Neural Computing and Applications
https://doi.org/10.1007/s00521-022-07755-8

ORIGINAL ARTICLE

Improved firefly algorithm for feature selection with the ReliefF-based initialization and the weighted voting mechanism

Xin Yong[1] · Yue-lin Gao[2]

Received: 21 March 2022 / Accepted: 17 August 2022

Abstract
Feature selection has become popular in data mining tasks currently for its ability of improving the performance of the algorithm and gaining more information about the dataset. Although the firefly algorithm is a well-performed heuristic algorithm, there is still much room for improvement as to the feature selection problem. In this research, an improved firefly algorithm designed for feature selection with the ReliefF-based initialization method and the weighted voting mechanism is proposed. First of all, a feature grouping initialization method that combines the results of the ReliefF algorithm and the cosine similarity is designed to take place of random initialization. Then, the direction of the firefly is modified to move toward the optimal solution. Finally, inspired by the ensemble algorithm, a weighted voter is proposed to build recommended positions for fireflies, which is also integrated with the elite crossover operator and the mutation operator to improve the diversity of the population. Selected from the mixed swarm, a new population is constructed to replace the original population in the next stage. To verify the effectiveness of the algorithm proposed in this paper, 18 datasets are utilized and 9 comparison algorithms (e.g., Black Hole Algorithm, Grey Wolf Optimizer and Pigeon Inspired Optimizer) from state-of-the-art related works are selected for the simulating experiments. The experimental results demonstrate the superiority of the proposed algorithm applied to the feature selection problem.

Keywords Firefly algorithm · Feature selection · Feature grouping · ReliefF algorithm · Weighted voting

1 Introduction

Data mining has become an important topic among application fields of industry [1]. With the development of data acquisition technology and data storage capability, how to deal with high-dimensional datasets is one of the key problems in data mining tasks [2, 3]. Researches have shown that a quantity of irrelevant and redundant features may have an effect on the efficiency and accuracy of the data mining algorithms [4]. In order to increase the prediction performance and decrease the computational complexity of data mining algorithms, dimensionality reduction technologies including feature selection and feature extraction have been widely used in real applications.

Feature selection is a technique that is intended to elect the most relevant and important features for building a data mining model [5]. The main objective of the feature selection research is to remove redundant and irrelevant features, improve classification accuracy and make the number of features appropriate. Although feature selection is used as a pre-processing step in developing the models, it plays a vital role in the whole process [6]. As noted by Dash in [7], feature selection helps to reduce the time complexity of the algorithms and contributes to getting a better perception of real-world applications. According to whether the classifiers get involved, feature selection methods can be classified into two categories: wrapper methods and filter methods [2, 8].

✉ Yue-lin Gao
1993001@nun.edu.cn

Xin Yong
yongxin_Azure@163.com

[1] School of Computer Science and Engineering, North Minzu University, Wenchang North Street, Yinchuan 750021, Ningxia Province, China

[2] Ningxia Province Key Laboratory of Intelligent Information and Data Processing, North Minzu University, Wenchang North Street, Yinchuan 750021, Ningxia Province, China

Published online: 09 September 2022

Filter methods with designed evaluation criteria generally work by judging the features or feature subsets [9]. The calculation of the statistical measures only requires the information of datasets, so the filter methods can be independent of the learning algorithms in modeling. The advantage of the filter methods is that they are generally faster and the results generated are more easier to be utilized as the inputs of any modeling algorithms. As mentioned in [5], filter methods can be divided into univariate and multivariate methods. The former category only considers the relationships of features and the target class while the latter one also takes the dependency of the features into consideration. The typical measures of filter methods are information, distance, correlation and so on.

Take the typical filter method Relief algorithm as an example. Proposed in 1992 [10], the algorithm employs a distance measure to evaluate a statistic for each feature. It is worth mentioning that the original Relief algorithm is limited to two-class problems; therefore, Igor [11] proposed a generalized Relief algorithm, the ReliefF algorithm, which can deal with the problems of multiple classes. However, a repeatedly mentioned disadvantage of such algorithms is that they may not remove the feature redundancies, which results in the inefficiency of the solutions [9].

Among the articles before, filter methods ignore the combination effect between the feature subsets and learning algorithms. On the other hand, redundant features should be treated with caution when a filter-based method is used for feature selection. In contrast, the wrapper methods utilize classifiers to evaluate a given feature subset, which leads to a better performance than the filter methods [8]. As an NP-hard problem, searching for the best feature subset among 2^N candidates costs greatly concerning a high-dimensional dataset [12]. To tackle this problem, nature-inspired heuristic algorithms are introduced in dealing with the optimization of the searching process [13]. Compared with the complete search, greedy search and random search, the meta-heuristic algorithms show better global optimization ability in the process of solving such combinatorial optimization problems [14]. Therefore, the utilization of meta-heuristic algorithms as a search strategy for a wrapper-based method has been gradually accepted by researchers in this field. The algorithms widely studied contain differential evolution (DE) [15, 16], particle swarm optimization (PSO) [17–20], genetic algorithm (GA) [21], grey wolf optimization (GWO) [22, 23], grasshopper optimization algorithm (GOA) [24, 25], salp swarm algorithm (SSA) [14, 26, 27], butterfly optimization algorithm (BOA) [8], artificial bee colony (ABC) [28], whale optimization algorithm (WOA) [29, 30], Harris Hawks optimization (HHO) [31], gravitational search algorithm (GSA) [3] and so on.

As an effective global optimal solution search algorithm, the heuristic algorithm can be used in the wrapper algorithm to help search for the optimal feature subset. However, using various heuristic algorithms to solve the feature selection problem may result in some differences in precision and complexity. Here in this paper, we choose the firefly algorithm (FA) as the studied method. FA is a creative proposed swarm intelligent algorithm developed by Yang in 2008 [32]. Since its proposal, FA has received widespread attention and interest from scholars. In recent years, improvements in different aspects of this algorithm have been proposed [33–35] and extensively utilized to optimize problems in medicine [36], engineering [37] and other fields [38, 39]. As for the feature selection problem, researchers recently have proposed DBFA [40], Rc-BBFA [41], MIFA [39] and so on. However, according to the no-free lunch theorem [29], there exists no single optimization algorithm which can solve all the optimization problems perfectly. Due to the weaknesses of the FA, there remains much scope for improvement and a method designed for feature selection based on its characteristics is needed. To solve the problems of insufficient local development ability, low convergence accuracy and premature convergence, we proposed a novel improved FA.

The proposed method is motivated by the problems mentioned above, and it provides a better solution for feature selection. The main contributions of this paper are summarized as follows:

(1) Based on the ReliefF algorithm and the cosine similarity, a new feature grouping algorithm is designed to select initial feature subsets and replace the traditional random generation. The proposed initialization method can impressively increase population diversity, improve the quality of the initial population and accelerate the convergence speed of the algorithm.
(2) The direction of the fireflies is modified to move toward the optimal solution. At the same time, a weighted voter that is inspired by the ensemble learning algorithm is proposed for the application of the FA in terms of the feature selection problem. The recommended positions generated by the voters are modified by the elite crossover operator and the mutation operator to improve the diversity of the population. Then, the new swarm is selected from the mixed population for the next generation. With the proposed method, the performance of FA is improved in terms of convergence accuracy and capability of searching for the global optimal solution.

(3) The efficiency of the proposed method is validated by experiments based on several datasets. A comprehensive study on comparison with the algorithms selected from state-of-the-art related works is conducted, and the experimental results demonstrate the superiority of the proposed algorithm applying to the feature selection problem.

This paper is organized as follows. Section 2 provides a preliminary knowledge of the FA and ReliefF algorithm. In Sect. 3, the proposed algorithm is more detailed, while the experimental results and corresponding analysis are given in Sect. 4. Finally, the conclusions of the research and several future directions are suggested in Sect. 5.

2 Preliminaries

2.1 Continuous FA

FA is a kind of swarm intelligent optimization algorithm that simulates the luminous behavior of fireflies in nature to find the optimal solution. The algorithm itself only abstracts the glowing characteristics of fireflies to search for the solution space region. During the whole process, every firefly constantly moves toward the brighter fireflies in the solution space and then finally converges to the optimal solution.

The key points of this algorithm lie in the brightness and attractiveness of fireflies. The brightness and the attractiveness of the other fireflies determine the direction and distance of the current firefly's movement separately. In each iteration, the better the quality of the solution, the higher the brightness of the firefly. However, the brightness of other fireflies seen in the field of view of each firefly is also affected by the relative position distance. The shorter the relative distance between the fireflies, the higher the relative brightness. Furthermore, each individual's attraction to other fireflies is related to relative brightness and it is also inversely proportional to the distance between the individual fireflies. Firefly populations search for optimal solutions by exchanging information through brightness and attraction, which means each firefly is more susceptible to being attracted by the surrounding brighter fireflies and finally converges through this mechanism.

Suppose there are N fireflies searching in the d-dimensional space, then the location of a firefly is denoted as $X_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,d}]$. The algorithm is described as follows:

$$I_{oi} = f(X_i) \tag{1}$$

where I_{oi} is the brightness of the i th firefly and $f(X_i)$ represents the fitness function value of this firefly. As mentioned before, the brightness of each firefly in the field of view of the other firefly also depends on the relative distance that can be represented as:

$$I_i = I_{oi} \cdot e^{-\gamma \cdot r_{ij}} \tag{2}$$

where I_i is the relative brightness, γ is the light absorption coefficient. $\langle \cdot \rangle$ is an element by element multiplication. In FA, γ controls the variation in the light intensity and it is often set as a constant number. r_{ij} is the distance between the i th and j th fireflies, which can be defined by Eq. (3).

$$r_{ij} = \|X_i - X_j\| = \sqrt{\sum_{n=1}^{d} (X_{i,n} - X_{j,n})^2} \tag{3}$$

Then, the attractiveness of firefly can be defined as:

$$\beta = \beta_0 \cdot e^{-\gamma \cdot r_{ij}^2} \tag{4}$$

where β_0 is the attractiveness at the distance $r_{ij} = 0$. If the i th firefly is attracted by j th firefly, its movement is formulated by Eq. (5).

$$X_i^{t+1} = X_i^t + \beta \cdot \left(X_j^t - X_i^t\right) + \alpha \cdot (rand - 1/2) \tag{5}$$

where t represents the iteration number, α is the step parameter and $rand$ is a random generator uniformly distributed in $[0, 1]$. $\alpha \cdot (rand - 1/2)$ is added to increase the space range of the search domain and prevent the algorithm from falling into premature convergence. In each iteration, every firefly is executed cyclically using Eq. (5) to update, and the algorithm will converge to the optimal value eventually.

2.2 FA for feature selection

For the purpose of applying FA to feature selection problem, certain modifications to the position representation, fitness function and update formulas should be made because the algorithm is originally designed for continuous optimization problem. The main objective of feature selection problem is to get as fewer features as possible to maximize the performance of subsequent algorithms. Therefore, a representation applied to this problem is needed to build connections between continuous values and discrete questions. At present, a binary representation is the most popular method among the articles in this field [3, 28–31]. As shown in Fig. 1, each element represents whether the feature is selected, that is, "0" means not

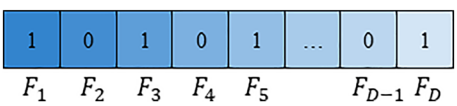


Fig. 1 Binary representation

selected, "1" means selected, so that a subset of features can be represented as a binary combination. Using the binary representation, each firefly can be regarded as a solution to the feature selection problem.

The feature selection problem can be regarded as an optimization problem in essence, and the design of its objective function is also worth considering. According to previous studies, the fitness function of FA is set by the objective function of the problem. In FA, the fitness function is used to evaluate the quality of the locations of the firefly individuals to guide the convergence process. The most essential factors that need to be considered are mainly the classification performance obtained by the feature subset and the number of features it contains. Therefore, throughout the comprehensive consideration, the fitness function can be designed as Eq. (6):

$$fitness = f(X_i) = \alpha_w \cdot err(X_i) + \beta_w \cdot \frac{|X_i|}{d} \tag{6}$$

where $f(X_i)$ represents the fitness value of X_i, $err(X_i)$ is the error rate of the classifier when using the subset of X_i, and $|X_i|$ is the total number of the selected features. α_w and β_w can be regarded as the importance of the two parts in the formula, $\alpha_w \in [0, 1]$ and $\alpha_w + \beta_w = 1$. It is easy to infer that the minimum of the objective function represents the best quality of the solutions.

In addition, how to convert the calculation process of continuous values to the binary representation in FA is also a question worth noting. [30] proposed the transform method with the sigmoid function that is formulated in Eq. (7). Sigmoid function is a common transformation function that smoothly maps continuous values to $[0, 1]$. After introducing a random threshold and executed by Eq. (8), the result values calculated by Eq. (5) further become a set of 0 s and 1 s, so that the fireflies are bound to a search space of limited values effectively.

$$Sigmoid(v_i) = \frac{1}{1 + e^{-v_i}} \tag{7}$$

$$X_i = \begin{cases} 0, & if \quad rand < Sigmoid(v_i) \\ 1, & if \quad rand \geq Sigmoid(v_i) \end{cases} \tag{8}$$

2.3 ReliefF algorithm

In this paper, the ReliefF algorithm is chosen because there are multiple labels in the datasets we are dealing with. Therefore, in order to adapt the algorithm to more application datasets, the ReliefF algorithm is integrated into the proposed algorithm. The ReliefF algorithm and the Relief algorithm are inextricably linked. The main idea of the Relief algorithm is to estimate features based on how the feature helps distinguish instances that are close to each other [11]. For each instance selected, the algorithm searches its two nearest neighbors: one from the same class, called the nearest hit, and the other from a different class, called the nearest miss. The Relief algorithm gives more weights to those features that distinguish the instances from different classes, while the ReliefF algorithm is constructed based on the same rationale. After introducing the concepts mentioned above, the estimate of a feature F can be defined by the difference of probabilities as follows:

$$\begin{aligned} W(F) = {} & P(different\ value\ of\ F | different\ class) \\ & - P(different\ value\ of\ F | same\ class) \end{aligned} \tag{9}$$

As pointed out in [9], the Relief-based algorithm is capable of detecting feature dependencies and is relatively fast to get the results, but not removing feature redundancies may be a limitation. However, how to deal with feature redundancies is still controversial and it is also hard to distinguish whether there exists information lost during the process. Inspired by the studies above, we consider building a filter-based wrapper algorithm that utilizes the estimates of the ReliefF algorithm as expert knowledge to guide the process of initialization.

3 The proposed improved FA (IFA)

In this section, the proposed improved FA (IFA) is introduced and detailed in three parts. First of all, the method of feature grouping based on the ReliefF algorithm and cosine similarity is discussed. Then, we modified the moving mechanism of the classical FA to simplify and optimize the process. Last but not the least, a weighted voter mechanism simulating the ensemble algorithm integrated with the elite crossover operator and mutation operator is proposed. The flowchart of the proposed algorithm is shown in Fig. 2.

3.1 Feature grouping based on the ReliefF algorithm and cosine similarity (FG-RC)

Random initialization strategy is the most universal selection of the researches about tackling feature selection problem by heuristic algorithms. As an important part of the algorithms, an initial solution of better quality is more likely to help the algorithm converge to the optimal solution. [42] proposed three new initialization strategies for PSO solving the feature selection problem, which has shown a better performance than the traditional methods.

