



Recent Advances in Butterfly Optimization Algorithm, Its Versions and Applications

Sharif Naser Makhadmeh¹ · Mohammed Azmi Al-Betar¹ · Ammar Kamal Abasi² · Mohammed A. Awadallah^{3,4} · Iyad Abu Doush^{5,6} · Zaid Abdi Alkareem Alyasseri^{7,8,9} · Osama Ahmad Alomari¹⁰

Received: 16 February 2022 / Accepted: 15 October 2022 / Published online: 3 November 2022
© The Author(s) under exclusive licence to International Center for Numerical Methods in Engineering (CIMNE) 2022

Abstract

The butterfly optimization algorithm (BOA) is a recent successful metaheuristic swarm-based optimization algorithm. The BOA has attracted scholars' attention due to its extraordinary features. Such as the few adaptive parameters to handle and the high balance between exploration and exploitation. Accordingly, the BOA has been extensively adapted for various optimization problems in different domains in a short period. Therefore, this paper reviews and summarizes the recently published studies that utilized the BOA for optimization problems. Initially, introductory information about the BOA is presented to illustrate the essential foundation and its relevant optimization concepts. In addition, the BOA inspiration and its mathematical model are provided with an illustrative example to prove its high capabilities. Subsequently, all reviewed studies are classified into three main classes based on the adaptation form, including original, modified, and hybridized. The main BOA applications are also thoroughly explained. Furthermore, the BOA advantages and drawbacks in dealing with optimization problems are analyzed. Finally, the paper is summarized in conclusion with the future directions that can be investigated further.

Abbreviations

ABC	Artificial bee colony	GWO	Grey wolf optimizer
ACO	Ant colony optimization	KH	Krill herd
ANN	Artificial neural networks	MVO	Multi-verse optimizer
BOA	Butterfly optimization algorithm	NFL	No free lunch
CHIO	Coronavirus herd immunity optimizer	PSO	Particle swarm optimization
CS	Cuckoo search	PV	Photovoltaic
FA	Firefly algorithm	SI	Swarm intelligence
FS	Feature selection	WSN	Wireless sensor networks
GA	Genetic algorithm		

✉ Sharif Naser Makhadmeh
s.makhadmeh@ajman.ac.ae

¹ Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman, United Arab Emirates

² Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence (MBZUAI), Abu Dhabi, United Arab Emirates

³ Department of Computer Science, Al-Aqsa University, P.O. Box 4051, Gaza, Palestine

⁴ Artificial Intelligence Research Center (AIRC), Ajman University, Ajman, United Arab Emirates

⁵ Computing Department, College of Engineering and Applied Sciences, American University of Kuwait, Salmiya, Kuwait

⁶ Computer Science Department, Yarmouk University, Irbid, Jordan

⁷ ECE Department-Faculty of Engineering, University of Kufa, P.O. Box 21, An Najaf, Iraq

⁸ Department of Business Administration, College of Administrative and Financial Sciences, Imam Ja'afar Al-Sadiq University, Baghdad, Iraq

⁹ Information Technology Research and Development Center (ITRDC), University of Kufa, An Najaf 54001, Iraq

¹⁰ MLALP Research Group, University of Sharjah, P.O. Box 27272, Sharjah, United Arab Emirates

1 Introduction

Optimization is ordinarily concerned with finding the optimal configurations for a set of decision variables to minimize or maximize specific criteria formulated in an objective function [1]. Optimization problems can be utilized in different variants to deal with various types of search space, such as binary, continuous, discrete, and hierarchical [2]. The search space includes the set of all possible solutions to any optimization problem. An optimization problem can be unimodal or multimodal, where the front has only one niche while the rear has several search space niches with different local optima [3]. The optimization method can navigate the search space niches using two main processes: exploration and exploitation. Exploration is the ability to navigate not-yet-visited niches, while exploitation is the ability to drill down/up the already-visited niches [4]. Exploration and exploitation are contradictory, where the primary purpose is to find a suitable trade-off between them during the search for powerful utilization.

The optimization problems are initially tackled by calculus-based methods, such as linear programming [5], integer programming [6], branch and bound [7], and many others. These traditional methods are potent for problems with small dimension search spaces. However, their performance is reduced with larger dimensions of the search space [8]. Therefore, the heuristic and metaheuristic-based methods have gained substantial attention from the optimization research communities due to their efficiency and easy-to-use features. Heuristic-based methods are very fast in tackling optimization problems. However, they are problem-dependent and solve the problem as a constraint satisfaction method with less consideration of the solution quality [9]. The metaheuristic-based methods work with high dimensional search space trying to find a solution with good-enough quality [1, 10].

Metaheuristics are a group of algorithms that work as general optimization frameworks to explore and exploit problems in search spaces using intelligent operators making use of accumulative knowledge gained during the search controlled by tuned parameters and initiated by some algorithmic parameters [9]. They are conventionally categorized in different ways: natural-inspired vs non-natural-inspired, memory-use vs memory-less, static vs dynamic objective function, population-based vs trajectory-based [8]. Recently, the research community agreed to categorize the population-based meta-heuristic based on their inspiration into human-based, physical-based, swarm intelligence, and evolutionary-based [1, 11]. More information about these types of metaheuristics categories can be found in [2].

The swarm intelligence (SI) category has recently gained substantial attention from the optimization research community due to its simple features. SI algorithms are normally inspired by the natural living behavior of some animals that live in groups or swarms, such as birds, fishes, wolves, salps, ants, bees, bacteria, and insects [12]. The common behavior of the SI algorithms is the interaction between the swarm members, where normally they follow the leader(s) to reach the food source or to hunt the prey [13]. The base algorithm of SI is particle swarm optimization (PSO) which was proposed in 1996 by Kennedy and Eberhart [13]. It mimics the motion behavior of bird flocks and schooling fish. The ant colony optimization (ACO) is also a powerful SI algorithm proposed by Dorigo, Marco, Mauro Birattari, and Thomas Stutzle [14]. Other SI algorithms include krill herd optimization (KH) [15], cuckoo search (CS) [16], multi-verse optimizer (MVO) [17], group search optimizer [18], grey wolf optimizer (GWO) [19], Coronavirus herd immunity optimizer (CHIO) [20], and others reported in [13].

The butterfly optimization algorithm (BOA) is a recent SI algorithm that imitates the butterfly food foraging behaviour in nature [21]. The butterfly techniques in searching for food are formulated as an optimization method based on global and local movements. The BOA reveals powerful features as other SI algorithms: it is simple in adaptation for any optimization problem, easy to use, derivative-free, less tuned parameters, flexible, scalable, and sound-and-complete. Therefore, BOA has been adapted and utilized for a wide range of optimization problems, such as feature selection [22], photovoltaic models [23], early search blindness [24], energy consumption [25], image segmentation [25], scheduling [26], medical data classification [27], sentiment analysis [28], engineering problems [29] and many others as summarized in Table 2.

As stated by the No Free Lunch (NFL) theorem [30], no single algorithm can yield a superior outcome for all optimization problems or even for instances of the same problem. Therefore, the basic form of any SI algorithm, such as BOA, can be modified or hybridized to align with the problem-solving requirements. Thus, several problem-dependent versions of BOA has been recently proposed to enhance its searching behavior, such as binary BOA [31], discrete BOA [26], chaotic BOA [32], adaptive BOA [23], bidirectional BOA [33], dynamic BOA [34], and others [29, 35–37].

This review paper intensively analyzes all works related to the BOA in which six pillars are addressed as follows:

1. The growth of BOA in the literature in terms of the number of BOA-related articles, citations, authors, countries, institutions, and topics since 2016 is presented in Sect. 2.
2. The original foundation BOA is demonstrated in terms of inspirations, procedural steps, exploration, and

exploitation. A simple example with step-by-step explanations is given in Sect. 3.

- All versions of BOA are discussed with their related works to show the adaptability features of BOA in Sect. 4.
- All applications tackled by BOA-versions are also illustrated in Sect. 5.
- The behavior of BOA in tackling optimization problems is criticized and analyzed to show the pros and cons of BOA. In addition, the motivation behind the variations of BOA is shown in Sect. 6.
- Finally, the fruitful review of the current works and the directions for future enhancements are presented in Sect. 7.

2 The Growth of Butterfly Optimization Algorithm in the Literature

To study the progression and growth of BOA in the literature, this section summarizes the usage of BOA in terms of scholarly publications. Scopus database is the most effective indexer nowadays to collect such materials. The search in Scopus is based on the following criteria:

- The query in the search is “Butterfly optimization algorithm”
- The dates are limited to the articles from 2016 until 2022.
- The date of collecting the results is 15/02/2022.

In summary, the whole query of searching executed in Scopus database is as follows: TITLE (“Butterfly optimization algorithm”) AND (LIMIT-TO (PUBYEAR, 2022) OR LIMIT-TO (PUBYEAR, 2021) OR LIMIT-TO (PUBYEAR, 2020) OR LIMIT-TO (PUBYEAR, 2019) OR LIMIT-TO (PUBYEAR, 2018) OR LIMIT-TO (PUBYEAR, 2017) OR LIMIT-TO (PUBYEAR, 2016))

Initially, the growth of BOA in terms of publication per year is shown in Fig. 1. Researchers in different domains considered the BOA the main algorithm during the years mentioned. This consideration is sharply increased, where the number of BOA-based publications was two in 2016 and grew to 34 by the end of 2021. By the end of 2022, BOA will gain much attention from the research community, which might duplicate the number of published articles in 2021.

Regarding citation gain for BOA publications shown in Fig. 2, the number grew from 11 citations in 2017 to 411 in 2021. The total number of citations is expected to be duplicated by the end of 2022. This exponential growth in BOA citations results from the BOA’s efficiency and effectiveness when solving optimization problems.

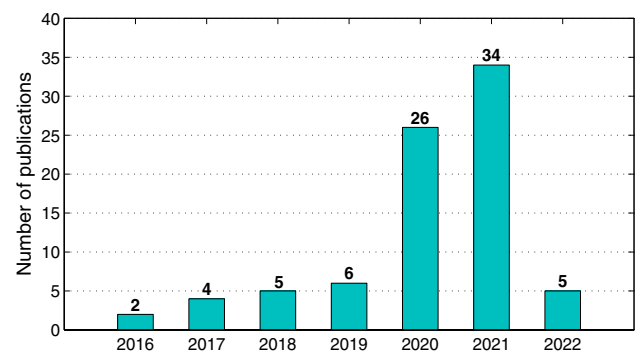


Fig. 1 The number of BOA-based publications per year

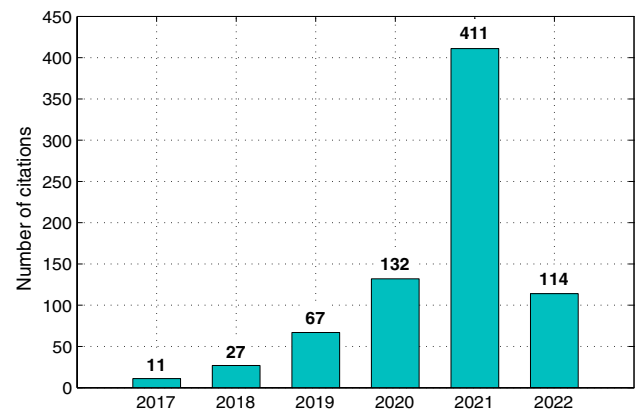


Fig. 2 The number of citations per year for BOA-based publications

In the pie chart shown in Fig. 3, the growth of BOA is visualized concerning the research topics, and each slice in the pie chart reflects the number of articles per topic. As shown in the pie chart, the most significant percentage of articles is published on the computer science topic, which is equal to 30%. The second topic that gained much interest is engineering with 28%. The mathematics topic is ranked third with 12%. The remaining publication percentage is covered by other topics such as medicine, chemistry, business, energy, etc.

The BOA-based articles are published in well-reputed journals, conferences, and book chapters on well-established publishers, such as Springer-Link, ScienceDirect, MDPI, IEEE, world scientific, Taylor & Francis, Hindawi, etc., as shown in Fig. 4. Publishing on these publishers with a high reputation is due to the solid theory of BOA and its impressive characteristics.

