



Article

An Improved African Vulture Optimization Algorithm for Feature Selection Problems and Its Application of Sentiment Analysis on Movie Reviews

Aitak Shaddeli ¹, Farhad Soleimanian Gharehchopogh ^{1,*} , Mohammad Masdari ¹ and Vahid Solouk ^{1,2}

¹ Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia 5756151818, Iran

² Faculty of Information Technology and Computer Engineering, Urmia University of Technology, Urmia 5756151818, Iran

* Correspondence: bonab.farhad@gmail.com; Tel.: +98-914-176-427

Abstract: The African Vulture Optimization Algorithm (AVOA) is inspired by African vultures' feeding and orienting behaviors. It comprises powerful operators while maintaining the balance of exploration and efficiency in solving optimization problems. To be used in discrete applications, this algorithm needs to be discretized. This paper introduces two versions based on the S-shaped and V-shaped transfer functions of AVOA and BAOVAH. Moreover, the increase in computational complexity is avoided. Disruption operator and Bitwise strategy have also been used to maximize this model's performance. A multi-strategy version of the AVOA called BAVOA-v1 is presented. In the proposed approach, i.e., BAVOA-v1, different strategies such as IPRS, mutation neighborhood search strategy (MNSS) (balance between exploration and exploitation), multi-parent crossover (increasing exploitation), and Bitwise (increasing diversity and exploration) are used to provide solutions with greater variety and to assure the quality of solutions. The proposed methods are evaluated on 30 UCI datasets with different dimensions. The simulation results showed that the proposed BAOVAH algorithm performed better than other binary meta-heuristic algorithms. So that the proposed BAOVAH algorithm set is the most accurate in 67% of the data set, and 93% of the data set is the best value of the fitness functions. In terms of feature selection, it has shown high performance. Finally, the proposed method in a case study to determine the number of neurons and the activator function to improve deep learning results was used in the sentiment analysis of movie viewers. In this paper, the CNNEM model is designed. The results of experiments on three datasets of sentiment analysis—IMDB, Amazon, and Yelp—show that the BAOVAH algorithm increases the accuracy of the CNNEM network in the IMDB dataset by 6%, the Amazon dataset by 33%, and the Yelp dataset by 30%.

Keywords: African Vulture Optimization Algorithm; binary; multi-strategies; mutation neighborhood search strategy; emotion analysis; movie viewers



Citation: Shaddeli, A.; Soleimanian Gharehchopogh, F.; Masdari, M.; Solouk, V. An Improved African Vulture Optimization Algorithm for Feature Selection Problems and Its Application of Sentiment Analysis on Movie Reviews. *Big Data Cogn. Comput.* **2022**, *6*, 104. <https://doi.org/10.3390/bdcc6040104>

Academic Editor: Isaac Triguero

Received: 22 July 2022

Accepted: 21 September 2022

Published: 28 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Feature selection is usually used as a pre-processing step to find an optimal subset of features considered in a set of all features in the data analysis process. It can be said that the problem is to find the optimal subset of features for better prediction of class tags in a data set with machine learning algorithms and deep learning by removing irrelevant or redundant features. The primary purpose of feature selection methods is to find an optimal subset of features from the original data set [1,2]. It improves the performance of machine learning algorithms and saves memory and CPU time. A data set with n samples is provided that $i = 1, 2, \dots, n$. Each sample consists of d different features that $= 1, 2, \dots, D$, and A_{ij} refer to samples i and j . Moreover, each sample consists of a class or tag, defined by $i = 1, 2, \dots, m$, as shown in Figure 1.

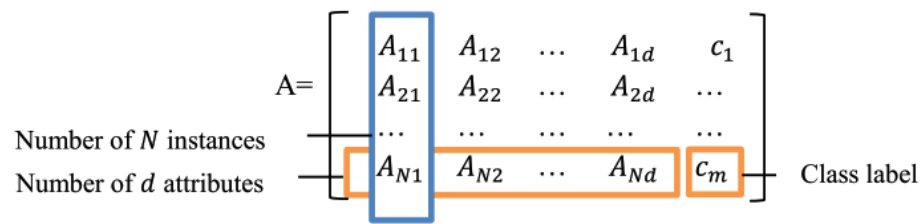


Figure 1. An example of a dataset with features.

Figure 1 shows that class samples may have similar values and features. Moreover, the samples of different classes do not have various features [3] (in terms of number, type of features). Assuming that the original set consists of n samples and d features, then $D \subseteq \mathbb{R}^N$ contains all the features of d . Moreover, feature selection methods aim to find the optimal feature subset from the large D set. Thus, the binary mode for feature selection can be shown as $X = (y_{i1}, y_{i2}, y_{i3}, \dots, y_{id})$. Here, y_{ij} can contain 0 and 1; zero means the feature is not selected, and one means the feature is selected. Solution x is considered a binary solution for feature selection. Feature selection methods can be divided into two general categories [4]: filter-based and wrapper-based. Filter-based methods use mathematical and statistical methods to find essential and dependent output features [5]. However, wrapper-based methods use machine learning and deep learning classification algorithm to evaluate optimal features. Filter-based methods are independent of classification algorithms and are relatively fast. Wrapper-based methods achieve better results than filter-based methods due to classification algorithms in their evaluation model. The challenging discussion in wrapper-based methods is the selection of a subset of features by the optimization algorithm and evaluating this set on a classifier in each step.

A heuristic is an automatic method of selecting or generating a set of heuristics. Hyper-heuristic was divided into two general and main categories [6]. These two categories are selection hyper-heuristic and generation hyper-heuristic. These categories are named ‘heuristics to select heuristics’ and ‘heuristics to generate’ heuristics’ defined and expressed. Hyper-heuristics and selection and generation can be selected or generated based on the nature of heuristics and are divided into two other categories: constructive or perturbative hyper-heuristics [7]. A constructive hyper-heuristic incrementally creates a perfect solution from scratch, while a disruptive hyper-heuristic iteratively improves an existing solution by using its chaotic mechanisms. The heuristics selected or generated by hyper-heuristic are called low-level heuristics (LLHs). A hyper-heuristic model based on selection includes two general levels. The low level has problem representation, evaluation function(s), and specific LLH problems. In the case of the high level, the high level manages which of the LLHs is selected to generate the solution for the new solution. The LLH Generates a new solution to be used, then decides to accept this new solution or not.

A cross-domain assessment was proposed for multi-objective optimization issues in the real world [8]. This paper uses five multi-objective evolutionary algorithms as Low-Level meta-Heuristics (LLHs) to govern a set of election-based, reinforcement learning, and mathematical function hyper-heuristics. According to their findings, hyper-heuristics perform better across domains than a single meta-heuristic, making them an excellent choice for novel multi-objective optimization problems. The authors of [9] proposed a new algorithm for feature selection concerning metaheuristics and hyper-heuristic algorithms. They use hyper-heuristics to give an efficient technique for dealing with complicated optimization challenges for feature selection in industrial and scientific-based domains for text classification. It can be done by using the power of the internet. This paper introduced a brand-new Hyper-Heuristic Feature Selection technique for locating an effective suitable feature subset [10]. Two categories of low-level heuristics are distinguished: the exploiters, which effectively exploit the search space by enhancing the quality of the candidate solution at hand, and the explorers, which thoroughly explore the solution space by focusing on random disturbances.

Recently many meta-heuristic algorithms have been used for the feature selection problem based on the wrapper-based model. Moreover, show high performance compared to traditional feature selection methods. Optimal meta-heuristic algorithms can select the optimal subset or close to an optimal subset of features in good time [11]. Searching for all subsets requires creating possible subcategories to find the solution, which is almost impractical and time-consuming work on a large data set. Most meta-heuristic optimization algorithms are designed to solve continuous optimization problems [12]. In addition to that, to solve the feature selection problem, these algorithms need to be designed with binary operators. Besides this, due to the lack of attention to the features of the feature selection problem, the operators of updating the optimization and optimal meta-heuristic algorithms in high dimensions are of collision constraint. They cannot solve the problem of feature selection well. Therefore, designing an optimization algorithm with powerful operators and its performance in a large data set is a big challenge.

The AVOA is a new meta-heuristic algorithm inspired by African vultures' feeding and orienting behaviors. It consists of powerful operators while maintaining the balance of exploration and efficiency in solving continuous optimization problems. Feature selection can be defined as identifying relevant features and removing irrelevant and repetitive features to observe a subset of features that describe the problem well with minimal loss of efficiency. The importance of feature selection includes understanding the data, gaining knowledge about the process and helping to visualize it and reduce overall data, limiting storage requirements, possibly helping reduce costs, and gaining speed. Therefore, it encouraged us to offer a binary version of the AVOA. In addition, determining the values of the hyper-parametric algorithms of deep learning is a big challenge for analyzing the emotions of the film viewer, which has not yet provided a suitable and accurate solution. Because the parameters of a deep neural network have always been essential, the function of the deep neural network depends mainly on them. Finding the optimal values of deep parameters requires much work, experience, and time, which can be considered an optimization problem. One of the effective methods is to use binary meta-heuristic algorithms in bits to determine the appropriate amount of deep learning Artificial Neural Network hyper-parameters in analyzing movie viewers' emotions. In this paper, to solve these challenges and problems and further evaluate and determine the appropriate value of hyper parameters, we used the BAOVAH approach, presented in the paper's final section with its results. The contribution of this article is as follows:

- It introduces three new binary versions of the AVOA algorithm.
- Introduced a new hybrid version of AVOA with the sine cosine algorithm.
- Presenting a new model of combining meta-heuristic algorithms based on hyper-heuristics.
- Using the Disruption operator mechanism to maximize the performance of the proposed model.
- Using IPRS strategy to generate quality initial population based on ranking strategy.
- Using mutation neighborhood search strategy (MNSS) to maintain the balance between exploration and exploitation in the AVOA algorithm.
- Using multi-parent crossover strategy to increase exploitation and properly implement exploitation step in AVOA algorithm.
- Using the Bitwise method to binaries the operators of the AVOA algorithm.
- Evaluating proposed approaches on 30 UCI datasets, including small, medium, and large datasets.
- Comparing the proposed approaches with filter-based methods.
- Designing CNNEM deep network for emotion analysis.
- Optimizing the parameters of the deep learning method in emotion analysis using the BAOVAH approach.

Section 2 examines the previous works of different types of binary versions of meta-heuristic algorithms. Section 3 describes the AVOA. Section 4 presents three new binary versions of the AVOA algorithm. Section 5 evaluates the proposed methods and com-

compares them with other algorithms. Section 6 considers a case study of in-depth learning and sentiment analysis. The paper's final section presents conclusions and future work in general.

2. Related Works

Researchers have developed many wrapper-based methods based on meta-heuristic algorithms to solve the feature selection problem [13]. We will discuss some of the essential algorithms in the following. The operators of meta-heuristic methods usually work continuously; the authors have attempted to use different strategies to binarize these algorithms to solve the feature selection problem. Transfer functions are often considered the most straightforward methods and usually do not impose any change on the structure of meta-heuristic algorithms.

Moreover, there is sometimes a lack of proper exploration and exploitation in this transfer function. Genetic operators such as mutation and crossover can be used to overcome this problem. However, we need different strategies to balance exploration and exploitation to use meta-heuristic methods while providing higher quality solutions and greater diversity. For this reason, some authors have used new operators to maintain a balance between exploration and exploitation. Some disadvantages are as follows:

- The inefficiency of algorithms in high dimensional datasets;
- Poor convergence of some algorithms due to weak operators;
- Use a single strategy to maintain a balance between exploration and exploitation;
- Getting trapped in local optima due to an imbalance between exploration and exploitation;
- Evaluation of algorithms on a few datasets.

The above disadvantages indicate that using only a few strategies can lead to poor results. However, using several different techniques can ensure the maintenance of a balance between exploration and exploitation and higher quality solutions and diversity. Therefore, the proposed method applies different strategies such as IPRS, mutation neighborhood search strategy (MNSS) (balancing exploration and exploitation), multi-parent crossover (increasing exploitation), and Bitwise strategy (increasing diversity and inquiry) to provide solutions with greater variety and to ensure the quality of the solution. We also proposed the AVOA BAOVAS and BAOVAV for the two versions based on the S-shaped and V-shaped transfer functions, which do not change the structure and operators.

In [14], only the S-shape and V-shape transfer functions are simply used to binary the PSO algorithm. No data sets and feature selection were used to test and evaluate the proposed method, but benchmark mathematical functions were used. Besides this, in this paper, new binary operators are not designed due to the feature selection, and the basic PSO algorithm is used with the transfer function. In addition, no particular operator is designed to escape the local optimization. In [15], a binary version of the GWO algorithm is presented with a change in the structure of the GWO algorithm to solve the feature selection problem. However, the proposed BGWO method has been evaluated on a relatively small and medium-sized data set. Moreover, no special operator is designed to escape the local optimization. In [16], a binary version of the BALO algorithm is presented by changing the structure of the BALO algorithm to solve the feature selection problem. The proposed BALO method is evaluated on a relatively small and medium-sized data set.

In [17], only simple V-shape transfer functions are used to binary the CSA algorithm. Moreover, no changes have been made to the algorithm's structure to increase exploration and productivity. The proposed BCSA method was evaluated on a small dimension and numbers data set. In addition, fewer criteria have been assessed for the proposed method and other algorithms. In [18], only the new transfer functions TV-S shape and TV-V shape are used to binary the BDA algorithm. The innovation of this paper is more related to new transfer functions, but no changes have been made to the operators of the BDA algorithm. The proposed BDA method is evaluated on a relatively small and medium-sized data set. In addition, no special operator is designed to escape local optimization.

In [19], the author tries to use a version based on transfer functions and genetic operators to binary the SSA algorithm. Although no new operator is provided for the feature selection problem, this algorithm has compared algorithms with more and better statistical criteria. In [20], only an S-shape transfer and chaotic functions are used to binary the CSA algorithm. The innovation of this article is more about using ten chaotic maps, and no changes have been made to the operators of the CSA algorithm. The proposed algorithm has not been evaluated in the data set with dimensions of more than 82 items. In [21], only the S-shape and V-shape transfer functions are used to binary the BBO algorithm. The innovation of this paper is more related to transfer functions, but no changes have been made to the operators of the BBO algorithm. The proposed BBO method is evaluated on a relatively small and medium-sized data set. Moreover, no special operator is designed to escape local optimization. In [22], they tried to use opposition and local search operators to improve the SA algorithm in feature selection. However, the proposed algorithm was evaluated on a smaller data set. It also achieved worse results in several datasets in terms of several features.

In [23], only one V-shape transfer function is used to binary the COA algorithm, and there is no innovation in the structure of the COA algorithm. In [24], transfer functions and two mutation steps are used to binary the GWO algorithm to solve the feature selection problem. Indeed, the evaluation of the proposed method is done only on the data set with low dimensions (less than 62). In [25], a new version of the HHO algorithm is presented based on the mutation and opposition operator to solve the feature selection problem. The proposed algorithm may have a more complex time than the base version of the HHO algorithm.

In [26], only the S-shape and V-shape transfer functions are used to binary the WOA algorithm. The innovation of this paper is more related to transfer functions, but no changes have been made to the operators of the WOA algorithm. Moreover, no changes have been made to the operators of the WOA algorithm. A new binary version of the GSKO algorithm for feature selection is proposed [22]. The evaluation of the proposed algorithm is not based on a real problem. In [27], two binary versions of the SOS algorithm with an S-shape transfer function and new binary operators are presented. This paper evaluates both proposed methods with a smaller data set, and a special operator is designed to escape the local optimization.

In [28], four S-shape and four V-shape transfer functions are used to binary the EPO algorithm. The innovation of this paper is more related to transfer functions, but no changes have been made to the operators of the BBO algorithm. In addition, no special operator is designed to escape local optimization. In [29], two binary versions of the FFA algorithm with an S-shape transfer function and new binary operators are presented. This paper evaluates both proposed methods with a smaller data set. Moreover, no special operator is designed to escape local optimization.

In this paper, due to the deficiency and disadvantages of other methods, we have tried to provide a powerful algorithm for solving feature selection. Therefore, this paper presents a practical study of the AVOA algorithm using transfer functions, four different strategies, and new binary operators.

3. Enhanced AVOA with Hyper-Heuristic (Approach-1)

In this section, the mechanisms used in the first proposed model, as well as how to integrate the above mechanisms, are discussed.

3.1. AVOA

The AVOA is a new meta-heuristic algorithm inspired by African vultures' feeding and orientation behaviors [30]. According to the conduct of vultures in nature, vultures are divided into two categories based on physical strength. Moreover, the desire to eat of vultures and hours of searching for food by vultures causes escaping from the hungry trap. Moreover, two of the best solutions are considered the strongest and best vultures. In this

section, the general steps of this algorithm are summarized, which can be referred to [30] for more details.

3.1.1. Determining the Best Vulture in Each Group

In the first step of the AVOA algorithm, the first and second vultures are selected for global optimization based on the fitness function. Then other solutions according to the following formula show the possibility of choosing the vultures.

$$R(i) = \begin{cases} \text{Best Vulture}_1 & \text{if } p_i = \alpha \\ \text{Best Vulture}_2 & \text{if } p_i = \beta \end{cases} \quad (1)$$

In Equation (1), α and β are quantified parameters before the search operation. The value of these parameters is between zero and one, and the sum of the importance of both parameters is equal to 1. Furthermore, choose any of the best solutions by using the roulette wheel. If α is close to one, it increases exploitation. Vulture vigor rate due to vulture behavior, lack of energy, and aggressive behavior while searching for food is formulated with a mathematical form in Equation (2).

$$F = (2 \times rand_1 + 1) \times y \times \left(1 - \frac{It_i}{maxIt}\right) \quad (2)$$

In Equation (2), f indicates the degree of satiety of the vulture, It_i Indicates the current iteration number, $maxIt$ represents the total number of iterations, and y is a random number between -1 and 1 that changes each time it is repeated. $rand_1$ is a random number between 0 and 1 ; when the value of y is less than 0 , the vulture is hungry; otherwise, vultures are total. In the AVOA algorithm, the value of the variable F strikes a balance between exploration and exploitation. When the value of $|F|$ is more than 1 , it enters the exploration phase; otherwise, it enters the exploitation phase [30].

