



OPEN

# Advancing forensic-based investigation incorporating slime mould search for gene selection of high-dimensional genetic data

Feng Qiu<sup>1</sup>, Ali Asghar Heidari<sup>2</sup>, Yi Chen<sup>3</sup>, Huiling Chen<sup>1✉</sup> & Guoxi Liang<sup>4✉</sup>

Modern medicine has produced large genetic datasets of high dimensions through advanced gene sequencing technology, and processing these data is of great significance for clinical decision-making. Gene selection (GS) is an important data preprocessing technique that aims to select a subset of feature information to improve performance and reduce data dimensionality. This study proposes an improved wrapper GS method based on forensic-based investigation (FBI). The method introduces the search mechanism of the slime mould algorithm in the FBI to improve the original FBI; the newly proposed algorithm is named SMA\_FBI; then GS is performed by converting the continuous optimizer to a binary version of the optimizer through a transfer function. In order to verify the superiority of SMA\_FBI, experiments are first executed on the 30-function test set of CEC2017 and compared with 10 original algorithms and 10 state-of-the-art algorithms. The experimental results show that SMA\_FBI is better than other algorithms in terms of finding the optimal solution, convergence speed, and robustness. In addition, BSMA\_FBI (binary version of SMA\_FBI) is compared with 8 binary algorithms on 18 high-dimensional genetic data from the UCI repository. The results indicate that BSMA\_FBI is able to obtain high classification accuracy with fewer features selected in GS applications. Therefore, SMA\_FBI is considered an optimization tool with great potential for dealing with global optimization problems, and its binary version, BSMA\_FBI, can be used for GS tasks.

**Keywords** Forensic-based investigation, High-dimensional genetic data, Gene selection, Slime mould algorithm, Global optimization

With the rapid development of modern information technology and biomedical fields, researchers can study biomedical fields at the genetic level<sup>1-3</sup>. DNA microarray technology is a technique that enables the rapid and efficient detection and analysis of the expression levels of millions of genes<sup>4,5</sup>. These gene expression data are an essential source of information for studying the current physiological state of cells, and these data have been widely used for early diagnosis and prognosis prediction of diseases. The microarray technology shines as an emerging and valuable technology in generating high-dimensional data from a few samples<sup>6,7</sup>. However, for this type of dataset, the high feature dimensionality makes it prone to dimensional catastrophe; at the same time, the relatively small sample size causes the algorithm to be prone to overfitting, making it difficult to extract accurate patterns and correlations from it. Therefore, feature selection (FS) is performed for gene data to reduce the data dimensionality, i.e., gene selection (GS) is a method to solve this problem.

Advances in healthcare information systems require more efficient algorithms to process the various data in the system, and techniques such as machine learning and FS are well used in this field<sup>8-10</sup>. FS of gene data is becoming more prominent as a screening method to reduce irrelevant microarray gene data and improve classification accuracy. FS efficiently identifies and extracts the most valuable features from a large data set<sup>11,12</sup>. This approach allows researchers and analysts to focus on the most relevant and informative features, which improves the accuracy and efficiency of subsequent analyses and models and helps to make informed decisions in different disciplines, such as genetics and biology. In reducing the number of features, elimination of redundant and

<sup>1</sup>Institute of Big Data and Information Technology, Wenzhou University, Wenzhou 325035, China. <sup>2</sup>School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran. <sup>3</sup>Department of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China. <sup>4</sup>Department of Artificial Intelligence, Wenzhou Polytechnic, Wenzhou 325035, China. ✉email: chenhuiling.jlu@gmail.com; guoxiliang2017@gmail.com

invalid features plays a vital role and is especially beneficial in dealing with high-dimensional datasets<sup>13,14</sup>. As the number of features in a dataset increases, model training becomes more challenging. Unnecessary features can significantly lengthen the training time and negatively affect the model's performance. Therefore, FS prior to training can improve the effectiveness of the model. Especially in high-dimensional genetic datasets, FS is an essential step because it provides a valuable way to reduce unnecessary features.

FS is a crucial technique in data processing and machine learning. Its main goal is to select the best feature subset from the original data, thus reducing the dimensionality of the data. Based on different approaches, FS can be categorized into five types: filtering, wrapping, embedding, ensemble, and hybrid<sup>15,16</sup>. Among them, wrapping methods rely on classification algorithms to select a subset of features and can achieve more desirable results.

Current optimization algorithms mainly include the exact solution method and approximate solution method<sup>17</sup>. The exact solution method is usually applied to optimization problems with small problem sizes and relatively simple structures. However, due to the limitation of computational complexity, it may face computational difficulties when dealing with large-scale and complex problems. Approximate solution algorithms can find an approximate optimal solution in a fair time, and this class of methods is more suitable for large-scale and complex problems. Metaheuristic algorithms (MAs) are an important branch of approximate solution methods, and they have also been one of the most active research areas in computer science in the past few years<sup>18</sup>. In recent years, researchers have proposed a large number of MA, including Differential Evolution (DE)<sup>19</sup>, Bat Algorithm (BA)<sup>20</sup>, Grasshopper Optimization Algorithm (GOA)<sup>21</sup>, Colony Predation Algorithm (CPA)<sup>22</sup>, weighted mean of vectors (INFO)<sup>23</sup>, Harris Hawk Algorithm (HHO)<sup>24</sup>, Sine Cosine Algorithm (SCA)<sup>25</sup>, Ant Colony Algorithm (ACO)<sup>26</sup>, Runge Kutta Optimizer (RUN)<sup>27</sup>, Slime Mould Algorithm (SMA)<sup>28</sup>, Gravity Search Algorithm (GSA)<sup>29</sup>, Hunger Games Search (HGS)<sup>30</sup>, Rime Optimization algorithm (RIME)<sup>31</sup>, Parrot Optimizer (PO)<sup>32</sup>, and Liver Cancer Algorithm (LCA)<sup>33</sup>, among others. These methodologies are aimed at solving problems across various fields, such as networking<sup>34,35</sup>, image segmentation<sup>36</sup>, and the Internet of Things<sup>37,38</sup>, among others. As optimization problems become more complex, there is a need for more intelligent algorithms that can use their exploration and exploitation capabilities to deal with problems that are nonlinear and contain multiple locally optimal solutions.

Recently, researchers have proposed several hybrid techniques incorporating different search strategies for feature selection problems, providing suitable references and ideas for this paper. Hussain et al.<sup>39</sup> proposed a hybrid optimization approach integrating the sine-cosine algorithm into HHO while dynamically adjusting the candidate solutions to avoid the solution stagnation problem in HHO. Through testing on the CEC 2017 test set as well as 16 feature selection datasets containing both low and high dimensions, the results demonstrate that the proposed hybrid variant of HHO produces efficient search results without increasing the computational cost. A new variant of SSA for the feature selection problem is introduced by Neggaz et al.<sup>40</sup>. The authors use a sinusoidal mathematical function inspired by the sine-cosine algorithm to update the positions of the followers in the SSA while using a disrupt operator to increase the diversity of the population. Evaluation of 20 datasets containing 4 high dimensions demonstrates that the algorithm performs well in accuracy, sensitivity, specificity, and number of selected features. Building on previous work, our study focuses on high-dimensional gene selection data, so the selected datasets are almost all high-dimensional small-sample gene data.

The investigation-localization-tracking process is often involved in criminal investigations by police officers involved in criminal investigations. Inspired by this process, Chou et al.<sup>41</sup> proposed a Forensic-based Investigation (FBI) algorithm in 2020. FBI has several advantages, such as no predefined parameters, good performance, robustness, and stability. As a result, it has received much attention. It has been successfully used in diverse fields, including resource-constrained scheduling problems<sup>41</sup>, solar cell models<sup>42</sup>, framework hyperparameter optimization<sup>43</sup>, optimal parameters for fuel cell MPPT<sup>44</sup>, and lateral vibration of ground-based systems<sup>45</sup>, and edge smart grids<sup>46</sup>. In this study, an FBI-based algorithm will be used for FS.

However, like other MAs, the original FBI has shortcomings in solving high-dimensional data and real-world problems. Therefore, to alleviate the problems and balance the exploitation and exploration capabilities of the FBI, many improved variants of the FBI have been proposed recently. Kaveh et al.<sup>47</sup> proposed a frequency-constrained dome truss optimization design method based on Enhanced Forensic Investigations (EFBI), which modifies the original FBI to enhance the link between the investigation and tracing teams in the algorithm. EFBI is used in three dome trusses with frequency constraints optimization problems, and comparisons are made with other optimization algorithms to verify the improvement of the improved algorithm. Kuyu et al.<sup>48</sup> focused on the search process of the FBI algorithm, i.e., Step A and Step B stages. The algorithm's behavior is improved by enhancing the diversity of the population by incorporating dyadic-based learning in Stage A and integrating the Cauchy variation mechanism in Step B to improve the ability to jump out of the local optimal solution. Validated on two benchmark test sets and six real-world problems, the dyadic-based learning and the Cauchy variation mechanism positively affect the FBI. Malika et al.<sup>49</sup> suggested a quasi-oppositional forensic investigation (QOFBI)-inspired approach for producing optimal outcomes for DG allocation and sizing that incorporates all the system's operating limitations. To demonstrate the enhanced capabilities of the suggested method, IEEE 33-bus, IEEE 69-bus, and IEEE 85-bus test systems are simulated. Several metrics of performance for power, voltage, and stability have been calculated for various degrees and kinds of DG penetration, and a comparative study finding with the previously recommended strategy are provided and examined. The outcomes illustrate that the approach is more robust and efficient, and it operates superior with less computing cost.

Nguyen et al.<sup>50</sup> devised a framework integrating Building Information Modeling (BIM), Multi-Objective Optimization (MOO), and Multi-Criteria Decision Making (MCDM) to address resource tradeoffs in project scheduling. Initially, BIM construction management software is leveraged to develop a comprehensive 3D model, facilitating the generation of a quantity list delineating the necessary project resources. A novel approach, termed Multi-Objective Forensic Investigation Method (MOFBI), is introduced to achieve the most favorable outcome. Subsequently, an evidence-driven multi-criteria decision analysis technique is applied to ascertain the ideal project execution schedule. The efficacy and potency of this framework are authenticated through

the examination of three distinct project scheduling challenges. Tolba et al.<sup>51</sup> presented a robust, improved forensic-based investigation (mFBI) optimization method for calculating the most efficient location of distributed generators (DGs) in electricity distribution networks (EDNs) to minimize the loss of power, as well as voltage deviations. Furthermore, hierarchical analysis is employed to derive the most relevant weighting factors for the multi-objective function (MOF). The efficacy of the proposed mFBI technique is validated and demonstrated through an investigation into the impact of DG integration on 118 IEEE EDN nodes and real Delta-Egypt EDN nodes. Chou et al.<sup>52</sup> suggested a forensic-based multi-objective investigation method for the multi-objective engineering optimization problem. Within this algorithm, the population undergoes initialization via chaotic mapping. Subsequently, Lévy flights, two elite groups, and a fixed-size file are employed to regulate the activities of investigators and police officers during offender search and frisking procedures. Simultaneously, a control time mechanism is integrated into MOFBI to harmonize exploration and exploitation, thereby attaining a Pareto-optimal solution within the multi-objective search space. Experiments show that MOFBI can approximate the Pareto-optimal frontier more accurately than other algorithms.

Although many advanced and improved FBIs have been proposed, most of the existing improved algorithms still suffer from the question of slow convergence and the greater probability of falling into local optimal when solving some specific cases. In the original FBI algorithm, two phases are included in the process of criminal investigation: one is the investigation phase, and the other is the tracing phase. The two phases perform independent searches in their respective populations, and trapping into the same local optimal is possible. The search agent generated by the SMA can adaptively go beyond the local optimal and better find the optimal solution through the positive and negative feedback mechanism. This article is based on the original FBI version, and the search phase of the slime mould algorithm has been added to assist with the solution. During the proposed algorithm, the slime mould search mechanism is integrated as an independent inspector group that compensates for the shortcomings of the investigation and pursuit groups. The slime mould search mechanism dynamically adjusts the search patterns according to the probability of the suspect being at the location. When the probability is high, the slime mould search mechanism uses an area-restricted search methodology, which focuses on the identified area. If the probability of the suspect being at the location is initially found to be low, the slime mould search mechanism controls the search to jump out of the current area and look for other locations with a high probability of the suspect being there. Thus, this strategy can significantly increase the convergence speed of the algorithm and the capability of skipping local optimal. A new variant of the FBI, called SMA\_FBI, is developed by incorporating the SMA strategy into the original FBI. After that, the binary version of the algorithm, i.e., BSMA\_FBI, is obtained utilizing a conversion function, which is applied to the GS problem with high-dimensional data.

The remainder of this paper is divided into two sections: “Overview (FBI and SMA)” section describes the original FBI and SMA. In “Proposed SMA\_FBI” section, SMA\_FBI and the algorithm’s time complexity are given a detailed description. “Experiments” section gives and analyzes the experimental results. “Discussion” section discusses the experiments as well as the results. “Conclusion and future work” section summarizes the conclusions and gives some future directions.

