Received: April 23, 2024. Revised: October 12, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Proposed an Accurate Optimization Algorithm Using Butterfly Optimization and Sine-Cosine Optimization Algorithms

Mustafa Salam Kadhm[1]* **Mamoun Jassim Mohammed[2]** **Sufyan Othman Zaben[2]**

[1]*Computer Department, College of Basic Education, Mustansiriyah University, Iraq*
[2]*Department of Computer Engineering, College of Engineering, Al-Iraqia University, Iraq*
* Corresponding author's Email: mst.salam@mustansiriyah.edu.iq

Abstract: Feature selection consider one of the essential pre-processing stage of the classification task in machine learning. The datasets that used in classification contain irrelevant features that may directly affect the performance of the used classifiers. The classification accuracy could be race using appropriate feature selector by reducing the number of the extracted features from the datasets. The common and the powerful algorithms that successfully used for feature selection task is the optimization algorithms. Based on the searching strategy of butterflies, the Butterfly Optimization Algorithm (BOA) is a meta-heuristic swarm intelligence algorithm. Because of its performance, BOA has been applied to a wide range of optimization problems. However, BOA has limitations such as reduced population variety and a tendency to become locked in a local optimum. Besides, it suffers in converges speed, accuracy, and precision of the optimal objective value when optimizing high dimensional problems. Therefore, this paper proposed an accurate algorithm based on BOA and Sine-Cosine Algorithm called BOA-SC. The BOA first improved via the update equations then hybrid with SC to enhance the local search stage for better optimization results. Using the improvement strategy and SC enhance the performance of BOA and solve the lower coverage and local optima issues that BOA suffers from. The performance of the proposed hybrid algorithm is evaluated using two assessments via converges speed, the accuracy, and precision of the optimal objective value. First, 23 benchmark functions used to evaluate proposed algorithm that achieved a high optimization result comparing with six most recent metaheuristic algorithms puzzle optimization algorithm (POA), northern goshawk optimization (NGO), coati optimization algorithm (COA), swarm bipolar algorithm (SBA), apiary organizational-based optimization algorithm (AOOA), and swarm space hopping algorithm (SSHA). The obtained results show that, BOA-SC is better than POA, NGO, COA, SBA, AOOA, and SSHA, in 5, 6, 8, 13, 18, 22, and 23 functions. In the second evaluation, the proposed algorithm compared with four BOA variants algorithms s-shaped binary butterfly optimization algorithm (S-bBOA), dynamic butterfly optimization algorithm (DBOA), chaotic butterfly optimization algorithm (CBOA), and optimization and extension of binary butterfly optimization approaches (OEbBOA) which are employed for feature selections methods. The results of BOA-SC are funnier than S-bBOA, DBOA, CBOA, and OEbBOA in three distinct datasets (Sonar, Waveform, and Spect) by archiving a high classification accuracy 97%, 86%, and 87% as a feature selection algorithm for the classification task.

Keywords: BOA, SC, Optimization, Classification, Feature selection.

1. Introduction

Many biological systems have been developed by nature which is assisting them in their survival for millions of years. For many real-world problems, over time, these natural systems have grown to be so resilient and effective [1]. The real-world combinatorial or global optimization problems are address by several metaheuristic algorithms, including Artificial Bee Colony (ABC) [2], Cuckoo Search (CS) [3], Firefly Algorithm (FA) [4], Gray Wolf Optimization (GWO) [5], Sine-Cosine (SC) [6], Butterfly Optimization (BOA) [7], and Particle Swarm Optimization (PSO) [8]. In recent years many metaheuristics algorithms based on swarm intelligence to solve various optimization problems. These algorithms are puzzle optimization algorithm

(POA) [9], northern goshawk optimization (NGO) [10], coati optimization algorithm (COA) [11], swarm bipolar algorithm (SBA) [12], apiary organizational-based optimization algorithm (AOOA) [13], and swarm space hopping algorithm (SSHA) [14] and so on.

A widely used and very effective algorithm of optimization is BOA. Butterfly Optimization Algorithm (BOA) is a nature inspired metaheuristics that solve the global optimization problems via mimic's food search and mating behaviour of butterflies. BOA successfully applied in many real-word problems and achieved a satisfied result. BOA has been applied in classification task as a feature selection stage using different datasets. Besides, it's also performed better than other algorithms on several engineering problems, such as (gear train design, spring design, and welded beam design [7].

Numerous research studies take feature selection into account using different optimization strategies. An improved whale optimization algorithm (WOA) in [15-16] used for feature selection, in [17] an improved version of particle swarm optimization (PSO) used in classification task for selecting the required features, in [18] Grasshopper optimization algorithm (GOA), in [19] used Firefly algorithm (FFA) for feature selection, and in [20] Differential Evolution (DE) for features selection.

BOA has been used successfully to solve feature selection challenges in machine learning and data analysis [4]. The process of discovering and selecting a subset of relevant features or variables from a broader collection of available features is known as feature selection. Enhancing machine learning models' efficiency, lowering overfitting, and expediting the training process are the objectives of working with a more condensed and informative feature set. Number of researches improved the performance of BOA and succussfuly applied for feature selection task, like s-shaped binary butterfly optimization algorithm (S-bBOA) [21], dynamic butterfly optimization algorithm (DBOA) [22], chaotic butterfly optimization algorithm (CBOA) [23], and optimization and extension of binary butterfly optimization approaches (OEbBOA) [24].

This paper proposes an improved BOA approach based on the SC algorithm. BOA was improved first by utilizing a modified equation, then by combining it with the SC algorithm. The proposed solution tackles BOA's low accuracy, coverage area, and local search concerns. Twenty-seven benchmark functions were used to assess the proposed technique. In addition, the performance of the proposed approach is compared to that of existing modified algorithms for classification tasks using five different datasets from the UCI repository. The main contributions of the proposed algorithm are represented below:

- BOA is improved via update equation.
- BOA is hybrid with SC to solve the local search problem that BOA suffer from.
- The hybrid approach leads for better converges speed, and accuracy of the optimal objective value.
- BOA-SC is validated and evaluated using 23 benchmark functions.
- The obtained results of BOA-SC are compared with six most recent optimization algorithms.
- BOA-SC is employed for feature selection task using five datasets and the results are compared with four BOA variants algorithms.

