



Binary Arithmetic Optimization Algorithm Using a New Transfer Function for Fusion Modeling

Zaynab Ayham Almishlih¹, Omar Saber Qasim¹, Zakariya Yahya Algamal²

¹Department of Mathematics, University of Mosul, Mosul, Iraq

²Department of Statistics and Informatics, University of Mosul, Mosul, Iraq

Emails: zaynabayham92@uomosul.edu.iq; omar.saber@uomosul.edu.iq; zakariya.algamal@uomosul.edu.iq

Abstract

Organizations use fusion data modeling to integrate multiple data sources and build precise representations that achieve better organizational clarity. One recent method that has proven effective in many benchmark tests is the arithmetic optimization algorithm (AOA). AOA applies basic distribution behavior to arithmetic operations such as multiplication, division, addition, and subtraction. This paper focuses on the innovative application of AOA in addressing the feature selection problem. The binary version of this algorithm (BAOA) is introduced to solve problems of binary nature. The main part of this version is the transfer function that converts a continuous search space into a discrete search space. Therefore, a new Fountain-shaped transfer function is proposed to enhance global exploration and local exploitation in the BAOA algorithm. The performance of the proposed Fountain-shaped transfer function has been compared with V-shaped and S-shaped transfer functions. Based on ten public datasets, the performance of the proposed transfer function is validated. The Experimental results show the superiority of the proposed Fountain-shaped transfer function not only in getting high classification accuracy with few selected features but also requires inexpensive computational costs.

Keywords: Feature selection; Transfer function; Binary optimization algorithm; Arithmetic optimization algorithm; Fountain-shaped transfer function; Data fusion

1. Introduction

An important class of combinatorial optimization problems is binary optimization problems (BOPs) [1-13], such as 0-1 knapsack problem (0-1KP) [9, 12], maximum coverage problem (MCP) [3, 7], incapacitated facility location problem (UFLP) [5, 6], feature selection problem (FSP) [8], knapsack problem with a single continuous variable (KPC) [4, 13], and software and hardware partitioning problem (HW/SW) [2] and so on. Our research addresses the problem of feature selection, as recently, data representation has become very important in its impact on the performance of classification models. Today, the amount of high-dimensional data collected from data acquisition processes is increasing. This is challenging for data mining due to the complexities of large dimensions [14]. In addition, the presence of unnecessary or unrelated features leads to degraded system performance and increased computational costs. As a result, feature selection becomes crucial to the data mining process. Selecting an excellent set of possible features that provide a better realization of the classification models is the primary goal in feature selection. In addition to improving the prediction accuracy, feature selection also reduces the redundant dimensions of the data [15, 16]. One of the methods used in feature selection is the arithmetic optimization algorithm AOA. Using this technique, it is possible to extract only important and reliable features.

AOA is a mathematically inspired and newly designed metaheuristic optimization technique. This technique draws its inspiration from the natural behavior of the distribution of arithmetic functions, as it was proposed by Aboalegah et al. [17]. AOA relies on basic arithmetic operations such as addition, subtraction, multiplication, and division as basic operators to explore the desired function and find the optimal solution. In reality, many problems require binary research to improve. Therefore, developing binary optimization algorithms is essential to solve these problems. Min Xu et al. in 2023 [18] have presented binary arithmetic optimization algorithms (BAOAs) using various strategies. In such algorithms, the transfer function plays an important role, as it is used to transform a

continuous search space into a discrete search space restricted between zero and one. That is, the transfer function performs a discretization process, and then a process is designed to convert that discrete space into a binary space, which is a process that makes the position of a particle either zero or one.

The study of transfer functions in binary optimization problems has a rich history, with several significant contributions by pioneering researchers. These contributions have shaped the development of this field and paved the way for the advancements we present in this paper. In 2018, Panwar et al. [19] presented an innovative approach to solve the unit commitment (UC) problem using the heuristic binary method. This method is concerned with estimating the continuous updates and values of the wolfs in order to reach the solution that the global and optimal, then comes the sigmoid transformation. Through simulation results, it is shown that the BGWO algorithm excels in solving UC problems for small and medium systems compared to other available heuristic and binary methods. In the same year, Reddy et al. [20] turned their attention to the natural binary PBUC problem by transforming the continuous position of the whale into a binary search field using various of transformation functions. They proposed three different types of binary whale optimization algorithm (BWOA), in which they used hyperbolic sequence, sigmoid transfer, and inverse tangent. Consistency of results between different variants, convergence and quality of solutions of the BWOA algorithm are discussed, and preference and statistical importance are given to the proposed method over existing methods. In 2020, Hussain et al. [21] modified the initial version of the WOA algorithm to deal with binary optimization challenges. To achieve their goal, they mapped a continuous search field to a discrete search field through two transfer functions (V-shaped and S-shaped). The performance of a new whale class optimization algorithm (BWOA) has been studied and evaluated for its annotation, and has been applied to 22 objective functions, as well as solving 3 engineering problems and a global business problem. Finally, reliable results were achieved and validated.

Although using the transfer function for separation is a powerful and universal method, two issues are still important to study and discuss in more depth [22] :

- i. From a subspace of R^n with cardinal number \aleph to discrete space $\{0, 1\}^n$ with cardinal number 2^n , all transfer functions are surjective. In theory, there are unnumbered such surjections since \aleph is much larger than 2^n . Therefore, investigating how to design efficient transfer functions to discretize BOPs is an issue worthy of discussion.
- ii. All transfer functions assign each element of the one-dimensional real vector to values 0 or 1 in turn, in order to discretization. This mapping not only increases the complexity of executing the algorithm, but also increases the time required for it. Therefore, reducing the difficulty of calculating transfer functions is a problem worthy of serious investigation.

Because of the above points, a new class of transfer functions, known as ‘‘Fountain-shaped transfer function’’, is proposed, using the benchmark datasets in the paper. In addition, a new version of the BAOA, which is based on Fountain-shaped transfer function, has been developed to efficiently solve large-scale problems (BOPs).

In this paper, the content is organized according to the following structure. Section 2 presents the arithmetic optimization algorithm, while Section 3 presents the binary arithmetic optimization algorithm. Section 4 comes up with a new proposal, which is an innovative transfer function. In Section 5, experimental results are evaluated and discussed by performing them on ten benchmark datasets. Finally, section 6 summarizes the entire.

