | $y_1$     | $y_2$     | $y_3$     | $y_4$    | $y_5$     | $y_6$     | $y_7$   | Min- cost      | Rank |
|------------|-----------|-----------|-----------|----------|-----------|-----------|---------|----------------|------|
| <b>BBO</b> | 150.00000 | 147.63481 | 199.41697 | 1.72781  | 147.84613 | 125.30214 | 2.43379 | <b>2.97112</b> | 1    |
| <b>GWO</b> | 148.93549 | 143.56337 | 134.49691 | 3.80565  | 80.74199  | 141.43387 | 2.25668 | 4.83446        | 3    |
| <b>PSO</b> | 155.81850 | 121.37676 | 162.69311 | 30.71219 | 70.41388  | 157.17438 | 2.38838 | 4.48748        | 2    |

## 4 Conclusion

In this paper, comprehensive mathematical foundations of BBO, GWO, and PSO are presented in this work. This comparative understanding offers researchers and practitioner's helpful direction when choosing appropriate optimization techniques for specific engineering challenges. The CEC 2017 benchmark test suites are thoroughly optimized in this research. These methods are also used to optimize well-known problems of engineering design, including "Robot Gripper Problems", "Pressure Vessel Design", "Cantilever Beam Design", and "Speed Reducer Design Problem". The outcomes of this study show that the BBO is the top optimizer for optimizing unimodal and multimodal problems, according to CEC 2017 findings. On the other hand, GWO algorithm is superior in handling the hybrid modal. The PSO algorithm has the worst outcome in solving these problems. That is why BBO is superior in solving the multimodal engineering design problem. In the future works, PSO will be hybridized with BBO to improve PSO's effectiveness and performance in resolving such these multimodal challenges.

## DECLARATIONS

**Funding.** This paper has no fund in which the publication fees is paid by Dr. Hisham A. Shehadeh.

**Conflicts of interest.** The authors declare no conflict of interest.

**Ethics approval.** Ethics approval was not required, as this study did not involve human participants, animals, or personally identifiable data.

**Informed consent.** 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 algorithm, namely, BBO is available in the following link: <https://www.mathworks.com/matlabcentral/fileexchange/184076-bombardier-beetle-optimizer-bbo>

**Author contributions.** The concept, design and evaluation of the proposed method is done by Hisham A. Shehadeh, Dr. Shehadeh structures the paper and devices experiments with selected benchmarking; Widi contributes in the statistical evaluation.

**Acknowledgements.** We acknowledge that this paper has no fund.

## REFERENCES

- [1] Houssein, E. H., Saeed, M. K., Hu, G., & Al-Sayed, M. M. (2024). Metaheuristics for Solving Global and Engineering Optimization Problems: Review, Applications, Open Issues and Challenges: EH Houssein et al. *Archives of computational methods in engineering*, 31(8), 4485-4519.
- [2] Camacho-Vallejo, J. F., Corpus, C., & Villegas, J. G. (2024). Metaheuristics for bilevel optimization: A comprehensive review. *Computers & Operations Research*, 161, 106410.
- [3] Lin, M., Dai, Z., Qiu, Z., Chen, H., & Lin, H. (2025). Multi-strategy enhanced sand cat swarm optimization algorithm and its engineering applications. *Scientific Reports*, 15(1), 34391.
- [4] Shehadeh, H. A. (2025). Bermuda Triangle Optimizer (BTO): A Novel Metaheuristic Method for Global Optimization. *International Journal of Advances in Soft Computing & Its Applications*, 17(2).
- [5] Zhao, W., Wang, L., & Zhang, Z. (2019). A novel atom search optimization for dispersion coefficient estimation in groundwater. *Future Generation Computer Systems*, 91, 601-610.
- [6] Shehadeh, H. A. (2023). Chernobyl disaster optimizer (CDO): a novel meta-heuristic method for global optimization. *Neural Computing and Applications*, 35(15), 10733-10749.
- [7] Seyyedabbasi, A., & Kiani, F. (2023). Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. *Engineering with computers*, 39(4), 2627-2651.
- [8] 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).
- [9] El-Kenawy, E. S. M., Khodadadi, N., Mirjalili, S., Abdelhamid, A. A., Eid, M. M., & Ibrahim, A. (2024). Greylag goose optimization: nature-inspired optimization algorithm. *Expert Systems with Applications*, 238, 122147.
- [10] Al-Betar, M. A., Awadallah, M. A., Braik, M. S., Makhadmeh, S., & Doush, I. A. (2024). Elk herd optimizer: a novel nature-inspired metaheuristic algorithm. *Artificial Intelligence Review*, 57(3), 48.
- [11] Dhivyaprabha, T. T., Subashini, P., & Krishnaveni, M. (2018). Synergistic fibroblast optimization: a novel nature-inspired computing algorithm. *Frontiers of Information Technology & Electronic Engineering*, 19(7), 815-833.
- [12] Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341-359.
- [13] Tan, Y., & Zhu, Y. (2010, June). Fireworks algorithm for optimization. In *International conference in swarm intelligence* (pp. 355-364). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [14] Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *simulation*, 76(2), 60-68.
- [15] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, 69, 46-61.
- [16] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). ieee.
- [17] Shehadeh, H. A., Idris, M. Y. I., Jebri, I. H., & Mustafa, H. M. (2026). Bombardier Beetle Optimizer: A Novel Bio-Inspired Algorithm for Global Optimization. In *2026 2<sup>nd</sup> International Conference on Computational Intelligence Approaches and Applications (ICCIAA)*. IEEE.
- [18] Wu, G., Mallipeddi, R., & Suganthan, P. N. (2017). Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization. *National University of Defense Technology*,

Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report, 9, 2017.

[19] Aribowo, W., & Shehadeh, H. A. (2025). A Comparative Study of Metaheuristic Optimization Algorithms in Solving Engineering Designing Problems. *Journal of Robotics and Control (JRC)*, 6(4), 1885-1898.