Using country as another dimension, the number of BOA-based articles published by Indian researchers is equal to 27, where BOA and its versions gained the highest interest, as shown in Fig. 5. China is the second country that considered BOA and its variants to solve optimization problems, published 25 articles since 2016. Other countries have published

Fig. 3 The number of BOA-based publications published by each subject

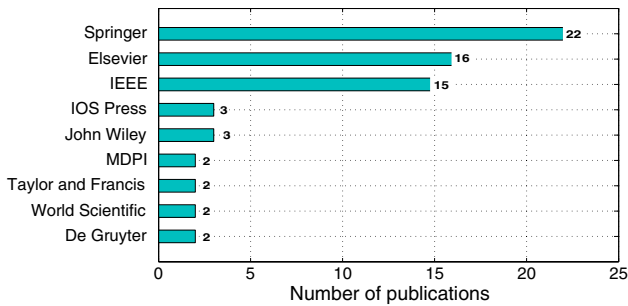
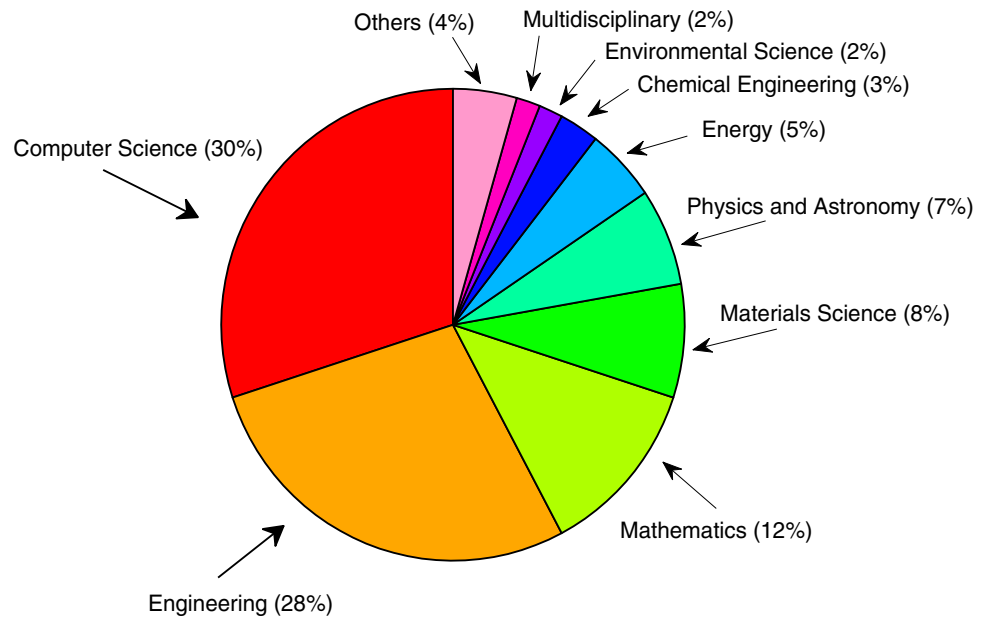


Fig. 4 The number of BOA-based publications published by each publisher

BOA in less than 15 articles each, as shown in Fig. 5. This shows the popularity of applying BOA by researchers from different countries.

The top ten authors who published BOA articles are shown in Fig. 6. As can be noticed, the BOA founders have published five articles on various topics and different optimization problems.

Finally, the top ten institutions that considered the BOA-based articles in their research are bar-charted in Fig. 7. The National Natural Science Foundation of China produced the highest number of BOA-based articles, which are 13 articles. Guizhou Key Laboratory of Big Data Statistics Analysis is

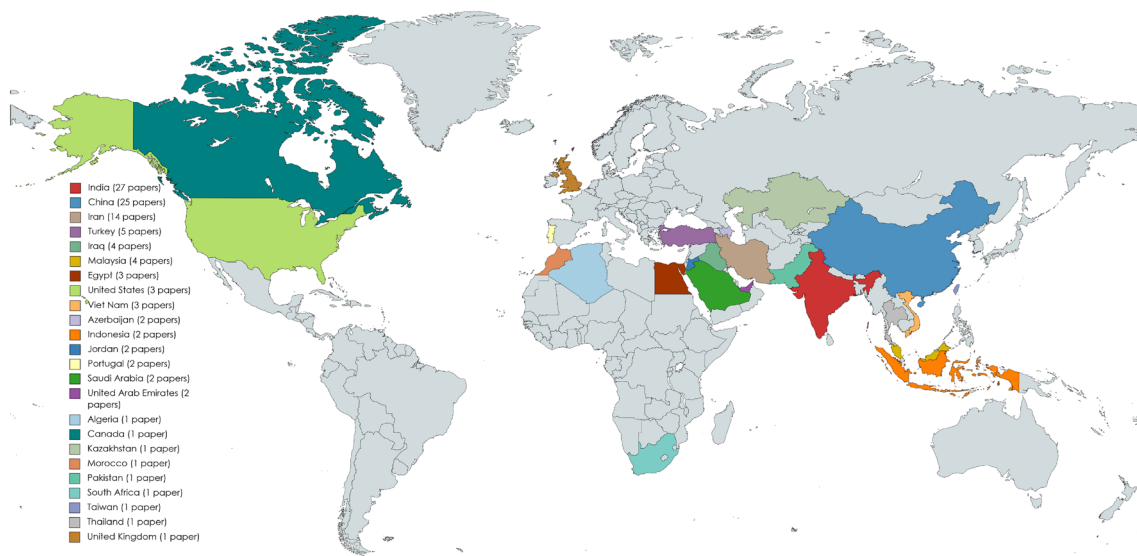


Fig. 5 The number of BOA-based publications of each country

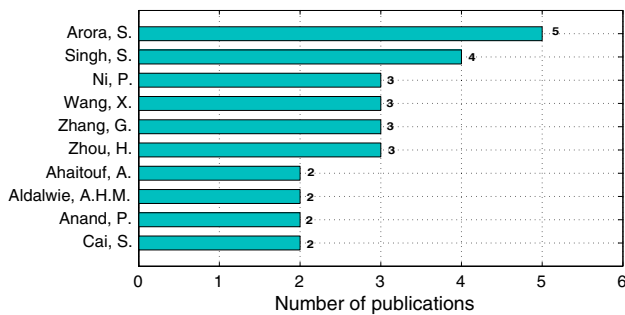


Fig. 6 The number of BOA-based publications of top eight authors

the second-highest institution that published BOA-based articles, which are two. Other institutions are shown in the bar-chart visualized in Fig. 7.

3 Basic Concepts of Butterfly Optimization Algorithm

The BOA is a robust swarm-based metaheuristic that mimics butterfly food foraging behaviour. This section illustrates the inspiration of BOA and its mathematical model. In addition, the BOA’s capabilities in exploring and exploiting search spaces are presented. Furthermore, an illustrative example that describes the algorithm searching steps is exhibited.

3.1 Butterfly Optimization Algorithm Inspiration

Butterflies have more the 18,000 species around the world [38]. These species can interact with each other using five senses that let them smell, sight, taste, touch, and hear. In addition, these senses can help them change home, reproduce, and find food. The smell sense is essential due to its effect on finding food and nectar from long-distance [39]. The butterflies have sense receptors scattered over the butterfly’s body used to smell. These receptors contain nerve

cells, called chemoreceptors, which are also used to find the best mating partner [40].

Generally, butterflies can locate the fragrance source accurately and distinguish different fragrances. Butterflies move from their location to other locations with more nectar. The butterflies produce fragrance while moving to share their current location and personal information with other butterflies. Subsequently, butterflies will follow the produced fragrance to find a better location with more nectar [21]. The general behavior of butterflies is presented in Fig. 8.

3.2 Butterfly Optimization Algorithm Mathematical Model

The butterflies’ inspiration and movement behaviour can be formulated as an optimization method, where butterflies present the search agents and the produced fragrances denote the fitness values.

In BOA, the butterflies/search agents can generate fragrance/fitness value with some power to be distinguished from other fragrances. Such behaviour can help other search agents update their position in the search space. Once the

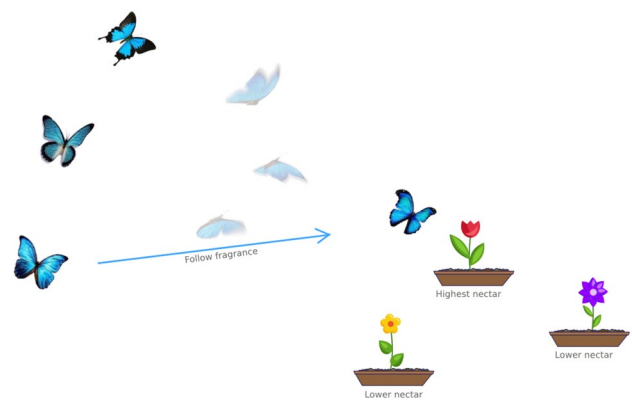
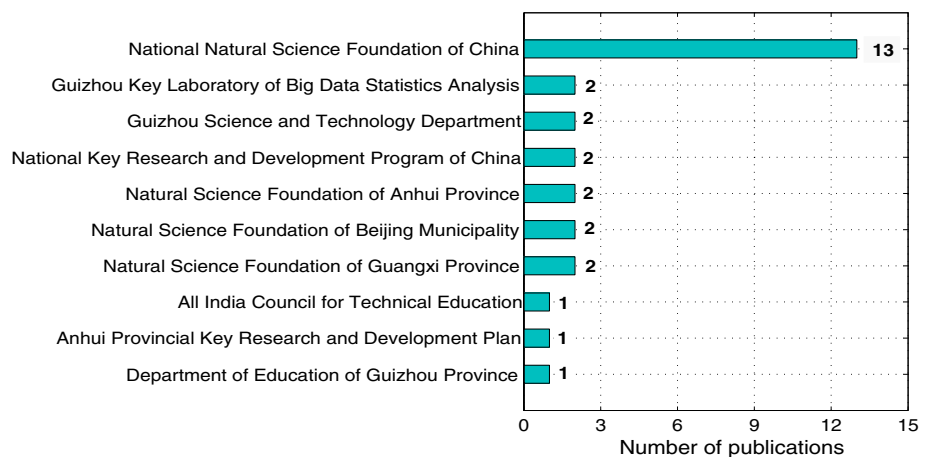


Fig. 8 Butterflies movement behavior

Fig. 7 The number of BOA-based publications per Institution



butterfly which finds the best nectar food in the search space produces a fragrance, all neighbourhood butterflies will move towards the best butterfly location. Such an updating mechanism is called global search in BOA. On the other hand, the butterflies will move randomly in the search space if other butterflies' fragrances are detected, known as local search in BOA.

The fragrance intensity is mathematically modeled as follows:

$$pf_i = cI^a, \tag{1}$$

where pf_i denotes the fragrance strength of i th butterfly, I presents the stimulus intensity, c denotes the sensory modality, and a is the power exponent depending on modality, which presents a varying absorption degree. Each butterfly's location is presented as a vector of particular problem values. Such a location can be updated in attempting to find a better location using the following formula:

$$x_i^{t+1} = x_i^t + F_i^{t+1}, \tag{2}$$

where x_i^t denotes the current position of butterfly i in iteration t , x_i^{t+1} is the next position of butterfly i and F_i^{t+1} denotes fragrance that utilized by x_i to update its position during iterations.