3.1.2. Exploration Phase

Population exploration and diversity are formulated at this phase of the AVOA algorithm. For this phase, two strategies can be adjusted according to a parameter called $p1$ with a value between 0 and 1 . To choose any of the strategies in the exploration phase $p1$, which is a random number between 0 and 1 , if this number is more significant than or equal to parameter 1 , Equation (3) is used, but if $randp1$ is smaller than parameter $p1$, Equation (4) is used.

$$\begin{aligned} V(i+1) &= R(i) - D(i) \times F \\ D(i) &= |X \times R(i) - V(i)| \end{aligned} \quad (3)$$

According to Equation (3), $V(i+1)$ indicates the vector position of the vulture in the next iteration, f indicates the degree of the vulture satiety, $R(i)$ indicates one of the best-chosen vultures and x , which means the random movement of the lead vulture to protect food from other vultures. X is used as a coefficient vector that increases a random motion that changes with each iteration and is obtained using the $X = 2 \times Rand$ formula that $Rand$ is a random number between zero and one. V_i is the current vector position of the vulture [30].

$$V(i+1) = R(i) - F + rand_2 \times ((ub - lb) \times rand_3 + lb) \quad (4)$$

In Equation (4), $rand_3$ is a random number between zero and one. Moreover, ub and lb represent the upper and lower bounds. $rand_3$ has been used to increase the coefficient of random quiddity.

3.1.3. Exploitation Phase

At this phase of the AVOA algorithm, efficiency and convergence are formulated. For this phase, two strategies with values between 0 and 1, considering two parameters, p_2 and p_3 , are adjusted [30]. In the following, each strategy for the exploitation phase is listed.

Phase 1

In the first phase, two different strategies of rotation flight and siege and quiet quarrel are performed. To determine the selection of each strategy, the p_2 parameter is formulated as Equation (5) [30]. That $rand_{p_2}$ is a random number between zero and one.

$$V(i+1) = \begin{cases} \text{Equation (6)} & \text{if } P_3 \geq rand_{p_2} \\ \text{Equation (8)} & \text{if } P_3 < rand_{p_2} \end{cases} \quad (5)$$

Siege and a quiet quarrel: At this phase of the AVOA algorithm, the severe quarrels of weak vultures on food acquisition and vital vultures are formulated, as shown in Equation (6).

$$V(i+1) = D(i) \times (F + rand_4) - d(t) \quad (6)$$

$$d(t) = R(i) - V(i) \quad (7)$$

In Equation (6), $rand_4$ is a random number between 0 and 1 increases the random coefficient. R_i is one of the best-selected vultures of the two groups. V_i is the current vector position of the vulture. Using this equation, the distance between the vulture and one of the best vultures of the two groups is obtained [30].

Rotational movement of vultures: In this step of the AVOA algorithm, rotational motion is modeled using spiral motion. The distance between the vulture and one of the two best vultures is first calculated in this method. Moreover, a spiral equation between the vulture and one of the best vultures is created. The motion is proportional to f .

$$V(i+1) = D(i) \times e^{SF} \times \cos(2\pi F) + R(i) \quad (8)$$

In Equation (8), S is a parameter to determine the logarithmic model of spiral shape.

Phase 2

In this phase of the AVOA algorithm, the movement of all vultures towards the food source is examined. To determine the selection rate of each of the strategies, the p_2 parameter is formulated as Equation (9), which $rand_{p_3}$ is a random number between zero and one.

$$V(i+1) = \begin{cases} Eq11 & \text{if } P_3 \geq rand_{p_3} \\ Eq12 & \text{if } P_3 < rand_{p_3} \end{cases} \quad (9)$$

The gartering of several types of vultures in the food source: In this phase, the movement of all vultures towards the food source is examined. There are times when vultures are hungry, and there is much competition for food, and several types of vultures may accumulate on the same food source, which is modeled according to Equations (11) and (12).

$$\begin{aligned} A_1 &= \text{Best Vulture}_1(i) - D(i) \times F \\ A_2 &= \text{Best Vulture}_2(i) - D(i) \times F \end{aligned} \quad (10)$$

In Equation (10), best vulture 1(i) is the best vulture of the first group in the current iteration, and best vulture 2(i) is the best vulture of the second group in the current iteration.

$$V(i+1) = \frac{A_1 + A_2}{2} \quad (11)$$

Finally, all vultures are aggregated using Equation (11).

Siege and fierce quarrels: At this phase of the AVOA algorithm, the lead vultures are hungry and weak; on the other hand, vultures have also become aggressive. Moreover,

move in several directions towards the leading vulture. Equation (12) has been used to model this movement [30].

$$V(i+1) = R(i) - |d(t)| \times F \times Levy(d) \quad (12)$$

In Equation (12), $d(t)$ indicates the distance of the vulture to one of the best vultures of the two groups. The flight patterns of the levy flight function have been used to increase the effectiveness of the AVOA algorithm in Equation (12).

3.2. Sine Cosine Algorithm (SCA)

SCA is one of the powerful meta-heuristic algorithms presented by Mirjalili [31], which has been used to solve many problems. This algorithm uses two mathematical functions, Sine and Cosine, for optimization operations. In the SCA algorithm, Equation (13) is used to change the position of the search agent vector.

$$x_i(t+1) = \begin{cases} x_i(t) + r_1 \times \sin(r_2) \times |r_3 P_i(t) - x_i(t)|, & r_4 < 0.5 \\ x_i(t) + r_1 \times \cos(r_2) \times |r_3 P_i(t) - x_i(t)|, & r_4 \geq 0.5 \end{cases} \quad (13)$$

In Equation (13), $x_i(t)$ shows the current location of the i th particle and the t th iteration. r_1 , r_2 and r_3 are random numbers and also r_3 are random numbers between 0 and 1. P_i is the destination of the particle. SCA meta-heuristic Algorithm 1 has four main parameters r_1 , r_2 , r_3 , and r_4 . The parameter r_1 represents the next position of the particle. This place may be inside the answer space or outside it. The parameter r_2 shows how much the particle's movement moves towards the destination or against the direction of the destination. The parameter r_3 gives a certain weight to the destination position, and finally, the parameter r_4 indicates whether to use the sine component or the cosine component. Equation (14) is used to balance the extraction and exploration phases.

$$r_1 = a - t \frac{a}{T} \quad (14)$$

In Equation (14), t is the current iteration number, T represents the total number of iterations, and a is a constant value.

Algorithm 1: Pseudo-code of SCA algorithm.

```

Initialize a set of search agents (solutions)(X)
Do
  Evaluate each of the search agents by the objective function
  Update the best solution obtained so far ( $P = X^*$ )
  Update  $r_1$ ,  $r_2$ ,  $r_3$ , and  $r_4$ 
  Update the position of search agents using Equation (14)
While ( $t < \text{maximum number of iterations}$ )
Return the best solution obtained so far as the global optimum

```

3.3. Modified Choice Function

In [32], the authors presented a hyper-heuristic based on the Choice function and scoring approach. The values of each LLH are calculated based on the previous performance of the LLH. The value of each LLH is calculated based on three separate criteria f_1 , f_2 and f_3 . The f_1 criterion, which is the recent performance of each LLH, is calculated using Equation (15):

$$f_1(h_j) = \sum_n \alpha^{n-1} \frac{I_n(h_j)}{T_n(h_j)} \quad (15)$$

In Equation (15), h_j is the same as LLH^j. $I_n(h_j)$ Shows the difference value between the current solution and the new solution with the n^{th} application of h_j , $T_n(h_j)$ expresses the amount of time spent by the n^{th} application of h_j to propose a new solution, α is a parameter

between zero and one that prioritizes recent performance. The second criterion, f_2 , indicates the dependence between a consecutive pair of LLHs, calculated using Equation (16).

$$f_2(h_k, h_j) = \sum_n \beta^{n-1} \frac{I_n(h_k, h_j)}{T_n(h_k, h_j)} \quad (16)$$

In Equation (16), $I_n(h_k, h_j)$, shows the amount of difference between the current solution and the new solution using the n^{th} consecutive application of h_k and h_j (that is, h_j is executed right after h_k), $T_n(h_k, h_j)$ specifies the amount of time spent by the n^{th} consecutive application of h_k and h_j to propose a new solution, β is a parameter between zero and one that prioritizes the recent performance. These two criteria f_1 and f_2 are the components of the Choice function's escalation, which increase the selection of LLHs with better performance. The third criterion, f_3 , the time elapsed since the last execution of a particular LLH is calculated using Equation (17).

$$f_3(h_j) = \tau(h_j) \quad (17)$$

In Equation (17), $\tau(h_j)$ represents the elapsed time since the last execution of h_j (in seconds). Note that f_3 plays a role as a diversity component in the selection function and prioritizes those LLHs that have not been used for a long time. The score of each LLH is calculated using the sum of the weights of all three criteria f_1 , f_2 , and f_3 as shown in Equation (18).

$$F(h_j) = \alpha f_1(h_j) + \beta f_2(h_k, h_j) + \delta f_3(h_j) \quad (18)$$

In Equation (18), α , β and δ are the parameters that show the weights of f_1 , f_2 and f_3 criteria, which are fixed values in the initial model. In [33], the authors presented an improved version of the hyper-heuristic to increase efficiency and performance. In this version, the parameters can be dynamically controlled during execution. In this version, if an LLH improves the solution, the values of the α and β parameters increase compared to the improvement of the new solution with the previous solution. Still, if the selected LLH does not improve the solution, the values of the α and β parameters decrease due to the difference in the costs of the new solution and the previous solution. In this version of the selection function, parameters α and β are combined in a single parameter called μ , and finally, the score of each LLH is calculated using Equation (19).

$$F_t(h_j) = \mu_t [f_1(h_j) + f_2(h_k, h_j)] + \delta f_3(h_j) \quad (19)$$

If an LLH improves the solution, the resonance component is prioritized, and the μ parameter becomes the maximum static value close to one. At the same moment, the δ parameter decreases to the minimum static value close to zero. If LLH does not improve the solution, the μ parameter is penalized linearly, and the lower limit is 0.01. This mechanism causes the δ parameter to grow at a uniform and low rate so that the resonance components do not lose their effectiveness quickly. The parameters μ and δ are calculated using Equations (20) and (21), which show the difference between the cost of the previous solution and the cost of the new solution in Equation (20).

$$\mu_t(h_j) = \begin{cases} 0.99, & d > 0 \\ \max[0.01, \mu_{t-1}(h_j) - 0.01], & d \leq 0 \end{cases} \quad (20)$$

$$\delta_t(h_j) = 1 - \mu_t(h_j) \quad (21)$$

In the proposed model, the Modified Choice Function has been used to change the optimization algorithm during optimization so that the performance and efficiency of the proposed model can be greatly increased by using this mechanism.

3.4. Disruption Operator (DO)

DO is used to increase population diversity, inspired by astrophysical phenomena. This operator has been used to improve the ability in the search process of the proposed model in the problem space and to create a balance between exploration and exploitation mechanisms. The perturbation operator is formulated using Equation (22).

$$D_{op} = \begin{cases} D_{is,i,j} \times \delta(-2.2). & \text{if } D_{is,i,best} \geq 1 \\ 1 + D_{is,i,best} \times \delta\left(\frac{-10^{-4}}{2}, \frac{10^{-4}}{2}\right). & \text{otherwise} \end{cases} \quad (22)$$

In Equation (22), $D_{is,i,j}$ represents the Euclidean distance between the i th and j th solutions, where the j th solution is located near and adjacent to the i th solution. The Euclidean distance is between the i th and the best solution. Moreover, $\delta(x,y)$ is a random number generated in $[x,y]$ interval [34].

3.5. Bitwise Strategy (BS)

To escape from the local optimal points and solve the problem of low population diversity, two operators, bitwise OR operation and bitwise AND operation, have been used. In the first step, a random solution is created in these two operators. Then a bitwise AND operation is performed between the generated new, unexpected solution and the best solution. The main goal in the bitwise AND operator is to obtain good features in the best and the random new solutions.

Meanwhile, the solution obtained from AND operation and the new solution generated from the proposed model are used as input for the OR operation. Moreover, the primary purpose of the OR operation is to transfer features that are better and more useful than other features that come from the AND operation, which leads to the production of new and high-quality solutions. In addition, Bitwise operations can lead to escape from local optimal points because random shuffles can increase population diversity and improve solution quality. Figure 2 shows the Bitwise operation.

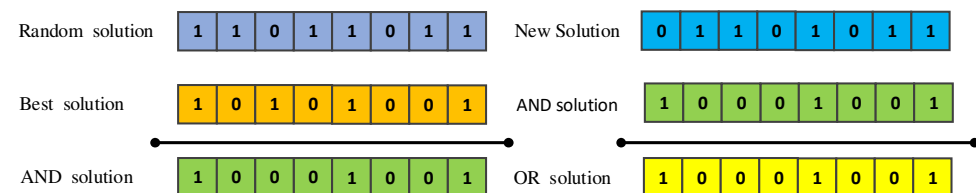


Figure 2. Bitwise operations.

4. Hyper-Heuristic Binary African Vultures Optimization Algorithm (Approach-1)

The AVOA and its steps are described in Section 3. The standard AVOA is designed for continuous problems, and the solutions produced in this algorithm include decimal values. On the other hand, according to the nature of the problem, it is necessary to integrate certain mechanisms into the AVOA to achieve a powerful model with the ability to reach quality solutions. For this reason, we have used and benefited from several mechanisms in the proposed model, which are fully explained in Section 3. According to the feature of the AVOA, initially presented to solve continuous optimization problems, it focuses more on exploration mechanisms at the beginning of the optimization operation. Still, in the later stages, it tends more towards exploitation mechanisms. This approach is suitable for solving continuous problems, but the AVOA can get stuck in local optimal points in solving discrete problems.

We have integrated the AVOA with the sine and cosine algorithms. The reason for our choice of the sine–cosine algorithm is the unique features of this algorithm in increasing the diversity component. By using this algorithm, diversity can be optimized in all stages increased. However, to integrate the sine-cosine algorithm with the African vulture algorithm, the Modified Choice Function has been used. The main reason for choosing MCF is

the intelligent choice of AVOA and SCA algorithms. You can switch between algorithms because each can perform differently in different optimization stages. In the next step, after updating the solutions using optimization algorithms, two other mechanisms are used, which are DO and BS, because these two mechanisms can significantly increase the solutions' quality. To increase the computational complexity of the proposed model, after updating the solutions using optimization algorithms, only one of the DO and BS mechanisms is used to update the solutions. A random number between 0 and 1 is generated to select these mechanisms. If the generated random number is greater than 0.5, the DO mechanism is selected; otherwise, the BS mechanism is selected.

Due to their simplicity and lack of complexity, the transfer functions are used with the help of random thresholding to convert continuous space to binary space. In general, transfer functions are S-shaped or V-shaped, and each of these functions has different types. Figures 3 and 4 illustrate the output of S-shaped and V-shaped transfer functions, respectively. This section presents two versions of the S-shaped and V-shaped proposed methods, BAOVAS and BAOVAV. In the BAOVAS version, we used four S-shaped transfer functions to binarize the AVOA, as described in the following.

$$SG1(V_i^{t+1}) = \frac{1}{1 + e^{-2V_i^{t+1}}} \quad (23)$$

$$SG2(V_i^{t+1}) = \frac{1}{1 + e^{-V_i^{t+1}}} \quad (24)$$

$$SG3(V_i^{t+1}) = \frac{1}{1 + e^{(-\frac{V_i^{t+1}}{2})}} \quad (25)$$

$$SG3(V_i^{t+1}) = \frac{1}{1 + e^{(-\frac{V_i^{t+1}}{3})}} \quad (26)$$

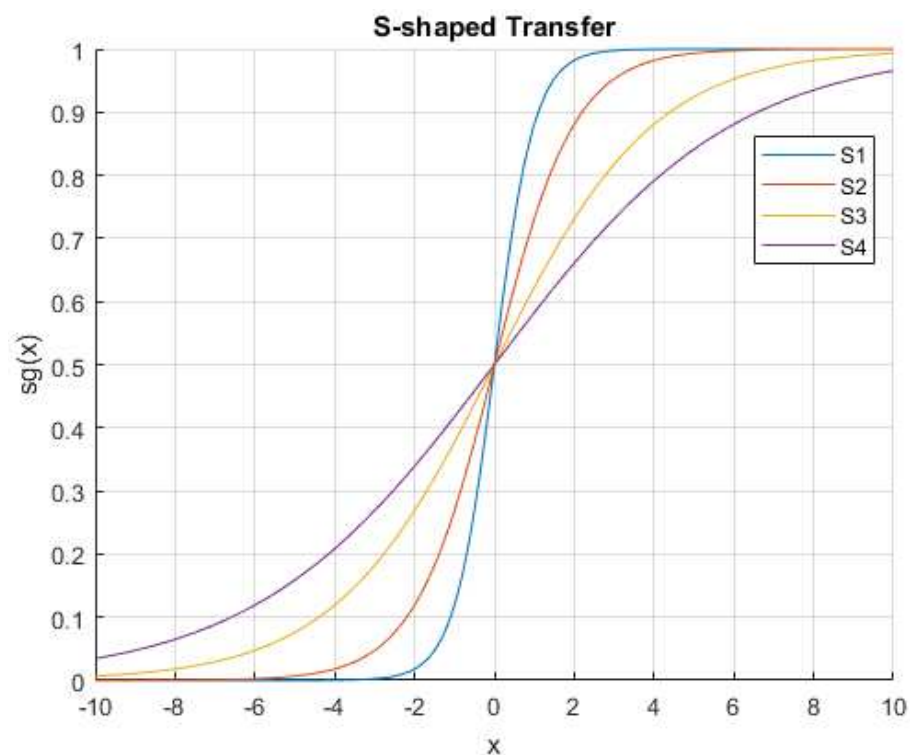


Figure 3. A graphic view of various types of S-shaped transfer function.

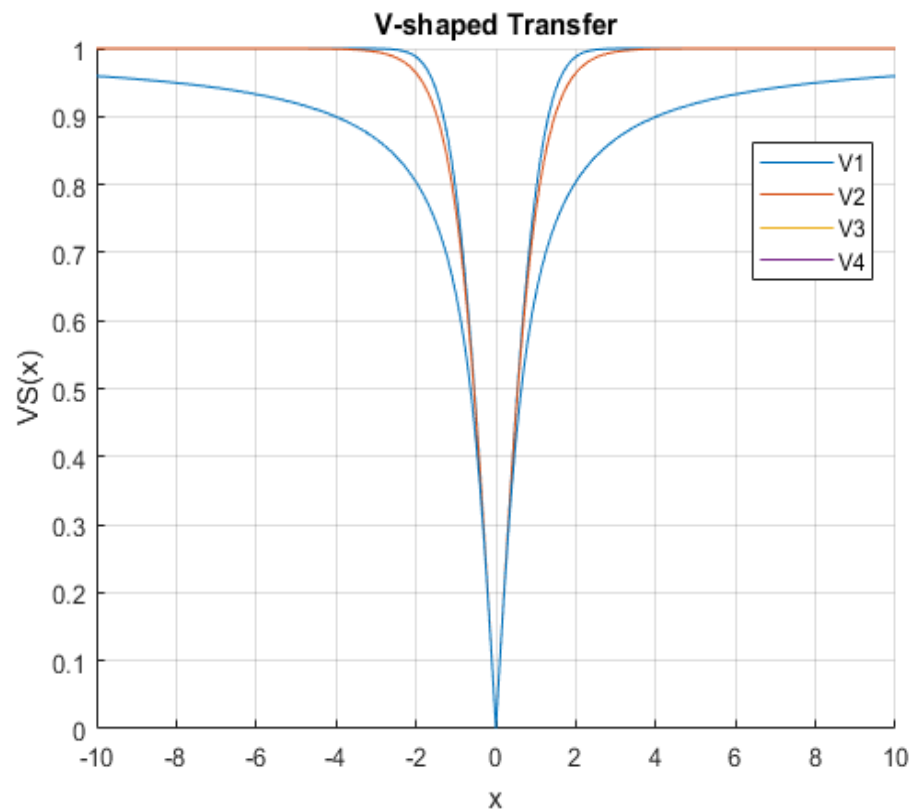


Figure 4. A graphic view of various types of V-shaped transfer function.