## Overview (FBI and SMA)

This section provides a detailed description of the FBI and SMA.

### Forensic-based investigation (FBI)

The forensic-based investigation algorithm was inspired by Chou et al. from the investigation-localization-pursuit process of pursuing suspects by police officers involved in criminal investigations. It consists of two phases: the investigation phase (Step A) and the pursuit phase (Step B). The investigation phase is responsible for determining the location interval of the suspect in a general direction, while the pursuit phase requires a detailed search at the suspect location. The search space of the algorithm is defined as all possible suspect locations with the probability of locating the suspect as a metaphor for the objective function. In the search space, the investigator analyzes and evaluates the collected information to determine the identity and location of suspects, based on which the police can carry out an arrest. During this process, investigators and pursuers shift the direction of the search based on the latest evidence, thus requiring them to coordinate closely with each other throughout the process. The pseudo-code of the FBI appears in Algorithm 1. The flowchart for the FBI is pictured in Fig. 1.

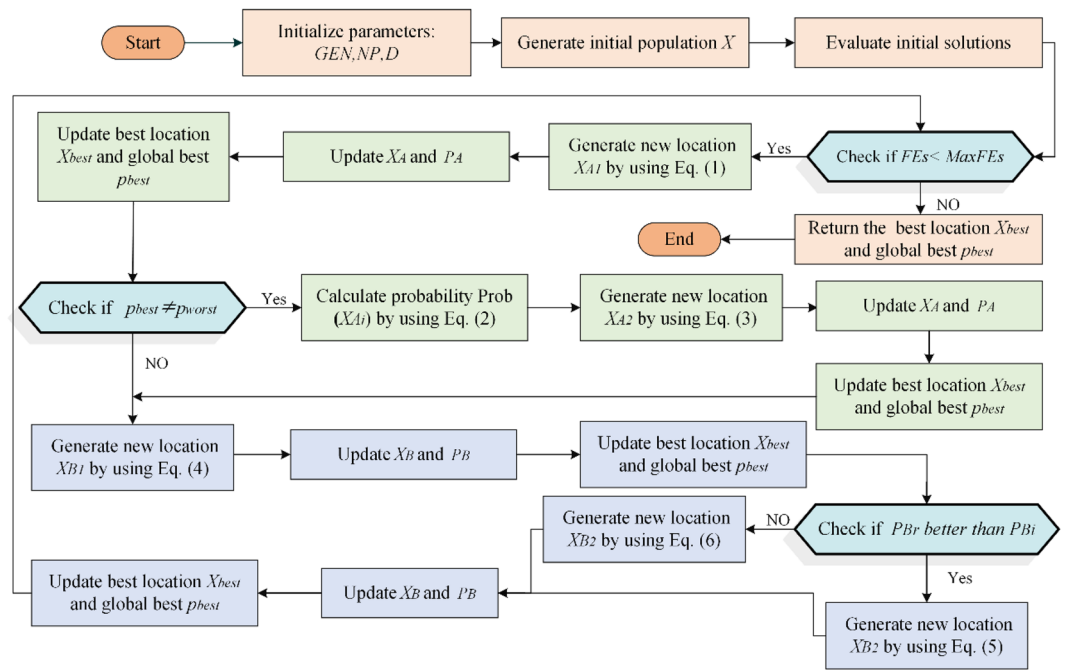
The original FBI can be categorized into several imperative steps:

**Step A1** Interpretation of Discovered Information. During this step, the investigative team analyzes the collected findings and initially identifies possible suspect locations, which can be inferred based on information related to  $X_{A_i}$  and other suspect points. In this work, each individual is influenced by the others. The new suspected location  $X_{A_{1i}}$  is represented in Eq. (1).

$$X_{A_{1i}} = X_{A_{ij}} + (r_1 - 0.5) * 2 * \left( X_{A_{ij}} - \left( X_{A_{kj}} + X_{A_{hj}} \right) / 2 \right) \quad (1)$$

where  $X_{A_i}$  denotes the  $i$  th suspicious point,  $i = 1, 2, \dots, NP$ ;  $NP$  is the overall size of the population.  $j = 1, 2, \dots, D$ ;  $D$  is the number of dimensions.  $r_1$  represents a random number in the range of  $[0, 1]$ ,  $((r_1 - 0.5) * 2$  indicates a random value between  $-1$  and  $1$ .  $k, h, j$  are the three randomly selected suspicious locations,  $\{k, h, j\} \in \{1, 2, \dots, NP\}$ .

**Step A2** Determine the direction of the investigation. To establish the most likely suspect site, the investigator compares the probability of each suspect location to each other.  $p_{A_i}$  indicates the likelihood (objective value) that the suspect is located at position  $X_{A_i}$ , i.e.,  $p_{A_i}$  denotes the objective value for the location  $X_{A_i}$  (i.e.,  $p_{A_i} = f_{objective}(X_{A_i})$ ). The investigator evaluates the likelihood of the new suspect’s position and compares it to the location



**Figure 1.** The flowchart of FBI.

of the current entrance. The site with the higher probability (objective value) of the suspect’s presence will be reserved, while the other location will be discarded. The probability of each location is calculated by Eq. (2).

$$Prob(X_{A_i}) = (p_{worst} - p_{A_i}) / (p_{worst} - p_{best}) \tag{2}$$

where  $p_{worst}$  is the minimum probability of the existence of a suspect,  $p_{best}$  is the maximum probability, and  $X_{best}$  is the optimal location. The update of the search position will be affected by other suspicious positions, the direction of random selection is introduced on the basis of the optimal individual  $X_{best}$ , to increase the diversity of the search area and expand the search space. The position update formula is shown below:

$$X_{A_{2ij}} = X_{best} + X_{A_{dj}} + r_2 * (X_{A_{ej}} - X_{A_{fj}}) \tag{3}$$

where  $X_{best}$  indicates the best position obtained in step A1,  $r_2$  is a random value between 0 and 1; and  $d, e, f$  are three random values indicating the three suspicious positions.

**Step B1** Begin the operation. In this stage, the arresting officer approaches the target location and arrests the suspect based on the best location provided by the investigation team. Each  $B_i$  (pursuing officer) approaches the location with the best likelihood and updates the location if the newly approached location yields a better likelihood than the likelihood of the old location.

$$X_{B_{1ij}} = r_3 * X_{B_{ij}} + r_4 * (X_{best} - X_{B_{ij}}) \tag{4}$$

where  $r_3$  and  $r_4$  express two random values in the range of 0 and 1,  $j = 1, 2, \dots, D$ .

**Step B2** Real-time position updates based on actions. While the tracking team is taking pursuit actions, it reports new suspect points to the headquarters in real-time. The headquarters will update the position and direct the tracking team to approach the suspect point. Each tracking team member works closely and interacts with each other. Agent  $B_i$  approaches the target point and receives influence from team member  $B_r$  at the same time. When the likelihood of  $B_r$  is greater than the likelihood of  $B_i$ , a new suspect location is generated according to Eq. (5); and vice versa according to Eq. (6).

$$X_{B_{2ij}} = X_{B_{rj}} + r_5 * (X_{B_{rj}} - X_{B_{ij}}) + r_6 * (X_{best} - X_{B_{rj}}) \tag{5}$$

$$X_{B_{2ij}} = X_{B_{ij}} + r_7 * (X_{B_{ij}} - X_{B_{rj}}) + r_8 * (X_{best} - X_{B_{ij}}) \tag{6}$$

where  $X_{best}$  denotes the best position provided by step B1,  $r_5, r_6, r_7$  and  $r_8$  denote random numbers between 0 and 1;  $r$  and  $i$  indicate two random agents,  $\{r, i\} \in \{1, 2, \dots, NP\}$ .

**/\*Initialization phase\*/**

Initialize the parameters:  $GEN$ ,  $NP$ ,  $D$ ;

Generate the initial population  $X_i (i = 1, 2, \dots, NP)$ ;

$X_{A_i} = X_{B_i} = X_i$ ;

**/\*Start the cyclical investigation and pursuit processes\*/**

**While**  $g \leq GEN$

**/\*Investigation team - Team A\*/**

**For**  $i = 1: NP$  **/\*Step A1\*/**

**For**  $j = 1: D$

Generate new location  $X_{A1ij}$  by using Eq. (1);

**End for**

Calculate  $p_{A1i}$  (the objective value for location  $X_{A1i}$ );

Update  $X_{A_i}$  and  $p_{A_i}$ ;

**End for**

Update best location  $X_{best} = X_{A1_{best}}$  and global best  $p_{best} = p_{A1_{best}}$ ;

**If**  $p_{best} \neq p_{worst}$  **/\*Step A2\*/**

Calculate probability  $\text{Prob}(X_{A_i})$  by using Eq. (2);

**For**  $i = 1: NP$

**If**  $\text{rand} > \text{Prob}(X_{A_i})$

**For**  $j = 1: D$

**If**  $\text{rand} < \text{rand}$

Generate new location  $X_{A2ij}$  by using Eq. (3);

**End if**

```

        End for
        Calculate  $p_{A2_i}$  (the objective value for location  $X_{A2_i}$ );
        Update  $X_{A_i}$  and  $p_{A_i}$ ;
    End if
End for
Update best location  $X_{best} = X_{A2_{best}}$  and global best  $p_{best} = p_{A2_{best}}$ ;
End if
/*Investigation team - Team B*/
For  $i = 1: NP$  /*Step B1*/
    For  $j = 1: D$ 
        Generate new location  $X_{B1_{ij}}$  by using Eq. (4);
    End for
    Calculate  $p_{B1_i}$ ;
    Update  $X_{B_i}$  and  $p_{B_i}$ ;
End for
Update best location  $X_{best} = X_{B1_{best}}$  and global best  $p_{best} = p_{B1_{best}}$ ;
For  $i = 1: NP$  /*Step B2*/
    Randomly select  $X_{B_r}$ ;
    If  $p_{B_r}$  better than  $p_{B_i}$ 
        For  $j = 1: D$ 
            Generate new location  $X_{B2_{ij}}$  by using Eq. (5);
        End for
    Else
        For  $j = 1: D$ 
            Generate new location  $X_{B2_{ij}}$  by using Eq. (6);
        End for
    End if
    Calculate  $p_{B2_i}$ ;
    Update  $X_{B_i}$  and  $p_{B_i}$ ;
End for
Update best location  $X_{best} = X_{B2_{best}}$  and global best  $p_{best} = p_{B2_{best}}$ ;
End While
/*End the cyclical investigation process*/
Return the best location  $X_{best}$  and global best  $p_{best}$ 

```

**Algorithm 1.** Pseudo-code of FBI.

### Slime mould algorithm (SMA)

Many newly developed algorithms based on their respective properties and search mechanisms can balance exploitation and exploration well. SMA is an effective population-based algorithm proposed by Li et al.<sup>28</sup>. SMA principally emulates the behavioral and morphological changes of slime moulds when they feed. The algorithm uses weights to model the positive and negative feedback slime moulds generate during the foraging process, resulting in three different morphological types. Previous studies in many application scenarios<sup>53–56</sup> have demonstrated the superior performance of SMA in exploration and exploitation.

#### Approaching food

Slime moulds can approach food through odors in the environment. The following equation is used to model this contraction pattern of approaching food:

$$X(t+1) = \begin{cases} X_b(t) + vb(t) \cdot (W \cdot X_A(t) - X_B(t)), & r < p \\ vc(t) \cdot X(t), & r \geq p \end{cases} \quad (7)$$

where  $vb$  is a parameter between  $-a$  and  $a$ , and  $vc$  exhibits a linear decline from one to zero.  $T$  represents the contemporary iterations, and  $X_b$  indicates the position with utmost concentration of odor discovered thus far.  $X$  refers the location of the slime moulds.  $X_A$  and  $X_B$  signify two fortuitously chosen individuals from the slime moulds.  $W$  symbolizes the weights assigned to the slime moulds.  $p$  is parameterized as follows:

$$p = \tanh|S(i) - DF| \tag{8}$$

where  $i \in 1, 2, \dots, n$ ,  $S(i)$  signifies the fitness evaluation of  $X$ , while  $DF$  indicates the most superior fitness achieved across all iterations.  $vb$  is given by the following equation:

$$vb = [-a, a] \tag{9}$$

$$a = \operatorname{arctanh}\left(-\left(\frac{t}{Max_t}\right) + 1\right) \tag{10}$$

$W$  relies on the utilization of the following formula:

$$W(\text{SmellIndex}(i)) = \begin{cases} 1 + r \cdot \log\left(\frac{bF - \text{SmellOrder}(i)}{bF - wF} + 1\right), & \text{condition} \\ 1 - r \cdot \log\left(\frac{bF - \text{SmellOrder}(i)}{bF - wF} + 1\right), & \text{otherwise} \end{cases} \tag{11}$$

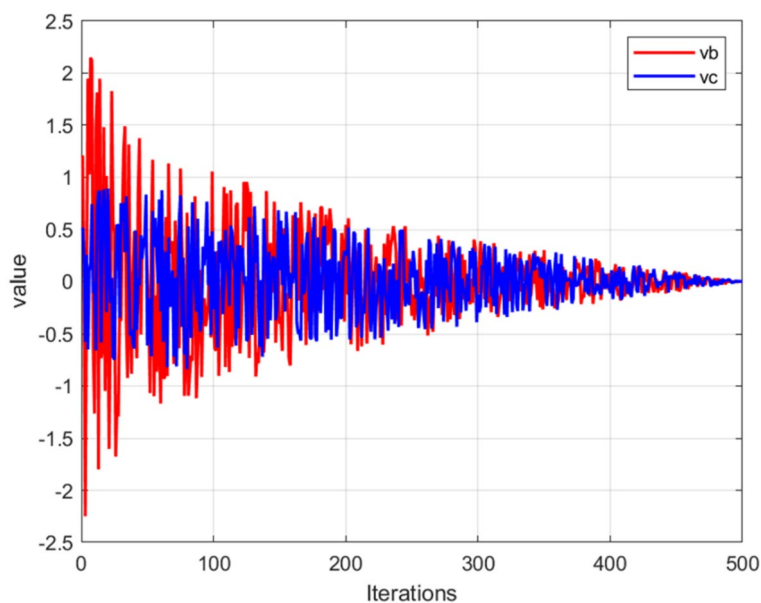
$$[\text{SmellOrder}, \text{SmellIndex}] = \text{sort}(S) \tag{12}$$

where condition refers to when  $S(i)$  is ranked within the superior half of the population,  $r$  signifies a randomly selected value between 0 and 1.  $max\_t$  represents the maximum iterations, and  $bF$  and  $wF$  refer the best fitness and worst fitness, respectively, achieved during the current iteration.  $smellIndex$  indicates the sequence of fitness values for the ranking (which goes up in the minima problem).