As mentioned previously, the updating mechanism can be in two phases, including local and global searches. In the global search, the butterfly i moves toward the fittest butterfly g^* , which can be represented as:

$$F_i^{t+1} = (r^2 \times g^* - x_i^t) \times pf_i, \tag{3}$$

where r is a random number in $[0, 1]$. In local search, the updating movement can be formulated as follows:

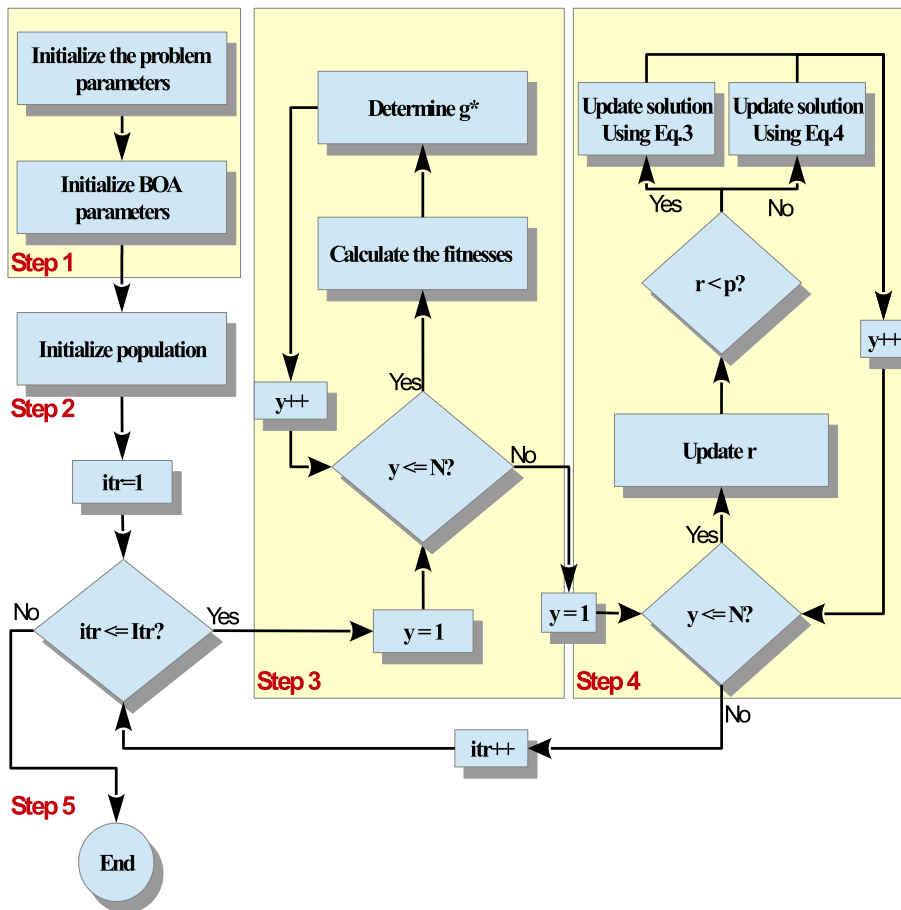
$$F_i^{t+1} = (r^2 \times x_j^t - x_k^t) \times pf_i, \tag{4}$$

where x_j^t and x_k^t denote the positions of j th and k th butterflies in the search space. A new parameter, called switch probability p , is utilized in BOA to switch the algorithm's behaviour between local and global search to get the best balance between exploration and exploitation.

3.3 Butterfly Optimization Algorithm Optimization Steps

In this section, the BOA general optimization steps are presented and illustrated. The BOA consists of five main steps, which are graphically presented in Fig. 9 and thoroughly discussed below.

Fig. 9 Flowchart of the general BOA steps



Step 1: BOA and the problem parameters initialization.

In this step, all BOA and the problem parameters are initialized. BOA has five parameters, including population size (N), number of iterations (Itr), c , a , and p .

Step 2: Population initialization.

All solutions are generated randomly by the BOA in this step. The solutions are presented as vectors of length equal to the problem dimension d .

All solutions are located in a matrix to create the population, as shown in Eq. 5.

$$\text{Population} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^N & x_2^N & \dots & x_d^N \end{bmatrix}, \quad (5)$$

Step 3: Fitness value calculation.

In this step, all solutions are evaluated based on the objective function of the optimization problem. Subsequently, the best solution is assigned to g^* .

Step 4: Update the population.

In this step, all solutions are updated using the BOA to find better solutions based on the fitness values obtained in Step 3. In the BOA, a random number r is generated and compared with p to lead the searching behaviour locally or globally. In case r is less than p , the butterfly moves globally using Eq. 3; otherwise, it moves locally using Eq. 4. Subsequently, if the new solution is better than the old one, it replaces the old solution. Finally, g^* will be updated.

Step 5: Check the stop condition

Steps 3 and 4 are reproduced until the maximum number of iterations is reached.

Algorithm 1 shows the pseudo-code of the general BOA steps. The BOA is coded in MATLAB, which is released as open-source [41]. The code addressed standard benchmark problems, which increased the BOA popularity.

Algorithm 1 BOA Pseudo-code

```

1: Step 1
2: Initialize the problem parameters
3: Initialize the BOA parameters ( $Itr, N, c, a, p$ )
4: Step 2
5: Initialize population matrix
6: Step 3
7: while ( $itr \leq Itr$ ) do
8:   for each solution do
9:     Calculate the solution's fitness value
10:     $g^*$  = the best solution
11:   end for
12:   Step 4
13:   for each solution do
14:     Generate  $r$  (random numbers in  $[0, 1]$ )
15:     if  $r < p$  then
16:       Update the solution using Eq.3
17:     else
18:       Update the solution using Eq.4
19:     end if
20:     If the solution is better, update the population.
21:     Update  $c$ 
22:   end for
23:   Step 5
24:   if  $Itr$  is not reached then
25:      $itr = itr + 1$ 
26:   end if
27: end while
28: Return  $g^*$ 

```

3.4 Exploration vs. Exploitation in Butterfly Optimization Algorithm

This section analyzes the BOA exploration and exploitation behaviour due to its significant effect on optimizing problems. The success of metaheuristics stands for their capabilities in achieving the best balance between exploration and exploitation. Such two terms are conflicting in their searching behaviour. In the explorations phase, the algorithm has a high ability to explore and navigate not-visited search space regions, whereas, in the exploitation phase, the algorithm's processes concentrate on searching deeply in the navigated search space regions. Typically, the algorithm searches randomly in the exploration phase and accumulatively in the exploitation phase.

In the context of BOA, r and p operators are in charge of determining the searching phase, whether exploration or exploitation, as discussed in Sect. 3.2. In the initial stage of the search, the distances between the decision variables' values are usually high due to the randomness of the initial population; thus, the exploration capability is high. In the

last stage of the search, the distances are lower due to the similar population situation; thus, the exploitation capability is high. Figure 10 shows an example of the distance reduction between the decision variables in five solutions from the initial iteration to the last one. Notably, during the first few iterations, the average distance between the decision variables is high; thus, the exploration is high. Subsequently, the average distance is reduced iteratively until the last iteration, where the distance is low.

Furthermore, the convergence behaviour is studied and evaluated for the same problem and solutions to show the BOA efficiency in optimizing optimization problems, as shown in Fig. 11. Note that the BOA reduced most of the objective function values significantly while maintaining the balance between exploration and exploitation.

3.5 Illustrative Example of Butterfly Optimization Algorithm

The Ackley 30-dimensional function is utilized to provide a complete picture of the convergence behaviour of BOA. This function can be represented as:

$$\begin{aligned} \min f(x) = & -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) \\ & - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e \end{aligned} \tag{6}$$

where $x_i \in [1, 30]$ and the best objective optima value is 20.6678. The population size is 5 as shown in Table 1. For each solution, the 1st, 2nd, 3th, and 30th dimensions were extracted in the 1st, 2nd, 10th, 20th, 30th, and 40th iterations with their objective function values. Table 1 elaborates that the solutions in the initial population are very scattered and far from each other. This shows that the BOA tends to explore many regions in the search space in the initial search. Later after iteration #30, the BOA exploits the search space by reducing the distance between the solutions (i.e., following the best solution) until the global optima is reached.

4 Recent Variants of Butterfly Optimization Algorithm

The BOA is a manageable optimization algorithm where it has a simple searching process with high-efficiency [21]. The BOA is primarily proposed and applied for continuous search spaces. Several studies have adjusted its processes to suit different search spaces. In addition, the BOA behaviour

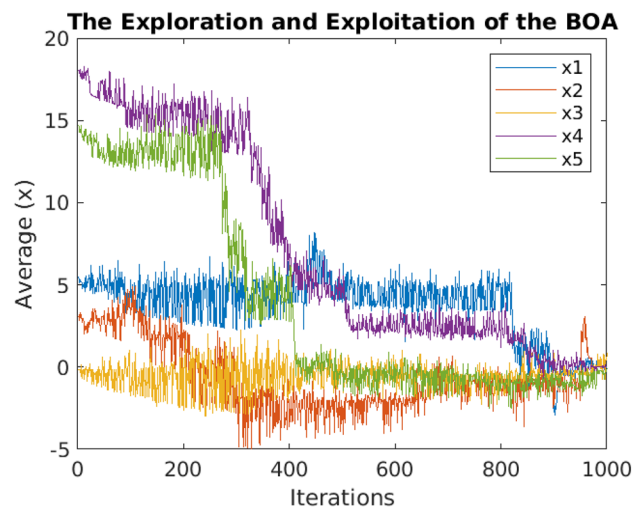


Fig. 10 The exploration and exploitation of the BOA

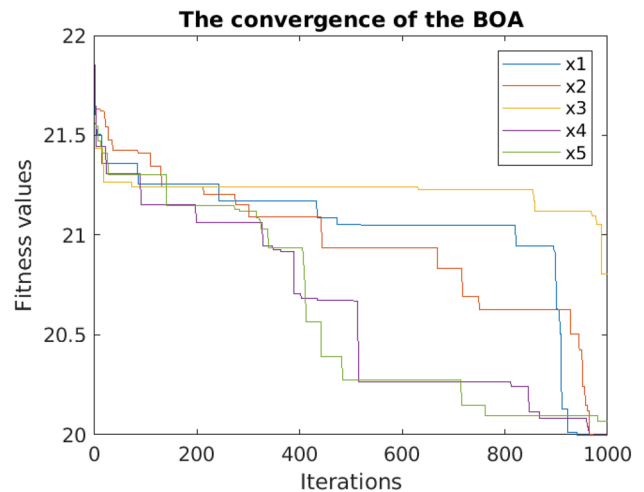


Fig. 11 The convergence of the BOA

is modified to improve its searching rules and obtain better solutions. The modifications are proposed in two variants, including improved and hybridized versions. In the following sections, several BOA variants, including original, improved, and hybridized, are discussed for different problems. The studies that are proposed in each variant are presented in Fig. 12.

4.1 Original Butterfly Optimization Algorithm

This section presents the adaptation of the original BOA to show its performance in addressing optimization problems and achieving the best of their results.

Fadaee et al. [42] investigated the capability of the original BOA by comparing its performance with the genetic algorithm (GA) to increase the accuracy of machine learning models, including adaptive neuro-fuzzy inference system, multiple linear regression, and artificial neural network. Several independent parameters were added to the machine learning models to increase prediction accuracy. The experimental results showed that the original machine learning models obtained similar results, whereas integrating the models with the BOA and GA could improve the results. The integration results proved the BOA essential performance compared with that of the GA.

The original BOA was adapted to find the maximum power point tracking of the photovoltaic (PV) systems under partial shading conditions [43]. The BOA was implemented for three different scenarios to investigate its performance using various situations. Also, the performance of BOA was compared with that of PSO, GWO, and the gravitational search algorithm to verify its efficiency. The results showed that the BOA performed better, where it outperformed all compared algorithms in terms of accuracy and tracking speed.

Abdul-Rashid and Alawode [44] adapted the BOA to optimize the designed Lead-Lad Controller parameters. The main aim of such an optimization is to enhance the stability of the control system. In the evaluation results, a numerical analysis was provided to evaluate the BOA efficiency in optimizing the objective function. In addition, the BOA's results were compared with that of the GA and differential evolution algorithm to investigate their performance. The BOA outperformed all compared algorithms in optimizing the control system's damping coefficient.

The BOA was used by Arora and Singh [45] to estimate the position of sensor nodes in wireless sensor networks (WSNs). The researchers described the BOA-based localization approach and summarized the findings by comparing the algorithm to others, such as the firefly algorithm and PSO, based on the computing time, localized nodes, and localization error. The computation results demonstrated that the proposed method is an efficient refinement method for node localization. In terms of accuracy and computing time, BOA beat the other algorithms tested.