In the BAOVAS version, the solution V_i^{t+1} is transferred from the continuous state to the new space by each vs. transfer function between 0 and 1. In the BAOVAS version, we used four V-shaped transition functions to binary the AVOA, described in the following four functions.

$$VS1(X_i^{t+1}) = \left| \operatorname{erf}\left(\frac{\sqrt{\pi}}{2} V_i^{t+1}\right) \right| \quad (27)$$

$$VS2(X_i^{t+1}) = \left| \tanh(V_i^{t+1}) \right| \quad (28)$$

$$VS3(X_i^{t+1}) = \left| \frac{V_i^{t+1}}{\sqrt{1 + V_i^{t+1}{}^2}} \right| \quad (29)$$

$$VS4(X_i^{t+1}) = \left| \frac{2}{\pi} \arctan\left(\frac{\pi}{2} V_i^{t+1}\right) \right| \quad (30)$$

In the BAOVAS version, the solution V_i^{t+1} is transferred from the continuous state to the new space by each vs. transfer function between 0 and 1. After each solution in the AVOA becomes a constant space between zero and one, the following interface can convert binary vectors.

$$X_i^{t+1} = \begin{cases} 0 & \text{if } r < VS|SG(V_i^{t+1}) \\ 1 & \text{if } r \geq VS|SG(V_i^{t+1}) \end{cases} \quad (31)$$

In Equation (31), r is a number between 0 and 1, which determines the final value of each of the dimensions of the solution V_i^{t+1} , which eventually produces a new binary solution. BAOVAH based S-shape and V-shape are shown in Algorithm 2. And flowchart of the this approach are shown in Figure 5.

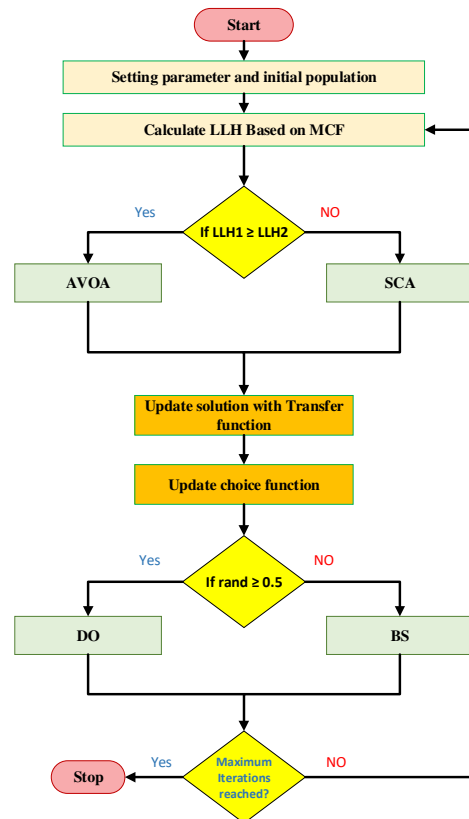
Algorithm 2: BAOVAH based S-shape and V-shape.

```

01: setting parameter
02: Initialize the random binary population  $V_i (i = 1. 2. \dots N)$ 
03: For  $it = 1: \text{MaxIt}$ , do
04:   Calculate the fitness according to Equation (23)
05:   get first and second Vulture Best_vulture1 and Best_vulture2
06:   for  $i = 1: N$ , do
07:     Calculate LLH Based On MCF.
08:     if  $(LLH1 \geq LLH2)$ , then
09:       Select the AVOA algorithm
10:     Else
11:       Select the SCA algorithm
12:     End if
13:     Update the location  $VS|SG(V_i^{t+1})$  According to Equation (31)
14:     
$$V_i^{t+1} = \begin{cases} 0 & \text{if } r < VS|SG(V_i^{t+1}) \\ 1 & \text{if } r \geq VS|SG(V_i^{t+1}) \end{cases}$$

15:     updateChoiceFunction(selectedLLH) // Equation (19)
16:     if  $(rand \geq 0.5)$ , then
17:       Apply  $D_{op}$  the formula for all individuals and their opposite makes more suitable positions for them
18:     Else
19:       Generate a random solution
20:       Apply AND operation (Selected Leader, random solution)
21:       Apply OR operation (AND Solution,  $V_{binary}(t + 1)$ )
22:       If fitness (OR solution) < fitness (Leader) then Update Leader = OR solution
23:     End if

```

**Figure 5.** Flowchart of the proposed BAOVAH approach.

5. Multi-Strategy Binary African Vultures Optimization Algorithm (Approach-2)

This paper presents a new approach to the AVOA, i.e., BAVOA-v1, using several robust strategies for solving feature selection and binarization problems. In the proposed BAVOA-v1 approach, techniques such as IPRS strategy mutation neighborhood search strategy (MNSS) (balancing exploration and exploitation), multi-parent crossover strategy (increasing exploitation), and Bitwise strategy (increasing diversity and exploration) are applied to provide solutions with a greater variety and to ensure the quality of the solution. Considering these four different strategies and the exploration and exploitation steps of the AVOA, each of the operators is correctly implemented in the desired step; see Figure 6 to better understand the flowchart of the proposed BAVOA-v1 approach.

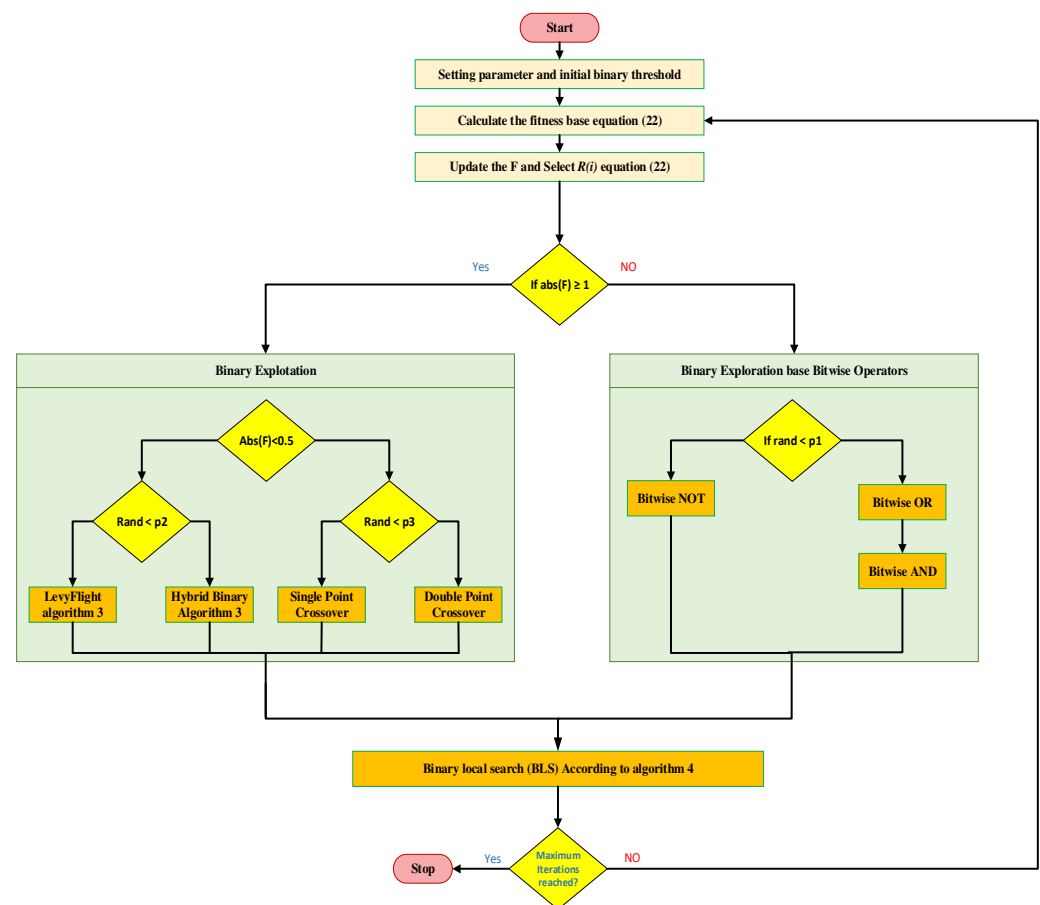


Figure 6. Flowchart of the proposed BAVOA-v1 approach.

5.1. Initial Population Based on Ranking Strategy (IPRS)

The first step in any meta-heuristic algorithm is randomly generating the initial population. Nonetheless, considering the nature of the problem, there are different ways to improve the quality of the solution produced by AVOA from the beginning. Feature selection seeks two goals: selecting fewer features and acquiring higher accuracy. Suppose these two essential goals are intelligently incorporated into the AVOA from the start in the initial population generation step. In that case, they can play a critical role in the exploitation and convergence rate of the AVOA. In the proposed algorithm, we use the Information Gain Ranking (IGR) strategy to rank the features of each dataset. The IGR strategy helps the AVOA in the selection of more efficient features and thus the production of a higher quality initial solution. IGR is an entropy-based metric that measures the importance of a feature (the amount of information each feature carries). It is a good criterion for determining feature relevance for classification. As a statistical method, IG

assigns the weight of features according to the correlation between features and classes. Assuming a dataset called D , the gain of feature F can be expressed by Equation (32).

$$Gain_f = Entropy(Data) - \sum_v \frac{|Data_v|}{|Data|} Entropy(Data_v) \quad (32)$$

In Equation (32), the number of elements $Data$ and $Data_v$ are respectively shown as $|Data|$, and $|Data_v|$; v is all unique values of the feature. After obtaining each feature's IG, it can be represented for the k feature as $GF = [Gainf1, Gainf2, \dots, Gainfk]$. The production of an initial solution can be expressed as Equation (34).

$$SG = \sum_{j=1}^k GF_j \quad (33)$$

$$Prop_j = \frac{GF_j}{SG} \quad (34)$$

$$X(i,j) = \begin{cases} 1 & \text{if } rand < Prop_j \\ 0 & \text{if } rand > Prop_j \end{cases} \quad (35)$$

In Equation (34), the GF represents the feature rankings by IG, and SG is the sum of the rankings of all the features. In addition, $Prop_j$ calculates the probability of selection of a feature based on SG and GF . Finally, in Equation (35), we used a clever and relatively random technique to generate the initial solution. The higher the ranking of a feature by IG, the more likely it is to be selected by the variable $Prop_j$. Moreover, we used a $rand$ to generate relatively random solutions based on higher-ranking features.

5.2. Mutation Neighborhood Search Strategy (MNSS)

The Mutation Neighborhood Search Strategy (MNSS) is an essential technique for balancing global and local search, first proposed by Das et al. [35] in 2009. The critical point in MNSS is that the mutation operator is used to search the neighborhood of the best solution (small area). In this paper, we use this strategy to improve the results of the BAVOAH algorithm. Note that, to control MNSS, we use the greedy rule: first, the mutation is carried out on the best current solution, then if the new position is better than the position of the current best solution, the current best solution is replaced by the MNS mutated solution; thereby, MNS-based local search is implemented. In the MNS strategy, to create neighbors for the first and last features, all the solutions are generated in the form of rings so that it is possible to make mutations to the right and left quickly. Figure 7 makes it easier to understand the MNS strategy in the form of a ring and a right neighborhood.

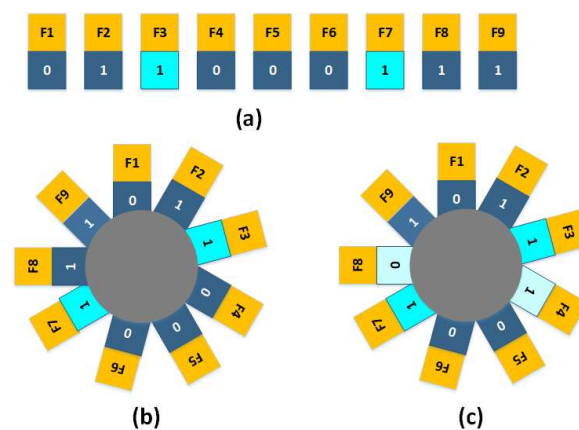


Figure 7. Mutation neighborhood search strategy (a) feature solution, (b) ring feature solution, (c) right neighbors MNSS.

According to Figure 7, first, a solution is usually selected at two mutation points (Figure 7a); then, the solution is considered as a ring (Figure 7b); finally, the MNS strategy produces a new solution based on the right neighbor's neighborhood from two points (Figure 7c). Note that the MNS strategy can also use the left neighborhood. So, the MNS strategy first generates a right neighborhood, then compares them to each other, then generates a left neighborhood on the final solution and compares the two solutions.

5.3. Multi-Parent Crossover Strategy (MPCS)

Genetic operators (selection, mutation, crossover) are inseparable in most discrete problems, including feature selection. Crossing over two parents, the crossover operator plays a vital role in exploitation. However, the point to be noted while using this operator is that if the parents are diverse, it will generate more diverse new solutions, and this will cause late convergence of the algorithm. On the other hand, if the parents are very similar in different iterations, it generates a repetitive and similar solution, thus causing early convergence. A multi-parent crossover operator was proposed by Al-Sharhan et al. [36] in 2019 to overcome this problem, where three parents can be defined as follows:

$$\sigma_1 = X_1 + \beta \times (X_2 - X_3) \quad (36)$$

$$\sigma_2 = X_2 + \beta \times (X_3 - X_1) \quad (37)$$

$$\sigma_3 = X_3 + \beta \times (X_1 - X_2) \quad (38)$$

In Equation (36), β is a randomly selected position of the first parent used in crossover operations. In the AVOA algorithm, we have three main solutions in the exploitation phase (first solution: first-best solution; second solution: second-best solution; third solution: current solution); we can use the MPCS operator at this stage, which can be expressed as:

$$\sigma_1 = Vulture_{Best1} + \beta \times (Vulture_{Best2} - V_{current}) \quad (39)$$

$$\sigma_2 = Vulture_{Best2} + \beta \times (V_{current} - Vulture_{Best1}) \quad (40)$$

$$\sigma_3 = V_{current} + \beta \times (Vulture_{Best1} - Vulture_{Best2}) \quad (41)$$

In the exploitation step in the basic AVOA algorithm (important sub-steps of the accumulation of several vultures around the food source, the siege, and aggressive quarrel), the current solution is updated by the best solution in the current iteration $Vulture_{Best1}$ (i) and the second-best current solution $Vulture_{Best2}$ (ii). Therefore, in the binary version of BAVOA-v1, we designed a binary operator based on the same behavior, so that these three solutions can be combined to generate a binary value (Figure 8).

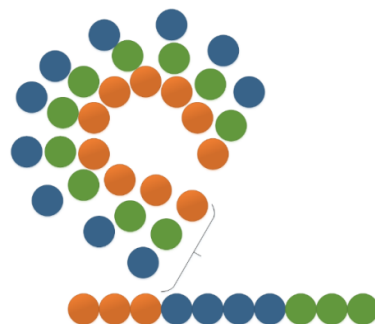


Figure 8. Multi-parent Crossover strategy with three solutions.

Since we use the two best solutions at this stage, premature convergence may occur in the algorithm. The current solution is updated by one of the best solutions and levying light on the siege and aggressive quarrel sub-steps to avoid it. Therefore, in the binary version of BAVOA-v1, we used the same behavior due to a direct levy flight and converted

all of its values to binary with thresholding. The pseudo-code of this binary behavior is shown in Algorithm 3 (8:6).

Algorithm 3: Binary exploitation phase 2.

```

01: dim = length( $V_{current}$ );
02: if abs(F)<0.5
03:   if rand<p2
04:     Multi parent Crossover strategy:  $Vulture_{Best1}$ ,  $Vulture_{Best2}$ ,  $V_{current}$ 
05:   else
06:     update  $V_{current}$  by levy Flight and  $V_{random}$ 
07:     convert binary  $V_{current}$  by threshold
08:   end if

```

The first exploitation stage in the basic AVOA algorithm consists of two sub-stages: siege, gentle quarrel, and rotational movement of vultures. We can implement the exploitation step using the crossover operator. Therefore, we used a variety of crossover operators to implement this step in binary. As a result, the first exploitation step in the binary version of BAVOA-v1 is defined as a single-point and a double-point crossover operator, as shown in Equation (42).

$$V(i+1) = \begin{cases} \text{SinglePointCrossover} & \text{if } P_3 \geq \text{rand}_{p2} \\ \text{DoublePointCrossover} & \text{if } P_3 < \text{rand}_{p2} \end{cases} \quad (42)$$

In Equation (42), the Single Point Crossover and Double Point Crossover operators are applied to increase exploitation in the binary version of BAVOA-v1.

5.4. Bitwise Strategy (BS)

The meta-heuristic algorithm uses the Bitwise operator to increase population diversity and avoid getting trapped in local optima. On the other hand, in [37], it has been proven that the Bitwise operator performs better regarding the final solution accuracy, convergence speed, and robustness. The operators AND, OR, and NOT are three critical Bitwise operators that can play a crucial role in generating a new binary solution (example in Figure 9). Therefore, we also used the three operators for exploration in this step. The binary pseudo-code of the exploration step of the BAVOA-v1 algorithm is shown in Algorithm 3.

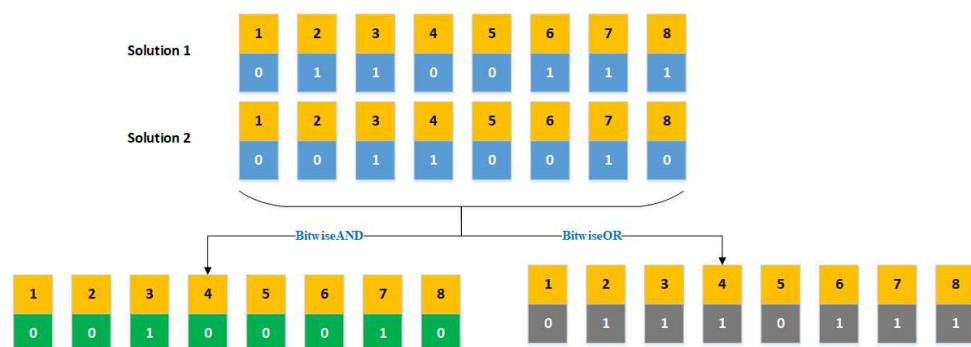


Figure 9. Bitwise Operators (AND, OR).