The remaining of the paper structured as following: Section 2 describe the literature review and Section 3 BOA and SC algorithms are described. The proposed algorithm is described in Section 4 and the obtained results in Section 4. Last, in Section 6 the conclusion is stated.

2. Literature review

Number of researches proposed new metaheuristics algorithms in recent years for solving different optimization problems. These algorithms based on various mathematic models to mimic the behaviour of the swarms. Besides, some researches consider improving the BOA for better optimization results and classification accuracy. In this part both aspect is reviewed and discussed.

In gaming based metaheuristic group, Puzzle Optimization Algorithm (POA) is proposed to solve various optimization problems. The main idea of the algorithm is to solve the puzzle via the cooperation between the players by putting the puzzle pieces into its right places (pattern) which is consider the optimal goal of the puzzle. The best player leads the others for the optimal solutions. Two steps are used for completing the puzzle process using POA process. First, each player imitates other players to complete his/her puzzle. In the second step, the player who is not able his puzzle, will get a help from other players to complete the puzzle pieces [9].

Another recent algorithm is northern goshawk optimization (NGO) which is proposed to solve different optimization and real word problems. NOG is a bird of hunting prey that using the hunting strategy as the optimization process. Each population member considers a solution in NGO to determine the variables values. The hunting strategy has two phases. The first phase is identifying the prey via global search (exploration) by randomly selecting the prey then attack it directly. Thus, aim to select the optimal area. While the second phase is tailing and chasing the prey. NGO chase the prey when its escape using the high speed ability which make the NOG chasing the prey in every situation via the local search (exploitation). After updating the population members in the two phases and completing the last iteration, the best solution of NGO [10].

Another new metaheuristic algorithm is the Coati Optimization Algorithm (COA) which is simulates the behaviour of coati in nature. In COA two nature behaviour is used. First one is attacking and hunting iguanas (exploration). The second one is escaping from the predators (exploitation). When the COA attack the prey, its separated into two groups. The first group scare the prey by climbing the tree, and the second group catch the fallen prey under the tree. The position of prey in COA design, is considered the best member of the population. However, the two COA behaviours depend on the only position information [11].

One the most recent metaphor-free metaheuristic search algorithm is Swarm bipolar algorithm (SBA). SBA used swarm intelligence and bipolar disorder to solve the complex problems. In order to diversify the searching process, SBA split the swarm into two equal size sub swarms. In each sub swarms there will be an elected leader based on its quality. Four references are used in SBA which are: the finest swarm members, the finest sub swarm members, the middle between two sub swarm members, and the randomly selected sub swarm members from the opposite one. Splitting the swarm and using four references enhance the exploration of the SBA to avoid the local optimal by diversify the motion of the swarm [12].

Another recent algorithm is apiary organizational-based optimization algorithm (AOOA) which is inspired by the complex organizational behaviour of the honeybees inside the apiary. The mathematical model of AOOA is developed using seven phases that translated from the queen, workers, and drones, activities inside the apiary. These phases are: initialization, drones exchange, queen fertilization, worker's lifecycle, queen investiture, fading out, swarming. The seven phases work together to find the problems optimal solutions [13].

Swarm space hopping algorithm (SSHA) is another recent algorithm that inspired by space hoping where the agent makes a large jump in the search space in order to explore a new region. SSHA contains three different searches. First search is the directed search which is the motion toward the high quality agents. The second search is the directed search which is the motion toward the resultant of the better agents. The third search is crossover search which is the crossover between the agent and randomized solution in the first or second half of the space [14].

In another hand, number of algortihms are proposed to improve the performance of BOA and applied in the classification task of the machine learning.

Arora et al. (2020) method used two-objective fitness function (maximizing the classification accuracy, and minimizing the number of selection features) for calculating the fitness of solutions. A sigmoid function was used to keep the solution in binary space. Researchers proved the adaptive mechanism in the S-bBOA algorithm is capable to accelerate the convergence with respect to the number of iterations, make the exploration and exploitation balanced in order to avoid a large number of local solutions in feature selection problems, and find an accurate estimation of the best solution. Therefore, this method demonstrated a better performance in comparison to some other optimization algorithms. [21].

Mohammad et al. (2020) proposed a solution for feature selection problem via a dynamic BOA called DBOA in order to reduce the number of extracted features, thus maximize the classification accuracy in the machine learning approaches. The local optimum is overcome by improving the solutions diversity via Local Search Algorithm based Mutation (LSAM). 20 UCI benchmarks datasets are used to evaluate the performance of DBOA. Authors mentioned that, DBOA achieved a high accuracy results comparing to other optimization algorithms results [22].

Asmaa et al. (2020) proposed chaotic butterfly optimization algorithm (CBOA) by integrating chaotic maps and BOA to increase the diversity and avoiding the local minima. In CBOA, chaotic maps invoked for updating the butterfly positions instead of using the random variables in the standard BOA. CBOA then transferred into binary search for better search results. CBOA tested using 16 benchmark datasets, compared with 6 metaheuristic algorithms and achieved a high accuracy results 95% [23].

In Zhang et al. (2020), the authors proposed a novel improved binary butterfly optimization approaches with various strategies for feature selection called Optimization and Extension of Binary Butterfly Optimization Approaches (OEbBOA). The proposed approaches try improve the structure of the binary Butterfly Optimization (bBOA) to enhance its classification accuracy, dimension reduction and reliability in feature selection task for who are interested in the fields of data mining and pattern recognition. These approaches are applied to reduce the randomness of bBOA's initialization and local search process. The proposed approaches (OEbBOA) are tested with the K nearest neighbor classier in which twenty UCI datasets and seven recent algorithms are utilized to assess the performance of the OEbBOA algorithm. The experimental results and nonparametric Wilcoxons rank sum test confirm the efficiency of the proposed OEbBOA in maximizing classification accuracy while minimizing the number of features selected [24].