As mentioned before, the filtering feature selection algorithms are reliable and fast enough, but the articles suggest that the wrapper methods, especially those that combined the intelligent optimization algorithm, perform better. Therefore, intending to combine the advantages of filter methods and wrapper methods and build a filter-based

wrapper algorithm, this paper chooses to utilize the ReliefF algorithm as expert knowledge to help group and cluster the features for initialization, which can provide a better initial solution than random selection. In addition, the proposed method continues to select the features for grouping by cosine similarity each time until every feature has been arranged. The formula of cosine similarity is as follows:

$$cosine\ similarity = \frac{F_1 \cdot F_2}{F_1 \cdot F_2} = \frac{\sum_{i=1}^{S} F_{1,i} \times F_{2,i}}{\sqrt{\sum_{i=1}^{S} \left(F_{1,i}\right)^2} \times \sqrt{\sum_{i=1}^{S} \left(F_{2,i}\right)^2}} \quad (10)$$

The greater the cosine similarity between two features, the closer their directions are. Similar features should be classified into the same group for feature grouping. Inspired by [43], blank features are added in each group to allow zero features can be selected per group each time in order to add more randomness. Finally, the completed feature grouping is utilized with subsequent random selections to generate the initial population, which is delivered to the algorithm in the next step.

Figure 3 shows the total process of the proposed FG-RC method, where the squares marked as $F_i(i = 1, 2, \ldots)$ represents the i th feature or the blank feature and the circles marked as $X_i(i = 1, 2, \ldots)$ represent the i th firefly. To be clearer, the pseudo-code of the algorithm is shown in Algorithm 1.

Algorithm 1. Pseudo-code of the FG-RC

Dataset with d features and s samples

Define the number of starting groups k, the number of generated fireflies N and the random control parameter r_1

Sort all the d features in descending order according to the estimation of ReliefF algorithm

Calculate the cosine similarity matrix $C_{D\times D}$

For $i \leftarrow 1$ to k do:

    Tag the ith feature in order according to the sort given before

$count = d - k$

While $count > 0$:

    Choose the maximum value in C and its index represents the feature i and feature j

    If feature i has been tagged then

        If feature j hasn't been tagged then

            Tag the feature j as the same tag with feature i

    Else if feature j hasn't been tagged then

            Tag feature i and feature j with a new tag

        Else

            Tag the feature j as the same tag with feature i

    Update the matrix C

Add blank features in each feature group

for $i \leftarrow 1$ to N do

    for each group do

        for $j \leftarrow 1$ to the length of the group do

            if $rand < r_1$ and the randomly selected feature is not blank then

                add the selected feature to the feature subset of ith firefly

Output the generated initial population

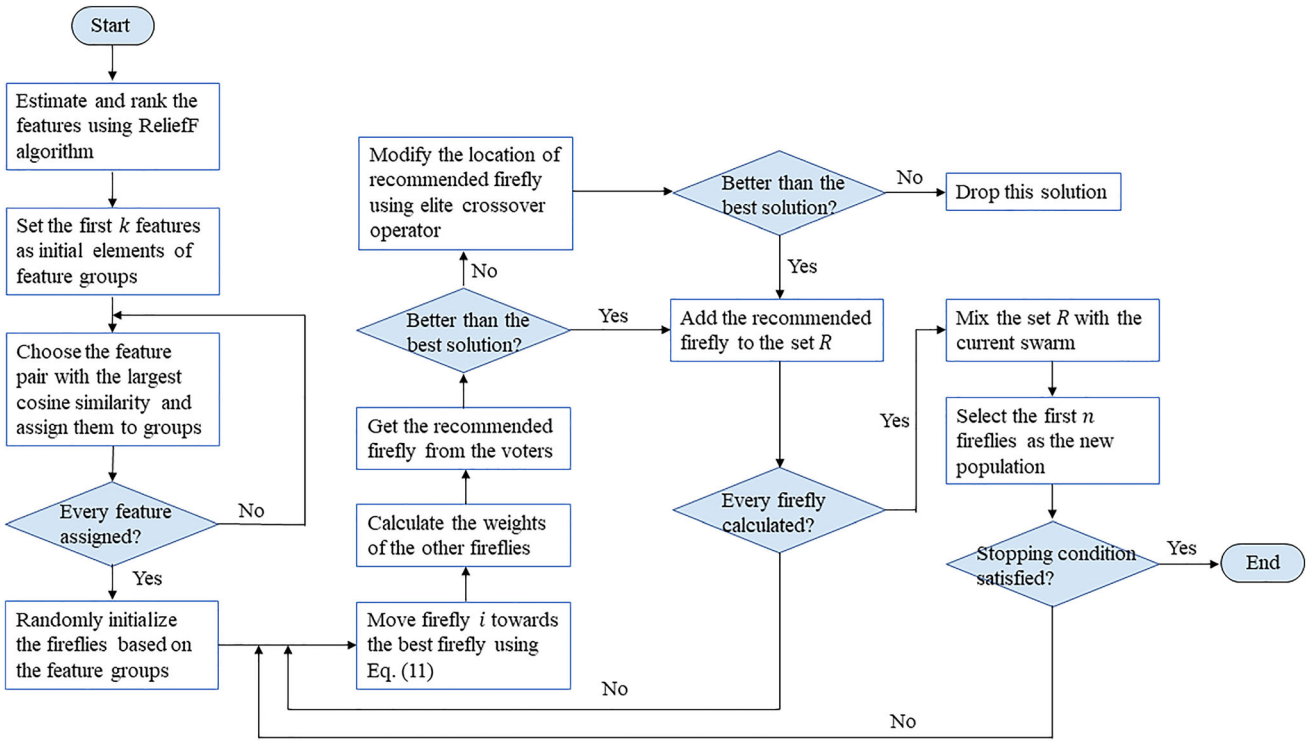


Fig. 2 Flowchart of the proposed algorithm

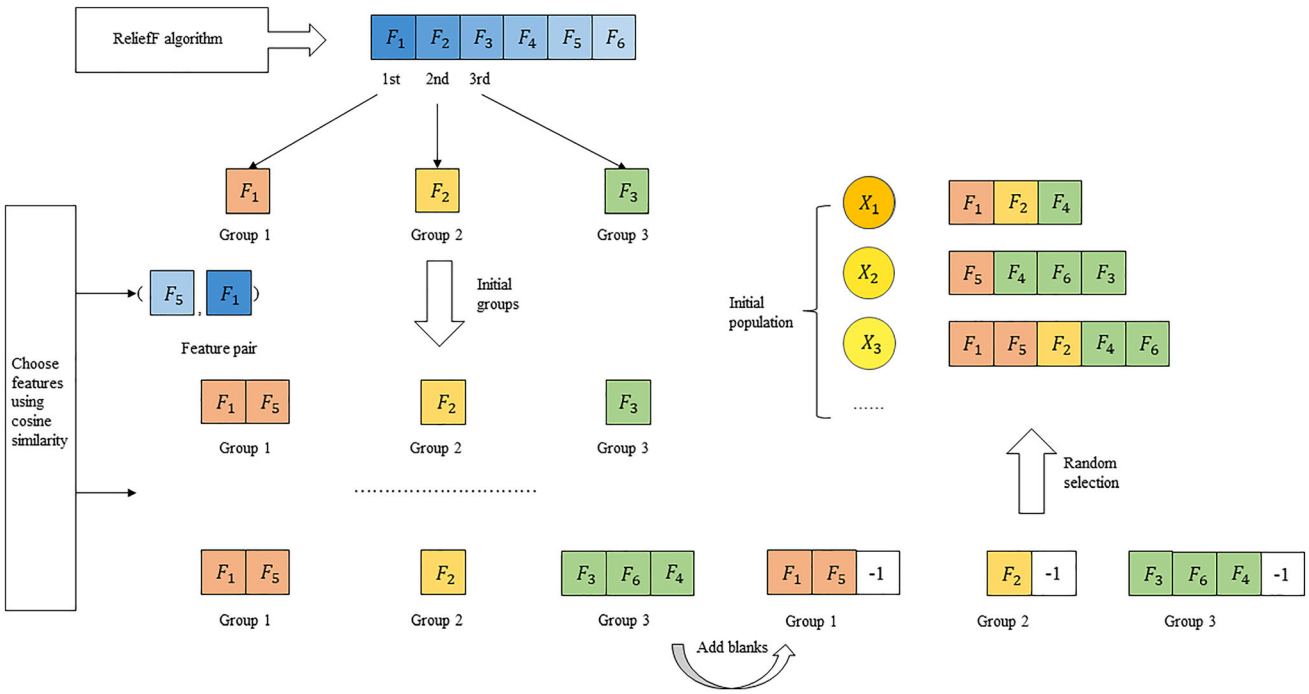


Fig. 3 Total process of the proposed feature grouping method (FG-RC)

3.2 Movement behavior toward the best firefly individual

In the traditional FA, the fireflies with higher brightness generally only have a strong attraction effect on individuals within a relatively short distance. Therefore, most fireflies have the probability to be guided to move toward positions of poor solution quality. On the other hand, for each iteration, there are the indiscriminate movements of a firefly toward each firefly whose brightness is higher than itself. This situation not only increases the complexity of the algorithm but also leads to a low convergence accuracy and a higher possibility of falling into the local optimal solution.

The optimal solution for the population in each iteration has great potential for the leading role of the population. The more the population tends to the optimal individuals generated, the easier it is to approach the optimal solution. Take the black hole search algorithm (BHA), for example, the stars (the agents) are attracted by the black hole (the best solution position) and keep approaching it, which shows a fast convergence speed and good performance. Inspired by the BHA [44], we modify the updating mechanism of the classical FA. The fireflies are forced to move toward the optimal individual each time, which will improve the convergence of the algorithm in this paper. Different from the random attraction model [45] or the global attraction model, the movement of the optimal solution direction can ensure that the individual fireflies always move in a better direction. The corresponding update formula is modified as Eq. (11).

$$X_i^{t+1} = X_i^t + \beta \cdot \left(X_{best}^t - X_i^t\right) + \alpha \cdot (rand - 1/2) \tag{11}$$

where X_{best}^t represents the best individual in iteration t and the β needs to be modified accordingly as follows:

$$\beta = \beta_0 \cdot e^{-\gamma \cdot r_{i,best}^2} \tag{12}$$

where $r_{i,best}$ is the distance between the i th firefly and the best firefly. As discussed in [46], the order of fireflies significantly influences the performance of the algorithm because each firefly utilizes the preceding updated fireflies to calculate its position. In consequence, once we remove this effect, the fireflies will have more freedom to move toward the optimal location. However, this adjustment runs the risk of falling into a local optimum, which may reduce the precision of the original algorithm. Therefore, we introduce a weighted voter that simulates ensemble learning to improve the population diversity of the algorithm to help jump out of the local optimum.

3.3 Weighted voting mechanism

Ensemble algorithm is an important category of machine learning algorithms. For purpose of improving accuracy rates, ensemble methods combine several base models with a designed voting mechanism to build models and its performance is often better than a single classifier's. There are three main categories according to the relationship between learners, such as boosting, bagging and stacking. The advantages of the ensemble algorithm depend mostly on the voting mechanism and the combination method.

Considering the Adaboost algorithm, we find that FA has a certain similarity with it: (1) The training process of the Adaboost algorithm is one classifier after another, and the latter trainer uses the results of the previous trainer. FA also updates one firefly after another, and the later firefly is influenced by the firefly updated before. (2) Both algorithms aim to get a better solution by performing computations for each classifier/firefly in each iteration. Of course, the essential idea that the Adaboost algorithm adjusts the weights of the samples and classifiers in each iteration so that the samples with poor classification results receive attention is not reflected in FA. Inspired by the design idea mentioned above, this paper investigates the application characteristics of FA to the feature selection problem and designs a weighted voting mechanism to optimize the algorithm. In a similar way, we try to apply this voting mechanism to FA, which is not a complete imitation or copy of the Adaboost algorithm, but a design improvement that draws on the ideas combined with the needs of practical applications.

In this mechanism, we give the fireflies with better performance greater weights and let them vote to build a new recommended firefly for the current firefly. The location of recommended firefly is adjusted by the weights and the locations of the other fireflies. However, there is also a certain problem that similar voting devices are more likely to give the same optimal results, which leads to trapping in the local optima and getting worse accuracy. Therefore, to increase the diversity of the population, the elite crossover operator and mutation operator are introduced as correction methods after each recommendation is given. If the quality of the corrected solution cannot be improved, the recommendation will be abandoned. Otherwise, they will be added to a new population set R. Then, mix R into the original population P and select the best fireflies of size N for retention. The preserved fireflies are built as a new swarm for the next iteration. The pseudo-code of the total proposed algorithm is shown in Algorithm 2.

Algorithm 2. Pseudo-code of the proposed improved firefly algorithm (IFA)

Dataset with d features and s samples

Define the maximum number of iterations T, parameters including k, r_1,w_a, r_2

Generate the initial population P of size N using FG-RC.

Sort the population and set the optimal solution as X_{best}.

For $t \leftarrow 1$ to T do:

  For $i \leftarrow 1$ to N do:

    Update the position of X_i by Eq. (11).

    Transfer the new position to the binary representation by Eq. (8).

    For $j \leftarrow 1$ to N do:

      If $f(X_j) < f(X_i)$ do:

        Record X_j in set Z.

    Calculate the weight vector W_z by Eq. (12) using Z.

    Normalize W_z by Eq. (13).

    Initialize the recommend position as d-dimensional zero vector l_r.

    If $|Z| > 1$ do:

      For $k \leftarrow 1$ to $|Z|$ do:

        l_r+= $W_z[k] \cdot Z[k]$

      Transfer the position l_r to the binary representation by Eq. (14).

      If $f(l_r) > f(X_{best})$ do:

        Modify l_r by the elite crossover operator in 3.3.3.

      Else if $f(l_r) = f(X_{best})$ do:

        Modify l_r by the mutation operator in 3.3.4.

      If $f(l_r) < f(X_{best})$ do:

        Add l_r to the recommended population set R.

  Mix and sort the population P with the recommended population set R. Select the best N agents as the next swarm.

3.3.1 Design of weights

It may be inappropriate that FA only uses the distance factor between fireflies without considering the brightness factor of fireflies in the update process of fireflies [41], which results in fireflies tending to move toward those who are closer rather than those with higher brightness. This phenomenon may lead to the swarm trapping in a local optimum and the convergence accuracy being reduced. Therefore, when designing the weights, we tend to consider both the distance and brightness between fireflies. Define Z is a set of fireflies that are brighter than the current firefly. The weight design is formulated as follows:

$$w_j^i = \begin{cases} w_a \cdot \left(e^{f(X_i)-1}\right) + w_b \cdot \dfrac{1}{r_{ij}}, else \\ 0, \ \text{if } X_j \notin Z \end{cases} \tag{13}$$

where w_j^i is the weight of j th firefly when the i th firefly is updated, the former part represents brightness and the latter part represents distance. $|Z|$ is the total number of fireflies in Z. The parameters w_a and w_b control the importance of the two parts, which means the corresponding part of the greater one represents is more decisive. w_a and w_b are in $[0, 1]$, $w_a + w_b = 1$. Since the fitness value of a firefly is no more than 1 and greater than 0 as well as the inverse of distance distributes in the same way, the value of w_j^i will be less than 2 and normalization is needed as shown in Eq. (14).

$$W_j^i = \frac{1}{|Z|}\sum_{i=1}^{|Z|} w_j^i \tag{14}$$

3.3.2 Voting mechanism for recommended locations

For each firefly in each iteration, the fireflies that are brighter than themselves are used to vote for a recommended location that is likely to be a better solution. Figure 4 shows that the current location firefly only moves forward to the optimal location and the recommended location firefly is built by the voting of the fireflies that are brighter than the current firefly. As for the way to create a recommended location, consider a vector l_r that has the same length d with fireflies,$l_r = [0, 0, \ldots, 0]$. Each element of l_r is $e_i (i = 1, 2, \ldots, D)$, which is updated by Eq. (15).

$$e_i = \begin{cases} 0, & \text{if } \sum_{z \in Z} W_z \cdot X_z > 0.5 \text{ and } rand > r_2 \\ 1, & else \end{cases} \tag{15}$$

where W_z is the weight of z th firefly in Z and r_2 is a parameter that adds more randomness. r_2 can be defined between 0 and 1, and each value of location only depends on the weights and locations of voting fireflies when r_2 is 1.

