# New binary whale optimization algorithm for discrete optimization problems

**5 authors**, including:

Abdelazim G. Hussien
Fayoum University
**9** PUBLICATIONS **136** CITATIONS

Aboul Ella Hassanien
Cairo University
**1,011** PUBLICATIONS **10,220** CITATIONS

Essam H Houssein
Minia University
**75** PUBLICATIONS **621** CITATIONS

Ahmad Taher Azar
Prince Sultan University
**426** PUBLICATIONS **6,542** CITATIONS

Some of the authors of this publication are also working on these related projects:

Intelligent approaches for medical diagnosis View project

Document Analysis View project

# New binary whale optimization algorithm for discrete optimization problems

Abdelazim G. Hussien, Aboul Ella Hassanien, Essam H. Houssein, Mohamed Amin & Ahmad Taher Azar

Published online: 26 Jun 2019.

Submit your article to this journal 

View Crossmark data

Taylor & Francis
Taylor & Francis Group

Check for updates

# New binary whale optimization algorithm for discrete optimization problems

Abdelazim G. Hussien [a], Aboul Ella Hassanien [b,c], Essam H. Houssein [d], Mohamed Amin[e] and Ahmad Taher Azar [f,g]

[a]Faculty of Science, Fayoum University, Faiyum, Egypt; [b]Faculty of Computers and Artificial Intelligence, Cairo University, Cairo, Egypt; [c]Scientific Research Group in Egypt (SRGE) Cairo, Egypt; [d]Faculty of Computers and Information, Minia University, Minya, Egypt; [e]Faculty of Science, Menoufia University, Menoufia, Egypt; [f]College of Engineering, Prince Sultan University, Riyadh, Saudi Arabia; [g]Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt

**ABSTRACT**

The whale optimization algorithm (WOA) is an intelligence-based technique that simulates the hunting behaviour of humpback whales in nature. In this article, an adaptation of the original version of the WOA is made for handling binary optimization problems. For this purpose, two transfer functions (S-shaped and V-shaped) are presented to map a continuous search space to a binary one. To illustrate the functionality and performance of the proposed binary whale optimization algorithm (bWOA), its results when applied on twenty-two benchmark functions, three engineering optimization problems and a real-world travelling salesman problem are found. Furthermore, the proposed bWOA is compared with five well-known metaheuristic algorithms. The experimental results show its superiority in comparison with other state-of-the-art metaheuristics in terms of accuracy and speed. Finally, Wilcoxon's rank-sum non-parametric statistical test is carried out at the 5% significance level to judge whether the results of the proposed algorithm differ from those of the other comparison algorithms in a statistically significant way.

## 1. Introduction

Several optimization problems dealing with discrete binary search space have been applied in a wide variety of domains such as feature selection (Emary and Zawbaa 2018), dimensionality reduction (Zawbaa *et al.* 2018), chemical activity (Hussien, Hassanien, and Houssein 2017), wind turbine placement (Beşkirli *et al.* 2018), mass transit services, unit commitment (Yuan *et al.* 2014) and resource allocation (Fan, You, and Li 2013). Recently, various metaheuristic algorithms have been proposed to solve complicated computational problems (Hassanien and Emary 2016; Tharwat *et al.* 2017) such as particle swarm optimization (PSO) (Kennedy and Eberhart 1995), genetic algorithms (GAs) (Holland 1975), bat algorithms (BAs) (Yang 2010b), grasshopper optimization algorithms (GOAs) (Saremi, Mirjalili, and Lewis 2017) and the ant lion optimizer (ALO) (Mirjalili 2015). Moreover, the flexibility of these algorithms to deal with different problems compared with conventional optimization techniques makes these algorithms popular among researchers.

Classical mathematical techniques and methods always fail in solving complex optimization problems in a reasonable time. However, metaheuristic algorithms are able to solve $\mathcal{NP}$-problems

---

owing to their simplicity, ease of implementation and ability to avoid local optima (Saka, Hasançebi, and Geem 2016). Strictly speaking, metaheuristic optimization algorithms are classified into three groups: physics-based (Geem, Kim, and Loganathan 2001), evolution-based (Gong *et al.* 2014) and swarm-intelligence-based (Krause *et al.* 2013).

Most metaheuristic algorithms were designed to handle continuous problems (Mirjalili 2015; Saremi, Mirjalili, and Lewis 2017). For this reason, the need for binary algorithms became an urgent task for their solution. Many research efforts have been made in order to adapt algorithms for solving binary optimization problems. Therefore, various optimization problems with a binary domain such as feature selection (Pal and Maiti 2010) and multi-class support vector machines (Babaoglu, Findik, and Ülker 2010; Lee and Lee 2015) still need more study and are regarded as hot topics. Most metaheuristic algorithms, such as the dragonfly algorithm (Mirjalili 2016), the gravitational search algorithm (GSA) (Rashedi, Nezamabadi-Pour, and Saryazdi 2009), and the magnetic optimization algorithm (MOA) (Tayarani-N and Akbarzadeh-T 2008), have binary versions that enable them to solve binary optimization problems.

In recent years, several binary metaheuristic algorithms have been developed to tackle continuous problems while conserving the concepts of the search process. For instance, Emary, Zawbaa, and Hassanien (2016a, 2016b) have proposed binary ant lion and binary grey wolf optimization for feature selection. To solve optimization problems, various metaheuristic algorithms have been proposed by for instance Obagbuwa and Abidoye (2016), who presented binary cockroach swarm optimization for combinatorial optimization problems. Dahi, Mezioud, and Draa (2016) proposed a binary flower pollination algorithm and Beheshti, Shamsuddin, and Yuhaniz (2013) presented a binary accelerated particle swarm algorithm for solving discrete optimization problems. Also, Mirjalili, Mirjalili, and Yang (2014) proposed a binary bat algorithm (BBA). Further, Holland (1992) proposed a GA that was inspired by natural evolution and has been widely used for solving combinatorial optimization problems (COPs) (Lau *et al.* 2010).

The whale optimization algorithm (WOA) (Mirjalili and Lewis 2016) has gained huge interest since its appearance in 2016. Jadhav and Gomathi (2018) proposed a hybrid of the grey wolf algorithm and the whale algorithm called the WGC algorithm. Also, Mafarja and Mirjalili (2017) proposed a hybrid whale optimization algorithm with simulated annealing and applied it to feature selection. Wang *et al.* (2017) proposed a multi-objective version of the WOA and applied it to wind speed forecasting. El Aziz, Ewees, and Hassanien (2017) applied the WOA to multilevel thresholding image segmentation.

Eid (2018) developed a new binary whale optimization algorithm (bWOA) to estimate the parameters of photovoltaic cells solving the feature selection (FS) problem. In Reddy K. *et al.* (2018), the bWOA has been used for profit-based unit commitment problems in marketing.

In the same context, a binary whale optimization algorithm and an S-shaped binary whale optimization algorithm have been presented by Hussien, Houssein, and Hassanien (2017) and Hussien *et al.* (2019) in order to solve the feature selection problem.

Obviously, the no-free-lunch (NFL) theorem (Wolpert and Macready 1997) makes this field of study highly active, which results in enhancing current approaches and proposing new metaheuristics every year. This theorem has revealed that no one metaheuristic optimization algorithm in particular is best suited for solving all optimization problems. Also, the superior ability of the WOA (Mirjalili and Lewis 2016) to deal with different problems, such as engineering design problems and mathematical optimization problems, makes this algorithm popular compared with conventional as well as metaheuristic algorithms.

The WOA was proposed to solve the continuous search space problem—it cannot deal with binary problems directly. For this reason, a promising way to cope with this issue is regarded as the main motivation for this article, which proposes two binary variants of the WOA, called the bWOA-S and the bWOA-V, for solving discrete optimization problems. In order to achieve this, the two proposed algorithms will force the whales to move in binary search spaces by shifting their positions by zero or one. Therefore, two sigmoid transfer functions are applied to obtain a new position. Even-

tually, the statistical results prove that the proposed algorithms are very competitive compared with three well-known metaheuristic algorithms, namely the BBA (Mirjalili, Mirjalili, and Yang 2014), the bPSO (Kennedy and Eberhart 1997), the binary grey wolf optimization algorithms bGWO1 and bGWO2 (Emary, Zawbaa, and Hassanien 2016b) and a GA (Holland 1992).