All the abouve mentioned optimization algorithms still suffer from the local minima, converges speed, and accuracy issues. Although, several algorithms improved the obtained results for solving the optimization problems via various enhancement, the problems still need an optimal and accurate algorithm. This open research is, inspired us for propose the BOA-SC algorithm to overcame the gaps in other optimization algorithms.

3. Butterfly optimization algorithm (BOA) [6]

BOA is a new metaheuristic that was introduced in 2015 (Arora and Singh). The recently developed nature inspired meta-heuristic known as BOA imitates the searching and pairing habits of butterflies. The following features of the butterfly are expected in a BOA framework:

- Butterflies are drawn to one another by a scent that releases.
- Butterflies search randomly or in reaction to the one with the strongest scent.
- The butterfly stimulus's strength is found using the landscape of the desired function.

Eq. (1) provides the scent (f), which is computed as a function of butterfly motivation strength.

$$f = cI^{a} \tag{1}$$

Where the power exponent that is dependent on modality is represented by a, and the sensory modality by c. The a and c have values that fall between 0 and 1, or $c \in [0, 1]$. I stand for intensity of stimuli.

The two main stages of BOA are global and local search. In the global search stage, the butterfly uses the following Eq. (2) to move toward the ideal location ($g*$) depending on the fitness threshold of the objective function:

$$x_i^{t+1} = x_i^t + \left(r^2 \times g^* - x_i^t\right) \times f_i \tag{2}$$

The optimal current location is indicated by g*, and x is a vector that reflects the i_{th} butterfly location at time t. r is a randomly produced value that falls between 0 and 1. The following Eq. (3) provides the local search phase:

$$x_i^{t+1} = x_i^t + \left(r^2 \times x_j^t - x_k^t\right) \times f_i \tag{3}$$

Where x and r, are vectors that appear as j_{th} and k_{th} butterflies' location at time t. r and f are random number and scent correspondingly.

Given that the butterfly can forage and look for a mate in any phase (global) or (local) any step can be used. However, some natural disasters may have an impact on this, therefore an extra probability parameter (p) serves as an alternate of local and everywhere search. The complete steps of the BOA algorithm are illustrated in the Alg. (1) below.

Algorithm 1: BOA

Input: Populations
Output: Best solution

Begin
set up the population of n butterflies (b)
$X = (x1, x2 \ldots xn)$ with (d).
set the values of parameters $(a, c, and\ p)$
assess the strength of stimulus (Ii) at xi
While some termination criteria not met do
    For each b in X **do**
        A random number r is generated
        If $r < p$ **then**
            Perform global search using Eq. (2)
        Else
            Local search using Eq. (3)
        End If
    End For
    Update a 's value
End While
Optimal results achieved.
End

4. Sine cosine algorithm (SC) [7]

SC is a population-based meta-heuristic method that was recently developed and depend on the mathematical criteria of sin-cos functions. Using the below equations, this algorithm adjusts automatically the locations after producing the random starting solutions:

$$\begin{cases} x_i^{t+1} = x_i^t + A \\ \times \sin(r_1) \times \left| r_2 \times x_{Best} - x_i^t \right| \; if \; r_3 < 0.5 \\ x_i^{t+1} = x_i^t + A \\ \times \cos(r_1) \times \left| r_2 \times x_{Best} - x_i^t \right| \; otherwise \end{cases} \quad (4)$$

x_i^t denotes the i_{th} solution's location at iteration t, xBest denotes the population's best solution, r_1 is a random number between 0 and 2, r_2 is the best solution's random weight within a range of -2 to 2, r_3 is a random number between 0 and 1, and the symbol | | stands for absolute value. When the value of r_3 is smaller than 0.5, the potential solution updates its location using the sin-cos function. The function that helps maintain a balance between the exploration and exploitation of a search space represented by the parameter A as the following equation:

$$A = 2 - 2\left(\frac{t}{t_{max}}\right) \quad (5)$$

The sine-cosine algorithm's complete steps are listed in Alg. (2) below.

Algorithm 2: Sine-Cosine (SC)

Input: Populations
Output: Best solution

Begin
Init.: Set up a collection of search agents (or solutions) (X).
Do assess every search agent according to the goal function**.**
Adjust the best answer found thus far (P=X^*).
Refresh r_1, r_2, r_3, and r_4.
Utilizing Eq. (4), adjust the search agents' locations.
While (t < max total of iterations)
Return the global optimum, which is the best solution found thus far.
End

5. Proposed work (BOA-SC)

The proposed approach enhances the BOA algorithm with the SC algorithm. The notation table has been given in Table 1. Because the BOA algorithm has two key stages, global search and local search, the butterfly uses Eq. (2) to find the fittest butterfly/solution g* in the global search stage. The local search, on the other hand, is represented by Eq. (3), which is a local random walk.

The BOA was able to achieve high accuracy results in the majority of test cases during the global search stage.

Table 1. Notation table

Notation	Meaning
f	The perceived magnitude of the fragrance
c	The sensory modality
I	The stimulus intensity
a	The power exponent dependent on modality
x_i^t	The solution vector x_i for ith butterfly in iteration number t
g^*	The current best solution found among all the solutions in current iteration
f_i	Fragrance of ith butterfly
r	A random number in [0, 1]
x_j^t	jth butterflies from the solution space
x_k^t	kth butterflies from the solution space
p	Switch probability between common global search to intensive local search
r_1	Random number in [0, 2]
r_2	Random number in [-2, 2]
r_3	Random number in [0, 1]
x_i^t	The position of the current solution in *i-th* dimension at *t-th* iteration
p_i	Position of the destination point in *i-th* dimension
t	The current iteration
T	The maximum number of iterations
rm	A random number in [0.3, 0.9]
TP	True positive
TN	True negative
FP	False positive
FN	False negative

However, during the local search stage, the random walk may result in the worst solutions. Besides, BOA suffer from converges speed, accuracy, and precision of the optimal objective value. In the proposed algorithm BOA-SC, two stages will be applied to improve the performance of the standard BOA. First, adding a setting variable to the local search phase in the original equation. Second, using SC algorithm to address the BOA problem issue by employing Eqs. (4) and (5) for a better search process during the local search stage. Fig. 1 illustrates the main steps of the BOA-SC when the contributions stage is ssurrounded by red line.