2. Arithmetic optimization algorithm

The arithmetic optimization algorithm (AOA) is a metaheuristic algorithm recently proposed by Abualigah et al. [17]. It is mainly based on the four arithmetic operations: subtraction $-$, addition $+$, division \div , and multiplication \times . By using these operations in a set of solutions, a significant advantage can be achieved by mathematical optimization. The algorithm uses division and multiplication for the exploration process, as these operators can make significant changes. However, these operators cause increased dispersion if used in local search, so they are not suitable for local search. Therefore, to perform local search or exploitation, subtraction and addition operators are used. AOA is also a population-based technique. In this method, a set of solutions, called individuals, are initially generated randomly. These individuals are evaluated based on their suitability for solving the problem in question. The most suitable individuals are selected and are subjected to various mathematical processes to produce new individuals based on the objective required values in the previous two stages. These new individuals are then assessed for suitability, and these processes are repeated until the desired levels of improvement are reached [23].

AOA aims to generate randomly initial solutions represented by $x_i = [x_i^1, x_i^2, \dots, x_i^d]$ in a d-dimensional search space, using Eq. (1).

$$x_i^j = x_{min}^j + r(x_{max}^j - x_{min}^j) \quad i = \{1, 2, \dots, N\}, \quad j = \{1, 2, \dots, d\} \quad (1)$$

where N represents the population size, x_i represents the i^{th} solution, x_i^j represents the j^{th} dimension of the i^{th} solution, x_{max}^j , and x_{min}^j are the upper and lower bounds in the search space for j^{th} dimension, and for randomness, r represents a random value in the range from 0 to 1. In addition, the initial solution X can also be symbolized by the matrix as indicated in Eq. (2).

$$X = \begin{bmatrix} x_1^1 & \dots & x_1^d \\ \vdots & \ddots & \vdots \\ x_N^1 & \dots & x_N^d \end{bmatrix} \tag{2}$$

In making decisions regarding exploitation and exploration, the Math Optimizer Accelerated (MOA) function is calculated according to the Eq. (3). It gives a parameter that depends on the current iteration C_{Iter} , which is used in the all search phases.

$$MOA(C_{Iter}) = Min + C_{Iter} \times \left(\frac{Max - Min}{M_{Iter}} \right) \tag{3}$$

Given C_{Iter} which indicates the current repetition and M_{Iter} which indicates the maximum number of repetitions, and Min and Max , which are constants respectively, indicate the minimum and maximum MOA values. MOA is designed in such a way that it favors exploration in the initial stages and exploitation in subsequent iterations. A random number $r_1 \in [0,1]$ is generated and its value is compared to MOA. If $r_1 > MOA$, exploration is performed; Otherwise, it moves to exploitation. During the exploration stage, division and multiplication operators are used to explore the solution space. One of the division or multiplication operators with equal probability is randomly chosen to perform the exploration. The new solution is calculated according to the Eq. (4).

$$x_i^j(C_{Iter} + 1) = \begin{cases} B(x^j) \div (MOP + \epsilon) \times ((x_{max}^j - x_{min}^j) \times \mu + x_{min}^j), & r_2 < 0.5 \\ B(x^j) \times MOP \times ((x_{max}^j - x_{min}^j) \times \mu + x_{min}^j), & otherwise \end{cases} \tag{4}$$

In the next iteration, $x_i^j(C_{Iter} + 1)$ represents the j^{th} dimension of the i^{th} solution, while $B(x^j)$ represents the j^{th} dimension of the current best solution. ϵ is denoted by a small non-zero number, while μ is defined as a control parameter to adjust the search process which is set to a value of 0.5. In addition r_2 is a random number between $[0,1]$. As for MOP, it is the function of the mathematical optimizer, which is calculated using Eq. (5) in each iteration.

$$MOP(C_{Iter}) = 1 - \frac{C_{Iter}^{1/\alpha}}{M_{Iter}^{1/\alpha}} \tag{5}$$

The value of the sensitivity parameter α is set to 5.

Now, at the exploitation stage, a deep search is done for the appropriate solution, that is, the solution that is close to the best. The factors used in this research depend on addition and subtraction operations. As exploration, operators are also equally likely to choose exploitation. New solutions are calculated according to the following Eq. (6).

$$x_i^j(C_{Iter} + 1) = \begin{cases} B(x^j) - MOP \times ((x_{max}^j - x_{min}^j) \times \mu + x_{min}^j), & r_3 < 0.5 \\ B(x^j) + MOP \times ((x_{max}^j - x_{min}^j) \times \mu + x_{min}^j), & otherwise \end{cases} \tag{6}$$

where the value of r_3 is set as a random number in the range between 0 and 1.

3. Binary arithmetic optimization algorithm

In this section, binary variants proposed as binary arithmetic optimization algorithm with V-shaped transfer function BAOA-V and binary arithmetic optimization algorithm with S-shaped transfer function BAOA-S are presented in AOA. According to the research of Kennedy and Eberhart [24], the transfer function (TF) is effectively used to convert the continuous algorithm into binary form. A transfer function is used to convert given continuous vectors into binary vectors, setting the probability that the element in the solution vector changes to 0 or 1 based on the value of the step vector. Two families of transfer functions: V-shaped (hyperbolic tangent) and S-shaped (sigmoid) are applied to convert AOA to BAOA-V and BAOA-S respectively using Eqs. (7) and (8) in general.

$$TF_V(x_i^j) = (|tanh(x_i^j)|) \tag{7}$$

$$TF_S(x_i^j) = \frac{1}{1 + e^{x_i^j}} \tag{8}$$

where x_i^j represents the j^{th} dimension of the i^{th} solution. In BAOA-V / BAOA-S, the position of i^{th} solution is updated using Eqs. (4) and (6), based on the values of MOA, r_2 , and r_3 . The resulting solution acts as a step function and consists of real values. To convert it to a binary vector, TF_V or TF_S is applied to the generated solution.

Currently, a random number r is generated from the standard uniform distribution. The real values in the solution vectors are set to either 0 or 1 based on the value of TF and the random number r and according to Eqs. (9) and (10) for V-shaped and S-shaped TF respectively.

$$x_i^{i,b}(C_{Iter} + 1) = \begin{cases} \sim x_i^j, r < TF_V(x_i^j(C_{Iter} + 1)) \\ x_i^j, r \geq TF_V(x_i^j(C_{Iter} + 1)) \end{cases} \quad (9)$$

$$x_i^{i,b}(C_{Iter} + 1) = \begin{cases} 1, r < TF_S(x_i^j(C_{Iter} + 1)) \\ 0, r \geq TF_S(x_i^j(C_{Iter} + 1)) \end{cases} \quad (10)$$

where $x_i^{i,b}(C_{Iter} + 1)$ is the binary vector, it is obtained by setting the step or real value solution of $x_i^{i,b}(C_{Iter} + 1)$ using TF . Pseudo-code of BAOA is summarized in Algorithm 1.