The article is organized as follows: a short overview of the basic whale optimization algorithm is presented in Section 2. Section 3 discusses the proposed binary whale optimization algorithm. Benchmark test functions used in the performance evaluation process are presented in Section 4. Sections 5 and 6 provide experimental results and a discussion. Finally, Section 7 concludes the article and suggests some future work.

## 2. Whale optimization algorithm

Mirjalili and Lewis (2016) have developed the WOA inspired by the behaviour of whales. Mathematically, in the assumption model of the WOA, the current best candidate solution is the target prey. Other whales try to update their position to reach the best according to Equation (1):

$$D = |CX^*(t) - X(t)| \tag{1}$$

$$X(t + 1) = X^*(t) - A \cdot D, \tag{2}$$

where $t$ represents the current iteration, $C$ and $A$ are coefficient vectors, $X^*$ represents the position vector of the best solution, and $X$ is the position vector. $A$ and $C$ values are calculated by the following equations:

$$A = 2 \cdot a \cdot r - a \tag{3}$$

$$C = 2 \cdot r, \tag{4}$$

where $a$ is linearly decreased from two to zero over iterations and $r \in [0, 1]$. The exploitation phase is simulated mathematically as follows.

(1) **Shrinking encircling:** obtained by decreasing $a$ values according to Equation (4). Remark that $a$ is a random value in $[-a, a]$.
(2) **Spiral updating:** calculates the distance between the prey and the whale. Equation (5), calculates the spiral that mimics the spiral movement as follows:

$$X(t + 1) = D^l e^{bl} \cdot \cos(2\pi l) + X^*(t) \tag{5}$$

where $b$ is fixed and $l$ is a random number in $[-1, 1]$. To choose either the spiral model or the shrinking encircling mechanism model, a probability of 50% is assumed as follows:

$$X(t + 1) = \begin{cases} X^*(t) - A \cdot D & \text{if } p < 0.5 \\ D^l \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) & \text{otherwise,} \end{cases} \tag{6}$$

where $p$ is a random number in a uniform distribution. On the other hand, in the exploration phase, $1 \prec A \prec -1$ is used to force the agent to move away from this location. Equations (7) and (8) represent the exploration phase mathematically as follows:

$$D = |C \cdot X_{rand} - X| \tag{7}$$

$$X(t + 1) = X_{rand} - A \cdot D. \tag{8}$$

## 3. Binary whale optimization algorithm

In continuous WOA, each whale changes its position to any point using Equation (2), while in bWOA, to update a position means you switch between either zero or one. Moreover, in the binary version, Equation (2) becomes in this case insufficient to perform the position updating process. So the major point is changing the agent position according to the probability of its distance (Kennedy and Eberhart 1997). To achieve this, a transfer function is required to map the values of the distance to probability values to update the positions. Figure 1 demonstrates the flow chart of the binary WOA version.

S-shaped (sigmoid) and V-shaped (hyperbolic tan) functions are used to squash solutions by Equations (9) or (10), respectively. Then, a threshold is applied in the case of S-shaped functions using Equation (11), while using Equation (12) in the case of V-shaped functions.

Figure 2 illustrates the mathematical curve of S-shaped and V-shaped functions, and Algorithm 1 shows the pseudo-code of the binary versions.
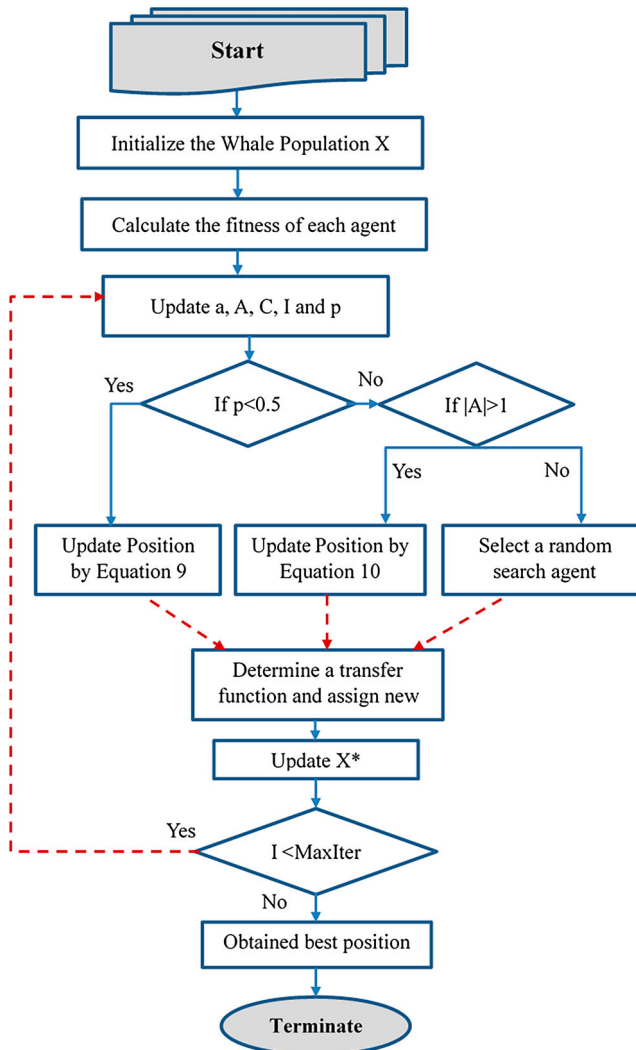


**Figure 1.** Binary whale optimization algorithm diagram.

$$S(x_i^k(t)) = \frac{1}{1 + e^{-d_i^k}(t)} \tag{9}$$

$$V(x_i^k(t)) = |\tanh(x_i^k(t))|, \tag{10}$$
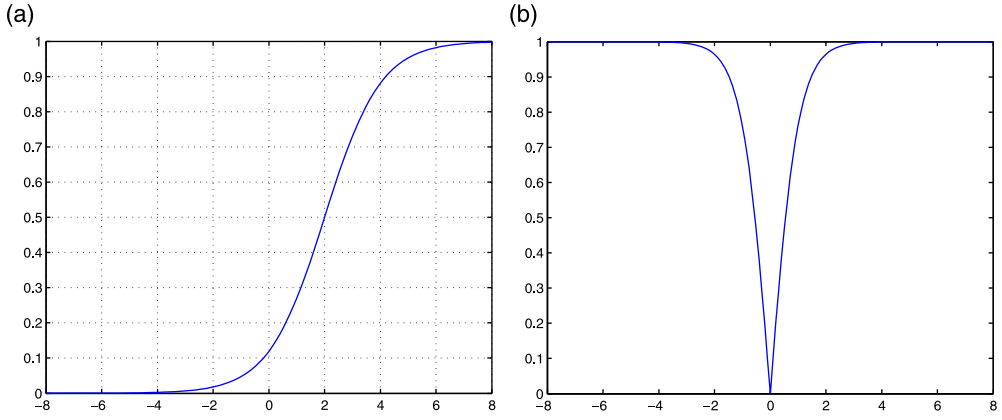