In order to identify the attractive areas in the search area, various solutions are integrated using the SC algorithm amid the exploration phase, which has a high randomness rate via an abrupt set of results. However, during the exploitation phase a progressive change in the random solutions is observed, and

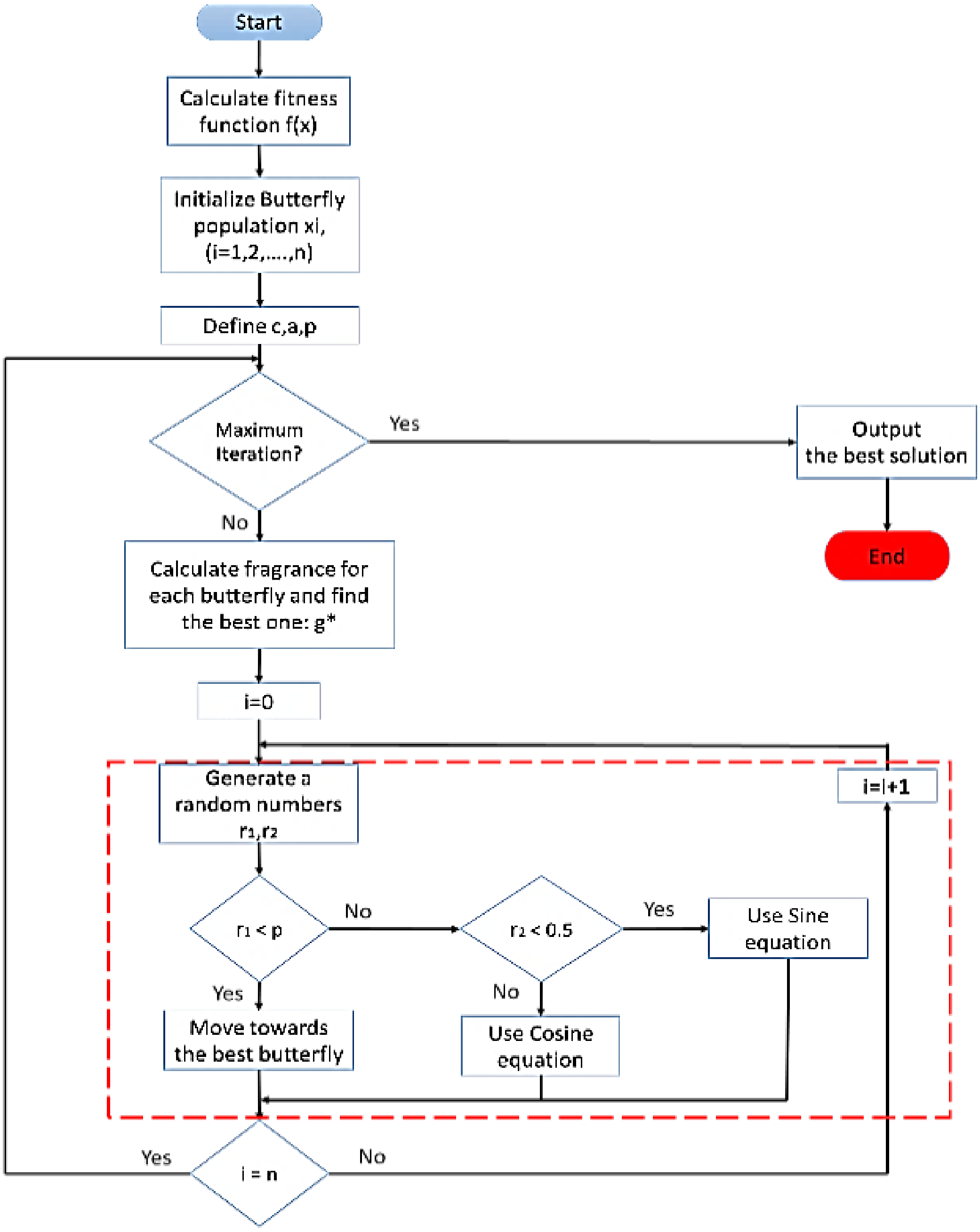

Figure. 1 Proposed Algorithm

random variations are significantly lower than during the exploration phase.

To avoid the local minima, issue in the BOA algorithm, Eq. (3) in the proposed approach is modified by adding a random variable (rm) in the range (0.3 - 0.9) which are specified after many experiments. The BOA algorithm's modified equation is shown below:

$$x_i^{t+1} = x_i^t + \left(r^2 \times x_j^t - x_k^t\right) \times f_i + rm \quad (6)$$

Adding the *rm* variable leads for better diversity and search space which make the BOA search in different space if there is no better solution in the current space, thus avoiding the local search weakness.

Eqs. (4) and (5) of the sine and cosine are used for location updating in SC algorithm. These two equations are combined to be used as in Eqs. (7) and (8).

$$\mathrm{x}_\mathrm{i}^{\mathrm{t}+2} = \mathrm{x}_\mathrm{i}^{\mathrm{t}+1} + \mathrm{A} \times \sin(\mathrm{r}_1) \times \left|\mathrm{r}_2 \times \mathrm{x}_{\mathrm{Best}} - \mathrm{x}_\mathrm{i}^\mathrm{t}\right| \quad (7)$$

$$\mathrm{x}_\mathrm{i}^{\mathrm{t}+2} = \mathrm{x}_\mathrm{i}^{\mathrm{t}+1} + \mathrm{A} \times \cos(\mathrm{r}_1) \times \left|\mathrm{r}_2 \times \mathrm{x}_{\mathrm{Best}} - \mathrm{x}_\mathrm{i}^\mathrm{t}\right| \quad (8)$$

Here the x_i^{t+2} is first updated based on the current solution of the BOA algorithm that obtained by Eqs. (2) and (3), then the rest of the results will be based on the sine cosine Eqs. (4) and (5) which will achieve the best possible coverage search area.