In Algorithm 4, the basic AVOA algorithm is applied so that V_{random} is one of the best solutions in the population, selected using Equation (1) in each iteration; $V_{current}$ represents the position of the current solution. Based on the exploration step of the basic AVOA algorithm, the new p value-based solution may be updated by the solution of V_{random} or take a new value according to the upper (ub) and the lower (lb) bounds. In the binary version of BAVOA-v1, considering these principles of the basic AVOA algorithm,

this step is performed in two ways: (1) If p_1 is larger than $rand$, then the two solutions V_{random} and $V_{current}$ are updated by two operators “AND” and “OR”, and thereby a binary vector is generated based on these two solutions; (2) If p_1 is less than $rand$, then the solution $V_{current}$ is updated by the operator “NOT” with a probability of 50%, and thereby, a binary vector is generated based on these two solutions.

Algorithm 4: Binary Exploration base Bitwise Operators.

```

01: dim = length( $V_{current}$ );
02: if rand < p1%—use “AND” and “OR” Bitwise—
03:    $Vec_{or}$  = BitwiseOR( $V_{current}$ ,  $V_{random}$ )
04:    $Vec_{and}$  = BitwiseAND( $V_{current}$ ,  $V_{random}$ )
05:   for i = 1 to dim
06:     r = rand;
07:     if (r < 0.25)
08:        $V_{current}(i)$  =  $Vec_{and}(i)$ ;
09:     elseif (r < 0.5)
10:        $V_{current}(i)$  =  $Vec_{or}(i)$ ;
11:     else
12:        $V_{current}(i)$  =  $V_{current}(i)$ 
13:     end if
14:   end for
15: else if %— use “NOT” Bitwise—
16:   for i=1 to dim
17:     if (rand > 0.5)
18:        $V_{current}(i)$  = BitwiseNOT ( $V_{current}(i)$ )
19:     end if
20:   end for
21: end if

```

5.5. Fitness Function

According to Equation (43), the fitness function in the feature selection problem can be a combination of reducing classifier error and reducing the number of selected features. The fitness function states that the number of selected features should be minimized so that the accuracy of the classification algorithm increases. The fitness function is described in detail in Equation (43).

$$Fitness = \alpha \times (1 - Classifier_{acc}) + (1 - \alpha) \times \frac{fs}{S} \quad (43)$$

In Equation (43), $Classifier_{acc}$ refers to the accuracy of a classification algorithm such as KNN. The α parameter represents the importance of classification accuracy, and $(1 - \alpha)$ shows the importance of the number of selected features. F_s refers to the number of selected features, and S is the total number of features in a dataset.

6. Results and Evaluation

In this section, the proposed algorithm and other meta-heuristic algorithms are evaluated. This section’s tests have been performed in a MATLAB software environment and on a system with five processors and 6 GB of RAM. The value of α in Equation (43) is equal to 99.0, and the classifier is also set in the KNN classification. Comparative algorithms include BBA [38], V-shaped BPSO [14], BGWO [39], BCCSA [20], and BFFA [29]. The proposed algorithm and other algorithms on 30 data sets are in Table 1.

Table 1. Different types of data sets have been used in this paper.

ID	Type (Size)	Name	No. of Instances	No. of Features	No. of Classes	Area
Data1	low	abalone	4177	9	28	Life
Data2		breastcancerw	699	10	2	Biology
Data3		tictactoe	958	10	2	Game
Data4		glass	214	11	6	Physical
Data5		heart	270	14	2	Life
Data6		wine	178	14	3	Chemistry
Data7	medium	letterrecognition	20,000	17	26	computer
Data8		seismicbumps	2584	19	2	-
Data9		hepatitis	155	20	2	Life
Data10		waveform	5000	22	3	Physical
Data11		spect	267	23	2	Life
Data12		german	1000	25	2	Financial
Data13		breastEW	569	31	2	Biology
Data14		Steel	1941	34	2	Physical
Data15		Dermatology	366	35	6	Biology
Data16		ionosphere	351	35	2	Physical
Data17		soybean	307	36	19	Life
Data18		krvskp	3195	37	2	Game
Data19		lungcancer	32	57	2	Life
Data20		spambase	4601	58	2	computer
Data21	high	sonar	208	61	2	Physical
Data22		audiology	199	71	24	Life
Data23		libras	360	91	15	-
Data24		LSVT	125	311	2	Life
Data25		PersonGaitDataSet	48	322	16	Computer
Data26		pd_speech	756	755	2	Computer
Data27		ORL	400	1025	40	image
Data28		warppie	210	2421	10	image
Data29		lung	203	3313	5	voice
Data30		SMK-CAN-187	187	19,994	2	Biology

6.1. Data Set

This phase will introduce the data set taken from the UCI source. This paper considered different data sets regarding the number of features, classes, and samples. In this paper, we divide the data set into three categories: small dimensions (less than 15) and medium (between 15 and 50 features), and large (with more than 50 features). A list of 24 different datasets with the total number of samples, number of features, and number of classes is given in Table 1.

6.2. Setting Parameters

This section sets the basic parameters of the proposed algorithms and five other well-known binary meta-heuristic algorithms. In all experiments, the population number of the proposed algorithm and different algorithms is set to 10 and the number of iterations to 100, unless, for a specific experiment, the number is stated separately in the same section. Table 2 shows the initial parameter value in comparative algorithms.

Table 2. Initial value of parameters in comparative algorithms.

Algorithm	Parameter	Value
BBA [40]	A	0.5
	r	0.5
	Qmin	0
	Qmax	2
	C1	2.05
BPSO [14]	C2	2.05
	W	2
BGWO [15]	-	-
BDA [41]	-	-
BCCSA [20]	AP	0.1
	f1	2
	W	1
BFFAG [29]	Q	0.7
	R	0.9
BAVOAV (hyper-heuristic)	p1	0.6
	p2	0.4
BAVOAS (hyper-heuristic)	p3	0.6
	alpha	0.8
BAVOA (multi-strategy)	betha	0.2
	gamma	0.25

Table 3 shows that the three proposed approaches—BAOVAH-S, BAOVAH-V, and BAVOA-v1—have the same parameters. First, BAOVAS and BAOVAV versions are compared in eight different versions, and then one will be selected as the version based on transfer functions.

Table 3. Comparison of proposed BAVOA methods based on S-shaped and V-shaped based on the criterion of the average number of features.

Datasets	BAOVAH-S1	BAOVAH-S2	BAOVAH-S3	BAOVAH-S4	BAOVAH-V1	BAOVAH-V2	BAOVAH-V3	BAOVAH-V4
Data1	5	3.5	3.0	3.2	3.7	3.5	5.5	5.3
Data2	4.6	5.2	4.6	4.2	4	4.1	4.5	5.8
Data3	3.1	4.4	4.9	2.4	4.2	4.2	4.5	4.6
Data4	8.6	8.3	8.3	8.6	4.2	4.7	4.1	3.6
Data5	6.6	8.5	6.3	10.4	5.8	4.8	5.6	6.1
Data6	4.4	7.2	5.7	7.2	7.5	6.3	6.8	7.5
Data7	6.3	61	5.5	7.7	9.6	7.2	10.4	11.3
Data8	14.9	16.2	16.1	16.5	8.9	6.7	7.3	6.7
Data9	9.3	11.1	16.5	11.6	9.4	10.1	7.5	6.1
Data10	10.3	8.1	6.2	5.6	9.3	10.8	12.9	15.2
Data11	13.4	12.5	18.7	18.2	7.7	8.1	8.4	10.2
Data12	16.4	9.2	13.1	11.4	11.3	7.1	13.1	15.3
Data13	16.2	15.1	12.3	13.4	11.2	12.3	10.2	14.1
Data14	19.1	17.5	16.2	17.5	16.4	14.1	16.3	16.2
Data15	14.3	17.2	12.1	11.5	14.1	15.1	21.3	20.4
Data16	23.1	25.3	27.5	25.3	9.2	13.1	14.1	15.3
Data17	12.4	10.2	17.3	13.4	15.3	18.4	18.3	18.5
Data18	15.5	15.4	17.6	18.8	19.3	18.6	16.7	13.2
Data19	45.5	28.2	39.7	36.1	26.2	19.7	19.1	22.9
Data20	22.1	30.9	19.5	19.1	20.8	28.9	28.4	41.5

Table 3. *Cont.*

Datasets	BAOVAH-S1	BAOVAH-S2	BAOVAH-S3	BAOVAH-S4	BAOVAH-V1	BAOVAH-V2	BAOVAH-V3	BAOVAH-V4
Data21	34.9	31.8	28.4	33.2	28.1	14.5	35.4	38.1
Data22	31.5	28.4	35.2	33.4	35.1	30.8	36.3	29.8
Data23	61.5	43.2	54.3	58.8	41.1	40.1	47.8	66.5
Data24	202.8	163.1	153.7	151.4	133	134.8	163.5	133
Data25	224.9	189.5	223.7	197.1	93	120.5	95.6	93
Data26	212.7	349.5	404.5	335.6	337.8	297.5	312.8	444.6
Data27	561.8	660.8	669.9	719.5	552.8	546.4	438.2	422.1
Data28	2047.2	1648.8	1620.3	1732.5	996.7	871.5	1081.2	1498.6
Data29	2791.6	2077.6	2291.7	2831.7	1319.7	1337.4	1491.4	1251.6
Data30	16324.4	14680.2	14373.9	15590.2	7649.3	8405.6	8426.2	7665.9
Rank-low	30 01	30 01	30 02	30 06	30 04	30 05	30 03	30 12
Rank-mid	30 01	30 01	30 02	30 06	30 04	30 05	30 03	30 12
Rank-high	30 01	30 01	30 02	30 06	30 04	30 05	30 03	30 12
Ranking all	30 01	30 01	30 02	30 06	30 04	30 05	30 03	30 12

6.3. Evaluating the Three Proposed Methods of BAOVAH-S and BAOVAH-V

This section compares BAOVAH-S1, BAOVAH-S2, BAOVAH-S3, and BAOVAH-S4 with BAOVAH-V1, BAOVAH-V2, BAOVAH-V3, and BAOVAH-V4. Then, an approach is selected as the final approach based on the transfer functions. In these experiments, the iteration number was set to 100, and the population was assigned to 20. Moreover, all methods based on S-shaped and V-shaped are evaluated based on the criteria of an average number of selected features, classification accuracy, and fitness function. The first experiment evaluates the BAOVAH-S1, BAOVAH-S2, BAOVAH-S3, and BAOVAH-S4 approaches with BAOVAH-V1, BAOVAH-V2, BAOVAH-V3, and BAOVAH-V4 in terms of the number of features selected. The results of this experiment are shown in Table 3.

According to Table 3, V-shaped methods' performance is much more successful and efficient in identifying valuable and essential features. With the BAOVAS4-based method, the results are more effective and efficient than other methods based on S-shaped. Moreover, BAOVAS2 and BAOVAS3 methods provided tolerable performance among the 30 datasets case study. The weakest result is in the S-shaped method related to BAOVAS1.

In the V-shaped-based method, the BAOVAV4 algorithm obtains the most successful results. BAOVA-V1, BAOVA-V2, and BAOVA-V3 algorithms also incur average results in the average number of selected features. The following results of S-shaped- and V-shaped-based algorithms based on the average accuracy evaluation criterion are reviewed. The results of this experiment are shown in Table 4. In the following, we will analyze and compare the methods based on S-shaped- and V-shaped-based methods in terms of average accuracy.

Table 4. Comparison of proposed BAOVA methods based on S-shaped and V-shaped based on average criteria.

[illegible]

Table 4. Cont.

Datasets	BAOVAH-S1	BAOVAH-S2	BAOVAH-S3	BAOVAH-S4	BAOVAH-V1	BAOVAH-V2	BAOVAH-V3	BAOVAH-V4
Data9	0.716	0.745	0.708	0.729	0.729	0.729	0.755	0.729
Data10	0.775	0.78	0.778	0.781	0.785	0.778	0.794	0.797
Data11	0.766	0.765	0.782	0.782	0.782	0.774	0.768	0.773
Data12	0.715	0.72	0.734	0.721	0.724	0.719	0.73	0.725
Data13	0.944	0.946	0.944	0.947	0.952	0.952	0.946	0.967
Data14	0.998	0.998	0.997	0.998	0.999	1	0.999	1
Data15	0.965	0.965	0.977	0.965	0.976	0.979	0.979	0.985
Data16	0.91	0.922	0.917	0.927	0.929	0.935	0.911	0.935
Data17	0.905	0.912	0.912	0.925	0.925	0.931	0.944	0.924
Data18	0.953	0.954	0.965	0.953	0.978	0.975	0.975	0.976
Data19	0.94	0.94	0.94	0.94	0.94	0.94	1	1
Data20	0.913	0.91	0.906	0.915	0.91	0.909	0.923	0.917
Data21	0.906	0.906	0.887	0.906	0.906	0.945	0.925	0.902
Data22	0.74	0.75	0.71	0.75	0.8	0.78	0.79	0.79
Data23	0.814	0.819	0.841	0.824	0.824	0.824	0.835	0.83
Data24	0.827	0.858	0.858	0.843	0.843	0.875	0.859	0.875
Data25	0.627	0.627	0.669	0.627	0.752	0.669	0.585	0.705
Data26	0.879	0.887	0.89	0.886	0.888	0.888	0.89	0.896
Data27	0.907	0.913	0.913	0.917	0.917	0.926	0.917	0.926
Data28	0.915	0.926	0.926	0.926	0.932	0.926	0.944	0.944
Data29	0.962	0.972	0.972	0.962	0.972	0.98	0.972	0.972
Data30	0.629	0.619	0.626	0.619	0.637	0.619	0.637	0.639
Rank-low	30 05	30 07	30 07	30 07	30 08	30 13	30 12	30 18
Rank-mid	30 05	30 07	30 07	30 07	30 08	30 13	30 12	30 18
Rank-high	30 05	30 07	30 07	30 07	30 08	30 13	30 12	30 18
Ranking all	30 05	30 07	30 07	30 07	30 08	30 13	30 12	30 18

According to Table 4, it can be seen that the performance of V-shaped-based methods is much more efficient and better in obtaining the values of the fitness function. BAOVAS4-based methods have achieved more successful results than other S-Shaped-based methods. BAOVAS2 and S3-BAOVA methods also provided acceptable performance among the 30 datasets studied. The weakest effect in the average fitness function is also related to BAOVAS1 in the S-shaped method. The most successful results in the BAOVAV4 algorithm are in the V-shaped-based method. BAOVAV1 and BAOVAV2 algorithms have also achieved relatively weak results. As a result, the experiments prove the superiority of BAOVAV4 in terms of average selected features, accuracy, and fitness function. Figures 8–12 show the degree of convergence of each method based on S-shaped and V-shaped.

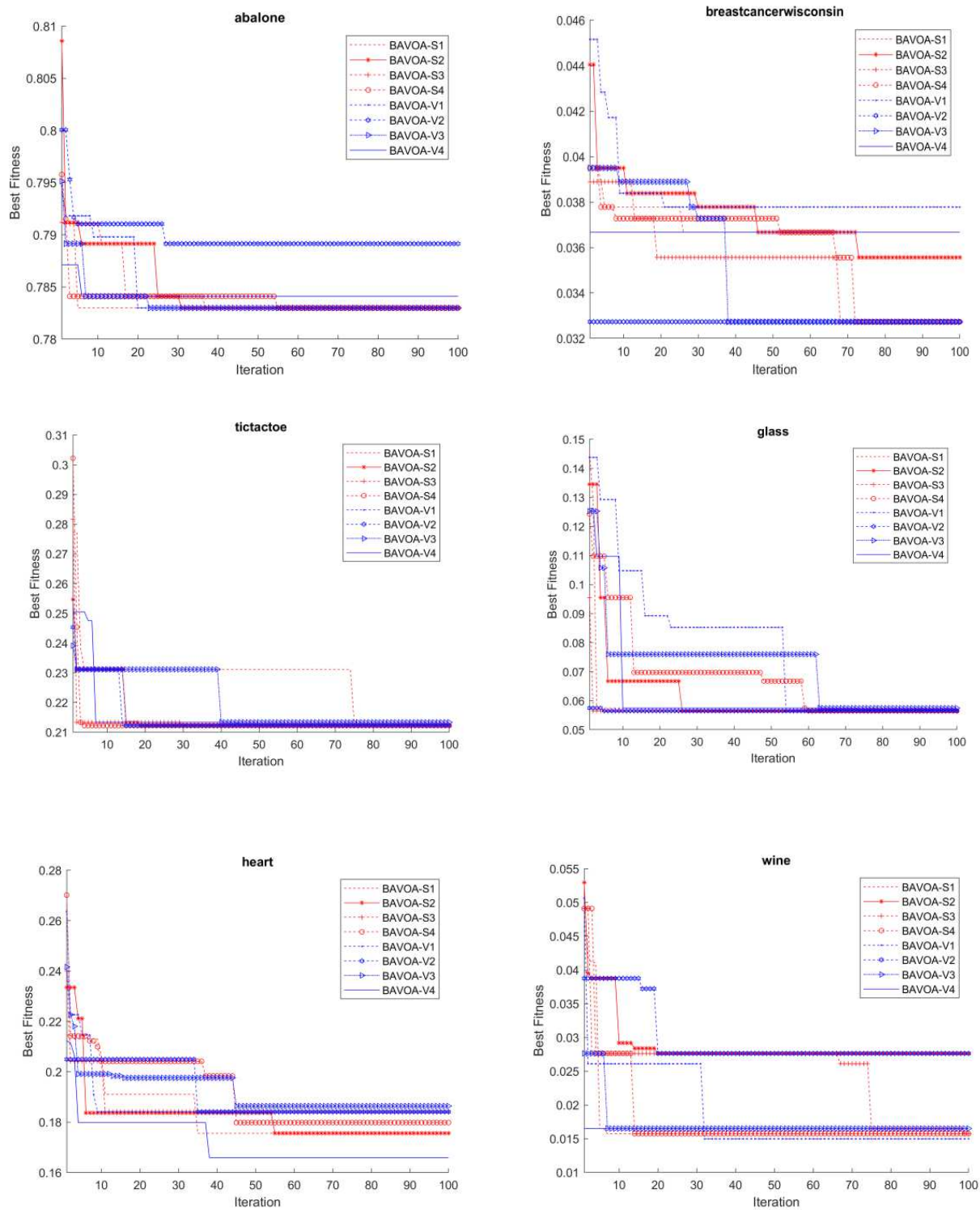


Figure 10. Comparison of the convergence of S-shaped- and V-shaped-based methods on the first six sets.