**Figure 2.** S-shaped and V-shaped transfer functions. (a) V-shaped and (b) S-shaped.

---

**Algorithm 1** Binary whale optimization algorithms (bWOA-S and bWOA-V)

---

1: **Input:** $n$ Number of whales in the population.
2: *MaxIter* Number of iterations for optimization.
3: **Output:** Optimal whale position
4: Initialize $a$, the population of $n$ whales.
5: Find $X^* = best$ search agent threads.
6: **while** stopping criteria not meet to **do**
7:     **for** $whale_i$ belong to whales **do**
8:         Calculate and Update $a; A, C, p$ and $l$.
9:         **if** $p < 0.5$ **then**
10:             **if** $(|A| < 1)$ **then**
11:                 Update position by Equation (2)
12:             **else** $(|A| \geq 1)$
13:                 Select a random search agent $(X_{rand})$
14:                 Update position by Equation (8)
15:             **end if**
16:         **else** $(p \geq 0.5)$
17:             Update position by Equation (5)
18:         **end if**
19:         Squash solution using Equation (9) or Equation (10)
20:         Update $X(t + 1)$ from Equation (11) or Equation (12)
21:     **end for**
22:     Calculate the agent fitness
23:     Update $X^*$ if there is a better solution
24:     $t = t + 1$
25: **end while**

---

where $d_i^k(t)$ is the distance of the particle

$$x_i^k(t+1) = \begin{cases} 0 & \text{if } r \, and \, < S(x_i^k(t)) \\ 1 & \text{otherwise} \end{cases} \tag{11}$$

$$X_i^k(t+1) = \begin{cases} (x_i^k(t))^{-1} & r \, and \, < V(x_i^k(t)) \\ x_i^k(t) & \text{otherwise,} \end{cases} \tag{12}$$

where $x_i^k(t)$ and $x_i^k(t+1)$ illustrate the position of the $i$th particle at specific iterations and dimensions, and $x_i^k(t)^{-1}$ is the complement of $(x_i^k(t))$.

## 4. Evaluation criteria and benchmark test functions

Twenty-two benchmark unconstrained functions (unimodal, multimodal and composite functions) and three engineering problems are used to evaluate the performance of the proposed bWOA-S and bWOA-V algorithms.

### 4.1. Evaluation criteria

The proposed algorithm is evaluated using three different statistical measurements as follows.

(1) **Average (Ave)** is the average of a stochastic optimization algorithm applied $N$ times as shown in Equation (13):

$$\text{Ave} = \frac{1}{N} \sum_{i=1}^{N} f_*^i, \tag{13}$$

where $f_*^i$ is the optimal solution that resulted at the $i$th agent of the algorithm.
(2) **Median (Med)** is the middle value of the ordered data.
(3) **Standard deviation (Std)** if the standard deviation is too small, then it means that the optimizer converges to the same solution. Otherwise, if it has large values, then it means that it is close to random results, as shown in Equation (14):

$$\text{Std} = \sqrt{\left(\frac{1}{N-1}\right) \sum (f_*^i - \text{Mean})^2}. \tag{14}$$

### 4.2. Benchmark test functions

To investigate the performance of the bWOA, twenty-two standard benchmark functions with different characteristics are selected (Mirjalili, Mirjalili, and Yang 2014) in addition to three engineering design problems. Numerical experiments are carried out for twenty-five case studies including seven unimodal functions, nine multimodal functions, six composite functions and three engineering design problems.

Tables 1 and 2 demonstrate the aforementioned functions, where '*Dim*' indicates the number of dimensions, 'Range' is the boundary of the function's search space, and $f_{\min}$ is the minimum value obtained from the function.

Moreover, three optimization engineering design problems, namely tension/compression spring design, welded beam design and pressure vessel design, are presented in Table 3.

**Table 1.** Unimodal and multimodal functions.

| Function | Range | Dim | Minimum |
|---|---|---|---|
| $F1 = \sum_{i=1}^{n} x_i^2$ | $[-100, 100]$ | 30 | 0 |
| $F2 = \sum_{i=1}^{n} |x_i| + \Pi_{i=1}^{n} |x_i|$ | $[-10, 10]$ | 5 | 0 |
| $F3 = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$ | $[-100, 100]$ | 5 | 0 |
| $F4 = \max_i \{|x_i|, 1 \leq i \leq n\}$ | $[-100, 100]$ | 5 | 0 |
| $F5 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | $[-30, 30]$ | 5 | 0 |
| $F6 = \sum_{i=1}^{n} ([x_i + 0.5])^2$ | $[-100, 100]$ | 5 | 0 |
| $F7 = \sum_{i=1}^{n} i x_i^4 + random[0, 1]$ | $[-1.28, 1.28]$ | 5 | 0 |
| $F8 = \sum_{j=1}^{n} -Z_j \sin(\sqrt{|Z_j|})$ | $[-500, 500]$ | 30 | $-418.98 \times 5$ |
| $F9 = \sum_{i=1}^{n} [z_j^2 - 10\cos(2\pi z_j) + 10]$ | $[-5.12, 5.12]$ | 30 | 0 |
| $F10 = -20 \exp(-0.2\sqrt{\frac{1}{n} \sum_{j=1}^{n} z_j^2}) - \exp(\frac{1}{n} \sum_{j=1}^{n} \cos(2\pi Z_j)) + 20 + e$ | $[-32, 32]$ | 30 | 0 |
| $F11 = \frac{1}{4000} \sum_{j=1}^{n} z_j^2 - \prod_{j=1}^{n} \cos(z_j/\sqrt{j}) + 1$ | $[-600, 600]$ | 30 | 0 |
| $F12 = \frac{\pi}{n} \{10\sin(\pi y_1) + \sum_{j=1}^{n-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{j+1})] + (y_n - 1)^2\}$ $+ \sum_{j=1}^{n} u(x_j, 10, 100, 4),$ | $[-50, 50]$ | 30 | 0 |
| $y_j = 1 + \dfrac{x_j + 1}{4}, \quad u(x_j, a, k, m) = \begin{cases} k(x_j - a)^m & x_j \succ a \\ 0 & -a \prec x_j \prec a \\ k(-x_j - a)^m & -a \succ x_j \end{cases}$ | | | |
| $F13 = 0.1\{\sin^2(3\pi z_1) + \sum_{j=1}^{n} (z_j - 1)^2 [1 + \sin^2(3\pi z_j + 1)] + (z_n - 1)^2$ $[1 + \sin^2(2\pi z_n)]\} + \sum_{j=1}^{n} u(z_j, 5, 100, 4)$ | $[-50, 50]$ | 30 | 0 |
| $F14 = -\sum_{i=1}^{n} \sin(x_i) \times (\sin(i \cdot x_i^2/\pi))^{2m}, \quad m = 10$ | $[-100, 100]$ | 2 | $-1$ |
| $F15 = [\exp(-\sum_{i=1}^{n} (x_i/\beta)^{2m}) - 2\exp(-\sum_{i=1}^{n} x_i^2)] \times \prod_{i=1}^{n} \cos^2 x_i, \quad m = 5$ | $[-100, 100]$ | 2 | 0 |
| $F16 = \{[\sum_i^n \sin^2(x_i)] - \exp(-\sum_{i=1}^{n} x_i^2)\} \times \exp[-\sum_{i=1}^{n} \sin^2\sqrt{|x_i|}]$ | $[-5, 5]$ | 2 | $-1.0316$ |

## 5. Experimental result and discussion

The statistical results are illustrated in Table 4. Thirty independent runs are applied and averaged to depict the results. Three measures are used for investigating the performance of the suggested algorithm: (1) the average (Ave), (2) the standard deviation (Std) and (3) the median (Med).

### 5.1. Unimodal functions

According to the nature of unimodal functions, each function has one global solution only. Consequently, to examine metaheuristic algorithms in terms of convergence rate, unimodal functions are appropriate. The quantitative results for unimodal test functions are depicted in Table 4. This table reveals that the bWOA outperforms the other three metaheuristic algorithms in terms of the mean, median and standard deviation on all the unimodal test functions.