Hybrid the sine-cosine equation with the BOA equations for the local search stage makes the address the converges speed issue that BOA suffer from via testing the incoming parameters and select the write equations to feed that parameter for better optimization results. The BOA-SC start with initializing the population X_i then defining the required parameters like population size, maximum number of iterations, switch probability, sensor modality, and power exponent. After that the fitness function *f(x)* for each butterfly is computing to find the possible optimal solution. The scent is then calculated *f* for each butterfly.

Here the movement of the butterfly will depend first on the scent, probability switch, and sine-cosine equations. For the global search the butterfly will move to the best solution with a probability of 1-p. However, in the local search the butterfly will teste via other condition to move towards sine or cosine equations. The position and scent then updated after the movement process then best solution is tracked during the selected iterations. Alg. (3) illustrates the proposed algorithm steps.

Algorithm 3: Proposed BOA-SC

Input: Populations
Output: Best solution

```
# Begin
Start the population Xi (i=1....n).
Determine the fitness function f(x).
switch probability (p), Sensor modality (c) and
power exponent (a) are all defined.
While requirements for halting are not satisfied do
    For each butterfly (g*) in the population,
    generate a random number r, r1, r2.
      If r1 < (p)
            Move to the best butterfly Eq. (6)
        Else If r2 < 0.5
                Move using sine Eq. (7)
        Else
                Move using cosine Eq. (8)
         End if
    End for
   Update a value
End while
Return best solution
# End
```

Using the hybrid approach in the proposed BOA-SC algorithm by modifying the local search equation then using sine-cosine equations enhancing the performance of the proposed algorithm by selecting a minimum number of relevants features from the extracted features in the classification task. This selection reduces the number of the selected features, reduce the classification time and maximize the classification accuracy.

6. Results

The suggested technique is tested using 23 benchmark procedures within a Windows environment using MATLAB software. Several well-known metrics (mean, standard deviation, best, and worst) are used for evaluation. Table 2 shows the impact of using the improved version of BOA algorithm on the obtained results.

Table 2. Result of BOA-SC algorithm

Fun.	Best	Worst	SD	Mean
f_1	0	0	0	0
f_2	0	0	0	0
f_3	0	0	0	0
f_4	0	0	0	0
f_5	4.7x10^{-06}	3.4x10^{-06}	2.8x10^{-05}	2.38x10^{-06}
f_6	0	0	0	0
f_7	0	0	0	0
f_8	2.3x10^{-18}	2.9x10^{-18}	2.77x10^{-21}	2.66x10^{-20}
f_9	4.9x10^{-02}	8.2x10^{-02}	9.88x10^{-05}	6.62x10^{-01}
f_{10}	-9.4x10	− 9.6x10	-2.39x10	-9.56x10
f_{11}	0	0	0	0
f_{12}	0	0	0	0
f_{13}	0	0	0	0
f_{14}	3.3x10^{+02}	8.7x10^{+02}	2.12x10^{-05}	6.06x10^{-02}
f_{15}	-1.0x10	-1.0x10	0	-1.0x10
f_{16}	-4.5x10	-1.3x10	2.48x10^{-11}	-2.97x10^{+01}
f_{17}	0	0	0	0
f_{18}	2.8 x10^{-15}	8.4x10^{-15}	7.8x10^{-22}	5.6x10^{-2}
f_{19}	0	0	0	0
f_{20}	-1.9x10	-8.5x10	-1.9x10^{+02}	-5.23x10^{+02}
f_{21}	0	0	0	0
f_{22}	0	0	0	0
f_{23}	0	0	0	0