Figure 4 shows how the voting mechanism works in the proposed method. It can be seen from the figure that, for the fireflies currently to be updated, only the fireflies with higher brightness are given weights and they can vote for the recommended positions. The recommended position is mainly determined by the positions and weights of the voters. Random control parameter r_2 and the generated random number also have an influence on the results.

However, the occurrence of duplicate values among recommended locations is observed and predictable. The reason may be that fireflies with high weights may dominate the process and then make the recommended fireflies more similar. Therefore, in order to increase population diversity and help modify the recommended locations to get better results, the elite crossover operator and mutation operator are introduced.

3.3.3 Elite crossover operator

The strategy of GA is to select the pairs of individuals to cross and retain the best individuals to make the population move toward the optimal solution, which increases the diversity of the population solution and improves the possibility of jumping out of the local optimal solution. The performance of the genetic algorithm proves the superiority of its crossover and mutation operators. For the swarm intelligence optimization algorithm, the elite individuals in the population have an important guiding role and the potential to find the optimal solution. Theoretically, compared to the traditional application of only retaining elite individuals in the iterative process, using elite individuals to participate in the crossover operation to spread their dominant gene fragments is more suitable. The advantage of making use of the elite individuals is that they owned partial better gene expression position compared with non-elite individuals so the results of the crossover operator will be greatly improved. In view of this, the crossover operator of elite participation is proposed in this paper to help modify the recommended locations mentioned before.

Taking l_r (a recommended firefly given before) as an example, $l_r = [e_1, e_2, \ldots, e_d]$, X_{best} is the best individual and $X_{best} = [x_{b,1}, x_{b,2}, \ldots, x_{b,d}]$. The steps of the crossover operator of elite participation are provided as follows:

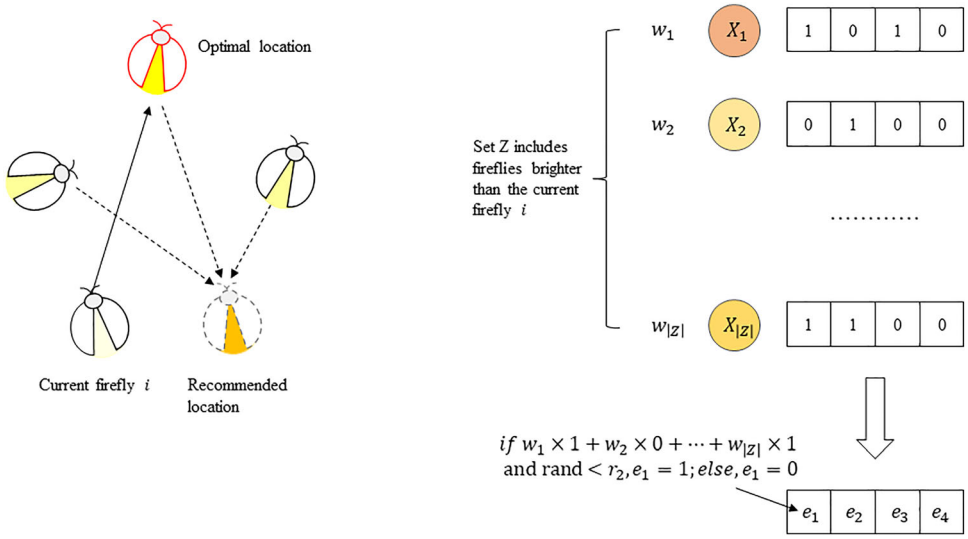


Fig. 4 Process of voting mechanism for recommended locations

Step1: Generate random crossover position q.
Step2: Divide $l_{r,i}$ into $[e_1, e_2, \ldots, e_q]$ and $[e_{q+1}, e_{q+2}, \ldots, e_d]$ and divide X_{best} into $[x_{b,1}, x_{b,2}, \ldots, x_{b,q}]$ and $[x_{b,q+1}, x_{b,q+2}, \ldots, x_{b,d}]$. Take out the gene segments $[e_{q+1}, e_{q+2}, \ldots, e_d]$ and $[x_{b,q+1}, x_{b,q+2}, \ldots, x_{b,d}]$.
Step3: Swap two gene segments and generate children c_1 and c_2. c_1 can be represented as $[e_1, e_2, \ldots, e_q, x_{b,q+1}, x_{b,q+2}, \ldots, x_{b,d}]$ and c_2 is $[x_{b,1}, x_{b,2}, \ldots, x_{b,q}, e_{q+1}, e_{q+2}, \ldots, e_d]$.
Step4: Calculate the fitness values of children.
Step5: Compare the children and the parents, then select the best individual to output.

The operator is simple to implement but has a remarkable effect that it can update the position of fireflies quickly and obtain a relatively better solution easily. However, suppose the algorithm falls into a local optimum and the majority of the population are already elite individuals, then the proposed crossover operator may distinctly have no effect. In order to avoid this problem, a mutation operator is proposed to help improve the situation.

3.3.4 Mutation operator

For the purpose of improving the diversity of the algorithm without affecting the quality of the solution, the mutation operator is utilized to further explore the region of the search space that has never been reached. Inspired by [47], the mutation operator designed in this paper is also divided into two parts and each time it is randomly selected from the set of 0 or the set of 1 for modification. Taking l_r (a recommended firefly given before) as an example and suppose $l_r = [1, 0, 1, 1, 0]$. The detailed steps are shown as follows:

Step 1: Record the position of 1 and 0 as set *One* and *Zero*. In this example, $One = \{0, 2, 3\}$ and $Zero = \{1, 4\}$.
Step 2: Randomly select a number from the set $\{0, 1\}$. If the chosen number is 1, then randomly select a location number from *One*. If not, randomly select a location number from *Zero*.
Step 3: Exchange 1 and 0, which depends on the results of step 2. For example, 2 is selected from the set *One*, then l_r will be converted to $[1, 0, 0, 1, 0]$.

3.3.5 Brief summary

In a brief summary, taking the current updating firefly X_i as an example, the weights of fireflies other than X_i are calculated as shown in Eq. (13) and normalized in Eq. (13). It can be seen from the equation that the fireflies with better quality or closer distance to X_i have more weights, while the weights of the others are 0. These fireflies (only fireflies better than X_i are used in fact) vote to give a recommended position by Eq. (15). As shown in Fig. 4, the modification operators (elite crossover operator and mutation operator) are used to make corrections because of the duplication problem. The above two modification methods are added to expand the search scope and increase the diversity of the population. Only the modified solutions that are better than the optimal solution are kept in recommended population set. That is to say, each time a firefly updating process is completed, a new firefly that is better than the existing optimal solution may be added to the recommended population. For the next step, the current swarm is mixed with the recommended population, and then, the best N agents are selected to be the new swarm of the next iteration. The brief process of this part is shown in Fig. 5.

To facilitate the illustration of the improvement in the population diversity by the above modification method, a part of the experiments conducted later is shown here. As shown in Fig. 6, the curves represent the cumulative number of solutions searched (i.e., the number of feature subsets already searched) during the same number of iterations of different algorithms running on the same dataset, respectively. These algorithms are IFA without modifiers, the complete IFA and the classical FA, respectively. As can be seen from the figure, the complete IFA searches more solutions than IFA without the modifiers, while the classical FA searches far more solutions than the first two. The dataset used here is the QSAR biodegradation dataset selected from the UCI Repository [48] and the total number of its features is 41, which means that the total number of feature subsets is $2^{41} \approx 2.2 \times 10^{12}$. Here, the number of different solutions searched by the algorithm can be considered as a rough measure of population diversity. Although the search space of an algorithm is limited, it is more desirable to search for the optimal solution properly without wasting resources by searching for too many solutions rather than searching for as many as possible solutions. In the case of the experiments in this paper (IFA outperforms FA for the most part), the fact that FA searches a significantly larger number of feature subsets than IFA means that IFA is less complex than FA while maintaining or even improving the search accuracy.

4 Experiments and results

This section presents the experimental results of verifying the effectiveness of the proposed IFA. The comparisons are divided into two parts: the comparison of initialization

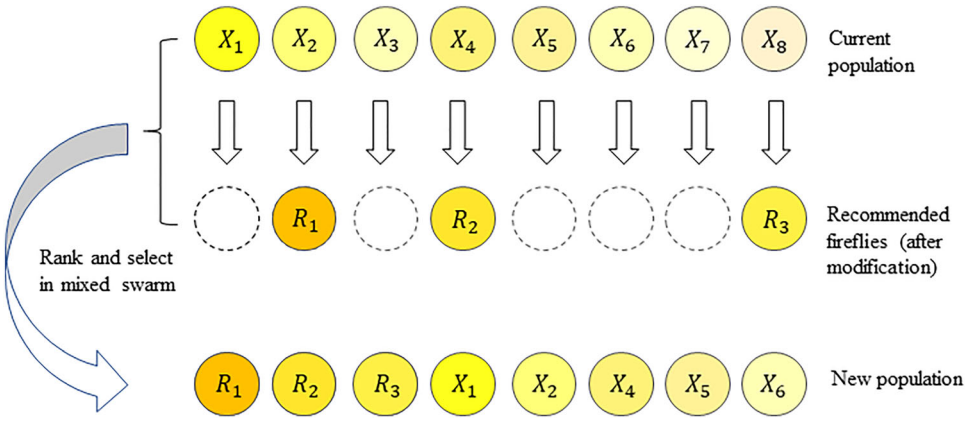


Fig. 5 Total process of the weighted voting mechanism

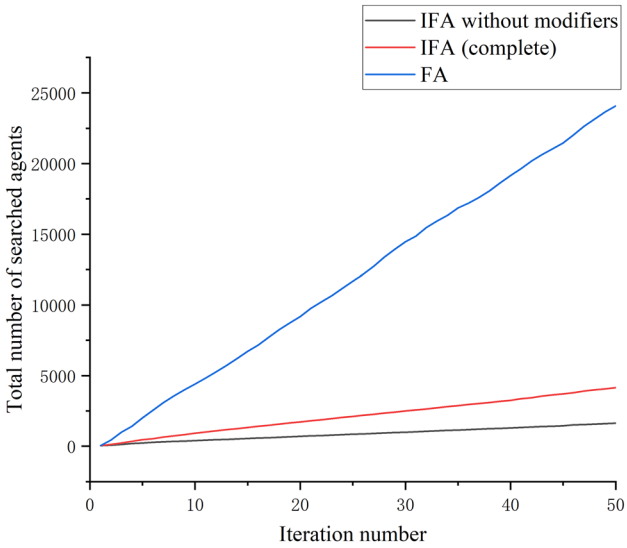


Fig. 6 Curves of the total number of searched agents for different algorithms

methods and the comparison of algorithm performance. All the experiments mentioned here are conducted on AMD Ryzen 5 5600H with Radeon Graphics and 16 GB of RAM. The programming language used in this article is Python with version 3.9.7.

4.1 Data description

In order to make the datasets as representative as possible, this paper selects 18 datasets from diverse sources with different numbers of features and samples for experiments. The details of the datasets used in the experiments are given in Table 1. Except for the Crab Age, Housing Prices, and Housing Prices datasets, which are from the Kaggle website, all other datasets are from the UCI data repository. For more details, please refer to the website of Kaggle (https://www.kaggle.com) and the UCI Repository (http://archive.ics.uci.edu/ml/index.php) [48].

The data preprocessing steps can greatly improve the accuracy of the algorithm in most situations. In this paper, the data preprocessing steps include removing obvious redundant features (such as the feature 'id'), filling missing values, data discretization and data normalization. Normalization is essential for the ReliefF algorithm.

4.2 Experiment settings

For one part of the experiments, we compared the traditional random selection algorithm with the initialization method proposed in this paper on 12 datasets. Each algorithm is conducted 50 times independently and for each time the algorithm generates a population of 32 fireflies. By comparing the fitness values of the populations generated by the two algorithms, the effectiveness of the proposed algorithm compared with the traditional random selection algorithm is verified.

For the other part of the experiments, a few of the algorithms from state-of-the-art related researches are selected as comparison algorithms which are listed as follows: MBPSO [49], bGWO [22], DbFA [40], bFA [40], BBHA [44], GOA [25], bWOA-S [50], GWO2 [47], PIO [51]. These comparison experiments are conducted on all 18 datasets. In order to make all algorithms perform well on the datasets, we set the parameters with reference to the recommendations of the original literatures. The parameters of the proposed method include the original parameters of FA and the newly added parameters. The original parameters of FA are set as the same as bFA, that is $\beta = 1$, $\gamma = 0.1$ and $\alpha = 0.1$. The sensitivity analysis of the newly added parameters is presented in 4.4. Except for the parameters mentioned above for this paper, all other parameters do not need to be set and specified manually.

Since the articles selected for comparison combined wrapper methods with heuristics algorithms, the setting of a classifier is necessary for all the methods. On the one hand, there is no uniform and standard classifier for

Table 1 Datasets used in the experiments

No	Dataset	Number of features	Number of instances
1	Housing Prices	12	545
2	Wine	13	178
3	Heart failure clinical records	13	299
4	Japanese Credit Screening	14	689
5	Zoo	17	101
6	Lymphography	18	148
7	Image Segmentation	19	2310
8	Mobile Price	20	2000
9	Anuran Calls (MFCCs)	22	7195
10	Parkinsons	23	197
11	Audit Data	24	776
12	Steel Plates Faults	32	1941
13	Dermatology	33	366
14	Chess (King-Rook vs. King-Pawn)	36	3196
15	QSAR biodegradation	41	1055
16	Divorce Predictors	54	170
17	Spambase	57	4601
18	Arrhythmia	279	452

wrapper feature selection methods in these studies. On the other hand, experiments need to ensure fairness to have application value and credibility. Therefore, we set the classifiers of all algorithms as KNN classifiers with the tenfold cross-validation method. As for the parameter K of the KNN classifier, it is set as 5 as suggested by most articles [3, 8, 22, 24]. The population of all algorithms is set to 32, the maximum number of iterations is set to 50, and every algorithm is independently run 30 times on each dataset.

4.3 Evaluation criteria

In this paper, three kinds of indicators are selected to examine the performance of the algorithms: the average accuracy, the fitness value (including the average, best and worst) and the average number of selected features.

Average accuracy: represents the average of the accuracy calculated by the classifier using the subset of selected features, which is formulated in Eq. (16). It is the most intuitive demonstration of the effect of the feature selection algorithm on improving the performance of the classifier.

$$AvgAccuracy = \frac{1}{T_{runs}} \sum_{j=1}^{T_{runs}} \frac{1}{s} \sum_{i=1}^{s} Match(C_i, L_i) \tag{16}$$

where T_{runs} is the number of runs for the algorithm to find the optimal solution, s is the number of dataset instances and *Match* is the function that outputs 1 when the predicted class C_i is the same as the actual class L_i and outputs 0 in the other situations.

Average fitness value: represents the average of the fitness value when the fitness function is defined as Eq. (6). In the research problem, a smaller fitness value means a better quality of the feature subset.

$$AvgFitness = \frac{1}{T_{runs}} \sum_{i=1}^{T_{runs}} f(X_*^i) \tag{17}$$

where $f(X_*^i)$ represents the fitness value of the optimal solution obtained by the algorithm at the i th time.

Worst fitness value: represents the maximum of the fitness values about the results for the T_{runs} times of the algorithm. The worst fitness value indicates a pessimistic situation and it can be defined as:

$$WorstFitness = \max_{i=1,2,\ldots,T_{runs}} (X_*^i) \tag{18}$$

Best fitness value: represents the minimum of the fitness values about the results for the T_{runs} times of the algorithm. The best fitness value indicates an optimistic situation and it can be defined as:

$$BestFitness = \min_{i=1,2,\ldots,T_{runs}} (X_*^i) \tag{19}$$

The average number of selected features: represents the average of the numbers of the features in the optimal solution feature subsets for the T_{runs} times.

$$AvgFeatureNumbers = \frac{1}{T_{runs}} \sum_{i=1}^{T_{runs}} \left| X_*^i \right| \tag{20}$$

where $\left| X_*^i \right|$ represents the number of the optimal solution feature subsets obtained by the algorithm at the i th time.