### 5.2. Multimodal benchmark functions

Each multimodal benchmark function has many local minima, so they are appropriate for measuring the ability of the algorithm to avoid local minimum points. The quantitative results in Table 4 show that the bWOA outperforms another algorithm on four of the multimodal benchmark test functions,

**Table 2.** Composite functions.

| No. | Function | | Range | *Dim* | Minimum |
|---|---|---|---|---|---|
| F17 | $f_1, f_2, f_3, \ldots, f_{10}$ | = | [−100, 100] | 10 | 0 |
| | Sphere function$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}]$ | = | | | |
| | $[1, 1, 1, \ldots, 1][\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ | = | | | |
| | $[5/100, 5/100, 5/100, \ldots, 5/100]$ | | | | |
| F18 | $f_1, f_2, f_3, \ldots, f_{10}$ | = | [−10, 10] | 10 | 0 |
| | Griewank's function$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}]$ | = | | | |
| | $[1, 1, 1, \ldots, 1][\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ | = | | | |
| | $[5/100, 5/100, 5/100, \ldots, 5/100]$ | | | | |
| F19 | $f_1, f_2, f_3, \ldots, f_{10}$ | = | [−100, 100] | 10 | 0 |
| | Griewank's function$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}]$ | = | | | |
| | $[1, 1, 1, \ldots, 1][\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ | = | | | |
| | $[1, 1, 1, \ldots, 1]$ | | | | |
| F20 | $f_1, f_2 =$ Ackley's function $f_3, f_4$ | = | [−100, 100] | 10 | 0 |
| | Rastrigin's function $f_5, f_6$ | = | | | |
| | Weierstrass's function $f_7, f_8$ | = | | | |
| | Griewank's function $f_9, f_{10}$ | = | | | |
| | Sphere function$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}]$ | = | | | |
| | $[1, 1, 1, \ldots, 1][\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ | = | | | |
| | $[5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$ | | | | |
| F21 | $f_1, f_2 =$ Rastrigin's function $f_3, f_4$ | = | [−30, 30] | 10 | 0 |
| | Weierstrass's function $f_5, f_6$ | = | | | |
| | Griewank's function $f_7, f_8$ | = | | | |
| | Ackley's function $f_9, f_{10}$ | = | | | |
| | Sphere function$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}]$ | = | | | |
| | $[1, 1, 1, \ldots, 1][\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ | = | | | |
| | $[1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$ | | | | |
| F22 | $f_1, f_2 =$ Rastrigin's function $f_3, f_4$ | = | | | |
| | Weierstrass's function $f_5, f_6$ | = | | | |
| | Griewank's function $f_7, f_8$ | = | | | |
| | Ackley's function $f_9, f_{10}$ | = | | | |
| | Sphere function$[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}]$ | = | | | |
| | $[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1][\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}]$ | = | | | |
| | $0.1 \times 1/5, 0.2 \times 1/5, 0.3 \times 5/0.5, 0.4 \times$ | | | | |
| | $5/0.5, 0.5 \times 5/100, 0.6 \times 5/100,$ | | | | |
| | $0.7 \times 5/32, 0.8 \times 5/32, 0.9 \times 5/100, 1 \times 5/100$ | | [−100, 100] | 10 | 0 |

*i.e.* F9, F11, F15 and F16, for all measures. The BBA and the binary particle swarm optimization (BPSO) algorithms have the best results on four (F10, F12, F13 and F14) and one (F8) test functions, respectively. According to Table 4, the proposed algorithm is capable of avoiding local minima.

### 5.3. Composite benchmark functions

The optimization of composite benchmark test functions is a very challenging task because both exploration and exploitation allow local optima to be avoided. The quantitative results are touted in Table 4 and Figure 3. The results show that the bWOA has the second-best values on F19, F20 and F22.

### 5.4. Engineering optimization problems

In addition to the previous results, the bWOA is evaluated with three engineering design problems; these problems have different constraints and characters. The quantitative results depicted in Table 5 make a fair and just comparison with the native WOA, various penalty function (Yang 2010a) constraint handling strategies being applied. The parameter settings are shown in Table 6.

**Table 3.** Engineering optimization problems.

| No. | Test problems | Functions |
|-----|---------------|-----------|
| E1 | Tension/compression spring (Belegundu and Arora [1985]) | Minimize: $f(x) = (x_3 + 2)x_2x_1^2$<br><br>where $g_1(x) = 1 - (x_2^3 x_3 / 71,785 x_1^4) \leq 0$,<br><br>$g_2(x) = [4x_2^2 - x_1 x_2 / 12,566(x_2 x_1^3 - x_1^4) + (1/5108 x_1^2)] - 10 \leq 0$,<br><br>$g_3(x) = 1 - (140.45 x_1 / x_2^2 x_3) \leq 0$,<br><br>$g_4(x) = (x_2 + x_1)/1.5 - 1 \leq 0$, $0.05 \leq x_1 \leq 2.00$, $0.25 \leq x_2 \leq 1.30$, $2.00 \leq x_3 \leq 15.00$ |
| E2 | Welded beam (Ray and Liew [2002]) | Minimize: $f_1(x) = 1.104,71x_1^2 x_2 + 0.048,11x_3 x_4(14.0 + x_2)$<br><br>where $g_1(x) = \tau - 13,600$, $g_2(x) = \sigma - 30,000$, $g_3(x) = \delta(x) - 0.25$,<br><br>$g_4(x) = x_1 - x_4$, $g_5(x) = 6,000 - p$, $g_6(x) = 0.125 - x_1$,<br><br>$g_7(x) = 1.104,71x_1^2 + 0.048,11x_3 x_4(14.0 + x_2)$,<br><br>$0.125 \leq x_1 \leq 5$, $0.1 \leq x_2 \leq 10$, $0.1 \leq x_3 \leq 10$, $0.125 \leq x_4 \leq 5$<br><br>$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\dfrac{x_2}{2R} + (\tau'')^2}$, $\tau' = \dfrac{p}{\sqrt{2x_1 x_2}}$, $\tau'' = \dfrac{MR}{J}$, $M = P(L + \dfrac{x_2}{2})$,<br><br>$R = \sqrt{\dfrac{x_2^2}{4} + (\dfrac{x_1 + x_3}{2})^2}$, $J = 2\sqrt{2x_1 x_2}[\dfrac{x_2^2}{4} + (\dfrac{x_1 + x_3}{2})^2]$, $\sigma(x) = \dfrac{6PL}{x_4 x_3^2}$, $\tau(x) = \dfrac{6PL^3}{Ex_3^2 x_4}$,<br><br>$P_c = \dfrac{4.013E\sqrt{\dfrac{x_3^2 x_4^6}{36}}}{L^2}(1 - \dfrac{x_3}{2L}\sqrt{\dfrac{E}{4G}})$, $L = 14$, $E = 30 \times 10^6$, $G = 12 \times 10^6$ |
| E3 | Pressure vessel (Kramer [1994]) | Minimize: $f(x) = 0.622,4x_1 x_3 x_4 + 1.778,1x_2 x_3^2 + 3.166,1x_1^2 x_4 + 19.84x_1^2 x_3$<br><br>where $g_1(x) = -x_1 + 0.019,3x$, $g_2(x) = -x_2 + 0/009,54x_3 \leq 0$,<br><br>$g_3(x) = -\pi x_3^2 x_4 - (4/3)\pi x_3^3 + 1,296,000 \leq 0$, $g_4(x) = x_4 - 240 \leq 0$<br><br>$0 \leq x_i \leq 100$, $i = 1, 2$, $10 \leq x_i \leq 200$, $i = 3, 4$ |

### Welded beam

Optimization results and the statistical results are given in Table 5. Therefore, this table reveals that the bWOA shows better performance compared with the native WOA.

### Tension/compression spring

Table 5 depicts the optimization and statistical results, showing that the bWOA algorithm is superior to the WOA on average.

### Pressure vessel

The statistical results and optimization results of utilizing the bWOA and the WOA are presented in Table 5. According to the results in this table, once more, both the bWOA algorithms are superior to the WOA.

