



Article

# Mutational Slime Mould Algorithm for Gene Selection

Feng Qiu <sup>1</sup>, Pan Zheng <sup>2</sup> , Ali Asghar Heidari <sup>1</sup>, Guoxi Liang <sup>3</sup> , Huiling Chen <sup>1,\*</sup> , Faten Khalid Karim <sup>4</sup>, Hela Elmannai <sup>5</sup> and Haiping Lin <sup>6</sup>

- <sup>1</sup> Department of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China  
<sup>2</sup> Information Systems, University of Canterbury, Christchurch 8014, New Zealand  
<sup>3</sup> Department of Information Technology, Wenzhou Polytechnic, Wenzhou 325035, China  
<sup>4</sup> Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia  
<sup>5</sup> Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia  
<sup>6</sup> Department of Information Engineering, Hangzhou Vocational & Technical College, Hangzhou 310018, China  
\* Correspondence: chenhuiling.jlu@gmail.com

**Abstract:** A large volume of high-dimensional genetic data has been produced in modern medicine and biology fields. Data-driven decision-making is particularly crucial to clinical practice and relevant procedures. However, high-dimensional data in these fields increase the processing complexity and scale. Identifying representative genes and reducing the data's dimensions is often challenging. The purpose of gene selection is to eliminate irrelevant or redundant features to reduce the computational cost and improve classification accuracy. The wrapper gene selection model is based on a feature set, which can reduce the number of features and improve classification accuracy. This paper proposes a wrapper gene selection method based on the slime mould algorithm (SMA) to solve this problem. SMA is a new algorithm with a lot of application space in the feature selection field. This paper improves the original SMA by combining the Cauchy mutation mechanism with the crossover mutation strategy based on differential evolution (DE). Then, the transfer function converts the continuous optimizer into a binary version to solve the gene selection problem. Firstly, the continuous version of the method, ISMA, is tested on 33 classical continuous optimization problems. Then, the effect of the discrete version, or BISMA, was thoroughly studied by comparing it with other gene selection methods on 14 gene expression datasets. Experimental results show that the continuous version of the algorithm achieves an optimal balance between local exploitation and global search capabilities, and the discrete version of the algorithm has the highest accuracy when selecting the least number of genes.

**Keywords:** gene selection; slime mould algorithm; Cauchy mutation; crossover and mutation; medical diagnosis



**Citation:** Qiu, F.; Zheng, P.; Heidari, A.A.; Liang, G.; Chen, H.; Karim, F.K.; Elmannai, H.; Lin, H. Mutational Slime Mould Algorithm for Gene Selection. *Biomedicines* **2022**, *10*, 2052. <https://doi.org/10.3390/biomedicines10082052>

Academic Editors: Thomas Mohr and Alexander Tolios

Received: 22 June 2022

Accepted: 16 August 2022

Published: 22 August 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

Microarray technology [1,2] is a new analytical tool that simultaneously measures the expression levels of thousands of genes in a single experiment, greatly helping researchers understand disease at the genetic level. However, the gene expression data are all high-dimensional, and the number of features is much larger than the number of samples [3,4]. A large number of unrelated and complex features will reduce the computational performance and waste computational resources, which is not conducive to the classification of gene expression [5–7]. The application of feature selection in genes, namely gene selection, is a screening technology to reduce unrelated genes and gene dimensions [8–10]. Through this technology, feature size can be effectively reduced, and classification performance can be improved [11–13].

Feature selection is an essential technology in data processing and machine learning [7,14]. The essence is to pick out the relatively optimal features from the raw data so that the

data go from high to low dimensions [15,16]. The commonly used (classical) feature selection methods can be divided into filter, wrapper, embedded, and hybrid methods [17]. Filter methods typically select features independently and evaluate individual features without providing a practical evaluation across feature subsets, which may ignore the correlation between feature combinations [18–21]. Because it does not use any algorithm, the computation is less, leading to the failure to find the optimal gene subset. The wrapper method relies on the classification algorithm to select the feature subset, which can obtain the ideal effect, but the calculation cost is high [22–24]. Embedded methods usually use some machine learning algorithms and models for training and then select the best feature subset through the classifier algorithm [25]. When extracting features, it needs to train the model to automatically obtain the corresponding threshold value, which is realized by the algorithm with a built-in feature selection method. The hybrid method combines the advantages of the filter and wrapper methods to determine the optimal subset of a given cardinality by independent measurement and select the final subset in the optimal subset using a mining algorithm [26–29].

Optimization methods can be approximated or deterministic [30], and their model can be single objective and multi-objective, including multiple objective algorithms that can deal with multiple objectives at once [31,32]. In recent years, since the wrapper method based on the meta-heuristic algorithm or its variants can find an acceptable solution, that is, the approximate optimal subset of features, it has been widely used in feature selection [33,34]. In this study, we tried to use an improved slime mould algorithm (SMA), called ISMA, to develop an efficient wrapper gene selection method for finding the smallest feature subset. The optimization algorithm proposed in this paper is aimed at the shortcomings and characteristics of the original SMA, using the main operators of the SMA, but some of the operators use binary conversion to adapt to the genetic selection problem because the original version of the algorithm was created to solve the continuity problem. SMA is a new meta-heuristic algorithm recently proposed by Li et al. [35], which is used to deal with continuous global optimization and engineering design problems. It is an optimal algorithm used to simulate the dynamic vibration behavior of slime mould in dispersive foraging and food searching. This method consists of three search patterns with different morphologic variations, which are mathematically expressed using a unique model. The mathematical model of SMA mainly adopts the adaptive weight to simulate the propagation wave of the biological oscillator and generates positive feedback during the optimization process, which helps form an optimal exploration trajectory of the optimal solution with good searchability. In addition, the survey and results confirm that SMA achieves a balanced competitive advantage between global exploration and local exploitation. Notably, it shows a superior tendency towards local exploitation. With the help of adaptive weighting and efficient and reasonable structure, SMA can provide significantly enhanced performance compared to many recognized advanced algorithms, such as whale optimization algorithm (WOA), gray wolf optimization (GWO), grasshopper optimization algorithm (GOA), moth-flame optimization (MFO), ant lion optimizer (ALO), bat algorithm (BA), salp swarm algorithm (SSA), sine cosine algorithm (SCA), particle swarm optimization (PSO), and differential evolution (DE) [36]. Other examples include biogeography-based learning particle swarm optimization (BLPSO) [37], comprehensive learning particle swarm optimizer (CLPSO) [38], improved grey wolf optimization (IGWO)[39], and binary whale optimization algorithm (BWOA) [40], etc. Therefore, SMA has been applied in engineering design problems [35,41], solar photovoltaic cell parameter estimation [42,43], multi-spectral image segmentation [44], numerical optimization [45], prediction problems [46,47], support vector regression parameter adjustment [48] and other aspects. This algorithm is a sufficiently effective meta-heuristic optimization algorithm, but it may have the shortcoming of local optimal convergence and slow convergence speed when dealing with some complex problems. Therefore, there are some challenges in improving the optimization capability of SMA and expanding its application value.

In order to alleviate the shortcomings of traditional SMA and strengthen the trend of coordination between global exploration and local exploitation, an advanced SMA variant was proposed based on the reasonable integration of Cauchy mutation (CM) and crossover mutation (MC). After the initial search agent is generated, the solution is updated in three phases. First, execute the search process of SMA and update the search agent. The Cauchy mutation strategy is adopted in the second stage to adjust the SMA-based search agent. Finally, the optimal search agent is selected from the previous generation of search agents through a crossover mutation strategy. In addition, we convert the continuous version of ISMA to a discrete ISMA with a transfer function. Tests on gene expression data sets have shown that BISMA has significant advantages over some advanced gene selection methods and is very effective. It shows that ISMA can effectively solve high-dimensional complex gene problems, which makes improving SMA more valuable.

The main contributions in this paper can be summarized as follows:

- (1) An improved slime mould algorithm (ISMA) is proposed to solve continuous global optimization problems and high-dimensional gene selection problems.
- (2) The performance of the ISMA algorithm is verified by comparing it with several famous optimization algorithms.
- (3) Different transfer functions are used to transform the proposed ISMA into a discrete version of BISMA, and they are compared to choose the most suitable transfer function for the binary ISMA optimizer.
- (4) The optimal BISMA version was selected as a gene selection optimizer to select the optimal gene subset from the gene expression data set.
- (5) The performance of the selected method is verified by comparing it with several other advanced optimizers.

The rest of this article is as follows: The second part introduces the work of gene selection and meta-heuristic algorithms. In the third section, Cauchy mutation and a crossover mutation strategy based on the DE algorithm are introduced in detail, and ISMA is proposed. In the fourth section, a series of comparative experiments between ISMA and other similar algorithms are introduced. In the fifth part, we design the structure of wrapper gene selection for discrete ISMA. In the sixth part, we discuss the application of BISMA and other related algorithms in gene selection. In the seventh part, we discuss a summary of the proposed work as well as its shortcomings and implications. The eighth part gives a brief description of the work of this paper and points out the future direction of the work.

## 2. Related Works

The dimensions of microarray data are often extremely asymmetric and highly redundant, and most genes are considered to be irrelevant to the category under study. Traditional classification methods cannot effectively process such data. Many researchers have achieved good results using machine learning techniques to process gene expression data sets.

### 2.1. Machine Learning for Gene Selection

Singh et al. [49] proposed a hybrid improved chaotic emperor penguin (CEPO) algorithm based on the Fisher criterion, ReliefF, and extreme learning machine (ELM) for microarray data analysis. In this paper, the Fisher criterion and ReliefF method were first used as gene selection filters, and then relevant data were used to train the ELM to obtain a better model. Banu et al. [50] used the fuzzy clustering method to assign initial values to each gene and then predicted the likelihood of belonging to each cluster to carry out gene selection. The comparative experimental results show that the fuzzy clustering algorithm performs well in gene prediction and selection. Chen et al. [51] proposed a support vector machine for binary tumor diagnosis, extending the three kinds of support vector machines to improve the performance of gene selection. At the same time, lasso, elastic net, and other sparse regression methods were introduced for cancer classification

and gene selection. Mahendran et al. [52] conducted an extensive review of recent work on machine learning-based selection and its performance analysis, classified various feature selection algorithms under supervised, unsupervised and semi-supervised learning, and discussed the problems in dealing with high and low sample data. Tan et al. [53] proposed an integrated machine learning approach to analyze multiple gene expression profiles of cervical cancer to find the genomes associated with it, with the expectation that it could help in diagnosis and prognosis. The gene expression data were identified effectively through the analysis of three steps.

Zhou et al. [54] proposed an improved discretized particle swarm optimization algorithm for feature selection. In their work, a modest pre-screening process is first applied to obtain fewer features; then, a better cutting combination is found through the encoding and decoding method based on PSO and the local search strategy guided by probability to obtain the desired feature subset. Zohre Sadeghian et al. [55] proposed a three-stage feature selection method based on the S-BBOA algorithm. In the first stage, the minimum redundancy—maximum new classification information (MRMNCI) feature selection was used to remove 80% of the irrelevant and redundant features. The best feature subset was chosen using IG-BBOA in the second step. Furthermore, the similarity ranking approach was used to choose the final feature subset. Veredas Coletto-Alcudia et al. [56] proposed a new hybridization method based on the dominance degree artificial bee colony algorithm (ABCD) to investigate the problem of gene selection. The method combines the first step of gene screening with the second part of the optimization algorithm to find the optimal subset of genes for the classification task. The first step is to use the Analytic Hierarchy Process (AHP) to select the most relevant genes in the dataset through five sequencing methods. In this way, gene filtering reduces the number of genes that need to be managed. For the second step, gene selection can be divided into two objectives: minimizing the number of selected genes and maximizing classification accuracy. Lee et al. [57] embedded the formal definition of correlation into Markov coverage (MB) and established a new multi-feature sequencing method, which was applied to high-dimensional microarray data, enhancing the efficiency of gene selection and, as a result, the accuracy of microarray data classification.

## 2.2. Swarm Intelligence for Gene Selection

Alok Kumar Shukla et al. [4] created TLBOGSA, a hybrid wrapper approach that combines the features of the Teaching Learning based Optimization (TLBO) and the Gravity Search Algorithm (GSA). TLBOGSA was updated with a new encoding approach that transformed the continuous search space into the binary search space, resulting in the binary TBSA. First, significant genes from the gene expression dataset were chosen using the minimal redundancy and maximum correlation (mRMR) feature selection approach. Then, using a wrapper strategy, information genes were chosen from the reduced data generated by the mRMR. They developed the gravitational seeking mechanism in the teaching stage to boost the evolutionary process's searching capabilities. The technique selected the most reasonable genes using a Naive Bayes classifier as a fitness function, which is useful for accurate cancer classification. Based on the phase diagram approach, Elahe Khani et al. [58] suggested a unique gene selection algorithm, and Ridge logistic regression analysis was performed to evaluate the likelihood that the genes belong to a stable group of genes with excellent classification ability. To address the problems, a methodology for the final selection of the selected set is suggested. The model's performance was assessed using the B632+ error estimation approach. To identify genes from gene expression data and valuable information genes from cancer data genes, a decision tree optimizer based on particle swarm optimization was presented by Chen et al. [59]. Experimental results demonstrate that this strategy outperforms different popular classifiers, including support vector machines, self-organizing mapping, and back propagation neural networks. Dabba et al. [10] developed the Quantum MFO (QMFOA), a swarm intelligent gene selection technique based on the fusion of quantum computing with the MFO, to discover

a relatively small subset of genes for high-precision sample classification. The QMFOA gene selection algorithm has two stages: the first is preprocessing, which acquires a preprocessing gene set by measuring the redundancy and correlation of genes, and the second is hybrid combination and gene selection, which utilizes several techniques such as MFO, quantum computing, and support vector machine. To select a limited, representative fraction of cancer-related genetic information, Mohamad et al. [60] developed an enhanced binary particle swarm optimization for gene selection. The velocity of particles is incorporated in this approach to give the rate of particle position change, and the particle position update rule is presented. The experimental findings show that the suggested technique outperforms the classic binary PSO in terms of classification accuracy while picking fewer genes (BPSO).

### 3. The Proposed ISMA

#### 3.1. SMA

Several swarm intelligence optimization techniques have appeared successively in recent years, such as slime mould algorithm (SMA) [35], Harris hawks optimization (HHO) [61], hunger games search (HGS) [62], Runge Kutta optimizer (RUN) [63], colony predation algorithm (CPA) [64], and weighted mean of vectors (INFO) [65]. Due to the simplicity and efficiency of swarm intelligence algorithms, they have been widely used in many different fields, such as image segmentation [66,67], the traveling salesman problem [68], feature selection [69,70], practical engineering problems [71,72], fault diagnosis [73], scheduling problems [74–76], multi-objective problems [77,78], medical diagnosis [79,80], economic emission dispatch problems [81], robust optimization [82,83], solar cell parameter identification [84], and optimization of machine learning models [85]. Among them, SMA is a new bionic stochastic optimization problem, simulating slime mold behavior and morphological changes during foraging. At the same time, SMA used weight to simulate the positive and negative feedback effects of slime mould propagation waves during foraging behavior to construct a venous network with different thicknesses. The morphology of the slime mould changed with the three search patterns: proximity to food, wrap around food, and oscillation.

From the brief description of SMA shown in Figure 1, the random value *rand* helps to find the optimal solution. The slime moulds were randomly distributed in any direction to search for solutions (food), and when  $rand < z$ , there was no venous structure. During the search phase, when  $rand \geq z$  and  $r < p$ , individuals form diffuse venous structures to access food. The adaptive change of decision parameter *p* ensures better adaptability of the transition from the exploration stage to the exploitation stage. During the exploitation phase, when  $r \geq p$ , the individual encapsulates the solution (food) through venous fibrillation.

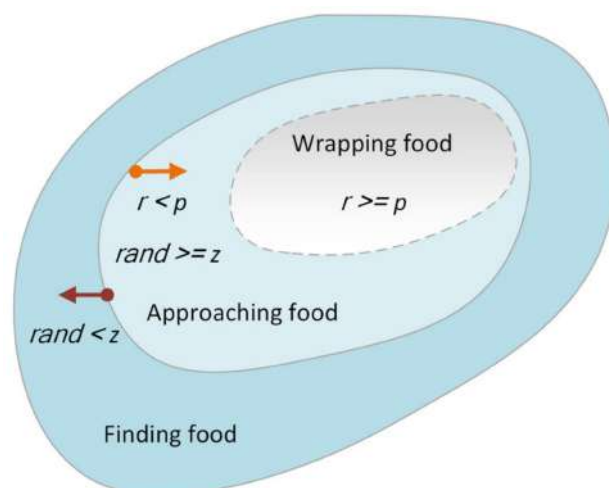


Figure 1. A brief description of SMA.

Based on the following significant parameters, a specific mathematical model of SMA can be constructed to represent the three contraction modes of slime mould:

$$X(t+1) = \begin{cases} rand \cdot (UB - LB) + LB, & rand < z \\ X_b(t) + vb(t) \cdot (W \cdot X_A(t) - X_B(t)), & r < p \\ vc(t) \cdot X(t), & r \geq p \end{cases} \quad (1)$$

where  $X(t)$  and  $X(t+1)$  represent the position vectors of slime mould during iteration  $t$  and  $(t+1)$ , respectively.  $UB$  and  $LB$  indicate the upper and lower boundaries of the search space, respectively.  $X_b$  denotes the position vector of the individual with the highest fitness (highest concentration).  $X_A(t)$  and  $X_B(t)$  indicate the position vectors of random individuals selected from the slime mould during iteration  $t$ .  $rand$  and  $r$  are random values between 0 and 1. The parameter  $z$  is set to 0.03 as in the original literature.

In addition, the decision parameter  $p$  can be calculated as follows:

$$p = \tanh \tanh |S(i) - DF| \quad (2)$$

where  $S(i)$  indicates the fitness of the  $i$ th individual in the slime mould  $X$ ,  $i \in 1, 2, \dots, N$ .  $N$  denotes the size of the population.  $DF$  represents the best fitness, which is attained during all of the iterations.

$W$  is the weight vector of slime mould, which can be obtained from the following equation. This vector mimics the rate at which slime mould shrinks around food for different food masses.

$$W(\text{SmellIndex}(i)) = \begin{cases} 1 + r \cdot \log\left(\frac{bF - \text{SmellOrder}(i)}{bF - wF} + 1\right), & \text{condition} \\ 1 - r \cdot \log\left(\frac{bF - \text{SmellOrder}(i)}{bF - wF} + 1\right), & \text{otherwise} \end{cases} \quad (3)$$

$$[\text{SmellOrder}, \text{SmellIndex}] = \text{sort}(S)$$

where  $bF$  and  $wF$  are the best and worst fitness obtained in the current iteration, respectively.  $\text{SmellIndex}$  and  $\text{SmellOrder}$  denote, respectively, the fitness sort order (smallest problems sorted in ascending order) and the corresponding fitness value. *condition* indicates the first half of  $\text{SmellOrder}$  and is also the overall fitness ordering value. *condition* simulates the individuals adjusting their search patterns dynamically according to the quality of things.