*Wrapping food*

This section models the shrinking pattern of the venous tissue structure as the slime mould searches for food. As the concentration of food that the vein is subjected to increases, the intensity of the wave produced by the biological oscillator amplifies, the speed of cytoplasmic flow accelerates, and the thickness of the vein augments. Equation (13) provides a mathematical representation of the positive and negative feedback relationship between the width of the veins of the slime mould and the food concentration, where the parameter  $r$  models the uncertainty in the contraction pattern of the veins. Including a logarithmic function slows the change in frequency so that no drastic changes occur in contraction frequency values. *condition* emulates the fact that the slime moulds dynamically tune their search pattern due to the concentration of food. In conditions of elevated food concentration, the weight of the neighborhood increases; vice versa, the weight of the vicinity decreases, prompting slime moulds to venture into alternative regions for exploration.

$$X(t + 1) = \begin{cases} \text{rand} \cdot (ub - lb) + lb, & \text{rand} < z \\ X_b(t) + vb(t) \cdot (W \cdot X_A(t) - X_B(t)), & r < p \\ vc(t) \cdot X(t), & r \geq p \end{cases} \tag{13}$$



**Figure 2.** The trends of  $vb$  and  $vc$ .

where  $lb$  and  $ub$  denote the upper and lower boundary within a search range.  $rand$  and  $r$  represent random variables encompassing values inclusively between 0 and 1.  $rand$  is used as a key parameter to control whether or not to enter the stochastic update, and  $r$  determines whether or not to entry into the exploration and exploitation phase. Additionally, to adhere to the original text, the parameter  $z$  is specifically assigned the value of 0.03.

**Oscillation**

Slime moulds rely heavily on propagating waves produced by a biological oscillator to manipulate the movement of cytoplasmic flow within the vein to a location that favors food concentration. To emulate the alterations in the pulse width of slime moulds,  $W$ ,  $vb$ ,  $vc$  were used.

$W$  mathematically modeled the rate of oscillations of slime moulds relative to varying food concentrations. Consequently, this facilitated the slime moulds' ability to approach regions of higher food quality swiftly. In conditions of the food concentration was lower at certain locations, the slime moulds approached the food more slowly, thus improving the capability of the slime moulds to select the best food source efficiently.

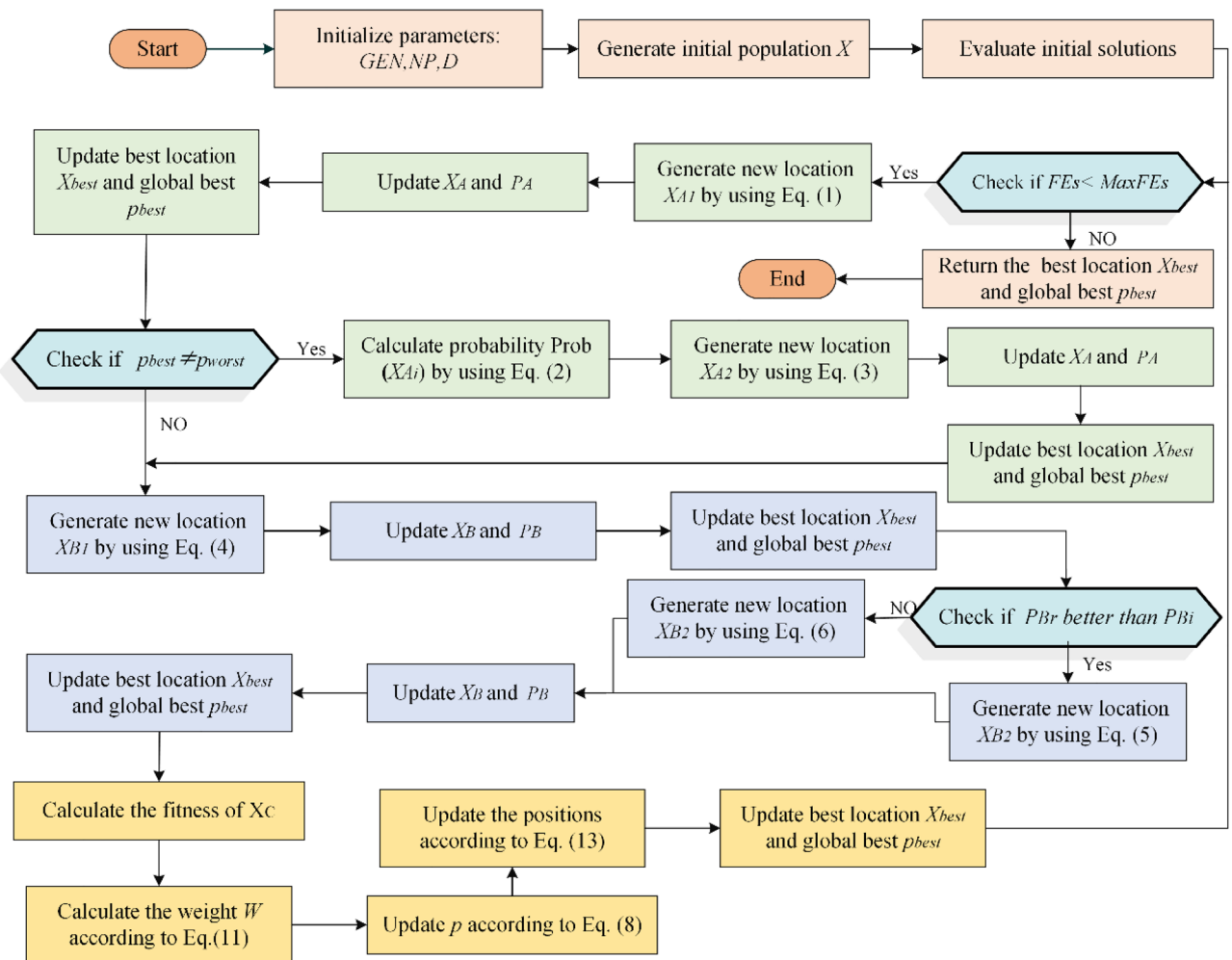
The variable  $vb$  exhibits random oscillations between  $[-a, a]$ , but steadily converges to 0 as the number of iterations increases. Similarly, the variable  $vc$  undergoes oscillations within the interval of  $[-1, 1]$ , and eventually converges to 0. The visual representation of this behavior can be observed in Fig. 2. The synergy between  $vb$  and  $vc$  simulates the selection patterns demonstrated by slime moulds. The slime moulds will explore some areas independently to find better food sources. The slime mould will branch out to search for better food sources instead of concentrating on one food source. This strategy ensures that the slimy bacteria algorithm will not easily fall into a local optimum.

**Proposed SMA\_FBI**

In this section, we provide a detailed description of the improved FBI and the time complexity of the algorithm in conjunction with the previous chapter.

**Enhanced FBI with the SMA (SMA\_FBI)**

Except for the population size and the evaluation of stopping conditions, the base FBI does not rely on predefined parameters, so the parameters do not affect the algorithm's behavior. At the beginning of the FBI algorithm,



**Figure 3.** The flowchart of SMA\_FBI.



the population was replicated into two parts, and each part independently searched for the optimal solution, interacting information through the current optimal position and the best fitness value. However, this led to barriers to communication within the algorithm, and the two parts did not exchange other information. Kaveh et al.<sup>47</sup> improved FBI by enhancing the exchange of information between the two parts. Instead of attempting to enhance communication, we added an inspector group to the investigating and searching groups, in which we introduced the search mechanism from the SMA.

When searching for food, the slime moulds can adaptively tune their search pattern due to varying levels of food concentration. In conditions of elevated food concentration, the slime moulds concentrate their search on the currently recognized food sources; if they find a low food concentration, the slime moulds depart from the food source and search for other food concentrations. Throughout this process, the front end of the slime mould extends out and is able to build a network of veins in the search space, and the quality of the food source affects the propensity of the slime mould to search.

Using this property of slime moulds in FBI, an inspector group can be reconstructed in addition to the original investigation group and pursuit group, which can make up for the deficiencies of the first two stages and enhance the capacity of the FBI. The flowchart of SMA\_FBI is displayed in Fig. 3. The pseudo code for the SMA\_FBI is provided by Algorithm 2.

```

/*Initialization phase*/
Initialize the parameters:  $GEN$ ,  $NP$ ,  $D$ ;
Generate the initial population  $X_i (i = 1, 2, \dots, NP)$ ;
 $X_{A_i} = X_{B_i} = X_{C_i} = X_i$ ;
Initialize control parameters  $a, z$ ;
/*Start the cyclical investigation, pursuit and inspector processes*/
While  $g \leq GEN$ 
  /*Investigation team - Team A*/
  For  $i = 1: NP$  /*Step A1*/
    For  $j = 1: D$ 
      Generate new location  $X_{A1ij}$  by using Eq. (1);
    End for
    Calculate  $p_{A1i}$  (the objective value for location  $X_{A1i}$ );
    Update  $X_{A_i}$  and  $p_{A_i}$ ;
  End for
  Update best location  $X_{best} = X_{A1best}$  and global best  $p_{best} = p_{A1best}$ ;
  If  $p_{best} \neq p_{worst}$  /*Step A2*/
    Calculate probability Prob ( $X_{A_i}$ ) by using Eq. (2);
    For  $i = 1: NP$ 
      If  $rand > Prob (X_{A_i})$ 
        For  $j = 1: D$ 
          If  $rand < rand$ 
            Generate new location  $X_{A2ij}$  by using Eq. (3);
          End if
        End for
        Calculate  $p_{A2i}$  (the objective value for location  $X_{A2i}$ );
        Update  $X_{A_i}$  and  $p_{A_i}$ ;
      End if
    End for
    Update best location  $X_{best} = X_{A2best}$  and global best  $p_{best} = p_{A2best}$ ;
  End if
  /*Pursuit team - Team B*/
  For  $i = 1: NP$  /*Step B1*/

```

```

For  $j = 1:D$ 
    Generate new location  $X_{B1ij}$  by using Eq. (4);
End for
    Calculate  $p_{B1i}$ ;
    Update  $X_{B_i}$  and  $p_{B_i}$ ;
End for
    Update best location  $X_{best} = X_{B1best}$  and global best  $p_{best} = p_{B1best}$ ;
For  $i = 1:NP$  /*Step B2*/
    Randomly select  $X_{B_r}$ ;
    If  $p_{B_r}$  better than  $p_{B_i}$ 
        For  $j = 1:D$ 
            Generate new location  $X_{B2ij}$  by using Eq. (5);
        End for
    Else
        For  $j = 1:D$ 
            Generate new location  $X_{B2ij}$  by using Eq. (6);
        End for
    End if
    Calculate  $p_{B2i}$ ;
    Update  $X_{B_i}$  and  $p_{B_i}$ ;
End for
    Update best location  $X_{best} = X_{B2best}$  and global best  $p_{best} = p_{B2best}$ ;
/* Inspector team - Team C*/
    Calculate the fitness of  $X_{C_i}$ ;
    Update  $p_{C_{best}}$ ,  $X_{best}$ ;
    Calculate the weight  $W$  according to Eq. (11);
For  $i = 1:NP$ 
    Update  $p$  according to Eq. (8);
    Update  $vb, vc$  respectively;
    Update the positions according to Eq. (13);
End for
    Update best location  $X_{best} = X_{C_{best}}$  and global best  $p_{best} = p_{C_{best}}$ ;
End While
/*End the cyclical investigation process*/
    Return the best location  $X_{best}$  and global best  $p_{best}$ ;

```

**Algorithm 2.** Pseudo-code of SMA\_FBI.

### Computational complexity analysis

The time complexity of SMA\_FBI is primarily associated with the dimension ( $D$ ), the number of suspect locations ( $NP$ ), and the number of evaluations ( $GEN$ ). Overall, the time complexity is calculated from 4 aspects: initialization, fitness assessment, location update, and slime mould update strategy. For the initialization of the suspect location, the time complexity is  $O(D * NP)$ , the time cost of the adaptation assessment is  $O(NP)$ , the location updating part includes the investigation phase as well as the pursuit phase, and the time complexity of each phase is  $2 * O(NP * D)$ , and the time consumption of the mucus updating strategy is  $O(NP * (1 + \log N + D))$ . Considering the total number of evaluations  $GEN$ , the total time complexity of SMA\_FBI is  $O(D * NP + NP + GEN * (4 * NP * D + NP * (1 + \log N + D)))$ .