Algorithm 1: pseudo code of BAOA

```

1. Load dataset, split into testing and training sets
2. AOA's parameters initialization  $\mu, \alpha$ 
3. initialized Randomly the positions of solutions ( $i=1, \dots, n$ )
4. While ( $C_{Iter} < M_{Iter}$ ) do
5.   Evaluate the fitness function.
6.   Get the obtained finest solution so far.
7.   Modify the values of MOA utilizing Eq. (3).
8.   Modify the values of MOP utilizing Eq. (5).
9.   For  $i=1$  to solutions do
10.    For  $j=1$  to solutions do
11.     Randomly generate values within the range  $[0, 1]$ : ( $r_1, r_2,$  and  $r_3$ )
12.     If  $r_1 > MOA$  then
13.      The stage of exploration
14.      If  $r_2 > 0.5$  then
15.       a) The division operation is applied
16.       Modify the locations of the  $i^{th}$  solutions applying the 1st rule in Eq. (4).
17.       Binarize the locations of the 1st rule of the  $i^{th}$  solutions applying Eq. (10).
18.      Else
19.       b) The multiplication operation is applied
20.       Modify the locations of the  $i^{th}$  solutions by applying the 2nd rule in Eq. (4).
21.       Binarize the locations of the 2nd rule of the  $i^{th}$  solutions applying Eq. (10).
22.      End if
23.     Else
24.      The stage of exploitation
25.      If  $r_3 > 0.5$  then
26.       a) The subtraction operation is applied
27.       Modify the locations of the  $i^{th}$  solutions applying the 1st rule in Eq. (6).
28.       Binarize the locations of the 1st rule of the  $i^{th}$  solutions applying Eq. (10).
29.      Else
30.       b) The addition operation is applied
31.       Modify the locations of the  $i^{th}$  solutions applying the 2nd rule in Eq. (6).
32.       Binarize the locations of the 2nd rule of the  $i^{th}$  solutions applying Eq. (10).
33.      End if
34.     End if
35.    End for
36.   End for
37.    $C_{Iter} = C_{Iter} + 1$ 
38. End while
39. Extract the solution of finest (maximum accuracy, minimum features)

```

In BAOA, features are selected by randomly generating a vector of binary values (containing one and zero) of the same length as the feature vector, where the feature corresponding to the value of one is selected, while the feature corresponding to zero is neglected, as shown in Figure 1.

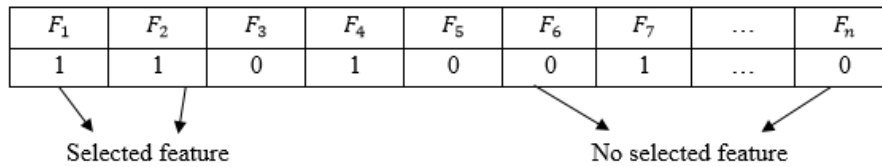


Figure 1. A representation of the features in BAOA [25].

4. Proposed Fountain-Shaped transfer function

Given the vital role of transfer functions in the discretization of BOPs, it highlights the importance of exploring and discussing the design of transfer functions in their diversity and practical effectiveness. And for this reason, the existing V-shaped transfer functions and S-shaped transfer functions [26-29] are firstly reviewed in this section. Then a new class of transfer functions is proposed, which are known as Fountain-shaped transfer function. Based on this idea, a new binary arithmetic optimization algorithm BAOA is proposed to solve binary optimization problems.

Table 1: V-Shaped and S-Shaped transfer functions.

	Name	Transfer Function
V-shaped	V_1	$V_1(x) = erf((\sqrt{\pi}/2)x) $
	V_2	$V_2(x) = tanh(x) $
	V_3	$V_3(x) = (x)/\sqrt{1+x^2} $
	V_4	$V_4(x) = (2/\pi)arctan((\pi/2)x) $
S-shaped	S_1	$S_1(x) = 1/(1+e^{-2x})$
	S_2	$S_2(x) = 1/(1+e^{-x})$
	S_3	$S_3(x) = 1/(1+e^{-x/2})$
	S_4	$S_4(x) = 1/(1+e^{-x/3})$

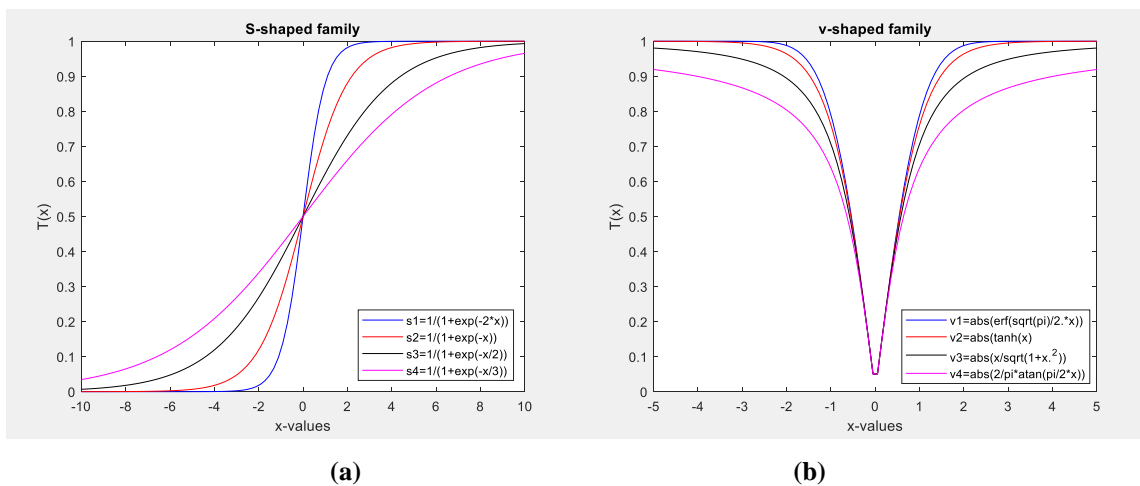


Figure 2. Transfer Functions (a) S-shaped and (b) V-shaped.