Table 3. Result of BOA-SC and other Algorithms

Fun.		BOA-SC	POA	NGO	COA	SBA	AOOA	SSHA
f_1	Mean	0	3.63	6.69×10^{3}	1.34×10^{3}	0	0	0.46
	SD	0	1.39×10^{1}	3.89×10^{3}	4.03×10^{2}	0	0	0.42
f_2	Mean	0	0	2.89×10^{41}	0	0	0	0
	SD	0	0	1.32×10^{42}	0	0	0	0
f_3	Mean	0	3.22×10^{4}	1.26×10^{5}	6.44×10^{4}	2.55	0	1.04×10^{3}
	SD	0	2.97×10^{4}	6.04×10^{4}	3.62×10^{4}	4.02	0	1.35×10^{3}
f_4	Mean	0	3.58×10^{1}	5.02x10	3.44×10^{1}	0	0	0.53
	SD	0	2.97×10^{1}	1.64x10	1.24×10^{1}	0	0	0.22
f_5	Mean	$\mathbf{2.38 \times 10^{-06}}$	8.23×10^{5}	2.40×10^{6}	3.16×10^{5}	4.89×10^{1}	3.24	5.42x10
	SD	2.80×10^{-05}	3.69×10^{6}	1.92×10^{6}	2.46×10^{5}	0.05	3.2	4.2573
f_6	Mean	**0**	6.99×10^{1}	6.22×10^{3}	1.27×10^{3}	9.93	4.11	1.15x10
	SD	0	2.99×10^{2}	4.46×10^{3}	5.28×10^{2}	0.45	0.86	0.76
f_7	Mean	0	0.21	2.59	0.55	0	0	0.03
	SD	0	0.63	2.09	0.30	0	0	0.02
f_8	Mean	$\mathbf{2.66 \times 10^{-20}}$	-2.23×10^{3}	-2.90×10^{3}	-4.53×10^{3}	-3.61×10^{3}	-6.50×10^{2}	-2.78×10^{3}
	SD	2.77×10^{-21}	4.41×10^{2}	4.87×10^{2}	8.61×10^{2}	4.56×10^{2}	8.70×10^{2}	3.50×10^{2}
f_9	Mean	6.62×10^{-01}	1.54	4.52×10^{2}	2.02×10^{2}	0	0	0.65
	SD	9.88×10^{-05}	6.72×10^{1}	5.10x10	5.07×10^{1}	0	0	0.83
f_{10}	Mean	-9.56x10	5.55	1.23x10	6.59	0	0	0.14
	SD	-2.39x10	7.25	2.68	1.01	0	0	0.09
f_{11}	Mean	0	0.85	4.87x10	1.65×10^{1}	0	0	0.17
	SD	0	2.08	2.52x10	7.6192	0	0	0.22
f_{12}	Mean	0	1.81	1.54×10^{6}	2.06×10^{3}	0	0.025	1.08
	SD	0	0.24	4.48×10^{6}	5.88×10^{3}	0.15	0.065	0.12
f_{13}	Mean	**0**	1.29×10^{6}	7.04×10^{6}	1.50×10^{5}	3.11	1.93	3.44
	SD	0	5.13×10^{6}	9.28×10^{6}	2.44×10^{5}	0.02	0.72	0.15
f_{14}	Mean	6.06×10^{-02}	9.73	2.89x10	9.54	7.77	0	8.67
	SD	2.12×10^{-05}	4.24	5.28x10	4.20	4.38	0.52	3.55
f_{15}	Mean	-1.00x10	0.10	0.03	0.011	0	0.01	0
	SD	0	0.05	0.02	0.011	0	0	0.02
f_{16}	Mean	$-2.97 \times 10^{+01}$	-0.43	0.83	-0.98	-1.02	0	-0.98
	SD	2.48×10^{-11}	0.45	0.23	0.14	0	0.06	0.07
f_{17}	Mean	0	2.59	1.42	0.53	0.44	0	0.45
	SD	0	3.03	1.75	0.33	0.09	0	0.13
f_{18}	Mean	$\mathbf{5.67 \times 10^{-20}}$	3.92×10^{1}	3.09x10	1.30×10^{1}	8.60	3.00	1.66x10
	SD	7.82×10^{-22}	0	2.07x10	2.12×10^{1}	1.04×10^{1}	0	1.48x10
f_{19}	Mean	0	-0.04	-0.04	-0.04	-0.04	-0.81	0
	SD	0	0	0	0	0	0	0
f_{20}	Mean	$-5.23 \times 10^{+02}$	-1.11	-2.15	-2.68	-2.69	0	-2.81
	SD	$-1.99 \times 10^{+01}$	0.56	0.46	0.35	0.35	0	0.26
f_{21}	Mean	0	-0.41	0	-2.10	-4.18	-4.7371	-2.84
	SD	0	0.09	0.64	0.35	1.47	0.14585	1.45
f_{22}	Mean	**0**	-0.47	-1.07	-2.01	-3.47	-7.5457	-2.97
	SD	0	0.19	0.70	0.93	0.98	0.9718	1.49
f_{23}	Mean	**0**	-0.61	-1.11	-2.47	-3.64	-6.5499	-3.10
	SD	0	0.22	0.31	1.14	1.66	0	1.71

Table 3 shows that the proposed algorithm outperformed the other algorithms in seven benchmark functions (F5, F6, F8, F13, F18, F22, and F23). Furthermore, all metrics show an improvement in the rest of the benchmark function results.

On the other hand, the proposed algorithm's results are compared to recent optimization algorithms such as puzzle optimization algorithm (POA), northern goshawk optimization (NGO), coati optimization algorithm (COA), swarm bipolar algorithm (SBA), apiary organizational-based optimization algorithm (AOOA), and swarm space hopping algorithm (SSHA). Experimental shows an outperforms of the proposed algorithm comparing to other algorithms in the majority of the twenty-three benchmark functions as shown in Table 3. The best results that clarify the performance of the proposed algorithm over other algorithms are obtained in the benchmark functions (F5, F6, F8, F13, F18, F22, and F23). However, in four functions (F9, F10, F14 and F16), the AOOA outperformed the proposed algorithm, while in (F9, F10 and F15), the standard SBA outperformed the proposed algorithm. Besides, SSHA outperformed the BOA-SC in (F15) functions.

6.1 Feature selection results

This section compares proposed BOA-SC algorithm as the best one among other algorithms. The standard BOA and SC are the algorithms used in this comparison in order to evaluate the performance of BOA-SC algorithm in the classification task.

For evaluating the performance of the propose algorithm, K-Nearest Neighbor (KNN) [26], and five datasets from the UCI machine learning repository [27] (Sonar, Waveform, Spect, Ionosphere, and Spambase), which are listed in Table 4 below are used.

In Table 5 the obtained classification accuracy of the shows that the proposed BOA-SC is outperformed the standard BOA and SC in all used datasets via selecting the appropriate features and maximize the classification accuracy.

Moreover, Table 5 shows the classification accuracy comparison for the five different datasets. On these datasets, in classification accuracy the proposed BOA-SC outperforms the standard SC and BOA algorithms in all five datasets.

Table 4. The Experimental of the Datasets

No.	Used Dataset	No. of Features	No. of Instances
1	Sonar	60	208
2	Waveform	21	5000
3	Spect	22	267
4	Ionosphere	34	351
5	Spambase	57	4601

Table 5. Accuracy of BOA-SC and Standard Optimization Algorithms

No.	Dataset	BOA-SC	BOA	SC
1	Sonar	**0.97156**	0.84202	0.85329
2	Waveform	**0.86221**	0.82687	0.83093
3	Spect	**0.87750**	0.76597	0.77908
4	Ionosphere	**0.94767**	0.88374	0.88091
5	Spambase	**0.93778**	0.90601	0.90626

Table 6. Accuracy of BOA-SC and BOA Variants

No.	Dataset	BOA-SC	S-bBOA [28]	DBOA [29]	CBOA [30]	OEbBOA [31]
1	Sonar	**0.97156**	0.93620	0.96138	0.94200	0.95140
2	Waveform	**0.86221**	0.74290	0.84423	0.80300	0.83100
3	Spect	**0.87750**	0.84630	0.86.546	0.82900	0.85160
4	Ionosphere	0.96567	0.90700	0.95488	**0.97700**	0.96650
5	Spambase	0.93778	0.91115	**0.94136**	0.90911	0.91400

BOA-SC finer than the BOA in classification accuracy from about 3-13% and SC about 3-12% for all datasets. The obtained results using Spect dataset yielded the highest accuracy when compared to other algorithms.