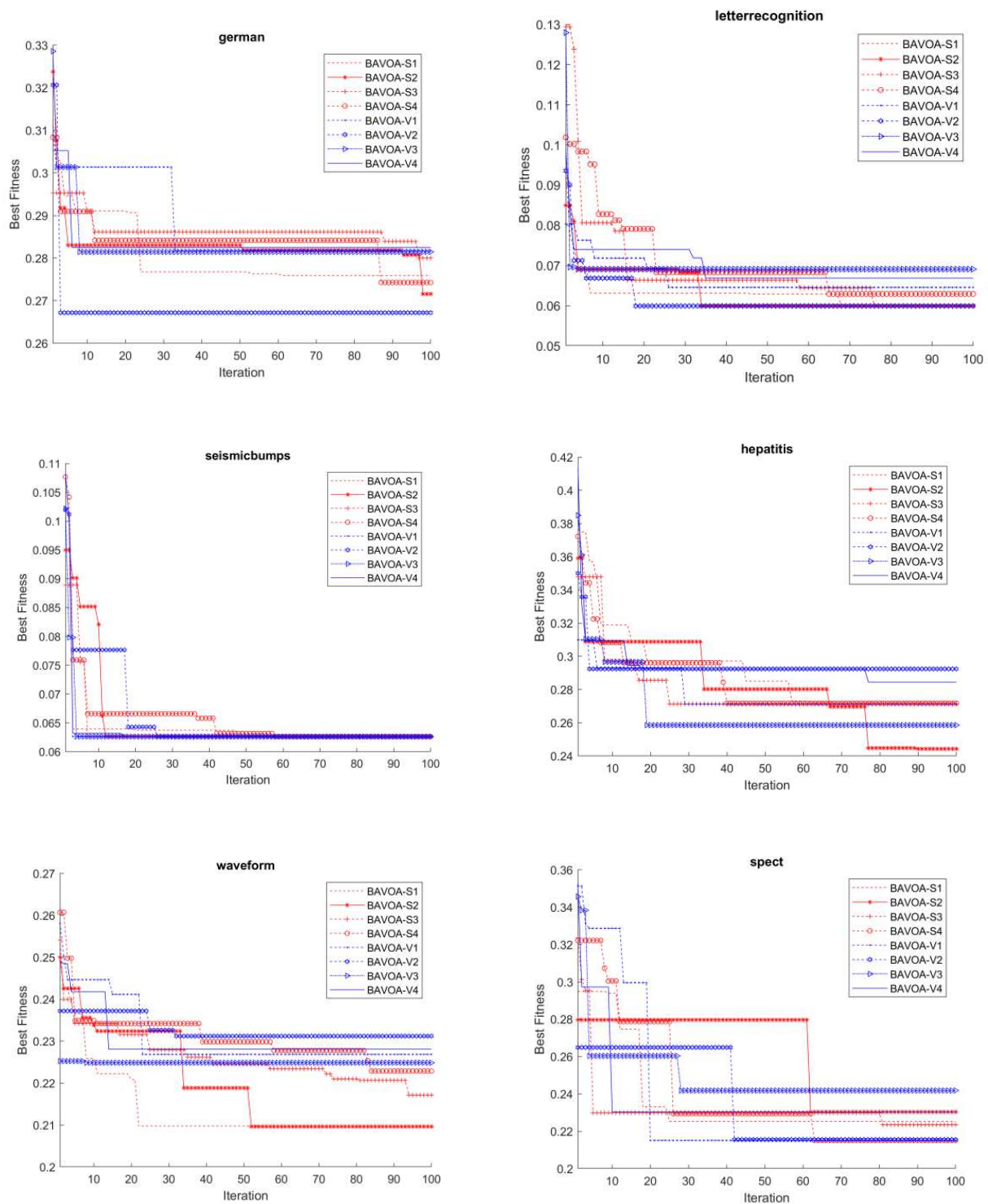


Figure 11. Comparison of the convergence of S-shaped- and V-shaped-based methods on the second 6 data sets.

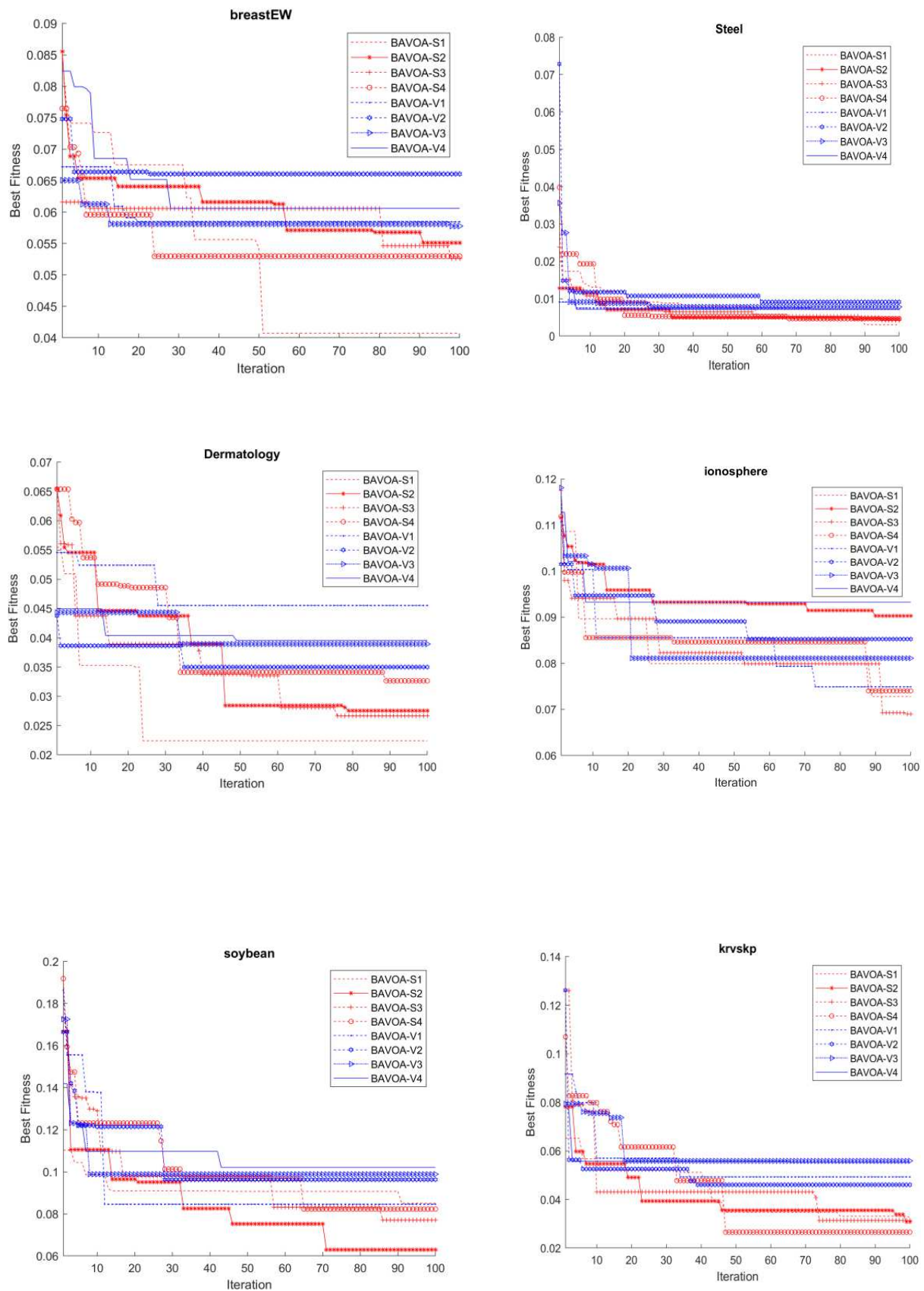


Figure 12. Comparison of the convergence of S-shaped- and V-shaped-based methods on the third six data sets.

It can be seen from Table 5 that the performance of V-shaped-based methods is much more efficient and successful in obtaining the average accuracy. BAOVAS1 and BAOVAS4-based methods have achieved more successful results than other S-shaped-based methods. The most successful results in the BAOVAV4 algorithm are in the V-shaped-based method. BAOVAV1 and BAOVAV2 algorithms also achieved relatively weak effects. The test results prove the superiority of the BAOVA-4 in terms of the mean of the selected features, the mean accuracy, and the mean of the fitness function. Figures 10–14 show the degree of convergence of the S-shaped- and V-shaped-based methods.

Table 5. Comparison of proposed BAOVA methods based on Z-shaped and V-shaped based on the average objective function criterion.

Datasets	BAOVAH-S1	BAOVAH-S2	BAOVAH-S3	BAOVAH-S4	BAOVAH-V1	BAOVAH-V2	BAOVAH-V3	BAOVAH-V4
Data1	0.805	0.81	0.81	0.811	0.809	10.724	0.809	0.801
Data2	0.069	0.067	0.073	0.06	0.067	0.096	0.072	0.044
Data3	20.321	0.389	0.37	0.925	0.377	0.31	0.985	10.247
Data4	0.279	0.273	0.319	0.288	0.254	20.227	0.326	0.22
Data5	0.918	0.307	0.316	0.294	0.963	0.302	0.948	0.952
Data6	0.153	0.108	0.264	0.117	0.071	0.086	0.102	0.058
Data7	0.404	0.428	0.576	0.402	0.131	0.327	0.148	0.131
Data8	0.122	0.114	0.116	0.122	0.113	0.984	0.91	0.106
Data9	0.442	0.468	0.444	0.467	0.417	20.347	0.389	0.954
Data10	0.339	0.354	0.392	0.415	0.322	0.302	0.265	0.262
Data11	0.459	0.409	0.406	0.417	0.405	0.364	0.383	0.932
Data12	0.353	0.364	0.373	0.362	0.335	0.307	0.349	0.323
Data13	0.11	0.108	0.115	0.119	0.107	0.114	0.204	0.101
Data14	0.216	0.205	0.186	0.275	0.163	0.175	0.102	0.095
Data15	0.195	0.166	0.22	0.304	0.223	0.138	0.098	0.107
Data16	0.129	0.127	0.129	0.125	0.133	0.133	0.901	0.12
Data17	0.4659	0.385	0.301	0.352	0.215	0.224	0.278	0.258
Data18	0.319	0.33	0.332	0.316	0.208	0.243	0.203	0.502
Data19	0.321	0.422	0.363	0.389	0.312	10.292	0.295	0.182
Data20	0.148	0.135	0.146	0.165	0.139	0.132	0.144	0.108
Data21	0.209	0.216	0.252	0.224	0.179	0.246	0.179	0.164
Data22	0.639	0.612	0.555	0.598	0.553	0.482	0.453	0.568
Data23	0.219	0.237	0.232	0.231	0.213	0.228	0.211	0.205
Data24	0.321	0.348	0.337	0.392	0.308	0.262	0.257	0.388
Data25	0.69	0.722	0.723	0.704	0.618	0.616	0.672	0.584
Data26	0.167	0.169	0.158	0.169	0.157	0.165	0.156	0.148
Data27	0.145	0.131	0.139	0.136	0.135	0.122	0.121	0.121
Data28	0.102	0.102	0.102	0.102	0.094	0.116	0.097	0.101
Data29	0.062	0.053	0.053	0.056	0.052	0.052	0.055	0.051
Data30	0.453	0.455	0.452	0.445	0.421	0.422	0.433	0.416
Rank-low	30 05	30 07	30 07	30 07	30 08	30 13	30 12	30 18
Rank-mid	30 05	30 07	30 07	30 07	30 08	30 13	30 12	30 18
Rank-high	30 05	30 07	30 07	30 07	30 08	30 13	30 12	30 18
Ranking all	30 05	30 07	30 07	30 07	30 08	30 13	30 12	30 18

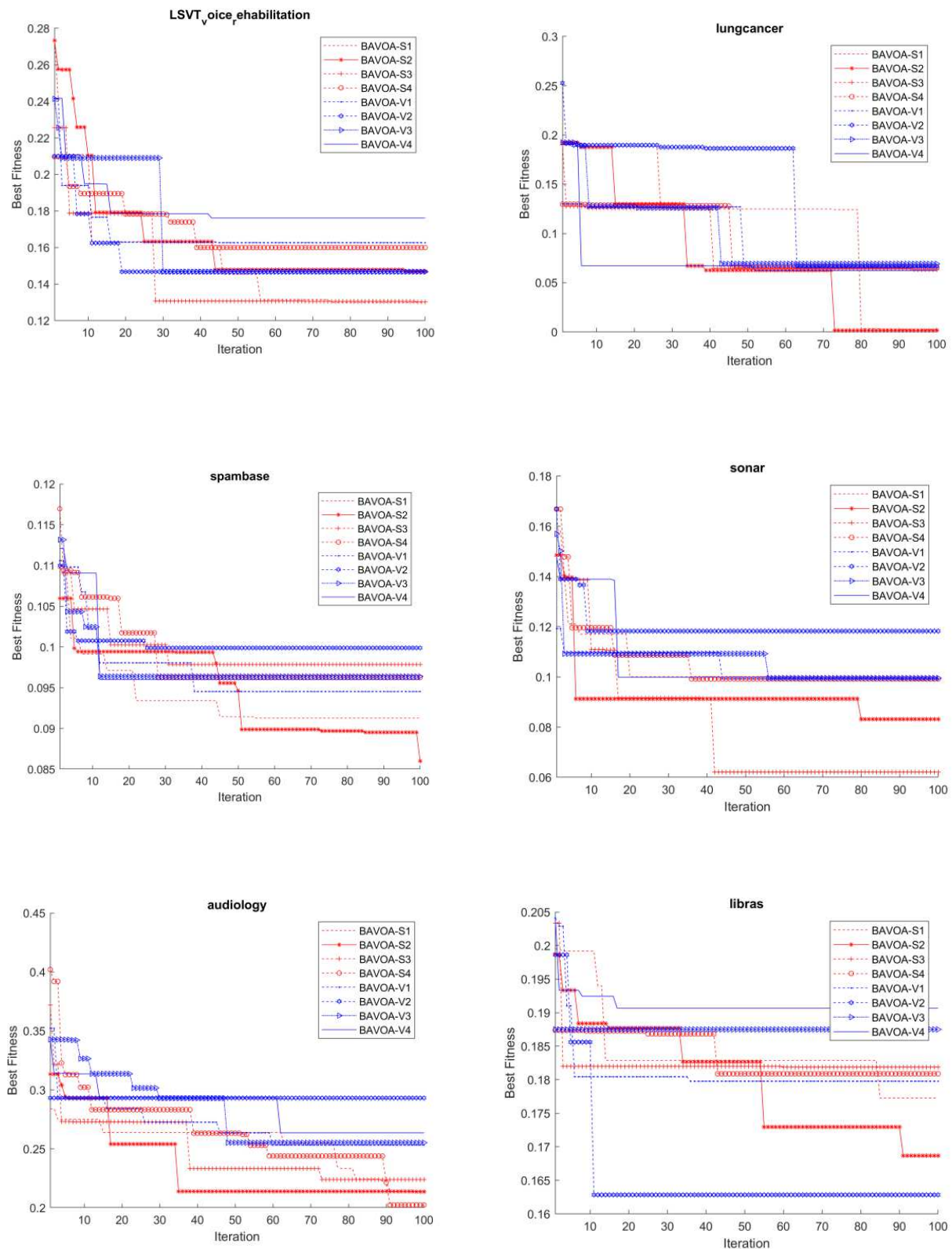


Figure 13. Comparison of the convergence of S-shaped- and V-shaped-based methods on the fourth six data sets.

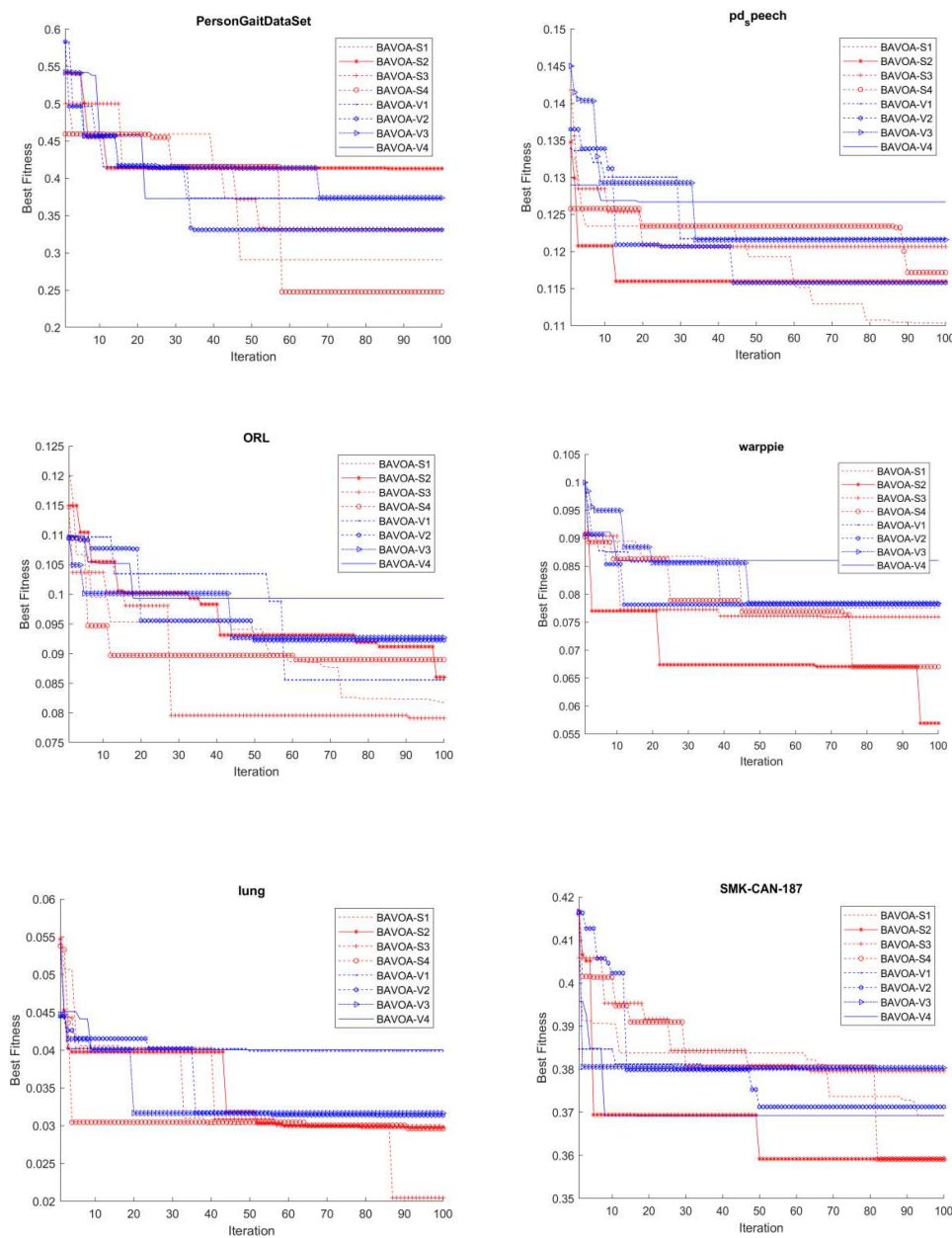


Figure 14. Comparison of the convergence of S-shaped- and V-shaped-based methods on the fifth six data sets.