Basically, V-shaped and S-shaped transfer functions [26, 27] are the most commonly used transfer functions. Among them, the more representative V-shaped transfer functions are V1–V4 and the more representative S-shaped transfer functions are S1–S4. The calculation formulas for these functions are given in Table 1, while the function curves are given in Fig.2.

Figure 2 makes it clear that S-shaped transfer functions are a type of elementary function produced by an exponential function. They contain a single calculation formula, which is $S(x) = \frac{1}{1+e^{bx+a}}$, where b and a are real numbers, and $b \neq 0$. It is generally referred to as the S-shaped transfer function since this kind of function has a "S" shaped curve. Also, V-shaped transfer functions, their curves show a "V" shape, so they are called V-shaped transfer functions, even though there is no unified mathematical formula for them, and they are not all elementary functions.

As we know, there are an infinite number of transformations from the real dimensional space R^n or its subspace to the discrete space $\{0, 1\}^n$. This transformation is often used as transfer functions for discretizing BOPs that still lack relevant theoretical results. Hence, individuals usually design and apply a transfer functions based on their experience to discretize BOPs. This is a recombination method using existing transfer functions, however, has a certain reference and guiding significance. Nowadays, existing transfer functions and their applicability [26-28, 30] are mainly evaluated according to the results obtained when calculating the discrete evolutionary algorithm they designed to solve BOPs. Based on this fact, it becomes clear that trying to develop a variety of transfer functions can be very beneficial, not only does this provide more options for the discretization of BOPs, but it also contributes to the accumulation of experience required to build a comprehensive theory for evaluating those functions. Since V-shaped transfer functions and S-shaped transfer functions [26-29] consist of a trigonometric function, an anti-trigonometric function, an exponential function, or a non-elementary function, their calculations require relatively more. This will have a negative impact on the execution time of the discretized evolutionary algorithm. For this purpose, a new transfer function is proposed. Since the function curve of the new transfer function resembles the shape of a fountain, it is called a Fountain-shaped transfer function and is defined as follows:

$$F(x) = \left| \frac{x}{(1 + |x|^2)^{1/2}} \right| \quad (11)$$

5. Simulation Results

This section provides precise details about the different data sets, and numerical results with statistical comparisons between them.

5.1. Datasets description

Tests of the proposed methods are conducted on ten benchmark datasets taken from the UCI repository [31]. The data includes a variety of instances from 148 to 9945, and a variety of features from 17 to 72 during the feature selection process. Each representation and detailed description of each dataset are shown in Table 2. Each data set is divided into 70% of the total samples as a training set, and 30% of the samples as a test set. All BAOA algorithms (S1–S4, V1–V4, and F) were implemented using MATLAB and processed on a 4-core Intel Core i5 CPU (8 GB RAM, 4 GHz processor).

Table 2: Description of datasets used

No.	Name	Instances	Features
1	Central Nervous System (CNS)	7129	60
2	Brain Tumor	7129	40
3	Children Tumors	9945	23
4	Leukemia-4c	7129	72
5	Hepatitis	155	19
6	Primary Tumor	339	17
7	Ionosphere	351	34
8	Soybean	307	35
9	Horse Colic	368	27
10	Lymphography	148	18

5.2. Evaluation of proposed Fountain-Shaped transfer function

In BAOA, KNN classifier is used to achieve classification accuracy, where the K value is set as 5, and the fitness function is defined as follows:

$$fitness = 0.8 ACC + 0.2 \frac{|q|}{|p|} \quad (12)$$

In the first phase of the experiment, the performance of the BAOA algorithm was verified on ten datasets. Common evaluation criteria are measured, including classification accuracy, number of features identified (feature size), and processing time (in seconds). The 9 proposed BAOA methods (S1–S4, V1–V4, and F) were compared to study the best binary version of AOA in classification accuracy, feature selection, and processing time. The following tables show the above, with the best results highlighted in bold.

Tables 3 and 4 presents the averaged classification accuracy of the used and proposed transfer functions for all dataset for training and testing dataset, respectively. Table 5 presents the average number of selected features. In addition, the standard deviation values are also provided. According to the classification accuracy, as we can see from Tables 3 and 4, the proposed fountain transfer function, F, achieves better results for all data sets. It achieved the highest classification accuracy comparing to S1, S2, S3, S4, V1, V2, V3, and V4. From Table 4, F function obtains the top classification accuracy at 95.33%, 92.55%, 85.09%, 98.73%, 95.56%, 88.15%, 96.16%, 95.42%, 94.01%, and 93.15% in the Central Nervous System, Brain Tumor, Children Tumors, Leukemia-4c, Hepatitis, Primary Tumor, Ionosphere, Soybean, Horse Colic, and Lymphography datasets, respectively. The stability of the proposed fountain transfer function is shown by small standard deviations in its results, which are smaller than those of S1, S2, S3, S4, V1, V2, V3, and V4 in all datasets.

As for the selected features, there is no doubt to say that with the proposed fountain transfer function for BAOA algorithm, it successfully addressed feature selection problems in all the datasets as shown in Table 5. In the same manner, the proposed fountain transfer function F, took lesser number of features than S1, S2, S3, S4, V1, V2, V3 and V4 transfer functions meaning thereby that the insignificant impact information as well as other irrelevant impact features were rejected. Looking at the standard deviation depending on the results of Table 5 we can mention that all of them are very low, that is why F shows relatively high stability.

Table 3: Average classification accuracy for the training set.

Dataset	S1	S2	S3	S4	V1	V2	V3	V4	F
Central Nervous System (CNS)	90.10±0.127	91.30±0.122	91.24±0.116	90.83±0.124	93.00±0.118	90.79±0.125	90.10±0.117	93.92±0.120	95.45±0.093
Brain Tumor	87.67±0.122	87.21±0.128	88.30±0.124	87.54±0.126	87.81±0.118	88.90±0.119	88.21±0.125	89.00±0.127	92.67±0.086
Children Tumors	83.36±0.120	83.38±0.123	83.25±0.126	83.80±0.119	83.14±0.124	83.65±0.125	83.24±0.118	83.65±0.127	87.21±0.087
Leukemia-4c	95.85±0.124	95.85±0.118	94.70±0.125	94.70±0.126	93.70±0.127	94.70±0.120	95.55±0.119	95.57±0.128	98.85±0.093
Hepatitis	92.37±0.126	92.68±0.118	91.53±0.127	92.69±0.125	93.12±0.128	93.39±0.124	90.12±0.123	93.16±0.120	95.68±0.089
Primary Tumor	85.12±0.123	84.28±0.119	84.68±0.128	86.28±0.120	84.12±0.121	85.81±0.127	86.15±0.125	86.16±0.118	88.27±0.094
Ionosphere	92.85±0.121	92.83±0.126	93.74±0.119	90.27±0.128	94.28±0.123	92.75±0.117	94.92±0.122	94.28±0.120	96.28±0.095
Soybean	93.56±0.128	90.17±0.119	91.37±0.120	90.16±0.127	90.25±0.125	92.45±0.118	90.13±0.124	90.53±0.126	95.54±0.089
Horse Colic	91.35±0.122	89.13±0.126	92.24±0.118	90.35±0.127	91.00±0.120	90.57±0.123	90.24±0.128	90.25±0.119	94.13±0.096
Lymphography	85.24±0.120	88.24±0.123	90.18±0.126	87.28±0.119	90.35±0.124	90.36±0.125	86.56±0.118	85.26±0.127	93.53±0.087