The collaborative interaction between the parameters  $vb$  and  $vc$  can simulate the selection behavior of slime mould.  $vb$  denotes a random value in the interval  $[-a, a]$ . The parameter  $vc$  represents a decrease in the number of iterations within the interval  $[-b, b]$ .

$$a = \text{arctanh}\left(1 - \left(\frac{t}{\text{Max\_iter}}\right)\right) \quad (4)$$

$$b = 1 - \left(\frac{t}{\text{Max\_iter}}\right) \quad (5)$$

where  $\text{Max\_iter}$  indicates the maximum number of iterations.

The simplified pseudo-code of SMA is listed in Algorithm 1. We can find more specific descriptions in the original literature.

---

**Algorithm 1:** Pseudo-code of SMA

---

Begin

  Initialize the parameters:  $\text{Max\_iter}, N$

  Initialize slime mould population  $X$

  While  $t \leq \text{Max\_iter}$

    Calculate the fitness of each individual in the slime mould

    Update best fitness and the  $X_b$

    Calculate the weight  $W$  according to Equation (3)

---

---

```

Calculate  $a$  according to Equation (4)
Calculate  $b$  according to Equation (5)
For  $i = 1, 2, \dots, N$  (each search agent)
    Update  $p$  according to Equation (2)
    Update  $vb, vc$  based on  $a$  and  $b$ , respectively
    Update the positions according to Equation (1)
EndFor
iteration = iteration + 1
EndWhile
Return the best fitness and  $X_b$ 
End

```

---

### 3.2. The Cauchy Mutation Operator

In this section, we will briefly introduce the Cauchy mutation. The Cauchy density function can be described as:

$$f_t(x) = \frac{1}{\pi} \frac{t}{t^2 + x^2}, -\infty < x < \infty \quad (6)$$

where  $t > 0$  and is the proportional parameter, and the distribution function is expressed as follows:

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{t}\right) \quad (7)$$

By increasing the search range in each generation, individuals can be guaranteed to find better solutions in a wider range, thus avoiding local optimization. Therefore, Cauchy mutation was selected as an improved mechanism.

In the original SMA based on Equations (6) and (7), the version using the Cauchy mutation operation is expressed as:

$$x_{i\_cauchy} = x_i \times (1 + Cauchy) \quad (8)$$

where *Cauchy* is the random number of the distribution obtained by the Cauchy distribution,  $x_i$  is a position in the SMA at the time of the current iteration,  $x_{i\_cauchy}$  is the corresponding position of  $x_i$  after Cauchy mutation. The introduction of the Cauchy mutation mechanism improves the foraging behavior of slime mould searching the unknown space, so the quality of SMA solutions can be enhanced by using the Cauchy operator in the simulation process.

### 3.3. The Mutation and Crossover Strategy in DE

During the optimization procedure, the major operations are mutation and crossover. Each solution  $x_i = \{x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}\}$  is a vector of  $n$  dimensions.

#### A. Mutation

A mutant vector can be generated via the mutation operator  $?_i$  according to selected components from randomly nominated vectors  $x_a, x_b$ , and  $x_c$ , where  $a \neq b \neq c \neq i$ . The mathematical equation can be represented as follows:

$$u_i = x_a + F * (x_b - x_c) \quad (9)$$

where  $F$  is a random number that is able to control the mutation's perturbation size.

#### B. Crossover

The crossover operator may construct a trial vector  $v_i$  by applying crossover to a mutant vector, where the trial vector is constructed by randomly selecting items from the

mutant  $u_i$  and the target vector  $x_i$  depending on the probability ratio  $P_c$ . The math formula appears such as this:

$$v_{ij} = \begin{cases} u_{ij}; & \text{rand} \leq P_c \text{ or } j = j_0 \\ x_{ij}; & \text{otherwise} \end{cases} \quad (10)$$

The probability feature  $P_c$  controls the diversity of the swarm and relieves the risk of local optima, and  $j_0$  is an index between  $[1, 2, 3, \dots, N_p]$ , which guarantees that  $v_i$  obtains at least one component from  $u_i$ .

### 3.4. The Hybrid Structure of the Proposed ISMA

Considering that the original SMA may not converge to some suboptimal solutions precociously or face the risk of falling into local optimal solutions, the improved algorithm ISMA proposed in this paper combines two strategies, Cauchy mutation and crossover mutation based on DE, to promote the coordination of global exploration and local exploitation and forms a new SMA variant, namely ISMA. The structure of the proposed ISMA is shown in Figure 2, which is demonstrated in Algorithm 2. Under the ISMA framework, these two strategies are used, in turn, to generate the new search agent and the best agent with the best solution in the current iteration. Figure 2 depicts the ISMA process. As illustrated in the picture, the position of each agent may be rebuilt when the location of each agent is updated according to Equation (1), implying that each agent achieves the best solution in a larger search space.

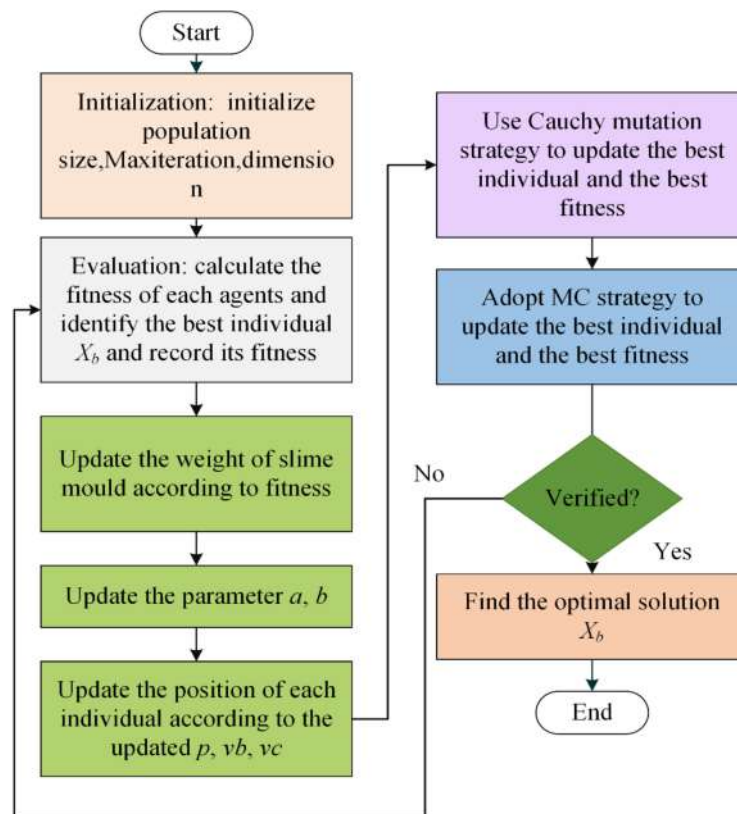


Figure 2. The framework of the proposed ISMA.

The position update based on SMA is to solve the position vector of slime mould according to the optimization rules of SMA, as detailed in Section 2.1. This phase produces a population based on SMA. The Cauchy-based mechanism and the crossover mutation mechanism are based on the behavior of the Cauchy-based mutation operation and the crossover mutation operation shown in Section 2.2 to adjust the position vector of an SMA-based individual to produce a new SMA-based population. In this stage, the advantages of Cauchy and the crossover mutation mechanism in the exploration stage are utilized to make



up for the shortcomings of the SMA exploration. Considering both mechanisms' effects on search ability, this means increasing population size and thus population diversity. The research shows that this stage not only helps to promote the coordination of exploration and exploitation capabilities but also helps to improve the quality of solutions and accelerate the convergence rate.

---

**Algorithm 2:** Pseudo-code of ISMA

---

```

Begin
  Initialize of the parameters:  $Max\_iter, N$ 
  Initialize of slime mould population  $X$ 
  While  $t \leq Max\_iter$ 
    Calculate the fitness for each individual in slime mould
    Update  $X_b$  and the best fitness
    Calculate the weight  $W_{a,b}$  according to Equations (3)–(5)
    For  $i = 1 : N$ 
      Update  $p$  using Equation (2)
      Update  $vb, vc$  based on  $a$  and  $b$ , respectively
      Update the positions by Equation (1)
    EndFor
    Use Cauchy mutation strategy to update the best individual and the best fitness
    Adopt MC strategy to update the best individual and the best fitness
    iteration = iteration + 1
  EndWhile
  Return the best fitness and  $X_b$  as the best solution
End

```

---

### 3.5. Computational Complexity

The proposed SMA structure mainly includes the following parts: initialization, fitness evaluation, fitness sorting, weight updating, position updating based on SMA strategy, position updating based on Cauchy mutation strategy, and position updating based on crossover mutation strategy, where  $N$  is the number of cells of slime mould,  $D$  is the dimension of function, and  $T$  is the maximum number of iterations. The computational complexity of initialization is  $O(D)$ . In the process of evaluation and sorting of fitness, the computational complexity is  $O(N + N \log N)$ . The computational complexity of updating the weight is  $O(N \times D)$ . The computational complexity of the location update process based on SMA is  $O(N \times D)$ . Similarly, the computational complexity of the location updating process based on the Cauchy mutation mechanism and cross mutation mechanism is  $O(N \times D)$ . Therefore, the total computational complexity of ISMA is  $O(D + T \times N \times (1 + 4D + \log N))$ .

## 4. Experimental Design and Analysis of Global Optimization Problem

To evaluate successive versions of ISMA, we considered two experiments to compare the methods presented in this section with several other competitors. We used 23 continuous benchmark functions (including 7 unimodal functions, 6 multimodal functions, and 10 fixed-dimensional multimodal functions) and 10 typical CEC2014 benchmark functions (2 hybrid functions and 8 composition functions) for a total of 33 benchmark cases. Experiment 1 is a series of SMA variants with different update strategies: ISMA, CSMA, and MCSMA. The best SMA variants are obtained by comparing them with the original SMA and DE algorithm. Experiment 2 is to compare the ISMA algorithm with 8 other advanced optimization algorithms, including multi-population ensemble differential evolution (MPEDE) [86], successful history-based adaptive DE variants with linear population size reduction (LSHADE) [87], particle swarm optimization with an aging leader and challengers (ALCPSO) [88], comprehensive learning particle swarm optimizer (CLPSO) [38], chaos-enhanced sine cosine-inspired algorithm (CESCA) [89], improved grey wolf optimization (IGWO) [39], whale optimization algorithm with  $\beta$ -hill climbing (BHC)

algorithm and associative learning and memory (BMWOA) [90], modified GWO with random spiral motions, simplified hierarchy, random leaders, oppositional based learning (OBL), levy flight (LF) with random decreasing stability index, and greedy selection (GS) mechanisms (OBLGWO) [91]. In this study, all experimental evaluations were conducted on a Windows 10(64-bit) operating system with 32GB RAM, Intel(R) Xeon(R) Silver 4110 CPU @ 2.40 GHz 2.10 GHz (dual processor), and MATLAB R2014a coding.

Tables A1–A4 contain information on 23 benchmark functions and 10 classic CEC2014 benchmark functions. It can be seen that the information of the 33 functions used in the experiment contains a wide variety of problems. These capabilities can be used not only to verify the local exploitation ability and global exploration ability but also to verify the ability to balance the two abilities. In addition, to reduce algorithm randomness's impact on the experiment [92], we conducted 30 independent tests for each test case. In order to exclude the influence of other factors on the experiment, all the test algorithms were run under the same settings and conditions [93–95]. The maximum function evaluation was set as 300,000, and the population size was 30.

In addition, statistical results such as mean and standard deviation (std) are used to represent the global optimization ability and robustness of the evaluation method. The Wilcoxon signed-rank test at the significance level of 0.05 was used to measure the degree of improvement, which was statistically significant. It is worth noting that the label '+/=/' in the results indicates that ISMA is significantly superior to, equal to, or worse than other competitors. For a comprehensive statistical comparison, the Friedman test was used to see whether the performance of all the comparison algorithms on the baseline function differed and was statistically significant. The mean ranking value (ARV) of the Friedman test was used to evaluate the average performance of the investigated method. It is worth noting that a reliable comparison must involve more than 5 algorithms for more than 10 test cases [96].

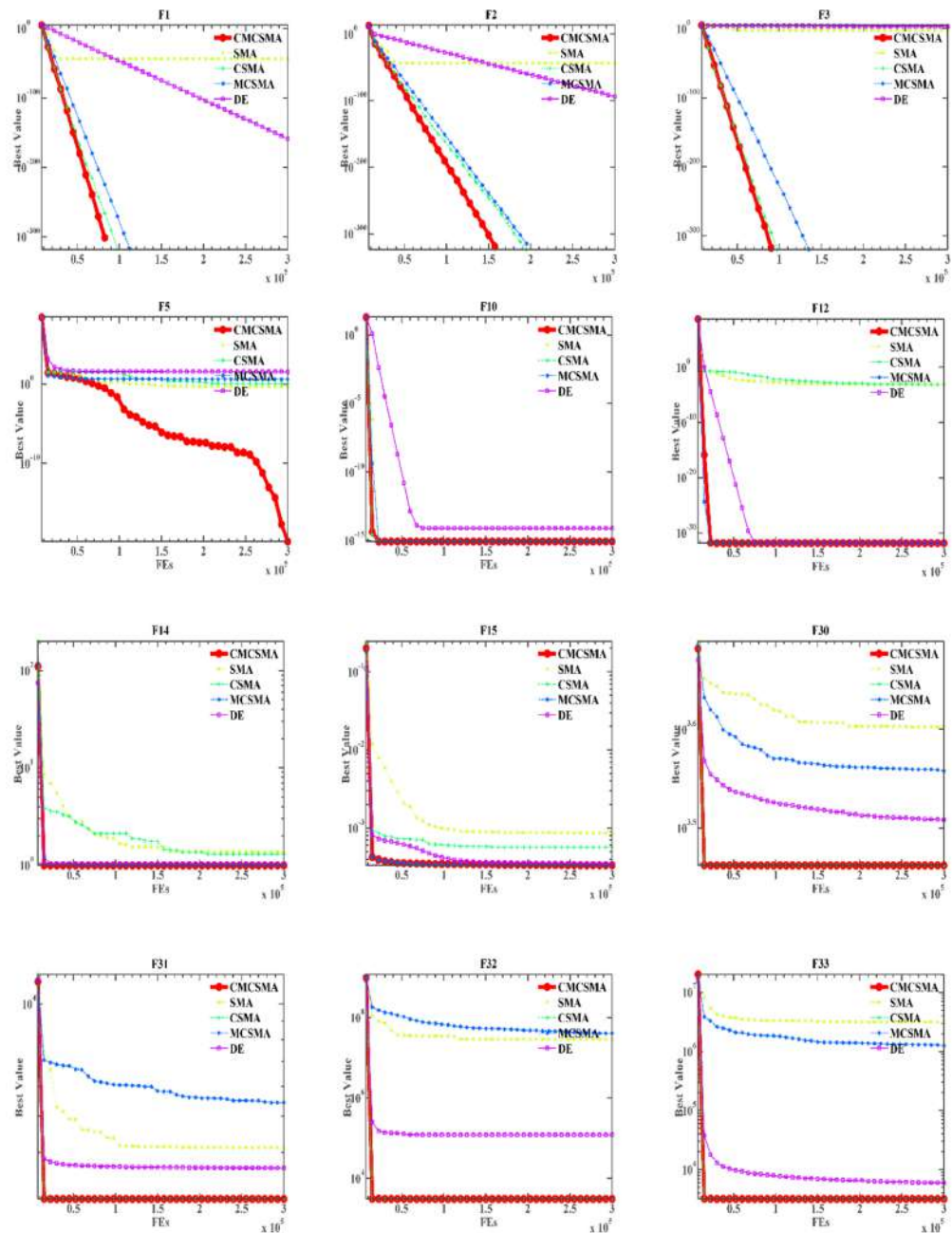
#### 4.1. Comparison between SMA Variant and Original SMA and DE Algorithm

In this section, to prove the superiority of the Cauchy mutation mechanism and the combination of mutation and crossover strategies in DE, we compare the three combinations of the two mechanisms and the original SMA with the DE algorithm. The comparison results are shown in Tables A5–A7, and the algorithm convergence curve is shown in Figure 3.

As the results show in Tables A5 and A6, ISMA clearly outperforms the other mechanism combinations and the original SMA and DE algorithms, as ISMA outperforms almost all algorithms in handling most of the test functions. As can be seen from the ARV of Friedman's test in Table A7, ISMA can be considered the first algorithm when comparing the five algorithms. Mean and std in Table A5 also indicate the superiority of ISMA in F1–F6, F9–F14, F26–28, and F30–33 functions. ISMA ranks 2nd in F7, F15–F17, F19–25, and F29. According to the statistical significance of  $p$ -values in Table A6, almost all values in the SMA column are less than 0.05, indicating that ISMA has significantly improved the original SMA algorithm. The final optimization effect of F1–3, F9–11 and F26–28 functions by CSMA and MCSMA is the same as that by ISMA. In summary, the results of the Wilcoxon signed-rank test show that, statistically, ISMA has significantly improved performance compared with other algorithms. The results show that the addition of the Cauchy mutation strategy and crossover mutation strategy based on DE is beneficial to ISMA's exploitation ability and exploration ability and the balance between ISMA's exploitation ability and exploration ability.

The convergence analysis can show which optimizer as an iterative method can reach better quality results within a shorter time [97,98]. Figure 3 shows the convergence curves of the comparison method on 12 functions. We can intuitively find that, compared with the original SMA, DE, and other two SMA variants, the ISMA using the two mechanisms has a better effect. Combining the two mechanisms makes the SMA avoid falling into the local optimal solution and can obtain the global optimal solution. The overall advantage of

ISMA is significant because of the positive effect of the Cauchy mutation mechanism and the crossover mutation strategy on SMA, which highlights the optimization capability of the proposed method.



**Figure 3.** Convergence curves of the SMA variants and the original SMA and DE algorithms on twelve functions.

#### 4.2. Comparison with Advanced Algorithms

In this experiment, we compare ISMA with several typical advanced algorithms, such as MPEDE [86], LSHADE [87], ALCP SO [88], CLPSO [38], BMWOA [90], CESCA [89], IGWO [39] and OBLGWO [91], in order to fully prove the ability of the proposed algorithm to avoid local optimal and global exploration. These include two superior DE variants, two often-computed PSO variants, and variants of WOA, GWO, and SCA.

Tables A8–A10 record the results of the comparison between ISMA and eight advanced algorithms. As can be seen from the comparison results in Table A10, among ISMA and 8 other advanced meta-heuristic algorithms, the average Friedman test result of ISMA is

3.7075758, ranking first, followed by CLPSO. The statistical results in Table A8 show that among all the comparison algorithms, the std of ISMA on more test functions is 0, so it can be seen that ISMA is more stable. In addition, the comparison results of specific functions show that ISMA has a stronger ability to deal with complex functions and mixed functions than other advanced algorithms. The mean and std in Table A8 also indicate the superiority of ISMA in F1–6, F9–15, F26–28 and F30–33 functions. ISMA also ranks high in the F7, F21–23. In addition, Table A9 shows the Wilcoxon signed-rank test results between ISMA and other advanced algorithms. It can be seen that ISMA outperforms other algorithms on most of the benchmark functions, especially CESSA, which outperforms CESSA on 90.9% of the functions. As a result, ISMA is superior to other strong competitors.