6.4. Evaluating the BAVOA-v1 Approach

In this subsection, we aim to evaluate the proposed approaches, BAOVAH (hyper-heuristic) and BAVOA-Version1 (multi-strategy). The BAOVAH method has been selected based on Section 5.3 as a transfer function-based method among eight transfer functions according to its results. This section will compare these two proposed methods with the basic algorithms BBA, BPSO, BGWO, and new algorithms such as BDA, BCCSA, and BCCSA regarding different statistical criteria. The fitness function specified in Equation (10) states that the proposed approach and other algorithms should consider feature selection and accuracy objectives. However, 99% of the scale is set on the accuracy of the classification. Therefore, in the first experiment, the average accuracy of the algorithms and the proposed approach on 30 data sets are presented in Table 6.

Table 6. Comparison of the proposed BAOVAH (hyper-heuristic) and BAVOA-V1 (multi-strategy) approach based on the average accuracy criterion.

	BBA	BPSO	BGWO	BDA	BCCSA	BFFAG	BAVOA-V1 (Multi-Strategy)	BAVOAH
Data1	0.174	0.174	0.206	0.213	0.206	0.207	0.181	0.207
Data2	0.934	0.934	0.965	0.97	0.965	0.574	0.927	0.972
Data3	0.524	0.524	0.783	0.785	0.777	0.588	0.607	0.787
Data4	0.736	0.736	0.871	0.924	0.944	0.544	0.716	0.878
Data5	0.675	0.675	0.804	0.807	0.807	0.52	0.673	0.824
Data6	0.874	0.874	0.969	0.958	0.966	0.526	0.860	0.954
Data7	0.559	0.559	0.942	0.944	0.931	0.547	0.945	0.945
Data8	0.904	0.904	0.927	0.937	0.937	0.547	0.934	0.934
Data9	0.576	0.576	0.729	0.754	0.685	0.524	0.732	0.739
Data10	0.672	0.672	0.793	0.785	0.756	0.516	0.797	0.797
Data11	0.57	0.57	0.731	0.75	0.769	0.546	0.771	0.694
Data12	0.65	0.65	0.715	0.715	0.696	0.516	0.624	0.742
Data13	0.901	0.901	0.958	0.951	0.933	0.532	0.876	0.945
Data14	0.737	0.737	0.997	0.991	0.982	0.532	0.790	0.992
Data15	0.856	0.856	0.977	0.976	0.945	0.529	0.845	0.955
Data16	0.878	0.878	0.904	0.903	0.909	0.527	0.881	0.935
Data17	0.647	0.647	0.942	0.942	0.87	0.523	0.686	0.946
Data18	0.696	0.696	0.957	0.921	0.914	0.51	0.638	0.922
Data19	0.703	0.703	0.928	0.984	0.875	0.576	0.583	0.929
Data20	0.848	0.848	0.91	0.9	0.899	0.566	0.835	0.920
Data21	0.797	0.797	0.917	0.927	0.865	0.553	0.778	0.876
Data22	0.494	0.494	0.784	0.831	0.68	0.528	0.792	0.833
Data23	0.781	0.781	0.825	0.825	0.825	0.536	0.786	0.786
Data24	0.707	0.707	0.817	0.854	0.841	0.554	0.664	0.900
Data25	0.338	0.338	0.54	0.679	0.583	0.57	0.672	0.550
Data26	0.841	0.841	0.914	0.875	0.876	0.57	0.835	0.903
Data27	0.874	0.874	0.922	0.901	0.91	0.584	0.864	0.924
Data28	0.901	0.901	0.923	0.926	0.926	0.926	0.926	0.926
Data29	0.951	0.951	0.97	0.96	0.971	0.565	0.960	0.971
Data30	0.56	0.56	0.624	0.624	0.617	0.575	0.599	0.636
Rank-low	06 00	06 00	06 01	06 02	06 01	06 00	06 00	06 04
Rank-mid	12 00	12 00	12 02	12 04	12 00	12 00	12 04	12 07
Rank-high	12 00	12 00	12 01	12 04	12 02	12 00	12 01	12 09
Ranking all	30 00	30 07	30 04	30 07	30 03	30 13	30 05	30 20

Comparing the mean accuracy of all methods in Table 6 shows that the BAOVAH method is more accurate than comparative algorithms out of 30 data sets in 20 sets, which is 67% of the data set has obtained the highest accuracy. Moreover, in the other 33%, it has shown perfect accuracy. Besides this, the BAVOA-V1 approach has shown relatively weak performance compared to the BAOVAH approach and different algorithms due to its easy use of the transfer function. After examining the algorithms' average accuracy, the features' average is considered in the following: in the fitness function, only 01% of the number of features are considered. Evaluation of the proposed approach and other algorithms in terms of the average number of selected features is given in Table 7.

Table 7. Comparison of proposed BAOVAH (hyper-heuristic) and BAVOA-V1 (Multi-strategy), an approach based on the criterion of the average number of features.

	BBA	BPSO	BGWO	BDA	BCCSA	BFFAG	BAVOA-V1 (Multi-Strategy)	BAVOAH
Data1	2.5	4.56	5.65	4.9	2.4	6.8	3.536	2.3
Data2	4.3	6.30	5.45	4	3.2	7.6	4.754	3.9
Data3	2.6	5.28	5.7	5.05	9	8.3	4.5	6.14
Data4	5	7.50	8.4	1	1	8.15	8.729	3.64
Data5	6.15	8.52	10.4	6.95	8	11.65	5.534	5.534
Data6	4.35	7.80	7.9	4.05	5	10.95	7.4	7.2
Data7	5.25	6.56	13.1	11	16	13.9	11.57	4.5
Data8	4.9	10.25	6.05	1	1	14.6	15.799	6.63
Data9	6.4	10.58	12.2	8.05	3	16.55	12.384	6.38
Data10	7.2	10.36	17.65	15.1	21	18.55	15.97	6.912
Data11	6.45	2	16.9	6.3	4	19.15	19.599	12.24
Data12	10.15	11.74	17.6	13.7	4	21.05	9.621	9.59
Data13	12.45	20.30	15.8	11.05	11.5	20.3	14.67	11.48
Data14	9.15	7.5	21.3	7.5	33	26.75	22.69	14.12
Data15	14.2	18.5	24	10.65	18	29.1	16.57	18.15
Data16	13.5	17.75	17.75	14.2	6	28.65	29.46	18.10
Data17	10.35	32	27.05	16.95	35	30.5	14.78	24.09
Data18	15.2	20.3	25.8	13.35	27	30.1	18.921	20.21
Data19	18.4	29.12	31.2	12.7	13	25.35	29.08	45.79
Data20	20.5	26.95	40.95	39.2	57	50.4	20.47	20.46
Data21	19.75	37.65	41.05	23.95	19	53.05	30.394	18.60
Data22	24.3	26.3	52.35	26.3	25	37.6	32.254	59.45
Data23	30.55	31.28	54.75	39.7	38	54.8	26.695	77.12
Data24	118.45	191.5	223.35	147.25	151	272	161.622	111.49
Data25	119.2	159.45	203.6	122.3	123	274.15	93.01	124.43
Data26	257.4	673.4	571.85	472.85	221	673.4	436.011	345.23
Data27	366.45	715.8	715.8	689.95	273	905.45	574.222	270.047
Data28	791.65	2042.83	1297.65	1053.35	987.5	2062.45	1563.67	652.325
Data29	964.6	1	1936.75	1375.1	863	2834.1	1960.24	2876.363
Data30	6042	1	14129.25	9682.8	9894	17652.25	11639	19742
Rank-low	06 00	06 00	06 01	06 02	06 01	06 00	06 00	06 04
Rank-mid	12 00	12 00	12 02	12 04	12 00	12 00	12 04	12 07
Rank-high	12 00	12 00	12 01	12 04	12 02	12 00	12 01	12 09
Ranking all	30 00	30 07	30 04	30 07	30 03	30 13	30 05	30 20

A comparison of the average choice of method features in Table 7 shows that the BAOVAH method obtained a better average of feature selection than comparative algorithms from 30 data sets in 12 data sets. At 40%, the data set has the lowest accuracy. In the other 60%, it has shown an excellent selection. Besides this, obtaining the optimal value or the best fitness function can determine the superiority of algorithms. To further prove the proposed method, Table 8 compares the proposed BAOVAH approach based on the best value of the fitness function.

Table 8. Comparison of the proposed BAOVAH (hyper-heuristic) and BAVOA-V1 (multi-strategy) an approach based on the criterion of the best value of the fitness function.

	BBA	BPSO	BGWO	BDA	BCCSA	BFFAG	BAVOA-V1 (Multi-Strategy)	BAVOAH
Data1	0.792	0.784	0.783	0.783	0.788	0.783	0.783	0.783
Data2	0.038	0.036	0.036	0.033	0.037	0.033	0.033	0.033
Data3	0.251	0.212	0.212	0.212	0.231	0.231	0.212	0.212
Data4	0.058	0.057	0.129	0.057	0.057	0.057	0.057	0.057
Data5	0.227	0.205	0.184	0.174	0.194	0.176	0.174	0.173
Data6	0.015	0.027	0.028	0.014	0.037	0.017	0.022	0.013
Data7	0.13	0.064	0.064	0.06	0.078	0.06	0.060	0.059
Data8	0.064	0.072	0.073	0.063	0.063	0.063	0.063	0.063
Data9	0.319	0.297	0.272	0.245	0.281	0.258	0.269	0.244
Data10	0.235	0.217	0.21	0.214	0.252	0.21	0.227	0.205
Data11	0.299	0.197	0.267	0.232	0.23	0.235	0.213	0.195
Data12	0.306	0.253	0.276	0.253	0.302	0.266	0.276	0.250
Data13	0.056	0.046	0.05	0.049	0.069	0.047	0.052	0.039
Data14	0.017	0.004	0.009	0.002	0.027	0.002	0.004	0.002
Data15	0.064	0.026	0.028	0.014	0.059	0.02	0.025	0.019
Data16	0.104	0.083	0.09	0.066	0.092	0.059	0.053	0.063
Data17	0.172	0.07	0.065	0.056	0.139	0.083	0.081	0.049
Data18	0.071	0.036	0.048	0.023	0.092	0.022	0.026	0.022
Data19	0.002	0.127	0.067	0.002	0.126	0.002	0.002	0.001
Data20	0.103	0.094	0.094	0.085	0.11	0.088	0.092	0.081
Data21	0.109	0.062	0.083	0.051	0.136	0.083	0.059	0.032
Data22	0.282	0.211	0.215	0.162	0.32	0.143	0.241	0.141
Data23	0.19	0.163	0.176	0.164	0.173	0.159	0.175	0.157
Data24	0.16	0.067	0.101	0.099	0.162	0.117	0.143	0.051
Data25	0.375	0.293	0.46	0.292	0.415	0.293	0.328	0.250
Data26	0.125	0.075	0.091	0.089	0.126	0.105	0.106	0.080
Data27	0.098	0.074	0.081	0.075	0.092	0.086	0.077	0.073
Data28	0.087	0.08	0.081	0.08	0.078	0.07	0.069	0.066
Data29	0.041	0.043	0.035	0.043	0.032	0.034	0.031	0.021
Data30	0.403	0.363	0.376	0.363	0.384	0.376	0.361	0.325
Rank-low	06 00	06 00	06 01	06 02	06 01	06 00	06 00	06 04
Rank-mid	12 00	12 00	12 02	12 04	12 00	12 00	12 04	12 07
Rank-high	12 00	12 00	12 01	12 04	12 02	12 00	12 01	12 09
Ranking all	30 00	30 07	30 04	30 07	30 03	30 13	30 05	30 20

Comparing the best value of the fitness function of all methods in Table 8 shows that the BAOVAH method has obtained a better fitness function value out of 30 data sets in 28 data sets than comparative algorithms. 93% of the data set, which has the best fitness function value, and the other 7%, showed perfect accuracy. Moreover, the BAVOA-V1 approach has a relatively weak performance compared to the BAOVAH approach and different algorithms due to the easy use of the transfer function. So, statistical results based on Table 8 show the approach. The proposed method of BAOVAH can be presented as a robust algorithm in feature selection. The degree of convergence of the algorithm shows how the algorithm has maintained a balance between exploration and exploitation. The convergence results of the proposed approach and other comparative algorithms to prove the convergence of the proposed method are shown in Figures 15–19 on 30 data sets, respectively.

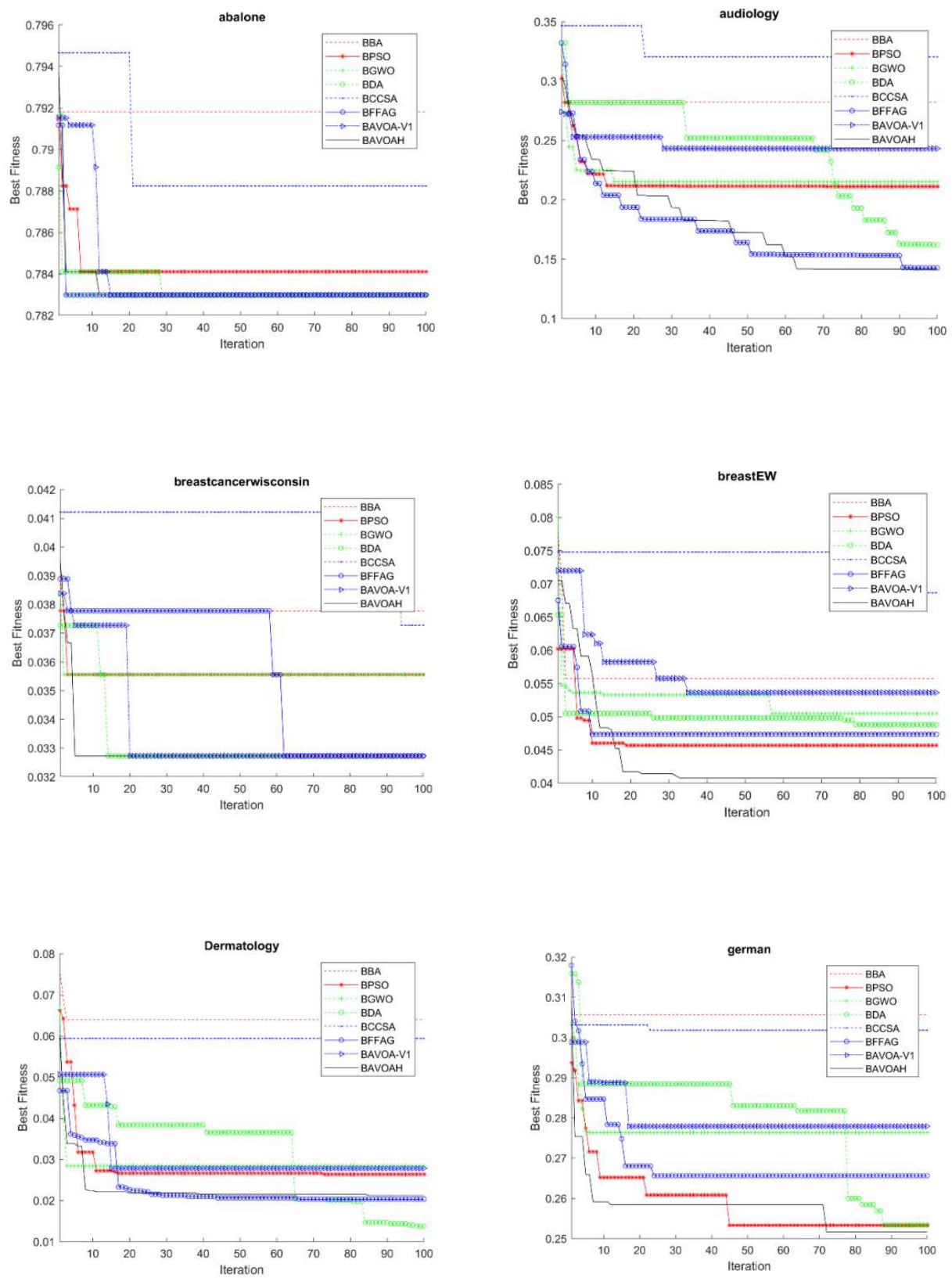


Figure 15. Comparison of the degree of convergence of the proposed BAOVAH (hyper-heuristic) and BAVOA-V1 (multi-strategy) method on the first six data sets.