### Experiments

In order to assess the efficacy of the proposed algorithm SMA\_FBI, a substantial quantity of experiments is undertaken in this section. Firstly, SMA\_FBI is compared with 10 other original and 10 improved algorithms on 30 benchmark functions in CEC2017<sup>57</sup>. Benchmark datasets serve as widely acknowledged instruments for assessing the performance of various technologies against uniform criteria<sup>58,59</sup>. These datasets facilitate the evaluation of different technological dimensions, determining which technology excels over others across multiple domains<sup>60,61</sup>. Secondly, the algorithms are tested on different number of evaluations as well as different population sizes while controlling for other variables, and the complexity of SMA\_FBI is also investigated and explained. Finally, the effectiveness of SMA\_FBI in practical applications is tested on the GS dataset.

To emphasize the impartiality of the experiments, all comparison algorithms undergo testing within the same hardware environment. Within the continuous optimization experiments, metaheuristic algorithm parameters are configured with a population size set at 30, and a maximum of 300,000 assessments. At the same time, to mitigate the influence of randomness on the experiments, all the algorithms are repeated on the test function for 30 times. Based on the experimental data, the capability of the comparison algorithms was evaluated using the mean (*avg.*) and standard deviation (*std.*) of the optimal function values. The best results in the data are shown in bold. The nonparametric statistical test Wilcoxon signed-rank test<sup>62</sup> was utilized to ascertain whether SMA\_FBI exhibits statistical superiority over other algorithms, with a significance level set at 0.05. The symbols "+/=-" denote the proposed algorithm's superiority, equality, or inferiority to the other algorithms. Consistency was assessed using the Friedman test<sup>63</sup> to rank the mean experimental results and list the average ranked value (ARV). Accurate validation of any proposed model or algorithm must be done based on known parameters and settings. In the next experiments, all the parameter settings of the compared algorithms will be listed separately.

The experimental results of SMA\_FBI in terms of function optimization are shown and analyzed in this subsection. Thirty functions of CEC2017 are selected as test functions, and the specifics of these functions can be found in Appendix A.1. Within this selection, F1-F3 represent unimodal functions, F4-F10 are associated with multimodal functions, F11-F20 pertain to hybrid functions, and F21-F30 are linked to composite functions. The unimodal function contains a single global optimal solution, which serves as a means to assess the algorithm's exploitation capabilities. Meanwhile, the multimodal function has multiple locally optimal solutions and is employed to evaluate the algorithm's capacity for global exploration. The hybrid and composite functions gauge the algorithm's equilibrium between exploitation and exploration.

All assessments were conducted on a Windows Server 2012 R2 datacenter operating system equipped with 128 GB of memory, utilizing an Intel (R) Xeon (R) E5-2650 v4 (2.20 GHz) CPU, within a MATLAB R2014b programming environment.

In "Parameter sensitivity analysis" section, experiments are analyzed for different evaluation numbers as well as population size. "Comparison with conventional algorithms" and "Comparison with state of the art algorithms" sections entail comparisons of SMA\_FBI with 10 original algorithms and 10 enhanced algorithms, respectively, aimed at substantiating SMA\_FBI's performance in addressing exploration and exploitation in the context of CEC 2017. Furthermore, in "Experiments on real world optimization of GS" section, SMA\_FBI is used to handle the GS problem for a dataset from the UCI database.

### Parameter sensitivity analysis

In order to enhance the analysis of the algorithm's parameter sensitivity, the impact of population size and the number of evaluations on the algorithm is examined by manipulating individual parameters while holding other variables constant.

During this phase of the experiment, in order to reflect the comprehensiveness of the experiment, four different functions, namely, unimodal function F3, multimodal function F7, hybrid function F14, and composite function F29, were selected for verification. The population sizes were set to 10, 30, 60, 100 and 200 to study the influence of population size on the algorithm's performance. Based on the findings presented in Appendix A.2, the optimization effectiveness of SMA\_FBI generally surpasses that of FBI. Moreover, when the population size is 30, the optimization effect of SMA\_FBI reaches the optimal value, while the optimization effect of SMA\_FBI is relatively poor when the population size deviates from 30.

Another pivotal factor influencing the experimental results is the number of algorithm evaluations. We selected five evaluation times, 50,000, 100,000, 150,000, 200,000, and 300,000, to investigate the effect of evaluation times on the property of SMA\_FBI. Similarly, the validation is carried out in four functions: unimodal function F3, multimodal function F7, hybrid function F14, and composite function F29. From Appendix A.3, it can be seen that SMA\_FBI has achieved the optimal value on the composite function before 50,000 evaluations. For the multimodal and hybrid functions, the optimal value is already close to the optimal value at 200,000 evaluations, but only at 300,000 evaluations can all the functions take the optimal value. In conclusion, we chose 300,000 evaluations.

### Comparison with conventional algorithms

Within this section, a comparison is conducted between SMA\_FBI and ten different traditional algorithms, including the original FBI<sup>41</sup> as well as SMA<sup>28</sup>, MFO<sup>64</sup>, BA<sup>20</sup>, MVO<sup>65</sup>, GSA<sup>29</sup>, SCA<sup>25</sup>, FA<sup>66</sup>, DE<sup>19</sup>, and PSO<sup>67</sup>. These ten algorithms used for comparison with the proposed algorithm include the original FBI as well as the SMA algorithm (for verifying the effectiveness of the improved algorithm), and also include the classical algorithms DE, PSO and different kinds of algorithms such as MVO, GSA, etc. (for verifying the superiority of the proposed algorithm). Table 1 showcases the precise parameter configurations employed by the comparison algorithms, with the specific parameter values remaining consistent with those used in the paper in which these algorithms were first presented. Appendix A.4, Appendix A.5 lists the detailed experimental data for the above algorithms on the F1-F30 test functions, and the algorithm convergence curves are shown in Fig. 4.

Appendix A.4, Appendix A.5 results show that SMA\_FBI clearly outperforms the original algorithms, including FBI, SMA, and DE. The detailed data in Appendix A.4 shows that SMA\_FBI performs best on 14 out of 30 functions and achieves suboptimal results on 8 functions, F1-2, F8, F11-12, F16-17, and F19. In the table, the optimal values are bolded. On the unimodal functions, SMA\_FBI does not obtain the optimal values, but it also obtains the suboptimal values or better values, which shows the strong exploitation ability of SMA\_FBI. Meanwhile, SMA\_FBI obtains optimal values on most multimodal functions, which shows that SMA\_FBI has strong global exploration ability. SMA\_FBI is slightly larger than DE on F11, slightly larger than FBI on F12, slightly larger than DE on F16, and slightly larger than FBI on F17 and F19. Moreover, SMA\_FBI obtains the optimal values on all the composite functions. This is due to the inclusion of the slime mould search mechanism, which improves the performance of FBI and ensures that the algorithm balances between exploration and exploitation within the search space.

The results of the Wilcoxon signed-rank test can be found in Appendix A.5. Observing the  $p$ -value from Appendix A.5 reveals that SMA\_FBI demonstrates a significant improvement compared to the other MAs, with its  $p$ -value consistently lower than 0.05 for all algorithms except FBI and SMA. In addition, it can be seen that SMA\_FBI also has some improvement effects for the FBI and SMA.

Based on the convergence curves of the 11 algorithms on 9 functions demonstrated in Fig. 4, we can observe that SMA\_FBI achieves a high convergence rate on most of the functions, and in F1 as well as F30, the optimal solution is achieved although at a relatively gradual pace. This finding substantiates the claim that the introduction of the slime mould search mechanism, the added inspector group in addition to the original two search groups, enhances the performance of the FBI.

In summary, SMA\_FBI demonstrates good overall superiority as well as robustness compared to other excellent original algorithms. Its comprehensive superiority and reliability are evident. Integrating the slime mould mechanism enhances the exploration and exploitation of the original FBI algorithm, leading to higher-quality solutions.

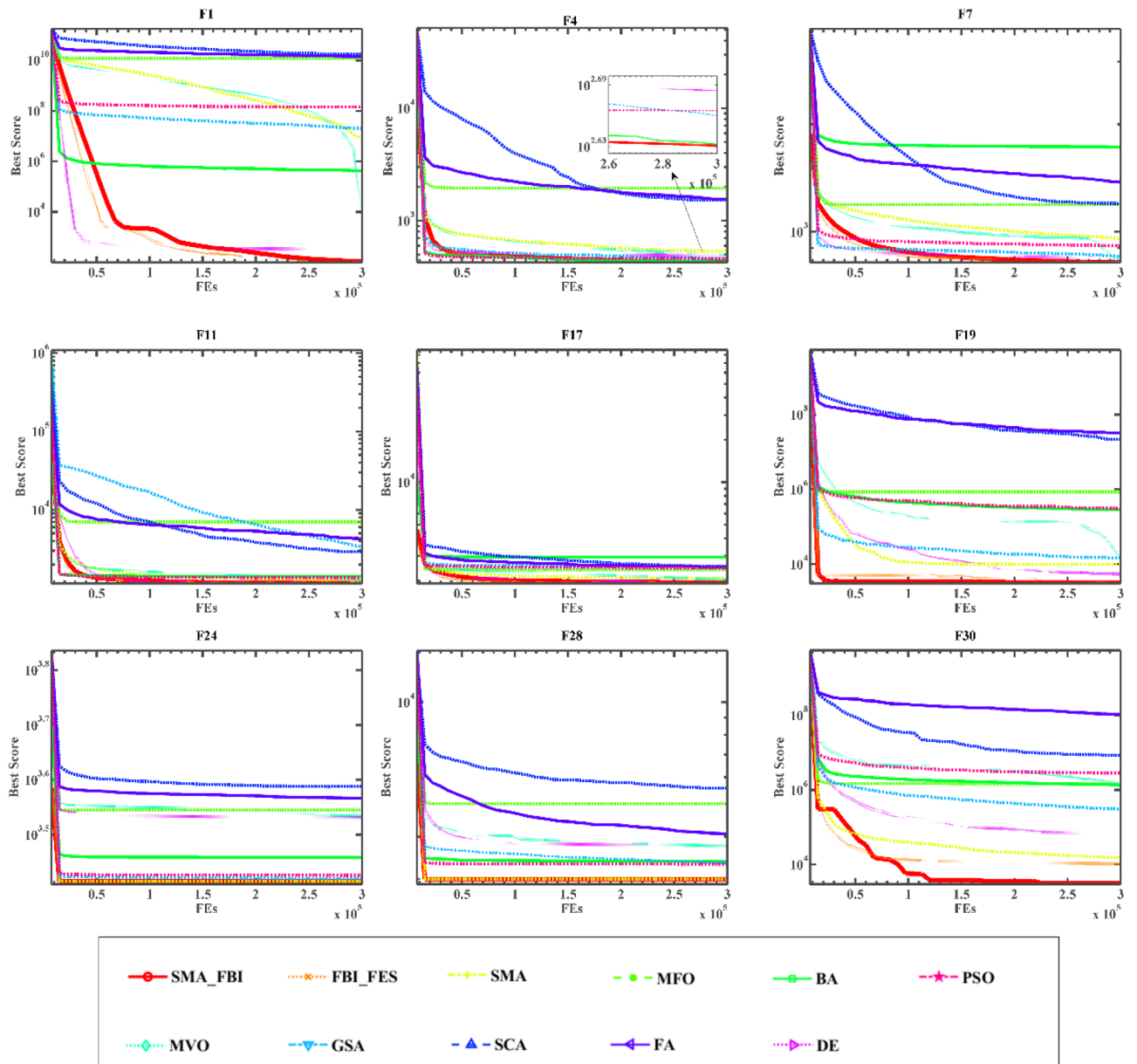
### Comparison with state-of-the-art algorithms

Within this experiment phase, the same CEC2017 benchmark function test set has been chosen to evaluate the capability of SMA\_FBI in correlation to 10 state-of-the-art algorithms, namely EPSDE<sup>68</sup>, ALCPPO<sup>69</sup>, BMWOA<sup>70</sup>, CLPSO<sup>71</sup>, IGWO<sup>72</sup>, CESCAs<sup>73</sup>, RDWOA<sup>74</sup>, LSHADE<sup>75</sup>, CBA<sup>76</sup>, and DECLS<sup>77</sup>. These 10 algorithms contain improved versions of various algorithms, especially of DE, PSO. EPSDE and LSHADE are two champion algorithms that have performed well in the field of evolutionary algorithms, and the superior performance of the proposed algorithm can be verified by comparing it with ten algorithms including these two. Table 2 shows the detailed parameter settings of the algorithms mentioned above. Appendix A.6 shows the comparison outcomes between SMA\_FBI and the above advanced algorithms.