Table 4: Average classification accuracy for the testing set.

Dataset	S1	S2	S3	S4	V1	V2	V3	V4	F
Central Nervous System (CNS)	89.98±0.102	91.18±0.120	91.12±0.098	90.71±0.117	92.88±0.113	90.67±0.124	89.98±0.119	93.80±0.122	95.33±0.088
Brain Tumor	87.55±0.119	87.09±0.115	88.18±0.121	87.42±0.114	87.69±0.118	88.78±0.122	88.09±0.120	88.88±0.123	92.55±0.086

Children Tumors	83.24±0.116	83.26±0.122	83.13±0.119	83.68±0.123	83.02±0.120	83.53±0.118	83.12±0.125	83.53±0.091	85.09±0.089
Leukemia-4c	95.73±0.121	95.73±0.118	94.58±0.122	94.58±0.119	93.58±0.116	94.58±0.123	95.43±0.117	95.45±0.120	98.73±0.092
Hepatitis	92.25±0.123	92.56±0.119	91.41±0.122	92.57±0.118	93.00±0.120	93.27±0.117	90.00±0.123	93.04±0.119	95.56±0.092
Primary Tumor	85.00±0.120	84.16±0.118	84.56±0.123	86.16±0.119	84.00±0.122	85.69±0.120	86.03±0.119	86.04±0.118	88.15±0.091
Ionosphere	92.73±0.122	92.71±0.117	93.62±0.121	90.15±0.120	94.16±0.119	92.63±0.123	94.80±0.122	94.16±0.118	96.16±0.094
Soybean	93.44±0.123	90.05±0.118	91.20±0.122	90.04±0.119	90.13±0.121	92.33±0.120	90.01±0.122	90.41±0.118	95.42±0.090
Horse Colic	91.23±0.120	89.01±0.123	92.12±0.118	90.23±0.122	90.88±0.121	90.45±0.123	90.12±0.119	90.13±0.120	94.01±0.094
Lymphography	85.24±0.122	88.24±0.119	90.18±0.123	87.26±0.120	90.18±0.118	87.28±0.124	90.35±0.119	90.36±0.120	93.15±0.091

Table 5: Average Number of Selected Features.

Dataset	S1	S2	S3	S4	V1	V2	V3	V4	F
Central Nervous System (CNS)	36±0.114	35±0.116	36±0.109	36±0.110	31±0.120	33±0.117	37±0.118	37±0.123	26±0.085
Brain Tumor	29±0.112	29±0.109	27±0.118	29±0.120	26±0.113	27±0.114	25±0.115	25±0.110	20±0.086
Children Tumors	12±0.115	11±0.119	14±0.122	11±0.110	15±0.117	15±0.120	12±0.116	11±0.121	9±0.093
Leukemia-4c	30±0.120	30±0.115	32±0.118	31±0.109	30±0.122	31±0.123	31±0.111	33±0.112	25±0.084
Hepatitis	10±0.105	14±0.120	10±0.109	10±0.116	11±0.108	13±0.117	11±0.114	10±0.122	6±0.093
Primary Tumor	8±0.108	8±0.116	9±0.112	9±0.117	7±0.105	8±0.120	10±0.114	7±0.122	2±0.083
Ionosphere	20±0.113	20±0.122	25±0.115	21±0.111	23±0.120	23±0.116	21±0.109	20±0.121	18±0.088
Soybean	21±0.116	22±0.119	21±0.113	21±0.110	23±0.118	21±0.122	22±0.117	22±0.123	20±0.095
Horse Colic	15±0.101	16±0.123	15±0.115	13±0.112	13±0.119	17±0.122	18±0.118	17±0.124	11±0.096
Lymphography	11±0.110	14±0.119	17±0.105	19±0.120	11±0.116	11±0.118	11±0.121	11±0.122	5±0.084

In terms of processing time, Figure 3 shows the average processing time measured in seconds. Apparently, F achieves comparable performance to S1, S2, S3, S4, V1, V2, V3 and V4 transfer functions. It is the most efficient transfer function, as it requires fewer seconds to achieve high classification accuracy with few features.

Further, Figure 4 shows the convergence curves for the average fitness functions of the proposed Fountain-shaped transfer function for all datasets. Based on the general analysis of BAOA algorithm exploration, it has features, which indicates that it represents fast convergence rate when using the Fountain-shaped transfer function as compared to the other comparative transfer functions. It implies that there is enhancement within the BAOA in the three phases to select the promising characteristics and a high classification accuracy rate of the most effective features. In the exploration phase, BAOA with Fountain-shaped transfer function is used to navigate a good search of graduates having gained sufficient experience to identify promising areas in the solutions space. Then the active exploration process slowly transforms to frameborder allocation and then BAOA with Fountain-shaped transfer function travels around the good solution to get the best solution in the exploitation phase.