The convergence curves of all nine algorithms over 12 functions shown in Figure 4 show that the convergence rate of ISMA is competitive with other more advanced methods, which always converge to local optimum earlier than ISMA. It can be proved that the ISMA algorithm has a strong ability to avoid local and global searches, and ISMA can produce more accurate solutions.

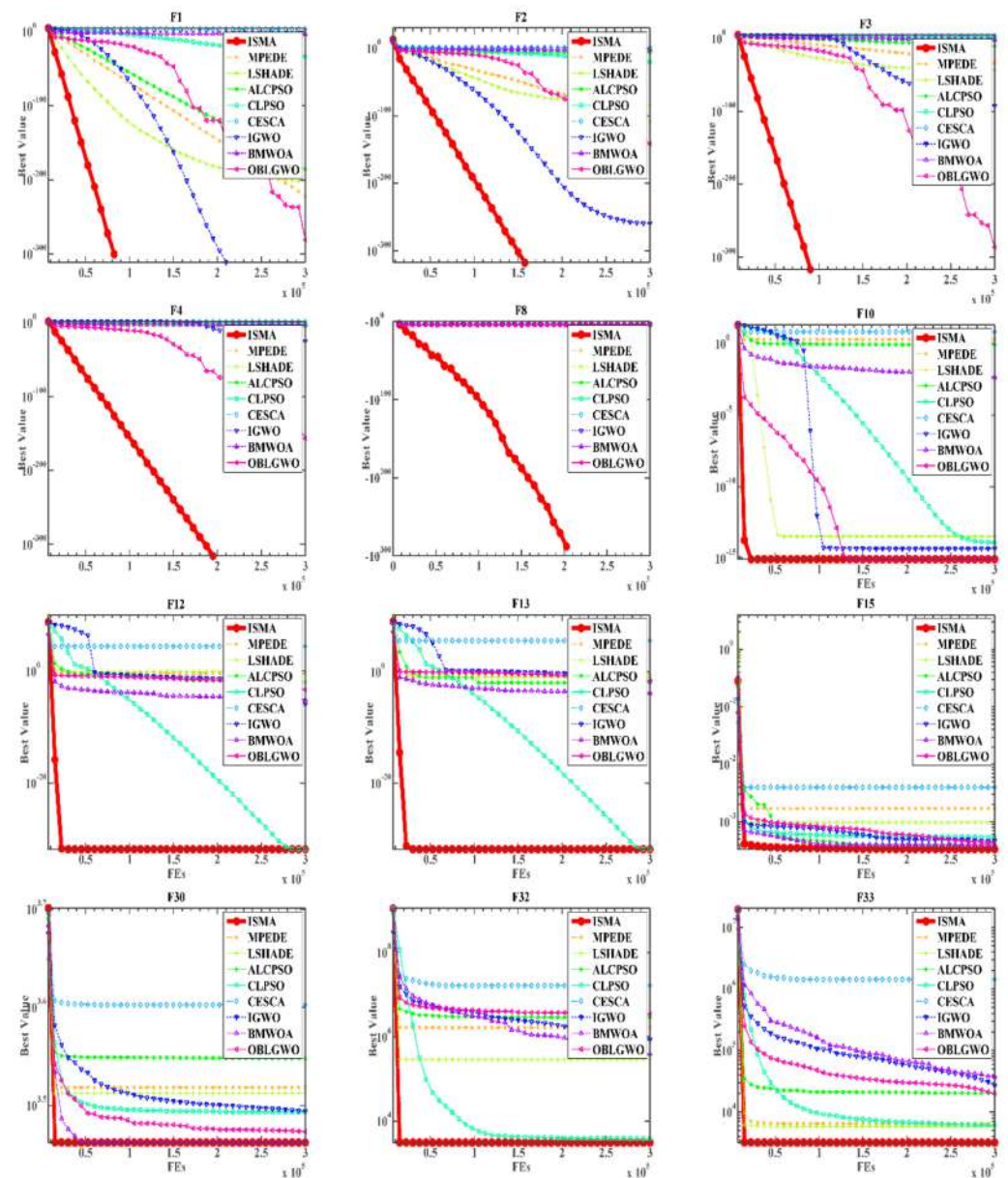


Figure 4. Convergence curves of the ISMA and the other advanced algorithms on twelve functions.

To sum up, the optimization power of ISMA is reflected in the overall superior performance of ISMA in different types of functions compared to the more challenging advanced methods. The combination of the Cauchy mutation mechanism and crossover mutation strategy based on the DE algorithm enables the proposed ISMA to obtain a higher quality solution in the optimization process and makes exploration and exploitation in a better equilibrium state.

## 5. The Proposed Technique for Gene Selection

In this section, the proposed ISMA is applied to the gene selection problem, which makes improving the proposed algorithm more practical. For this purpose, we transform the continuous ISMA into a discrete version, namely the BISMA of the wrapper method, to solve the gene selection problem for binary optimization tasks.

### 5.1. System Architecture of Gene Selection Based on ISMA

The procedure of selecting or generating some of the most significant features from a set of features in order to lower the dimension of the training dataset is known as feature selection. Many fields with large data sets want to be able to reduce the dimensions of application data, such as gene selection for high-dimensional gene expression data sets in the medical field. The task of gene selection is to reduce the number of irrelevant and unimportant genes, identify the most relevant genomes with the greatest classification accuracy, reduce the cost of high computing costs and improve the accuracy of disease analysis. The continuous ISMA optimizer is converted to binary ISMA (BISMA) using the transfer function (TF) for the gene selection problem. The machine learning algorithm was used as a classifier to evaluate the ability of BISMA to identify discriminant genes and eliminate irrelevant, redundant genes in high-dimensional gene expression datasets. In addition, cross-validation (CV) was used to evaluate the optimality of selected gene subsets for classification during the evaluation process.

### 5.2. Fitness Function

Gene selection is a process that uses the least subset of genes to obtain the optimal classification accuracy, and both goals need to be achieved simultaneously. Therefore, in order to meet each objective, the fitness function expressed in Equation (11) can be designed to comprehensively evaluate the candidate solutions by using classification accuracy and the number of selected genes.

$$fit = \alpha \times (1 - Acc) + \beta \times \left( \frac{D_R}{D} \right) \quad (11)$$

where *Acc* indicates the classification accuracy of the classifier (machine learning method), so  $(1 - Acc)$  is the error rate of the classifier. The weighting factors  $\alpha$  and  $\beta$  are the importance of error rate and the number of selected genes, respectively, and  $\alpha \in [0,1]$ ,  $\beta = 1 - \alpha$ .  $D$  is the total number of genes in the exponential data set, and the numerator  $D_R$  is the number of genes filtered by the proposed gene selection optimizer. In this study,  $\alpha$  and  $\beta$  were set to 0.95 and 0.05, respectively.

### 5.3. Implementation of Discrete BSSMA

The proposed ISMA optimizer searches for the optimal solution in a continuous search space in previous work. Gene selection is a binary problem. The transfer function restricts the continuous search space to 0 or 1. When the value is 0, it means not selected, and when the value is 1, it means selected.

Individuals with binary position vectors are initialized through a random threshold, as shown below:

$$x_i^d = \begin{cases} 0, & rand \leq 0.5 \\ 1, & rand > 0.5 \end{cases} \quad (12)$$

where  $x_i^d$  is the  $i$ -th gene on the  $d$ -th dimension of the position vectors of the slime mould.

In addition, the transfer function (TF) is a suitable converter that can convert a continuous optimization algorithm to a discrete version without changing the algorithm's structure because it is convenient and efficient [99]. There are 8 types of TFs, which can be divided into S-shaped and V-shaped according to their shapes. Their mathematical formulae and graphical descriptions are shown in Table A11.

For an S-shaped family, a gene of the position vector at the next iteration can be converted according to the TFS1-TFS4 shown in Table A11 as follows:

$$x_i^d(t+1) = \begin{cases} 1, & \text{rand} < T(x_i^d(t+1)) \\ 0, & \text{rand} \geq T(x_i^d(t+1)) \end{cases} \quad (13)$$

where  $T(x_i^d(t+1))$  represents the probability value of the  $i$ -th gene on the  $d$ -th dimension at the next iteration.

For a V-shaped family, the gene of the position vector at the next iteration can be converted according to the TFV1-TFV4 shown in Table A11 as follows:

$$x_i^d(t+1) = \begin{cases} -x_i^d(t+1), & \text{rand} < T(x_i^d(t+1)) \\ x_i^d(t+1), & \text{rand} \geq T(x_i^d(t+1)) \end{cases} \quad (14)$$

## 6. Experimental Design and Discussion on Gene Selection

### 6.1. Experimental Design

In this experiment, two kinds of comparison results are used to evaluate the optimization ability of the proposed algorithm. In the first assessment, we studied Bisma with different TFs to determine the best version of Bisma out of the eight TFs. The resulting Bisma is compared with other mature meta-heuristic optimizers in the second evaluation. Fourteen gene expression datasets were used in the two case studies. Table A12 lists the detailed characteristics of these microarray datasets, including the number of samples, the number of genes per sample, and the number of categories. These 14 representative gene datasets have been widely used to test a variety of gene selection optimizers to evaluate their performance.

In addition, to obtain more convincing results, this paper also considers the Leave-One-Out cross-validation (LOOCV) to validate the gene selection process. A sample in the data set is taken as the test set to verify the classification accuracy of the classifier, while the rest of the sample is taken as the training set to be trained with the classifier. The number of validations per dataset is equal to the size of the test dataset. The KNN classifier is used for classification tasks. For KNN, let the field size  $k$  in KNN be 1. The test method of distance  $D$  is as follows:

$$D(x, y) = \left( \sum_K^N (x_k - y_k) \right)^{\frac{1}{2}} \quad (15)$$

To be fair in comparison [100–102], each evaluation and comparison involving Bisma was performed on the same computing environment, namely Intel(R) Xeon(R) Silver 4110 CPU @ 2.40 GHz 2.10 GHz (two processors) and 8 GB RAM (Windows 10)(64-bit). MATLAB R2014a software was used to test the algorithm. For each algorithm, we set the maximum number of iteration agents and the number of search agents to 50 and 20, respectively. It was run 10 times independently. The initial parameters of all algorithms are identified as their original reference parameters.

### 6.2. The Proposed Bisma with Different TFs

Considering the effect of TF on the performance of the gene selection optimizer, we developed eight Bisma optimizers using eight different TFs and evaluated their effectiveness in finding the optimal gene from each gene dataset listed in Table A10. These eight TFs include four S-shaped and four V-shaped TFs, as shown in Table A11. This

assessment helps to obtain the best binary version of Bisma to the gene selection issue. Tables A13–A16 show the average number of selected genes, the average error rate, the average fitness, the average calculation time, and the corresponding std and AVR of the 8 developed versions of the Bisma optimizer.

The average number of selected genes produced by each version of Bisma on the 14 datasets is shown in Table A13. The number of genes required by Bisma based on V-shaped was the least among all versions of Bisma. As can be seen from the ARV value, the average number of selected genes of Bisma based on TFV4 was the least and ranked the first, while the four Bisma based on V-shaped were ranked as the first four, and the number of selected genes was significantly lower than that of Bisma based on S-shaped.

Table A14 records the average classification error rates of the eight versions of Bisma on the baseline gene dataset. Judging from the average ranking value, Bisma with TFV4 is significantly better than other competitors. The four V-shaped Bismas obtained an average error of 0 in 57% of the gene data sets, indicating the stability of feature selection based on V-shaped Bisma. Meanwhile, Bisma based on V4 obtained an error of 0 and a standard deviation of 0 on 85.7% of gene data sets. Therefore, from the average error rate, the ability of Bisma with V-shaped TFs to solve the gene selection task is due to its S-shaped TFs counterpart.

According to the average fitness test results reported in Table A15, it can be found that Bisma\_V3 achieved the best fitness on about 42.9% of the baseline gene data set, which was slightly better than Bisma\_V3 and significantly better than other competitors. However, from the ranking mean, Bisma\_V4 ranked first, followed by Bisma\_V3, Bisma\_V1, Bisma\_V2, Bisma\_V1, Bisma\_S2, Bisma\_S3, and Bisma\_S4. The fitness results also showed that TFs of the V-shaped family were better than that of the S-shaped family.

Similarly, it can be seen from the calculated time that, except for V1, the version of the V-shaped TFs takes less time to run than the version of the S-shaped TFs. In particular, the first-place V4 takes much less time on average than the second-place V3. The calculation overhead of Bisma\_V4 with the best average ranking value is lower than that of the other versions over all the benchmark datasets.

As shown in Tables A13–A16, the Bisma version with TFV4 was superior to other versions in terms of the average number of selected genes, average error rate, average fitness, and average time cost, and the Bisma version with TFV4 was far superior to the second in terms of average time cost. In comparing S-shaped and V-shaped, V-shaped can achieve better results than S-shaped. Therefore, the transfer function TFV4 was chosen as the best choice to establish a Bisma optimizer with better stability for genetic problems. In this case, Bisma\_V4 is used to represent Bisma, which is further evaluated by comparison in the following sections.

### 6.3. Comparative Evaluation with Other Optimizers

In this section, the superiority of the proposed Bisma optimizer is evaluated by comparing it with several state-of-the-art meta-heuristic approaches. These algorithms considered to be meta-heuristics are bGWO [103], BGSA [104], BPSO [99], bALO [105], BBA [106], BSSA [107], bWOA [108], BSMA, the binary form of the original SMA [35], and Bisma, the discrete version of the improved ISMA. Table A17 shows the parameter settings for the relative comparison optimizer.

Tables A18–A21 show the selected genes' statistical results in terms of length, error rate, fitness and computational time. According to the average gene length in Table A18, the proposed Bisma had the least number of selected genes on 57.1% of the gene datasets, while bWOA had the least number of selected genes on 42.9% of the gene datasets. It can be seen that in the 14 data sets, Bisma and bWOA are far more competitive than other algorithms in reducing the data dimensions.

The results of the mean error rate are shown in Table A19, which shows the superiority of the proposed Bisma. Bisma achieves the minimum mean error rate on 85.7% of the gene data sets and only performs slightly worse on Lung\_Cancer and Tumor\_14. bGWO

showed the best error rate on the Tumor\_14 gene dataset, while bWOA showed competing results on the Lung\_Cancer gene dataset. From the perspective of the ARV index, Bisma ranked first, followed by bWOA, bGWO, Bisma, BGSA, BPSO, bALO, BSSA, and BBA.

The fitness of the important measurements shown in Table A20 comprises the weighted error rate and the number of genes selected by weighting. It is clear that the performance of the proposed Bisma is superior to other competitors on 64.3% of the gene data sets. The average fitness of Bisma and bWOA on 14 gene datasets was significantly better than that of the other algorithms.

In addition, according to the std values shown in Tables A18–A20, Bisma showed better performance, satisfactory standard deviation and excellent average fitness values in most of the gene data sets tested, which indicated that Bisma was more stable than bALO, BSSA, BBA, etc. There is a big gap between the overall performance of Bisma, BSMA, bWOA, and bGWO and the overall performance of BGSA, BPSO, bALO, BBA, and BSSA, and the first four optimizers are obviously better than the last five.

As can be seen from the average calculation time results shown in Table A21, the proposed Bisma has the highest time cost, and the time complexity of BSMA and bWOA with better performance is also relatively high, indicating the increase in calculation time cost caused by the improvement of performance. The time cost of Bisma was influenced by the introduced Cauchy mutation and the crossover mutation strategy based on DE. As shown in Table A21, the calculation time of the original SMA is also relatively expensive, which is also the reason for the high cost of Bisma time.

Compared with other gene selection optimizers, it is found that Bisma is the best one. Although the result is not ideal in terms of calculation time, Bisma is expected to select the optimal gene subset on the vast majority of microarray data sets to obtain the best fitness and the best classification error rate without the loss of meaningful genes. This fact proves that the combination of Cauchy mutation and crossover mutation strategy based on DE guarantees the improvement of global exploration in the proposed Bisma to achieve a more effective balance between local exploitation and global exploration.

## 7. Discussions

In this part, the ISMA algorithm proposed in this paper is discussed, and its advantages and existing points can be improved. In the original SMA, the global exploration ability of slime moulds was not strong, and they would fall into local optimum in the face of some problems, limiting the algorithm's use. In this paper, Cauchy mutation (CM) and cross mutation are introduced to update the population, increasing the global exploration space and avoiding falling into local optimum. Experiments show that the effect of a dual mechanism is better than that of a single mechanism, and ISMA is better than some advanced optimization algorithms.

However, ISMA exposes some common shortcomings of random optimizers in certain areas. As seen in Tables A5 and A8, when processing some multimodal functions, the algorithm's performance is sometimes poor due to the randomness of the crossover mutation mechanism. The search speed is slow in global exploration and local exploitation.

A binary algorithm (Bisma) performs feature selection in feature selection optimization problems on 14 data sets. The experimental results show that the proposed algorithm exhibits smaller average fitness and lower classification error rates while selecting fewer features. However, the introduction of Cauchy mutation and cross mutation mechanism brings good effects but also leads to a long running time of the algorithm, and the time complexity is the highest among all comparison algorithms.

According to the study [109], Ornek et al. combined the position update of the sines and cosines algorithm with the slime mold algorithm. In these updates, various sines and cosines algorithms are used to modify the oscillation process of slime molds. Experimental results show that the algorithm has good exploration and exploitation ability. Gurses et al. [110] applied a new hybrid slime mold algorithm, the Simulated Annealing Algorithm (HSMA-SA), to structural engineering design problems. Experimental results



demonstrate the feasibility of the proposed algorithm in solving shape optimization problems. Cai et al. [111] proposed an artificial slime mold algorithm to solve the traffic network node selection problem, and the experimental results are of great significance to studying traffic node selection and artificial learning mechanisms. These ideas can be used as a reference to improve the shortcomings of ISMA in the future so that it can be applied in more fields, such as dynamic module detection [112,113], road network planning [114], information retrieval services [115–117], drug discovery [118,119], microgrids planning [120], image dehazing [121], location-based services [122,123], power flow optimization [124], disease identification and diagnosis [125,126], recommender system [127–130], human activity recognition [131], and image-to-image translation [132].

## 8. Conclusions

In this study, based on the basic SMA, an improved ISMA version is proposed, and the combination of the Cauchy mutation and crossover mutation strategy based on the DE algorithm is used to improve the SMA so as to achieve the coordination between global exploration and local exploitation. We first evaluate the effectiveness of the continuous version of the ISMA algorithm on 33 benchmark evaluation functions to deal with global optimization problems, compared with some advanced swarm intelligence algorithms. The results show that ISMA has a strong global exploration capability. In order to verify the performance of ISMA in practical application, the BISMA was obtained by mapping ISMA into binary space through the transfer function and then applied to the feature selection problem of 14 commonly used UCI datasets. In order to understand the optimal conversion function of the ISMA variant, we compared the number of selected genes, average error rate, average fitness, and computational cost. It can be seen that BISMA\_V4 is superior to other versions. Therefore, BISMA\_V4 is regarded as the final method to solve the gene selection problem. We compare BISMA\_V4 with binary SMA, binary GWO, and several other advanced methods. The experimental results show that BISMA can select fewer features and obtain higher classification accuracy.