As can be noted from the summed rankings in Appendix A.6, SMA\_FBI is still number one, even in the face of competition from the most highly acclaimed algorithms. In some of the previous functions of CEC2017, SMA\_FBI did not achieve the best result compared to the champion algorithm, but it took the better solution. Furthermore, SMA\_FBI achieves the optimal solution on most of the composite functions, i.e., F23, F25-30, and the sub-optimal solution on F24, and the std is 0 on all these functions, which indicates that SMA\_FBI is more stable as well as robust on the composite functions. It shows that introducing the search mechanism of the SMA makes the algorithm more balanced between exploitation and exploration.

Algorithms	Parameters and values
SMA	$z = 0.03$
MFO	$b = 1; t = [-1, 1]; a \in [-2, -1]$
BA	$a = 0.5; r = 0.5$
MVO	$WEO\_Max = 1; WEP\_Min = 0.2$
GSA	$G_0 = 100; \alpha = 20; final\_per = 2$
SCA	$a = 2$
FA	$\alpha = 0.5; \beta = 0.2; \gamma = 1$
DE	$\beta_{min} = 0.2; \beta_{max} = 0.8; pCR = 0.2$
PSO	$c1 = 2; c2 = 2; vMax = 6$

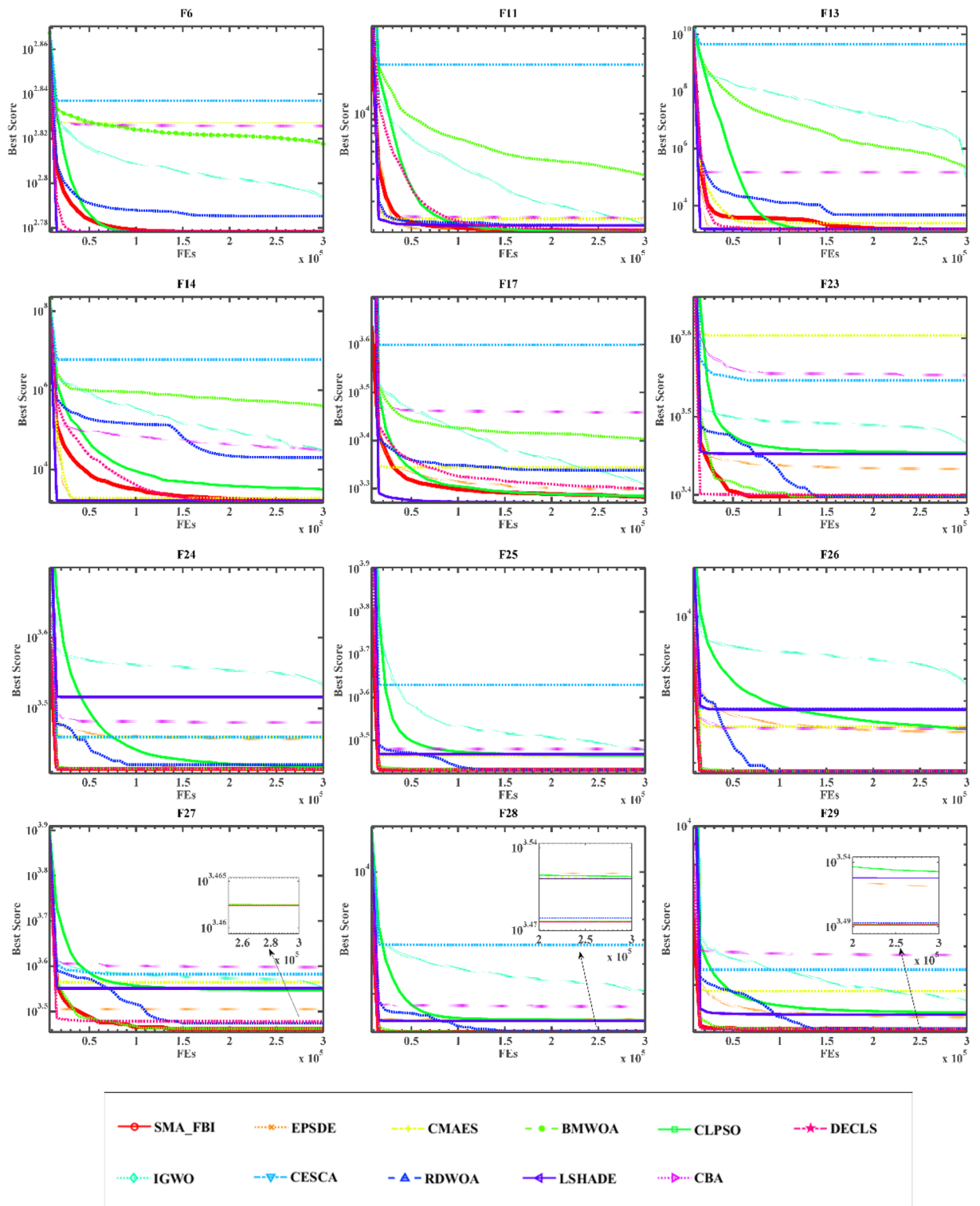
**Table 1.** The specific configuration of parameters.



**Figure 4.** Convergence curves of SMA\_FBI and ten conventional algorithms on nine functions.

Algorithms	Parameters and values
EPSDE	$FF = [0.4, 0.9]$ ; $CR = [0.1, 0.9]$
ALCPSO	$w = 0.4$ ; $c1 = c2$ ; $lifespan = 60$ ; $T = 2$ ; $pro = 1/n$
BMWOA	$a1 = [0, 2]$ ; $a2 = [-2, -1]$ ; $b = 1$
CLPSO	$w \in [0.9, 2]$ ; $c = 1.496$ ; $m = 5$
IGWO	$a \in [0, 2]$
CESCA	$a = 2$
RDWOA	$a1 = [0, 2]$ ; $a2 = [-2, -1]$ ; $b = 1$ ; $s = 0$
LSHADE	$p_{best\_rate} = 0.11$ ; $arc_{rate} = 1.4$ ; $L_{Rate} = 0.8$ ; $First_{class\_percentage} = 0.5$
CBA	$Q_{min} = 0$ ; $Q_{max} = 2$
DECLS	$\beta_{min} = 0.2$ ; $\beta_{max} = 0.8$ ; $pCR = 0.2$ ; $m = 1500$

**Table 2.** The detailed parameter settings.



**Figure 5.** Convergence curves of SMA\_FBI and ten state-of-the-art algorithms on twelve functions.

The results of the Wilcoxon signed-rank test, comparing SMA\_FBI with other state-of-the-art algorithms, are depicted in Appendix A.7. From the table in Appendix, it can be observed that in the experiments of SMA\_FBI with BMOA, IGWO, CESCO, CBA, the *p*-value is much less than 0.05, which proves that SMA\_FBI outperforms these algorithms. Meanwhile, compared with other algorithms, most are also less than 0.05, which shows that SMA\_FBI has apparent advantages over them.

As shown in Fig. 5, SMA\_FBI also shows competitive performance compared to the state-of-the-art and improved algorithms, proving that SMA\_FBI is more competitive. The comparison algorithms also contain

some improvements of DE and PSO algorithms, which further verifies that the introduction of the slime mould mechanism serves as a significant enhancement to the FBI.

In summary, in the face of competition from the challenging state-of-the-art algorithms, the optimization ability of SMA\_FBI is reflected in the overall optimization performance in different types of functions, especially in composite and hybrid functions. The slime mould search mechanism, as the third search scheme in the improved algorithm, enhances the algorithm's exploration and search capability as a whole.

### Experiments on real-world optimization of GS

In this section, we employ the proposed algorithm SMA\_FBI to address the GS problem and showcase the improved algorithm's effectiveness. Whereas the GS problem is a binary optimization task, we adapt the continuous SMA\_FBI into a discrete variant, i.e., BSMA\_FBI, to solve the high-dimensional GS problem.

#### Basic information

The GS problem requires selecting a set of most representative subsets from a collection of features for the purpose of dimensionality reduction of a dataset. GS can effectively reduce the computational cost of data, so many domains with large datasets wish to downsize application data.

In SMA\_FBI based GS algorithm,  $x = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$  represents a set of features, if  $x_{i,1} = 1$ , it implies that the  $i$ th feature is selected; otherwise, the feature is not selected. GS represents a discrete optimization problem; therefore, converting the SMA\_FBI algorithm to a binary version is necessary. We utilize a transfer function to convert continuous SMA\_FBI to binary SMA\_FBI (BSMA\_FBI). The machine learning algorithm is employed in a classification capacity, and its classification accuracy is utilized to evaluate the ability of BSMA\_FBI to screen important features in the dataset. In addition, during the evaluation process, cross-validation was employed to assess the optimum subset of features used for classification to avoid the impact of random elements on the experiment.

#### Fitness function and implementation of experiments

In previous work on continuous optimization, the proposed SMA\_FBI searches for optimal solutions in a continuous search space. Whereas the GS problem is a binary problem, such a problem requires that the solution must be binary, i.e., it can only take either 0 or 1. However, many optimization algorithms are inherently designed for continuous spaces. Therefore, we need a way to convert the outputs of these continuous optimizers to binary values to satisfy the requirements of the problem. The transfer function (or threshold function) is the key to this conversion. The basic idea is to set a threshold value for the output of the continuous optimizer, and then convert the output to 0 or 1 according to this threshold value, with 1 indicating selected and 0 indicating unselected. By adjusting the threshold, we can control the stringency of the selected genes. A higher threshold will result in fewer genes being selected, while a lower threshold may result in more genes being selected. Here we choose a threshold of 0.5 as shown below:

$$X_i^j = \begin{cases} 0 & \text{rand} \leq 0.5 \\ 1 & \text{rand} > 0.5 \end{cases} \quad (14)$$

$X_i^j$  is the  $i$ th value of the current search entity in the  $j$ th dimension within the discrete space.

The transfer function is a proper translator that converts a continuous optimization algorithm into a discrete variant of the algorithm without altering the structure of the algorithm, which is convenient and efficient. Within this paper,  $V$ -type transfer function is employed, and its transfer function is shown below:

$$T(x_i^j(t)) = \left| \tanh x_i^j(t) \right| \quad (15)$$

GS is a process of obtaining the lowest classification error rate employing the least subset of features, which needs to be achieved simultaneously. Evidently, the GS problem presents itself as a multi-objective optimization challenge, and to satisfy each objective, a fitness function can be designed using the classification error rate and the number of selected features to evaluate the chosen feature subset. The specific form of the fitness function is shown below:

$$\text{fitness} = a \cdot \text{error} + b \cdot \frac{l}{d} \quad (16)$$

where *error* represents the classification error rate computed by the K-Nearest Neighbor (KNN)<sup>78</sup> classifier,  $l$  signifies the size of the selected feature subset, and  $d$  is the total number of features in the dataset. Meanwhile,  $a$  and  $b$  serve as two weighting factors indicating the significance of the classification error and the subset length, respectively, to the GS problem. Our study asserts that the classification error rate deserves more attention than the feature subset length. Thus, we assign  $a$  to be 0.95 and  $b$  to be  $1 - a$ , i.e., 0.05. Each feature subset is evaluated based on fitness, with smaller fitness values indicating superior feature subsets.