## 5.5. Convergence test

To investigate the performance evaluation of the bWOA compared with the native WOA, overall benchmark test functions with five dimensions, tested to obtain convergence curve. Note that 30 independent runs are performed to obtain the convergence of the bWOA and the WOA. Also, all the

**Table 4.** Comparison of Ave, Std and Median for the benchmark functions.

| No. | Measure | WOA | bWOA-S | bWOA-V | bGWO1 | bGWO2 | BBA | BPSO | GA |
|---|---|---|---|---|---|---|---|---|---|
| F1 | Ave | 5.2987e−42 | **0** | **0** | **0** | **0** | 1.8518 | 5.2965 | 10.0705 |
| | Std | 1.180e−41 | 7.6828e−30 | **0** | **0** | **0** | 2.4981 | 2.7657 | 24.9445 |
| | Med | 2.979e−45 | **0** | **0** | **0** | **0** | 1.2037 | 4.6684 | 2.6534 |
| F2 | Ave | 4.4849e−36 | **0** | **0** | **0** | **0** | 0.0965 | 0.2292 | 0.269483 |
| | Std | 1.0005e−35 | 7.772e−35 | **0** | **0** | 1.085e−43 | 0.0646 | 0.0938 | 0.23788 |
| | Med | 1.503e−38 | **0** | **0** | **0** | **0** | 0.0880 | 0.2373 | 0.172363 |
| F3 | Ave | 5.0255e−29 | **0** | **0** | **0** | **0** | 7.8103 | 22.48915 | 555.9039 |
| | Std | 1.0798e−28 | **0** | **0** | 2.3841e−54 | | 9.7981e−24 | 14.11401 | 250.693 |
| | Med | 3.4899e−30 | **0** | **0** | **0** | **0** | 4.9511 | 19.09979 | 545.6876 |
| F4 | Ave | 6.7340e−18 | **0** | **0** | **0** | **0** | 1.1526 | 2.608854 | 1.59375 |
| | Std | 1.5042e−17 | 5.034e−36 | **0** | **0** | **0** | 0.6140 | 0.838937 | 1.21348 |
| | Med | 7.5e−23 | **0** | **0** | **0** | **0** | 1.5588 | 7.672516 | 4.671173 |
| F5 | Ave | 1.4953 | **0** | **0** | 0.2437 | **0** | 25.0743 | 148.0799 | 369.7545 |
| | Std | 0.001 | **0** | **0** | **0** | **0** | 28.4430 | 137.1896 | 342.8893 |
| | Med | 0.278 | **0** | **0** | **0** | **0** | 14.9324 | 96.0937 | 305.5475 |
| F6 | Ave | 5.3700e−08 | **0** | **0** | 0.0157 | **0** | 2.6993 | 8.496653 | 6.984222 |
| | Std | 4.5992e−08 | **0** | **0** | **0** | **0** | 2.7428 | 6.140883 | 7.010388 |
| | Med | 0.2764 | **0** | **0** | **0** | **0** | 1.0169 | 2.496094 | 1.71875 |
| F7 | Ave | 0.0024 | **0** | **0** | **0** | 1.2516e−04 | 0.0060 | 0.015542 | 0.047174 |
| | Std | 0.0015 | **0.0004** | 0.005 | 0.011 | 0.002 | 0.0044 | 0.007474 | 0.043587 |
| | Med | 0.0002 | **0** | **0** | **0** | **0** | 0.0057 | 0.014006 | 0.034778 |
| F8 | Ave | −1083.2 | −973.3 | **−1162.2** | −985 | −444.76 | −985.3203 | −988.355 | −929.324 |
| | Std | 171.002 | 127.98 | 98.02 | 1000.02 | 0.05 | 27.5790 | **14.21898** | 27.9523 |
| | Med | −1020.0653 | −971.23 | **−3220.28** | −434.23 | −789.32 | −994.8144 | −992.355 | −918.613 |
| F9 | Ave | 5.20 | **0** | **0** | **0** | **0** | 1.5850 | 4.977688 | 2.1896 |
| | Std | **0** | **0** | **0** | **0** | **0** | 1.3352 | 1.597929 | 1.883027 |
| | Med | 6.23 | **0** | **0** | **0** | **0** | 1.2686 | 5.282659 | 1.9901952 |
| F10 | Ave | **0** | **0** | **0** | **0** | **0** | 1.1560 | 2.7255 | 1.399853 |
| | Std | 1.9459e−15 | **0** | **0** | **0** | **0** | 0.7279 | 0.4721 | 1.338105 |
| | Med | 4.44e−15 | 8.88e−17 | **0** | **0** | **0** | 0.9589 | 2.7969 | 2.316849 |
| F11 | Ave | 3.0198e−15 | **0** | **0** | 8.8817e−16 | **0** | 0.2463 | 0.3873 | 0.7067 |
| | Std | 1.9459e−15 | **0** | **0** | 1.4222e−17 | **0** | 0.0839 | 0.1302 | 0.3223 |
| | Med | 0.0074 | **0** | **0** | **0** | 1.687e−23 | 0.2261 | 0.3862 | 0.7336 |
| F12 | Ave | 0.0054 | **0** | **0** | **0** | **0** | 0.2708 | 0.6213 | 0.191197 |
| | Std | 0.0053 | **0** | **0** | **0** | **0** | 0.3287 | 0.38857 | 0.244347 |
| | Med | 1.67e−7 | 0.057 | 0.0012 | 0.0030 | **0.0008** | 0.1506 | 0.4924 | 0.073291 |
| F13 | Ave | 0.0131 | **0.0336** | 0.212 | 0.621 | 0.112 | 0.1297 | 0.4444 | 0.193006 |
| | Std | **9.1228e−07** | 0.0076 | 0.007844 | 0.4351 | 0.096 | 0.0736 | 0.211701 | 0.254864 |
| | Med | 5.4e−8 | 0.0023 | **0.0020** | 0.214 | 0.098 | 0.427624 | 0.113689 | |
| F14 | Ave | −4.7062 | **0.0057** | −0.0023 | 0 | −0.002 | −3.642 | −3.6416 | −3.885 |
| | Std | 0.8910 | 0.0067 | 0.0052 | **0.0002** | 0.06 | 0.351 | 0.325 | 0.718 |
| | Med | −1.83 | −2.71 | −0.003 | **−0.0056** | | −3.608 | −3.582 | −4.076 |
| F15 | Ave | −0.4 | −1 | −2.654 | **0** | **0** | −0.5173 | −0.055483 | −0.474555 |
| | Std | 0.5477 | 0 | 0.69 | 39 | 0.87 | 0.3841 | **0.1351484** | 0.4856118 |
| | Med | −7.307 | −0.002 | −0.5 | −0.8 | −0.543 | −0.5908 | **−4.15e−109** | −0.414068 |
| F16 | Ave | −0.2 | −1 | −1.009 | 6.25 | **0.0001** | 3.198e−04 | 2.95e−04 | 0.001575 |
| | Std | 0.4472 | 0.2479 | 1.333 | **0.0001** | 0.0001 | 2.334e−04 | 0.000215 | 0.000818 |
| | Med | **5.95e−13** | −1 | 5.23 | 2.39 | 11.03 | 2.325e−04 | 0.000269 | 0.001325 |
| F17 | Ave | 100.09 | 1640.57 | **51.32** | 93.45 | 87.52 | 93.2475 | 194.8523 | 193.6682 |
| | Std | 122.47 | 88.023 | 87.36 | 77.677 | **53.987** | 64.2902 | 60.03402 | 121.9127 |
| | Med | 100.13 | 971.32 | 84.23 | **52.98** | 96.23 | 78.7049 | 176.0384 | 170.521 |
| F18 | Ave | 225.45 | 199.24 | **14.7** | 111.98 | 78.65 | 156.6317 | 146.7613 | 205.6785 |
| | Std | 115.23 | 113.58 | 231.245 | 33.54 | **9.54** | 31.8874 | 29.08005 | 160.9849 |
| | Med | 178.98 | 127.11 | **19.36** | 402 | 109.77 | 154.5892 | 140.6424 | 154.8682 |
| F19 | Ave | 438.99 | 324.25 | **9.02** | 147.33 | 89.574 | 149.6407 | 445.7764 | 384.7761 |
| | Std | 84.6687 | 12.36 | 8.64 | 89.23 | **11.47** | 38.7091 | 49.3449 | 118.0311 |
| | Med | 417.32 | 111.98 | **92.5** | 172.98 | 196.3 | 152.1527 | 443.3976 | 448.1912 |

(*continued*).

