Optimal Feature Selection Based on Discrete Grasshopper Optimization Algorithm and K-nearest Neighbor Classifier

Yu-Liang Qi, Jie-Sheng Wang*, Yu-Wei Song, Yu-Cai Wang, Hao-Ming Song, Jia-Ning Hou

Abstract—**In the majority of data mining tasks, feature selection serves as an essential pre-processing step. The most important attributes are selected to lower the dimensionality reduction of data set and enhance the precision of classification. Natural heuristic algorithms are extensively employed in the realm of encapsulated feature selection. Based on the wrapper feature selection method, seven natural heuristic algorithms are used to solve feature selection problems and perform performance comparison, which include Slime Mold Algorithm (SMA), Whale Optimization Algorithm (WOA), Harris Hawks Optimization Algorithm (HHO), Marine Predator Algorithm (MPA), Butterfly Optimization Algorithm (BOA), Cuckoo Search (CS) and Firefly Algorithm (FA). At the same time, performance tests are carried out on 21 standard UCI data sets to verify the functionality of various algorithms, and the convergence curves and accuracy boxplots of 7 natural heuristic algorithms on 21 data sets are given. The simulation outcomes were assessed utilizing the mean and standard deviation of fitness, as well as the number of chosen features, and the running time, with the optimal value in bold. By comparing the comprehensive performance indexes, MPA obtained the maximum mean fitness value in most data sets (16 data sets), followed by FA (6 data sets). SMA obtained the best performance and finds the minimum eigenvalues (20 data sets) in multiple data sets and has an advantage in computing time.**

Index Terms—**Feature Selection, Grasshopper Optimization Algorithm, KNN Classifier, Performance Evaluation**

I. INTRODUCTION

WITH the continuous upgrading of science and technology, massive data sets come into being, and their complexity and diversity are constantly increasing. However, multidimensional data has deficiencies such as duplicate data and excessive modeling duration, which brings great difficulty to data analysis [1]. Feature selection occupies a vital role in statistical analysis and artificial intelligence, which can improve classifier performance, reduce computational costs and enhance model interpretation [2]. As an effective means of data preprocessing, feature selection technology can eliminate redundant and unnecessary variables, reduce dimension processing of data sets, improve model generalization ability and reduce over-fitting [3]. Zhang et al. found that the feature selection method can significantly enhance the classification performance of deep neural networks, and also reduce the network complexity and training time [4]. The study of Li et al. also shows that the feature selection method can help the machine learning algorithm better capture important features in the data and improve the accuracy and stability of the classifier [5]. Feature selection is a data preprocessing phase in artificial intelligence methods. Extracting a subset of the most influential features of the original set can lower the dimensionality of the data, simplify the intricacy of the structure, and eliminate the features that interfere with the learning task, which is an important means to improve the performance of the algorithm [6]. With the deepening of the application of large-scale data, the size of the data has exploded, so dimension reduction has become an indispensable link in the advanced data preprocessing step. Feature extraction and feature selection are frequently employed as dimensionality reduction techniques. Since feature selection retains the original features of data and has good interpretability, it has become the main dimension reduction methods for data [7-8].

Feature selection is a very important problem in the realm of artificial intelligence and knowledge discovery. In the actual data analysis task, a significant quantity of samples and features are usually involved. Selecting the most important features for classification or regression can not only improve the classification effect and generalization performance of the machine learning model, but also reduce the computing overhead brought by dimensions and enhance the effectiveness and scalability of the algorithm. However, the traditional feature selection method has some shortcomings, such as high computational intricacy and difficulty to face the problem of high-dimensional data. Therefore, more and more research is now devoted to solve these difficult problems by optimizing algorithms. Zou et al. proposed a feature selection method rooted in genetic optimization and support vector machination to control the

Manuscript received July 22, 2023; revised October 28, 2023. This work was supported by the Postgraduate Education Reform Project of Liaoning Province (Grant No. LNYJG2022137).

Yu-Liang Qi is a postgraduate student at School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China (E-mail: 2281424682@qq.com).

Jie-Sheng Wang is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (Corresponding author, phone: 86-0412-2538355; fax: 86-0412-2538244; e-mail: wang_jiesheng@126.com).

Yu-Wei Song is a postgraduate student at School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China (E-mail: 2296087986@qq.com).

Yu-Cai Wang is a doctoral student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: wyc118117@163.com).

Hao-Ming Song is a doctoral student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: 823234816@qq.com).

Jia-Ning Hou is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: 1994905806@qq.com).

variation and crossover of individual features through genetic algorithm, thus selecting the optimal feature set [9]. Chu et al. put forth a feature selection method based on genetic algorithm and classifier integration, and achieved very good classification effect on multiple data sets [10]. In addition, integrating particle swarm optimization, simulated annealing, and differential evolution techniques are also used in the research and application of feature selection problems. These algorithms try to reduce the feature dimension, improve the classification accuracy and avoid over-fitting problems by optimizing search.

Grasshopper Optimization Algorithm (GOA) is a biometric algorithm proposed by Australian scholar Seyedali Mirjalili in 2017 [11]. It is a search algorithm abstracted by imitating the foraging habits of grasshopper groups in nature. Based on the action law of grasshopper population and individual, the distance between grasshoppers is divided into mutual exclusion zone, attraction zone and suitable zone, and the solution to the problem is obtained by judging the distance types between grasshoppers individuals. GOA has advantages in solving practical problems of unknown search space. GOA has strong searching ability, but its disadvantage is that it tends to converge towards local optima. In 2018, Arora et al proposed a global optimization Chaos GOA [12], in which chaotic mapping was used to successfully balance the exploration and development of grasshoppers and reduce the repulsion (attraction) among grasshoppers. By using 13 functions to test its feasibility, the findings demonstrate a significant enhancement in the efficiency of GOA. Aiming at the disadvantage of poor accuracy of GOA, Li et al. proposed the curve adaptive and simulated annealing GOA [13]. In this method, curve adaptive method is introduced to substitute the linear adaptive approach for the key parameters of GOA, which increases the global search efficiency of the algorithm. The simulated annealing algorithm is introduced to receive the inferior solution of GOA with a certain probability, and the ability of the algorithm to find the global optimal value is improved. The discrete GOA is a global optimization algorithm similar to the ant colony algorithm, which can efficiently steer clear of the issue of local optimal solution by adding random disturbance and local search methods [14]. At the same time, KNN classifier is also widely used in feature selection problems, which does not require prior knowledge and distribution hypothesis, and is suitable for multiple classification and nonlinear classification and other situations [15]. This work introduces a superior feature selection method with discrete GOA and KNN classifier. Through experimental verification on multiple datasets, the discrete GOA can discover the global optimal solution faster, and has better performance and computational efficiency. The outline of the paper is presented as below. The second section presents GOA, while the third section elaborates on the KNN classifier, fitness function and feature selection architecture. The fourth section presents experimental simulations and result examinations. Finally, the conclusion is drawn.