#### Experimental results and analysis of FS

The SMA\_FBI based GS method, which we refer to as BSMA\_FBI, will be in the face of competition from several state-of-the-art GS methods on 18 datasets, including bGWO<sup>79</sup>, BBA<sup>80</sup>, BGSA<sup>81</sup>, BPSO<sup>82</sup>, bALO<sup>83</sup>, BSSA<sup>84</sup>, bHHO<sup>85</sup>, and the original GS algorithm for the FBI, BFBI. These algorithms used for comparison are more classical algorithms and are commonly used in comparison experiments. They include many different kinds of

Algorithms	Parameters and values
bGWO	$a = [0, 2]$
BBA	$a = 0.5; r = 0.5$
BGSA	$wMax = 20; wMin = 1e - 10$
BPSO	$Max = 0.9; Min = 0.4$
BSSA	$a = 2$
bHHO	$a = [0, 2]$

**Table 3.** Parameter settings for the classifiers.

Datasets	Samples	Features	Categories	Datasets	Samples	Features	Categories
Parkinson	195	23	2	Leukemia1	72	5328	5
penglungEW	102	10,510	2	Leukemia2	72	11,226	3
clean2	6598	167	2	Lung_Cancer	203	12,601	3
Colon	62	2000	2	Lungcancer_3class	32	57	3
Leukemia	72	7131	2	Prostate_Tumor	102	10,509	2
Brain_Tumor1	90	5920	5	SRBCT	83	2309	4
Brain_Tumor2	50	10,368	4	Tumors_9	60	5726	9
CNS	60	7130	2	Tumors_11	174	12,534	11
DLBCL	77	5470	2	Tumors_14	308	15,010	26

**Table 4.** Characteristics of gene expression datasets.

algorithms such as nature-inspired algorithms, algorithms inspired by physical phenomena, and so on. Table 3 lists the detailed parameters of these classifiers. GS based on the SMA\_FBI algorithm is performed on each dataset and is run  $N$  times, and tenfold cross-validations are performed each time. The data samples are partitioned into training, validation, and test sets in the cross-validation procedure according to a certain ratio. In this paper, the KNN classifier is used for classification. The classifier initially undergoes training and classification on all the data within the training set, subsequently assessing and validating against the samples in the validation set, and ultimately applies the chosen features to the test data to ascertain computational accuracy.

Table 4 lists 18 detailed features from the UCI dataset, including the number of instances, features, and categories. As can be observed from the table, these datasets have 32–6598 samples, 23–15,010 features, and 2–26 classes. These datasets essentially represent different types of data, containing both small high-dimensional samples and large low-dimensional samples, which challenges the performance of the algorithm.

Appendix A.8–Appendix A.11 reflects the statistical findings of the means by the number of features selected, error rates, fitness values, and computation time. The bolded values represent the most favorable outcomes for the present comparison results. Examination of Appendix A.8 distinctly illustrates that the proposed BSMA\_FBI selects the fewest features across nearly all datasets and achieves the second least number of features on the Parkinson, Lungcancer\_3class dataset. By comparing the data of BSMA\_FBI with BFBI, we can also find that our improvement of FBI is very effective, and our proposed algorithm selects fewer features and fetches better results than the original algorithm. The ARV metric shows the ranking results of various algorithms on multiple datasets, and there is no doubt that BSMA\_FBI is ranked first. This shows that BSMA\_FBI is competitive in selecting the least features.

According to the ARV, comparison results in Appendix A.9, as a whole, BSMA\_FBI has not achieved the optimal results, but it has also achieved a suboptimal ranking, with an average error that is only a little bit higher than that of bGWO, and the average error value of BSMA\_FBI is noteworthy lower when compared with that of BFBI. The proposed algorithm achieves the least average error on more than half of the datasets and has the smallest standard deviation, even though many of them are 0. This indicates that the proposed algorithm is very stable, which also proves the algorithm's superior behavior. Of course, we can also see that BSMA\_FBI achieves relatively poor results on some algorithms, especially Tumors\_9, Tumors\_11, and Tumors\_14, and we speculate that it may be the fact that these three datasets contain too many categories, which leads to the algorithm's general effect.

Appendix A.10 demonstrates the fitness values for the algorithm comparison, i.e., the weighted results of the error rate versus the number of features, from which it is evident that the data is mainly in line with the trend in Appendix A.9, although BSMA\_FBI achieves the best results due to the addition of the number of features as a factor. The ARV results show that BSMA\_FBI outperforms bGWO, and both significantly outperform the other optimizers, with BFBI having the worst results. This proves that the entry of the slime mould algorithm as a mechanism that improves the effectiveness of the original algorithm in searching for suitable features in the feature space has a positive impact.



	<b>bGWO</b>	<b>BBA</b>	<b>BGSA</b>	<b>BPSO</b>	<b>bALO</b>	<b>BSSA</b>	<b>bHHO</b>	<b>BFBI</b>
Brain_Tumor1	1.00E+00	1.56E-02	4.38E-01	1.25E-01	1.25E-01	1.25E-01	2.50E-01	9.38E-02
Brain_Tumor2	5.00E-01	6.25E-02	2.50E-01	6.25E-02	1.25E-01	1.25E-01	2.50E-01	1.56E-02
clean2	1.95E-03	7.81E-03	1.95E-03	1.95E-02	1.95E-03	4.26E-01	4.75E-01	4.77E-01
CNS	1.00E+00	3.91E-03	2.50E-01	6.25E-02	3.13E-02	3.13E-02	6.25E-02	1.56E-02
Colon	3.75E-01	1.95E-03	3.13E-02	1.56E-02	7.81E-03	4.69E-02	1.56E-02	3.91E-02
DLBCL	1.00E+00	1.25E-01	1.00E+00	1.00E+00	1.00E+00	5.00E-01	1.00E+00	1.00E+00
Leukemia	1.00E+00	6.25E-02	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	5.00E-01
Leukemia1	1.00E+00	3.13E-02	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
Leukemia2	1.00E+00	1.25E-01	5.00E-01	5.00E-01	1.00E+00	1.00E+00	1.00E+00	5.00E-01
Lung_Cancer	6.25E-01	3.91E-03	5.31E-01	3.28E-01	7.50E-01	3.13E-01	1.25E-01	1.56E-02
lungcancer_3class	1.00E+00	1.56E-02	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
Parkinson	2.03E-01	1.56E-02	7.42E-01	6.25E-02	3.13E-02	1.72E-01	3.59E-01	6.72E-01
penglungEW	1.00E+00	3.13E-02	5.00E-01	1.00E+00	5.00E-01	5.00E-01	1.00E+00	3.13E-02
Prostate_Tumor	1.00E+00	1.95E-03	5.00E-01	1.25E-01	2.50E-01	2.50E-01	5.00E-01	3.13E-02
SRBCT	1.00E+00	3.13E-02	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
Tumors_11	1.41E-01	2.54E-02	5.47E-01	2.19E-01	8.44E-01	1.00E+00	7.62E-01	3.20E-01
Tumors_14	1.95E-03	1.31E-01	9.77E-03	7.70E-01	1.05E-01	9.22E-01	4.96E-01	1.05E-01
Tumors_9	1.00E+00	7.81E-03	1.88E-01	7.50E-01	5.00E-01	3.75E-01	5.00E-01	3.13E-02

**Table 5.** The  $p$ -value of the Wilcoxon test between the BSMA\_FBI and alternative GS optimizers on average error rate.

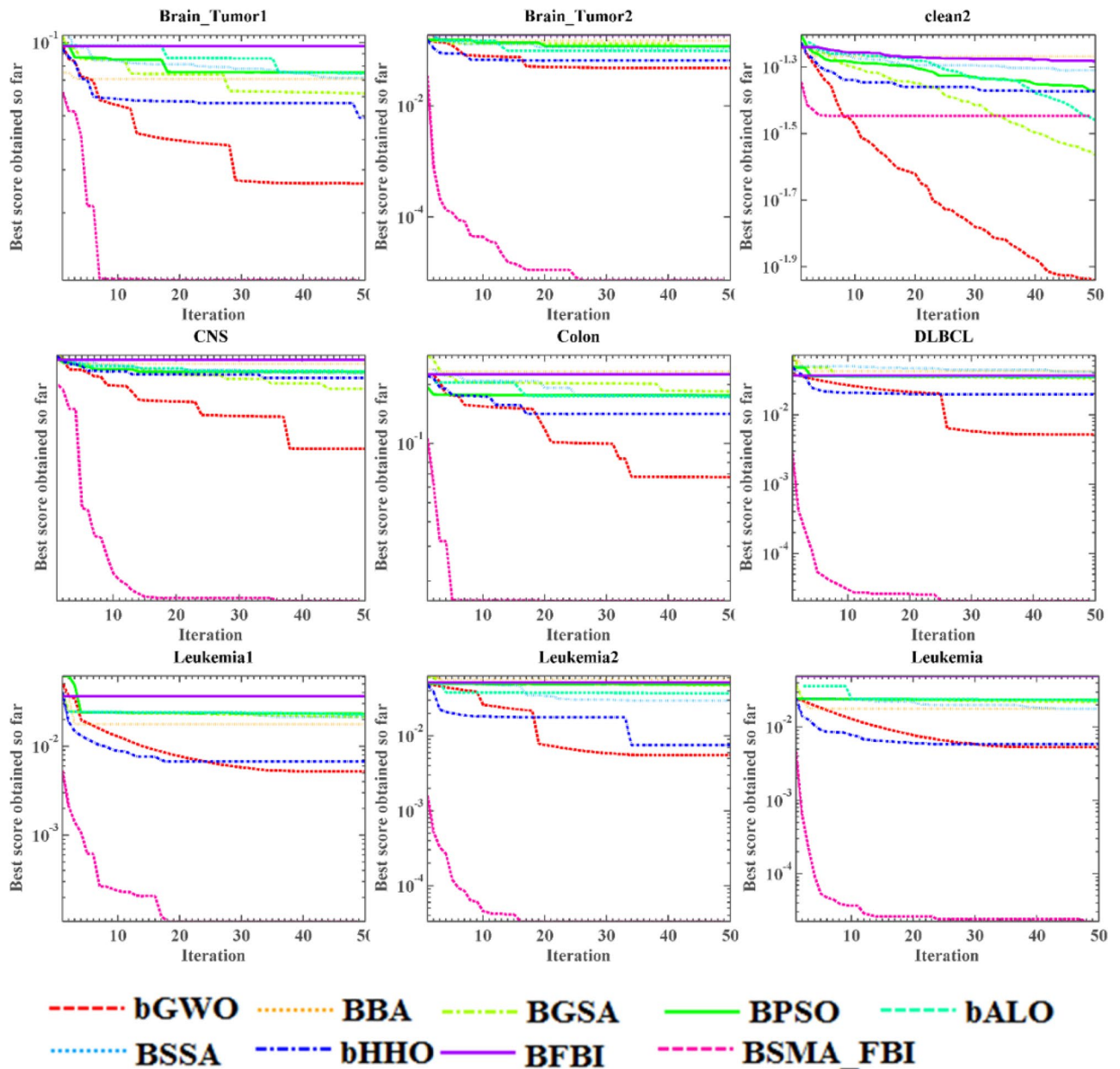
	<b>bGWO</b>	<b>BBA</b>	<b>BGSA</b>	<b>BPSO</b>	<b>bALO</b>	<b>BSSA</b>	<b>bHHO</b>	<b>BFBI</b>
Brain_Tumor1	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03
Brain_Tumor2	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03
clean2	2.03E-01	1.95E-03	1.95E-03	1.95E-03	1.95E-03	2.73E-02	3.91E-03	1.95E-03
CNS	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03
Colon	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03
DLBCL	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03
Leukemia	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03
Leukemia1	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03
Leukemia2	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	3.91E-03	1.95E-03	1.95E-03
Lung_Cancer	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03
lungcancer_3class	2.81E-01	1.95E-03	3.13E-02	1.95E-03	1.95E-03	1.95E-03	3.91E-03	1.95E-03
Parkinson	3.59E-01	1.95E-03	1.00E+00	4.06E-01	8.44E-01	1.95E-03	1.48E-01	1.95E-03
penglungEW	1.19E-01	1.95E-03	1.95E-03	1.95E-03	1.95E-03	7.81E-03	1.95E-03	1.95E-03
Prostate_Tumor	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03
SRBCT	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03
Tumors_11	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03	1.95E-03
Tumors_14	2.73E-02	1.95E-03	1.95E-03	1.95E-03	1.95E-03	3.91E-03	1.95E-03	1.95E-03
Tumors_9	8.40E-02	1.95E-03	1.95E-03	1.95E-03	1.95E-03	3.91E-03	9.77E-03	1.95E-03

**Table 6.** The  $p$ -value of Wilcoxon test between the BSMA\_FBI and alternative GS optimizers on average number of the selected features.

Based on the average computation time results in Appendix A.11, it can be observed that although the BSMA\_FBI algorithm has a high computation time, it is still superior to BFBI, which can also prove the value of the improvement side by side, reducing the time cost.

Tables 5 and 6 show the Wilcoxon signed-rank test results of BSMA\_FBI against other GS optimizers in terms of classification error as well as the number of features selected, respectively. From Table 5, it can be seen that there seems to be no significant difference between BSMA\_FBI and other GS optimizers in terms of classification error and only in a few tests is the  $p$ -value less than 0.05. However, it can be seen from Table 6 that there is a significant difference between BSMA\_FBI and other gene selection optimizers in terms of the number of features selected. This indicates that SMA\_FBI has a significant advantage over these algorithms.