**Table 4.** Continued.

| No. | Measure | WOA | bWOA-S | bWOA-V | bGWO1 | bGWO2 | BBA | BPSO | GA |
|-----|---------|-----|--------|--------|-------|-------|-----|------|-----|
| F20 | Ave | 629.23 | 479.35 | 800.75 | 872.32 | **102**.002 | 146.9480 | 479.9867 | 588.1262 |
|     | Std | 144.4 | 214.21 | 47.65 | 67.12 | 77.37 | **22**.9687 | 30.19361 | 102.3373 |
|     | Med | 678.95 | 516.89 | 847.02 | 637.12 | **119**.014 | 147.0495 | 477.0193 | 639.9012 |
| F21 | Ave | 403.91 | **373**.09 | 510.57 | 245.62 | 514.09 | 616.1212 | 172.0816 | 426.3021 |
|     | Std | 172.58 | 53.49 | **11**.87 | 24.78 | 99.23 | 49.8056 | 64.2674 | 183.5344 |
|     | Med | 429.852 | 390.879 | **118**.21 | 360.25 | 161.02 | 163.1106 | 140.9928 | 218.0127 |
| F22 | Ave | 694.60 | 724.35 | **92**.65 | 174.98 | 564.25 | 152.8125 | 691.65 | 914.5375 |
|     | Std | 208.020 | 46.32 | 87.65 | 74.78 | 158.23 | 33.6342 | 149.6255 | **12.32191** |
|     | Med | 800.025 | 900253 | **109**.23 | 187.05 | 202.98 | 145.5394 | 607.9773 | 908.362 |
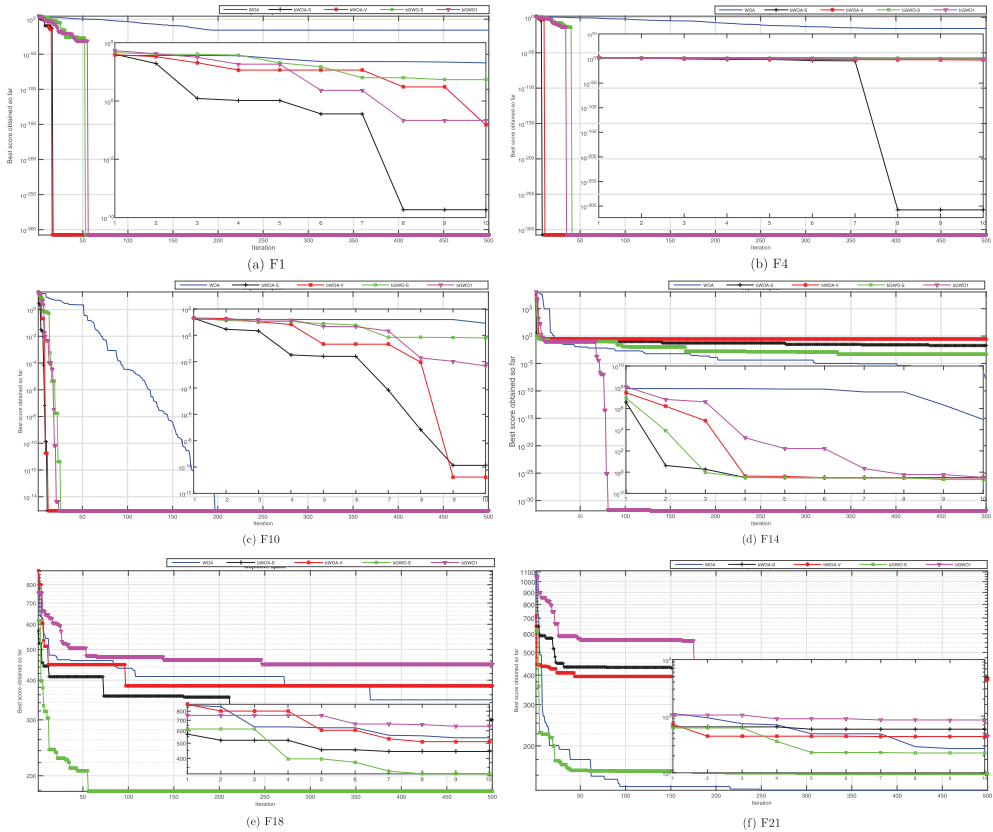


**Figure 3.** The convergence rate between bWOA and WOA on some benchmark functions. (*concluded*). (a) F1. (b) F4. (c) F10. (d) F14. (e) F18 and (f) F21.

convergence curves are averaged. Owing to space limitations and to improve the readability of the article, only six convergence curves, *i.e.* (F1, F4) from unimodal functions, (F10, F14) from multimodal functions and (F18, F21) from composite functions, are shown in Figure 3. This figure reveals that the bWOA outperforms the basic WOA and has faster convergence. Also, the bWOA is able to avoid local minima with a significant convergence rate for multimodal benchmark test functions. As shown by this figure, the bWOA has a faster convergence rate than the WOA and the capability of finding global solutions.

In summary, the statistical results of the 25 benchmark tests show that the proposed bWOA algorithm is very effective in solving constrained and engineering design problems.

**Table 5.** Optimization results and statistical results of applying the bWOA-S, the bWOA-V and the WOA to Engineering problems.

| | | Optimization results | | | | | Statistical results | |
|---|---|---|---|---|---|---|---|---|
| | Algorithm | *h* | *l* | *t* | *b* | Cost | Ave | Std |
| Welded beam design problem | WOA | 0.205396 | 3.484293 | 9.037426 | 0.206276 | 1.730499 | 1.7320 | 0.0226 |
| | bWOA-S | 0.21 | 4 | 8 | 0.28125 | **1.2246** | 1.727 | 0.0113 |
| | bWOA-V | 0.20 | 3.47 | 9.035 | 0.2011 | 1.724 | 1.725 | 0.004 |
| | | Optimization results | | | | | Statistical results | |
| | | *d* | *D* | *N* | | Cost | Ave | Std |
| Tension/compression design | WOA | 0.051207 | 0.345215 | 12.004032 | | 0.0126763 | 0.0127 | 0.1003 |
| | bWOA-S | 0.05 | 0.3125 | 11 | | **0.010281** | **0.01260** | 0.0180 |
| | bWOA-V | 0.05 | 0.3429 | 12.08 | | 0.01265 | 0.012699 | **0.001** |
| | | Optimization results | | | | | Statistical results | |
| | | $T_s$ | $T_h$ | **R** | *l* | Cost | Ave | Std |
| Pressure vessel | WOA | 0.812500 | 0.4375 | 42.0982 | 176.638 | 6059.7410 | 6068.05 | 65.6519 |
| | bWOA-S | 1.05 | 0.781 | 40.4525 | 198.002 | 5890.32 | **5658.964** | **29.6521** |
| | bWOA-V | 0.7780 | 0.3831 | 40.315 | 200 | **5880.1642** | 5891.964 | 327.007 |

**Table 6.** Parameter settings.

| Parameter | Value |
|---|---|
| Number of iterations | 500 |
| Number of search agents | 20 |
| Number of run repetitions | 30 |
| Crossover, mutation | 0.9, 0.05 |

**Table 7.** Wilcoxon's rank sum test.

| | No. | *p*-value | No. | *p*-value | No. | *p*-value |
|---|---|---|---|---|---|---|
| bWOA-S versus WOA | F1 | 0.000 | F9 | 0.000 | F17 | 0.237 |
| | F2 | 0.000 | F10 | 0.000 | F18 | 0.216 |
| | F3 | 0.000 | F11 | 0.000 | F19 | 0.326 |
| | F4 | 0.000 | F12 | 0.000 | F20 | 0.480 |
| | F5 | 0.000 | F13 | 0.102 | F21 | 0.932 |
| | F6 | 0.000 | F14 | 0.000 | F22 | 0.742 |
| | F7 | 0.000 | F15 | 0.652 | | |
| | F8 | 0.000 | F16 | 0.204 | | |
| bWOA-V versus WOA | F1 | 0.000 | F9 | 0.000 | F17 | 0.365 |
| | F2 | 0.000 | F10 | 0.000 | F18 | 0.932 |
| | F3 | 0.000 | F11 | 0.000 | F19 | 0.265 |
| | F4 | 0.000 | F12 | 0.082 | F20 | 0.265 |
| | F5 | 0.000 | F13 | 0.000 | F21 | 0.821 |
| | F6 | 0.000 | F14 | 0.000 | F22 | 0.661 |
| | F7 | 0.000 | F15 | 0.346 | | |
| | F8 | 0.000 | F16 | 0.000 | | |

### 5.6. Wilcoxon's rank sum

In order to compare the results of each run, a non-parametric statistical test known as Wilcoxon's rank sum test for independent samples (Wilcoxon 1945) is performed over all functions at the 5% significance level and the *p*-values are reported in Table 7.