Standard deviation: represents the degree of deviation of the results for the T_{runs} times of the algorithm. The standard deviation measures the stability of the algorithm and can be formulated in Eq. (21)

$$Std = \sqrt{\frac{1}{T_{runs}-1}\sum_{i=1}^{T_{runs}}\left[f(X_*^i) - AvgFitness\right]^2} \tag{21}$$

4.4 Sensitivity analysis

IFA is mainly divided into three parts: the first part is the feature grouping method (FG-RC), the second part is improved movement behavior, and the third part is the weighted voting mechanism. Among them, the first part introduces two parameters k and r_1, and the third part also introduces two parameters $w_a(w_b = 1 - w_a)$ and r_2. The parameters mentioned above all play an important role, and this section aims to analyze the sensitivity of the algorithm to these four parameters. To ensure fairness, the experiments conducted in this section also use the same KNN classifier (with the tenfold cross-validation method) and fitness function settings as described previously.

According to the proposed FG-RC, the parameter k controls the number of initial groups. Based on the strategy of the algorithm, the number of groups that the features can be divided into is at least one and at most $d/2$, so the parameter k should be an integer value in $[2, d/2]$. r_1 is a parameter that provides randomness in the process of random selection for initialization using the grouped features. Considering the range of random numbers generated is $[0, 1]$, the parameter r_1 takes value in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}. Datasets including Housing Prices, Heart failure clinical records, Lymphography, Mobile Price, Parkinsons and Audit Data are used for the experiments of sensitivity analysis. By adjusting the values of k and r_1 at the same time, the initialization algorithm runs 20 times independently on each dataset, and the algorithm generates 32 individuals each time. For the convenience of observation, we draw it into a 3D surface plot as shown in Fig. 7, in which the height of each point represents the average fitness value of the fireflies generated by FG-RC using the corresponding parameter combination.

It can be seen from Fig. 7 that the r_1 value has an impressive effect on the performance of the algorithm. As the value of r_1 increases from 0.1 to 0.9, the average fitness of the population gradually increases, which means the quality of the population gradually deteriorates. All of the datasets show that the population obtained when r_1 is set as 0.9 is significantly inferior to the population with r_1 between 0.1 and 0.5. Among them, when r_1 is 0.1, the performance of the initialization algorithm is the most stable. From the perspective of the parameter k, when r_1 is fixed, the value of k has no significant influence on the improvement in the average fitness value. By general analysis, the algorithm with a smaller k has a higher degree of freedom and performs better. Taking the above analysis into consideration, the subsequent comparative experiments in this paper will set $r_1 = 0.1$ and $k = 3$.

For the weighted voting mechanism, the parameter w_a represents the importance of the brightness in Eq. (13), while w_b represents the importance of distance and $w_a + w_b = 1$. Another parameter r_2 controls the generation of the recommended locations. Both of w_a and r_2 are limited in [0,1] and rarely take the values of boundaries. Therefore, they are selected from {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} for experiments here. By adjusting the values of w_a and r_2 at the same time, the proposed algorithm runs 20 times independently on each dataset when the other parameters are constant. In the experimental study, it was found that because the algorithm can always converge to a better result, the solutions obtained after convergence are mostly similar and cannot be compared. Based on the analysis of the method, the parameters mainly affect the convergence speed of the algorithm in the early stage. Therefore, in the experiments of sensitivity analysis for this part, the maximum number of iterations is set to only 7 for research. For the convenience of observation, the experiment results are also shown in a 3D surface plot in Fig. 8. The height of each point represents the average optimal fitness value obtained by the algorithm for each parameter combination.

It can be seen from Fig. 8 that the parameters do not affect the algorithm regularly due to the randomness. However, the quality of solution decreases when r_2 increases closely to 0.9, which is shown on Housing Price, Heart failure clinical record, Audit Data and Mobile Price datasets. Although the distribution of results seems uneven, it is still can be seen that when r_2 is between 0.3 and 0.6 as well as w_a is between 0.2 and 0.7, the algorithm is more likely to obtain better performance. Therefore, the subsequent comparative experiments in this paper will set $r_2 = 0.3$ and $w_a = 0.6$.

4.5 Numerical results and discussion

4.5.1 Comparison of the proposed initialization algorithm and random initialization

Figure 9 displays the boxplots of the proposed initialization method (FG-RC) and the random selection (RS), which shows the average fitness value of the population each time. As can be seen from Fig. 9, FG-RC is better than the commonly used RS initialization method completely on most datasets. The proposed FG-RC algorithm can more directly and effectively generate an initial

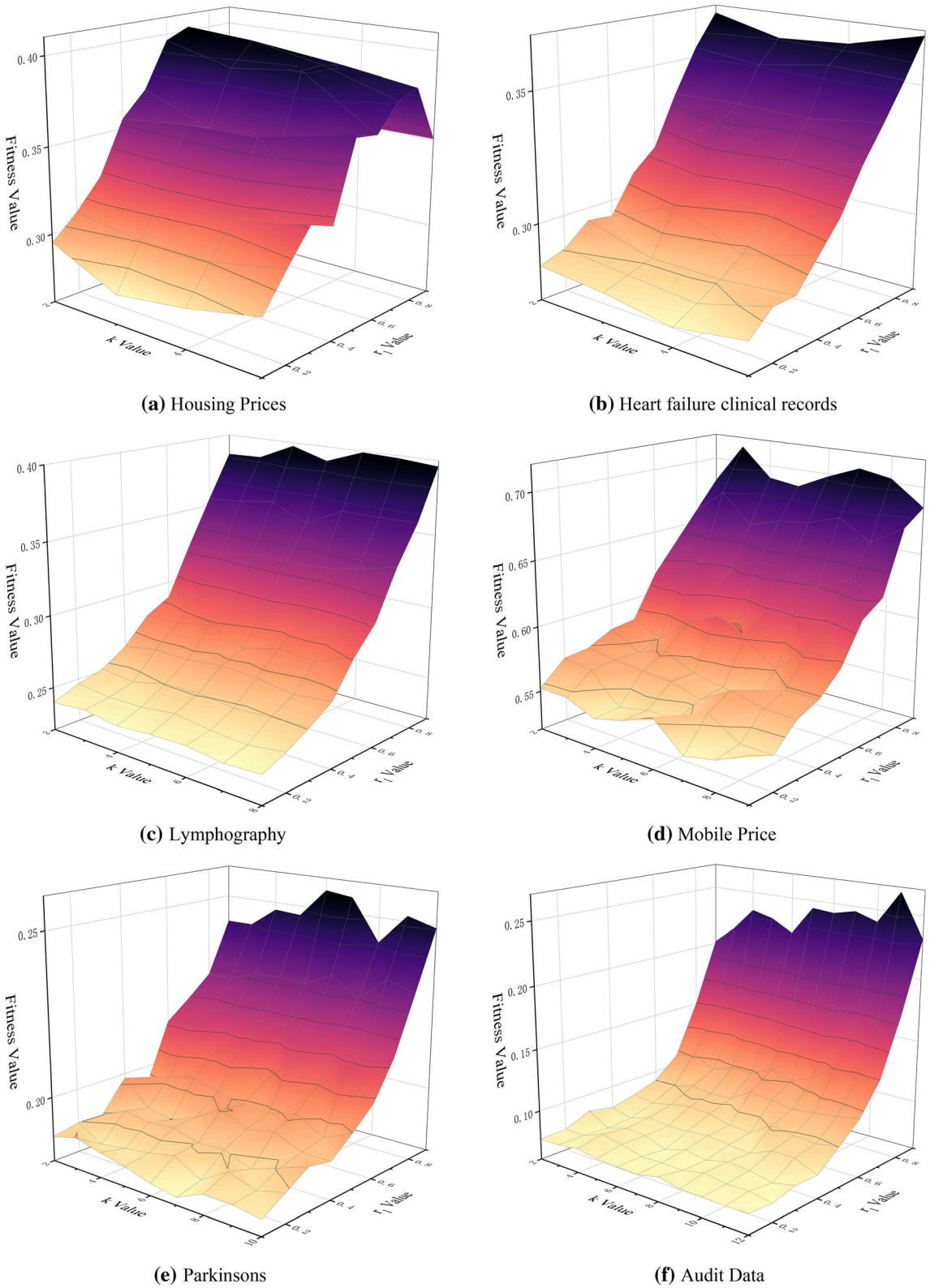


Fig. 7 Average fitness value when adjusting k and r_1 values

population with better performance for the feature selection problem. In detail, the median of the average values of the proposed method is significantly better than that of RS on most datasets. In addition, the maximum value of FG-RC

does not even exceed that of RS method, which is shown on the Wine, Heart failure clinical records, Japanese Credit Screening, Zoo and so on. It is worth noting that the minimum value of FG-RC breaks through that of RS, indicating that the population generated is closer to the global optimal solution and potentially contributes to the subsequent procedures. Although FG-RC performs better than RS on the datasets of QSAR biodegradation and Divorce Predictors, the superiority is not obvious because of the difference in maximum values between the two methods. We believe that this superiority is still meaningful, but some large deviations show the proposed method may not be stable enough.

For the Arrhythmia dataset, the performance of FG-RC is similar to RS and slightly better. Other experiments show that the best-performing feature subset for this dataset is hard to find at first due to the large number of features, which may be one of the reasons to account for this situation. On the other hand, the ReliefF algorithm used in this paper is more dependent on the sampling process. That is to say, the distribution of instances in this dataset may result in bias between the samplings, which should also be considered as a reason.

4.5.2 Comparison of the proposed algorithm and comparison algorithms

Table 2 presents the average fitness values of the proposed algorithm and comparison algorithms. It can be seen from the table that IFA is better than other comparison algorithms on all test datasets. The average fitness value of IFA is significantly smaller than that of other algorithms, which shows its excellent optimization ability. It should be noted that the bold number in Table 3 means the maximum value among the algorithms while the meaning is the minimum value in Tables 2, 4, 5, 6, 7.

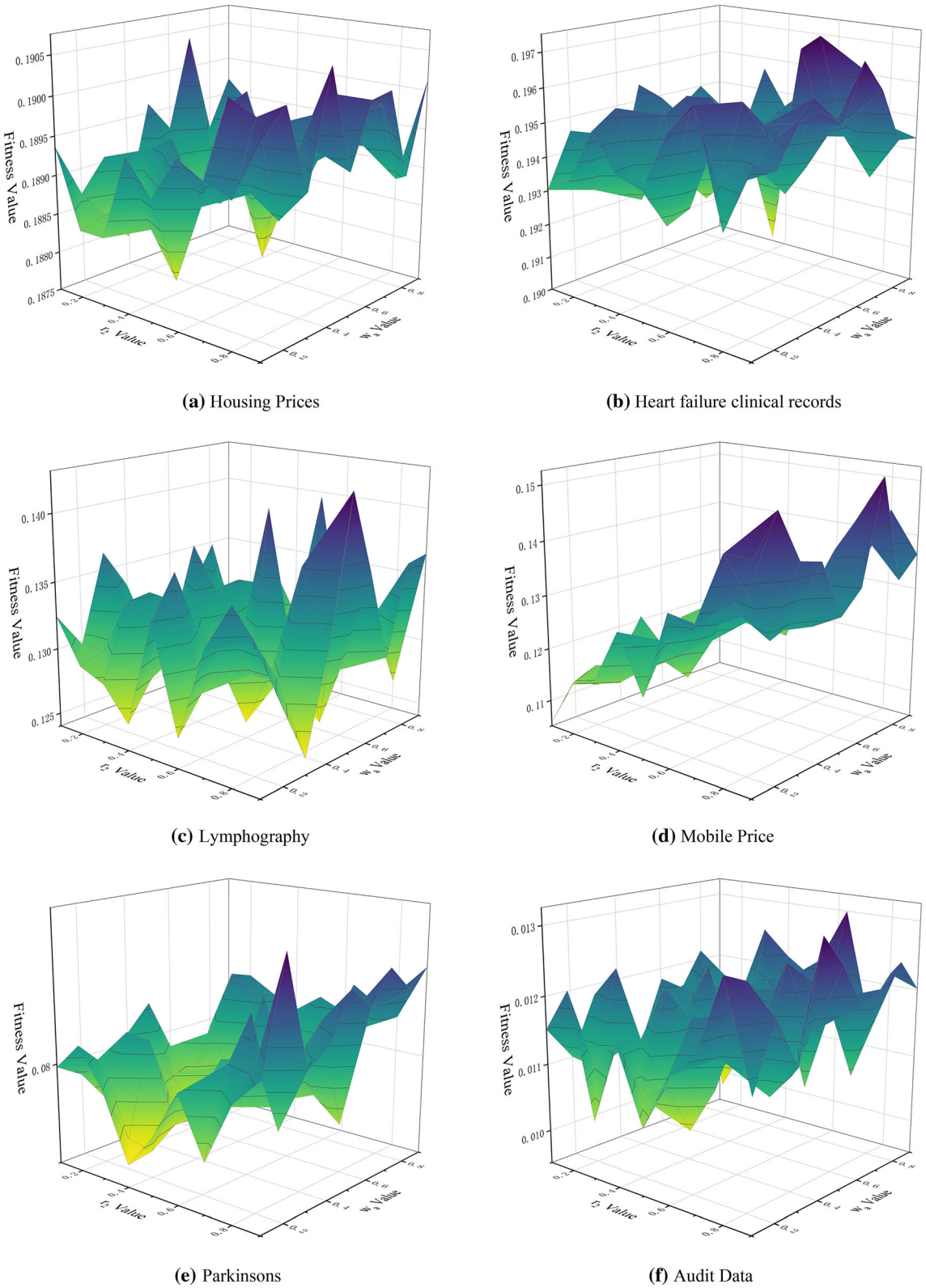
Similar results can be observed in the other tables. Tables 4 and 5 show the worst and the best fitness values over all the runs. In Table 4, the worst fitness obtained by IFA is still the smallest among the comparison algorithms on all datasets except the Divorce Predictors dataset. Table 5 shows that IFA can always find the optimal solution within the number of experimental tests. Other algorithms such as MBPSO and bGWO have shown similar capabilities on the Housing Prices dataset and Wine dataset, which may due to the smaller size of the datasets. GWO2 and PIO algorithms can also find the optimal solution on other larger datasets. However, only IFA shows the ability to find the optimal solution on each dataset. In view of the above points, the superiority of IFA proposed in this paper can be proved.

Tables 3 and 6 show the average classification accuracy and the average number of selected features of the algorithms. Combined with Tables 2, 3, and 6, there is an obvious phenomenon that even if IFA achieved the minimum of the average fitness values among all the compared algorithms, it is hard to get the highest classification accuracy or the smallest feature subset at the same time on some datasets. We suspect that the guidance of the fitness function may lead to deviation to some extent, which can be one of the reasons that explain this phenomenon. This may give us a direction that the fitness function still has much room for improvement. Both the higher accuracy and the lower number of features are crucial, so the algorithm should keep a balance rather than sacrifice one for another.

Table 7 shows the standard deviation of the algorithms. It can be seen that IFA is generally stable on most datasets. However, if an algorithm gets trapped in the same local optimal solution each time, the results of it will show more stable results with a poor performance. For example, the standard deviation of DbFA even get 0 on some datasets, but it performs worse than IFA. Although a lower deviation value is important, focusing too much on it may result in overlooking the essential performance, which deviates from the objective of the problem. Therefore, the standard deviation value of the proposed algorithm is still satisfactory.