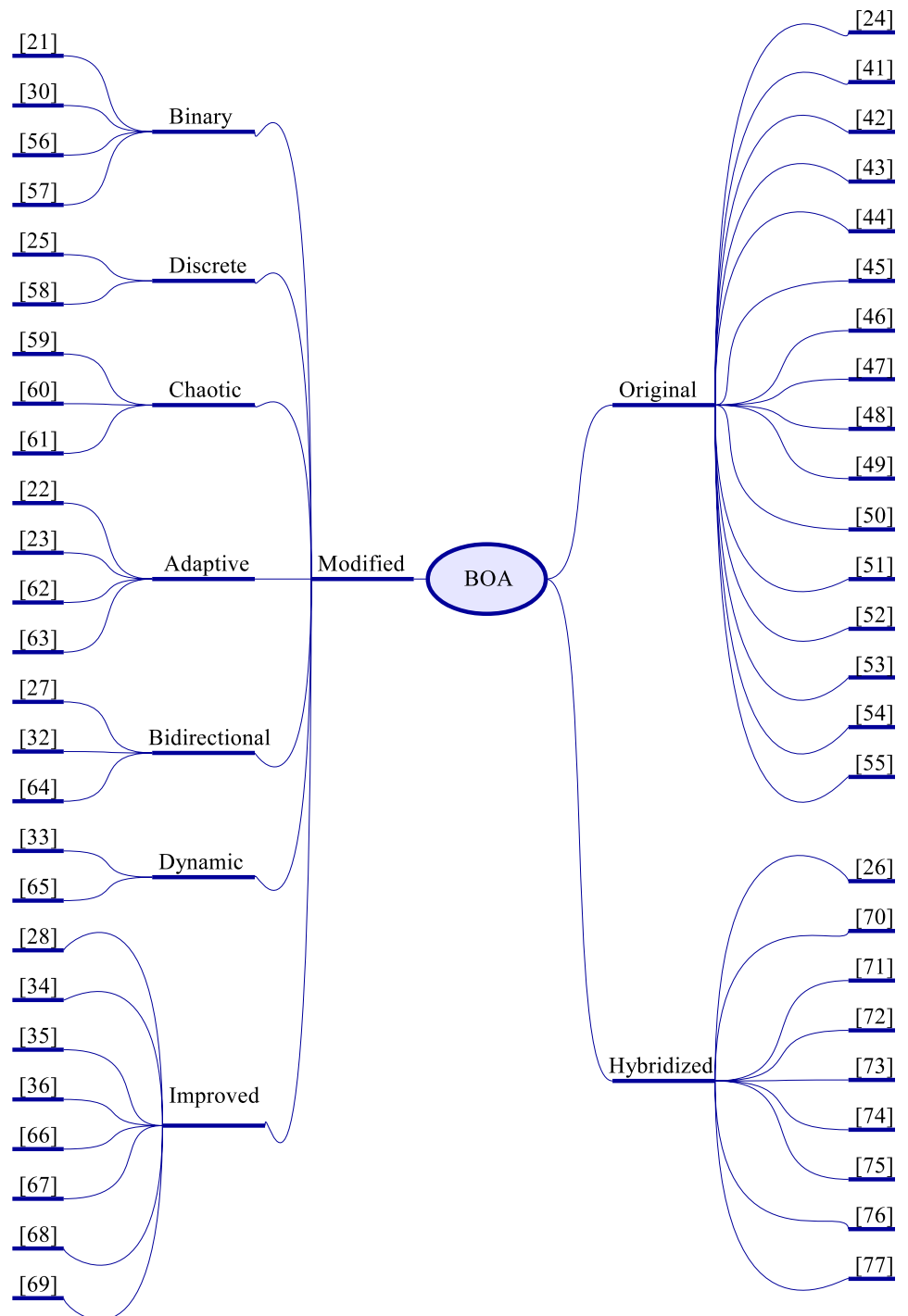
The BOA was adapted to optimize the network lifetime and overall energy consumption by selecting an optimal/near-optimal cluster head from a nodes' group in WSNs [25]. The optimal cluster head was selected considering several criteria, including the nodes' residual energy, distance to the base station, distance to the neighbours, node centrality,

Table 1 BOA results

Sol#	Initial population				Population after one iteration				Population after 10 iterations						
	Dim_1	Dim_2	Dim_3	Dim_30	f(x)	Dim_1	Dim_2	Dim_3	Dim_30	f(x)	Dim_1	Dim_2	Dim_3	Dim_30	f(x)
1	62.9447	-80.4919	-68.4774	-84.8067	21.6897	61.8210	-81.7335	-68.4356	-83.9485	21.6353	0.5369	-0.9969	-0.7460	-70.1018	21.2413
2	81.1584	-44.3004	94.1186	-52.0168	21.5792	80.1011	-43.6470	92.8373	-51.3449	21.5792	0.8246	-0.4805	100	-35.9860	21.4162
3	-74.6026	9.3763	91.4334	-75.3362	21.5480	-75.8805	8.1518	91.6619	-74.6324	21.5480	-0.9887	0.3187	100	-65.0665	21.4248
4	82.6752	91.5014	-2.9249	-75.3362	21.4704	81.5903	90.3007	-2.8865	-62.3889	21.4704	0.7357	0.7792	-0.0420	-56.8926	21.4087
5	26.4718	92.9777	60.0561	-75.3362	21.7334	27.6246	92.7750	58.9718	-51.0728	21.6772	0.0702	0.6820	0.5745	-39.2179	21.5374
Population after 20 iterations															
Sol#	Dim_1	Dim_2	Dim_3	Dim_30	f(x)	Dim_1	Dim_2	Dim_3	Dim_30	f(x)	Dim_1	Dim_2	Dim_3	Dim_30	f(x)
1	0.4680	-0.7648	-0.8868	-0.7221	21.2413	0.4729	-0.6514	-0.8609	-79.3701	21.2413	0.4842	-0.6626	-0.5611	-0.5852	21.1141
2	1.0000	-0.6970	0.6528	-0.2613	21.3488	0.7844	-0.6700	0.3712	-8.1758	21.2753	100	-100	-0.0821	-0.0013	20.6678
3	-0.8092	0.0731	0.8182	-0.5540	21.4187	-1.0000	0.5655	0.8795	-59.7101	21.3474	-100	-0.0678	100	-0.1009	20.9478
4	0.6477	0.5842	-0.0957	-0.4599	21.3339	0.8950	0.5589	-0.2047	-51.9939	21.3339	0.5559	0.4144	-0.1256	-0.5265	21.3240
5	0.0621	0.5474	0.3469	-0.2731	21.3948	-0.4317	0.7455	0.3928	-21.8815	21.3255	-0.6453	0.4955	0.4985	-0.0733	21.0968

Bold values indicate the best value

Fig. 12 The BOA variants



and node degree. In the experimental results, the BOA performance was evaluated and analyzed based on alive nodes, dead nodes, data packets, and energy consumption. The BOA showed a significant performance in optimizing the objectives.

Yıldız et al. [46] implemented the BOA to obtain a lighter vehicle design. Firstly, BOA optimized the coupling with a bolted rim problem. Secondly, the vehicle suspension arm

shape is optimized using BOA. The Kriging metamodeling method was used to obtain constraint function and objective equations in shape optimization. The BOA reduced the suspension arm's weight significantly. The findings demonstrated the BOA's potential to create better optimal automotive components.

Fathy [47] adapted the BOA to extract the global maximum power based on optimizing the reconfigure shaded

PV array under partial shadow conditions. The outcomes of BOA were compared to shade dispersion using GWO and novel structure-based arrangements, NS puzzle pattern, and series-parallel total-cross-tied. The proposed BOA achieved a maximum improvement compared to the series-parallel total-cross-tied system. Furthermore, the Wilcoxon test was performed for the proposed GWO and BOA results. Additionally, both approaches' statistical parameters were computed. The obtained findings revealed that the proposed BOA is suitable for optimally reconfiguring the PV array under PSCs.

Wen and Cao [48] introduced a novel intelligent model that combines prediction of driving forces and investigation. The BOA was used to optimize one classifier parameter (i.e., least-square support vector machine) and predict residential CO₂ emissions. By comparing prediction accuracy with existing models, the final experiments demonstrated the new model's superior performance.

A new classification approach was developed by Jalali et al. [49] to combine the BOA and artificial neural networks (ANN). As a new training approach, the BOA was used to optimize the biases and weights of ANN. Based on several evaluation measures and two well-known datasets, the proposed classification approach was compared with other state-of-the-art methods. The experimental findings demonstrated that the proposed method is superior to the other approaches.

The BOA was adapted to optimize the control horizon and prediction horizon of a new approach-based model predictive control and motion cueing algorithm by Qazani et al. [50]. The primary purpose of adapting the BOA is to reduce the tuning process's time of the proposed approach. In the experimental results, the BOA showed a significant performance in optimizing the proposed approach objectives and achieving the best results.

Parambil et al. [51] adapted the BOA to solve the economic load dispatch problem. The authors used the advantages of BOA in solving this kind of problem. To evaluate performance, the BOA was compared with GA and PSO. In terms of lowering fuel costs and increasing efficiency, the BOA strategy outperformed GA. BOA has a faster run time than PSO, which is an advantage in practical applications.

The BOA was utilized by Tamilarasan et al. [52] to optimize the parameters of the roller burnishing process. Three process parameters were considered: burnishing depth to achieve a higher surface hardness value, burnishing feed, and burnishing speed. Single-objective optimization was used to evaluate the BOA solutions to determine the optimal burnishing parameters. The percentage error for surface hardness is $\pm 2\%$, according to a comparison of experimental and optimization results.

The BOA was used by Bagchi et al. [53] to propose an Amended Adaptive Equalizer. The proposed method was

investigated on different Adaptive Equalizers based on evaluating the Bit Error Rate and the convergence rate. To evaluate the proposed approach, the constant modulus algorithm and least mean square algorithm was adapted and compared with the proposed approach. The proposed approach using the BOA showed better performance than all compared approaches.

The multi-layer radar absorbing material problem was addressed utilizing the BOA by Toktas and Ustun [54]. The BOA was used due to its high performance in optimizing this optimization problem. The problem was designed as a multi-objective optimization problem to optimize all objectives simultaneously, including total reflection and total thickness. The Pareto optimality approach was used to handle such a multi-objective problem. The experimental results proved the significant performance of the proposed method in optimizing the problem compared with other state-of-the-art methods.

ALTIRAİKİ and TEZEL [55] used the BOA to solve one of the critical challenges in the 5G network called environmental pollution. The authors utilized the BOA to maximize the average signal-to-noise index of the network and user-related vectors for service in a single cell to improve the communication power and output in the 5G networks. The researchers found the other methods, such as SPRS+WGC-PD, WGC-PD, SPRS, and Random, less efficient than the proposed BOA. In addition, each user's bit rate was higher than with previous methods.

The BOA was utilized by Mahboob and Moghaddam [56] to find the best features of the intrusion detection system. The NSL-KDD dataset was used in the evaluation step to investigate the system's efficiency with the BOA in detecting four types of attacks, including DoS, Probe, R2L, and U2R. The results proved the robust performance of the BOA in addressing the problem with high accuracy. In addition, the BOA performance was compared with that of state-of-the-art methods. The BOA showed significant improvement compared with other methods.

4.2 Modified Versions of Butterfly Optimization Algorithm

The BOA, like other algorithms, has drawbacks in the process of obtaining an optimal solution which may lead to not obtaining the global optima. Therefore, several studies have been proposed to modify the BOA searching processes to overcome low convergence speed, local optima stagnation, and imbalance between exploitation and exploration. In addition, the BOA is modified to deal with different kinds of search spaces, where it was initially proposed for only continuous search spaces.

In this section, the modified BOA versions are reviewed based on the modification types, where each type of modified BOA is presented in a separate section.

4.2.1 Binary Butterfly Optimization Algorithm

As discussed previously, the BOA was initially proposed to address continuous optimization problems. A new version of BOA was proposed by modifying its searching features to be suitable for binary optimization problems [31].

Zhang et al. [31] proposed an improved version of binary BOA to address one of the most popular binary optimization problems, known as the feature selection (FS) problem. The primary objectives of the proposed method are to improve dimension reduction, classification accuracy, and reliability in the FS problem. A new differential evolution initialization strategy was applied to minimize the binary BOA randomness. In addition, a new parameter is added to enhance the balance between exploration and exploitation of the binary BOA. Subsequently, the behaviour of BOA is modified to improve the quality of solutions by employing an evolution population dynamics mechanism. In the evaluation results, the proposed method was tested for twenty datasets and compared with seven algorithms to investigate its performance. The results proved the robust performance of the proposed method in classification accuracy maximization and the number of features selected minimization.

Two proposed binary variations of BOA were used by Arora and Anand [57] to find the best feature combination that reduces the number of features involved while maximizing classification accuracy. In these modifications, the basic BOA was used. Four high-performance optimization algorithms and five state-of-art methods were compared with the proposed binary algorithms. Various evaluation indicators were used to correctly evaluate these algorithms' performance and compare them over 21 datasets. The experiments confirmed that BOA is efficient in searching for the search space in the most informative classification attributes and that the proposed techniques are efficient compared to previous wrapper-based algorithms.