The *p*-values show that, on most functions except F15, F21 and F22 for the bWOA-S, and F22 for the bWOA-V, a great and significant difference is exhibited.

**Table 8.** Results of the proposed bWOA-S, bWOA-V and WOA over five TSP benchmarks.

| Dataset name | Optima | bWOA-S | bWOA-V | WOA |
|---|---|---|---|---|
| KroA100 | 21282 | **21345** | 21647 | 21989 |
| KroB100 | 22140 | 22406 | **22308** | 22842 |
| KroC100 | 20749 | **21070** | 21320 | 21476 |
| KroD100 | 21294 | **21596** | 22013 | 22083 |
| KroE100 | 22068 | 22771 | **22439** | 22970 |

## 6. The travelling salesman problem (TSP)

This algorithm can be applied to many real-world problems, such as the water pump switching problem and the optimal scheduling of a multiple dam system (Geem 2005, 2007). In this section, the TSP is used as a real-world application. The TSP is considered to be one of the most common and classical examples of a combinatorial optimization problem, and has been proved to be $\mathcal{NP}$-hard. The objective in the TSP is to find the salesman's optimal tour to visit all cities once and only once and return to the start city after travelling the shortest possible distance. Here, the symmetrical TSP is used, *i.e.* the distance from city u to city v is the same as from city v to city u (Zurada 1992). Five cities from benchmark designs especially for the TSP are used (Reinelt 1991). The name of the five benchmark datasets and their results are shown in Table 8. The results show that both the bWOA-S and the bWOA-V have better results than the original WOA algorithm in all datasets.

## 7. Conclusion and future work

Binary versions of the WOA, namely the bWOA-S and the bWOA-V, have been proposed via transfer functions. They are compared with five well-known metaheuristic algorithms, namely the BPSO, the BBA, the bGWO1, the bGWO2 and a GA, over 22 benchmark functions and three engineering design problems to investigate their evaluation performance. The statistical comparison results revealed that both novel versions give better performance than the comparison algorithms. Furthermore, the results prove that they are worthy of being classed as binary optimization algorithms. Moreover, the convergence curves for the proposed algorithms when compared with the native WOA reveal that the bWOAs have faster convergence rates. The focus of further research will apply the bWOAs to real-life optimization problems with different transfer functions.

## Disclosure statement

## ORCID

*Abdelazim G. Hussien* http://orcid.org/0000-0001-5394-0678
*Aboul Ella Hassanien* http://orcid.org/0000-0002-9989-6681
*Essam H. Houssein* http://orcid.org/0000-0002-8127-7233
*Ahmad Taher Azar* http://orcid.org/0000-0002-7869-6373

## References

Babaoglu, İsmail, Oğuz Findik, and Erkan Ülker. 2010. "A Comparison of Feature Selection Models Utilizing Binary Particle Swarm Optimization and Genetic Algorithm in Determining Coronary Artery Disease Using Support Vector Machine." *Expert Systems with Applications* 37 (4): 3177–3183.
Beheshti, Zahra, Siti Mariyam Shamsuddin, and Siti Sophiayati Yuhaniz. 2013. "Binary Accelerated Particle Swarm Algorithm (BAPSA) for Discrete Optimization Problems." *Journal of Global Optimization* 57 (2): 549–573. doi:10.1007/s10898-012-0006-1.
Belegundu, Ashok D., and Jasbir S. Arora. 1985. "A Study of Mathematical Programming Methods for Structural Optimization. Part I: Theory." *International Journal for Numerical Methods in Engineering* 21 (9): 1583–1599.

Beşkirli, Mehmet, İsmail Koç, Hüseyin Haklı, and Halife Kodaz. 2018. "A New Optimization Algorithm for Solving Wind Turbine Placement Problem: Binary Artificial Algae Algorithm." *Renewable Energy* 121: 301–308. doi:10.1016/j.renene.2017.12.087.

Dahi, Zakaria Abd El Moiz, Chaker Mezioud, and Amer Draa. 2016. "On the Efficiency of the Binary Flower Pollination Algorithm: Application on the Antenna Positioning Problem." *Applied Soft Computing* 47: 395–414.

Eid, Heba F. 2018. "Binary Whale Optimisation: An Effective Swarm Algorithm for Feature Selection." *International Journal of Metaheuristics* 7 (1): 67–79.

El Aziz, Mohamed Abd, Ahmed A. Ewees, and Aboul Ella Hassanien. 2017. "Whale Optimization Algorithm and Moth–Flame Optimization for Multilevel Thresholding Image Segmentation." *Expert Systems with Applications* 83: 242–256.

Emary, E., and Hossam M. Zawbaa. 2018. "Feature Selection via Lèvy Antlion Optimization." *Pattern Analysis and Applications*. doi:10.1007/s10044-018-0695-2.

Emary, Eid, Hossam M. Zawbaa, and Aboul Ella Hassanien. 2016a. "Binary Ant Lion Approaches for Feature Selection." *Neurocomputing* 213: 54–65.

Emary, Eid, Hossam M. Zawbaa, and Aboul Ella Hassanien. 2016b. "Binary Grey Wolf Optimization Approaches for Feature Selection." *Neurocomputing* 172: 371–381.

Fan, Kun, Weijia You, and Yuanyuan Li. 2013. "An Effective Modified Binary Particle Swarm Optimization (mBPSO) Algorithm for Multi-Objective Resource Allocation Problem (MORAP)." *Applied Mathematics and Computation* 221: 257–267.

Geem, Zong Woo. 2005. "Harmony Search in Water Pump Switching Problem." In *Proceedings of the First International Conference on Natural Computation (ICNC 2005)*, 751–760. Cham, Switzerland: Springer Nature.

Geem, Zong Woo. 2007. "Optimal Scheduling of Multiple Dam System Using Harmony Search Algorithm." In *Proceedings of the 9th International Work-Conference on Artificial Neural Networks (IWANN '2007)*, 316–323. Berlin: Springer. doi:10.1007/978-3-540-73007-1_39.

Geem, Zong Woo, Joong Hoon Kim, and Gobichettipalayam Vasudevan Loganathan. 2001. "A New Heuristic Optimization Algorithm: Harmony Search." *Simulation* 76 (2): 60–68.

Gong, Dunwei, Xinfang Ji, Jing Sun, and Xiaoyan Sun. 2014. "Interactive Evolutionary Algorithms with Decision-Maker's Preferences for Solving Interval Multi-Objective Optimization Problems." *Neurocomputing* 137: 241–251.

Hassanien, Aboul Ella, and Eid Emary. 2016. *Swarm Intelligence: Principles, Advances, and Applications*. Boca Raton, FL: CRC Press.

Holland, John H. 1975. *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence*. Ann Arbor, MI: University of Michigan Press.

Holland, John H. 1992. *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence*. Reprint Edition. Cambridge, MA: MIT/A Bradford Book.