The images are more intuitive and visual than the data in the tables. Figures 6 and 7 show the optimal fitness values calculated by multiple algorithms during the optimization process in the form of curves. The horizontal

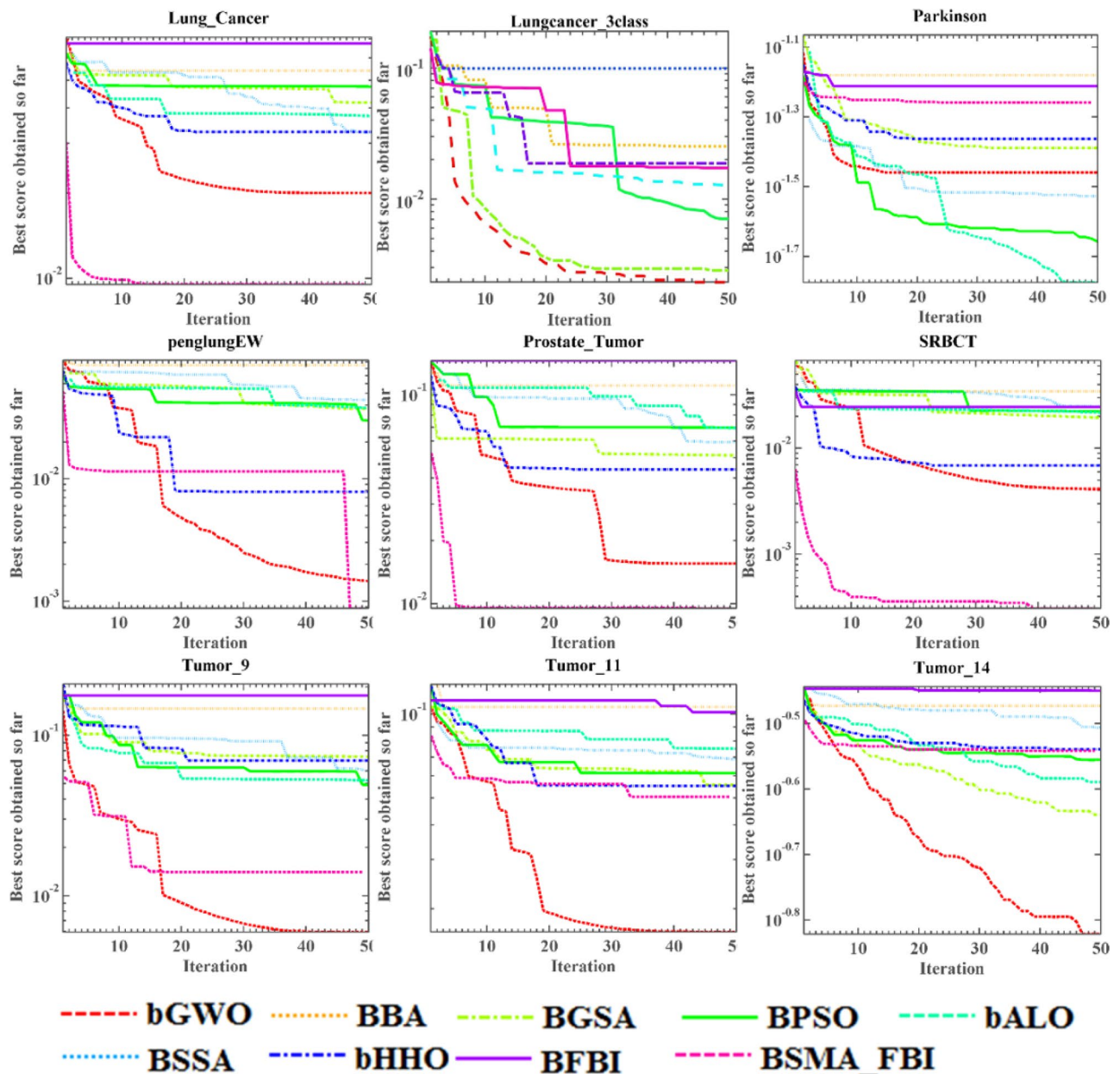


**Figure 6.** Convergence plots for the BSMA\_FBI and alternative binary metaheuristic algorithms across 9 datasets.

axis represents the iterations of the algorithms, and the vertical axis represents the average fitness value for every 10 runs of the nine algorithms. It can be seen that the best fitness values achieved by BSMA\_FBI are much smaller than those of the other algorithms on most of the datasets, such as the Brain\_Tumor1, Brain\_Tumor2, and CNS datasets. Of course, on some datasets, the fitness values obtained by BSMA\_FBI are relatively average, mainly due to trapping in the local optimum, such as the clean2, Lungcancer\_3class datasets, etc. The BSMA\_FBI algorithm's performance on the penglungEW dataset is relatively bright, and in the early stage of iteration, the fitness values obtained have been in the local optimum. Nevertheless, at the last moment of the iteration, it suddenly escapes from local optimality. It finds a better solution, which can be observed that introducing the slime mold algorithm as a mechanism enhances the diversity of the algorithm during the development period and the possibility of escaping from the local optimum.

### Discussion

This section summarizes the experimental findings of the proposed SMA\_FBI on continuous function optimization and GS problems and provides a detailed analysis of the nature of the algorithms involved as well as the experimental results. The experimental part in “Experiments” section can be divided into three aspects: (1) Comparative experiments on the population size and on the number of algorithm evaluations in function optimization to find the most suitable population size and the number of evaluations; (2) On the CEC 2017 dataset,



**Figure 7.** Convergence plots for the BSMA\_FBI and alternative binary metaheuristic algorithms across 9 datasets.

the correctness of the introduction of the slime mould search mechanism for the FBI is verified by comparing it to the base algorithm and state-of-the-art algorithms as well as the overall SMA\_FBI algorithm's superiority. (3) SMA\_FBI is applied to a high-dimensional GS optimization problem from UCI data to prove the algorithm's behavior for effective dimensionality reduction of high-dimensional data and addressing discrete combinatorial optimization challenges.

From the perspective of function optimization, it can be seen from Appendix A.4, Appendix A.5 as well as Fig. 4, that the algorithm after the integration of the slime mould mechanism is superior compared to the FBI, and its optimization ability is stronger. Secondly, the SMA\_FBI algorithm can occupy an obvious advantage, no matter it is compared with the classical DE, PSO, or the novel SMA. In addition, according to the data in Appendix A.6, Appendix A.7 and the curves in Fig. 5, we can see the comparison between SMA\_FBI and a variety of improved algorithms, which contain several champion algorithms (EPSDE, LSHADE), as well as other state-of-the-art and improved algorithms (e.g., ALCPSO, DECLS), and so on. We can see that SMA\_FBI significantly outperforms these state-of-the-art algorithms. At the same time, we can also see that SMA\_FBI is not optimal in some problems, especially in the hybrid function, which is not obvious when comparing with the basic algorithms, but when comparing with the state-of-the-art algorithms, this problem becomes evidently apparent.

In discrete combinatorial optimization, SMA\_FBI achieves satisfactory results for GS problems. We evaluated BSMA\_FBI (the binary version of the algorithm) as well as several GS optimizers using 18 datasets from the UCI repository (containing different types of data). Appendix A.8–Appendix A.11 quantitatively analyzes the performance of the algorithms in the four aspects of the number of selected features, classification errors, fitness values, and time cost, respectively. It is readily apparent that BSMA\_FBI surpasses the other optimization techniques, and the proposed algorithm maintains high classification accuracy while selecting fewer features. It can also be seen that BFBI and BSMA\_FBI algorithms are ranked in the bottom two positions in terms of time cost, while bGWO is also effective and has a higher time complexity. Given the significant reduction in time cost compared to the original algorithm, it is justifiable to take satisfaction in the performance of BSMA\_FBI. In addition, we can see that BSMA\_FBI is more effective on high-dimensional small-sample data but less effective on low-dimensional large-sample datasets and multi-classified data, which is the direction of our future improvement. In addition, Figs. 6 and 7 show that BSMA\_FBI exhibits elevated classification accuracy and convergence at a superior rate compared to its counterparts. Thus, it shows that BSMA\_FBI is a promising approach for discrete combinatorial optimization challenges in GS.

In brief, this article discusses the SMA\_FBI algorithm, which incorporates a slime mould search mechanism based on the original FBI to achieve improved algorithm performance. By comparing with other excellent algorithms on the function optimization problem, it is found that SMA\_FBI has a significant advantage in enhancing population diversity as well as convergence. In addition, compared with other GS methods on the GS problem, it is verified that BSMA\_FBI can obtain higher classification accuracy while selecting fewer features. Of course, there is also a problem that BSMA\_FBI has high time loss when performing GS, which is an optimization direction we need to consider afterward. Overall, SMA\_FBI shows good prospects in addressing diverse optimization and GS problems.

Improving the accuracy and efficiency of gene selection plays a crucial role in medical diagnosis and personalized therapy, and has a profound impact on the development of drug discovery and individualized treatment. By improving the accuracy of gene selection, we are able to more accurately identify genetic variants associated with diseases and thus diagnose them more accurately. This helps to avoid misdiagnosis and underdiagnosis, and provide patients with more precise and personalized treatment plans. Meanwhile, in the process of drug development, by accurately selecting relevant genes, we can study the mechanism of action of drugs in greater depth, accelerate the process of drug development, reduce the cost of research and development, and improve the efficiency of research and development, so as to assist in medical diagnosis and personalized treatment.

## Conclusion and future work

Throughout this manuscript, we propose a modification of FBI, SMA\_FBI, which significantly improves the capability of the original FBI. Based on the original two phases of the FBI, the slime mould search was introduced as the third phase to balance the equilibrium between exploitation and exploration better. Comparisons are made with multiple optimization algorithms on 30 CEC2017 datasets and 18 datasets from the UCI repository to validate the improvement and effectiveness of the algorithms.

The slime mould search mechanism adjusts its search pattern according to the current situation, which, in the early stage, helps to explore the search space quickly, while at a later stage, it helps to maintain the diversity of the population when developing of search space, thus avoiding falling into local optimality. Introducing the search mechanism in the FBI is equivalent to adding an inspector group to the original two investigation and search groups, significantly improving the algorithm's capability. This can be demonstrated by the experimental data given in “Experiments” section.

The SMA\_FBI algorithm achieves superior behavior in continuous optimization tasks. In 4.2 and 4.3, SMA\_FBI is compared with the original algorithms (DE, PSO, MFO, SMA, etc.) as well as the improved algorithms (EPSDE, LSHADE, ALCPSO, etc.), respectively, and is tested on the CEC2017 dataset, and the results show that SMA\_FBI possesses strong global search capability as well as fast convergence speed. Also, the proposed algorithm performs satisfactorily in high-dimensional genetic data GS tasks. In 4.4, the binary version of the suggested algorithm is contrasted with various algorithms on 18 well-known datasets from the UCI database, showing that the algorithm can achieve high classification accuracy while selecting fewer features.

In conclusion, the SMA\_FBI proposed in this paper provides excellent solutions to continuous optimization and discrete combinatorial optimization problems, encouraging the application of the algorithm to more complex and realistic challenges.

## Ethical statement

The manuscript has not been submitted to more than one journal for simultaneous consideration and has not been published elsewhere in any form or language.

## Data availability

The data involved in this study are all public data, which can be downloaded through [https://github.com/youny honey/SMA\\_FBI/tree/main](https://github.com/youny honey/SMA_FBI/tree/main).

Received: 2 January 2024; Accepted: 6 April 2024

Published online: 13 April 2024

## References

1. Chen, Y., Chen, L. & Zhou, Q. Genetic association between eNOS gene polymorphisms and risk of carotid atherosclerosis. *Herz* **46**, 253–264 (2021).