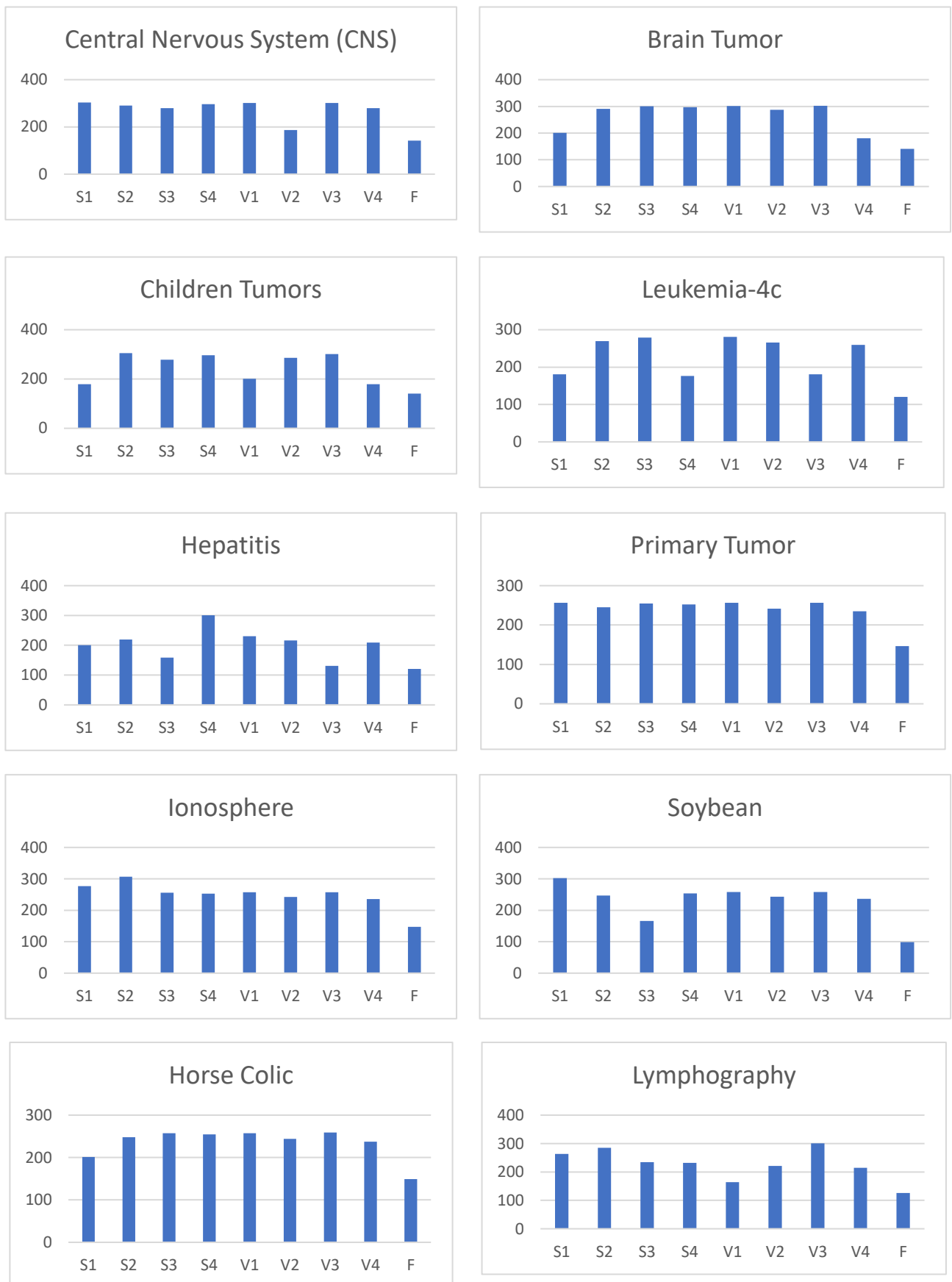
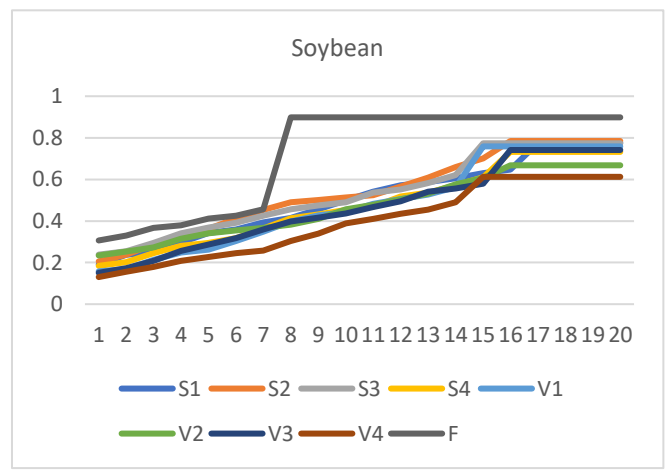
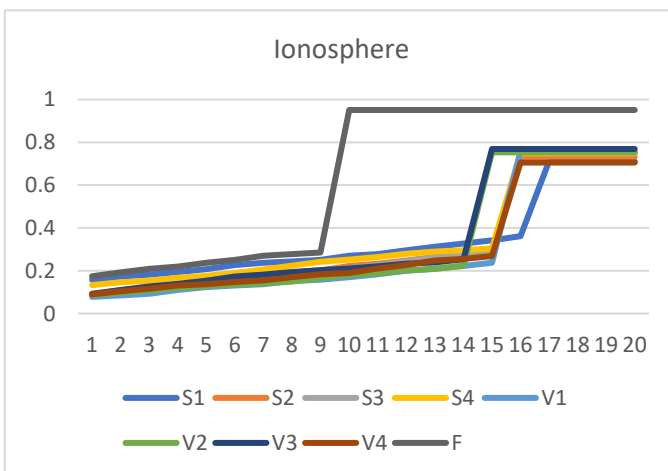
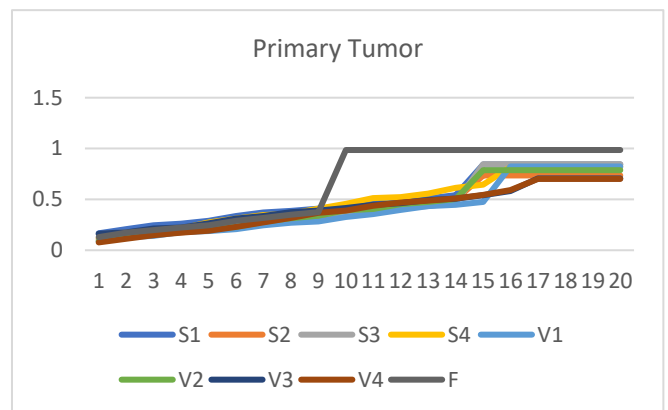
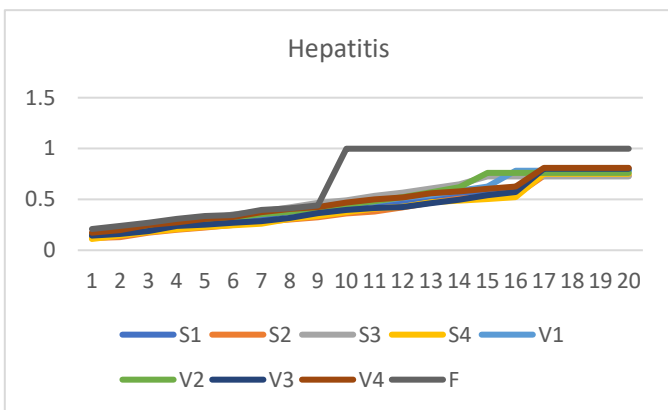
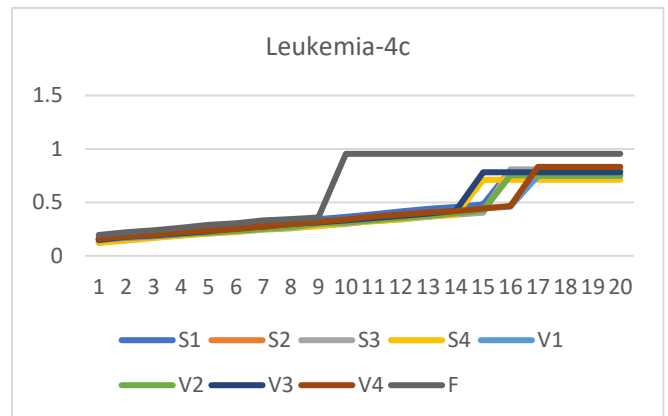
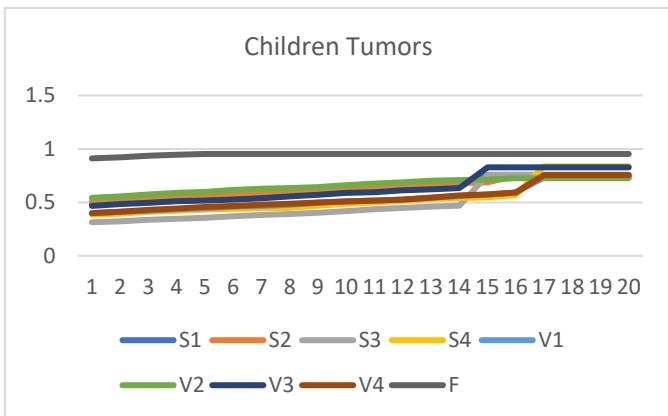
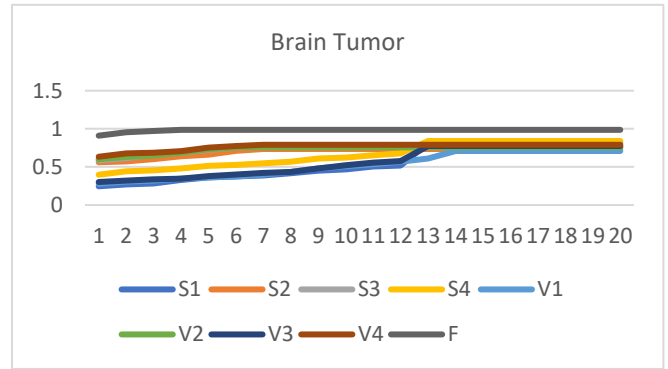
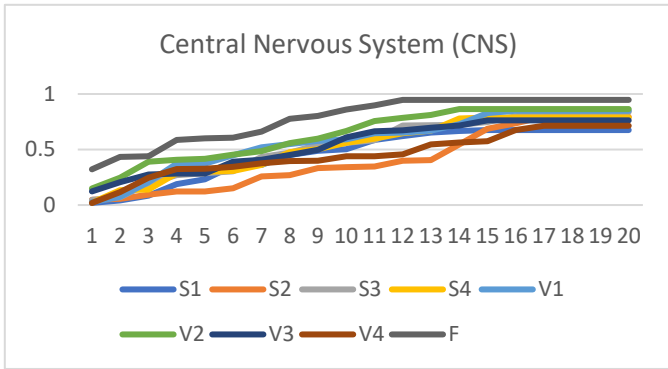