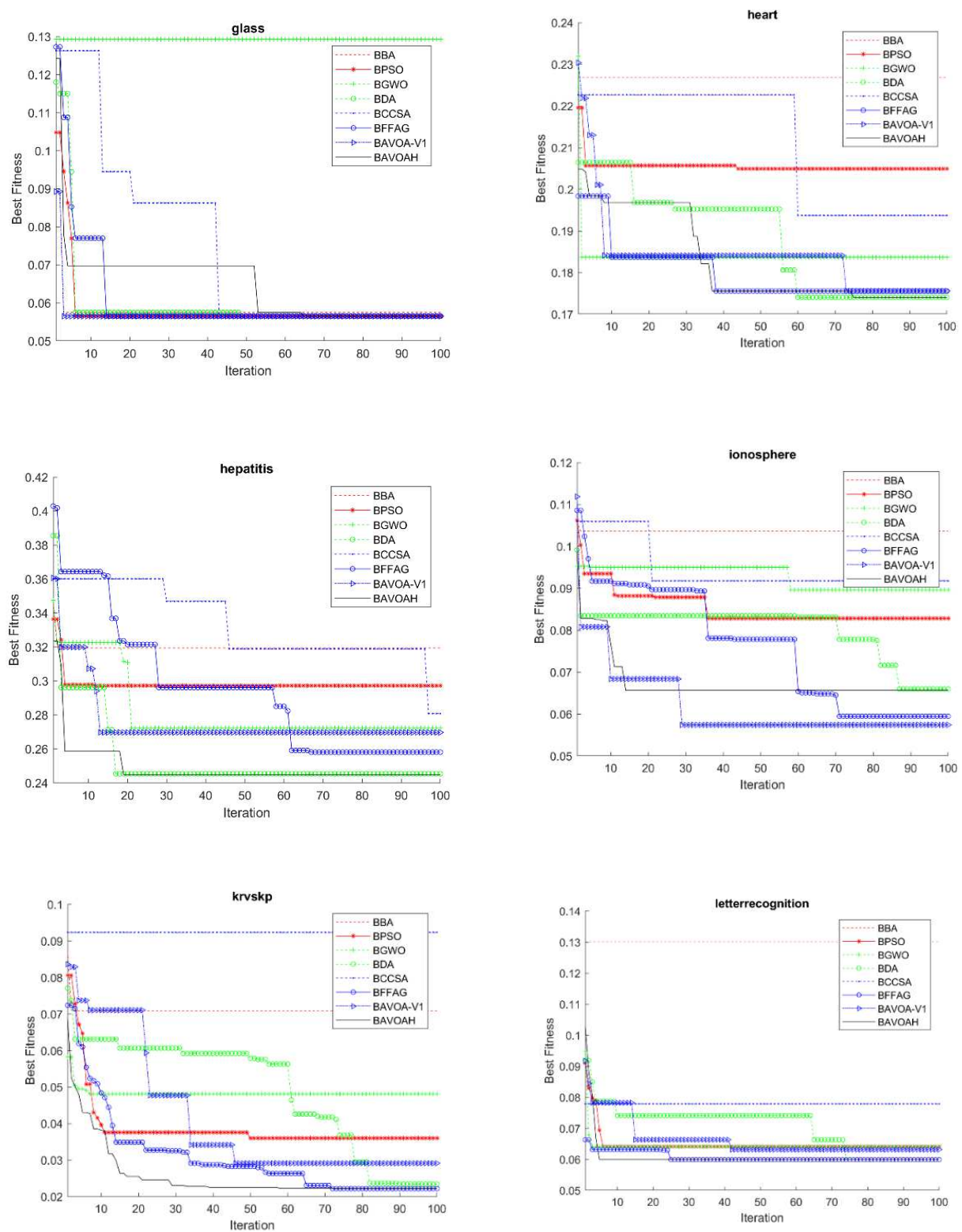


Figure 16. Comparison of the degree of convergence of the proposed BAOVAH (hyper-heuristic) and BAVOA-V1 (multi-strategy) method on the second six data sets.

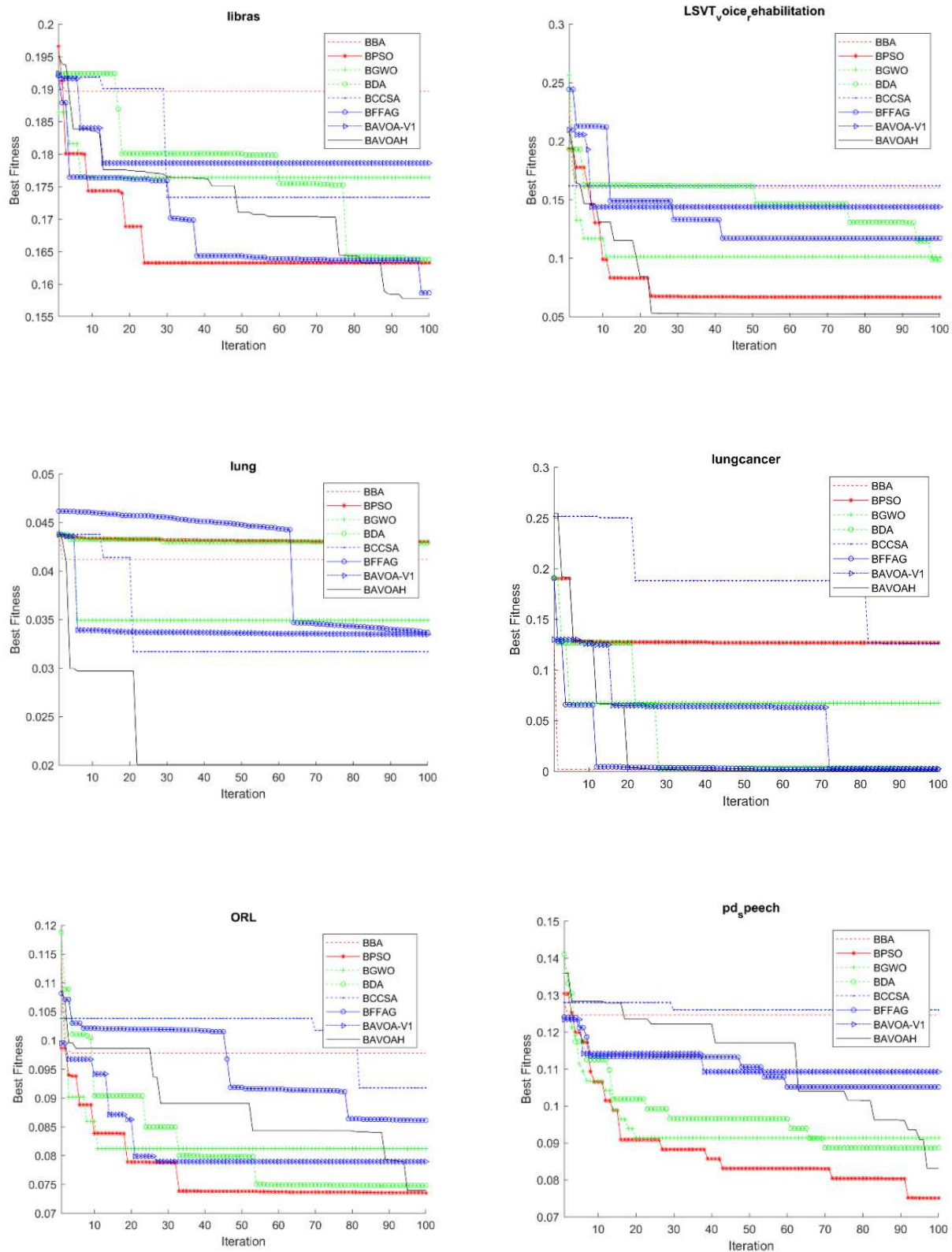


Figure 17. Comparison of the degree of convergence of the proposed BAOVAH (hyper-heuristic) and BAVOA-V1 (multi-strategy) method on the third six sets.

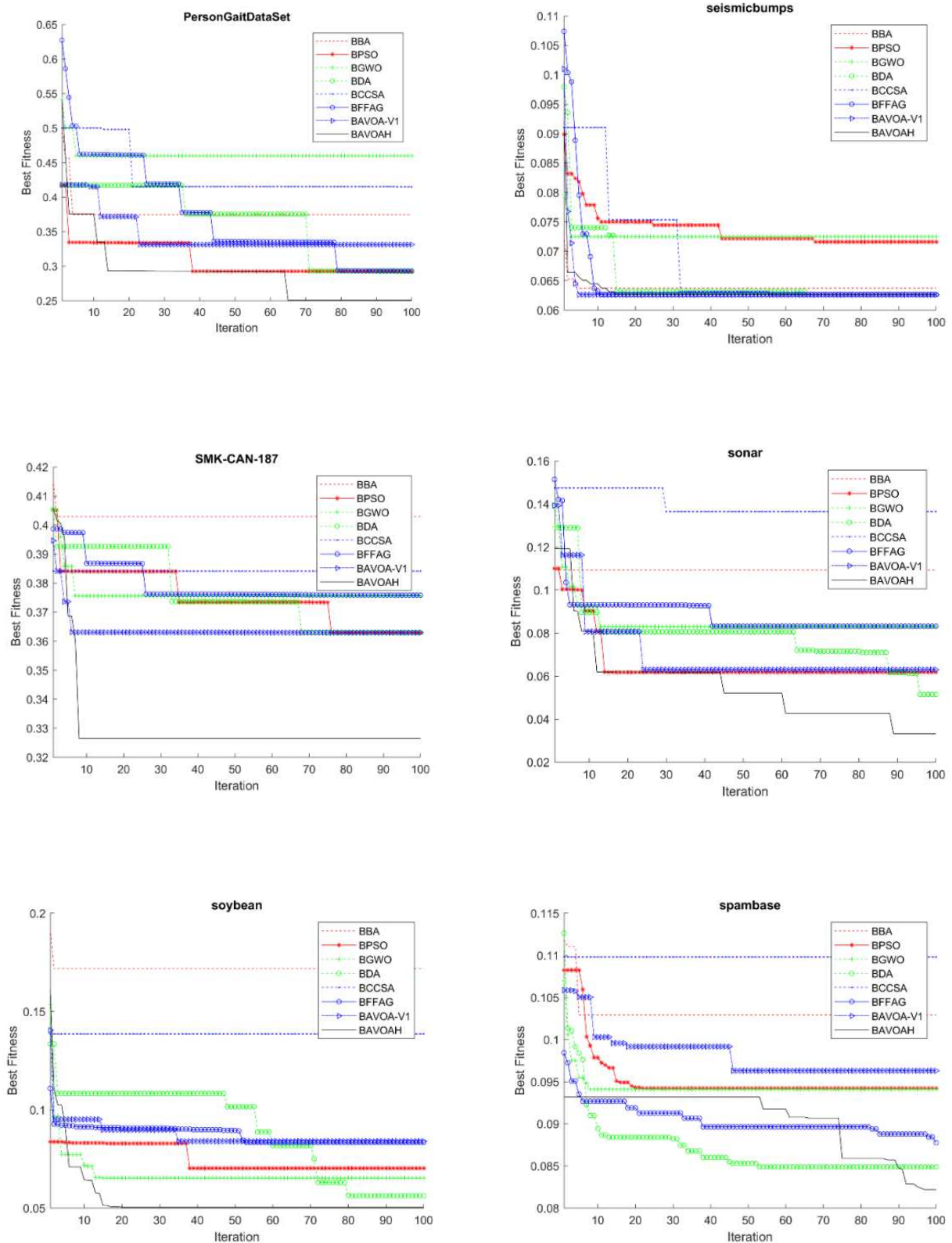


Figure 18. Comparison of the degree of convergence of the proposed BAOVAH (hyper-heuristic) and BAVOA-V1 (multi-strategy) method on the fourth six sets.

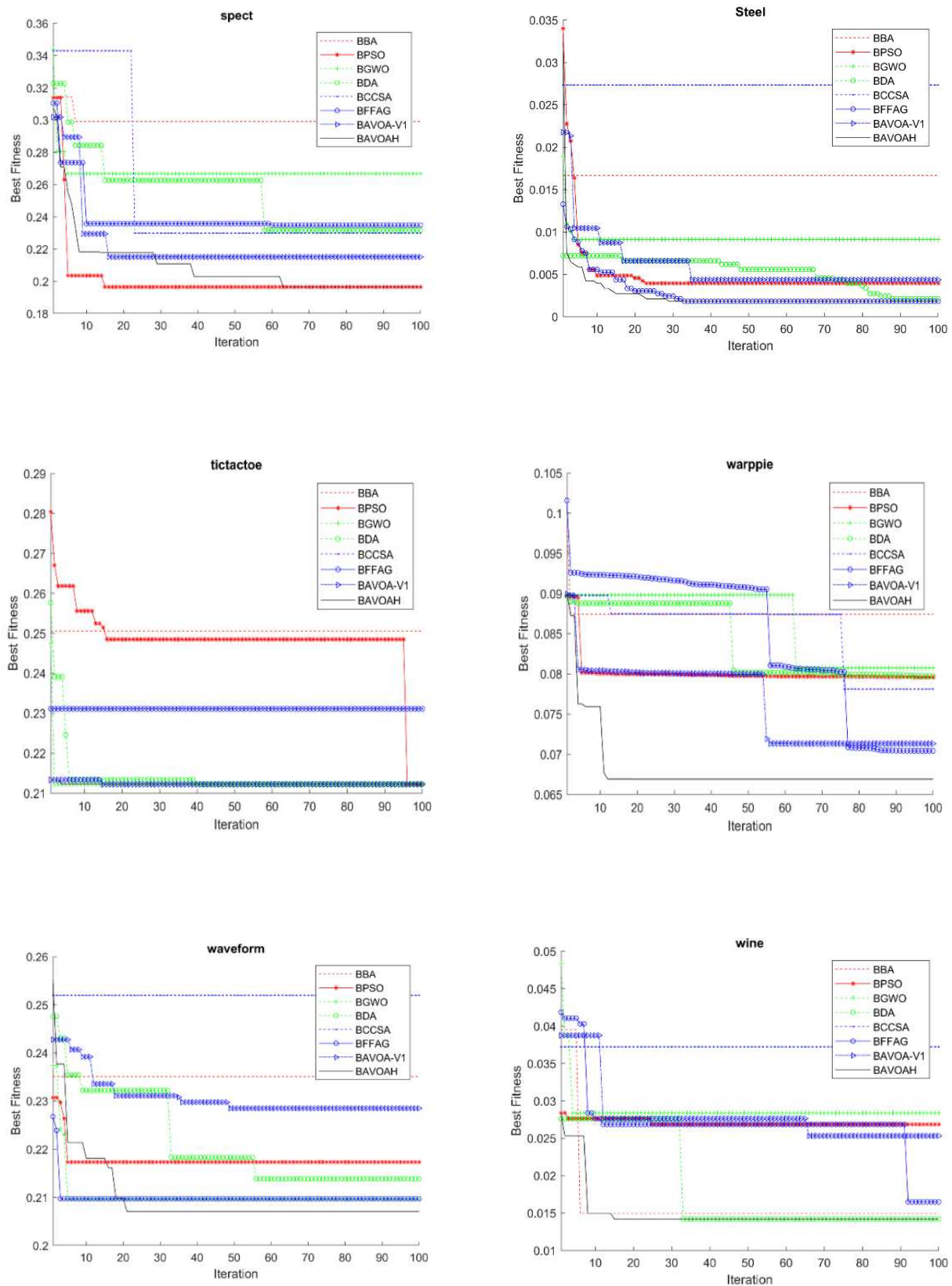


Figure 19. Comparison of the degree of convergence of the proposed BAOVAH (hyper-heuristic) and BAVOA-V1 (multi-strategy) method on the fifth six data sets.

The convergence function of the proposed method and other algorithms in the above figures shows that the BAOVAH approach has a relatively good convergence. It has been

able to maintain the balance between exploration and exploitation. Finally, in 93% of the data set, it has shown better convergence.

7. Case Study

This section aims to apply a practical study to evaluate further the BAOVAH approach in setting the parameters of deep learning algorithms in sentiment analysis. This section will combine the proposed BAOVAH approach with in-depth learning methods for sentiment analysis. In this section, BAOVAH's proposed approach for setting hyper-parameters of the deep learning algorithm, such as the number of neurons in each layer, activation function, and deep learning, has been used in sentiment analysis. This section will use the UCI Sentiment Analysis Textual Database, consisting of three different and primary sentiment analysis datasets. This dataset includes three datasets IMDB, Amazon, and Yelp. Each data set contains 1000 samples with 0 labels and 1. the results of this section have been done in the Google collab environment and with Python language. The Tensor Flow and Keras libraries have also implemented in-depth methods. We used the proposed BAOVAH approach, a binary algorithm in setting the parameters of deep learning algorithms in sentiment analysis that has advantages such as exploring and creating a variety of solutions and creating quality solutions over continuous solutions. In [42], a partial swarm quantum binary algorithm called BQPSO is used to determine the appropriate parameters of the deep convolution network. The experimental results of this paper prove that the proposed binary-based method can have better performance and stability than the traditional method. This method is more commonly known as fixed-length string binary. In [43], a binary string with a genetic algorithm is used as a three-step evolutionary process consisting of selection, synthesis, and mutation to optimize the structure of the DenseNet topology. The results of various experiments confirm the superiority of the proposed method. Moreover, the number of parameters has been significantly reduced. In [44], the BPSO is being used to optimize the parameter values of a deep CNN network. The bit string is used to encode this problem that the results of the CNN hybrid method or BPSO have a significant advantage over other methods.

7.1. Pre-Processing

Since this dataset is textual, a series of initial pre-processing is required; in this phase of the proposed method, a series of pre-processing operations, including data clearing, tokenization removing the stop words, and stemming, is being applied to any comments. In natural language processing, algorithms do not understand the text. Therefore, the first and most crucial step is to identify or separate the words (signs and words) and the unit maker's responsibility to separate their words. The next step is to remove the stop words. Stop words are repetitive words in the text that do not contain information and only connect words in a sentence [45]. Rooting is the last step that is done in the pre-processing phase. The root refers to the central meaning and concept of the word. Thus, a limited number of roots are formed in the natural language, and the rest of the words are derived from these roots [46,47]. The purpose of the root finder is to extract the root and delete the fixes attached to the word [48,49]. Hence, rooting is one of the main steps in natural language processing that must be done.

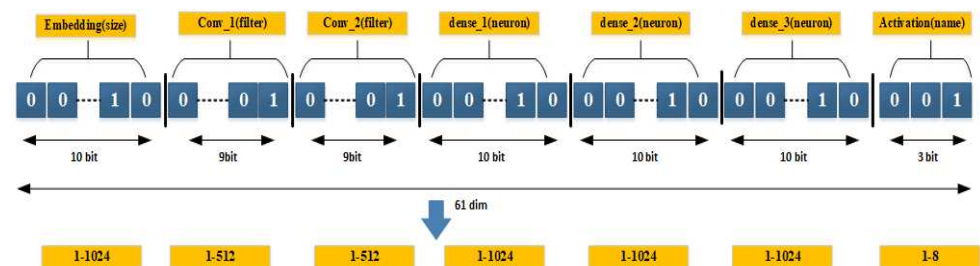
7.2. CNN Deep Neural Network Proposed Based on the Embedding Layer (CNNEM)

CNN is a particular type of multilayer perceptron that includes an input layer, an output layer, and a cannulation layer with several filters with different dimensions, followed by a pooling layer [50]. However, when this network is used for text and sentiment analysis, it is defined with the first layer of embedding. This embedding technique maps words and phrases from a dictionary to numeric vectors. We have designed a deep CNN model based on the embedding layer called CNNEM. The detailed specifications are given in Table 9. The CNNEM model shows some variable parameters in the Changeable column determined by the BAOVAH proposed approach.

Table 9. Specifications of CNNEM model and its modifiable neurons using proposed BAOVAH approach.