Therefore, we believe that the proposed BISMA is a promising gene selection technique. There are several ways to extend the work we have conducted. We can consider applying BISMA to other high-dimensional data sets and study the effectiveness of BISMA on other data sets. Secondly, other strategies can be used to improve the SMA and improve the coordination between the SMA global exploration and local exploration. Thirdly, interested researchers can apply SMA to more areas, such as financial forecasting, optimization of photovoltaic parameters, and other engineering applications. Finally, we can extend the application of ISMA to multi-objective optimization, image segmentation, machine learning, and other fields.

**Author Contributions:** Conceptualization, G.L. and H.C.; methodology, G.L. and H.C.; software, G.L. and H.C.; validation, F.Q., P.Z., A.A.H., G.L., H.C., F.K.K., H.E. and H.L.; formal analysis, F.K.K., H.E. and H.L.; investigation, F.Q., P.Z. and A.A.H.; resources, F.K.K., H.E. and H.L.; data curation, F.K.K., H.E. and H.L.; writing—original draft preparation, F.Q., P.Z. and A.A.H.; writing—review and editing, G.L. and H.C.; visualization, G.L. and H.C.; supervision, F.K.K., H.E. and H.L.; project administration, F.K.K., H.E. and H.L.; funding acquisition, G.L. and H.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research project was funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2022R300), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. Zhejiang University Students Science and Technology Innovation Activity Plan (2022R429B045), Graduate Innovation Fund of Wenzhou University (316202102088).

**Data Availability Statement:** The data involved in this study are all public data, which can be downloaded through public channels.

**Conflicts of Interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Appendix A**

See Tables A1–A21.

**Table A1.** Descriptions of unimodal benchmark functions.

Function	Dim	Range	$f_{\min}$
$f_1(x) = \sum_{i=1}^n x_i^2$	30	[−100, 100]	0
$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	[−10, 10]	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j\right)^2$	30	[−100, 100]	0
$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	[−100, 100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[−30, 30]	0
$f_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30	[−100, 100]	0
$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0,1]$	30	[−128, 128]	0

**Table A2.** Descriptions of multimodal benchmark functions.

Function	Dim	Range	$f_{\min}$
$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[−500, 500]	$-418.9829 \times 30$
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[−5.12, 5.12]	0
$f_{10}(x) = -20 \exp\{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i}\} - \exp\{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\} + 20 + e$	30	[−32, 32]	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[−600, 600]	0
$f_{12}(x) = \frac{\pi}{n} \{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 + \sum_{i=1}^n \mu(x_i, 10, 100, 4)\}$ $y_i = 1 + \frac{x_i + 1}{4}$	30	[−50, 50]	0
$\mu(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$f_{13}(x) = 0.01 \{ \sin^2(3\pi x_i) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] + \sum_{i=1}^n \mu(x_i, 5, 100, 4) \}$	30	[−50, 50]	0

**Table A3.** Descriptions of fixed-dimension multimodal benchmark functions.

Function	Dim	Range	$f_{\min}$
$f_{14}(x) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	[−65, 65]	1
$f_{15}(x) = \sum_{i=1}^{11} \left[ a_i - \frac{x_i (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[−5, 5]	0.00030
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5, 5]	−1.0316
$f_{17}(x) = \left( x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	[−5, 5]	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[−2, 2]	3
$f_{19}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij} (x_j - P_{ij})^2\right)$	3	[1, 3]	−3.86
$f_{20}(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij} (x_j - P_{ij})^2\right)$	6	[0, 1]	−3.32
$f_{21}(x) = -\sum_{i=1}^5 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.1532
$f_{22}(x) = -\sum_{i=1}^7 [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.4028
$f_{23}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]	−10.5363

**Table A4.** Descriptions of CEC2014 functions. (Search range: [−100, 100]D).

Function	Class	Functions	Optimum
F24	Hybrid	Hybrid Function 5 (N = 5)	2100
F25		Hybrid Function 6 (N = 5)	2200
F26		Composition Function 1 (N = 5)	2300
F27	Composition	Composition Function 2 (N = 3)	2400
F28		Composition Function 3 (N = 3)	2500
F29		Composition Function 4 (N = 5)	2600
F30		Composition Function 5 (N = 5)	2700
F31		Composition Function 6 (N = 5)	2800
F32		Composition Function 7 (N = 3)	2900
F33		Composition Function 8 (N = 3)	3000

**Table A5.** The SMA variants are compared with the original SMA and DE algorithms.

	F1		F2		F3	
	mean	std	mean	mean	std	mean
ISMA	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$
SMA	$3.2559 \times 10^{-44}$	$1.7833 \times 10^{-43}$	$1.7856 \times 10^{-44}$	$3.2559 \times 10^{-44}$	$1.7833 \times 10^{-43}$	$1.7856 \times 10^{-44}$
CSMA	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$
MCSMA	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$
DE	$1.8673 \times 10^{-159}$	$4.1198 \times 10^{-159}$	$1.3001 \times 10^{-94}$	$1.8673 \times 10^{-159}$	$4.1198 \times 10^{-159}$	$1.3001 \times 10^{-94}$
	F4		F5		F6	
	mean	std	mean	mean	std	mean
ISMA	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$1.5210 \times 10^{-20}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$1.5210 \times 10^{-20}$
SMA	$9.1947 \times 10^{-44}$	$5.0362 \times 10^{-43}$	$4.5273 \times 10^{-1}$	$9.1947 \times 10^{-44}$	$5.0362 \times 10^{-43}$	$4.5273 \times 10^{-1}$
CSMA	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$1.0735 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$1.0735 \times 10^0$
MCSMA	$5.5509 \times 10^{-247}$	$0.0000 \times 10^0$	$3.7675 \times 10^0$	$5.5509 \times 10^{-247}$	$0.0000 \times 10^0$	$3.7675 \times 10^0$
DE	$6.3804 \times 10^{-15}$	$1.3750 \times 10^{-14}$	$3.2827 \times 10^1$	$6.3804 \times 10^{-15}$	$1.3750 \times 10^{-14}$	$3.2827 \times 10^1$
	F7		F8		F9	
	mean	std	mean	mean	std	mean
ISMA	$5.2004 \times 10^{-5}$	$4.4680 \times 10^{-5}$	$6.5535 \times 10^4$	$5.2004 \times 10^{-5}$	$4.4680 \times 10^{-5}$	$6.5535 \times 10^4$
SMA	$1.8109 \times 10^{-3}$	$1.9112 \times 10^{-3}$	$-1.256 \times 10^4$	$1.8109 \times 10^{-3}$	$1.9112 \times 10^{-3}$	$-1.256 \times 10^4$
CSMA	$1.0466 \times 10^{-5}$	$7.1026 \times 10^{-6}$	$6.5535 \times 10^4$	$1.0466 \times 10^{-5}$	$7.1026 \times 10^{-6}$	$6.5535 \times 10^4$
MCSMA	$2.8153 \times 10^{-4}$	$1.4821 \times 10^{-4}$	$-1.256 \times 10^4$	$2.8153 \times 10^{-4}$	$1.4821 \times 10^{-4}$	$-1.256 \times 10^4$
DE	$2.4715 \times 10^{-3}$	$4.9474 \times 10^{-4}$	$-1.244 \times 10^4$	$2.4715 \times 10^{-3}$	$4.9474 \times 10^{-4}$	$-1.244 \times 10^4$
	F10		F11		F12	
	mean	std	mean	mean	std	mean
ISMA	$8.8818 \times 10^{-16}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$8.8818 \times 10^{-16}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$
SMA	$8.8818 \times 10^{-16}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$8.8818 \times 10^{-16}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$
CSMA	$8.8818 \times 10^{-16}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$8.8818 \times 10^{-16}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$
MCSMA	$8.8818 \times 10^{-16}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$8.8818 \times 10^{-16}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$
DE	$7.7568 \times 10^{-15}$	$9.0135 \times 10^{-16}$	$0.0000 \times 10^0$	$7.7568 \times 10^{-15}$	$9.0135 \times 10^{-16}$	$0.0000 \times 10^0$
	F13		F14		F15	
	mean	std	mean	mean	std	mean
ISMA	$1.3498 \times 10^{-32}$	$5.5674 \times 10^{-48}$	$9.9800 \times 10^{-1}$	$1.3498 \times 10^{-32}$	$5.5674 \times 10^{-48}$	$9.9800 \times 10^{-1}$
SMA	$4.8249 \times 10^{-3}$	$7.4218 \times 10^{-3}$	$1.3350 \times 10^0$	$4.8249 \times 10^{-3}$	$7.4218 \times 10^{-3}$	$1.3350 \times 10^0$
CSMA	$4.3078 \times 10^{-3}$	$6.3340 \times 10^{-3}$	$1.2955 \times 10^0$	$4.3078 \times 10^{-3}$	$6.3340 \times 10^{-3}$	$1.2955 \times 10^0$
MCSMA	$1.3498 \times 10^{-32}$	$5.5674 \times 10^{-48}$	$9.9800 \times 10^{-1}$	$1.3498 \times 10^{-32}$	$5.5674 \times 10^{-48}$	$9.9800 \times 10^{-1}$
DE	$1.3498 \times 10^{-32}$	$5.5674 \times 10^{-48}$	$1.0311 \times 10^0$	$1.3498 \times 10^{-32}$	$5.5674 \times 10^{-48}$	$1.0311 \times 10^0$

**Table A5.** *Cont.*

	F16		F17		F18	
	mean	std	mean	mean	std	mean
ISMA	$-1.032 \times 10^0$	$1.2770 \times 10^{-8}$	$3.9838 \times 10^{-1}$	$-1.032 \times 10^0$	$1.2770 \times 10^{-8}$	$3.9838 \times 10^{-1}$
SMA	$-8.2436 \times 10^{-1}$	$4.1923 \times 10^{-1}$	$4.1640 \times 10^{-1}$	$-8.2436 \times 10^{-1}$	$4.1923 \times 10^{-1}$	$4.1640 \times 10^{-1}$
CSMA	$-1.031 \times 10^0$	$1.1109 \times 10^{-3}$	$4.1829 \times 10^{-1}$	$-1.031 \times 10^0$	$1.1109 \times 10^{-3}$	$4.1829 \times 10^{-1}$
MCSMA	$-1.031 \times 10^0$	$6.5572 \times 10^{-4}$	$3.9865 \times 10^{-1}$	$-1.031 \times 10^0$	$6.5572 \times 10^{-4}$	$3.9865 \times 10^{-1}$
DE	<b><math>-1.031 \times 10^0</math></b>	<b><math>6.7752 \times 10^{-16}</math></b>	<b><math>3.9789 \times 10^{-1}</math></b>	<b><math>-1.031 \times 10^0</math></b>	<b><math>6.7752 \times 10^{-16}</math></b>	<b><math>3.9789 \times 10^{-1}</math></b>
	F19		F20		F21	
	mean	std	mean	mean	std	mean
ISMA	$-3.863 \times 10^0$	$1.1037 \times 10^{-4}$	$-3.163 \times 10^0$	$-3.863 \times 10^0$	$1.1037 \times 10^{-4}$	$-3.163 \times 10^0$
SMA	$-3.782 \times 10^0$	$9.4398 \times 10^{-2}$	$-2.958 \times 10^0$	$-3.782 \times 10^0$	$9.4398 \times 10^{-2}$	$-2.958 \times 10^0$
CSMA	$-3.795 \times 10^0$	$7.9965 \times 10^{-2}$	$-2.901 \times 10^0$	$-3.795 \times 10^0$	$7.9965 \times 10^{-2}$	$-2.901 \times 10^0$
MCSMA	$-3.861 \times 10^0$	$1.9880 \times 10^{-3}$	$-3.042 \times 10^0$	$-3.861 \times 10^0$	$1.9880 \times 10^{-3}$	$-3.042 \times 10^0$
DE	<b><math>-3.862 \times 10^0</math></b>	<b><math>2.7101 \times 10^{-15}</math></b>	<b><math>-3.321 \times 10^0</math></b>	<b><math>-3.862 \times 10^0</math></b>	<b><math>2.7101 \times 10^{-15}</math></b>	<b><math>-3.321 \times 10^0</math></b>
	F22		F23		F24	
	mean	std	mean	mean	std	mean
ISMA	$-1.040 \times 10^1$	$3.3560 \times 10^{-6}$	$-1.054 \times 10^1$	$-1.040 \times 10^1$	$3.3560 \times 10^{-6}$	$-1.054 \times 10^1$
SMA	$-1.032 \times 10^1$	$9.7684 \times 10^{-2}$	$-1.044 \times 10^1$	$-1.032 \times 10^1$	$9.7684 \times 10^{-2}$	$-1.044 \times 10^1$
CSMA	$-9.877 \times 10^0$	$1.2268 \times 10^0$	$-1.041 \times 10^1$	$-9.877 \times 10^0$	$1.2268 \times 10^0$	$-1.041 \times 10^1$
MCSMA	$-1.040 \times 10^1$	$6.2358 \times 10^{-6}$	$-1.054 \times 10^1$	$-1.040 \times 10^1$	$6.2358 \times 10^{-6}$	$-1.054 \times 10^1$
DE	<b><math>-1.040 \times 10^1</math></b>	<b><math>1.8067 \times 10^{-15}</math></b>	<b><math>-1.053 \times 10^1</math></b>	<b><math>-1.040 \times 10^1</math></b>	<b><math>1.8067 \times 10^{-15}</math></b>	<b><math>-1.053 \times 10^1</math></b>
	F25		F26		F27	
	mean	std	mean	mean	std	mean
ISMA	$3.4989 \times 10^3$	$2.2734 \times 10^2$	<b><math>2.5000 \times 10^3</math></b>	$3.4989 \times 10^3$	$2.2734 \times 10^2$	<b><math>2.5000 \times 10^3</math></b>
SMA	$1.0429 \times 10^4$	$2.8215 \times 10^4$	$2.5169 \times 10^3$	$1.0429 \times 10^4$	$2.8215 \times 10^4$	$2.5169 \times 10^3$
CSMA	$4.7397 \times 10^3$	$1.2900 \times 10^3$	$2.5000 \times 10^3$	$4.7397 \times 10^3$	$1.2900 \times 10^3$	$2.5000 \times 10^3$
MCSMA	$3.6251 \times 10^3$	$1.8988 \times 10^2$	$2.5000 \times 10^3$	$3.6251 \times 10^3$	$1.8988 \times 10^2$	$2.5000 \times 10^3$
DE	<b><math>2.3554 \times 10^3</math></b>	<b><math>8.2085 \times 10^1</math></b>	$2.6152 \times 10^3$	<b><math>2.3554 \times 10^3</math></b>	<b><math>8.2085 \times 10^1</math></b>	$2.6152 \times 10^3$
	F28		F29		F30	
	mean	std	mean	mean	std	mean
ISMA	<b><math>2.7000 \times 10^3</math></b>	<b><math>0.0000 \times 10^0</math></b>	$2.7147 \times 10^3$	<b><math>2.7000 \times 10^3</math></b>	<b><math>0.0000 \times 10^0</math></b>	$2.7147 \times 10^3$
SMA	$2.7000 \times 10^3$	$0.0000 \times 10^0$	$2.7732 \times 10^3$	$2.7000 \times 10^3$	$0.0000 \times 10^0$	$2.7732 \times 10^3$
CSMA	$2.7000 \times 10^3$	$0.0000 \times 10^0$	$2.7172 \times 10^3$	$2.7000 \times 10^3$	$0.0000 \times 10^0$	$2.7172 \times 10^3$
MCSMA	$2.7000 \times 10^3$	$0.0000 \times 10^0$	$2.7788 \times 10^3$	$2.7000 \times 10^3$	$0.0000 \times 10^0$	$2.7788 \times 10^3$
DE	$2.7066 \times 10^3$	$8.5796 \times 10^{-1}$	<b><math>2.7003 \times 10^3</math></b>	$2.7066 \times 10^3$	$8.5796 \times 10^{-1}$	<b><math>2.7003 \times 10^3</math></b>
	F31		F32		F33	
	mean	std	mean	mean	std	mean
ISMA	<b><math>3.0000 \times 10^3</math></b>	<b><math>0.0000 \times 10^0</math></b>	<b><math>3.1000 \times 10^3</math></b>	<b><math>3.0000 \times 10^3</math></b>	<b><math>0.0000 \times 10^0</math></b>	<b><math>3.1000 \times 10^3</math></b>
SMA	$4.1186 \times 10^3$	$1.9606 \times 10^3$	$2.8989 \times 10^7$	$4.1186 \times 10^3$	$1.9606 \times 10^3$	$2.8989 \times 10^7$
CSMA	$3.0000 \times 10^3$	$0.0000 \times 10^0$	$3.1000 \times 10^3$	$3.0000 \times 10^3$	$0.0000 \times 10^0$	$3.1000 \times 10^3$
MCSMA	$5.4386 \times 10^3$	$1.1178 \times 10^3$	$4.0742 \times 10^7$	$5.4386 \times 10^3$	$1.1178 \times 10^3$	$4.0742 \times 10^7$
DE	$3.6286 \times 10^3$	$2.4807 \times 10^1$	$1.2080 \times 10^5$	$3.6286 \times 10^3$	$2.4807 \times 10^1$	$1.2080 \times 10^5$

**Table A6.** Wilcoxon signed-rank test results between the SMA variants and the original SMA and DE algorithms.

Function	SMA	CSMA	MCSMA	DE
F1	$1.7344 \times 10^{-6}$	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$1.7344 \times 10^{-6}$
F2	$1.7344 \times 10^{-6}$	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$1.7344 \times 10^{-6}$
F3	$1.7344 \times 10^{-6}$	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$1.7344 \times 10^{-6}$