Figure 3. Average processing time (seconds).



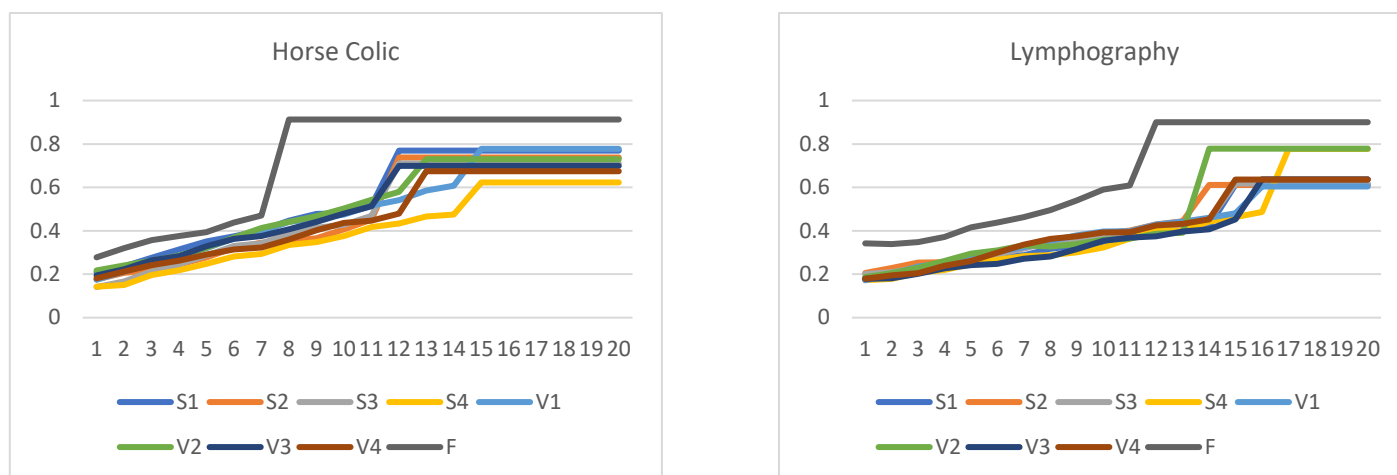


Figure 4. Convergence curves of proposed algorithms on datasets.