II. Grasshopper Optimization Algorithm

A. Basic Principles of Grasshopper Optimization Algorithm

GOA is an emerging natural heuristic algorithm that draws on the social behavior of insects such as grasshoppers, especially their gathering behavior for searching for food sources and large-scale migration. By simulating the movement characteristics of larvae and adults of grasshoppers, GOA carries out local development and global search respectively, so as to achieve efficient target search. In the larval stage, GOA simulates grasshopper jumping and columnar rotating flight, looking for food sources on the prey path with small steps and small moving speed.

In the adult stage, GOA uses the search strategy of large stride length and large moving speed to simulate the characteristics of large-scale migration of grasshoppers. Through this simulation of the grasshopper life cycle, GOA is able to better explore the exploration domain and increase target search competence. The advantages of this algorithm are simple operation, easy to implement, less sensitivity to initial value, and can be applied to various optimization problems. The experimental results for the standard function optimization show that the convergence of GOA is superior to the PSO. The mathematical model of grasshopper individual location is shown in Eq. (1).

$$X_i = S_i + G_i + A_i \tag{1}$$

In the equation, X_i represents the position of the i -th grasshopper, S_i represents the force between the i -th grasshopper and other grasshoppers, G_i signifies the gravity exerted on the i -th grasshopper, and A_i signifies the wind resistance experienced by the i -th grasshopper during flight. Eq. (1) can be recorded in the form of Eq. (2)

$$S_i = \sum_{j=1, j\neq 1}^{N} s(d_{ij})d'_{ij} \tag{2}$$

In the equation, d_{ij} represents the distance between the i -th and j -th grasshoppers, which is calculated by:

$$d_{ij} = |X_i - X_j| \tag{3}$$

The distance unit vector between individual i and individual j is $d'_{ij} = (x_j - x_i)/d_{ij}$, and the function s can be regarded as the strength of community force.

$$s(r) = fe^{\left(\frac{-r}{r}\right)} - e^{-r} \tag{4}$$

In the equation, f is the intensity of attraction, and l is the step size of attraction. In the process of searching for food, grasshoppers will establish three types of areas according to community forces: ease area, attractiveness area, and exclusion area. When the distance between grasshoppers gradually increases (more than 10), the function s tends to be unable to generate social forces, that is, there is no interaction between individuals. The general practice is to limit the position of individuals to $[1,4]$, while the position of individuals in the comfort zone will not be updated. G_i is defined as:

$$G_i = -ge'_g \tag{5}$$

In the equation, g represents the gravitational constant, while e'_g denotes the unit vector pointing towards the Earth's center. A_i is defined as:

$$A_i = ue'_w \tag{6}$$

In the equation, u denotes the drift constant, and e'_w is the wind direction unit vector. Substitute the relevant parameters into Eq. (1) to obtain:

$$X_i = \sum_{j=1, j\neq 1}^{N} s\left(\left|x_j - x_i\right|\right)\frac{x_j - x_i}{d_{ij}} - g e'_g + u e'_w \tag{7}$$

In the equation, N denotes the number of grasshoppers in the population.

In general, individuals are weakly affected by gravity G and wind A, and population individuals only update their positions according to the influence of community force S. Therefore, the location update method can be defined as:

$$X_{i,d} = c_1\left(\sum_{j=1, j\neq 1}^{N} c_2 \frac{ub_d - lb_d}{2} s\left(\left|x_j - x_i\right|\right)\frac{x_j - x_i}{d_{ij}}\right) + T_d^* \tag{8}$$

In the equation, ub_d serves as the upper threshold for D-dimensional space, lb_d represents the lower boundary of D-dimensional space, T_d^* is the optimal solution found in the current population, the function s is defined by Eq. (4), c_1 and c_2 are the adjustment coefficients, which are used to shrink and adjust the comfort area, attraction area and exclusion area, and this parameter changes linearly with the iteration.

$$c_1 = c_2 = c_{\max} - l \cdot \frac{c_{\max} - c_{\min}}{L} \tag{9}$$

In the equation, $c_{\max}$ represents the maximum adjustment coefficient, $c_{\min}$ denotes the minimum value, l is the current iteration count, and L signifies the maximum pseudo-iteration limit.

B. Flowchart of Grasshopper Optimization Algorithm

GOA iteratively solves the optimal solution in the solution domain by adjusting the position coordinates of the grasshopper. Its flowchart is presented in Fig. 1.

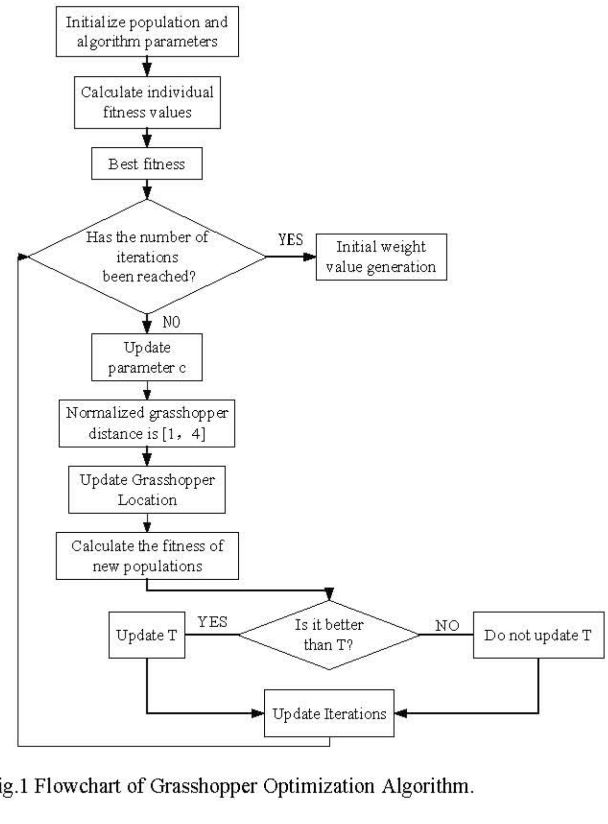


Fig.1 Flowchart of Grasshopper Optimization Algorithm.