**Table A6.** *Cont.*

Function	SMA	CSMA	MCSMA	DE
F4	$1.7344 \times 10^{-6}$	$1.0000 \times 10^0$	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$
F5	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$	$2.3438 \times 10^{-2}$	$1.7344 \times 10^{-6}$
F6	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$	$1.0000 \times 10^0$	$1.0000 \times 10^0$
F7	$2.3534 \times 10^{-6}$	$4.0715 \times 10^{-5}$	$2.6033 \times 10^{-6}$	$1.7344 \times 10^{-6}$
F8	$1.6503 \times 10^{-1}$	$1.2720 \times 10^{-1}$	$1.3851 \times 10^{-1}$	$1.6268 \times 10^{-1}$
F9	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$5.0000 \times 10^{-1}$
F10	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$1.0135 \times 10^{-7}$
F11	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$1.0000 \times 10^0$
F12	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$	$1.0000 \times 10^0$	$1.0000 \times 10^0$
F13	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$	$1.0000 \times 10^0$	$1.0000 \times 10^0$
F14	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$	$1.0000 \times 10^0$	$1.0000 \times 10^0$
F15	$2.8786 \times 10^{-6}$	$2.6033 \times 10^{-6}$	$6.7328 \times 10^{-1}$	$3.5888 \times 10^{-4}$
F16	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$
F17	$1.2381 \times 10^{-5}$	$8.4661 \times 10^{-6}$	$9.5899 \times 10^{-1}$	$1.7344 \times 10^{-6}$
F18	$7.3433 \times 10^{-1}$	$4.0483 \times 10^{-1}$	$1.1973 \times 10^{-3}$	$1.7344 \times 10^{-6}$
F19	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$	$2.6033 \times 10^{-6}$	$1.7344 \times 10^{-6}$
F20	$6.3391 \times 10^{-6}$	$6.3391 \times 10^{-6}$	$2.6033 \times 10^{-6}$	$1.7344 \times 10^{-6}$
F21	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$	$9.0993 \times 10^{-1}$	$3.1123 \times 10^{-5}$
F22	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$	$1.9569 \times 10^{-2}$	$1.7344 \times 10^{-6}$
F23	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$	$4.2843 \times 10^{-1}$	$1.7344 \times 10^{-6}$
F24	$6.9838 \times 10^{-6}$	$2.5967 \times 10^{-5}$	$3.1618 \times 10^{-3}$	$1.7344 \times 10^{-6}$
F25	$3.1123 \times 10^{-5}$	$1.1265 \times 10^{-5}$	$4.2767 \times 10^{-2}$	$1.7344 \times 10^{-6}$
F26	$2.5000 \times 10^{-1}$	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$4.3205 \times 10^{-8}$
F27	$5.0000 \times 10^{-1}$	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$1.7344 \times 10^{-6}$
F28	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$1.0000 \times 10^0$	$1.7344 \times 10^{-6}$
F29	$6.5213 \times 10^{-6}$	$1.8326 \times 10^{-3}$	$1.6789 \times 10^{-5}$	$1.7344 \times 10^{-6}$
F30	$3.7896 \times 10^{-6}$	$1.0000 \times 10^0$	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$
F31	$4.8828 \times 10^{-4}$	$1.0000 \times 10^0$	$2.5631 \times 10^{-6}$	$1.7344 \times 10^{-6}$
F32	$7.8125 \times 10^{-3}$	$1.0000 \times 10^0$	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$
F33	$3.7896 \times 10^{-6}$	$1.0000 \times 10^0$	$1.7344 \times 10^{-6}$	$1.7344 \times 10^{-6}$
+/=/-	25/8/0	16/16/1	15/18/1	16/7/10

**Table A7.** Average ranking values using the Friedman test.

Algorithm	ISMA	SMA	CSMA	MCSMA	DE
AVR	2.256060606	3.847979798	3.202525253	2.909595996	2.783838384
rank	1	5	4	3	2

**Table A8.** Comparison of the numerical results obtained by ISMA and other advanced methods.

	F1		F2		F3	
	mean	std	mean	mean	std	mean
ISMA	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$0.0000 \times 10^0$
MPEDA	$5.6838 \times 10^{-223}$	$0.0000 \times 10^0$	$2.0352 \times 10^{-109}$	$5.6838 \times 10^{-223}$	$0.0000 \times 10^0$	$2.0352 \times 10^{-109}$
LSHADE	$8.6954 \times 10^{-203}$	$0.0000 \times 10^0$	$2.6224 \times 10^{-85}$	$8.6954 \times 10^{-203}$	$0.0000 \times 10^0$	$2.6224 \times 10^{-85}$
ALCPSO	$4.5530 \times 10^{-186}$	$0.0000 \times 10^0$	$1.0128 \times 10^{-6}$	$4.5530 \times 10^{-186}$	$0.0000 \times 10^0$	$1.0128 \times 10^{-6}$
CLPSO	$2.7917 \times 10^{-34}$	$2.0632 \times 10^{-34}$	$5.6730 \times 10^{-21}$	$2.7917 \times 10^{-34}$	$2.0632 \times 10^{-34}$	$5.6730 \times 10^{-21}$
CESCA	$1.0264 \times 10^3$	$7.6509 \times 10^2$	$7.2069 \times 10^0$	$1.0264 \times 10^3$	$7.6509 \times 10^2$	$7.2069 \times 10^0$
IGWO	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$5.4179 \times 10^{-260}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$5.4179 \times 10^{-260}$
BMWOA	$8.7826 \times 10^{-4}$	$1.9389 \times 10^{-3}$	$8.5362 \times 10^{-3}$	$8.7826 \times 10^{-4}$	$1.9389 \times 10^{-3}$	$8.5362 \times 10^{-3}$
OBLGWO	$2.6476 \times 10^{-281}$	$0.0000 \times 10^0$	$5.6311 \times 10^{-142}$	$2.6476 \times 10^{-281}$	$0.0000 \times 10^0$	$5.6311 \times 10^{-142}$

Table A8. Cont.

	F4		F5		F6	
	mean	std	mean	mean	std	mean
ISMA	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$5.6931 \times 10^{-12}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$5.6931 \times 10^{-12}$
MPEDE	$1.3923 \times 10^{-5}$	$2.6447 \times 10^{-5}$	$1.1960 \times 10^0$	$1.3923 \times 10^{-5}$	$2.6447 \times 10^{-5}$	$1.1960 \times 10^0$
LSHADE	$1.3040 \times 10^{-4}$	$2.3249 \times 10^{-4}$	$5.3155 \times 10^{-1}$	$1.3040 \times 10^{-4}$	$2.3249 \times 10^{-4}$	$5.3155 \times 10^{-1}$
ALCPSO	$2.6029 \times 10^{-5}$	$3.4443 \times 10^{-5}$	$2.5603 \times 10^1$	$2.6029 \times 10^{-5}$	$3.4443 \times 10^{-5}$	$2.5603 \times 10^1$
CLPSO	$1.3451 \times 10^0$	$2.6110 \times 10^{-1}$	$6.5461 \times 10^{-1}$	$1.3451 \times 10^0$	$2.6110 \times 10^{-1}$	$6.5461 \times 10^{-1}$
CESCA	$2.0286 \times 10^1$	$7.5303 \times 10^0$	$2.4759 \times 10^5$	$2.0286 \times 10^1$	$7.5303 \times 10^0$	$2.4759 \times 10^5$
IGWO	$7.5149 \times 10^{-26}$	$4.1158 \times 10^{-25}$	$2.3186 \times 10^1$	$7.5149 \times 10^{-26}$	$4.1158 \times 10^{-25}$	$2.3186 \times 10^1$
BMWOA	$3.6139 \times 10^{-3}$	$3.9430 \times 10^{-3}$	$3.9781 \times 10^{-3}$	$3.6139 \times 10^{-3}$	$3.9430 \times 10^{-3}$	$3.9781 \times 10^{-3}$
OBLGWO	$2.7133 \times 10^{-157}$	$1.4861 \times 10^{-156}$	$2.6112 \times 10^1$	$2.7133 \times 10^{-157}$	$1.4861 \times 10^{-156}$	$2.6112 \times 10^1$
	F7		F8		F9	
	mean	std	mean	mean	std	mean
ISMA	$9.4873 \times 10^{-5}$	$6.6385 \times 10^{-5}$	$6.5535 \times 10^4$	$9.4873 \times 10^{-5}$	$6.6385 \times 10^{-5}$	$6.5535 \times 10^4$
MPEDE	$3.2148 \times 10^{-3}$	$1.6021 \times 10^{-3}$	$-1.187 \times 10^4$	$3.2148 \times 10^{-3}$	$1.6021 \times 10^{-3}$	$-1.187 \times 10^4$
LSHADE	$6.5393 \times 10^{-3}$	$5.0546 \times 10^{-3}$	$-1.895 \times 10^3$	$6.5393 \times 10^{-3}$	$5.0546 \times 10^{-3}$	$-1.895 \times 10^3$
ALCPSO	$9.6181 \times 10^{-2}$	$3.9035 \times 10^{-2}$	$-1.147 \times 10^4$	$9.6181 \times 10^{-2}$	$3.9035 \times 10^{-2}$	$-1.147 \times 10^4$
CLPSO	$2.6752 \times 10^{-3}$	$7.7407 \times 10^{-4}$	$-1.256 \times 10^4$	$2.6752 \times 10^{-3}$	$7.7407 \times 10^{-4}$	$-1.256 \times 10^4$
CESCA	$5.3895 \times 10^{-1}$	$3.4475 \times 10^{-1}$	$-3.901 \times 10^3$	$5.3895 \times 10^{-1}$	$3.4475 \times 10^{-1}$	$-3.901 \times 10^3$
IGWO	$2.7827 \times 10^{-4}$	$2.2936 \times 10^{-4}$	$-7.436 \times 10^3$	$2.7827 \times 10^{-4}$	$2.2936 \times 10^{-4}$	$-7.436 \times 10^3$
BMWOA	$1.1610 \times 10^{-3}$	$8.5016 \times 10^{-4}$	$-1.257 \times 10^4$	$1.1610 \times 10^{-3}$	$8.5016 \times 10^{-4}$	$-1.257 \times 10^4$
OBLGWO	$2.3640 \times 10^{-5}$	$2.4037 \times 10^{-5}$	$-1.253 \times 10^4$	$2.3640 \times 10^{-5}$	$2.4037 \times 10^{-5}$	$-1.253 \times 10^4$
	F10		F11		F12	
	mean	std	mean	mean	std	mean
ISMA	$8.8818 \times 10^{-16}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$8.8818 \times 10^{-16}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$
MPEDE	$2.0353 \times 10^0$	$6.7054 \times 10^{-1}$	$1.5065 \times 10^{-2}$	$2.0353 \times 10^0$	$6.7054 \times 10^{-1}$	$1.5065 \times 10^{-2}$
LSHADE	$3.3455 \times 10^{-14}$	$3.7417 \times 10^{-15}$	$1.2274 \times 10^{-2}$	$3.3455 \times 10^{-14}$	$3.7417 \times 10^{-15}$	$1.2274 \times 10^{-2}$
ALCPSO	$8.3257 \times 10^{-1}$	$8.5957 \times 10^{-1}$	$1.7674 \times 10^{-2}$	$8.3257 \times 10^{-1}$	$8.5957 \times 10^{-1}$	$1.7674 \times 10^{-2}$
CLPSO	$1.2138 \times 10^{-14}$	$2.4831 \times 10^{-15}$	$0.0000 \times 10^0$	$1.2138 \times 10^{-14}$	$2.4831 \times 10^{-15}$	$0.0000 \times 10^0$
CESCA	$6.7169 \times 10^0$	$1.9070 \times 10^0$	$1.0700 \times 10^1$	$6.7169 \times 10^0$	$1.9070 \times 10^0$	$1.0700 \times 10^1$
IGWO	$4.6777 \times 10^{-15}$	$9.0135 \times 10^{-16}$	$0.0000 \times 10^0$	$4.6777 \times 10^{-15}$	$9.0135 \times 10^{-16}$	$0.0000 \times 10^0$
BMWOA	$4.6994 \times 10^{-3}$	$5.2250 \times 10^{-3}$	$1.7612 \times 10^{-3}$	$4.6994 \times 10^{-3}$	$5.2250 \times 10^{-3}$	$1.7612 \times 10^{-3}$
OBLGWO	$8.8818 \times 10^{-16}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$	$8.8818 \times 10^{-16}$	$0.0000 \times 10^0$	$0.0000 \times 10^0$
	F13		F14		F15	
	mean	std	mean	mean	std	mean
ISMA	$1.3498 \times 10^{-32}$	$5.5674 \times 10^{-48}$	$9.9800 \times 10^{-1}$	$1.3498 \times 10^{-32}$	$5.5674 \times 10^{-48}$	$9.9800 \times 10^{-1}$
MPEDE	$3.2626 \times 10^{-1}$	$9.4775 \times 10^{-1}$	$9.9800 \times 10^{-1}$	$3.2626 \times 10^{-1}$	$9.4775 \times 10^{-1}$	$9.9800 \times 10^{-1}$
LSHADE	$1.1303 \times 10^{-1}$	$4.0369 \times 10^{-1}$	$9.9800 \times 10^{-1}$	$1.1303 \times 10^{-1}$	$4.0369 \times 10^{-1}$	$9.9800 \times 10^{-1}$
ALCPSO	$1.1403 \times 10^{-2}$	$3.4415 \times 10^{-2}$	$9.9800 \times 10^{-1}$	$1.1403 \times 10^{-2}$	$3.4415 \times 10^{-2}$	$9.9800 \times 10^{-1}$
CLPSO	$1.3498 \times 10^{-32}$	$5.5674 \times 10^{-48}$	$9.9800 \times 10^{-1}$	$1.3498 \times 10^{-32}$	$5.5674 \times 10^{-48}$	$9.9800 \times 10^{-1}$
CESCA	$4.2932 \times 10^5$	$6.0065 \times 10^5$	$3.0471 \times 10^0$	$4.2932 \times 10^5$	$6.0065 \times 10^5$	$3.0471 \times 10^0$
IGWO	$1.6832 \times 10^{-2}$	$3.2997 \times 10^{-2}$	$9.9800 \times 10^{-1}$	$1.6832 \times 10^{-2}$	$3.2997 \times 10^{-2}$	$9.9800 \times 10^{-1}$
BMWOA	$1.7335 \times 10^{-4}$	$5.7395 \times 10^{-4}$	$9.9800 \times 10^{-1}$	$1.7335 \times 10^{-4}$	$5.7395 \times 10^{-4}$	$9.9800 \times 10^{-1}$
OBLGWO	$2.4316 \times 10^{-2}$	$3.9405 \times 10^{-2}$	$9.9800 \times 10^{-1}$	$2.4316 \times 10^{-2}$	$3.9405 \times 10^{-2}$	$9.9800 \times 10^{-1}$
	F16		F17		F18	
	mean	std	mean	mean	std	mean
ISMA	$-1.032 \times 10^0$	$6.9699 \times 10^{-9}$	$3.9808 \times 10^{-1}$	$-1.032 \times 10^0$	$6.9699 \times 10^{-9}$	$3.9808 \times 10^{-1}$
MPEDE	$-1.032 \times 10^0$	$6.7752 \times 10^{-16}$	$3.9789 \times 10^{-1}$	$-1.032 \times 10^0$	$6.7752 \times 10^{-16}$	$3.9789 \times 10^{-1}$
LSHADE	$-1.032 \times 10^0$	$6.7752 \times 10^{-16}$	$3.9789 \times 10^{-1}$	$-1.032 \times 10^0$	$6.7752 \times 10^{-16}$	$3.9789 \times 10^{-1}$
ALCPSO	$-1.032 \times 10^0$	$5.6082 \times 10^{-16}$	$3.9789 \times 10^{-1}$	$-1.032 \times 10^0$	$5.6082 \times 10^{-16}$	$3.9789 \times 10^{-1}$
CLPSO	$-1.032 \times 10^0$	$6.4539 \times 10^{-16}$	$3.9789 \times 10^{-1}$	$-1.032 \times 10^0$	$6.4539 \times 10^{-16}$	$3.9789 \times 10^{-1}$
CESCA	$-1.026 \times 10^0$	$5.9057 \times 10^{-3}$	$7.0892 \times 10^{-1}$	$-1.026 \times 10^0$	$5.9057 \times 10^{-3}$	$7.0892 \times 10^{-1}$

Table A8. Cont.