From the dataset's point of view, it seems that the feature selection problem for datasets with fewer features is much easier. For the Housing Prices dataset, all the algorithms used here can find the optimal solution in 30 times except bGWO and GWO2. It also shows that there is more than one algorithm that can achieve the best solution on datasets with no more than 30 features. Consider the Arrhythmia dataset, there are more than 200 features and it is hard to find the best subset of the features. IFA achieves only 30.9 features with the highest average fitness value while the other algorithms get a number of features selected more than 100. That is to say, IFA can successfully obtain the required accuracy, but other algorithms cannot jump out of the local optimal values.

In order to compare the convergence ability of the proposed algorithm and other algorithms, the convergence curves of the above ten algorithms on all the datasets for 50 iterations are drawn in Fig. 10. The iterative curves show that IFA converges at a fast speed during operation to the vicinity of the optimal solution for searching. Meanwhile, the comparison algorithms are easy to be trapped in the local optimal solution sometimes. The initial solution of IFA is often better than random selection, which can help the algorithm determine the convergence range quickly and search for the optimal solution. During the iterative process, the improved strategy introduced by this paper can help the algorithm jump out of the local optimal area, obtain the optimal solution with higher convergence accuracy and then improve the performance of the



(a) Housing Prices

(b) Heart failure clinical records

(c) Lymphography

(d) Mobile Price

(e) Parkinsons

(f) Audit Data

Fig. 8 Average fitness value when adjusting w_a and r_2 values

algorithm. Compared with the other swarm intelligence optimization algorithms, the effectiveness of the IFA is verified.

Wilcoxon test is a nonparametric statistical test used primarily to calculate the difference between two paired groups to determine whether there is a statistical difference between them. However, the more statistical tests are performed, the more likely false positives are generated. One of the strategies to address this problem is the correction. The correction method used here is the "BH" [52] method based on the R language tool. The adjusted p-values of the Wilcoxon test attained for the comparison of the proposed method and the compared algorithms are reported in Table 8. From Table 8, the p values are below 0.05 for the majority of cases, which confirms that IFA is significantly different from the compared algorithms on the majority of the datasets.

Overall, the algorithm proposed in this paper significantly outperforms other compared algorithms on most datasets. The average fitness value of IFA is the lowest which means the method can find the most suitable feature subset to obtain the highest accuracy. However,

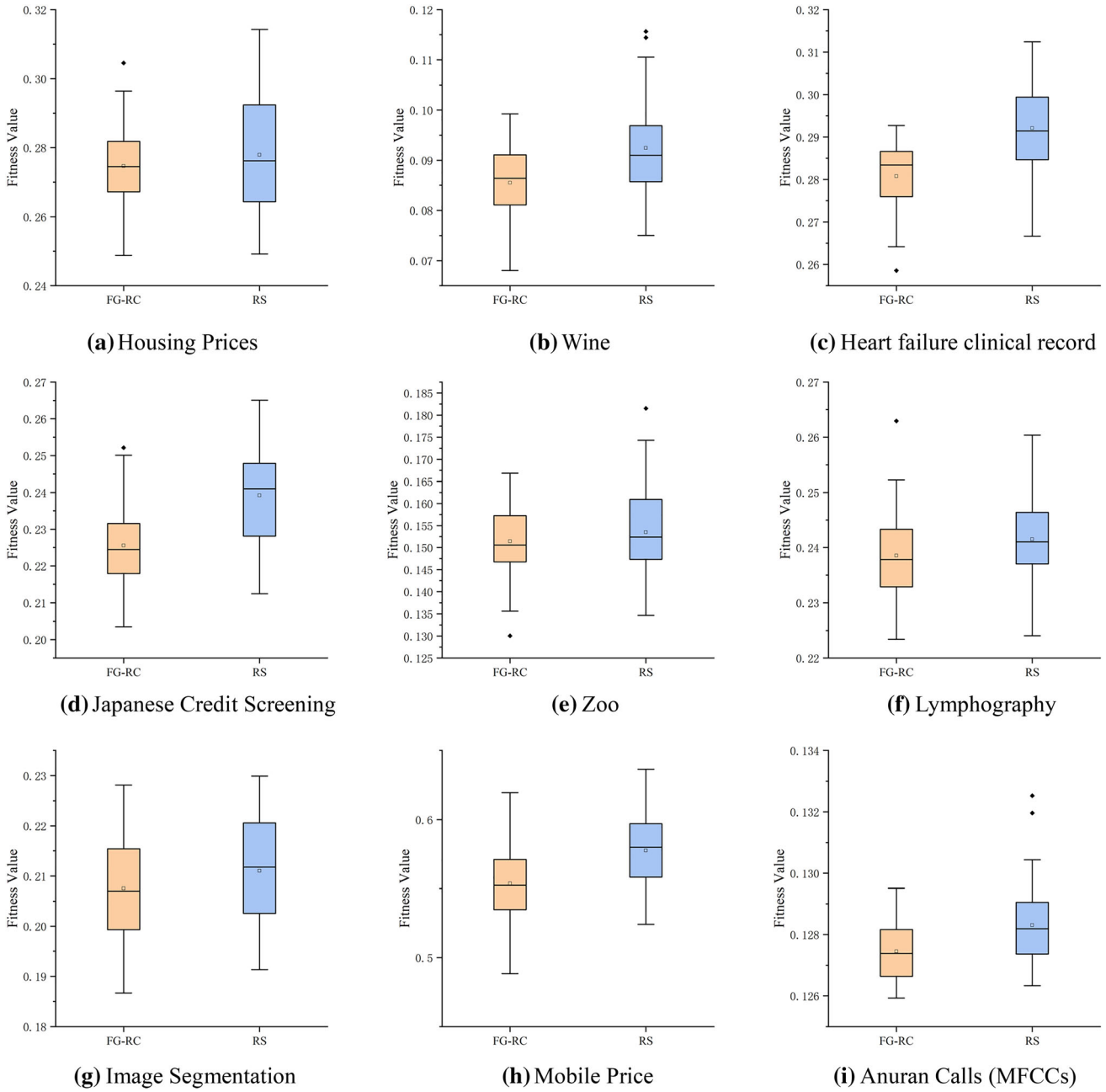


Fig. 9 Boxplots of the proposed initialization method and the random selection

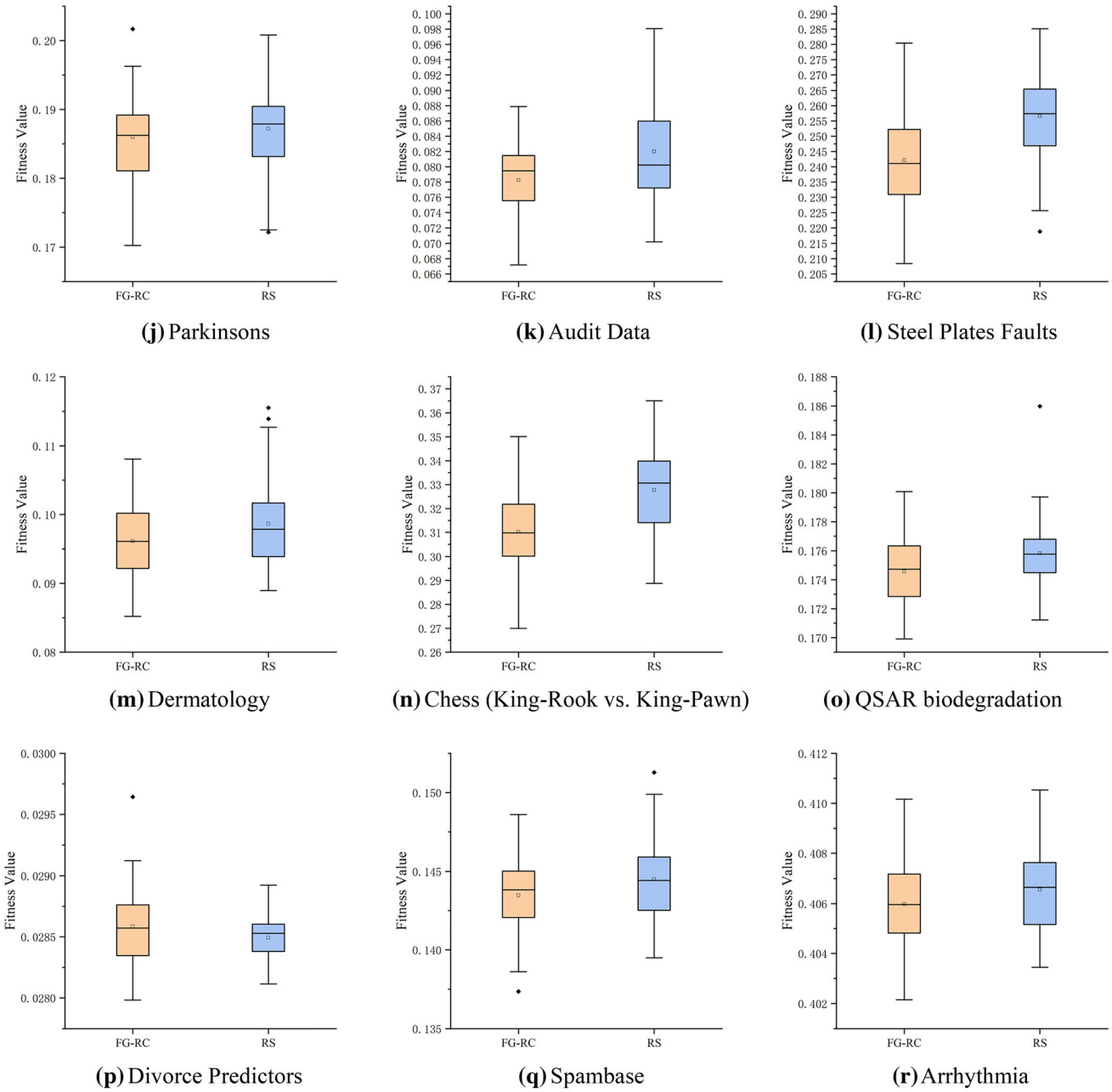


Fig. 9 continued

experiments show that there may be room for improvement in the design of the fitness function. In addition, the stability of the algorithm in this paper still needs to be improved and researched.

4.5.3 Time complexity

To evaluate the running efficiency of the algorithm proposed in this paper, the time complexity of IFA is discussed and analyzed in this section. As mentioned above, IFA is mainly divided into three parts, including the feature grouping method (FG-RC), improved movement behavior and the weighted voting mechanism. For the first part, it can be divided into three modules, respectively: sorting based on the ReliefF algorithm, feature grouping and generating new populations. The running time of ReliefF linearly depends on the size of dataset s and the number of features d, which can be written as $O(sd)$ [11]. The feature grouping module is only linearly related to d, and the last module is actually related to d and the population size N. In summary, the time complexity of FG-RC can be written as $O((s+1+N)\times d)$. The second part uses only one layer

Table 2 Comparison in terms of average fitness (Bold numbers indicate the minimum values)

Dataset	MBPSO	bGWO	DbFA	bFA	BBHA	GOA	bWOA-S	GWO2	PIO	IFA
Housing Prices	0.1885	0.1929	0.1855	0.1889	0.1895	0.1887	0.1887	0.1937	0.1892	**0.1854**
Wine	0.0172	0.0197	0.0154	0.0153	0.0177	0.0182	0.0167	0.0193	0.0170	**0.0152**
Heart failure clinical records	0.1972	0.20	0.1944	0.1944	0.1976	0.1976	0.1929	0.2041	0.1929	**0.1912**
Japanese Credit Screening	0.1420	0.1451	0.1355	0.1344	0.1389	0.1386	0.1414	0.1431	0.1411	**0.1324**
Zoo	0.0467	0.0557	0.0416	0.0401	0.0479	0.0471	0.0443	0.0549	0.0411	**0.0394**
Lymphography	0.1333	0.1356	0.1288	0.1265	0.1375	0.1357	0.1366	0.1361	0.1353	**0.1260**
Image Segmentation	0.0913	0.0951	0.0857	0.0862	0.0904	0.0912	0.0901	0.0877	0.0861	**0.0827**
Mobile Price	0.220	0.1732	0.1498	0.1399	0.1879	0.1870	0.1783	0.1372	**0.1053**	**0.1035**
Anuran Calls (MFCCs)	0.1034	0.1076	0.0995	0.1015	0.1024	0.1027	0.1005	0.1008	0.0941	**0.0922**
Parkinsons	0.0946	0.0975	0.0746	0.0769	0.1055	0.1066	0.0867	0.0851	0.0672	**0.0609**
Audit Data	0.0197	0.0244	0.0114	0.0103	0.0143	0.0148	0.0142	0.0075	0.0052	**0.0050**
Steel Plates Faults	0.0332	0.0372	0.0332	0.0324	0.0333	0.0332	0.0332	0.0305	0.0304	**0.0301**
Dermatology	0.0253	0.0237	0.0226	0.0209	0.0176	0.0266	0.0233	0.0242	0.0199	**0.0177**
Chess (King-Rook vs. King-Pawn)	0.0798	0.0779	0.0744	0.0708	0.0817	0.0827	0.0751	0.0736	0.0625	**0.0463**
QSAR biodegradation	0.1410	0.1382	0.1383	0.1371	0.1419	0.1421	0.1403	0.1381	0.1325	**0.1291**
Divorce Predictors	0.0229	0.0235	0.0209	0.0210	0.0212	0.0212	0.0212	0.01760	0.01744	**0.01707**
Spambase	0.1009	0.0960	0.1005	0.0959	0.0146	0.0813	0.0997	0.0913	0.0865	**0.0140**
Arrhythmia	0.3689	0.3521	0.3558	0.3548	0.3573	0.3572	0.3581	0.3331	0.3378	**0.2998**

Table 3 Comparison in terms of average classification accuracy (Bold numbers indicate the maximum values)

Dataset	MBPSO	bGWO	DbFA	bFA	BBHA	GOA	bWOA-S	GWO2	PIO	IFA
Housing Prices	0.8154	0.8121	0.8168	0.8164	0.8142	0.8147	0.8151	0.8106	0.8146	**0.8169**
Wine	0.9886	0.9860	0.9886	0.9885	0.9876	0.9871	**0.9888**	0.9860	0.9882	0.9886
Heart failure clinical records	0.8051	0.8026	0.8122	0.8111	0.8043	0.8043	0.8094	0.7981	0.8094	**0.8123**
Japanese Credit Screening	0.8627	0.8605	0.8663	0.8679	0.8638	0.8639	0.8614	0.8603	0.8622	**0.8701**
Zoo	0.9578	0.9496	0.9626	0.9640	0.9564	0.9576	0.9607	0.9495	0.9632	**0.9643**
Lymphography	0.8715	0.8698	0.8753	0.8776	0.8664	0.8683	0.8731	0.8681	0.8712	**0.8779**
Image Segmentation	0.9131	0.9103	0.9174	0.9169	0.9131	0.9123	0.9141	0.9151	0.9174	**0.9203**
Mobile Price	0.7812	0.8279	0.8512	0.8610	0.8130	0.8139	0.8227	0.8631	0.8955	**0.8975**
Anuran Calls (MFCCs)	0.9004	0.8970	0.9033	0.9015	0.9006	0.9006	0.9028	0.9015	0.9086	**0.9103**
Parkinsons	0.9092	0.9070	0.9274	0.9252	0.8972	0.8963	0.9166	0.9176	0.9349	**0.9408**
Audit Data	0.9844	0.9799	0.9912	0.9918	0.9886	0.9883	0.9886	0.9933	0.9957	**0.9960**
Steel Plates Faults	0.9716	0.9678	0.9716	0.9716	0.9714	0.9713	0.9716	0.9714	0.9716	**0.9717**
Dermatology	0.9808	0.9828	0.9826	0.9842	0.9877	0.9785	0.9827	0.9817	0.9848	**0.9870**
Chess (King-Rook vs. King-Pawn)	0.9247	0.9281	0.9298	0.9333	0.9223	0.9214	0.9295	0.9302	0.9412	**0.9578**
QSAR biodegradation	0.8637	0.8674	0.8650	0.8663	0.8617	0.8615	0.8641	0.8659	0.8709	**0.8746**
Divorce Predictors	0.9817	0.9794	0.9823	0.9823	0.9823	0.9823	0.9821	**0.9833**	0.9835	**0.9833**
Spambase	0.9040	0.9103	0.9033	0.9083	0.9884	0.9223	0.9045	0.9112	0.9172	**0.9889**
Arrhythmia	0.6333	0.6517	0.6454	0.6463	0.6439	0.6440	0.6436	0.6683	0.6636	**0.6983**

of loops to move the fireflies toward the optimal solution fireflies, so the time complexity can be written as $O(N)$. The third part still retains the double layer loop of the classical FA, but the execution contents inside and outside the loop of these two algorithms are different. Besides, the part where IFA adds the recommended positions and their modifications is related to the randomness of the algorithm, but it is still considered as the linear complexity of N