The detailed steps are outlined below.

(1) Initialization. Firstly, initialize the grasshopper population and the overall parameters of the algorithm, such as the boundary values $c_{\max}$ and $c_{\min}$ of parameter c, the iterative count L, and the population quantity N. Define the fitness function.

(2) Measure the fitness value of each individual based on the fitness function and select the optimal fitness T.

(3) Check if the maximum number of iterations has been reached. If so, the algorithm concludes, and the present solution turns out to be the optimal one.

(4) Update parameter c according to Eq. (9).

(5) Normalize the distance between grasshoppers to $[1,4]$.

(6) According to Eq. (8), revise the grasshopper's position and compute the refreshed individual fitness. Contrast it with T. If the fitness is better than T, update T, otherwise it remains unchanged.

(7) Renew the iteration number and repeat Step (3)-(7) above.

III. Feature Selection Based on Discrete Grasshopper Optimization Algorithm and KNN Classifier

A. K-nearest Neighbor Classifier

Discrete grasshopper optimization algorithm (DGOA) is usually used to solve Discrete optimization problems. K nearest neighbor (KNN) classifier is an instance based supervised learning algorithm. It selects the nearest K sample points as the nearest neighbor of the sample points by computing the remoteness between the sample points to be categorized and the sample points of the known category. Then, based on the class information of these K nearest neighbors, the sample points are classified using majority voting or weighted majority voting. The description of its algorithm is described as follows.

(1) Calculate the gap between the test data and numerous training data examples;

(2) Organize based on the ascending distance correlation;

(3) Choose the K points with the minimal distance;

(4) Calculate the repetitiousness of the initial K points within the category;

(5) Select the category that appears most often among the first K points as the forecast-ed classification for the test data.

In the experiment, KNN serves for classification purposes, calculating the Euclidean distance D_E between the training set and test data, and determining the K closest samples, as shown in Eq. (10).

$$D_E = \sqrt{\sum_{i=1}^{K} (Train_{Fi} - Test_{Fi})^2} \tag{10}$$

B. Fitness Function

Fitness function is usually determined by the goal function of the current problem itself, which can be the numerical value of the objective function or related indicators, or it can be an evaluation function based on the characteristics of the problem and individual solutions. The feature subset involved in the issue is composed of 1 and 0, represented by a binary vector is employed, where 1 signifies the feature is picked and 0 signifies it is not chosen. In this paper, the aforementioned two opposing objectives are embodied in the fitness function displayed in in Eq. (11).

$$fitness = h_1 \gamma_R(D) + h_2 \frac{|M|}{|N|} \quad (11)$$

In the equation, $\gamma_R(D)$ stands for the classification mistake rate associated with the chosen feature subset by the classifier, $|M|$ indicates the quantity of selected features, $|N|$ denotes the overall number of features, and h_1 and h_2 are two weight factors that reflect the classification efficiency and the subset's length, satisfying $h_1 + h_2 = 1$. Due to the need for an exact categorization model, the categorization exactness is assigned a significant inertia weight. In this article, h_1 and h_2 are set as 0.99 and 0.01, respectively.

C. Architecture for Feature Selection

FS aims to identify the least representative subset of features from the original set while maintaining the classifier's classification accuracy. In essence, FS is a dimension reduction method that utilizes the classifier's accuracy to assess the effectiveness of dimension reduction. The architecture for feature selection is shown in Fig. 2.

IV. Simulation Experiment and Result Analysis

A. Selection of Experimental Data

Fifteen datasets were selected from the UCI data set for classification studies. These datasets containing diverse instances, feature numbers and categories can see the effectiveness of the suggested feature selection method based on discrete GOA and KNN separator in various datasets from different perspectives. Table I provides detailed information on these datasets. In the simulation process, the K-nearest neighbor (KNN) algorithm with K=5 was employed to determine the classification precision in the fitness function, as KNN has been proven to become faster and more straightforward. During the experiment, different random populations were used to repeat 20 times.

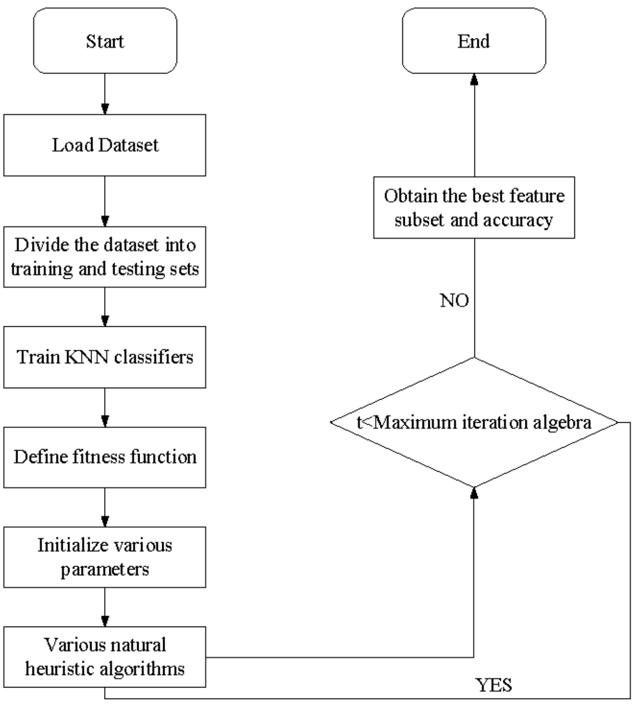


Fig. 2 FS architecture diagram of Feature selection.