IGWO	$-1.032 \times 10^0$	$2.2583 \times 10^{-13}$	$3.9789 \times 10^{-1}$	$-1.032 \times 10^0$	$2.2583 \times 10^{-13}$	$3.9789 \times 10^{-1}$
BMWOA	<b><math>-1.031 \times 10^0</math></b>	<b><math>4.4024 \times 10^{-16}</math></b>	$3.9789 \times 10^{-1}$	<b><math>-1.031 \times 10^0</math></b>	<b><math>4.4024 \times 10^{-16}</math></b>	$3.9789 \times 10^{-1}$
OBLGWO	$-1.032 \times 10^0$	$9.0832 \times 10^{-9}$	$3.9801 \times 10^{-1}$	$-1.032 \times 10^0$	$9.0832 \times 10^{-9}$	$3.9801 \times 10^{-1}$
	F19		F20		F21	
	mean	std	mean	mean	std	mean
ISMA	$-3.863 \times 10^0$	$9.7215 \times 10^{-5}$	$-3.159 \times 10^0$	$-3.863 \times 10^0$	$9.7215 \times 10^{-5}$	$-3.159 \times 10^0$
MPEDE	$-3.863 \times 10^0$	$2.7101 \times 10^{-15}$	$-3.271 \times 10^0$	$-3.863 \times 10^0$	$2.7101 \times 10^{-15}$	$-3.271 \times 10^0$
LSHADE	$-3.863 \times 10^0$	$1.3042 \times 10^{-4}$	$-1.952 \times 10^0$	$-3.863 \times 10^0$	$1.3042 \times 10^{-4}$	$-1.952 \times 10^0$
ALCPSO	<b><math>-3.862 \times 10^0</math></b>	<b><math>2.5243 \times 10^{-15}</math></b>	$-3.274 \times 10^0$	<b><math>-3.862 \times 10^0</math></b>	<b><math>2.5243 \times 10^{-15}</math></b>	$-3.274 \times 10^0$
CLPSO	$-3.863 \times 10^0$	$2.7101 \times 10^{-15}$	<b><math>-3.322 \times 10^0</math></b>	$-3.863 \times 10^0$	$2.7101 \times 10^{-15}$	<b><math>-3.322 \times 10^0</math></b>
CESCA	$-3.610 \times 10^0$	$1.6803 \times 10^{-1}$	$-2.176 \times 10^0$	$-3.610 \times 10^0$	$1.6803 \times 10^{-1}$	$-2.176 \times 10^0$
IGWO	$-3.863 \times 10^0$	$1.0500 \times 10^{-9}$	$-3.251 \times 10^0$	$-3.863 \times 10^0$	$1.0500 \times 10^{-9}$	$-3.251 \times 10^0$
BMWOA	$-3.863 \times 10^0$	$1.5134 \times 10^{-14}$	$-3.290 \times 10^0$	$-3.863 \times 10^0$	$1.5134 \times 10^{-14}$	$-3.290 \times 10^0$
OBLGWO	$-3.863 \times 10^0$	$1.3281 \times 10^{-6}$	$-3.223 \times 10^0$	$-3.863 \times 10^0$	$1.3281 \times 10^{-6}$	$-3.223 \times 10^0$
	F22		F23		F24	
	mean	std	mean	mean	std	mean
ISMA	$-1.040 \times 10^1$	$5.9774 \times 10^{-6}$	$-1.054 \times 10^1$	$-1.040 \times 10^1$	$5.9774 \times 10^{-6}$	$-1.054 \times 10^1$
MPEDE	$-9.542 \times 10^0$	$2.2747 \times 10^0$	$-9.817 \times 10^0$	$-9.542 \times 10^0$	$2.2747 \times 10^0$	$-9.817 \times 10^0$
LSHADE	$-1.023 \times 10^1$	$9.6292 \times 10^{-1}$	<b><math>-1.053 \times 10^1</math></b>	$-1.023 \times 10^1$	$9.6292 \times 10^{-1}$	<b><math>-1.053 \times 10^1</math></b>
ALCPSO	$-9.876 \times 10^0$	$1.6093 \times 10^0$	$-9.997 \times 10^0$	$-9.876 \times 10^0$	$1.6093 \times 10^0$	$-9.997 \times 10^0$
CLPSO	$-1.040 \times 10^1$	$5.7155 \times 10^{-9}$	$-1.054 \times 10^1$	$-1.040 \times 10^1$	$5.7155 \times 10^{-9}$	$-1.054 \times 10^1$
CESCA	$-1.091 \times 10^0$	$4.2964 \times 10^{-1}$	$-1.172 \times 10^0$	$-1.091 \times 10^0$	$4.2964 \times 10^{-1}$	$-1.172 \times 10^0$
IGWO	$-9.166 \times 10^0$	$2.2815 \times 10^0$	$-1.018 \times 10^1$	$-9.166 \times 10^0$	$2.2815 \times 10^0$	$-1.018 \times 10^1$
BMWOA	<b><math>-1.040 \times 10^1</math></b>	<b><math>9.4634 \times 10^{-11}</math></b>	$-1.054 \times 10^1$	<b><math>-1.040 \times 10^1</math></b>	<b><math>9.4634 \times 10^{-11}</math></b>	$-1.054 \times 10^1$
OBLGWO	$-1.040 \times 10^1$	$3.5332 \times 10^{-5}$	$-1.054 \times 10^1$	$-1.040 \times 10^1$	$3.5332 \times 10^{-5}$	$-1.054 \times 10^1$
	F25		F26		F27	
	mean	std	mean	mean	std	mean
ISMA	$3.4696 \times 10^3$	$1.5041 \times 10^2$	<b><math>2.5000 \times 10^3</math></b>	$3.4696 \times 10^3$	$1.5041 \times 10^2$	<b><math>2.5000 \times 10^3</math></b>
MPEDE	$2.5483 \times 10^3$	$2.1545 \times 10^2$	$2.6152 \times 10^3$	$2.5483 \times 10^3$	$2.1545 \times 10^2$	$2.6152 \times 10^3$
LSHADE	$2.4214 \times 10^3$	$1.2400 \times 10^2$	$2.6152 \times 10^3$	$2.4214 \times 10^3$	$1.2400 \times 10^2$	$2.6152 \times 10^3$
ALCPSO	$2.6317 \times 10^3$	$1.8339 \times 10^2$	$2.6153 \times 10^3$	$2.6317 \times 10^3$	$1.8339 \times 10^2$	$2.6153 \times 10^3$
CLPSO	<b><math>2.4055 \times 10^3</math></b>	<b><math>8.0140 \times 10^1</math></b>	$2.6152 \times 10^3$	<b><math>2.4055 \times 10^3</math></b>	<b><math>8.0140 \times 10^1</math></b>	$2.6152 \times 10^3$
CESCA	$5.5650 \times 10^3$	$9.4857 \times 10^2$	$3.0675 \times 10^3$	$5.5650 \times 10^3$	$9.4857 \times 10^2$	$3.0675 \times 10^3$
IGWO	$2.5661 \times 10^3$	$1.8331 \times 10^2$	$2.6206 \times 10^3$	$2.5661 \times 10^3$	$1.8331 \times 10^2$	$2.6206 \times 10^3$
BMWOA	$2.9003 \times 10^3$	$1.9433 \times 10^2$	$2.5005 \times 10^3$	$2.9003 \times 10^3$	$1.9433 \times 10^2$	$2.5005 \times 10^3$
OBLGWO	$2.6973 \times 10^3$	$2.3782 \times 10^2$	$2.6188 \times 10^3$	$2.6973 \times 10^3$	$2.3782 \times 10^2$	$2.6188 \times 10^3$
	F28		F29		F30	
	mean	std	mean	mean	std	mean
ISMA	<b><math>2.7000 \times 10^3</math></b>	<b><math>0.0000 \times 10^0</math></b>	$2.7181 \times 10^3$	<b><math>2.7000 \times 10^3</math></b>	<b><math>0.0000 \times 10^0</math></b>	$2.7181 \times 10^3$
MPEDE	$2.7112 \times 10^3$	$4.6410 \times 10^0$	$2.7202 \times 10^3$	$2.7112 \times 10^3$	$4.6410 \times 10^0$	$2.7202 \times 10^3$
LSHADE	$2.7056 \times 10^3$	$3.3938 \times 10^0$	$2.7104 \times 10^3$	$2.7056 \times 10^3$	$3.3938 \times 10^0$	$2.7104 \times 10^3$
ALCPSO	$2.7124 \times 10^3$	$5.0481 \times 10^0$	$2.7553 \times 10^3$	$2.7124 \times 10^3$	$5.0481 \times 10^0$	$2.7553 \times 10^3$
CLPSO	$2.7072 \times 10^3$	$9.5781 \times 10^{-1}$	<b><math>2.7004 \times 10^3</math></b>	$2.7072 \times 10^3$	$9.5781 \times 10^{-1}$	<b><math>2.7004 \times 10^3</math></b>
CESCA	$2.7206 \times 10^3$	$8.6833 \times 10^0$	$2.7123 \times 10^3$	$2.7206 \times 10^3$	$8.6833 \times 10^0$	$2.7123 \times 10^3$
IGWO	$2.7107 \times 10^3$	$2.5492 \times 10^0$	$2.7007 \times 10^3$	$2.7107 \times 10^3$	$2.5492 \times 10^0$	$2.7007 \times 10^3$
BMWOA	$2.7000 \times 10^3$	$1.1250 \times 10^{-2}$	$2.7006 \times 10^3$	$2.7000 \times 10^3$	$1.1250 \times 10^{-2}$	$2.7006 \times 10^3$
OBLGWO	$2.7000 \times 10^3$	$0.0000 \times 10^0$	$2.7005 \times 10^3$	$2.7000 \times 10^3$	$0.0000 \times 10^0$	$2.7005 \times 10^3$
	F31		F32		F33	
	mean	std	mean	mean	std	mean
ISMA	<b><math>3.0000 \times 10^3</math></b>	<b><math>0.0000 \times 10^0</math></b>	<b><math>3.1000 \times 10^3</math></b>	<b><math>3.0000 \times 10^3</math></b>	<b><math>0.0000 \times 10^0</math></b>	<b><math>3.1000 \times 10^3</math></b>
MPEDE	$3.9778 \times 10^3$	$3.4239 \times 10^2$	$1.6519 \times 10^6$	$3.9778 \times 10^3$	$3.4239 \times 10^2$	$1.6519 \times 10^6$
LSHADE	$3.7470 \times 10^3$	$8.7552 \times 10^1$	$2.9248 \times 10^5$	$3.7470 \times 10^3$	$8.7552 \times 10^1$	$2.9248 \times 10^5$





Table A11. Cont.

V-Shaped Family		
Name	TFs	Graphs
TFV1	$T(x_i^j(t)) = \left  \operatorname{erf}\left(\frac{\sqrt{\pi}}{2} x_i^j(t)\right) \right  = \left  \frac{\sqrt{\pi}}{2} \int_0^{2x_i^j(t)} e^{-t^2} dt \right $	
TFV2	$T(x_i^j(t)) =  \tanh \tanh x_i^j(t) $	
TFV3	$T(x_i^j(t)) = \left  \frac{x_i^j(t)}{\sqrt{1+x_i^j(t)^2}} \right $	
TFV4	$T(x_i^j(t)) = \left  \frac{2}{\pi} \arctan\left(\frac{\sqrt{\pi}}{2} x_i^j(t)\right) \right $	

Note:  $x_i^j(t)$  denotes the  $i$ -th element on  $j$ -th dimension in the position vector.

Table A12. Characteristics of gene expression datasets.

Datasets	Samples	Genes	Categories
Colon	62	2000	2
SRBCT	83	2309	4
Leukemia	72	7131	2
Brain_Tumor1	90	5920	5
Brain_Tumor2	50	10,367	4
CNS	60	7130	2
DLBCL	77	5470	4
Leukemia1	72	5328	5
Leukemia2	72	11,225	3
Lung_Cancer	203	12,601	3
Prostate_Tumor	102	10,509	2
Tumors_9	60	5726	9
Tumors_11	174	12,533	11
Tumors_14	308	15,009	26

Table A13. Overall results of eight versions of BISMA according to S-shaped and V-shaped TFs in terms of average number of the selected genes.

Datasets	Metrics	BISMA_S1	BISMA_S2	BISMA_S3	BISMA_S4	BISMA_V1	BISMA_V2	BISMA_V3	BISMA_V4
Colon	std	143.6448	157.4435	173.4187	162.0243	<b>0.4216</b>	0.9718	0.6992	0.6992
	avg	307.5000	464.5000	476.5000	498.0000	<b>1.0000</b>	1.0000	1.0000	1.0000
	std	138.2114	95.9528	156.2727	154.4375	2.9515	2.9364	1.9322	<b>1.4337</b>
SRBCT	avg	376.5000	465.5000	566.0000	565.0000	4.0000	5.0000	4.5000	<b>4.5000</b>
	std	589.3556	296.6164	135.8554	64.6241	0.9487	1.2517	0.3162	<b>0.3162</b>
	avg	1595.5000	1359.0000	1738.5000	1755.0000	1.0000	1.0000	1.0000	<b>1.0000</b>
Brain_Tumor1	std	926.7275	778.2962	44.3653	560.9920	147.6392	8.0939	<b>11.1679</b>	19.3724
	avg	1050.0000	1319.5000	1451.5000	1461.5000	2.0000	3.0000	<b>2.5000</b>	2.5000
	std	755.7944	978.0951	955.6762	430.0868	1.7512	0.9944	1.2293	<b>0.4831</b>
Brain_Tumor2	avg	1938.0000	2509.5000	2510.0000	2529.5000	1.0000	2.0000	1.5000	<b>1.0000</b>
	std	504.4472	867.2775	732.4766	489.2598	2.2136	0.5164	<b>0.0000</b>	0.4216
	avg	1685.0000	1720.5000	1805.0000	1935.0000	1.0000	1.0000	<b>1.0000</b>	1.0000
DLBCL	std	292.2214	169.4024	129.8839	79.6573	<b>0.0000</b>	0.6750	0.3162	0.6325
	avg	490.5000	1295.0000	1334.5000	1371.5000	<b>1.0000</b>	1.0000	1.0000	1.0000
	std	348.2874	536.8715	66.7750	77.7810	1.3499	1.8135	<b>1.1005</b>	1.2472
Leukemia1	avg	1163.0000	1271.5000	1283.0000	1328.5000	2.0000	2.0000	<b>2.0000</b>	2.0000
	std	731.5217	497.7822	232.6141	929.4172	3.7357	1.6633	<b>1.4142</b>	1.4181
	avg	1255.5000	2532.5000	2673.5000	2737.5000	3.0000	2.5000	<b>1.5000</b>	3.0000
Lung_Cancer	std	1191.4138	1241.8645	1162.5447	623.9975	19.8161	<b>16.1593</b>	29.0746	93.8666
	avg	3066.0000	3122.0000	3111.0000	3162.0000	23.5000	<b>19.0000</b>	16.5000	15.5000
	std	1573.8463	1270.5976	1119.6290	1279.6201	6.2405	37.9867	<b>1.0750</b>	1.8529
Prostate_Tumor	avg	2540.0000	2709.0000	2631.5000	2760.5000	3.5000	2.0000	<b>2.5000</b>	2.5000
	std	785.7851	856.2383	533.6090	595.3492	243.1681	<b>42.0502</b>	595.2484	139.8144
	avg	1376.5000	1409.5000	1698.0000	1421.0000	1.0000	<b>2.0000</b>	2.5000	4.0000
Tumors_9	std	1040.6752	1660.6726	1391.3213	1285.5454	<b>108.9483</b>	288.1741	948.9861	248.4647
	avg	3118.5000	4607.0000	4642.0000	3287.0000	<b>210.0000</b>	304.5000	374.5000	233.0000
	std	2353.3411	1657.2601	974.4708	1551.2076	1520.8509	930.6287	<b>618.4779</b>	966.3795
Tumors_14	avg	4920.0000	7469.0000	7450.0000	6775.0000	1143.5000	760.5000	<b>540.5000</b>	569.5000
	ARV	5.7143	6.3893	6.8143	7.0393	2.5286	2.6536	2.4464	<b>2.4143</b>
	Rank	5	6	7	8	3	4	2	<b>1</b>



Table A16. Cont.

Datasets	Metrics	BISMA_S1	BISMA_S2	BISMA_S3	BISMA_S4	BISMA_V1	BISMA_V2	BISMA_V3	BISMA_V4
Leukemia	std	4.6463	7.1863	8.4485	4.176	5.6477	8.9345	7.1624	5.6551
	avg	288.1026	369.0878	418.859	300.3361	312.7438	281.3737	263.277	262.0363
Brain_Tumor1	std	15.0486	5.518	5.9123	3.6483	6.5985	7.0206	5.2887	9.7095
	avg	257.1141	329.2853	355.1649	265.6545	268.0102	235.1687	226.2937	221.4106
Brain_Tumor2	std	26.0923	16.4663	5.829	5.9109	5.7821	6.9892	8.4514	4.4363
	avg	394.7483	557.1407	417.7936	408.0532	429.047	378.0446	403.66	366.1612
CNS	std	18.5258	8.2571	4.9416	4.7855	5.2549	5.1788	6.3485	2.5764
	avg	282.115	399.2233	297.2468	292.9844	305.7291	270.9286	305.3575	257.4227
DLBCL	std	13.4459	7.3986	3.7698	3.2965	6.6934	6.881	5.9037	6.3564
	avg	229.0604	318.173	239.9178	235.4501	243.1863	222.2096	206.4178	207.6545
Leukemia1	std	13.0915	7.1145	4.2194	3.5637	4.3786	5.3226	3.4246	4.1366
	avg	221.6625	306.661	230.06	226.9516	236.7801	206.948	201.3261	199.8014
Leukemia2	std	27.9557	27.8952	7.3565	6.2185	9.6514	10.0649	7.2691	9.5984
	avg	454.5811	626.5679	467.8857	467.3684	482.2521	424.8834	411.641	408.7297
Lung_Cancer	std	40.0181	14.4133	21.3431	26.7837	47.3963	37.8825	48.654	32.9511
	avg	835.7816	1064.939	847.6348	828.0133	677.3208	558.4493	534.8904	521.5364
Prostate_Tumor	std	25.1417	10.4573	6.7311	10.3808	19.9367	16.2087	12.0796	24.8174
	avg	470.1901	659.3352	485.7299	477.1534	464.5947	415.6169	390.0605	389.1298
Tumors_9	std	13.5588	8.8614	3.2316	4.0011	2.6109	4.0259	4.3268	3.487
	avg	231.0626	333.3597	240.6433	238.7621	246.7118	220.015	208.5856	206.8161
Tumors_11	std	39.8624	15.6572	18.8506	15.9373	46.4145	36.5902	20.4801	15.6274
	avg	744.1785	985.7713	758.463	752.3035	630.7326	555.8758	502.5768	483.1388
Tumors_14	std	73.1984	62.0491	69.1097	103.7124	77.7274	74.0669	49.1133	57.7599
	avg	1560.365	1901.44	1556.638	1541.604	1087.476	880.2872	760.1826	723.8812
ARV		4.7	7.5143	6.4071	5.0571	5.8929	2.8643	2.1357	1.4286
Rank		4	8	7	5	6	3	2	1

Table A17. Parameter settings.

Optimizers	Parameters	Value
bGWO	$a_{max}$	2
	$a_{min}$	0
BPSO	Min inertia weight	0.4
	Min inertia weight	0.9
	$c1, c2$	0.2
bWOA	$a_{max}$	2
	$a_{min}$	0

Table A18. Comparison of BISMA with other gene selection optimizers in terms of average number of the selected genes.

Datasets	Metrics	BISMA	BSMA	bGWO	BGSA	BPSO	bALO	BBA	BSSA	bWOA
Colon	std	0.5164	29.5727	15.9753	23.2178	18.7901	26.9081	57.2076	413.9399	1.6499
	avg	1	46	153.5	769	899	876	818	424.5	2
SRBCT	std	1.2649	20.5721	15.2567	28.0515	17.2321	21.7348	88.7612	234.9426	1.8974
	avg	3	33.5	192	898.5	1023	996	936	1073.5	4
Leukemia	std	0.42164	21.9699	41.075	22.2264	31.8531	27.3595	180.2885	1254.8997	0.91894
	avg	1	36	791.5	3106	3354	3288	2850	3427	2
Brain_Tumor1	std	3.1429	78.3272	37.8001	45.6636	31.3739	42.0132	104.9288	1333.051	1.2649
	avg	3.5	65	631	2559	2766	2737	2449.5	2646.5	3
Brain_Tumor2	std	1.7029	240.6062	75.5373	55.0019	55.9691	46.9871	135.9838	2454.5883	1.1785
	avg	2.5	156	1148.5	4672.5	4914.5	4864.5	4209	2946.5	2.5
CNS	std	0.31623	136.7067	42.7265	96.6304	35.9623	50.9117	198.0223	1551.1952	3.2335
	avg	1	87.5	852	3171	3386.5	3344.5	2985	3293	2
DLBCL	std	0.42164	33.4865	23.7957	48.9182	24.6162	37.7601	156.4013	833.0272	0.99443
	avg	1	40.5	571.5	2329.5	2522.5	2489	2245	2625.5	2
Leukemia1	std	0.8165	25.3588	33.1832	39.1324	20.8017	31.6665	190.6413	1124.43	1.2649
	avg	2	40	550.5	2303	2473.5	2419	2132	2538.5	3.5
Leukemia2	std	1.2649	22.3617	46.4113	57.6102	51.1196	42.9973	252.5475	2534.8708	1.1972
	avg	2.5	55	1245.5	5021.5	5320.5	5272.5	4592	5412.5	3
Lung_Cancer	std	27.247	240.5198	66.0041	77.9308	42.3663	48.9689	688.2611	2587.9333	13.898
	avg	10	172	1504	5750.5	6030	5947.5	5097.5	6092	5.5



Table A20. Cont.

Datasets	Metrics	BISMA	BSMA	bGWO	BGSA	BPSO	bALO	BBA	BSSA	bWOA
Tumors_9	std	$8.9737 \times 10^{-4}$	$6.7888 \times 10^{-2}$	$8.5899 \times 10^{-2}$	$6.2023 \times 10^{-4}$	$6.6747 \times 10^{-2}$	$7.7087 \times 10^{-2}$	$1.9970 \times 10^{-1}$	$1.2797 \times 10^{-1}$	$2.6422 \times 10^{-5}$
	avg	$6.9857 \times 10^{-5}$	$1.5194 \times 10^{-3}$	$5.8854 \times 10^{-3}$	$2.2083 \times 10^{-2}$	$2.4162 \times 10^{-2}$	$2.3411 \times 10^{-2}$	$2.3214 \times 10^{-2}$	$2.4703 \times 10^{-2}$	$2.6196 \times 10^{-5}$
Tumors_11	std	$2.1319 \times 10^{-2}$	$5.7197 \times 10^{-2}$	$1.9912 \times 10^{-2}$	$5.4119 \times 10^{-2}$	$4.6341 \times 10^{-2}$	$4.8083 \times 10^{-2}$	$5.9891 \times 10^{-2}$	$5.7273 \times 10^{-2}$	$3.4943 \times 10^{-2}$
	avg	$1.3923 \times 10^{-3}$	$6.6026 \times 10^{-3}$	$6.4171 \times 10^{-3}$	$2.3604 \times 10^{-2}$	$4.9392 \times 10^{-2}$	$7.6599 \times 10^{-2}$	$1.2055 \times 10^{-1}$	$6.7590 \times 10^{-2}$	$2.6123 \times 10^{-2}$
Tumors_14	std	$5.5130 \times 10^{-2}$	$5.3345 \times 10^{-2}$	$5.7317 \times 10^{-2}$	$6.8032 \times 10^{-2}$	$3.5029 \times 10^{-2}$	$5.2960 \times 10^{-2}$	$7.0141 \times 10^{-2}$	$9.5849 \times 10^{-2}$	$6.4709 \times 10^{-2}$
	avg	$2.5180 \times 10^{-1}$	$2.7576 \times 10^{-1}$	$1.7527 \times 10^{-1}$	$2.1745 \times 10^{-1}$	$2.8210 \times 10^{-1}$	$2.5053 \times 10^{-1}$	$3.1859 \times 10^{-1}$	$2.7012 \times 10^{-1}$	$2.2178 \times 10^{-1}$
ARV Rank		1.6964 1	1.6964 1	3.7571 3	4.2286 4	5.9 5	7.1286 9	6.8143 8	6.8071 7	6.6429 6

Table A21. Comparison of BISMA with other gene selection optimizers in terms of average computational time.