Sadeghian et al. [22] proposed a new version of binary BOA with Information Gain to maximize the mean of mutual information between class labels and sample features as well as the accuracy of classification. The proposed method was utilized in a three-phase method, called ensemble information theory-based bBOA, which aimed to reduce the number of selected features. Using the minimal redundancy-maximum new classification information FS, 80% of redundant and irrelevant features were eliminated in the first phase. The best feature subset was selected using the proposed method in the second phase. Finally, the final features subset was selected using a similarity-based ranking method. Six

standard datasets were used to conduct the experiments. The results showed that the proposed strategy effectively selects the best optimal features subset with the lowest number of features and improves classification accuracy in most cases.

Shahbandegan and Naderi [58] modified the BOA searching behaviour to deal with a binary multidimensional knapsack optimization problem. The binary BOA was proposed in six versions using three S-shaped and three V-shaped transfer functions to determine the most effective version. In addition, an initial population generator and a repair operator were proposed to enhance the method's performance. In the evaluation step, 11 medium-scale and large-scale benchmark problems were used to investigate the proposed method's performance. Also, a comparison study was conducted with other competitive methods.

4.2.2 Discrete Butterfly Optimization Algorithm

The BOA was also modified to suit the discrete search spaces of different optimization problems.

Hosseinzadeh et al. [26] proposed a discrete version of BOA to deal with the discrete values in the search space of the workflow scheduling problem. The primary purpose of addressing such a problem is to find the optimal task execution order in the scientific workflows. The Levy flight method was combined with the proposed discrete BOA for further enhancement in addressing the problem by improving the BOA local search ability and convergence speed. The evaluation results proved the significant performance of the proposed method in outperforming the compared methods and achieving the results.

Another discrete version of BOA was proposed by Lansky et al. [59] to address a workflow scheduling problem and achieve the best schedule optimally. In addition, the Levy flight was utilized with the BOA to improve the BOA convergence speed and prevent the local optima problem. In the experimental evaluation step, the proposed method was examined using well-known scientific workflow scheduling problems with different dimensions. The results proved the proposed method's robust performance, where it outperformed all compared methods in optimizing the objectives.

4.2.3 Chaotic Butterfly Optimization Algorithm

New BOA versions were proposed by integrating the BOA searching behaviour with the chaotic maps method to improve its performance and searching speed. The chaotic maps method is a mathematical method that deals with nonlinear dynamical systems and plays an essential role in improving optimization methods' performance [32].

A new modified version of BOA based on the chaotic maps method was proposed by Awad et al. [60] to increase its diversity and avoid stagnation in local optima. The new

method was mainly proposed to address the optimization problems with high complexity and dimensionality, such as the FS optimization problem. The proposed chaotic BOA's performance was investigated using 16 benchmark datasets and evaluated by comparing it with six optimization algorithms. The chaotic BOA outperformed all compared algorithms in optimizing the objectives.

The BOA was modified by Arora and Singh [61] by introducing the chaotic maps method into its searching behaviour to improve its performance in terms of convergence speed and local optima avoidance. To improve the BOA's performance, ten chaotic maps are used. On engineering design problems and multimodal and unimodal benchmark test functions. According to the findings, chaotic maps can considerably improve BOA's performance.

A modified version of the BOA based on the chaotic map method was proposed by Zhi et al. [62] for improving the heat recovery system's performance. The primary objective of modifying the BOA is to increase the BOA convergence speed. Based on a multi-criteria assessment, the authors proposed an optimal heat recovery system for power generation, heating, and combined cooling, a gas compressor, a humidifier, a small absorption chiller, and a 5 kW proton exchange membrane stack. In the experiment results, the modified BOA proved its efficiency in improving the system and achieved the best performance. In addition, the BOA reduced the annual operation cost.

4.2.4 Adaptive Butterfly Optimization Algorithm

The adaptive parameter tuning concept in the optimization context refers to automatically tuning parameters during the search process by the algorithm. This approach could help the optimization method to find the values for the parameters and find the best solutions.

Long et al. [23] proposed a new improved version of the BOA, called Enhanced adaptive BOA, to improve the searching behaviour of the original BOA and achieve a better balance between exploration and exploitation by adding a new position search equation. The proposed method was evaluated to achieve accurate, reliable, and quick parameter identification for PV models. In the experimental results, the performance and efficiency of the proposed method were tested using 12 classical benchmark test problems and three benchmark test PV models, including single diode, double diode, and PV module models. The proposed method proved its high performance in obtaining the results, where it outperformed most of the compared methods in terms of reliability and accuracy.

An adaptive BOA was proposed by Hu et al. [24] to address the early search blindness and the relatively poor adaptability of the sensory modality. The performance of the adaptive BOA was evaluated using 14-benchmark test

functions. The results showed that the proposed method obtained accuracy and stability competitively. In addition, the adaptive BOA was tested using the support-vector-machine prediction model based on an engineering problem. Also, the proposed method's performance was compared with other optimization algorithms. The adaptive BOA performed better than all compared algorithms.

Ustun [63] proposed an adaptive BOA to enhance the BOA searching capability by emphasizing local and global search and thus optimize the balance between exploration and exploitation. The proposed adaptive BOA was tested using CEC-2017 benchmark functions. In addition, the adaptive BOA was tested using three constrained engineering problems, including welded beam design, spring design, and gear train design. In the evaluation results, the proposed method was evaluated by comparing its performance with that of the other four methods. The adaptive BOA outperformed all compared methods in achieving their optimal solutions.

An adaptive BOA was proposed by Singh and Anand [64], which utilized a phenomenon of adjusting the sensory modality of BOA throughout the optimization process to produce better results than standard BOA. Seventeen common benchmark functions were investigated with the proposed method. The new method's performance was compared with existing standard optimization algorithms, such as BOA, FA, and artificial bee colony (ABC) algorithm. The findings show that the proposed method with enhanced factor calculation procedure outperformed the competition to achieve the global optimal solution and convergence rate efficiently.

4.2.5 Bidirectional Butterfly Optimization Algorithm

The bidirectional modification could improve the algorithms' searching performance by allowing the parameters to perform the local search in forward and backward directions [65]. Therefore, this type of improvement attracts the scholar's attention to modify optimization algorithms, such as BOA, in different domains.

Sharma et al. [33] integrated the bidirectional approach into the original BOA to improve the convergence rate and avoid stagnation in local optima. The proposed method was tested using three real-life problems and seven unimodal benchmark datasets. The simulation results demonstrated the improved version of BOA competency in optimizing the objectives. In addition, the proposed method outperformed most of the compared state-of-the-art methods.

A new modified variant of BOA was introduced by Sharma [65] based on a bidirectional searching behaviour. The primary purpose of the introduced method is to address the BOA drawbacks in emphasizing exploitation, where the BOA is usually stagnant in local optima. In addition, the

greedy selection method was applied in selecting the search direction. The proposed method was tested using CEC2006 and CEC2014 benchmark problems. The results showed the significant performance of the algorithm's improvements, where it almost outperformed all compared methods.

The bidirectional BOA was proposed by Elangovan and Subedha [28] for sentiment analysis based on a kernel extreme learning machine. The kernel extreme learning machine method was utilized as a classifier to classify the samples of positive or negative user reviews. In addition, the proposed bidirectional BOA was used to set up the parameters to improve the classifier performance. To evaluate the outcome of the proposed method, substantial testing was carried out on a benchmark dataset, with the outcomes examined from many perspectives. The proposed method showed encouraging results in addressing the problem and optimizing the objective.

4.2.6 Butterfly Optimization Algorithm with Dynamic Parameters

As most of the optimization algorithm's parts can be modified and improved, the behaviour of its parameters can be changed in attempting to enhance results. The parameters of the BOA can be modified to operate dynamically to move flexibly in search spaces and find better solutions.

Dynamic BOA was proposed by Tubishat et al. [34] to address the FS problem. In this paper, the invention of a local search based on mutation was used to improve the diversity of BOA solutions and avoid local optima problems. Twenty benchmark datasets were used to illustrate the efficiency and quality of the proposed method. The method and its competitor algorithms are notified of convergence curves, statistical results, the number of selected functions, fitness values, and classification accuracy. The results showed that DBOA outperformed most performance metrics significantly with comparing algorithms.

Tan et al. [66] proposed an improved BOA for data training of one of the unsupervised machine learning approaches, called wavelet neural networks. The proposed improved BOA was constructed based on combining a local search equation and a dynamic switch probability with BOA parameters to enhance its performance in achieving optimal/near-optimal results. The proposed method's performance was evaluated based on experimental comparisons with three robust algorithms, including the original BOA, PSO, and momentum backpropagation algorithm. Also, the compared algorithms' efficiency was evaluated using statistical analyses. The obtained results proved the high performance of the proposed method in terms of accuracy, robustness, and convergence.

4.2.7 Improved Versions of Butterfly Optimization Algorithm

Several other improved versions of BOA were proposed to enhance its search performance.

A modified version of BOA was proposed by Arora et al. [35] to address mechanical design optimization problems. The improvement emphasized a more intensive exploitation phase. The proposed method provided the solutions with a better ability to improve. The proposed algorithm's performance was validated using three engineering design problems and fifteen benchmark test functions with different decision variables, constraints, and objectives. The obtained results were compared to those published in the literature. The results showed that the proposed method outperformed other known optimization methods in terms of performance.

In order to overcome low accuracy and to stack in local optima problems of the basic BOA, an improved BOA was proposed by Arora and Singh [36] that utilized a variable sensory modality parameter strategy. The proposed algorithm's performance was evaluated using a variety of benchmark functions. The proposed method was evaluated against three other metaheuristic algorithms: PSO, FA, and ABC, as well as the standard BOA. The results obtained demonstrated that the proposed method is superior to, or at least equivalent to, the existing algorithm in terms of convergence rate and final solution quality.

Arora and Anand [67] utilized learning automata in the BOA to increase the potential of the BOA, which focuses on exploration in the early stages and exploitation in the later stages of optimization. To ensure the right balance between the local and global search, a learning automatic takes the function of configuring the butterfly's behaviour. The proposed algorithm was evaluated on three classic engineering design problems with various real-world applications and seventeen benchmark test functions to validate its effectiveness. The findings show that the learning automata enhanced the global convergence to the true global optimum while maintaining the basic BOA's fundamental feature. The findings also demonstrated that including learning automata into BOA considerably improved BOA's performance in avoiding local optimum trapping and achieving true global optimums.

Li et al. [29] used the cross-entropy approach to enhance the original BOA. Based on a co-evolution strategy, the modification strategy found an appropriate balance between exploitation and exploration to avoid stacking into an optimum local while improving global search efficiency. The proposed method was tested on three classic engineering design problems and 19 well-known benchmark test functions. The findings of the engineering problems demonstrated that the BOA fits to solve complex problems with unknown search space and constraints. Besides, test function results showed that the proposed algorithm could produce

extremely competitive results concerning enhancing the convergence rate, optimal local avoidance, and global search.

Fan et al. [37] modified the BOA searching behaviour to address its drawbacks, including parameter complexities and poor exploration, by improving the iteration capability and the searching accuracy. The method was proposed with a simple structure, fewer parameters, and strong precision. The proposed method was tested using various optimization problems, including low-high dimensional optimization functions and constrained problems. To show the methods' efficiency, its achieved results were compared with well-known optimization algorithms. The results proved the high performance of the proposed method in achieving the objectives compared with the other algorithms.

Shams et al. [68] proposed a new modified BOA for partial shading patterns, solar intensity, uniform shading, and load variation conditions with fast convergence speed. The BOA was modified using one dynamic parameter to reduce the algorithm's complexity. In addition, a new skipping method was used with the BOA to reduce the high-dimension of the problem search space. In the evaluation results, the proposed method showed high achievements in tracking time, steady-state efficiency, and response to load variation.

Chen [69] used the recurrent neural network model to address one of the motivating research problems, known as the emotion recognition problem. The data in the used model was trained using an improved BOA version due to the BOA's high performance in handling such training data. The proposed improved BOA was constructed to handle the BOA drawbacks and enhance its searching capability. The experimental evaluation showed encouraging results, where the proposed method addresses the problem with a maximum accuracy rate.