2. Li, J. *et al.* The genetic association between apolipoprotein E gene polymorphism and Parkinson disease: A meta-Analysis of 47 studies. *Medicine* **97**(43), e12884 (2018).
3. Tang, L., *et al.*, Genetic association between HER2 and ESR2 polymorphisms and ovarian cancer: A meta-analysis. *OncoTargets Ther.* 1055–1066 (2018).
4. Zhang, X., *et al.* A survey of gene selection and classification techniques based on cancer microarray data analysis. In 2018 IEEE 4th International Conference on Computer and Communications (ICCC) (2018).
5. Bonilla-Huerta, E. *et al.* Hybrid framework using multiple-filters and an embedded approach for an efficient selection and classification of microarray data. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **13**(1), 12–26 (2016).
6. Souza, J. T. D., Francisco, A. C. D. & Macedo, D. C. D. Dimensionality reduction in gene expression data sets. *IEEE Access* **7**, 61136–61144 (2019).
7. Hsueh, S. Y., *et al.* Oral cancer risk prediction by using gray wolf optimizer-based multifactor dimensionality reduction. In 2023 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan) (2023).
8. Babenko, L., Shumilin, A. & Alekseev, D. Development of the algorithm to ensure the protection of confidential data in cloud medical information system. In 2021 14th International Conference on Security of Information and Networks (SIN) (2021).
9. Zhu, Y. *et al.* Deep learning-based predictive identification of neural stem cell differentiation. *Nat. Commun.* **12**(1), 2614 (2021).
10. He, B. *et al.* TOOme: A novel computational framework to infer cancer tissue-of-origin by integrating both gene mutation and expression. *Front. Bioeng. Biotechnol.* **8**, 394 (2020).
11. Zhou, P. *et al.* Unsupervised feature selection for balanced clustering. *Knowl.-Based Syst.* **193**, 105417 (2020).
12. Huang, H. *et al.* SLNL: A novel method for gene selection and phenotype classification. *Int. J. Intell. Syst.* **37**(9), 6283–6304 (2022).
13. Li, S. & Wei, D. Extremely high-dimensional feature selection via feature generating samplings. *IEEE Trans. Cybern.* **44**(6), 737–747 (2014).
14. Happy, S. L., Mohanty, R. & Routray A. An effective feature selection method based on pair-wise feature proximity for high dimensional low sample size data. In 2017 25th European Signal Processing Conference (EUSIPCO) (2017).
15. Dokeroglu, T., Deniz, A. & Kiziloz, H. E. A comprehensive survey on recent metaheuristics for feature selection. *Neurocomputing* **494**, 269–296 (2022).
16. Chen, K. *et al.* An evolutionary multitasking-based feature selection method for high-dimensional classification. *IEEE Trans. Cybern.* **52**(7), 7172–7186 (2022).
17. Liu, L. *et al.* Multi-point shortest path planning based on an Improved Discrete Bat Algorithm. *Appl. Soft Comput.* **95**, 106498 (2020).
18. Teghem, J. Metaheuristics: From design to implementation. *Eur. J. Oper. Res.* **205**(2), 486–487 (2010).
19. Storn, R. & Price, K. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**(4), 341–359 (1997).
20. Yang, X. & He, X. Bat algorithm: Literature review and applications. *Int. J. Bio-Inspired Comput.* **5**(3), 141–149 (2013).
21. Saremi, S., Mirjalili, S. & Lewis, A. Grasshopper optimisation algorithm: Theory and application. *Adv. Eng. Softw.* **105**, 30–47 (2017).
22. Tu, J. *et al.* The colony predation algorithm. *J. Bionic Eng.* **18**(3), 674–710 (2021).
23. Ahmadianfar, I. *et al.* INFO: An efficient optimization algorithm based on weighted mean of vectors. *Expert Syst. Appl.* **195**, 116516 (2022).
24. Heidari, A. A. *et al.* Harris hawks optimization: Algorithm and applications. *Fut. Gen. Comput. Syst.-Int. J. Escience* **97**, 849–872 (2019).
25. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl.-Based Syst.* **96**, 120–133 (2016).
26. Dorigo, M., Maniezzo, V. & Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **26**(1), 29–41 (1996).
27. Ahmadianfar, I. *et al.* RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Syst. Appl.* **181**, 115079 (2021).
28. Li, S. *et al.* Slime mould algorithm: A new method for stochastic optimization. *Fut. Gen. Comput. Syst. Int. J. Escience* **111**, 300–323 (2020).
29. Rashedi, E., Nezamabadi-Pour, H. & Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **179**(13), 2232–2248 (2009).
30. Yang, Y. *et al.* Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Syst. Appl.* **177**, 114864 (2021).
31. Su, H. *et al.* RIME: A physics-based optimization. *Neurocomputing* **532**, 183–214 (2023).
32. Lian, J., *et al.*, Parrot optimizer: Algorithm and applications to medical problems. *Comput. Biol. Med.* 108064 (2024).
33. Houssein, E. H. *et al.* Liver cancer algorithm: A novel bio-inspired optimizer. *Comput. Biol. Med.* **165**, 107389 (2023).
34. Chen, L. *et al.* Energy-saving access point configurations in WLANs: A swarm intelligent approach. *J. Supercomput.* **79**(17), 19332–19364 (2023).
35. Yue, Y. G. *et al.* Improved whale optimization algorithm and its application in heterogeneous wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **17**(5), 15501477211018140 (2021).
36. Shi, M. R. *et al.* A grade-based search adaptive random slime mould optimizer for lupus nephritis image segmentation. *Comput. Biol. Med.* **160**, 106950 (2023).
37. Lakhani, A. *et al.* Blockchain multi-objective optimization approach-enabled secure and cost-efficient scheduling for the Internet of Medical Things (IoMT) in fog-cloud system. *Soft Comput.* **26**(13), 6429–6442 (2022).
38. Lakhani, A. *et al.* Federated Learning-Aware Multi-Objective Modeling and blockchain-enable system for IIoT applications. *Comput. Electr. Eng.* **100**, 107839 (2022).
39. Hussain, K. *et al.* An efficient hybrid sine-cosine Harris hawks optimization for low and high-dimensional feature selection. *Expert Syst. Appl.* **176**, 114778 (2021).
40. Neggaz, N. *et al.* Boosting salp swarm algorithm by sine cosine algorithm and disrupt operator for feature selection. *Expert Syst. Appl.* **145**, 113103 (2020).
41. Chou, J.-S. & Nguyen, N.-M. FBI inspired meta-optimization. *Appl. Soft Comput.* **93**, 106339 (2020).
42. Shaheen, A. M. *et al.* A forensic-based investigation algorithm for parameter extraction of solar cell models. *IEEE Access* **9**, 1–20 (2021).
43. Cao, M. T. *et al.* Automatic recognition of concrete spall using image processing and metaheuristic optimized LogitBoost classification tree. *Adv. Eng. Softw.* **159**, 103031 (2021).
44. Fathy, A., Rezk, H. & Alanazi, T. M. Recent approach of forensic-based investigation algorithm for optimizing fractional order PID-based MPPT with proton exchange membrane fuel cell. *Ieee Access* **9**, 18974–18992 (2021).
45. Chen, S. S., Shi, J. Y. & Doan, M. T. A meta-heuristic optimization-based approach for 3D simplified parametric analysis of embedded soil-foundation systems undergoing coupled horizontal-rocking vibrations. *Comput. Geotechn.* **155**, 105198 (2023).
46. Alkhiari, A. M., Mishra, S. & AlShehri, M. Blockchain-based SQKD and IDS in edge enabled smart grid network. *CMC-Comput. Mater. Continua* **70**(2), 2149–2169 (2022).
47. Kaveh, A., Hamedani, K. B. & Kamalinejad, M. An enhanced Forensic-Based Investigation algorithm and its application to optimal design of frequency-constrained dome structures. *Comput. Struct.* **256**, 106643 (2021).

48. Kuyu, Y. Ç. & Vatanserver, F. Modified forensic-based investigation algorithm for global optimization. *Eng. Comput.* **38**(4), 3197–3218 (2022).
49. Malika, B. K. *et al.* Quasi-oppositional forensic-based investigation for optimal DG selection for power loss minimization. *Process Integr. Optim. Sustain.* **7**(1), 73–106 (2023).
50. Nguyen, D.-T., Chou, J.-S. & Tran, D.-H. Integrating a novel multiple-objective FBI with BIM to determine tradeoff among resources in project scheduling. *Knowl.-Based Syst.* **235**, 107640 (2022).
51. Tolba, M. A. *et al.* Optimizing the distributed generators integration in electrical distribution networks: Efficient modified forensic-based investigation. *Neural Comput. Appl.* **35**, 8307–8342 (2022).
52. Chou, J. S. & Truong, D. N. Multiobjective forensic-based investigation algorithm for solving structural design problems. *Autom. Constr.* **134**, 104084 (2022).
53. Liu, L. *et al.* Performance optimization of differential evolution with slime mould algorithm for multilevel breast cancer image segmentation. *Comput. Biol. Med.* **138**, 104910 (2021).
54. Zhou, X. *et al.* Boosted local dimensional mutation and all-dimensional neighborhood slime mould algorithm for feature selection. *Neurocomputing* **551**, 126467 (2023).
55. Hu, J. *et al.* Dispersed foraging slime mould algorithm: Continuous and binary variants for global optimization and wrapper-based feature selection. *Knowl.-Based Syst.* **237**, 107761 (2022).
56. Liu, Y. *et al.* Boosting slime mould algorithm for parameter identification of photovoltaic models. *Energy* **234**, 121164 (2021).
57. Sallam, K. M., *et al.* Multi-method based orthogonal experimental design algorithm for solving CEC2017 competition problems. In *2017 IEEE Congress on Evolutionary Computation (CEC)* (2017).
58. Liu, X., Huang, H. & Xiang, J. A personalized diagnosis method to detect faults in gears using numerical simulation and extreme learning machine. *Knowl.-Based Syst.* **195**, 105653 (2020).
59. Li, J. & Lin, J. A probability distribution detection based hybrid ensemble QoS prediction approach. *Inf. Sci.* **519**, 289–305 (2020).
60. Zhou, P. *et al.* Self-paced clustering ensemble. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(4), 1497–1511 (2021).
61. Fan, M. *et al.* Adaptive data structure regularized multiclass discriminative feature selection. *IEEE Trans. Neural Netw. Learn. Syst.* **33**, 5859–5872 (2021).
62. García, S. *et al.* Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Inf. Sci.* **180**(10), 2044–2064 (2010).
63. Alcalá-Fdez, L. S., García, S., del Jesus, M. J., Ventura, S., Garrell, J. M., Otero, J., Romero, C., Bacardit, J., Rivas, V. M., Fernández, J. C. & Herrera, F. KEEL: A software tool to assess evolutionary algorithms for data mining problems (2009)
64. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **89**, 228–249 (2015).
65. Erlich, I., Venayagamoorthy, G. K., Worawat, N. A Mean-Variance Optimization algorithm. In *IEEE Congress on Evolutionary Computation* (2010).
66. Yang, X. S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspired Comput.* **2**(2), 78–84 (2010).
67. Kennedy, J. & Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (1995).
68. Mallipeddi, R. *et al.* Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl. Soft Comput.* **11**(2), 1679–1696 (2011).
69. Chen, W. *et al.* Particle swarm optimization with an aging leader and challengers. *Ieee Trans. Evol. Comput.* **17**(2), 241–258 (2013).
70. Heidari, A. *et al.* An enhanced associative learning-based exploratory whale optimizer for global optimization. *Neural Comput. Appl.* **32**(9), 5185–5211 (2020).
71. Liang, J. *et al.* Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *Ieee Trans. Evol. Comput.* **10**(3), 281–295 (2006).
72. Cai, Z. *et al.* Evolving an optimal kernel extreme learning machine by using an enhanced grey wolf optimization strategy. *Expert Syst. Appl.* **138**, 112814 (2019).
73. Lin, A. *et al.* Predicting intentions of students for master programs using a chaos-induced sine cosine-based fuzzy k-nearest neighbor classifier. *Ieee Access* **7**, 67235–67248 (2019).
74. Chen, H. *et al.* An efficient double adaptive random spare reinforced whale optimization algorithm. *Expert Syst. Appl.* **154**, 113018 (2020).
75. Tanabe, R. & Fukunaga, A. S. Improving the search performance of SHADE using linear population size reduction. In *2014 IEEE Congress on Evolutionary Computation (CEC)* (2014).
76. Adarsh, B. R. *et al.* Economic dispatch using chaotic bat algorithm. *Energy* **96**, 666–675 (2016).
77. Jia, D. L., Zheng, G. X. & Khan, M. K. An effective memetic differential evolution algorithm based on chaotic local search. *Inf. Sci.* **181**(15), 3175–3187 (2011).
78. Jadhav, S., He, H. & Jenkins, K. Information gain directed genetic algorithm wrapper feature selection for credit rating. *Appl. Soft Comput.* **69**, 541–553 (2018).
79. Emary, E., Zawba, H. & Hassanien, A. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **172**, 371–381 (2016).
80. Mirjalili, S., Mirjalili, S. & Yang, X. Binary bat algorithm. *Neural Comput. Appl.* **25**(3–4), 663–681 (2014).
81. Rashedi, E., Nezamabadi-pour, H. & Saryzadi, S. BGSA: Binary gravitational search algorithm. *Natural Comput.* **9**(3), 727–745 (2010).
82. Mirjalili, S. & Lewis, A. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm Evol. Comput.* **9**, 1–14 (2013).
83. Emary, E., Zawba, H. & Hassanien, A. Binary ant lion approaches for feature selection. *Neurocomputing* **213**, 54–65 (2016).
84. Faris, H. *et al.* An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems. *Knowl.-Based Syst.* **154**, 43–67 (2018).
85. Peng, L. *et al.* Hierarchical Harris hawks optimizer for feature selection. *J. Adv. Res.* **53**, 261–272 (2023).

## Acknowledgements

This work was supported by National Natural Science Foundation of China (62076185, 62301367) and Zhejiang Provincial Natural Science Foundation of China (LTGY24C060004). We acknowledge the comments of the reviewers.

## Author contributions

Feng Qiu: Writing – Original Draft, Writing – Review & Editing, Software, Visualization, Investigation. Ali Asghar Heidari: Writing – Review & Editing, Software, Visualization, Investigation. Yi Chen: Writing – Original Draft, Writing – Review & Editing, Software, Visualization, Investigation. Huiling Chen: Conceptualization, Methodology, Formal Analysis, Investigation, Writing – Review & Editing, Funding Acquisition, Supervision,

Project administration. Guoxi Liang: Conceptualization, Methodology, Formal Analysis, Investigation, Writing – Review & Editing, Funding Acquisition, Supervision, Project administration.

### Competing interests

The authors declare no competing interests.

### Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-024-59064-w>.

**Correspondence** and requests for materials should be addressed to H.C. or G.L.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

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



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024