In addition, to prevent over-fitting, the fold cross validation method was adopted, and the dataset was divided into the training set and the testing set. In the initial iteration, 80% of the feature vectors were allotted for training, with the remaining 20% reserved for testing. Subsequently, 20% of the feature vectors were set aside for testing, while the remaining 80% formed the training set. Repeat the above process until all feature vectors are used for testing. Finally, statistical analysis was conducted on the results of 20 independent runs.

All tests were performed utilizing MATLAB R2020a, running on an Intel Core i5-8300H machine with a CPU of 2.30 GHz, RAM of 8GB and Windows 10 operating system. In this investigation, each algorithm's population was configured to 10, with a maximum of 100 iterations. The common parameters of the eight algorithms remained consistent. The dimension of the searching space is equal to the total number of features. According to previous research by scholars, the classifier has the best classification performance when the hyper-parameter h_1 is set to 0.99[16].

B. Performance Evaluation of Feature Selection

Metrics are commonly employed when assessing and interpreting the outcomes of feature selection problems. These evaluation criteria include the fitness value, classification precision, and the average number of chosen features. Eq. (12)-(17) are the calculation methods for mean categorization precision, mean count of chosen features, mean fitness and standard deviation.

$$Mean\_accuracy = \frac{1}{20} \sum_{i=1}^{20} Accuracy_i \quad (12)$$

where, $Mean\_accuracy$ represents the mean classification precision achieved by executing the algorithm 20 separate times, and $Accuracy_i$ denotes the classification accuracy for each run. The classification accuracy is computed as follows:

$$Accuracy = \frac{1}{N} \sum_{i=1}^{N} match(Pl_i, Al_i) \quad (13)$$

where, N stands for the quantity of test set-points, that is, the number of instances of the dataset; Pl_i is the class label for the predicted class data point i, Al_i is the reference class label for the actual class i in the annotated data, and $match(Pl_i, Al_i)$ is a comparison discriminant function.

When $Pl_i == Al_i$, $match(Pl_i, Al_i) = 1$; Otherwise, $match(Pl_i, Al_i) = 0$.

$$Mean\_feature = \frac{1}{20} \sum_{i=1}^{20} feature_i \quad (14)$$

where, $Mean\_feature$ is the average of the number of selected features obtained by running the algorithm M times independently, and $feature_i$ is the value of the number of selected features obtained by each run.

$$Mean\_fitness = \frac{1}{20} \sum_{i=1}^{20} fitness_i \quad (15)$$

where, $Mean\_feature$ is the average fitness of the algorithm running independently M times, and f_i is the best fitness obtained for each run. The fitness value is shown in Eq. (16).

$$fitness = 0.99*(1-Accuracy)+0.01*\frac{|Selected\ features\quad Count|}{|Total\ features\quad Count|} \tag{16}$$

where, $Accuracy$ is the classification accuracy.

$$Std\_fitness = \sqrt{\frac{1}{20}\sum(fitness_i - Mean\_fitness)^2} \tag{17}$$

where, $Std\_fitness$ is the standard deviation of the fitness value, $fitness_i$ is the fitness value obtained for the i-th time, and $Mean\_fitness$ is calculated by Eq. (17).

C. Experimental simulation results and analysis

For different 15 UCI datasets, GOA and 7 commonly used natural heuristic algorithms, including slime mold algorithm (SMA), whale optimization algorithm (WOA), Harris Hawk algorithm (HHO), marine predator algorithm (MPA), butterfly optimization algorithm (BOA), Cuckoo search (CS) and Firefly algorithm (FA), are employed to conduct simulation experiments, whose results are shown in Table II-IV. The bold figures denote the optimal outcomes. Table II presents the mean and standard deviation of fitness for the eight natural heuristic algorithms. Tables III-IV respectively provide a comparison of the accuracy values of different algorithms and the mean of the number of selected features. In the above table, the outstanding results are emphasized in bold type. In Table II, GOA achieved the maximum average fitness value across the majority of datasets (8 datasets). From Table III, it can be seen that GOA has a significant advantage in accuracy. Table IV shows the number of features selected by all natural heuristic algorithms, in which GOA wins by absolute advantage. By drawing the convergence curves and precision box plot of the optimal classification accuracy calculated by the KNN classifier, the differences between 8 different natural heuristic algorithms are more intuitively and vividly displayed. The convergence curves of the 8 natural heuristic algorithms on 15 datasets are shown in Fig. 3. The horizontal axis of Fig. 3 denotes the number of iterations for the algorithm, while the vertical axis indicates the average accuracy figures for each algorithm following 20 separate executions. The box plot of accuracy values is shown in Fig. 4. As can be observed in Fig. 4 that GOA has the best convergence performance in most cases.

TABLE I. 15 DATA SETS USED IN THE SIMULATION EXPERIMENTS

Number	Datasets	Features	Instances	Classes
1	Algerian Forest Fires	12	244	2
2	Clean1	167	476	2
3	Climate Model Simulation Crashes	18	540	2
4	Connectionist Bench	60	208	2
5	Diabetic Retinopathy Debrecen	20	1151	2
6	Forest type mapping	27	326	4
7	Heart	13	303	2
8	Im	270	1000	20
9	Ionosphere	34	351	2
10	Page Blocks Classification	10	5473	5
11	Parkinson Disease	754	756	2
12	Pima Indians Diabetes	8	768	2
13	Planning Relax	13	182	2
14	QSAR biodegradation	41	1055	2
15	Semeion	256	1593	2