Table 4 Comparison in terms of the worst fitness (Bold numbers indicate the minimum values)

Dataset	MBPSO	bGWO	DbFA	bFA	BBHA	GOA	bWOA-S	GWO2	PIO	IFA
Housing Prices	0.1905	0.2008	0.1889	0.1889	0.1967	0.1932	0.1920	0.2104	0.1950	**0.1854**
Wine	0.0203	0.0285	**0.0164**	0.0171	0.0219	0.0226	0.0182	0.0258	0.0219	**0.0164**
Heart failure clinical records	0.2049	0.2172	0.1944	0.1944	0.2035	0.2044	0.2039	0.2263	0.2039	**0.1932**
Japanese Credit Screening	0.1457	0.1502	0.1401	0.1409	0.1460	0.1449	0.1470	0.1471	0.1470	**0.1398**
Zoo	0.0548	0.0832	0.0523	0.0452	0.0544	0.0572	0.0558	0.0775	0.0504	**0.0439**
Lymphography	0.1399	0.1487	0.1344	0.1335	0.1491	0.1464	0.1385	0.1556	0.1445	**0.1311**
Image Segmentation	0.0948	0.1178	0.0937	0.0937	0.0995	0.0964	0.0948	0.1047	0.0964	**0.0854**
Mobile Price	0.2881	0.2896	0.1797	0.1624	0.2411	0.2267	0.2233	0.2020	0.1252	**0.1035**
Anuran Calls (MFCCs)	0.1078	0.1166	0.1014	0.1028	0.1081	0.1074	0.1051	0.1116	0.1003	**0.0985**
Parkinsons	0.1029	0.1272	0.0855	0.0850	0.1173	0.1172	0.1032	0.1083	0.1078	**0.0692**
Audit Data	0.0283	0.0383	0.0137	0.0122	0.0197	0.0220	0.0222	0.0313	0.0113	**0.0096**
Steel Plates Faults	0.0341	0.0466	0.0341	0.0332	0.0358	0.0352	0.0341	0.0319	0.0314	**0.0311**
Dermatology	0.0301	0.0288	0.0254	0.0245	0.0258	0.0305	0.0261	0.0311	0.0259	**0.0213**
Chess (King-Rook vs. King-Pawn)	0.0911	0.1103	0.0865	0.0763	0.0973	0.0948	0.0858	0.1026	0.0853	**0.0546**
QSAR biodegradation	0.1453	0.1491	0.1398	0.1421	0.1461	0.1470	0.1441	0.1513	0.1387	**0.1372**
Divorce Predictors	0.0269	0.0266	0.0213	0.0211	0.0222	0.0224	0.0258	**0.0191**	0.0229	0.0238
Spambase	0.1072	0.1026	0.1013	0.0971	0.0192	0.0865	0.1047	0.1017	0.1002	**0.0184**
Arrhythmia	0.3741	0.3622	0.3625	0.3612	0.3650	0.3647	0.3668	0.3495	0.3592	**0.3470**

Table 5 Comparison in terms of the best fitness (Bold numbers indicate the minimum values)

Dataset	MBPSO	bGWO	DbFA	bFA	BBHA	GOA	bWOA-S	GWO2	PIO	IFA
Housing Prices	**0.1854**	**0.1854**	**0.1854**	**0.1854**	0.1869	0.1869	0.1869	0.1888	0.1869	**0.1854**
Wine	**0.0152**	0.0164	**0.0152**	**0.0152**	**0.0152**	**0.0152**	**0.0152**	0.0164	**0.0152**	**0.0152**
Heart failure clinical records	**0.190**	**0.190**	0.1944	0.1944	0.1944	0.1944	**0.190**	**0.190**	**0.190**	**0.190**
Japanese Credit Screening	0.1345	0.1409	**0.1279**	0.1281	0.1281	0.1281	**0.1279**	0.1302	**0.1279**	**0.1279**
Zoo	0.0427	0.0443	0.0331	0.0331	0.0331	0.0324	**0.0318**	0.0324	**0.0318**	**0.0318**
Lymphography	0.1257	0.1258	0.1250	0.1250	0.1250	0.1256	0.1252	0.1256	0.1256	**0.1194**
Image Segmentation	0.0848	0.0843	**0.0775**	**0.0775**	0.0785	0.0854	0.0785	**0.0775**	**0.0775**	**0.0775**
Mobile Price	0.1599	**0.1035**	0.1208	0.1213	0.1208	0.1416	0.1346	**0.1035**	**0.1035**	**0.1035**
Anuran Calls (MFCCs)	0.0942	0.0995	0.0969	0.1007	0.0934	0.0966	0.0940	0.0921	**0.0894**	**0.0894**
Parkinsons	0.0857	0.0796	0.0595	0.0692	0.0898	0.0953	0.0640	**0.0586**	**0.0586**	**0.0586**
Audit Data	0.0137	0.0067	0.0074	0.0050	0.0080	0.0092	0.0054	**0.0017**	**0.0017**	**0.0017**
Steel Plates Faults	0.0323	0.0320	0.0317	0.0317	0.0320	0.0314	0.0320	**0.0298**	**0.0298**	**0.0298**
Dermatology	0.0203	0.0173	0.0191	0.0153	0.0152	0.0184	0.0192	0.0192	0.0184	**0.0137**
Chess (King-Rook vs. King-Pawn)	0.0579	0.0581	0.0627	0.0638	0.0601	0.0663	0.0632	0.0524	0.0494	**0.0382**
QSAR biodegradation	0.1364	0.1311	0.1375	0.1314	0.1328	0.1335	0.1356	0.1284	0.1215	**0.1213**
Divorce Predictors	0.0208	0.0195	0.0202	0.0209	0.0204	0.0196	0.0198	0.0122	0.0124	**0.0063**
Spambase	0.0907	0.0849	0.1001	0.0944	0.0105	0.0762	0.0921	0.0769	0.0782	**0.0103**
Arrhythmia	0.3650	0.3421	0.3459	0.3441	0.3485	0.3461	0.3405	0.3202	0.3213	**0.2760**

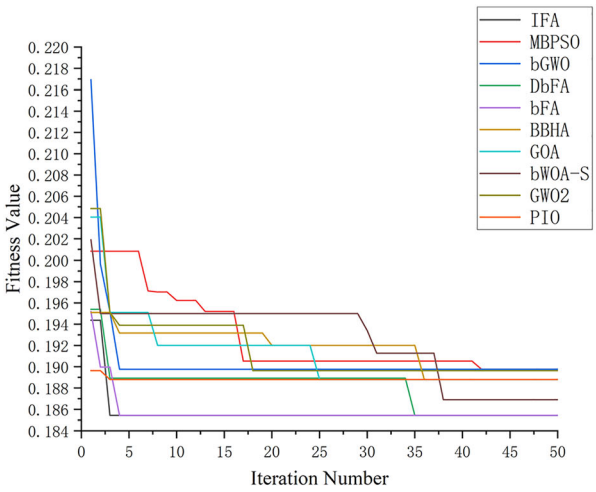
theoretically. To summarize, the time complexity of this part is noted as $O(N^2 + N)$. Therefore, the time complexity of IFA proposed in this paper can be written as $O((s + 1 + N) \times d + N^2 + 2N)$, which can be abbreviated as $O(sd + Nd + N^2)$. That is, the time complexity of the algorithm is related to the size of the dataset and the

Table 6 Comparison in terms of average number of selected features (Bold numbers indicate the minimum values)

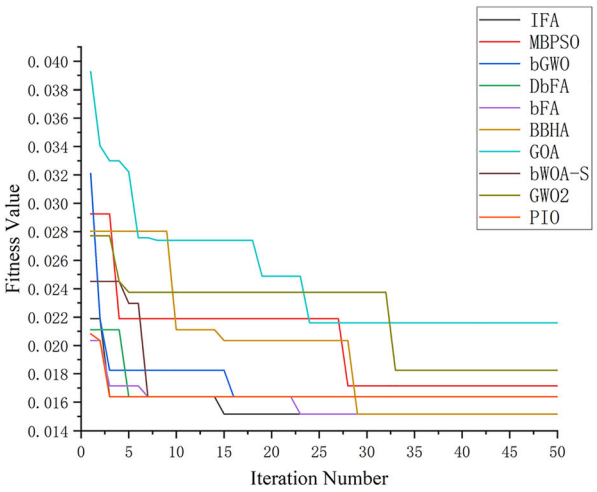
Dataset	MBPSO	bGWO	DbFA	bFA	BBHA	GOA	bWOA-S	GWO2	PIO	IFA
Housing Prices	7.20	8.2333	5.06	5.50	6.7333	5.80	6.80	7.50	6.80	**5.0**
Wine	9.80	7.6667	5.3333	5.2333	7.10	6.2333	7.3667	7.2333	7.0333	**5.0667**
Heart failure clinical records	6.90	5.4333	5.0	5.0	**4.7333**	5.8667	5.1333	5.0333	5.0333	4.9333
Japanese Credit Screening	10.0333	10.60	6.60	6.6333	6.3333	7.6333	6.50	7.40	7.1666	**5.7667**
Zoo	9.50	9.5333	7.8667	**6.6**	7.8333	8.2333	8.6667	8.0333	7.70	7.60
Lymphography	11.2667	12.0667	9.5333	10.2333	9.5333	8.8667	10.9667	10.1667	8.9667	**8.9333**
Image Segmentation	11.7333	12.1667	8.4667	9.2667	8.50	9.40	9.60	6.8333	8.4333	**7.2667**
Mobile Price	11.60	5.70	4.8333	4.5667	5.5333	10.40	5.6333	**3.40**	4.0	4.0
Anuran Calls (MFCCs)	13.40	12.40	8.6667	9.0	9.0667	10.40	9.7667	**7.60**	8.1667	7.6333
Parkinsons	12.7667	11.90	6.0667	6.2667	8.10	11.10	9.0333	7.7333	6.2333	**5.3333**
Audit Data	15.40	11.80	7.0	5.60	7.7333	12.7667	7.70	**2.30**	2.3667	2.6333
Steel Plates Faults	20.90	17.9333	15.20	14.9667	16.80	17.3333	15.5667	8.2333	6.60	**6.0**
Dermatology	21.8667	22.90	18.0667	18.2667	**7.0333**	17.3333	20.90	20.80	16.50	16.70
Chess (King-Rook vs. King-Pawn)	21.80	24.4333	17.6667	18.80	17.60	17.50	19.4667	16.4667	16.8333	**15.5667**
QSAR biodegradation	27.1667	28.90	**19.3333**	19.80	20.7333	20.4667	24.1667	21.9667	19.8667	20.5667
Divorce Predictors	32.1667	16.90	19.40	19.3333	20.5333	27.70	19.3667	5.9667	6.0667	**5.1**
Spambase	35.4667	41.60	27.6667	28.0	**8.30**	25.6333	29.5667	19.80	26.40	13.2333
Arrhythmia	172.9667	203.3333	134.50	132.7333	135.6667	138.3333	146.10	131.4333	134.0333	**30.90**

Table 7 Comparison in terms of the standard deviation (Bold numbers indicate the minimum values)

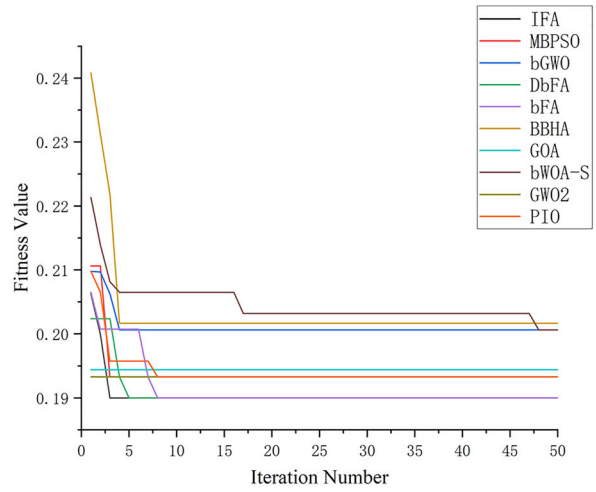
Dataset	MBPSO	bGWO	DbFA	bFA	BBHA	GOA	bWOA-S	GWO2	PIO	IFA
Housing Prices	0.0017	0.0039	**0.0**	**0.0**	0.0023	0.0018	0.0012	0.0051	0.0019	0.0014
Wine	0.0011	0.0038	0.0004	0.0004	0.0018	0.0024	0.0007	0.0027	0.0017	**0.0002**
Heart failure clinical records	0.0043	0.0078	**0.0**	**0.0**	0.0031	0.0030	0.0045	0.0082	0.0038	0.0016
Japanese Credit Screening	**0.0024**	0.0026	0.0038	0.0056	0.0049	0.0051	0.0038	0.0039	0.0061	0.0036
Zoo	0.0045	0.0088	**0.0044**	0.0046	0.0064	0.0058	0.0060	0.0115	0.0069	0.0045
Lymphography	0.0041	0.0069	0.0031	0.0015	0.0058	0.0054	0.0035	0.0069	0.0035	**0.0017**
Image Segmentation	**0.0028**	0.0055	0.0046	0.0037	0.0051	0.0029	0.0029	0.0063	0.0037	0.0031
Mobile Price	0.0301	0.0448	0.0174	0.0089	0.0305	0.0219	0.0202	0.0307	0.0057	**0.0**
Anuran Calls (MFCCs)	0.0031	0.0047	0.0019	**0.0009**	0.0032	0.0024	0.0033	0.0051	0.0034	0.0027
Parkinsons	0.0047	0.0128	0.0065	0.0039	0.0060	0.0057	0.0076	0.0153	0.0143	**0.0034**
Audit Data	0.0036	0.0072	**0.0013**	0.0016	0.0027	0.0027	0.0029	0.0083	0.0028	0.0029
Steel Plates Faults	0.0004	0.0048	0.0005	0.0004	0.0008	0.0008	0.0005	0.0005	**0.0002**	0.0003
Dermatology	0.0020	0.0032	0.0018	0.0022	0.0024	0.0032	0.0018	0.0033	0.0026	**0.0016**
Chess (King-Rook vs. King-Pawn)	0.0074	0.0131	0.0058	0.0044	0.0085	0.0078	0.0068	0.0147	0.0099	**0.0042**
QSAR biodegradation	0.0027	0.0043	**0.0010**	0.0026	0.0029	0.0027	0.0019	0.0053	0.0041	0.0043
Divorce Predictors	0.0014	0.0026	**0.0003**	0.0008	0.0004	0.0005	0.0010	0.0021	0.0016	0.0033
Spambase	0.0041	0.0045	**0.0005**	0.0011	0.0021	0.0021	0.0031	0.0057	0.0061	0.0021
Arrhythmia	0.0026	0.0056	0.0038	**0.0040**	**0.0040**	0.0050	0.0052	0.0166	0.0101	0.0068