Datasets	Metrics	BISMA	BSMA	bGWO	BGSA	BPSO	bALO	BBA	BSSA	bWOA
Colon	std	0.93215	0.55407	0.098098	0.11018	0.076472	0.069336	<b>0.19183</b>	0.23189	0.4158
	avg	79.1933	35.9194	14.2079	7.2215	4.2471	4.1295	<b>13.9446</b>	23.2622	26.0384
SRBCT	std	2.1619	0.51149	0.15702	0.11402	0.13534	0.16851	<b>0.25667</b>	0.3163	0.37599
	avg	93.6061	41.244	16.2856	8.8596	5.4073	5.2877	<b>16.3119</b>	27.1393	29.9446
Leukemia	std	7.2303	1.8022	0.3074	0.42745	0.27052	0.35454	<b>0.51794</b>	0.93854	1.2515
	avg	256.4992	122.7257	44.8501	23.5815	12.5313	12.2151	<b>45.0914</b>	79.2949	89.7865
Brain_Tumor1	std	6.9684	1.0527	0.278	0.45035	0.47769	0.32493	<b>0.49618</b>	1.069	1.2276
	avg	220.5351	103.2569	38.7039	21.6636	13.2493	12.7106	<b>40.4449</b>	68.416	74.6861
Brain_Tumor2	std	4.4718	2.0085	0.40876	0.4924	0.4705	0.34669	<b>0.57963</b>	1.4683	1.9697
	avg	354.245	176.7797	63.4666	29.9924	13.3176	12.3337	<b>60.1049</b>	110.4912	131.5993
CNS	std	4.7455	1.4788	0.51911	0.26787	0.19856	0.2563	<b>0.6056</b>	0.94437	1.1606
	avg	248.504	122.8625	44.5969	22.0423	10.8899	10.3101	<b>43.5037</b>	77.9093	89.9202
DLBCL	std	5.3919	0.93684	0.23064	0.16063	0.29357	0.16353	<b>0.47267</b>	0.61759	1.1042
	avg	200.6234	94.9785	35.3048	18.7326	10.6001	10.3269	<b>35.7383</b>	61.9793	69.2286
Leukemia1	std	4.3623	1.1156	0.31638	0.35829	0.27141	0.19161	<b>0.52207</b>	0.86822	0.85614
	avg	194.386	92.0658	34.0793	17.7135	9.9582	9.5621	<b>34.4391</b>	60.048	66.7794
Leukemia2	std	7.374	3.0726	0.49382	0.61819	0.5241	0.50114	<b>0.61738</b>	1.9491	2.7261
	avg	399.2129	192.557	69.7935	36.9327	19.1668	18.0316	<b>69.7835</b>	123.9545	144.0557
Lung_Cancer	std	41.6799	4.5389	1.0736	3.7226	4.3019	3.6112	<b>4.3848</b>	2.089	2.7099
	avg	515.1456	233.6435	99.2504	93.316	77.8452	75.9999	<b>127.7409</b>	190.8128	167.6969
Prostate_Tumor	std	17.4231	2.7956	0.56853	0.55518	0.82953	0.5596	<b>1.1379</b>	1.3348	1.8037
	avg	383.2946	183.4421	68.6138	42.0024	25.8001	25.2016	<b>72.0151</b>	122.9783	133.436
Tumors_9	std	2.9367	1.1039	0.25397	0.356	0.42194	0.23273	<b>0.39294</b>	0.77615	1.0106
	avg	203.6074	98.814	36.386	18.1879	9.237	8.8404	<b>35.6073</b>	62.8321	71.9579
Tumors_11	std	11.6164	3.5569	1.045	2.7062	2.8401	3.2718	<b>3.7758</b>	1.9198	1.6793
	avg	465.5284	226.3375	93.2904	78.8383	61.7486	60.114	<b>113.7264</b>	175.511	163.5641
Tumors_14	std	78.3081	9.9354	<b>2.1979</b>	13.6846	7.0692	10.4616	8.8472	4.4011	5.6434
	avg	664.032	309.3361	<b>159.1758</b>	202.5748	176.6571	176.1249	235.2403	308.2242	212.441
ARV Rank		9 9	7.95 8	4.1857 4	3.1 3	1.8929 2	<b>1.2571</b> 1	4.6643 5	6.2643 6	6.6857 7

References

- Ye, M.; Wang, W.; Yao, C.; Fan, R.; Wang, P. Gene Selection Method for Microarray Data Classification Using Particle Swarm Optimization and Neighborhood Rough Set. *Curr. Bioinform.* **2019**, *14*, 422–431. [CrossRef]
- Wang, S.; Kong, W.; Zeng, W.; Hong, X. Hybrid Binary Imperialist Competition Algorithm and Tabu Search Approach for Feature Selection Using Gene Expression Data. *Biomed Res. Int.* **2016**, *2016*, 9721713. [CrossRef] [PubMed]
- Uthayan, K. A novel microarray gene selection and classification using intelligent dynamic grey wolf optimization. *Genetika* **2019**, *51*, 805–828. [CrossRef]
- Shukla, A.; Singh, P.; Vardhan, M. Gene selection for cancer types classification using novel hybrid metaheuristics approach. *Swarm Evol. Comput.* **2020**, *54*, 100661. [CrossRef]
- Sharma, A.; Rani, R. C-HMOSHSSA: Gene selection for cancer classification using multi-objective meta-heuristic and machine learning methods. *Comput. Methods Programs Biomed.* **2019**, *178*, 219–235. [CrossRef]
- Mohamad, M.; Omatu, S.; Deris, S.; Yoshioka, M.; Abdullah, A.; Ibrahim, Z. An enhancement of binary particle swarm optimization for gene selection in classifying cancer classes. *Algorithms Mol. Biol.* **2013**, *8*, 15. [CrossRef]
- Mabu, A.; Prasad, R.; Yadav, R. Gene Expression Dataset Classification Using Artificial Neural Network and Clustering-Based Feature Selection. *Int. J. Swarm Intell. Res.* **2020**, *11*, 65–86. [CrossRef]
- Jin, C.; Jin, S. Gene selection approach based on improved swarm intelligent optimisation algorithm for tumour classification. *Int. J. Syst. Biol.* **2016**, *10*, 107–115. [CrossRef]
- Dabba, A.; Tari, A.; Meftali, S.; Mokhtari, R. Gene selection and classification of microarray data method based on mutual information and moth flame algorithm. *Expert Syst. Appl.* **2021**, *166*, 114012. [CrossRef]

10. Dabba, A.; Tari, A.; Meftali, S. Hybridization of Moth flame optimization algorithm and quantum computing for gene selection in microarray data. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *12*, 2731–2750. [[CrossRef](#)]
11. Xu, X.; Li, J.; Chen, H.-L. Enhanced support vector machine using parallel particle swarm optimization. In Proceedings of the 2014 10th International Conference on Natural Computation (ICNC), Xiamen, China, 19–21 August 2014.
12. Alshamlan, H.; Badr, G.; Alohal, Y. mRMR-ABC: A Hybrid Gene Selection Algorithm for Cancer Classification Using Microarray Gene Expression Profiling. *Biomed Res. Int.* **2015**, *2015*, 604910. [[CrossRef](#)]
13. Alshamlan, H.; Badr, G.; Alohal, Y. Genetic Bee Colony (GBC) algorithm: A new gene selection method for microarray cancer classification. *Comput. Biol. Chem.* **2015**, *56*, 49–60. [[CrossRef](#)]
14. Liu, B.; Tian, M.; Zhang, C.; Li, X. Discrete Biogeography Based Optimization for Feature Selection in Molecular Signatures. *Mol. Inform.* **2015**, *34*, 197–215. [[CrossRef](#)]
15. Best, M.; Sol, N.; In't Veld, S.G.J.G.; Vancura, A.; Muller, M.; Niemeijer, A.N.; Fejes, A.V.; Tjon Kon Fat, L.A.; Huis In't Veld, A.E.; Leurs, C.; et al. Swarm Intelligence-Enhanced Detection of Non-Small-Cell Lung Cancer Using Tumor-Educated Platelets. *Cancer Cell* **2017**, *32*, 238. [[CrossRef](#)]
16. Best, M.; In't Veld, S.; Sol, N.; Wurdinger, T. RNA sequencing and swarm intelligence-enhanced classification algorithm development for blood-based disease diagnostics using spliced blood platelet RNA. *Nat. Protoc.* **2019**, *14*, 1206–1234. [[CrossRef](#)]
17. Ang, J.; Mirzal, A.; Haron, H.; Hamed, H. Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection. *IEEE-Acm Trans. Comput. Biol. Bioinform.* **2016**, *13*, 971–989. [[CrossRef](#)]
18. Sun, Y.; Lu, C.; Li, X. The Cross-Entropy Based Multi-Filter Ensemble Method for Gene Selection. *Genes* **2018**, *9*, 258. [[CrossRef](#)]
19. Mundra, P.; Rajapakse, J. SVM-RFE With MRMR Filter for Gene Selection. *IEEE Trans. Nanobioscience* **2010**, *9*, 31–37. [[CrossRef](#)]
20. Li, J.; Su, L.; Pang, Z. A Filter Feature Selection Method Based on MFA Score and Redundancy Excluding and Its Application to Tumor Gene Expression Data Analysis. *Interdiscip. Sci.-Comput. Life Sci.* **2015**, *7*, 391–396. [[CrossRef](#)]
21. Kim, Y.; Yoon, Y. A genetic filter for cancer classification on gene expression data. *Bio-Med. Mater. Eng.* **2015**, *26*, S1993–S2002. [[CrossRef](#)]
22. Chandrashekar, G.; Sahin, F. A survey on feature selection methods. *Comput. Electr. Eng.* **2014**, *40*, 16–28. [[CrossRef](#)]
23. Bolon-Canedo, V.; Sanchez-Marono, N.; Alonso-Betanzos, A. A review of feature selection methods on synthetic data. *Knowl. Inf. Syst.* **2013**, *34*, 483–519. [[CrossRef](#)]
24. Lee, S.; Xu, Z.; Li, T.; Yang, Y. A novel bagging C4.5 algorithm based on wrapper feature selection for supporting wise clinical decision making. *J. Biomed. Inform.* **2018**, *78*, 144–155.
25. Al-Thanoon, N.; Qasim, O.; Algarnal, Z. Tuning parameter estimation in SCAD-support vector machine using firefly algorithm with application in gene selection and cancer classification. *Comput. Biol. Med.* **2018**, *103*, 262–268. [[CrossRef](#)]
26. Yang, A.; Cao, T.; Li, R.; Liao, B. A Hybrid Gene Selection Method for Cancer Classification Based on Clustering Algorithm and Euclidean Distance. *J. Comput. Theor. Nanosci.* **2012**, *9*, 611–615. [[CrossRef](#)]
27. Wang, L.; Han, B. Hybrid feature selection method for gene expression analysis. *Electron. Lett.* **2014**, *50*, 1269–1270.
28. Sungheetha, A.; Sharma, R. Extreme Learning Machine and Fuzzy K-Nearest Neighbour Based Hybrid Gene Selection Technique for Cancer Classification. *J. Med. Imaging Health Inform.* **2016**, *6*, 1652–1656. [[CrossRef](#)]
29. Lu, H.; Chen, J.; Yan, K.; Jin, Q.; Xue, Y.; Gao, Z. A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing* **2017**, *256*, 56–62. [[CrossRef](#)]
30. Cao, B.; Zhao, J.; Lv, Z.; Yang, P. Diversified personalized recommendation optimization based on mobile data. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 2133–2139. [[CrossRef](#)]
31. Cao, B.; Fan, S.; Zhao, J.; Tian, S.; Zheng, Z.; Yan, Y.; Yang, P. Large-scale many-objective deployment optimization of edge servers. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 3841–3849. [[CrossRef](#)]
32. Zhang, M.; Chen, Y.; Susilo, W. PPO-CPQ: A privacy-preserving optimization of clinical pathway query for e-healthcare systems. *IEEE Internet Things J.* **2020**, *7*, 10660–10672. [[CrossRef](#)]
33. Wang, L.; Wang, Y.; Chang, Q. Feature selection methods for big data bioinformatics: A survey from the search perspective. *Methods* **2016**, *111*, 21–31. [[CrossRef](#)]
34. Prasartvit, T.; Banharnsakun, A.; Kaewkamnerdpong, B.; Achalakul, T. Reducing bioinformatics data dimension with ABC-kNN. *Neurocomputing* **2013**, *116*, 367–381. [[CrossRef](#)]
35. Li, S.; Chen, H.; Wang, M.; Heidari, A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst. -Int. J. Escience* **2020**, *111*, 300–323. [[CrossRef](#)]
36. Mirjalili, S.; Dong, J.S.; Lewis, A. *Nature-Inspired Optimizers: Theories, Literature Reviews and Applications*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 811.
37. Chen, X.; Tianfield, H.; Mei, C.; Du, W.; Liu, G. Biogeography-based learning particle swarm optimization. *Soft Comput.* **2017**, *21*, 7519–7541. [[CrossRef](#)]
38. Liang, J.; Qin, A.; Suganthan, P.; Baskar, S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans. Evol. Comput.* **2006**, *10*, 281–295. [[CrossRef](#)]
39. Cai, Z.; Gu, J.; Luo, J.; Zhang, Q.; Chen, H.; Pan, Z.; Li, Y.; Li, C. Evolving an optimal kernel extreme learning machine by using an enhanced grey wolf optimization strategy. *Expert Syst. Appl.* **2019**, *138*, 112814. [[CrossRef](#)]
40. Reddy, K.; Panwar, L.; Panigrahi, B.; Kumar, R. Binary whale optimization algorithm: A new metaheuristic approach for profit-based unit commitment problems in competitive electricity markets. *Eng. Optim.* **2019**, *51*, 369–389. [[CrossRef](#)]

41. Kouadri, R.; Slimani, L.; Bouktir, T. Slime mould algorithm for practical optimal power flow solutions incorporating stochastic wind power and static var compensator device. *Electr. Eng. Electromechanics* **2020**, *45*–54. [[CrossRef](#)]
42. Mostafa, M.; Rezk, H.; Aly, M.; Ahmed, E. A new strategy based on slime mould algorithm to extract the optimal model parameters of solar PV panel. *Sustain. Energy Technol. Assess.* **2020**, *42*, 100849. [[CrossRef](#)]
43. Kumar, C.; Raj, T.; Premkumar, M. A new stochastic slime mould optimization algorithm for the estimation of solar photovoltaic cell parameters. *Optik* **2020**, *223*, 165277. [[CrossRef](#)]
44. Abdel-Basset, M.; Chang, V.; Mohamed, R. HSMA\_WOA: A hybrid novel Slime mould algorithm with whale optimization algorithm for tackling the image segmentation problem of chest X-ray images. *Appl. Soft Comput.* **2020**, *95*, 106642. [[CrossRef](#)] [[PubMed](#)]
45. Sun, K.; Jia, H.; Li, Y.; Jiang, Z. Hybrid improved slime mould algorithm with adaptive beta hill climbing for numerical optimization. *J. Intell. Fuzzy Syst.* **2021**, *40*, 1667–1679. [[CrossRef](#)]
46. Zubaidi, S.; Abdulkareem, I.H.; Hashim, K.S.; Al-Bugharbee, H.; Ridha, H.M.; Gharghan, S.K.; Al-Qaim, F.F.; Muradov, M.; Kot, P.; Al-Khaddar, R. Hybridised Artificial Neural Network Model with Slime Mould Algorithm: A Novel Methodology for Prediction of Urban Stochastic Water Demand. *Water* **2020**, *12*, 2692. [[CrossRef](#)]
47. Zhang, Y.; Liu, R.; Heidari, A.A.; Wang, X.; Chen, Y.; Wang, M.; Chen, H. Towards augmented kernel extreme learning models for bankruptcy prediction: Algorithmic behavior and comprehensive analysis. *Neurocomputing* **2021**, *430*, 185–212. [[CrossRef](#)]
48. Chen, Z.; Liu, W. An Efficient Parameter Adaptive Support Vector Regression Using K-Means Clustering and Chaotic Slime Mould Algorithm. *IEEE Access* **2020**, *8*, 156851–156862. [[CrossRef](#)]
49. Baliarsingh, S.; Vipsita, S. Chaotic emperor penguin optimised extreme learning machine for microarray cancer classification. *Let Syst. Biol.* **2020**, *14*, 85–95. [[CrossRef](#)]
50. Banu, P.; Azar, A.; Inbarani, H. Fuzzy firefly clustering for tumour and cancer analysis. *Int. J. Model. Identif. Control.* **2017**, *27*, 92–103. [[CrossRef](#)]
51. Chen, L.; Li, J.; Chang, M. Cancer Diagnosis and Disease Gene Identification via Statistical Machine Learning. *Curr. Bioinform.* **2020**, *15*, 956–962. [[CrossRef](#)]
52. Mahendran, N.; Vincent, P.; Srinivasan, K.; Chang, C. Machine Learning Based Computational Gene Selection Models: A Survey, Performance Evaluation, Open Issues, and Future Research Directions. *Front. Genet.* **2020**, *11*, 603808. [[CrossRef](#)]
53. Tan, M.; Chang, S.; Cheah, P.; Yap, H. Integrative machine learning analysis of multiple gene expression profiles in cervical cancer. *PeerJ* **2018**, *6*, e5285. [[CrossRef](#)] [[PubMed](#)]
54. Zhou, Y.; Lin, J.; Guo, H. Feature subset selection via an improved discretization-based particle swarm optimization. *Appl. Soft Comput.* **2021**, *98*, 106794. [[CrossRef](#)]
55. Sadeghian, Z.; Akbari, E.; Nematzadeh, H. A hybrid feature selection method based on information theory and binary butterfly optimization algorithm. *Eng. Appl. Artif. Intell.* **2021**, *97*, 104079. [[CrossRef](#)]
56. Coletto-Alcudia, V.; Vega-Rodriguez, M. Artificial Bee Colony algorithm based on Dominance (ABCD) for a hybrid gene selection method. *Knowl.-Based Syst.* **2020**, *205*, 106323. [[CrossRef](#)]
57. Lee, J.; Choi, I.; Jun, C. An efficient multivariate feature ranking method for gene selection in high-dimensional microarray data. *Expert Syst. Appl.* **2021**, *166*, 113971. [[CrossRef](#)]
58. Khani, E.; Mahmoodian, H. Phase diagram and ridge logistic regression in stable gene selection. *Biocybern. Biomed. Eng.* **2020**, *40*, 965–976. [[CrossRef](#)]
59. Chen, K.; Wang, K.; Wang, K.; Angelia, M. Applying particle swarm optimization-based decision tree classifier for cancer classification on gene expression data. *Appl. Soft Comput.* **2014**, *24*, 773–780. [[CrossRef](#)]
60. Mohamad, M.; Omatu, S.; Deris, S.; Yoshioka, M. A Modified Binary Particle Swarm Optimization for Selecting the Small Subset of Informative Genes From Gene Expression Data. *IEEE Trans. Inf. Technol. Biomed.* **2011**, *15*, 813–822. [[CrossRef](#)]
61. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst. -Int. J. Escience* **2019**, *97*, 849–872. [[CrossRef](#)]
62. Yang, Y.; Chen, H.; Heidari, A.A.; Gandomi, A.H. Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Syst. Appl.* **2021**, *177*, 114864. [[CrossRef](#)]
63. Ahmadianfar, I.; Heidari, A.A.; Gandomi, A.H.; Chu, X.; Chen, H. RUN Beyond the Metaphor: An Efficient Optimization Algorithm Based on Runge Kutta Method. *Expert Syst. Appl.* **2021**, *181*, 115079. [[CrossRef](#)]
64. Tu, J.; Chen, H.; Wang, M.; Gandomi, A.H. The Colony Predation Algorithm. *J. Bionic Eng.* **2021**, *18*, 674–710. [[CrossRef](#)]
65. Ahmadianfar, I.; Heidari, A.A.; Noshadian, S.; Chen, H.; Gandomi, A.H. INFO: An Efficient Optimization Algorithm based on Weighted Mean of Vectors. *Expert Syst. Appl.* **2022**, *195*, 116516. [[CrossRef](#)]
66. Hussien, A.G.; Heidari, A.A.; Ye, X.; Liang, G.; Chen, H.; Pan, Z. Boosting whale optimization with evolution strategy and Gaussian random walks: An image segmentation method. *Eng. Comput.* **2022**. [[CrossRef](#)]
67. Yu, H.; Song, J.; Chen, C.; Heidari, A.A.; Liu, J.; Chen, H.; Zaguia, A.; Mafarja, M. Image segmentation of Leaf Spot Diseases on Maize using multi-stage Cauchy-enabled grey wolf algorithm. *Eng. Appl. Artif. Intell.* **2022**, *109*, 104653. [[CrossRef](#)]
68. Lai, X.; Zhou, Y. Analysis of multiobjective evolutionary algorithms on the biobjective traveling salesman problem (1, 2). *Multimed. Tools Appl.* **2020**, *79*, 30839–30860. [[CrossRef](#)]
69. Hu, J.; Chen, H.; Heidari, A.A.; Wang, M.; Zhang, X.; Chen, Y.; Pan, Z. Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection. *Knowl.-Based Syst.* **2021**, *213*, 106684. [[CrossRef](#)]