6. Conclusion

To enhance global exploration and local exploitation in the BAOA algorithm, a new Fountain-shaped transfer function is proposed in this paper. Ten benchmark datasets in the UCI Machine Learning Repository were used to validate the proposed algorithms. The performance of the proposed algorithms is evaluated based on the best value of feature size, classification accuracy, and processing time to implement the algorithm. Among the BAOA algorithms, BAOA with Fountain-shaped transfer function provided the optimal performance in the present work. Moreover, experimental results indicate that BAOA with Fountain-shaped transfer function can typically achieve the highest classification accuracy, smallest feature size, and lowest algorithm execution processing time when competing in feature selection tasks.

References

- [1] A. P. Punnen, P. Pandey, and M. Friesen, "Representations of quadratic combinatorial optimization problems: A case study using quadratic set covering and quadratic knapsack problems," *Comput. Oper. Res.*, vol. 112, p. 104769, 2019.
- [2] J. Staunstrup and W. Wolf, *Hardware/software co-design: Principles and practice*. Springer, 2013.
- [3] C. Revelle and K. Hogan, "The maximum reliability location problem and α -reliable p-center problem: Derivatives of the probabilistic location set covering problem," *Ann. Oper. Res.*, vol. 18, no. 1, pp. 155–173, 1989.
- [4] H. Marchand and L. A. Wolsey, "The 0-1 knapsack problem with a single continuous variable," *Math. Program.*, vol. 85, pp. 15–33, 1999.
- [5] A. A. Kuehn and M. J. Hamburger, "A heuristic program for locating warehouses," *Manag. Sci.*, vol. 9, no. 4, pp. 643–666, 1963.
- [6] M. S. Kiran, "The continuous artificial bee colony algorithm for binary optimization," *Appl. Soft Comput.*, vol. 33, pp. 15–23, 2015.
- [7] B. Jayalakshmi and A. Singh, "A hybrid artificial bee colony algorithm for the cooperative maximum covering location problem," *Int. J. Mach. Learn. Cybern.*, vol. 8, pp. 691–697, 2017.
- [8] R. A. Ibrahim et al., "An opposition-based social spider optimization for feature selection," *Soft Comput.*, vol. 23, pp. 13547–13567, 2019.
- [9] D. Du, K.-I. Ko, and X. Hu, *Design and analysis of approximation algorithms*, vol. 62. Springer, 2012.
- [10] T. Dokeroglu, E. Sevinc, and A. Cosar, "Artificial bee colony optimization for the quadratic assignment problem," *Appl. Soft Comput.*, vol. 76, pp. 595–606, 2019.
- [11] H. Djelloul, S. Sabba, and S. Chikhi, "Binary bat algorithm for graph coloring problem," in *Proc. 2014 2nd World Conf. Complex Syst. (WCCS)*, 2014, pp. 1–6.
- [12] G. B. Dantzig, "Discrete-variable extremum problems," *Oper. Res.*, vol. 5, no. 2, pp. 266–288, 1957.
- [13] M. Büther and D. Briskorn, "Reducing the 0-1 knapsack problem with a single continuous variable to the standard 0-1 knapsack problem," *Int. J. Oper. Res. Inf. Syst.*, vol. 3, no. 1, pp. 1–12, 2012.
- [14] B. Tran, B. Xue, and M. Zhang, "Genetic programming for multiple-feature construction on high-dimensional classification," *Pattern Recognit.*, vol. 93, pp. 404–417, 2019.
- [15] H. Jia et al., "Spotted hyena optimization algorithm with simulated annealing for feature selection," *IEEE Access*, vol. 7, pp. 71943–71962, 2019.

- [16] C. E. M. Qiu, "A novel multi-swarm particle swarm optimization for feature selection," *Genet. Program.*, vol. 20, no. 4, pp. 503–529, 2019.
- [17] L. Abualigah et al., "The arithmetic optimization algorithm," *Comput. Methods Appl. Mech. Eng.*, vol. 376, p. 113609, 2021.
- [18] M. Xu et al., "Binary arithmetic optimization algorithm for feature selection," *Soft Comput.*, 2023.
- [19] L. K. Panwar et al., "Binary grey wolf optimizer for large scale unit commitment problem," *Swarm Evol. Comput.*, vol. 38, pp. 251–266, 2018.
- [20] S. Reddy K et al., "Binary whale optimization algorithm: A new metaheuristic approach for profit-based unit commitment problems in competitive electricity markets," *Eng. Optim.*, vol. 51, no. 3, pp. 369–389, 2019.
- [21] A. G. Hussien et al., "New binary whale optimization algorithm for discrete optimization problems," *Eng. Optim.*, vol. 52, no. 6, pp. 945–959, 2020.
- [22] Y. He et al., "Novel binary differential evolution algorithm based on Taper-shaped transfer functions for binary optimization problems," *Swarm Evol. Comput.*, vol. 69, p. 101022, 2022.
- [23] G. Hu et al., "An enhanced hybrid arithmetic optimization algorithm for engineering applications," *Comput. Methods Appl. Mech. Eng.*, vol. 394, p. 114901, 2022.
- [24] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. 1997 IEEE Int. Conf. Syst. Man Cybern.*, 1997, pp. 4104–4108.
- [25] O. S. Qasim and Z. Y. Algamal, "Feature selection using different transfer functions for binary bat algorithm," *Int. J. Math. Eng. Manag. Sci.*, vol. 5, no. 4, p. 697, 2020.
- [26] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary particle swarm optimization," *Swarm Evol. Comput.*, vol. 9, pp. 1–14, 2013.
- [27] M. Mafarja et al., "S-shaped vs. V-shaped transfer functions for ant lion optimization algorithm in feature selection problem," in *Proc. Int. Conf. Future Netw. Distrib. Syst.*, 2017.
- [28] M. Mafarja et al., "Binary dragonfly optimization for feature selection using time-varying transfer functions," *Knowl.-Based Syst.*, vol. 161, pp. 185–204, 2018.
- [29] J. Liu et al., "A binary differential search algorithm for the 0–1 multidimensional knapsack problem," *Appl. Math. Model.*, vol. 40, no. 23–24, pp. 9788–9805, 2016.
- [30] K. K. Ghosh et al., "Binary social mimic optimization algorithm with x-shaped transfer function for feature selection," *IEEE Access*, vol. 8, pp. 97890–97906, 2020.
- [31] A. Frank, "UCI machine learning repository," 2010. [Online]. Available: <http://archive.ics.uci.edu/ml>.