A modified version of the BOA based on gbest-guided search strategy and pinhole-imaging-based learning was proposed by Long et al. [70]. The gbest-guided search strategy was embedded in the update position equation of BOA, and then the pinhole-imaging-based learning was utilized to avoid premature convergence and to search for unknown regions effectively. The efficiency of the proposed method was investigated further using 60 complex optimization tasks from CEC 2014 and CEC 2017 and 23 classic problems. On benchmark test functions, the comparative findings showed that the proposed method outperformed the majority of the compared algorithms. Finally, in a real-world wind turbine, the proposed method was used to tackle fault diagnosis and FS problems. In terms of classification accuracy, the results demonstrated that the proposed method outperformed other competitors.

The percentage of modified BOA versions is shown in Fig. 13, where the improved BOA has the highest percentage.

4.3 Hybridized Versions of Butterfly Optimization Algorithm

The BOA searching performance can be improved using the hybridization approach. The BOA parameters and searching behaviour can be combined with components of another method(s) to be modified and enhanced for solutions' quality improvements. Several studies are proposed to hybridize the BOA's components with other methods to enhance solutions' quality.

A new hybrid optimization algorithm was proposed by Arora and Singh [71], which combined the basic BOA with the basic ABC algorithm. To establish a balance between exploration and exploitation, the proposed method used the strengths of both algorithms. Ten benchmark functions with a wide variety of dimensions and difficulty were used to evaluate the proposed algorithm. The simulation results showed that the proposed method outperformed the basic BOA and ABC in terms of accuracy and convergence speed in finding optimal solutions.

Sharma et al. [72] enhanced the BOA performance by hybridizing its searching behaviour with another optimizing algorithm, called symbiotic organisms search algorithm. The primary aim of proposing such hybridization was to improve the BOA searching behaviour by emphasizing both exploration and exploitation and optimizing the balance between them. The proposed method was evaluated using twenty-five benchmark functions. The proposed method's performance was tested by comparing its results with well-known state-of-the-art methods. In addition, statistical tests based on the Friedman rank test and t-test were provided. The proposed method showed a significant performance in optimizing the problem and achieving the objectives compared with the other methods.

The FS optimization problem was addressed by Dubey [73] by adapting and hybridizing the BOA components with a recent swarm-based optimization algorithm, called the lion algorithm. In addition, a deep learning method was used to solve the multi-disease prediction problem. In the evaluation results, different standard datasets were used to evaluate the proposed method and prove its performance compared with traditional and original methods.

Zhang et al. [74] proposed a new hybrid method based on BOA and PSO to address the BOA drawbacks, including low accuracy and slow convergence. In addition, a new approach was introduced to enhance the BOA's performance, including a cubic one-dimensional map and a nonlinear parameter control strategy. The proposed methods were evaluated using 26 benchmark functions. The comparison evaluation proved the high efficiency of the proposed hybrid method in addressing optimization problems, where it outperformed all compared methods.

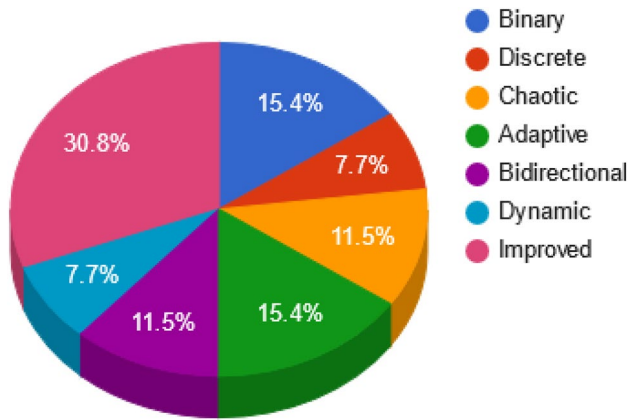


Fig. 13 The BOA modified versions

The BOA was hybridized with one of the recent popular swarm-based optimization algorithms, known as symbiosis organisms search, to emphasize the exploration and exploitation capabilities of the BOA and achieve the best balance between them [75]. The proposed method was tested using 26 classical benchmark functions for a global optimization approach. The proposed method was evaluated by comparing its performance with well-known optimization algorithms. The proposed method obtained the best results, where it outperformed all compared methods.

Sharma and Saha [76] hybridized the BOA with the sine-cosine algorithm to improve the optimization strategy of the BOA by developing a balance between local exploitation and global exploration abilities. The researchers used 25 benchmark functions to investigate the robustness and efficiency of the proposed method. In addition, comprehensive analysis with state-of-art algorithms was conducted. Besides, the researchers applied the proposed algorithm to two real-world problems to investigate the performance of the proposed method in an unknown search space. The results of these real-world problems were compared with various other algorithms, and in real-world optimization problems, the proposed method proved superior in most cases.

Zhou et al. [77] proposed a new hybrid method based on the components of the BOA and differential evolution algorithms. The primary purposes of proposing the method were to emphasize the exploitation and exploration capabilities and avoid stagnation in local optima. The proposed method was constructed in two main steps: the first step is an improvement based on the differential evolution parameters, and the second step introduces dynamic adaptive operators into the BOA. In the evaluation results, the proposed method was tested using different datasets. The proposed method showed high and robust performance in optimizing the objectives compared with the original versions.

Utama et al. [78] proposed a new hybrid version of BOA by combining the searching behaviour of the BOA with one

of the most popular local search algorithms, known as the tabu search algorithm. Such hybridization aimed to enhance the BOA exploitation capability and the solutions' quality. The proposed HBOA addressed the green vehicle routing problem and achieved its objectives optimally, including minimizing the distribution costs considering carbon costs, fuel costs, and vehicle use costs. In the evaluation results, the proposed method was compared with other methods to investigate its performance. The obtained results proved the proposed method's significant performance, where it outperformed all compared methods in optimizing the objectives.

The BOA searching parameters were hybridized with that of the multi-layer perceptron model to address one of the healthcare optimization problems [27]. In addition, the proposed method was combined with the oppositional-based learning model for performance enhancement. The primary aim of this study was to find the best classification for medical data, which was passed into three steps, including preprocessing, classification, and parameter tuning. In the evaluation results, the proposed method proved its performance in optimizing objectives.

5 Applications of Butterfly Optimization Algorithms

The BOA effectiveness and efficiency allow it to flourish when utilized for tackling different real-world applications. The algorithm tackled a wide range of problems: FS [22, 31, 34, 57, 60, 60], numerical optimization [33, 37, 64, 70, 71, 75], PV models [23, 47], early search blindness [24], energy consumption [25], image segmentation [72], scheduling [26, 59], multi disease prediction [73], medical data classification [27], optimal capacity of gas production [76], sentiment analysis [28], roller burnishing process parameters [52], pilot contamination in massive systems for 5G communication networks [55], optimum shape design [46], combined cooling heating and power generation system [62], household CO₂ emissions mitigation strategies [48], solving elliptic partial differential equations [66], reliability optimization problems [65], suspended sediment prediction [42], maximum power point tracking [43, 68], engineering problems [29, 61, 63, 67], green vehicle routing problem [78], high-dimensional optimization problems [74], model predictive control [50], crowd behaviour recognition [69], structural damage detection [77], anomaly-based intrusion detection [56], multidimensional knapsack problem [58], amended adaptive design [53], multilayer radar absorbing composite material [54], economic load dispatch problem [51], optimization of artificial neural networks [49], robustness evaluation of a control system [44], mechanical design optimization problems [35], node localization in wireless sensor networks [45], and global optimization [36, 55].

The list of the BOA versions used to tackle optimization problems is presented in Table 2. The table shows the problems and their types, the utilized methods and variants, the compared methods, and the published year.

The major applications addressed by the BOA are thoroughly briefly illustrated below.

FS means selecting the most favorable features that can be used to present the entire data. Solving the FS problem can help tackle a wide range of real-world applications, such as classifications and pattern recognition. BOA and modified BOA are used to tackle this problem [22, 31, 34, 57, 60, 60]. Numerical optimization is considered one of the mathematical techniques that are used to solve many engineering problems. The BOA and its variations are used to tackle these kinds of problems [33, 37, 64, 70, 71, 75]. The optimization of PV cells is vital to provide clean energy. Several researchers utilize BOA for developing PV models [23, 47].

The field of communication has several problems that are addressed by BOA, such as the efficient consumption of energy in WSNs [25], pilot contamination in massive MIMO systems for 5G communication networks [55], green vehicle routing problem [78], and node localization in wireless sensor networks [45].

BOA and its modifications undertake power and electrical system optimization problems. This encompasses economic emission dispatch [51], maximum power point tracking [43, 68], and combined cooling heating and power generation system [62].

Scheduling and planning tasks are considered NP combinatorial problems that try assigning a set of resources while adhering to a set of constraints to accomplish a predefined objective. Many research works targeted toward resolving such problems, such as data placement and scheduling in edge computing environments [26] and scientific workflow scheduling in mobile edge computing [59].

Design optimization is a famous engineering design problem that formulates such a problem mathematically, intending to achieve an optimal design from different possible alternatives. The BOA and its variations is used to tackle these problems such as optimum shape design [46], amended adaptive design [53], and mechanical design optimization problems [35].

6 Critical Analysis of BOA Theory

This section provides the analysis of BOA. The BOA's versions and applications are summarized in the previous sections. BOA has attracted much attention since it can solve a wide range of optimization problems efficiently. Such high performance is because of the BOA's features, including few control parameters, simplicity in construction, efficacy in

tackling real-time problems, and ease of implementation. However, the BOA has a few limitations and disadvantages despite all of these advantages.

The primary disadvantage of the BOA is associated with the NFL theorem of optimization [30, 79–81] which is also a problem facing other optimization algorithms. According to the NFL, no superior optimization algorithm can outperform all other optimization algorithms for all varieties of optimization problems or perhaps all cases of the same optimization problem. Thus, the convergence of the BOA is significantly related to the nature of the problem search space. Therefore, it is necessary to hybridize or modify the BOA, particularly in an unknown search space, to address the optimization problem.

The second disadvantage is related to the problem domain. As noted previously, the original BOA was designed for optimization problems with continuous search spaces, unconstrained objective functions, and single objectives. However, for various fields and nature, such as multi and many objectives, dynamic, combinatorial, binary, and discrete, BOA must expand its applications [26, 31].

The third disadvantage of BOA is related to the behaviour of its population. BOA can expand the search to the best global solution during a search process and closely move into an extensive range and variety of search regions via its operators, which decreases the distance with the best solutions. The BOA searching behaviour is based on inherited values, where its operators do not pay much attention to each region in the search space. Such an issue reduces the exploitation capabilities when dealing with multi-modal landscape problems. Thus, the BOA should be combined with other local search-based algorithms to improve its exploitation capabilities. BOA's combination of local search algorithms allows it to stay focused on promising search space regions [78].

Like the other optimization algorithms, the performance of the BOA is closely associated with the level of dimensionality (i.e., the size of the search space). As the number of problem explanatory variables, the algorithm's performance degrades. Therefore, dimensionality reduction is appropriate for specific problems to produce more efficient outcomes. Accordingly, other techniques for scalability and transformation should be considered [27].

Another recurrent drawback is BOA's reliance on stochastic operators that are essentially pseudo-random. The pseudo-randomness has an artificial cyclic aspect, which implies that some patterns will repeat themselves in the long term, and the search pattern will adapt a cyclic behaviour as well, potentially leading to an unending loop of search. However, BOA's behaviour is comparable to that of Simplex algorithms and Nelder-Mead. On the other hand, the BOA has stochastic components that limit its ability to cover multidimensional search spaces without cyclic behaviour [54].