Another comparison is performed between the proposed algorithm and other BOA variants that proposed by other authors as illustrated in Table 6. Form the obtained results have observed that, the proposed algorithm achieved a high classification accuracy in three datasets (Sonar, Waveform, and Spect). However, in Spambase dataset the outcomes of the proposed algorithm are lower than DBOA and in Ionosphere dataset its lower than CBOA. In summary, the results demonstrate that BOA-SC can achieve higher classification accuracy on most datasets than other current feature selection algorithms.

In Spambase dataset, the proposed BOA-SC algorithm achieved the second best classification results with higher than CBOA result by 3.3% and lower than the highest results by 1.6% that obtained by DBOA. While in Ionosphere dataset, the proposed BOA-SC algorithm achieved the third best classification results with higher than s-bBOA result by 6% and lower than the highest results by 1.1% that obtained by CBOA.

7. Performance analysis

In this section in-depth discussion and analysing of the BOA-SC results for solving the optimization problems is presented. The discussion grouped into two parts of the performance analysis: benchmarks functions and feature selection results comparison.

In the first part, BOA-SC is superior from POA, COA, and SSHA in solving 6 out of 7 unimodal benchmark functions F1-F7. However, it outperformed NGO in 7 of 7 functions F1-F7 in mean fitness. While it outperformed SBA in 5 out of 7 of the F1-F7 unimodal benchmark functions. Besides, it's finer in 2 out of 7 functions than AOOA.

In another hand, BOA-SC outperformed POA, COA, NGO, and SSHA in 6 out of 6 multimodal benchmark functions F8-F13. While it is outperformed SBA in 5 out of 6 multimodal functions. Meanwhile it is outperformed AOOA in 3 out of 6 multimodal functions.

In the fixed dimension multimodal functions, the BOA-SC was finer than POA, COA in 10 out of 10 in mean fitness of fixed dimension multimodal functions. The performance of BOA-SC was better than NGO and SBA in 9 out of 10 functions F14-F23. While it was better than SSHA in 8 out of 10 fixed dimension multimodal functions. However, BOA-SC overcame AOOA in solving 6 out of 10 functions F14-F23. Thus, BOA-SC was finer in solving 5, 6, 8, 13, 18, 22, and 23 functions than all other compared optimization functions.

In the second analysis part, the effectiveness of the proposed BOA-SC algorithm is assed using the classification accuracy in Eq. (9).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{9}$$

Where *TP* is the true positive class that correctly classified, *TN* is true negative class that classified correctly, *FP* is false positive class that incorrectly classified, *FP* is false negative class the classified incorrectly by the classifier.

The obtained classification results in Table 6 show that BOA-SC is highly accepted due its superiority among other algorithms using 5 standard datasets. BOA-SC is outperformed S-bBOA in 5 out of 5 classification accuracy. While it was outperformed OEbBOA, DBOA and CBOA in 4 out of 5 classification accuracy.

In general, BOA-SC is outperformed S-bBOA, DBOA, CBOA, and OEbBOA in 3 out of 5 datasets which are (Sonar, Waveform, and Spect) by within classification accuracy 97%, 86%, and 87%.

8. Conclusion

An accurate optimization algorithm based on the BAO and SC algorithms for solving the features selection problem in data classification called BOA-SC is proposed. It solves the low coverage and local optima issues that BOA suffer from based on the behaviour of butterfly and the sine cosine algorithms. The proposed algorithm is evaluated using 23 benchmark functions and the results showed that BOA-SC being better than POA, NGO, COA, SBA, AOOA, and SSHA, in 5, 6, 8, 13, 18, 22, and 23 functions respectively. Besides, BOA-SC applied in classification task using five different datasets which are (Sonar, Waveform, Spect, Ionosphere, and Spambase) and achieved a high classification accuracy 97%, 86%, and 87% in three datasets (Sonar, Waveform, and Spect). The improvement in the algorithm solves the low coverage issue and give the algorithm better diversity to find other are of solutions using behaviour search of SC. The proposed algorithm can be employed in future work for solving a high dimension problem in real world via working with the modern optimization algorithms and deep learning models.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The contribution for each author in this research is as follows: First and second are responsible for conceptualization, methodology, writing original draft preparation, writing review and editing, visualization. Third author is responsible for data curation, formal analysis, investigation, and validation.

Acknowledgments

We gratefully acknowledge the, College of Pharmacy at Mustansiriyah University and College of Engineering at Al-Iraqia University for providing research facilities and resources that facilitated this study. Their assistance was instrumental in the completion of this research project.

References

[1] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier", *Computer Networks*, Vol. 174, No. 107247, pp. 1-21, 2020, doi: 10.1016/j.comnet.2020.107247.

[2] J. Yang, Z. Ye, L. Yan, and R. Wang, "Modified naive bayes algorithm for network intrusion detection based on artificial bee colony algorithm", In: *Proc. of the 2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, Lviv, Ukraine, pp. 35-40, 2018, doi: 10.1109/IDAACS-SWS.2018.8525758.

[3] I. Syarif, R. F. Afandi, and F. A. Saputra, "Feature selection algorithm for intrusion detection using cuckoo search algorithm", In: *Proc. of the 2020 International Electronics Symposium (IES)*, Surabaya, Indonesia, pp. 430-435, 2020, doi: 10.1109/IES50839.2020.9231840.

[4] A. S. Mahboob, and M. R. O. Moghaddam, "An anomaly-based intrusion detection system using butterfly optimization algorithm", In: *Proc. of the 2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, Mashhad, Iran, pp. 1-6, 2020, doi: 10.1109/ICSPIS51611.2020.9349537.