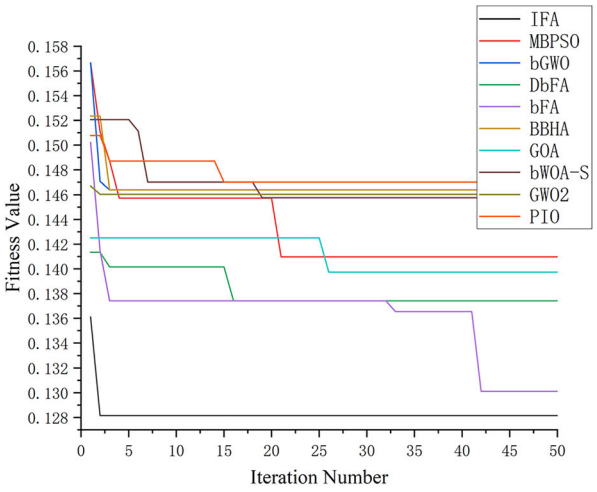
(a) Housing Prices



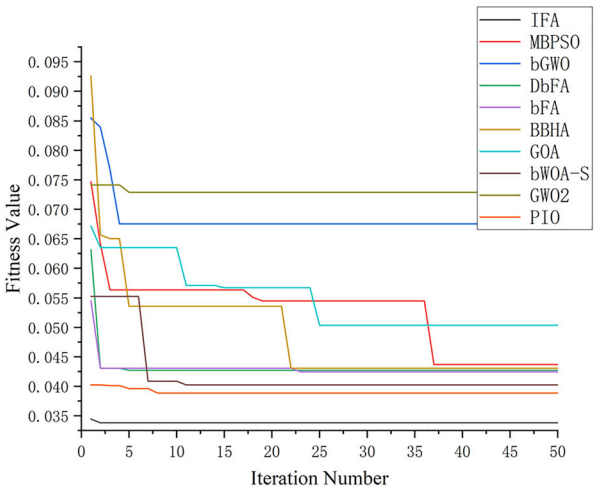
(b) Wine



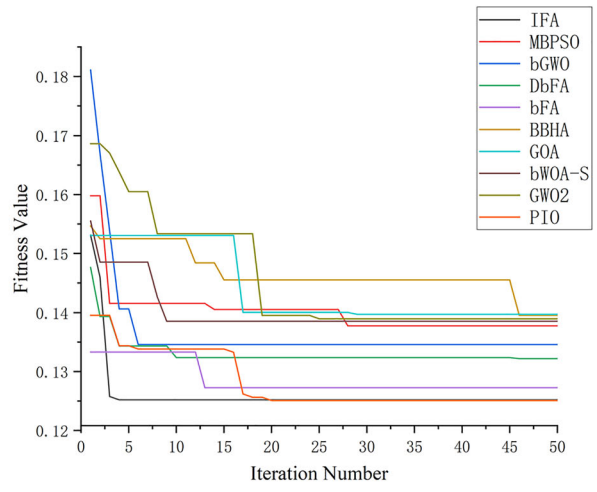
(c) Wine



(d) Japanese Credit Screening

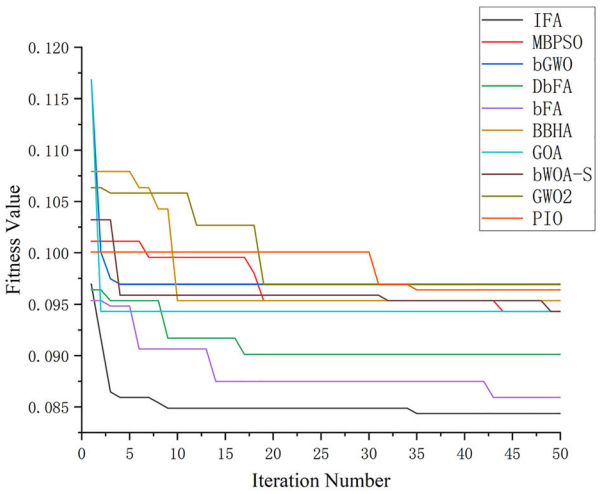


(e) Zoo

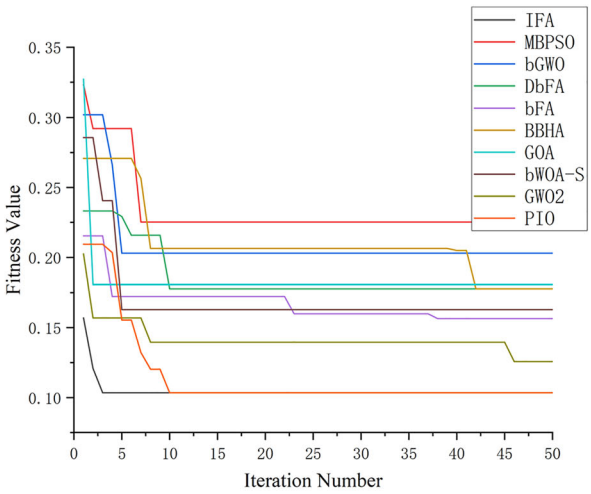


(f) Lymphography

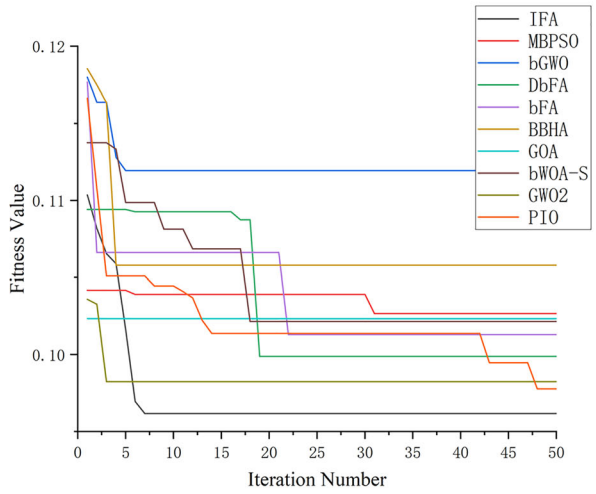
Fig. 10 Iterative curves for different datasets



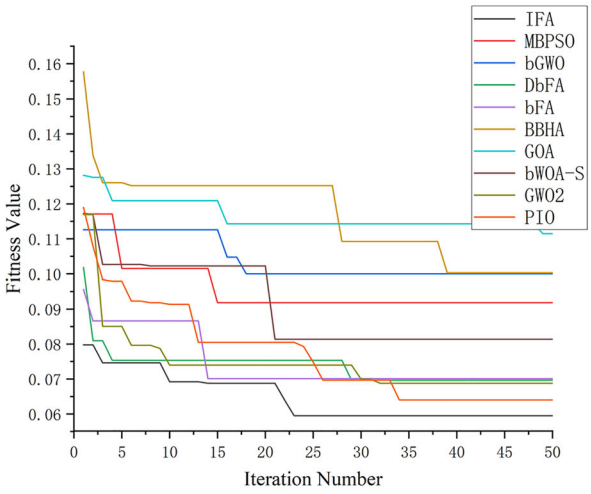
(g) Image Segmentation



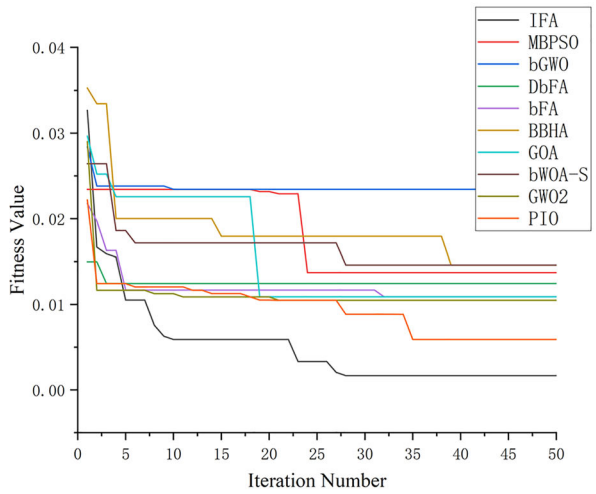
(h) Mobile Price



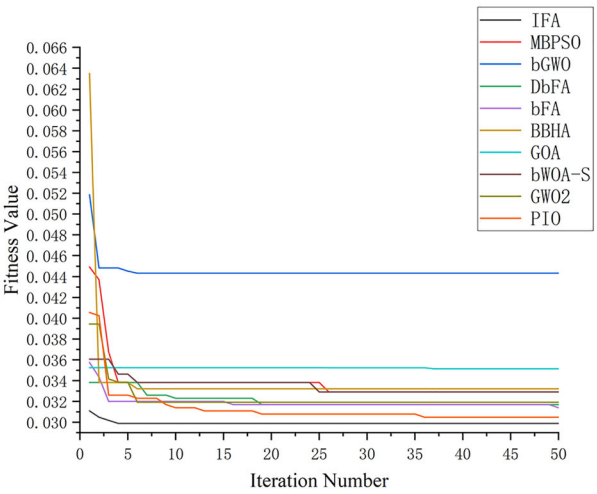
(i) Anuran Calls (MFCCs)



(j) Parkinsons

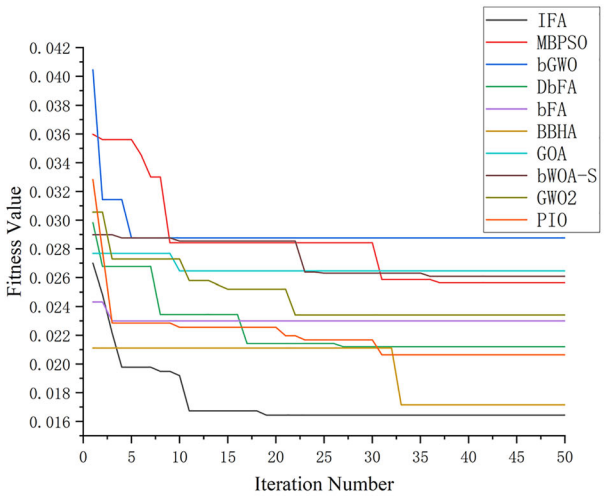


(k) Audit Data

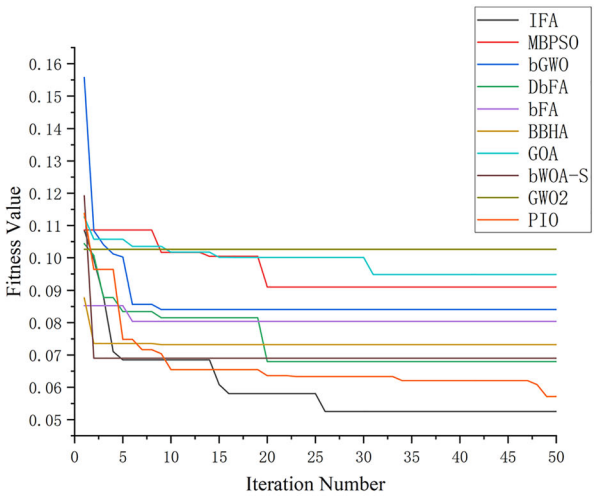


(l) Steel Plates Faults

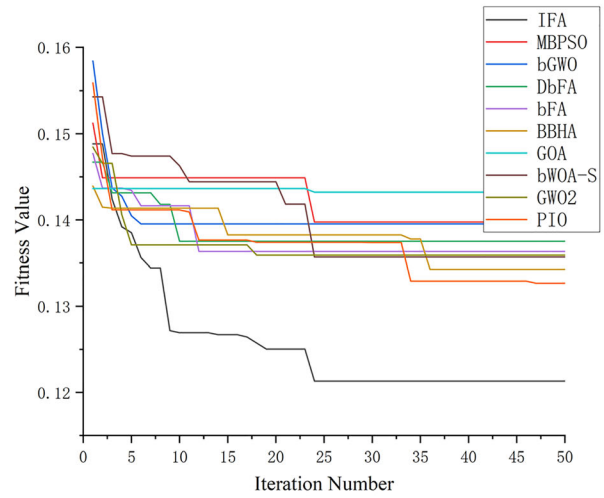
Fig. 10 continued



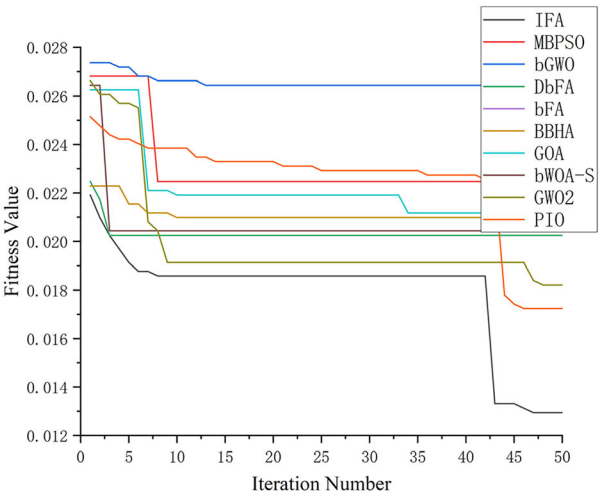
(m) Dermatology



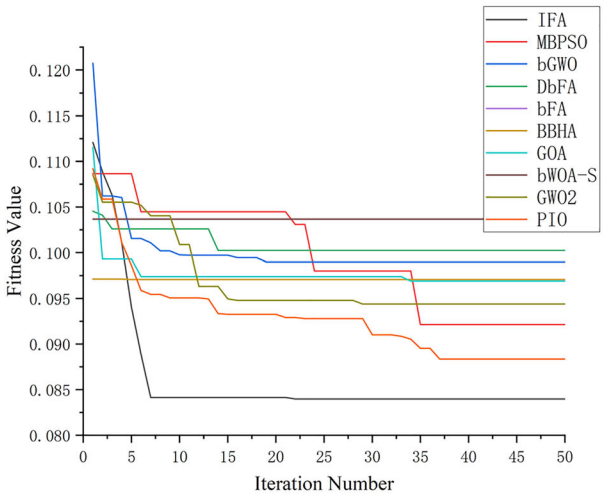
(n) Chess (King-Rook vs. King-Pawn)



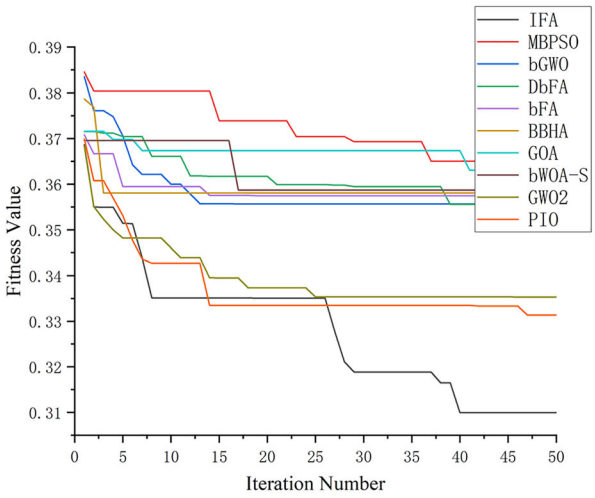
(o) QSAR biodegradation



(p) Divorce Predictors



(q) Spambase



(r) Arrhythmia

Fig. 10 continued

Table 8 Adjusted p values of the Wilcoxon test of the proposed IFA *vs* other algorithms ($p \geq 0.05$ are underlined)