TABLE II. STANDARD DEVIATION OF AVERAGE FITNESS AND ACCURACY

Dataset	Measure	SMA	WOA	HHO	MPA	BOA	CS	FA	GOA
Algerian	AVG	0.0050	0.0104	0.0212	0.0008	0.0273	0.0033	0.0036	0.0019
	STD	0.0127	0.0144	0.0195	0.0000	0.0156	0.0065	0.0064	0.0006
Clean1	AVG	0.1064	0.0924	0.0970	0.0601	0.1066	0.0729	0.0825	0.0842
	STD	0.0126	0.0156	0.0158	0.0112	0.0095	0.0078	0.0113	0.0097
Climate	AVG	0.0579	0.0693	0.0763	0.0436	0.0744	0.0508	0.0438	0.0512
	STD	0.0129	0.0162	0.0105	0.0045	0.0080	0.0092	0.0084	0.0057
Connectionist	AVG	0.1280	0.1255	0.1387	0.0612	0.1380	0.0849	0.1000	0.0400
	STD	0.0220	0.0371	0.0266	0.0213	0.0291	0.0143	0.0122	0.0118
Diabetic	AVG	0.3227	0.3221	0.3400	0.2976	0.3355	0.3077	0.3026	0.2469
	STD	0.0134	0.0140	0.0209	0.0098	0.0155	0.0101	0.0132	0.0077
Forest	AVG	0.0827	0.0873	0.0919	0.0681	0.0944	0.0683	0.0683	0.0959
	STD	0.0072	0.0085	0.0093	0.0071	0.0063	0.0040	0.0072	0.0075
Heart	AVG	0.1458	0.1500	0.1703	0.1159	0.1729	0.1169	0.1153	0.0837
	STD	0.0234	0.0313	0.0411	0.0064	0.0241	0.0065	0.0043	0.0081
Im	AVG	0.2104	0.1880	0.1876	0.1484	0.2100	0.1660	0.1819	0.1905
	STD	0.0082	0.0150	0.0181	0.0119	0.0128	0.0112	0.0085	0.0089
Ionosphere	AVG	0.0786	0.0997	0.1037	0.0526	0.1153	0.0921	0.0967	0.0879
	STD	0.0163	0.0221	0.0234	0.0148	0.0115	0.0105	0.0108	0.0055

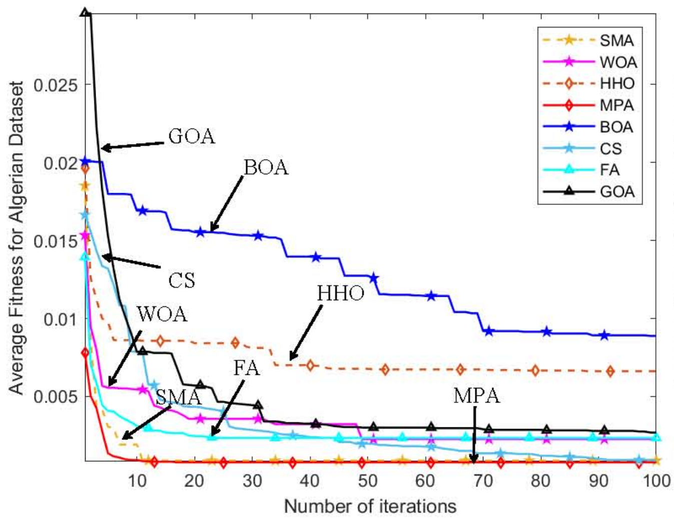
Page	AVG	0.0381	0.0388	0.0397	0.0376	0.0394	0.0376	0.0379	0.0352
	STD	0.0005	0.0007	0.0021	0.0002	0.0010	0.0000	0.0004	0.0008
Parkinson	AVG	0.1895	0.1948	0.1915	0.1505	0.2121	0.2554	0.2380	0.1021
	STD	0.0152	0.0269	0.0188	0.0142	0.0271	0.0392	0.0370	0.0045
Pima	AVG	0.2213	0.2276	0.2331	0.2141	0.2295	0.2143	0.2150	0.1922
	STD	0.0085	0.0072	0.0141	0.0049	0.0129	0.0054	0.0061	0.0024
Planning	AVG	0.2291	0.2467	0.2501	0.2020	0.2432	0.2114	0.2013	0.1843
	STD	0.0231	0.0204	0.0263	0.0119	0.0212	0.0164	0.0139	0.0137
QSAR	AVG	0.1120	0.1205	0.1204	0.0925	0.1249	0.0963	0.1018	0.1054
	STD	0.0071	0.0084	0.0076	0.0054	0.0081	0.0088	0.0046	0.0041
Semeion	AVG	0.0214	0.0192	0.0153	0.0058	0.0198	0.0111	0.0131	0.0101
	STD	0.0029	0.0042	0.0046	0.0017	0.0033	0.0026	0.0017	0.0018

TABLE III. COMPARISON OF ACCURACY VALUES BETWEEN DIFFERENT ALGORITHMS

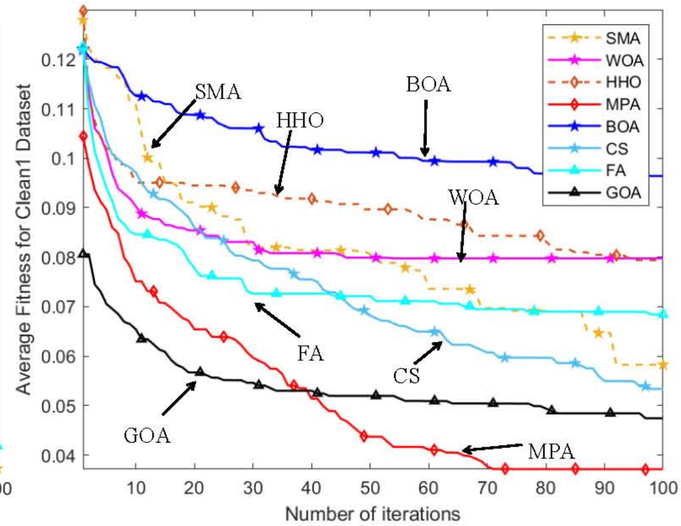
Dataset	SMA	WOA	HHO	MPA	BOA	CS	FA	GOA
Algerian	0.9958	0.9906	0.9802	0.9698	0.9750	0.9563	0.9500	1.0000
Clean1	0.8958	0.9111	0.9063	0.9416	0.8963	0.8732	0.8789	0.9211
Climate	0.9435	0.9333	0.9259	0.9532	0.9287	0.9157	0.9167	0.9537
Connectionist	0.8720	0.8768	0.8634	0.9256	0.8634	0.8134	0.8220	0.9655
Diabetic	0.6754	0.6774	0.6596	0.6883	0.6643	0.6433	0.6430	0.7561
Forest	0.9192	0.9159	0.9115	0.9332	0.9087	0.9014	0.9048	0.9076
Heart	0.8556	0.8528	0.8315	0.8676	0.8287	0.7981	0.7824	0.9231
Im	0.7920	0.8168	0.8160	0.8515	0.7920	0.7755	0.7760	0.8140
Ionosphere	0.9214	0.9007	0.8971	0.9243	0.8857	0.8536	0.8564	0.9171
Page	0.9664	0.9666	0.9651	0.9672	0.9663	0.9657	0.9658	0.9683
Parkinson	0.8086	0.8036	0.8070	0.8480	0.7871	0.6901	0.6957	0.9033
Pima	0.7817	0.7775	0.7706	0.7807	0.7732	0.7542	0.7631	0.8110
Planning	0.7708	0.7542	0.7514	0.7750	0.7569	0.7167	0.7125	0.8181
QSAR	0.8898	0.8829	0.8834	0.9092	0.8777	0.8664	0.8654	0.9002
Semeion	0.9835	0.9863	0.9890	0.9969	0.9841	0.9781	0.9800	0.9961