[5] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer", *Advances in Engineering Software*, Vol. 69, pp. 46-61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[6] S. Arora, and S. Singh, "Butterfly optimization algorithm: a novel approach for global optimization", *Soft Computing*, Vol. 23, pp. 715-734, 2019, doi:10.1007/s00500-018-3102-4.

[7] S. Mirjalili, "SCA: A Sine Cosine Algorithm for Solving Optimization Problems", *Knowledge-Based Systems*, Vol. 69, pp. 120-133, 2016, doi: 10.1016/j.knosys.2015.12.022.

[8] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization", In: *Proc. of the IEEE International Conference on Neural Networks*, Perth, WA, Australia, pp. 1942-1948, 1995, doi: 10.1109/ICNN.1995.488968.

[9] F. A. Zeidabadi and M. Dehghani, "POA: Puzzle Optimization Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 1, pp. 273-281, 2022, doi: 10.22266/IJIES2022.0228.25.

[10] M. Dehghani, S. Hubalovsky, and P. Trojovsky, "Northern Goshawk Optimization: A New Swarm-Based Algorithm for Solving Optimization Problems", *IEEE Access*, Vol. 9, pp. 162059-162080, 2021.

[11] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems", *Knowl Based Syst*, Vol. 259, 2023, doi: 10.1016/j.knosys.2022.110011.

[12] P. D. Kusuma, and A. Dinimaharawati, "Swarm Bipolar Algorithm: A Metaheuristic Based on Polarization of Two Equal Size Sub Swarms", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 2, pp. 377-389, 2024, doi: 10.22266/ijies2024.0430.31.

[13] M. A. Al-Sharqi, A. T. Sadiq Al-Obaidi, and S. O. Al-mamory, "Apiary Organizational-Based Optimization Algorithm: A New Nature-Inspired Metaheuristic Algorithm", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 3, pp. 783-801, 2024, doi: 10.22266/ijies2024.0630.61.

[14] P. D. Kusuma, and M. Kallista, "Swarm Space Hopping Algorithm: A Swarm-based Stochastic Optimizer Enriched with Half Space Hopping Search", *International Journal of Intelligent Engineering and Systems*, Vol. 17, No. 2, pp. 670-682, 2024, doi: 10.22266/ijies2024.0430.54.

[15] Y. Zheng, Y. Li, G. Wang, Y. Chen, Q. Xu, J. Fan, and X. Cui, "A novel hybrid algorithm for feature selection based on whale optimization algorithm", *IEEE Access*, Vol. 7, pp. 14908-14923, 2019, doi: 10.1109/ACCESS.2018.2879848.

[16] M. Tubishat, M. A. M. Abushariah, N. Idris, and I. Aljarah, “Improved whale optimization algorithm for feature selection in Arabic sentiment analysis”, *Applied Intelligence*, Vol. 49, No. 5, pp. 1688-1707, 2019, doi: 10.1007/s10489-018-1334-8.

[17] B. Wei, W. Zhang, X. Xia, Y. Zhang, F. Yu, and Z. Zhu, “Efficient feature selection algorithm based on particle swarm optimization with learning memory”, *IEEE Access*, Vol. 7, pp. 166066-166078, 2019, doi: 10.1109/ACCESS.2019.2953298.

[18] I. Aljarah, A. Z. Ala’M, H. Faris, M. A. Hassonah, S. Mirjalili, and H. Saadeh, “Simultaneous feature selection and support vector machine optimisation using the grasshopper optimisation algorithm”, *Cognit. Com- put.*, Vol. 10, pp. 478-495, 2018, doi: 10.1007/s12559-017-9542-9.

[19] B. Selvakumar and K. Muneeswaran, “Firefly algorithm-based feature selection for network intrusion detection”, *Comput. Secur.*, Vol. 81, pp. 148155, 2019, doi: 10.1016/j.cose.2018.11.005.

[20] E. Hancer, “Differential evolution for feature selection: A fuzzy wrapper-filter approach”, *Soft Comput.*, Vol. 23, No. 13, pp. 5233-5248, 2019, doi: 10.1007/s00500-018-3545-7.

[21] S. Arora, and P. Anand, “Binary butterfly optimization approaches for feature selection”, *Expert Systems with Applications*, Vol. 116, pp. 147-160, 2019, doi: 10.1016/j.eswa.2018.08.051.

[22] M. Tubishat, M. Alswaitti, S. Mirjalili, M. A. Al-Garadi, and T. A. Rana, “Dynamic butterfly optimization algorithm for feature selection”, *IEEE Access*, Vol. 8, pp. 194303-194314, 2020, doi: 10.1109/ACCESS.2020.3033757.

[23] A. A. Awad, A. F. Ali, and T. Gaber, “Feature selection method based on chaotic maps and butterfly optimization algorithm”, In: *Proc. of Joint Eur.-US Workshop Appl. Invariance Comput. Vis.*, pp. 159-169, 2020, doi: 10.1007/978-3-030-44289-7_16.

[24] B. Zhang, X. Yang, B. Hu, Z. Liu and Z. Li, “OEbBOA: A Novel Improved Binary Butterfly Optimization Approaches with Various Strategies for Feature Selection”, *IEEE Access*, Vol. 8, pp. 67799-67812, 2020, doi: 10.1109/ACCESS.2020.2985986.

[25] D. E. Goldberg, and J. H. Holland, “Genetic algorithms and machine learning”, *Mach Learn* Vol. 3, No. 2, pp. 95-99, 1988, doi: 10.1023/A:1022602019183.

[26] P. Cunningham and S. J. Delany, “K-Nearest Neighbour Classifiers - A Tutorial”, *ACM Comput. Surv*., Vol. 54, No. 128, pp. 1-25, 2022, doi: 10.1145/3459665.

[27] A. Asuncion and D.J. Newman, “UCI machine learning repository”, *School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA*, 2007, Available Online: http://www.ics.uci.edu/~mlearn/MLRepository.html.