Dataset	MBPSO	bGWO	DbFA	bFA	BBHA	GOA	bWOA-S	GWO2	PIO
Housing Prices	0.0002	6.02E−06	0.0013	0.0048	3.29E−05	4.56E−06	2.62E−05	3.89E−06	2.50E−05
Wine	9.78E−06	3.75E−06	0.0489	0.0490	1.46E−05	1.08E−05	5.89E−06	4.48E−06	6.27E−06
Heart failure clinical records	4.09E−05	1.77E−05	3.75E−06	3.75E−06	3.75E−06	3.75E−06	0.0953	6.10E−06	0.0238
Japanese Credit Screening	1.78E−05	3.75E−06	0.0465	0.2338	8.56E−05	0.0010	1.83E−05	9.41E−06	0.0008
Zoo	4.56E−06	3.75E−06	0.0138	0.0472	1.45E−05	6.82E−05	0.0036	7.89E−06	0.2411
Lymphography	3.75E−06	4.56E−06	0.0001	0.0145	3.89E−06	4.91E−06	4.81E−05	4.15E−06	0.0081
Image Segmentation	3.75E−06	4.56E−06	0.0194	0.0017	1.64E−05	3.75E−06	3.75E−06	0.0003	0.0008
Mobile Price	3.75E−06	4.56E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	5.48E−06	0.1063
Anuran Calls (MFCCs)	3.75E−06	3.75E−06	4.31E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	6.17E−06	0.0112
Parkinsons	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	9.78E−06	0.2070
Audit Data	3.75E−06	3.75E−06	3.75E−06	4.15E−06	3.75E−06	3.75E−06	4.43E−06	0.0489	0.0519
Steel Plates Faults	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	0.0014	0.6388
Dermatology	3.89E−06	7.22E−06	3.75E−06	4.15E−06	0.6883	3.75E−06	3.75E−06	3.75E−06	0.0031
Chess (King-Rook vs. King-Pawn)	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	4.91E−06
QSAR biodegradation	3.75E−06	6.68E−06	3.75E−06	6.17E−06	3.75E−06	3.75E−06	3.75E−06	6.17E−06	0.0140
Divorce Predictors	3.89E−06	3.75E−06	1.45E−05	1.32E−05	4.91E−06	7.22E−06	1.71E−05	0.0435	0.5505
Spambase	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06
Arrhythmia	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.75E−06	3.89E−06	3.75E−06	4.43E−06	0.0519

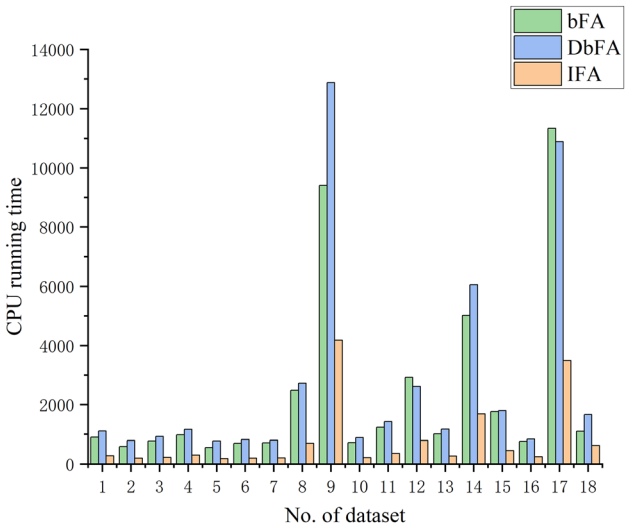


Fig. 11 CPU running time of FA-based algorithms

number of populations used by the algorithm. Compared to the traditional FA, even though IFA introduces the linear complexity of other operations while reducing the complexity inside the double layer loop, to ensure fairness, a comparison of CPU running time of the FA-based algorithms such as bFA, DbFA and IFA is presented in Fig. 11. The abscissa represents datasets, because of image size limitations; the no. of datasets is used instead of dataset names. The ordinate is the CPU running time. The execution time of these algorithms includes the generation of the initial population. As can be seen from the figure, the IFA proposed in this paper has a more significant improvement in time complexity than the classical FA and DbFA in this problem, which is one of its advantages.

5 Conclusion and future work

In this research, an improved firefly algorithm for feature selection with ReliefF-based initialization method and weighted voting mechanism is proposed and utilized to solve the feature selection problem. Experimental studies on 18 datasets show that the proposed algorithm is effective, and it also outperforms other comparison algorithms.

For future studies, the algorithm proposed in this paper has the potential for more in-depth research. On the one hand, the proposed initialization method can be effectively applied to the feature selection problem. Therefore, future research work should focus on the balance of the grouping algorithm and the possibility of combining other filtering algorithms with this initialization method. On the other hand, the weighted voting mechanism can also be applied to other application fields as a component of improved binary FA, which may provide inspiration for the design of new algorithms.

Funding This work is supported by the Key Project of Ningxia Natural Science Foundation (2022AAC02043), Major scientific Research Project of Northern University for Nationalities (ZDZX201901), the Natural Science Foundation of NingXia Hui Autonomous Region (2021AAC03185), Research Startup Foundation of North Minzu University (2020KYQD23), National Natural Science Foundation of China (61561001) and First-class Discipline Construction Fund project of Ningxia Higher Education (NXYLXK2017B09).

Data availability The datasets used during the current study are available in the Kaggle (https://www.kaggle.com) and the UCI Repository (http://archive.ics.uci.edu/ml/index.php).

Declarations

Conflict of interest The authors declare that there is no conflict of interests; we do not have any possible conflicts of interest.

References

1. Alican D, Derya B (2021) Machine learning and data mining in manufacturing. Expert Syst Appl 166:114060. https://doi.org/10.1016/j.eswa.2020.114060
2. Jie C, Jiawei L, Shulin W, Sheng Y (2018) Feature selection in machine learning: a new perspective. Neurocomputing 300:70–79. https://doi.org/10.1016/j.neucom.2017.11.077
3. Mohammad T, Majdi MM, Ali AH, Hossam F, Ibrahim A, Seyedali M, Hamido F (2019) An evolutionary gravitational search-based feature selection. Inf Sci 497:219–239. https://doi.org/10.1016/j.ins.2019.05.038
4. Lei Y, Huan L (2014) Efficient feature selection via analysis of relevance and redundancy. J Mach Learn Res 5, 1205–1224 (2004). Springer Nature 2021 LATEX template Article Title 19 https://doi.org/10.1023/B:JODS.0000045365.56394.b4
5. Mehrdad R, Kamal B, Elahe N, Saman F (2021) Review of swarm intelligence-based feature selection methods. Eng Appl Artif Intell 100:104210. https://doi.org/10.1016/j.engappai.2021.104210
6. Gao W, Hu L, Zhang P, He J (2018) Feature selection considering the composition of feature relevancy. Pattern Recogn Lett 112:70–74. https://doi.org/10.1016/j.patrec.2018.06.005
7. Manoranjan D, Huan L (1997) Feature selection for classification. Intell Data Anal 1:1–4. https://doi.org/10.1016/S1088-467X(97)00008-5
8. Sankalap A, Priyanka A (2019) Binary butterfly optimization approaches for feature selection. Expert Syst Appl 116:147–160. https://doi.org/10.1016/j.eswa.2018.08.051
9. Ryan JU, Melissa M, William GLC, Randal SO, Jason HM (2018) Relief-based feature selection: introduction and review. J Biomed Inform 85:189–203. https://doi.org/10.1016/j.jbi.2018.07.014
10. Kira K, Rendell LA (1992) The feature selection problem: Traditional methods and a new algorithm. In: Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI Press, San Jose, California, pp 129–134
11. Igor K (1994) Estimating attributes: analysis and extensions of RELIEF. Paper presented at the 94th European Conference on Machine Learning, Catania, Italy, 6–8 April 1994
12. Girish C, Ferat S (2014) A survey on feature selection methods. Comput Electr Eng 40:16–28. https://doi.org/10.1016/j.compeleceng.2013.11.024
13. Wan Y, Wang M, Ye Z, Lai X (2016) A feature selection method based on modified binary coded ant colony optimization algorithm. Appl Soft Comput 49:248–258. https://doi.org/10.1016/j.asoc.2016.08.011
14. Ibrahim A, Maria H, Hossam F, Nailah A, Ali AH, Majdi MM, Mohamed EAE, Seyedali M (2020) A dynamic locality multi-objective salp swarm algorithm for feature selection. Comput Ind Eng 147:106628. https://doi.org/10.1016/j.cie.2020.106628
15. Emrah H, Bing X, Mengjie Z (2018) Differential evolution for filter feature selection based on information theory and feature ranking. Knowl Based Syst 140:103–119. https://doi.org/10.1016/j.knosys.2017.10.028
16. Yong Z, DunWei G, XiaoZhi G, Tian T, Xiaoyan S (2020) Binary differential evolution with self-learning for multi-objective feature selection. Inf Sci 507:67–85. https://doi.org/10.1016/j.ins.2019.08.040
17. Ke C, Fengyu Z, Xianfeng Y (2019) Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection. Expert Syst Appl 128:140–156. https://doi.org/10.1016/j.eswa.2019.03.039
18. Maryam A, Behrouz MB (2018) Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism. Expert Syst Appl 113:499–514. https://doi.org/10.1016/j.eswa.2018.07.013
19. Bing X, Mengjie Z, Will NB (2012) New Fitness Functions in Binary Particle Swarm Optimisation for Feature Selection. Paper presented at the IEEE Congress on Evolutionary Computation, Brisbane, Australia, 10–15 June 2012
20. Bach HN, Bing X, Peter A (2019) PSO with surrogate models for feature selection: static and dynamic clustering-based methods. Memetic Comput 10:291–300. https://doi.org/10.1007/s12293-018-0254-9
21. Wang L, Gao Y, Gao S, Yong X (2021) A new feature selection method based on a self-variant genetic algorithm applied to android malware detection. Symmetry 13:1290. https://doi.org/10.3390/sym13071290
22. Eid E, Hossam MZ, Aboul EH (2016) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381. https://doi.org/10.1016/j.neucom.2015.06.083
23. Pei H, JengShyang P, ShuChuan C (2020) Improved Binary Grey Wolf Optimizer and Its application for feature selection. Knowl Based Syst 195:105746. https://doi.org/10.1016/j.knosys.2020.105746
24. Mafarja MM, Ibrahim A, Hossam F, Abdelaziz IH, Ala MA, Seyedali M (2019) Binary grasshopper optimisation algorithm approaches for feature selection problems. Expert Syst Appl 117:267–286. https://doi.org/10.1016/j.eswa.2018.09.015
25. Majdi MM, Ibrahim A, Ali AH, Abdelaziz IH, Hossam F, Ala MA, Seyedali M (2018) Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems. Knowl Based Syst 145:25–45. https://doi.org/10.1016/j.knosys.2017.12.037
26. Gehad IS, Ghada K, Mohamed HH (2018) A novel chaotic salp swarm algorithm for global optimization and feature selection. Appl Intell 48:3462–3481. https://doi.org/10.1007/s10489-018-1158-6
27. Hossam F, Majdi MM, Ali AH, Ibrahim A, Ala MA, Seyedali M, Hamido F (2018) An efficient binary salp swarm algorithm with crossover scheme for feature. Knowl Based Syst 154:43–67. https://doi.org/10.1016/j.knosys.2018.05.009
28. Emrah H, Bing X, Mengjie Z, Dervis K, Bahriye A (2015) A multi-objective artificial bee colony approach to feature selection using fuzzy mutual information. Paper presented at the IEEE Congress on Evolutionary Computation, Sendai, Japan, 25–28 May 2015

29. Majdi MM, Seyedali M (2017) Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing 260:302–312. https://doi.org/10.1016/j.neucom.2017.04.053
30. Majdi MM, Seyedali M (2018) Whale optimization approaches for wrapper feature selection. Appl Soft Comput 62:441–453. https://doi.org/10.1016/j.asoc.2017.11.006
31. Yanan Z, Renjing L, Xin W, Huiling C, Chengye L (2021) Boosted binary harris hawks optimizer and feature selection. Eng Comput 37:3741–3770. https://doi.org/10.1007/s00366-020-01028-5
32. Yang XS (2009) Firefly algorithms for multimodal optimization, stochastic algorithms: foundations and applications. SAGA 2009. Lecture notes in computer science. Springer, Berlin, Heidelberg, 5792.
33. Jinran W, YouGan W, Kevin B, YuChu T, Brodie L, Zhe D (2020) An improved firefly algorithm for global continuous optimization problems. Expert Syst Appl 149:113340. https://doi.org/10.1016/j.eswa.2020.113340
34. ChunFeng W, WenXin S (2019) A novel firefly algorithm based on gender difference and its convergence. Appl Soft Comput 80:107–124. https://doi.org/10.1016/j.asoc.2019.03.010
35. Aref Y, Cemal K (2018) A modified firefly algorithm for global minimum optimization. Appl Soft Comput 62:29–44. https://doi.org/10.1016/j.asoc.2017.10.032
36. Xingsi X (2020) A compact firefly algorithm for matching biomedical ontologies. Knowl Inf Syst 62:2855–2871. https://doi.org/10.1007/s10115-020-01443-6
37. Asma MA, Abdulqader MM, Abdullatif G (2019) An improved hybrid firefly algorithm for capacitated vehicle routing. Appl Soft Comput 84:1568–4946. https://doi.org/10.1016/j.asoc.2019.105728
38. Hui W, Wenjun W, Zhihua C, Xinyu Z, Jia Z, Ya L (2018) A new dynamic firefly algorithm for demand estimation of water resources. Inf Sci 438:95–106. https://doi.org/10.1016/j.ins.2018.01.041
39. Selvakumar B, Muneeswaran K (2019) Firefly algorithm based feature selection for network intrusion detection. Comput Secur 81:148–155. https://doi.org/10.1016/j.cose.2018.11.005
40. Long Z, Linlin S, Jianhua W (2017) Optimal feature selection using distance-based discrete firefly algorithm with mutual information criterion. Neural Comput Appl 28:2795–2808. https://doi.org/10.1007/s00521-016-2204-0
41. Yong Z, Xianfang S, DunWei G (2017) A return-cost-based binary firefly algorithm for feature selection. Inf Sci 418:561–574. https://doi.org/10.1016/j.ins.2017.08.047
42. Bing X, Mengjie Z, Will NB (2014) Particle swarm optimisation for feature selection in classification:novel initialisation and updating mechanisms. Appl Soft Comput 18:261–276. https://doi.org/10.1016/j.asoc.2013.09.018
43. Bach HN, Bing X, Ivy L, Mengjie Z (2014) PSO and statistical clustering for feature selection: A new representation. Paper presented at the 10th SEAL International Conference, Dunedin, New Zealand,15–18 December 2014
44. Elnaz P, Nizamettin A (2017) Binary black hole algorithm for feature selection and classification on biological data. Appl Soft Comput 56:94–106. https://doi.org/10.1016/j.asoc.2017.03.002
45. Hui W, Zhihua C, Hui S, Shahryar R, XinShe Y (2017) Randomly attracted firefly algorithm with neighborhood search and dynamic parameter adjustment mechanism. Soft Comput 21:5325–5339. https://doi.org/10.1007/s00500-016-2116-z
46. Bach HN, Bing X, Mengjie Z (2020) A survey on swarm intelligence approaches to feature selection in data mining. Swarm Evol Comput 54:100663. https://doi.org/10.1016/j.swevo.2020.100663
47. Mohamed AB, Doaa E, Ibrahim ME, Victor HCA, Seyedali M (2020) A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. Expert Syst Appl 139:112824. https://doi.org/10.1016/j.eswa.2019.112824
48. Dua D, Graff C (2019) UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science
49. Yudong Z, Shuihua W, Preetha P, Genlin J (2014) Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. Appl Soft Comput 64:22–31. https://doi.org/10.1016/j.knosys.2014.03.015
50. Hussien AG, Hassanien AE, Houssein EH, Bhattacharyya S, Amin M (2019) S-shaped binary whale optimization algorithm for feature selection. In: Bhattacharyya S, Mukherjee A, Bhaumik H, Das S, Yoshida K (eds) Recent trends in signal and image processing. Springer, Singapore, pp 79–87
51. Hadeel A, Ahmad S, Khair ES (2020) A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Expert Syst Appl 148:113249. https://doi.org/10.1016/j.eswa.2020.113249
52. Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J Roy Stat Soc: Ser B (Methodol) 57:289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x