TABLE IV. AVERAGE NUMBER OF SELECTED FEATURES

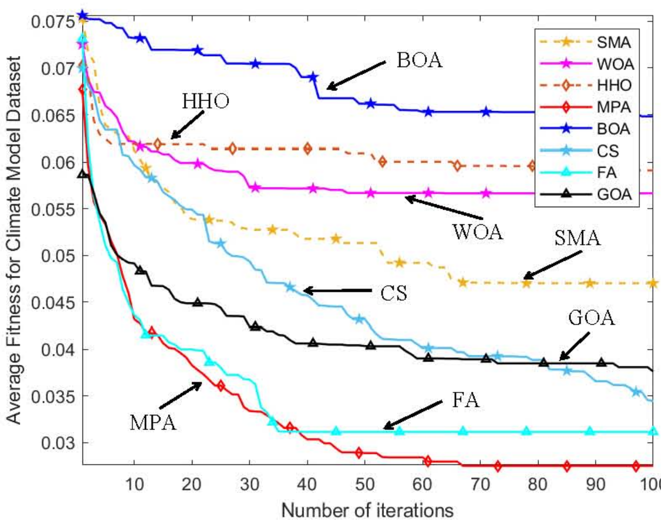
Dataset	SMA	WOA	HHO	MPA	BOA	CS	FA	GOA
Algerian	1.1000	1.5000	2.1500	1.0000	3.3500	1.5500	1.9500	2.6500
Clean1	53.1500	72.6000	71.7000	63.6500	65.8000	77.1000	81.0500	101.6000
Climate	4.0000	6.6000	6.0000	4.7500	7.7000	7.1500	7.9000	10.7500
Connectionist	7.6000	21.3500	21.1500	12.3000	16.6000	24.3000	27.4500	34.7500
Diabetic	2.6500	5.2500	5.6500	4.7500	6.0000	5.6500	5.6500	10.9000
Forest	7.3500	10.9500	11.8000	9.2000	10.8000	11.0000	10.9000	12.5500
Heart	3.7000	5.5000	4.4500	4.0500	4.2500	5.4000	5.7000	10.6000
Im	119.9000	179.0500	146.9500	131.3500	110.3000	131.0000	134.1500	172.6000
Ionosphere	2.6500	4.7000	6.2500	3.2500	7.3500	10.2000	11.4000	19.8000
Page	4.8500	5.8000	5.1500	5.0500	6.1000	5.2000	5.3500	3.8000
Parkinson	3.2500	31.6000	27.2500	54.4500	101.2000	325.4500	371.4000	480.0500
Pima	4.1500	5.8000	4.7500	4.0500	4.0000	4.2000	4.3000	4.1000
Planning	2.6500	4.0000	4.7500	3.1500	3.1000	4.5500	4.0000	5.0500
QSAR	12.0500	18.9500	20.4000	13.9500	15.8000	17.7500	20.0000	27.7500
Semeion	132.7500	148.6500	117.7000	100.9000	106.8500	121.3500	131.5500	166.6000