Hussien, Abdelazim G., Aboul Ella Hassanien, and Essam H. Houssein. 2017. "Swarming Behaviour of SALPS Algorithm for Predicting Chemical Compound Activities." In *Proceedings of the Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*. Piscataway, NJ: IEEE. doi:10.1109/INTELCIS.2017.8260072.

Hussien, Abdelazim G., Aboul Ella Hassanien, Essam H. Houssein, Siddhartha Bhattacharyya, and Mohamed Amin. 2019. "S-Shaped Binary Whale Optimization Algorithm for Feature Selection." In *Recent Trends in Signal and Image Processing*, 79–87. Singapore: Springer Nature. doi:10.1007/978-981-10-8863-6_9.

Hussien, Abdelazim G., Essam H. Houssein, and Aboul Ella Hassanien. 2017. "A Binary Whale Optimization Algorithm with Hyperbolic Tangent Fitness Function for Feature Selection." In *Proceedings of the Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, 166–172. Piscataway, NJ: IEEE.

Jadhav, Amolkumar Narayan, and N. Gomathi. 2018. "WGC: Hybridization of Exponential Grey Wolf Optimizer with Whale Optimization for Data Clustering." *Alexandria Engineering Journal* 57 (3): 1569–1584.

Kennedy, James, and Russell Eberhart. 1995. "Particle Swarm Optimization." In *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 4, 1942–1948. Piscataway, NJ: IEEE.

Kennedy, James, and Russell C. Eberhart. 1997. "A Discrete Binary Version of the Particle Swarm Algorithm." In *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics: Computational Cybernetics and Simulation*, Vol. 5, 4104–4108. Piscataway, NJ: IEEE.

Kramer, S. N. 1994. "An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design." *Journal of Mechanical Design* 116 (2): 405–411. doi:10.1115/1.2919393.

Krause, Jonas, Jelson Cordeiro, Rafael Stubs Parpinelli, and Heitor Silverio Lopes. 2013. "A Survey of Swarm Algorithms Applied to Discrete Optimization Problems." In *Swarm Intelligence and Bio-Inspired Computation*, 169–191. Amsterdam: Elsevier.

Lau, Henry C. W., T. M. Chan, W. T. Tsui, and W. K. Pang. 2010. "Application of Genetic Algorithms to Solve the Multidepot Vehicle Routing Problem." *IEEE Transactions on Automation Science and Engineering* 7 (2): 383–392.

Lee, Youngjoo, and Jeongjin Lee. 2015. "Binary Tree Optimization Using Genetic Algorithm for Multiclass Support Vector Machine." *Expert Systems with Applications* 42 (8): 3843–3851.

Mafarja, Majdi M., and Seyedali Mirjalili. 2017. "Hybrid Whale Optimization Algorithm with Simulated Annealing for Feature Selection." *Neurocomputing* 260: 302–312.

Mirjalili, Seyedali. 2015. "The Ant Lion Optimizer." *Advances in Engineering Software* 83: 80–98.

Mirjalili, Seyedali. 2016. "Dragonfly Algorithm: A New Meta-Heuristic Optimization Technique for Solving Single-Objective, Discrete, and Multi-Objective Problems." *Neural Computing & Applications* 27 (4): 1053–1073. doi:10.1007/s00521-015-1920-1.

Mirjalili, Seyedali, and Andrew Lewis. 2016. "The Whale Optimization Algorithm." *Advances in Engineering Software* 95: 51–67.

Mirjalili, Seyedali, Seyed Mohammad Mirjalili, and Xin-She Yang. 2014. "Binary Bat Algorithm." *Neural Computing and Applications* 25 (3–4): 663–681.

Obagbuwa, Ibidun Christiana, and Ademola Philips Abidoye. 2016. "Binary Cockroach Swarm Optimization for Combinatorial Optimization Problem." *Algorithms* 9 (3): 59. doi:10.3390/a9030059.

Pal, Avishek, and J. Maiti. 2010. "Development of a Hybrid Methodology for Dimensionality Reduction in Maha-lanobis–Taguchi System Using Mahalanobis Distance and Binary Particle Swarm Optimization." *Expert Systems with Applications* 37 (2): 1286–1293.

Rashedi, Esmat, Hossein Nezamabadi-Pour, and Saeid Saryazdi. 2009. "GSA: A Gravitational Search Algorithm." *Information Sciences* 179 (13): 2232–2248.

Ray, Tapabrata, and K. M. Liew. 2002. "A Swarm Metaphor for Multiobjective Design Optimization." *Engineering Optimization* 34 (2): 141–153.

Reddy K., Srikanth, Lokesh Panwar, B. K. Panigrahi, and Rajesh Kumar. 2018. "Binary Whale Optimization Algorithm: A New Metaheuristic Approach for Profit-Based Unit Commitment Problems in Competitive Electricity Markets." *Engineering Optimization* 51 (3): 369–389. doi:10.1080/0305215X.2018.1463527.

Reinelt, Gerhard. 1991. "TSPLIB—A Traveling Salesman Problem Library." *INFORMS Journal on Computing* 3 (4): 376–384.

Saka, Mehmet Polat, O. Hasançebi, and Zong Woo Geem. 2016. "Metaheuristics in Structural Optimization and Discussions on Harmony Search Algorithm." *Swarm and Evolutionary Computation* 28: 88–97.

Saremi, Shahrzad, Seyedali Mirjalili, and Andrew Lewis. 2017. "Grasshopper Optimisation Algorithm: Theory and Application." *Advances in Engineering Software* 105: 30–47.

Tayarani-N, Mohammad-H, and M. R. Akbarzadeh-T. 2008. "Magnetic Optimization Algorithms a New Synthesis." In *Proceedings of the IEEE World Congress on Evolutionary Computation (CEC 2008)—Computational Intelligence*, 2659–2664. Piscataway, NJ: IEEE. doi:10.1109/CEC.2008.4631155.

Tharwat, Alaa, Essam H. Houssein, Mohammed M. Ahmed, Aboul Ella Hassanien, and Thomas Gabel. 2017. "MOGOA Algorithm for Constrained and Unconstrained Multi-Objective Optimization Problems." *Applied Intelligence* 48 (8): 2268–2283. doi:10.1007/s10489-017-1074-1.

Wang, Jianzhou, Pei Du, Tong Niu, and Wendong Yang. 2017. "A Novel Hybrid System Based on a New Proposed Algorithm—Multi-Objective Whale Optimization Algorithm for Wind Speed Forecasting." *Applied Energy* 208: 344–360. doi:10.1016/j.apenergy.2017.10.031.

Wilcoxon, Frank. 1945. "Individual Comparisons by Ranking Methods." *Biometrics Bulletin* 1 (6): 80–83.

Wolpert, David H., and William G. Macready. 1997. "No Free Lunch Theorems for Optimization." *IEEE Transactions on Evolutionary Computation* 1 (1): 67–82.

Yang, Xin-She. 2010a. *Nature-Inspired Metaheuristic Algorithms*. Frome, UK: Luniver Press.

Yang, Xin-She. 2010b. "A New Metaheuristic Bat-Inspired Algorithm." *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 65–74. Berlin: Springer-Verlag. doi:10.1007/978-3-642-12538-6_6.

Yuan, Xiaohui, Bin Ji, Shuangquan Zhang, Hao Tian, and Yanhong Hou. 2014. "A New Approach for Unit Commitment Problem via Binary Gravitational Search Algorithm." *Applied Soft Computing* 22: 249–260. doi:10.1016/j.asoc.2014.05.029.

Zawbaa, Hossam M., Eid Emary, Crina Grosan, and Vaclav Snasel. 2018. "Large-Dimensionality Small-Instance Set Feature Selection: A Hybrid Bio-Inspired Heuristic Approach." *Swarm and Evolutionary Computation* 42: 29–42.

Zurada, Jacek M. 1992. *Introduction to Artificial Neural Systems*. Vol. 8. St Paul, MN: West Publishing.