Table 2 The BOA applications on optimization problems

Problem(S)	Proposed method	Variant	Type	Comparson with other methods	Year	References
Feature selection	EIT-bBOA	Hybrid	Binary	MR-MNCI, IG-bBOA, S-bBOA	2021	[22]
Numerical Optimization	BBOA	Modify	Continuous	BOA, BA, SSA, CSA, PSO, GA	2021	[33]
Global optimization and feature selection	PIL-BOA	Modify	Continuous + Binary	BSO, SCA, SSA, DA, WOA, ALO, GWO, PSO, GA	2021	[70]
Photovoltaic models	EABOA	Modify	Continuous	IGWO, HHO, BOA, WOA, OBSCA, mGWO, GWO	2021	[23]
Early search blindness	Adaptive BOA	Modify	Continuous	KCV-SVM model, back-propagation prediction model	2021	[24]
Energy consumption	BOA	Adaptive	Discrete	FLION, ALOC, BERA, CRHS, FUCHAR	2021	[25]
Image segmentation, Global optimization	MPBOA	Hybrid	Continuous		2021	[72]
Scheduling, Data Placement	DBOA	Modify	discrete		2021	[26]
Multi disease prediction	L-BOA	Hybrid	Binary		2021	[73]
Medical Data Classification	OBLBOA-MLP	Modify	Continuous	OlexGA, Logistic Regression, SVM, XGBoost, PSO, ACO, Decision Tree, Multi-Layer Perceptron, D-ACO, Fuzzy Neural Classifier, Adam-LR	2021	[27]
Scientific Workflow Scheduling	DBOA	Adaptive	discrete	DE, PSO, ACO	2021	[59]
capacity of gas production facilities, Gas transmission compressor design problem	BOSCA	Hybrid	Continuous	JAYA, PSO, DE, MBO, GA, FA, CS, ABC, SCA, BOA	2021	[76]
Sentiment Analysis	BBOA	Modify	Continuous		2021	[28]
Roller Burnishing Process Parameters	BOA	Adaptive	Continuous		2021	[52]
5G Communication Networks	BOA	Adaptive	Continuous	SPRS+WGC-PD, WGC-PD, SPRS, Random	2021	[55]
optimum shape design	BOA	Adaptive	Continuous		2020	[46]
CCHP driven	MBOA	Modify	Continuous	BOA, WCO, SSO, EPO, GSA	2020	[62]
Shaded photovoltaic array extracted power	BOA	Adaptive	Continuous	GWO, Shade dispersion with NS, NS puzzle pattern, SP_TCT	2020	[47]
Household CO2 emissions	EBOA-LSSVM	Adaptive	Continuous	SVM, CS-LSSVM, GA-LSSVM, PSO-LSSVM, BOA-LSSVM	2020	[48]
Feature Selection	DBOA	Modify	Binary	BOA, ALO, SCA, PSO, GA, GOA	2020	[34]
Feature Selection	bBOA	Modify	Binary	FSFOA, WOA-CM, HBBEPSO, WOASAT-2, BGOA_EPD_Tour, BGOA_M, PSO	2020	[31]
Elliptic partial differential equations	IBOA	Modify	Continuous	BOA, PSO, MBP	2020	[66]
Reliability optimization problems	BBOA	Modify	Continuous	BOA, DE, PSO, GA	2020	[65]
Numerical Optimization	SABOA	Modify	Continuous	SSA, SCA, LWOA, FPA, BOA	2020	[37]
Suspended Sediment Prediction	ANFIS-BOA	Adaptive	Continuous	MLR, ANN-BOA, ANN-GA, ANN-ML, ANFIS-GA, ANFIS-FCM	2020	[42]
Maximum Power Point Tracking	MBOA	Modify	Continuous	DE, FP, GWO, PSO	2020	[68]
Engineering problems	ABOA	Modify	Continuous	BOAL, MBOA, IBOA, BOA	2020	[63]
Green Vehicle Routing Problem	HBOA	Hybrid	discrete	GA, PSO, ACO, SA, TS, BOA	2020	[78]
Feature selection	CBOA	Modify	Binary	BBOA, PSO, GWO, GA, BSO, ALO	2020	[60]
High-Dimensional Optimization Problems	HPSOBOA	Hybrid	Continuous	IBOA, LBOA, PSOBOA, CABOA, BOA	2020	[74]
Numerical Optimization	hBOSOS	Hybrid	Continuous	JAYA, PSO, DE, SOS, BOA	2020	[75]

Table 2 (continued)

Problem(S)	Proposed method	Variant	Type	Comparison with other methods	Year	References
Model predictive control	BOA	Adaptive	Continuous	GA, TAE	2020	[50]
Feature selection	CBOA	Modify	Binary	BBOA, PSO, GWO, GA, BSO, ALO	2020	[60]
Crowd Behaviour Recognition	Enhanced-BOA	Modify	Continuous		2020	[69]
Structural damage detection	HBODEA	Hybrid	Continuous	DE, BOA	2020	[77]
An Anomaly-based Intrusion Detection	BOA	Adaptive	Binary		2020	[56]
Multidimensional Knapsack Problem	BBOA	Adaptive	Binary		2020	[58]
Amended Adaptive Design	BOA	Adaptive	Continuous		2020	[53]
Multilayer Radar Absorbing Composite Material	BOA	Adaptive	Continuous		2020	[54]
Economic Load Dispatch Problem	BOA	Adaptive	Continuous	PSO, GA	2020	[51]
Feature selection	S-bBOA	Adaptive	Binary	PSO, GWO, GA, BSO, ALO	2019	[57]
Artificial Neural Networks	BOA	Adaptive	Continuous	BOA, FPA, GOA, DE, PSO, GA	2019	[49]
Engineering Design Problems	BOA-CE	Modify	Continuous	BOA, SSA, CSA, GWO, BA, PSO, GA	2019	[29]
Maximum Power Point Tracking	BOA	Adaptive	Continuous	GWO, PSO-GSA	2019	[43]
Lead-Lad Controller	BOA	Adaptive	Continuous	DE, GA	2019	[44]
Mechanical design optimization problems	MBOA	Modify	Continuous	DE, HS, GA, ES, PSO, GSA, GWO	2018	[35]
Numerical Optimization	ABOA	Adaptive	Continuous	FA, ABC, BOA	2018	[64]
Engineering design problems	LABOA	Modify	Continuous	SaDE, FPA, CLPSO, BSO, BOA, BA, ABC	2018	[67]
Classical engineering problems	BOA with chaos	Modify	Continuous	NLDP, MIDCO, GA, MIDCP, SA, MVCP, GeneAS, SL, PSO, CS, CAPSO, CBOA	2017	[61]
Node Localization in Wireless Sensor Networks	BOA	Adaptive	Continuous	FA, PSO	2017	[45]
Numerical Optimization	Hybrid BOA-ABC	Hybrid	Continuous	BOA, ABC	2017	[71]
Global optimization	Improved BOA	Modify	Continuous	PSO, FA, ABC	2016	[36, 55]

Finally, one of the major BOA drawbacks is related to diversity control. Due to the loss of variety in the first course of execution, the BOA convergence behaviour tends to stagnate at local minima early. This is because the BOA's operators only select one solution (the best) per generation and ignore the other characteristics of the population's other solutions. Thus, the best solution will be affected by other solutions early. It will be attracted to the optimal solution while avoiding the drawbacks of the worst solution. In optimization, it is common knowledge that the search direction to the global optima is not always the strike to the optimal solution. As a result of ignoring most other solutions in the population, immaturity occurs, and loss of diversity. Several studies propose splitting the population into sub-populations and treating each sub-population with different BOA to limit diversity. It is also worth noting that parallel BOA can substantially speed up convergence without compromising solution quality [60].

7 Conclusion and Future Trends

The BOA has powerful and dynamic parameters that make it flexible and efficient to address various optimization problems in different domains. Accordingly, several studies have proposed using BOA to address optimization problems, particularly engineering optimization problems. However, BOA was initially proposed to address problems with continuous search spaces. In addition, although the BOA has robust parameters, it is still facing several problems that may lead not to obtaining optimal/near-optimal solutions in some cases. These drawbacks are low convergence speed, local optima stagnation, and an imbalance between exploitation and exploration. Therefore, several versions have been proposed to address these drawbacks and make the BOA handle different search spaces. These versions are classified into modified and hybridized classes. The modified class improved the BOA searching performance by modifying its components, whereas the hybrid class improved the

algorithm performance by combining it with a component from another algorithm.

In this paper, the BOA is studied and evaluated by reviewing recent studies from 2016 to 2022. Initially, general information regarding the optimization algorithms and the reasons behind the growth of BOA usage by the research community. Then, most of the studies that utilized the BOA in all its forms, including original, modified, and hybridized, are reviewed and summarized. For the modified class, the reviewed papers are categorized based on their modification approach, including binary BOA, discrete BOA, chaotic BOA, adaptive BOA, bidirectional BOA, dynamic BOA, and other modified versions. Each modified version was proposed to handle a drawback, such as the binary BOA and discrete BOA was proposed to change the BOA searching behaviour to deal with binary search spaces. In other versions, some approaches were added to the algorithm to improve its performance, such as chaotic.

This paper also provides the main applications addressed by the algorithm. In addition, a critical analysis is presented to show the BOA's advantages and disadvantages and its strong points.

Possible future directions can be considered to enhance the BOA's performance quality and achieve better optimal/near-optimal solutions in different domains as follows:

- *Selection mechanisms* The BOA focuses mainly on the best solution in the population in terms of global search. In this matter, several good solutions and directions that might improve the current solution are neglected. Accordingly, other solutions can be selected instead of the best solution using a selection method, such as tournament, proportional, and roulette wheel selection, for further improvement [82, 83].
- *Adaptive Parameters* In the BOA searching processes, three main parameters must be initialized before starting the searching processes, including c , a , and p . Some parameters are fixed, such as a and p , which may make the searching mechanism rigid. Therefore, modifying these parameters' nature can enhance the algorithm's performance in covering all search space regions [84].
- *BOA Modification* The BOA searching behavior can be improved to enhance the obtained solutions' qualities. Such improvement can be achieved by modifying or hybridizing the BOA's searching processes. These enhancement approaches were proposed in several studies in attempting to achieve the optimal solution to optimization problems. However, most of these studies obtained good and encouraging solutions but not the optimal ones. Therefore, an enhancement approach can be proposed for further solutions improvements, particularly in the exploitation phase, where the BOA has low exploitation capability in the initial searching stage, as shown in Sect. 3.4. Various local search optimization methods can be combined with the BOA to improve its local searchability, such as the Min-conflict algorithm [85].

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Fausto F, Reyna-Orta A, Cuevas E, Andrade ÁG, Perez-Cisneros M (2020) From ants to whales: metaheuristics for all tastes. *Artif Intell Rev* 53:753–810
2. Zitar RA, Al-Betar MA, Awadallah MA, Doush IA, Assaleh K (2021) An intensive and comprehensive overview of JAYA algorithm, its versions and applications. *Arch Comput Methods Eng* 2021:1–30
3. Eiben AE, Smith JE et al (2003) Introduction to evolutionary computing, vol 53. Springer, New York
4. Boussaïd I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. *Inf Sci* 237:82–117
5. Ignizio JP, Cavalier TM (1994) Linear programming. Prentice-Hall Inc, Hoboken
6. Conforti M, Cornuéjols G, Zambelli G et al (2014) Integer programming, vol 271. Springer, New York
7. Lawler EL, Wood DE (1966) Branch-and-bound methods: a survey. *Oper Res* 14:699–719
8. Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput Surv (CSUR)* 35:268–308
9. Hussain K, Salleh MNM, Cheng S, Shi Y (2019) Metaheuristic research: a comprehensive survey. *Artif Intell Rev* 52:2191–2233
10. Dokeroglu T, Sevinc E, Kucukyilmaz T, Cosar A (2019) A survey on new generation metaheuristic algorithms. *Comput Ind Eng* 137:106040
11. Meraihi Y, Ramdane-Cherif A, Acheli D, Mahseur M (2020) Dragonfly algorithm: a comprehensive review and applications. *Neural Comput Appl* 32:16625
12. Faris H, Aljarah I, Al-Betar MA, Mirjalili S (2018) Grey wolf optimizer: a review of recent variants and applications. *Neural Comput Appl* 30:413–435
13. Hussain K, Salleh MNM, Cheng S, Shi Y (2019) On the exploration and exploitation in popular swarm-based metaheuristic algorithms. *Neural Comput Appl* 31:7665–7683
14. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. *IEEE Comput Intell Mag* 1:28–39
15. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. *Commun Nonlinear Sci Numer Simul* 17:4831–4845
16. Yang X-S, Deb S (2009) Cuckoo search via lévy flights, in: world congress on nature & biologically inspired computing (NaBIC). *IEEE* 2009:210–214
17. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Comput Appl* 27:495–513
18. He S, Wu QH, Saunders J (2009) Group search optimizer: an optimization algorithm inspired by animal searching behavior. *IEEE Trans Evol Comput* 13:973–990

19. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
20. Al-Betar MA, Alyasseri ZAA, Awadallah MA, Doush IA (2021) Coronavirus herd immunity optimizer (chio). *Neural Comput Appl* 33:5011–5042
21. Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. *Soft Comput* 23:715–734
22. Sadeghian Z, Akbari E, Nematzadeh H (2021) A hybrid feature selection method based on information theory and binary butterfly optimization algorithm. *Eng Appl Artif Intell* 97:104079
23. Long W, Wu T, Xu M, Tang M, Cai S (2021) Parameters identification of photovoltaic models by using an enhanced adaptive butterfly optimization algorithm. *Energy* 229:120750
24. Hu K, Jiang H, Ji C-G, Pan Z (2021) A modified butterfly optimization algorithm: an adaptive algorithm for global optimization and the support vector machine. *Expert Syst* 38:e12642
25. Maheshwari P, Sharma AK, Verma K (2021) Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization. *Ad Hoc Netw* 110:102317
26. Hosseinzadeh M, Masdari M, Rahmani AM, Mohammadi M, Aldalwie AHM, Majeed MK, Karim SHT (2021) Improved butterfly optimization algorithm for data placement and scheduling in edge computing environments. *J Grid Comput* 19:1–27
27. PYogananda DL, Giri AA (2021) Oppositional butterfly optimization algorithm with multilayer perceptron for medical data classification. *Turki J Comput Math Educ* 12:2721–2731
28. Elangovan D, Subedha V (2021) Sentiment analysis and classification model using bidirectional butterfly optimization algorithm with kernel extreme learning machine. *J Comput Theor Nanosci* 18:664–673
29. Li G, Shuang F, Zhao P, Le C (2019) An improved butterfly optimization algorithm for engineering design problems using the cross-entropy method. *Symmetry* 11:1049
30. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1:67–82
31. Zhang B, Yang X, Hu B, Liu Z, Li Z (2020) Oebboa: a novel improved binary butterfly optimization approaches with various strategies for feature selection. *IEEE Access* 8:67799–67812
32. Shi P, Karimi HR, Su X, Yang R, Zhao Y (2014) Mathematical modeling, analysis, and advanced control of complex dynamical systems
33. Sharma TK, Sahoo AK, Goyal P (2021) Bidirectional butterfly optimization algorithm and engineering applications. *Mater Today Proc* 34:736–741
34. Tubishat M, Alswaiti M, Mirjalili S, Al-Garadi MA, Rana TA et al (2020) Dynamic butterfly optimization algorithm for feature selection. *IEEE Access* 8:194303–194314
35. Arora S, Singh S, Yetilmezsoy K (2018) A modified butterfly optimization algorithm for mechanical design optimization problems. *J Braz Soc Mech Sci Eng* 40:1–17
36. Arora S, Singh S (2016) An improved butterfly optimization algorithm for global optimization. *Adv Sci Eng Med* 8:711–717
37. Fan Y, Shao J, Sun G, Shao X (2020) A self-adaption butterfly optimization algorithm for numerical optimization problems. *IEEE Access* 8:88026–88041
38. Saccheri I, Kuussaari M, Kankare M, Vikman P, Fortelius W, Hanski I (1998) Inbreeding and extinction in a butterfly metapopulation. *Nature* 392:491–494
39. Blair RB, Launer AE (1997) Butterfly diversity and human land use: Species assemblages along an urban gradient. *Biol Conserv* 80:113–125
40. Pollard E, Yates TJ (1994) Monitoring butterflies for ecology and conservation: the British butterfly monitoring scheme. Springer, New York
41. BOA, butterfly optimization algorithm matlab (2021) <https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/b4a529ac-c709-4752-8ae1-1d172b8968fc/67a434dc-8224-4f4e-a835-bc92c4630a73/previews/BOA.m/index.html>. Accessed 16 June 2021
42. Fadaee M, Mahdavi-Meymand A, Zounemat-Kermani M (2020) Suspended sediment prediction using integrative soft computing models: on the analogy between the butterfly optimization and genetic algorithms. *Geocarto Int* 2020:1–17
43. Aygül K, Cikan M, Demirdelen T, Tumay M (2019) Butterfly optimization algorithm based maximum power point tracking of photovoltaic systems under partial shading condition. *Energy Sources Part A* 2019:1–19
44. Abdul-Rashid R, Alawode BO (2019) Robustness evaluation of the butterfly optimization algorithm on a control system. <http://arxiv.org/abs/1912.00185>
45. Arora S, Singh S (2017) Node localization in wireless sensor networks using butterfly optimization algorithm. *Arab J Sci Eng* 42:3325–3335
46. Yıldız BS, Yıldız AR, Albak Eİ, Abderazek H, Sait SM, Bureerat S (2020) Butterfly optimization algorithm for optimum shape design of automobile suspension components. *Mater Test* 62:365–370
47. Fathy A (2020) Butterfly optimization algorithm based methodology for enhancing the shaded photovoltaic array extracted power via reconfiguration process. *Energy Convers Manag* 220:113115
48. Wen L, Cao Y (2020) A hybrid intelligent predicting model for exploring household co2 emissions mitigation strategies derived from butterfly optimization algorithm. *Sci Total Environ* 727:138572
49. Jalali SMJ, Ahmadian S, Kebria PM, Khosravi A, Lim CP, Nahavandi S (2019) Evolving artificial neural networks using butterfly optimization algorithm for data classification. In: *International conference on neural information processing*. Springer, pp 596–607
50. Qazani MRC, Jalali SMJ, Asadi H, Nahavandi S (2020) Optimising control and prediction horizons of a model predictive control-based motion cueing algorithm using butterfly optimization algorithm. In: *IEEE congress on evolutionary computation (CEC)*. IEEE 2020, pp 1–8
51. Parambil RK et al (2020) Economic load dispatch problem using butterfly optimization algorithm. *Eur J Mol Clin Med* 7:2773–2778
52. Tamilarasan A, Arumugam S, Rajamani D, Vijayabhaskar S, Balakumar R, Reddy BT (2021) Butterfly optimization algorithm for optimization of roller burnishing process parameters. *Advances in materials and manufacturing engineering*. Springer, New York, pp 397–403
53. Bagchi M, Rej D, Sarkar S, Banerjee S (2020) Design of amended adaptive equalizer using butterfly optimization algorithm: extension to lte system. In *2020 3rd International conference on intelligent sustainable systems (ICISS)*. IEEE, pp 34–40
54. Toktas A, Ustun D (2020) Dual-objective design of multilayer radar absorbing composite material using butterfly optimization algorithm. In: *IEEE XXVth international seminar/workshop direct and inverse problems of electromagnetic and acoustic wave theory (DIPED)*. IEEE 2020, pp 77–81
55. ALTİRAİKİ S, TEZEL NS (2021) A new approach to pilot contamination in massive mimo systems for 5g communication networks with butterfly optimization algorithm. *Politeknik Dergisi*
56. Mahboob AS, Moghaddam MRO (2020) An anomaly-based intrusion detection system using butterfly optimization algorithm. In: *2020 6th Iranian conference on signal processing and intelligent systems (ICSPIS)*. IEEE, pp 1–6
57. Arora S, Anand P (2019) Binary butterfly optimization approaches for feature selection. *Expert Syst Appl* 116:147–160

58. Shahbandegan A, Naderi M (2020) A binary butterfly optimization algorithm for the multidimensional knapsack problem. In: 2020 6th Iranian conference on signal processing and intelligent systems (ICSPIS). IEEE, pp 1–5
59. Lansky J, Mohammadi M, Mohammed AH, Karim SHT, Rashidi S, Rahmani AM, Hosseinzadeh M (2021) Scientific workflow scheduling in mobile edge computing based on a discrete butterfly optimization algorithm
60. Awad AA, Ali AF, Gaber T (2020) Feature selection method based on chaotic maps and butterfly optimization algorithm. In: Joint European-US workshop on applications of invariance in computer vision. Springer, pp 159–169
61. Arora S, Singh S (2017) An improved butterfly optimization algorithm with chaos. *J Intell Fuzzy Syst* 32:1079–1088
62. Zhi Y, Weiqing W, Haiyun W, Khodaei H (2020) Improved butterfly optimization algorithm for CCHP driven by PEMFC. *Appl Therm Eng* 173:114766
63. Ustun D (2020) An enhanced adaptive butterfly optimization algorithm rigorously verified on engineering problems and implemented to ISAR image motion compensation. *Eng Comput*
64. Singh B, Anand P (2018) A novel adaptive butterfly optimization algorithm. *Int J Comput Mater Sci Eng* 7:1850026
65. Sharma TK (2020) Enhanced butterfly optimization algorithm for reliability optimization problems. *J Ambient Intell Hum Comput* 2020:1–25
66. Tan LS, Zainuddin Z, Ong P (2020) Wavelet neural networks based solutions for elliptic partial differential equations with improved butterfly optimization algorithm training. *Appl Soft Comput* 95:106518
67. Arora S, Anand P (2018) Learning automata-based butterfly optimization algorithm for engineering design problems. *Int J Comput Mater Sci Eng* 7:1850021
68. Shams I, Mekhilef S, Soon TK (2020) Maximum power point tracking using modified butterfly optimization algorithm for partial shading, uniform shading and fast varying load conditions. *IEEE Trans Power Electron* 36:5569
69. Chen Y (2020) Crowd behaviour recognition using enhanced butterfly optimization algorithm based recurrent neural network. *Multimed Res* 3:20
70. Long W, Jiao J, Liang X, Wu T, Xu M, Cai S (2021) Pinhole-imaging-based learning butterfly optimization algorithm for global optimization and feature selection. *Appl Soft Comput* 103:107146
71. Arora S, Singh S (2017) An effective hybrid butterfly optimization algorithm with artificial bee colony for numerical optimization. *Int J Interact Multimed Artif Intell* 4:14–21
72. Sharma S, Saha AK, Majumder A, Nama S (2021) Mpboa-a novel hybrid butterfly optimization algorithm with symbiosis organisms search for global optimization and image segmentation. *Multimed Tools Appl* 80:12035–12076
73. Dubey AK (2021) Optimized hybrid learning for multi disease prediction enabled by lion with butterfly optimization algorithm. *Sādhanā* 46:1–27
74. Zhang M, Long D, Qin T, Yang J (2020) A chaotic hybrid butterfly optimization algorithm with particle swarm optimization for high-dimensional optimization problems. *Symmetry* 12:1800
75. Sharma S, Saha AK, Ramasamy V, Sarkar JL, Panigrahi CR (2020) hbosos: an ensemble of butterfly optimization algorithm and symbiosis organisms search for global optimization. *Advanced computing and intelligent engineering*. Springer, New York, pp 579–588
76. Sharma S, Saha AK (2021) Bosca-a hybrid butterfly optimization algorithm modified with sine cosine algorithm. *Progress in advanced computing and intelligent engineering*. Springer, New York, pp 360–372
77. Zhou H, Zhang G, Wang X, Ni P, Zhang J (2020) A hybrid identification method on butterfly optimization and differential evolution algorithm. *Smart Struct Syst* 26:345–360
78. Utama DM, Widodo DS, Ibrahim MF, Dewi SK (2020) A new hybrid butterfly optimization algorithm for green vehicle routing problem. *J Adv Transp*. <https://doi.org/10.1155/2020/8834502>
79. Tzanetos A, Dounias G (2021) Nature inspired optimization algorithms or simply variations of metaheuristics? *Artif Intell Rev* 54:1841–1862
80. Bouchekara H (2021) Electric charged particles optimization and its application to the optimal design of a circular antenna array. *Artif Intell Rev* 54:1767–1802
81. Alsattar H, Zaidan A, Zaidan B (2020) Novel meta-heuristic bald eagle search optimisation algorithm. *Artif Intell Rev* 53:2237–2264
82. Mirjalili S (2019) Genetic algorithm. *Evolutionary algorithms and neural networks*. Springer, New York, pp 43–55
83. Al-Betar MA, Doush IA, Khader AT, Awadallah MA (2012) Novel selection schemes for harmony search. *Appl Math Comput* 218:6095–6117
84. Hinterding R, Michalewicz Z, Eiben A (1999) Parameter control in evolutionary algorithms. *IEEE Trans Evol Comput* 3:124–141
85. Makhadmeh SN, Khader AT, Al-Betar MA, Naim S, Abasi AK, Alyasseri ZAA (2021) A novel hybrid grey wolf optimizer with min-conflict algorithm for power scheduling problem in a smart home. *Swarm Evol Comput* 60:100793

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.