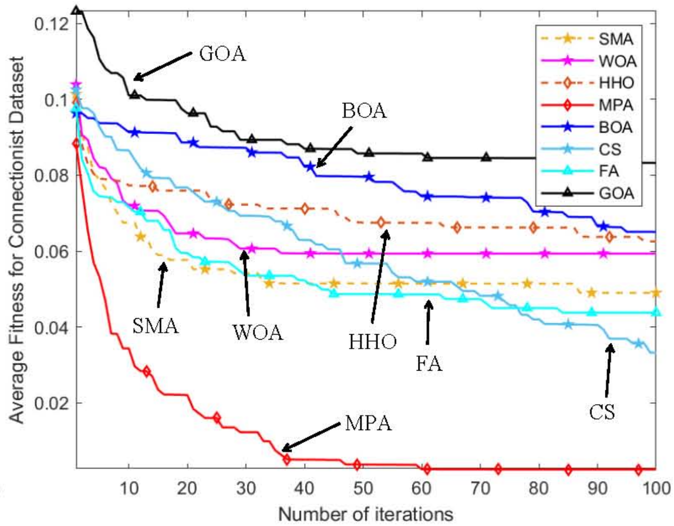
(a) Algerian



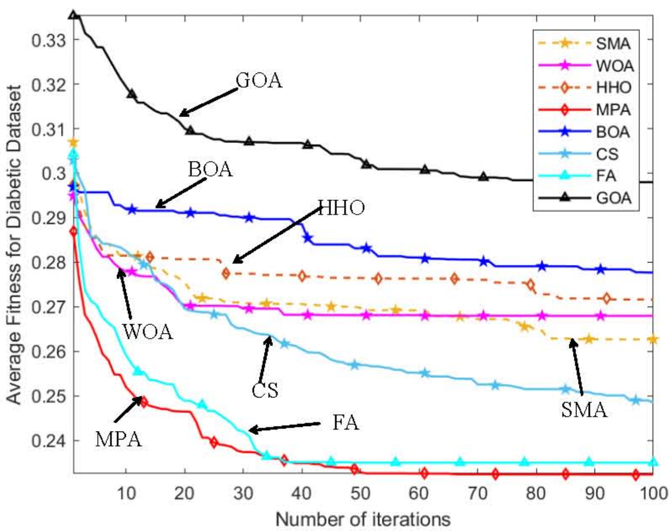
(b) Clean1



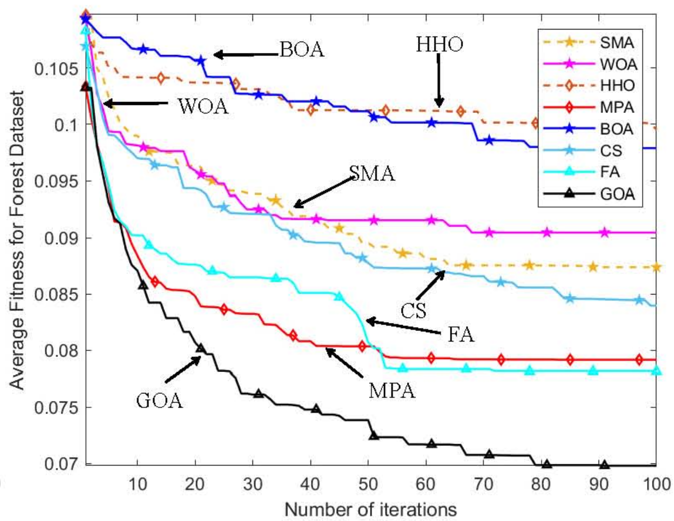
(c) Climate



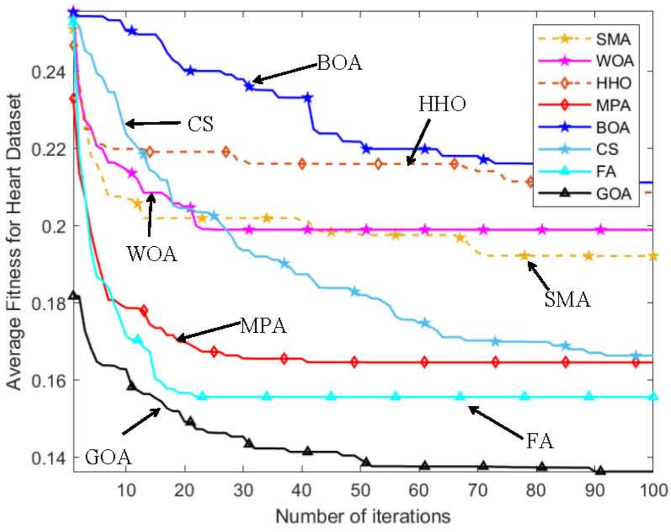
(d) Connectionist



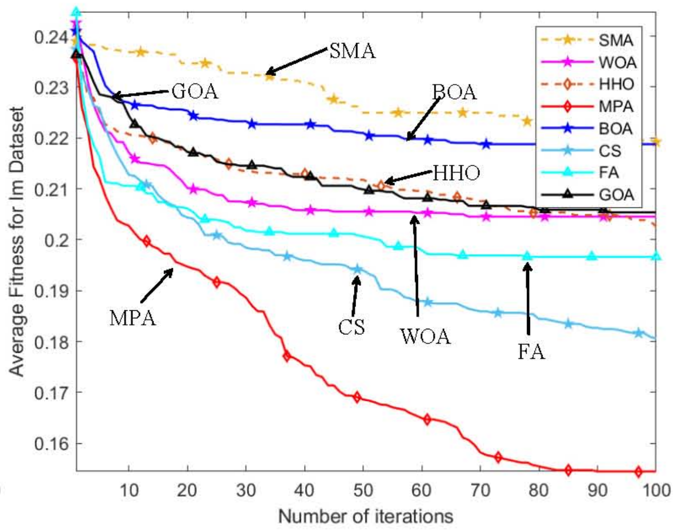
(e) Diabetic



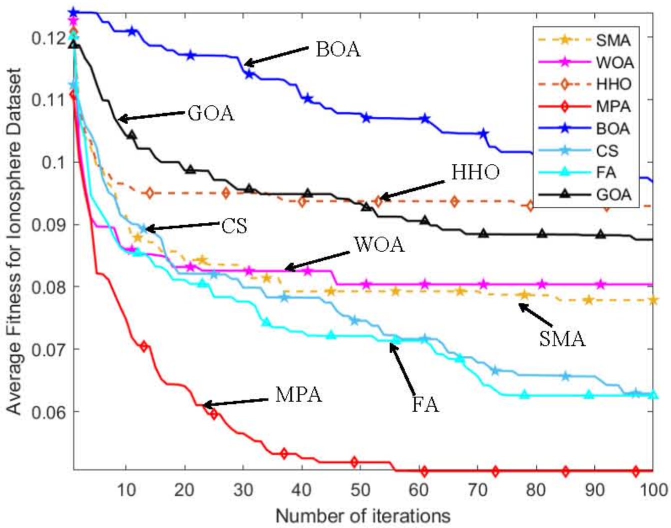
(f) Forest