70. Hu, J.; Gui, W.; Heidari, A.A.; Cai, Z.; Liang, G.; Chen, H.; Pan, Z. Dispersed foraging slime mould algorithm: Continuous and binary variants for global optimization and wrapper-based feature selection. *Knowl.-Based Syst.* **2022**, *237*, 107761. [[CrossRef](#)]
71. Chen, H.; Wang, M.; Zhao, X. A multi-strategy enhanced sine cosine algorithm for global optimization and constrained practical engineering problems. *Appl. Math. Comput.* **2020**, *369*, 124872. [[CrossRef](#)]
72. Yu, H.; Qiao, S.; Heidari, A.A.; Bi, C.; Chen, H. Individual Disturbance and Attraction Repulsion Strategy Enhanced Seagull Optimization for Engineering Design. *Mathematics* **2022**, *10*, 276. [[CrossRef](#)]
73. Yu, H.; Yuan, K.; Li, W.; Zhao, N.; Chen, W.; Huang, C.; Chen, H.; Wang, M. Improved Butterfly Optimizer-Configured Extreme Learning Machine for Fault Diagnosis. *Complexity* **2021**, *2021*, 6315010. [[CrossRef](#)]
74. Han, X.; Han, Y.; Chen, Q.; Li, J.; Sang, H.; Liu, Y.; Pan, Q.; Nojima, Y. Distributed Flow Shop Scheduling with Sequence-Dependent Setup Times Using an Improved Iterated Greedy Algorithm. *Complex Syst. Modeling Simul.* **2021**, *1*, 198–217. [[CrossRef](#)]
75. Gao, D.; Wang, G.-G.; Pedrycz, W. Solving fuzzy job-shop scheduling problem using DE algorithm improved by a selection mechanism. *IEEE Trans. Fuzzy Syst.* **2020**, *28*, 3265–3275. [[CrossRef](#)]
76. Wang, G.-G.; Gao, D.; Pedrycz, W. Solving multi-objective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm. *IEEE Trans. Ind. Inform.* **2022**. [[CrossRef](#)]
77. Deng, W.; Zhang, X.; Zhou, Y.; Liu, Y.; Zhou, X.; Chen, H.; Zhao, H. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Inf. Sci.* **2022**, *585*, 441–453. [[CrossRef](#)]
78. Hua, Y.; Liu, Q.; Hao, K.; Jin, Y. A Survey of Evolutionary Algorithms for Multi-Objective Optimization Problems With Irregular Pareto Fronts. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 303–318. [[CrossRef](#)]
79. Li, Q.; Chen, H.; Huang, H.; Zhao, X.; Cai, Z.; Tong, C.; Liu, W.; Tian, X. An Enhanced Grey Wolf Optimization Based Feature Selection Wrapped Kernel Extreme Learning Machine for Medical Diagnosis. *Comput. Math. Methods Med.* **2017**, *2017*, 9512741. [[CrossRef](#)]
80. Cai, Z.; Gu, J.; Wen, C.; Zhao, D.; Huang, C.; Huang, H.; Tong, C.; Li, J.; Chen, H. An Intelligent Parkinson's Disease Diagnostic System Based on a Chaotic Bacterial Foraging Optimization Enhanced Fuzzy KNN Approach. *Comput. Math. Methods Med.* **2018**, *2018*, 2396952. [[CrossRef](#)]
81. Dong, R.; Chen, H.; Heidari, A.A.; Turabieh, H.; Mafarja, M.; Wang, S. Boosted kernel search: Framework, analysis and case studies on the economic emission dispatch problem. *Knowl.-Based Syst.* **2021**, *233*, 107529. [[CrossRef](#)]
82. He, Z.; Yen, G.G.; Ding, J. Knee-based decision making and visualization in many-objective optimization. *IEEE Trans. Evol. Comput.* **2020**, *25*, 292–306. [[CrossRef](#)]
83. He, Z.; Yen, G.G.; Lv, J. Evolutionary multiobjective optimization with robustness enhancement. *IEEE Trans. Evol. Comput.* **2019**, *24*, 494–507. [[CrossRef](#)]
84. Ye, X.; Liu, W.; Li, H.; Wang, M.; Chi, C.; Liang, G. Modified Whale Optimization Algorithm for Solar Cell and PV Module Parameter Identification. *Complexity* **2021**, *2021*, 8878686. [[CrossRef](#)]
85. Chen, H.L.; Yang, B.; Wang, S.J.; Wang, G.; Li, H.Z.; Liu, W.B. Towards an optimal support vector machine classifier using a parallel particle swarm optimization strategy. *Appl. Math. Comput.* **2014**, *239*, 180–197. [[CrossRef](#)]
86. Wu, G.; Mallipeddi, R.; Suganthan, P.; Wang, R.; Chen, H. Differential evolution with multi-population based ensemble of mutation strategies. *Inf. Sci.* **2016**, *329*, 329–345.
87. Piotrowski, A. L-SHADE optimization algorithms with population-wide inertia. *Inf. Sci.* **2018**, *468*, 117–141. [[CrossRef](#)]
88. Chen, W.; Zhang, J.; Lin, Y.; Chen, N.; Zhan, Z.H.; Chung HS, H.; Li, Y.; Shi, Y.H. Particle Swarm Optimization with an Aging Leader and Challengers. *IEEE Trans. Evol. Comput.* **2013**, *17*, 241–258. [[CrossRef](#)]
89. Lin, A.; Wu, Q.; Heidari, A.A.; Xu, Y.; Chen, H.; Geng, W.; Li, C. Predicting Intentions of Students for Master Programs Using a Chaos-Induced Sine Cosine-Based Fuzzy K-Nearest Neighbor Classifier. *IEEE Access* **2019**, *7*, 67235–67248. [[CrossRef](#)]
90. Heidari, A.; Aljarah, I.; Faris, H.; Chen, H.; Luo, J.; Mirjalili, S. An enhanced associative learning-based exploratory whale optimizer for global optimization. *Neural Comput. Appl.* **2020**, *32*, 5185–5211. [[CrossRef](#)]
91. Heidari, A.; Abbaspour, R.; Chen, H. Efficient boosted grey wolf optimizers for global search and kernel extreme learning machine training. *Appl. Soft Comput.* **2019**, *81*, 105521. [[CrossRef](#)]
92. Wang, S.; Guo, H.; Zhang, S.; Barton, D.; Brooks, P. Analysis and prediction of double-carriage train wheel wear based on SIMPACK and neural networks. *Adv. Mech. Eng.* **2022**, *14*, 16878132221078491. [[CrossRef](#)]
93. Lv, Z.; Li, Y.; Feng, H.; Lv, H. Deep learning for security in digital twins of cooperative intelligent transportation systems. *IEEE Trans. Intell. Transp. Syst.* **2021**, 1–10. [[CrossRef](#)]
94. Lv, Z.; Chen, D.; Feng, H.; Zhu, H.; Lv, H. Digital twins in unmanned aerial vehicles for rapid medical resource delivery in epidemics. *IEEE Trans. Intell. Transp. Syst.* **2021**, 1–9. [[CrossRef](#)]
95. Zou, Q.; Xing, P.; Wei, L.; Liu, B. Gene2vec: Gene subsequence embedding for prediction of mammalian N6-methyladenosine sites from mRNA. *RNA* **2019**, *25*, 205–218. [[CrossRef](#)] [[PubMed](#)]
96. Faramarzi, A.; Heidarinejad, M.; Stephens, B.; Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowl.-Based Syst.* **2020**, *191*, 105190. [[CrossRef](#)]
97. Zhou, L.; Fan, Q.; Huang, X.; Liu, Y. Weak and strong convergence analysis of Elman neural networks via weight decay regularization. *Optimization* **2022**, 1–23. [[CrossRef](#)]
98. Fan, Q.; Zhang, Z.; Huang, X. Parameter Conjugate Gradient with Secant Equation Based Elman Neural Network and its Convergence Analysis. *Adv. Theory Simul.* **2022**, 2200047. [[CrossRef](#)]



99. Mirjalili, S.; Lewis, A. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm Evol. Comput.* **2013**, *9*, 1–14. [[CrossRef](#)]
100. Xu, Q.; Zeng, Y.; Tang, W.; Peng, W.; Xia, T.; Li, Z.; Teng, F.; Li, W.; Guo, J. Multi-task joint learning model for segmenting and classifying tongue images using a deep neural network. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2481–2489. [[CrossRef](#)]
101. Li, J.; Xu, K.; Chaudhuri, S.; Yumer, E.; Zhang, H.; Guibas, L. Grass: Generative recursive autoencoders for shape structures. *ACM Trans. Graph.* **2017**, *36*, 1–14. [[CrossRef](#)]
102. Zhao, H.; Zhu, C.; Xu, X.; Huang, H.; Xu, K. Learning practically feasible policies for online 3D bin packing. *Sci. China Inf. Sci.* **2022**, *65*, 1–17. [[CrossRef](#)]
103. Emary, E.; Zawba, H.; Hassanien, A. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **2016**, *172*, 371–381. [[CrossRef](#)]
104. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. BGSa: Binary gravitational search algorithm. *Nat. Comput.* **2010**, *9*, 727–745. [[CrossRef](#)]
105. Emary, E.; Zawbaa, H.; Hassanien, A. Binary ant lion approaches for feature selection. *Neurocomputing* **2016**, *213*, 54–65. [[CrossRef](#)]
106. Mirjalili, S.; Mirjalili, S.; Yang, X. Binary bat algorithm. *Neural Comput. Appl.* **2014**, *25*, 663–681. [[CrossRef](#)]
107. Reddy, K.; Panwar, L.; Panigrahi, B.; Kumar, R. A New Binary Variant of Sine-Cosine Algorithm: Development and Application to Solve Profit-Based Unit Commitment Problem. *Arab. J. Sci. Eng.* **2018**, *43*, 4041–4056. [[CrossRef](#)]
108. Mafarja, M.; Mirjalili, S. Whale optimization approaches for wrapper feature selection. *Appl. Soft Comput.* **2018**, *62*, 441–453. [[CrossRef](#)]
109. Örnek, B.N.; Aydemir, S.B.; Düzenli, T.; Özak, B. A novel version of slime mould algorithm for global optimization and real world engineering problems: Enhanced slime mould algorithm. *Math. Comput. Simul.* **2022**, *198*, 253–288. [[CrossRef](#)]
110. Gürses, D.; Bureerat, S.; Sait, S.M.; Yıldız, A.R. Comparison of the arithmetic optimization algorithm, the slime mold optimization algorithm, the marine predators algorithm, the salp swarm algorithm for real-world engineering applications. *Mater. Test.* **2021**, *63*, 448–452. [[CrossRef](#)]
111. Cai, Z.; Xiong, Z.; Wan, K.; Xu, Y.; Xu, F. A node selecting approach for traffic network based on artificial slime mold. *IEEE Access* **2020**, *8*, 8436–8448. [[CrossRef](#)]
112. Li, D.; Zhang, S.; Ma, X. Dynamic Module Detection in Temporal Attributed Networks of cancers. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2021**, *19*, 2219–2230. [[CrossRef](#)]
113. Ma, X.; Sun, P.G.; Gong, M. An integrative framework of heterogeneous genomic data for cancer dynamic modules based on matrix decomposition. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2020**, *19*, 305–316. [[CrossRef](#)]
114. Huang, L.; Yang, Y.; Chen, H.; Zhang, Y.; Wang, Z.; He, L. Context-aware road travel time estimation by coupled tensor decomposition based on trajectory data. *Knowl.-Based Syst.* **2022**, *245*, 108596. [[CrossRef](#)]
115. Wu, Z.; Li, R.; Xie, J.; Zhou, Z.; Guo, J.; Xu, X. A user sensitive subject protection approach for book search service. *J. Assoc. Inf. Sci. Technol.* **2020**, *71*, 183–195. [[CrossRef](#)]
116. Wu, Z.; Shen, S.; Lian, X.; Su, X.; Chen, E. A dummy-based user privacy protection approach for text information retrieval. *Knowl.-Based Syst.* **2020**, *195*, 105679. [[CrossRef](#)]
117. Wu, Z.; Shen, S.; Zhou, H.; Li, H.; Lu, C.; Zou, D. An effective approach for the protection of user commodity viewing privacy in e-commerce website. *Knowl.-Based Syst.* **2021**, *220*, 106952. [[CrossRef](#)]
118. Li, Y.; Li, X.X.; Hong, J.J.; Wang, Y.X.; Fu, J.B.; Yang, H.; Yu, C.Y.; Li, F.C.; Hu, J.; Xue, W.W.; et al. Clinical trials, progression-speed differentiating features and swiftness rule of the innovative targets of first-in-class drugs. *Brief. Bioinform.* **2020**, *21*, 649–662. [[CrossRef](#)]
119. Zhu, F.; Li, X.; Yang, S.; Chen, Y. Clinical success of drug targets prospectively predicted by in silico study. *Trends Pharmacol. Sci.* **2018**, *39*, 229–231. [[CrossRef](#)]
120. Cao, X.; Sun, X.; Xu, Z.; Zeng, B.; Guan, X. Hydrogen-Based Networked Microgrids Planning Through Two-Stage Stochastic Programming with Mixed-Integer Conic Recourse. *IEEE Trans. Autom. Sci. Eng.* **2021**, 1–14. [[CrossRef](#)]
121. Zhang, X.; Wang, J.; Wang, T.; Jiang, R. Hierarchical feature fusion with mixed convolution attention for single image dehazing. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 510–522. [[CrossRef](#)]
122. Wu, Z.; Li, G.; Shen, S.; Cui, Z.; Lian, X.; Xu, G. Constructing dummy query sequences to protect location privacy and query privacy in location-based services. *World Wide Web* **2021**, *24*, 25–49. [[CrossRef](#)]
123. Wu, Z.; Wang, R.; Li, Q.; Lian, X.; Xu, G. A location privacy-preserving system based on query range cover-up for location-based services. *IEEE Trans. Veh. Technol.* **2020**, *69*, 5244–5254. [[CrossRef](#)]
124. Cao, X.; Wang, J.; Zeng, B. A Study on the Strong Duality of Second-Order Conic Relaxation of AC Optimal Power Flow in Radial Networks. *IEEE Trans. Power Syst.* **2022**, *37*, 443–455. [[CrossRef](#)]
125. Tian, Y.; Su, X.; Su, Y.; Zhang, X. EMODMI: A multi-objective optimization based method to identify disease modules. *IEEE Trans. Emerg. Top. Comput. Intell.* **2020**, *5*, 570–582. [[CrossRef](#)]
126. Su, Y.; Li, S.; Zheng, C.; Zhang, X. A heuristic algorithm for identifying molecular signatures in cancer. *IEEE Trans. NanoBioscience* **2019**, *19*, 132–141. [[CrossRef](#)]
127. Wang, D.; Liang, Y.; Xu, D.; Feng, X.; Guan, R.J.K.-B.S. A content-based recommender system for computer science publications. *Knowl.-Based Syst.* **2018**, *157*, 1–9. [[CrossRef](#)]

128. Li, J.; Chen, C.; Chen, H.; Tong, C. Towards Context-aware Social Recommendation via Individual Trust. *Knowl.-Based Syst.* **2017**, *127*, 58–66. [[CrossRef](#)]
129. Li, J.; Lin, J. A probability distribution detection based hybrid ensemble QoS prediction approach. *Inf. Sci.* **2020**, *519*, 289–305. [[CrossRef](#)]
130. Li, J.; Zheng, X.-L.; Chen, S.-T.; Song, W.-W.; Chen, D.-R. An efficient and reliable approach for quality-of-service-aware service composition. *Inf. Sci.* **2014**, *269*, 238–254. [[CrossRef](#)]
131. Qiu, S.; Zhao, H.; Jiang, N.; Wang, Z.; Liu, L.; An, Y.; Zhao, H.; Miao, X.; Liu, R.; Fortino, G. Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges. *Inf. Fusion* **2022**, *80*, 241–265. [[CrossRef](#)]
132. Zhang, X.; Fan, C.; Xiao, Z.; Zhao, L.; Chen, H.; Chang, X. Random Reconstructed Unpaired Image-to-Image Translation. *IEEE Trans. Ind. Inform.* **2022**, *1*. [[CrossRef](#)]