Model: "CNNEM"			
Layer (Type)	Output Shape	Param #	Changeable
Embedding_layer (Embedding)	(None, 100, 100)	261,200	0–1024
Conv_1 (Conv1D)	(None, 97, 64)	25,664	0–512
drop_1 (Dropout)	(None, 97, 64)	0	-
MaxPool_1 (MaxPooling1D)	(None, 49, 64)	0	-
drop_2 (Dropout)	(None, 49, 64)	0	-
Conv_2 (Conv1D)	(None, 48, 32)	4128	0–512
drop_3 (Dropout)	(None, 48, 32)	0	-
MaxPool_2 (MaxPooling1D)	(None, 24, 32)	0	-
flatten_2 (Flatten)	(None, 768)	0	-
dense_1 (Dense)	(None, 16)	12,304	0–1024
drop_4 (Dropout)	(None, 16)	0	-
dense_2 (Dense)	(None, 8)	136	0–1024
drop_5 (Dropout)	(None, 8)	0	-
dense_3 (Dense)	(None, 4)	36	0–1024
output (Dense)	(None, 2)	10	-

To better understand the determination of CNNEM model parameters by the BAOVAH algorithm, we have shown it in Figure 18. In Figure 20, we mentioned the range of numbers, the number of bits required for each layer, and finally, the dimensions of the problem for the BAOVAH proposed approach. Finally, the CNNEM model requires 61 bits with these bits will be generated with zero and one by the BAOVAH algorithm and will optimize the CNNEM model. The improved model by the BAOVAH approach is named CNNEMBH.

**Figure 20.** Determining CNNEM in-depth model parameters and problem dimensions for the proposed BAOVAH approach.

7.3. The Results of Improving the CNNEM Model with the BAOVAH Proposed Approach

In this subsection, the parameters specified in Figure 20, including the size of the embedding layer, the numbers of the convolutional layer filters 1 and 2, and the dense layer of neurons one to three. The name of the activating function for each layer is determined by the BAOVAH approach to increase the accuracy of the CNNEM model and its efficiency in sentiment analysis. For the initial evaluation in this section, first, the CNNEM model with the basic parameters, shown in Table 10, is being performed, and we display its different results. Then we offer the CNNEM model with the help of the parameters implemented by the BAOVAH approach and its various effects. In this part, the base model with the initial CNNEM parameters is named, and the improved model is called CNNEMBH by the BAOVAH approach. A comparison of the accuracy of the two in-depth models, CNNEM and CNNEMBH, for the three sentiment analysis datasets is shown in Table 10.

Table 10. A comparison of the accuracy of the two in-depth models.

Model Dataset	CNNEM		CNNEMBH	
	Train	Test	Train	Test
IMDB	0.98	0.73	0.99	0.79
Amazon	0.54	0.45	0.98	0.78
Yelp	0.52	0.48	0.98	0.78

Comparing the accuracy of two in-depth models, CNNEM and CNNEMBH, for the three sets of sentiment analysis in Table 10 shows that the improved CNNEMBH method has been able to increase the accuracy to 79 in the IMDB dataset to 0.78 in the Amazon and Yelp datasets and has achieved better results compared to the basic model.

8. Conclusions and Future Work

This paper introduced and implemented two approaches to AVOAs. In the first proposed model, a hyper-heuristic method is used to increase the performance of the proposed model to combine and integrate the mechanisms of the sine and cosine algorithm with the African vulture algorithm. This combination's purpose is to choose both algorithms' mechanisms intelligently and avoid increasing the computational complexity. On the other hand, in the first proposed model, two mechanisms, the Disruption operator and the Bitwise strategy, have been used to maximize the capability and efficiency of the first proposed model.

Each used four S-shaped transfer functions and four V-shaped transfers to binarize the basic AVOA. In addition, this paper presented an improved version of the AVOA called BAVOA-v1, where four different strategies were applied to improve the performance of the AVOA in the feature selection problem. These four strategies included: IPRS, mutation neighborhood search strategy (MNSS) (balancing exploration and exploitation), multi-parent crossover strategy (increasing exploitation), and Bitwise strategy (increasing diversity and exploration). These strategies were used to provide solutions with more variety and to assure the quality of solutions. Each of these four strategies was designed and implemented in the exploration and exploitation step of the AVOA algorithm. Finally, the proposed BAOVAH approach was evaluated on 30 UCI datasets. The results of different simulations showed that the proposed BAOVAH algorithm performed better than the basic binary meta-heuristic algorithms, and it has shown better performance. Hence, the proposed BAVOA algorithm in 67% of the data set is the most accurate, and in 93% of the data set, it is the best fitness function value. In terms of feature selection, it has shown high performance. A practical study in which the BAOVAH approach was used to determine the appropriate values of hyper-parameters of the deep learning algorithm was performed in sentiment analysis.

The string fixed-length binary coding method was used, fitting deep learning algorithms. Moreover, a new deep convolutional network called CNNEM based on Embedded layers was designed. Determining the appropriate values of CNNEM hyper-parameters is done with the proposed BAOVAH approach. The results of various experiments on three basic sets of sentiment analysis called IMDB, Amazon, and Yelp show that the BAOVAH algorithm increases the accuracy of the CNNEM network in the IMDB dataset by 6%, in the Amazon dataset by 33%, and in the Yelp dataset by 30%. It has optimized the appropriate values of CNNEM hyper parameters well. Different primary population production strategies are considered new local search and binary operators for future work. The BAOVAH approach will increase the precision of profound algorithms in image processing and data with large dimensions. In addition, the BAOVAH approach can be used for various engineering and medical research.

Author Contributions: Conceptualization, A.S. and F.S.G.; writing—original draft preparation, A.S. and F.S.G.; Review and editing, M.M. and V.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors are grateful to the reviewers for providing advice and guidance as to how to improve the paper.

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

Abbreviations

ABC	Artificial Bee Colony
ALO	Ant Lion Optimization
AVOA	African Vulture Optimization Algorithm
BALO	Binary Ant Lion Optimization
BBO	Binary Butterfly Optimization
BAOVAH	Binary African Vulture Optimization Algorithm with Hyper-heuristic
BCCSA	Binary Chaotic Crow Search Algorithm
BCSA	Binary Crow Search Algorithm
BDA	Binary Dragonfly Algorithm
BGO	Binary Grasshopper Optimization
BGWO	Binary Gray Wolf Algorithm
BOA	Butterfly Optimization Algorithm
BSSA	Binary Salp Swarm Algorithm
CCSA	Chaotic Crow Search Algorithm
COA	Coyote Optimization Algorithm
CSA	Crow Search Algorithm
DA	Dragonfly Algorithm
EPO	Emperor Penguin Optimizer
FFA	Fruit Fly Algorithm
FFA	Farmland Fertility Algorithm
FS	Feature Selection
GA	Genetic Algorithm
GOA	Grasshopper Optimization Algorithm
GS	Gravitational Search
GSKO	Gaining–Sharing Knowledge-Based Optimization
GWO	Grey Wolf Optimization
HHO	Harris Hawks Optimization
MBA	Mine Blast Algorithm
PSO	Particle Swarm Optimization
SA	Simulated Annealing
SOS	Symbiotic Organisms Search
SPSA	Salp Swarm Algorithm
SSA	Salp Swarm Algorithm
WOA	Whale Optimization Algorithm
IPRS	Initial Population generation based on Ranking Strategy

References

1. Hancer, E.; Xue, B.; Zhang, M. A survey on feature selection approaches for clustering. *Artif. Intell. Rev.* **2020**, *53*, 4519–4545. [\[CrossRef\]](#)
2. Nadimi-Shahraki, M.H.; Banaie-Dezfouli, M.; Zamani, H.; Taghian, S.; Mirjalili, S. B-MFO: A binary moth-flame optimization for feature selection from medical datasets. *Computers* **2021**, *10*, 136. [\[CrossRef\]](#)
3. Abdollahzadeh, B.; Gharehchopogh, F.S. A multi-objective optimization algorithm for feature selection problems. *Eng. Comput.* **2021**, *38*, 1845–1863. [\[CrossRef\]](#)
4. Nadimi-Shahraki, M.H.; Zamani, H.; Mirjalili, S. Enhanced whale optimization algorithm for medical feature selection: A COVID-19 case study. *Comput. Biol. Med.* **2022**, *148*, 105858. [\[CrossRef\]](#) [\[PubMed\]](#)

5. Shaddeli, A.; Gharehchopogh, F.S.; Masdari, M.; Solouk, V. BFRA: A New Binary Hyper-Heuristics Feature Ranks Algorithm for Feature Selection in High-Dimensional Classification Data. *Int. J. Inf. Technol. Decis. Mak.* **2022**, 1–66. [\[CrossRef\]](#)
6. Hosseini, F.; Gharehchopogh, F.S.; Masdari, M. A Botnet Detection in IoT Using a Hybrid Multi-objective Optimization Algorithm. *New Gener. Comput.* **2022**, 1–35. [\[CrossRef\]](#)
7. Gharehchopogh, F.S. Advances in tree seed algorithm: A comprehensive survey. *Arch. Comput. Methods Eng.* **2022**, 1–24. [\[CrossRef\]](#)
8. de Carvalho, V.R.; Özcan, E.; Sichman, J.S. Comparative Analysis of Selection Hyper-Heuristics for Real-World Multi-Objective Optimization Problems. *Appl. Sci.* **2021**, *11*, 9153. [\[CrossRef\]](#)
9. Abiodun, E.O.; Alabdulatif, A.; Abiodun, O.I.; Alawida, M.; Alabdulatif, A.; Alkhaldeh, R.S. A systematic review of emerging feature selection optimization methods for optimal text classification: The present state and prospective opportunities. *Neural Comput. Appl.* **2021**, *33*, 15091–15118. [\[CrossRef\]](#)
10. Montazeri, M. HHFS: Hyper-heuristic feature selection. *Intell. Data Anal.* **2016**, *20*, 953–974. [\[CrossRef\]](#)
11. Gharehchopogh, F.S.; Nadimi-Shahraki, M.H.; Barshandeh, S.; Abdollahzadeh, B.; Zamani, H. CQFFA: A Chaotic Quasi-oppositional Farmland Fertility Algorithm for Solving Engineering Optimization Problems. *J. Bionic Eng.* **2022**, 1–26. [\[CrossRef\]](#)
12. Nadimi-Shahraki, M.H.; Fatahi, A.; Zamani, H.; Mirjalili, S. Binary Approaches of Quantum-Based Avian Navigation Optimizer to Select Effective Features from High-Dimensional Medical Data. *Mathematics* **2022**, *10*, 2770. [\[CrossRef\]](#)
13. Gharehchopogh, F.S. An Improved Tunicate Swarm Algorithm with Best-random Mutation Strategy for Global Optimization Problems. *J. Bionic Eng.* **2022**, 1–26. [\[CrossRef\]](#)
14. Mirjalili, S.; Lewis, A. S-shaped versus V-shaped transfer functions for binary particle swarm optimization. *Swarm Evol. Comput.* **2013**, *9*, 1–14. [\[CrossRef\]](#)
15. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **2016**, *172*, 371–381. [\[CrossRef\]](#)
16. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary ant lion approaches for feature selection. *Neurocomputing* **2016**, *213*, 54–65. [\[CrossRef\]](#)
17. De Souza, R.C.T.; dos Santos Coelho, L.; De Macedo, C.A.; Pierezan, J. A V-shaped binary crow search algorithm for feature selection. In Proceedings of the 2018 IEEE congress on evolutionary computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
18. Mafarja, M.; Aljarah, I.; Heidari, A.A.; Faris, H.; Fournier-Viger, P.; Li, X.; Mirjalili, S. Binary dragonfly optimization for feature selection using time-varying transfer functions. *Knowl. Based Syst.* **2018**, *161*, 185–204. [\[CrossRef\]](#)
19. Faris, H.; Mafarja, M.M.; Heidari, A.A.; Aljarah, I.; Ala'M, A.-Z.; Mirjalili, S.; Fujita, H. An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowl. Based Syst.* **2018**, *154*, 43–67. [\[CrossRef\]](#)
20. Sayed, G.I.; Hassanien, A.E.; Azar, A.T. Feature selection via a novel chaotic crow search algorithm. *Neural Comput. Appl.* **2019**, *31*, 171–188. [\[CrossRef\]](#)
21. Arora, S.; Anand, P. Binary butterfly optimization approaches for feature selection. *Expert Syst. Appl.* **2019**, *116*, 147–160. [\[CrossRef\]](#)
22. Tubishat, M.; Idris, N.; Shuib, L.; Abushariah, M.A.; Mirjalili, S. Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Syst. Appl.* **2020**, *145*, 113122. [\[CrossRef\]](#)
23. de Souza, R.C.T.; de Macedo, C.A.; dos Santos Coelho, L.; Pierezan, J.; Mariani, V.C. Binary coyote optimization algorithm for feature selection. *Pattern Recognit.* **2020**, *107*, 107470. [\[CrossRef\]](#)
24. Abdel-Basset, M.; El-Shahat, D.; El-henawy, I.; de Albuquerque, V.H.C.; Mirjalili, S. A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Syst. Appl.* **2020**, *139*, 112824. [\[CrossRef\]](#)
25. Sihwail, R.; Omar, K.; Ariffin, K.A.Z.; Tubishat, M. Improved harris hawks optimization using elite opposition-based learning and novel search mechanism for feature selection. *IEEE Access* **2020**, *8*, 121127–121145. [\[CrossRef\]](#)
26. Hussien, A.G.; Hassanien, A.E.; Houssein, E.H.; Amin, M.; Azar, A.T. New binary whale optimization algorithm for discrete optimization problems. *Eng. Optim.* **2020**, *52*, 945–959. [\[CrossRef\]](#)
27. Mohammadzadeh, H.; Gharehchopogh, F.S. An efficient binary chaotic symbiotic organisms search algorithm approaches for feature selection problems. *J. Supercomput.* **2021**, *77*, 9102–9144. [\[CrossRef\]](#)
28. Dhiman, G.; Oliva, D.; Kaur, A.; Singh, K.K.; Vimal, S.; Sharma, A.; Cengiz, K. BEPO: A novel binary emperor penguin optimizer for automatic feature selection. *Knowl. Based Syst.* **2021**, *211*, 106560. [\[CrossRef\]](#)
29. Hosseinalipour, A.; Gharehchopogh, F.S.; Masdari, M.; Khademi, A. A novel binary farmland fertility algorithm for feature selection in analysis of the text psychology. *Appl. Intell.* **2021**, *51*, 4824–4859. [\[CrossRef\]](#)
30. Abdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Comput. Ind. Eng.* **2021**, *158*, 107408. [\[CrossRef\]](#)
31. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **2016**, *96*, 120–133. [\[CrossRef\]](#)
32. Cowling, P.; Kendall, G.; Soubeiga, E. A hyperheuristic approach to scheduling a sales summit. In Proceedings of the International Conference on the Practice and Theory of Automated Timetabling, Konstanz, Germany, 16–18 August 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 176–190.
33. Cowling, P.; Kendall, G.; Soubeiga, E. A parameter-free hyperheuristic for scheduling a sales summit. In Proceedings of the 4th Metaheuristic International Conference, MIC, Fairfax, VA, USA, 26–28 July 2001; pp. 127–131.

34. Neggaz, N.; Ewees, A.A.; Abd Elaziz, M.; Mafarja, M. Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Syst. Appl.* **2020**, *145*, 113103. [[CrossRef](#)]
35. Das, B.; Prasad, K.E.; Ramamurty, U.; Rao, C. Nano-indentation studies on polymer matrix composites reinforced by few-layer graphene. *Nanotechnology* **2009**, *20*, 125705. [[CrossRef](#)] [[PubMed](#)]
36. Al-Sharhan, S.; Bimba, A. Adaptive multi-parent crossover GA for feature optimization in epileptic seizure identification. *Appl. Soft Comput.* **2019**, *75*, 575–587. [[CrossRef](#)]
37. Droste, S. Analysis of the (1 + 1) EA for a dynamically bitwise changing OneMax. In Proceedings of the Genetic and Evolutionary Computation Conference, Chicago, IL, USA, 9–11 July 2003; Springer: Berlin/Heidelberg, Germany, 2003; pp. 909–921.
38. Nakamura, R.Y.M.; Pereira, L.A.M.; Rodrigues, D.; Costa, K.A.P.; Papa, J.P.; Yang, X.-S. Binary bat algorithm for feature selection. In *Swarm Intelligence and Bio-Inspired Computation*; Elsevier: Amsterdam, The Netherlands, 2013; pp. 225–237.
39. Abdel-Basset, M.; Sallam, K.M.; Mohamed, R.; Elgendy, I.; Munasinghe, K.; Elkomy, O.M. An improved binary grey-wolf optimizer with simulated annealing for feature selection. *IEEE Access* **2021**, *9*, 139792–139822. [[CrossRef](#)]
40. Mirjalili, S.; Mirjalili, S.M.; Yang, X.-S. Binary bat algorithm. *Neural Comput. Appl.* **2014**, *25*, 663–681. [[CrossRef](#)]
41. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
42. Li, Y.; Xiao, J.; Chen, Y.; Jiao, L. Evolving deep convolutional neural networks by quantum behaved particle swarm optimization with binary encoding for image classification. *Neurocomputing* **2019**, *362*, 156–165. [[CrossRef](#)]
43. Fang, Z.; Ren, J.; Marshall, S.; Zhao, H.; Wang, S.; Li, X. Topological optimization of the densenet with pretrained-weights inheritance and genetic channel selection. *Pattern Recognit.* **2021**, *109*, 107608. [[CrossRef](#)]
44. Wang, B.; Sun, Y.; Xue, B.; Zhang, M. A hybrid differential evolution approach to designing deep convolutional neural networks for image classification. In Proceedings of the Australasian Joint Conference on Artificial Intelligence, Wellington, New Zealand, 11–14 December 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 237–250.
45. Rani, R.; Lobiyal, D. Automatic Construction of Generic Stop Words List for Hindi Text. *Procedia Comput. Sci.* **2018**, *132*, 362–370. [[CrossRef](#)]
46. Porter, M.F. An algorithm for suffix stripping. *Program* **1980**, *14*, 130–137. [[CrossRef](#)]
47. Xu, J.; Croft, W.B. Corpus-based stemming using cooccurrence of word variants. *ACM Trans. Inf. Syst. (TOIS)* **1998**, *16*, 61–81. [[CrossRef](#)]
48. Weikum, G. Foundations of statistical natural language processing. *ACM SIGMOD Rec.* **2002**, *31*, 37–38. [[CrossRef](#)]
49. Porter, M.F. Snowball: A Language for Stemming Algorithms. 2001. Available online: <http://snowball.tartarus.org/texts/introduction.html> (accessed on 10 September 2022).
50. Xu, S.; Shijia, E.; Xiang, Y. Enhanced attentive convolutional neural networks for sentence pair modeling. *Expert Syst. Appl.* **2020**, *151*, 113384. [[CrossRef](#)]