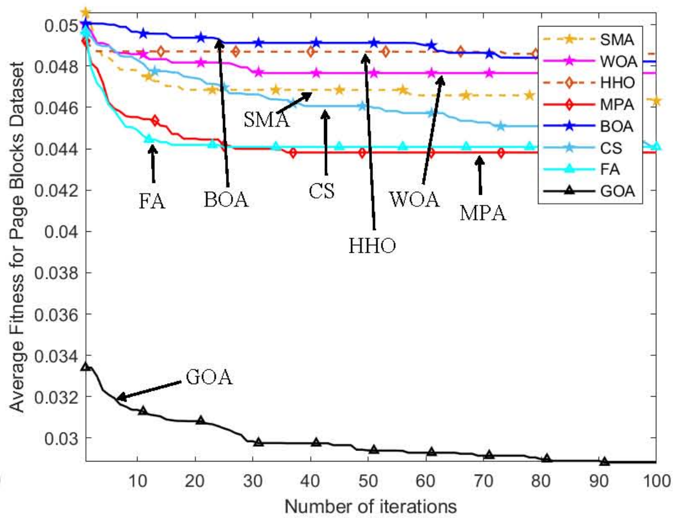
(g) Heart



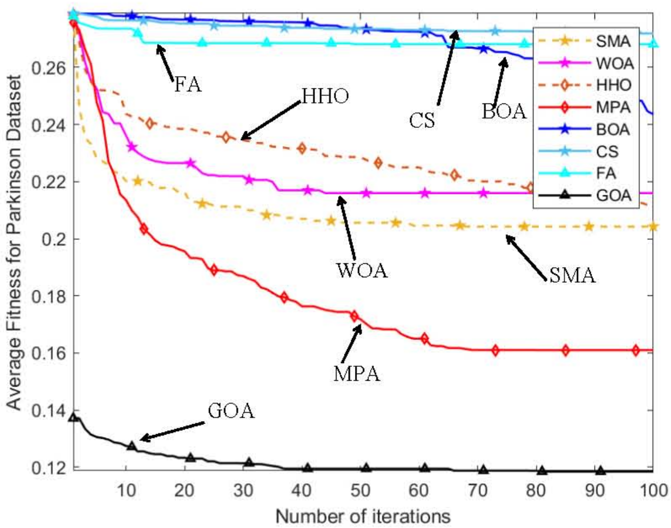
(h) Im



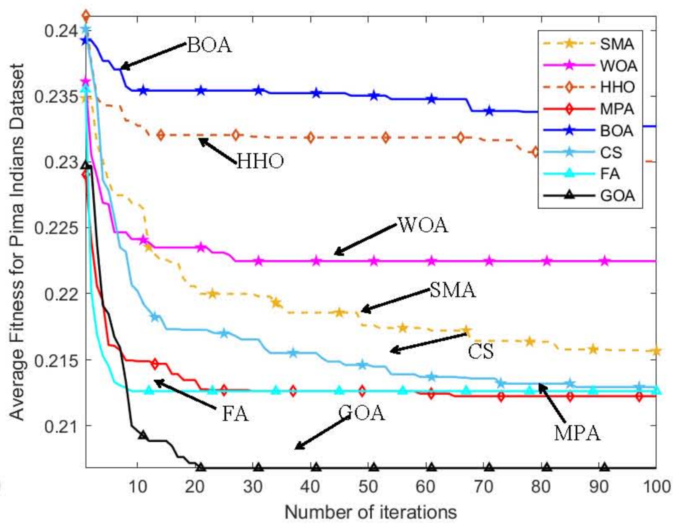
(i) Ionosphere



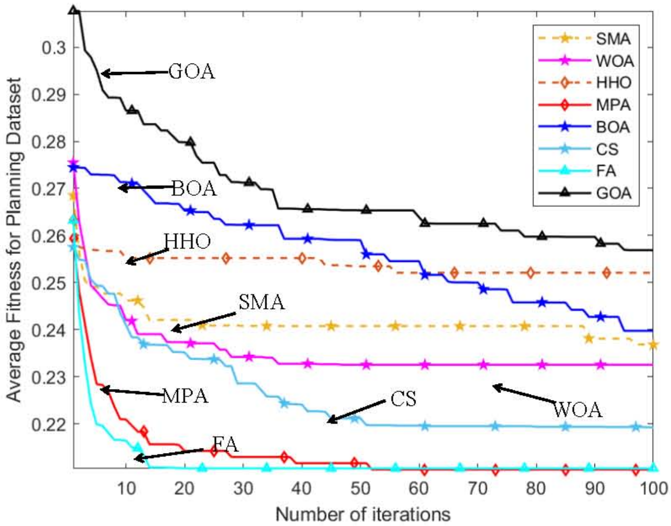
(j) Page



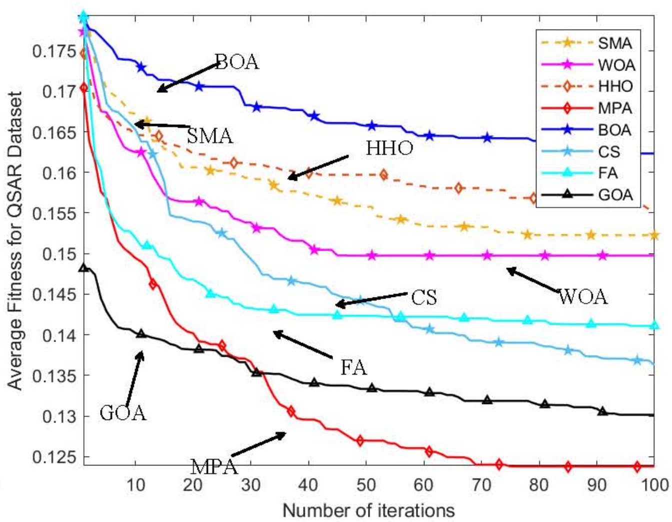
(k) Parkinson



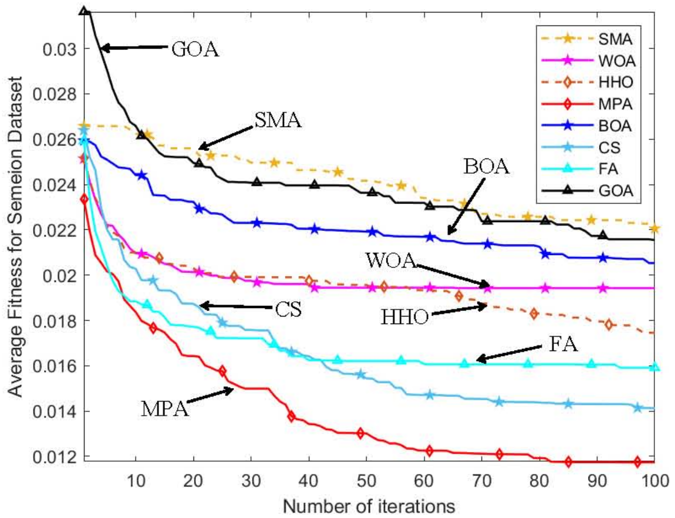
(l) Pima



(m) Planning

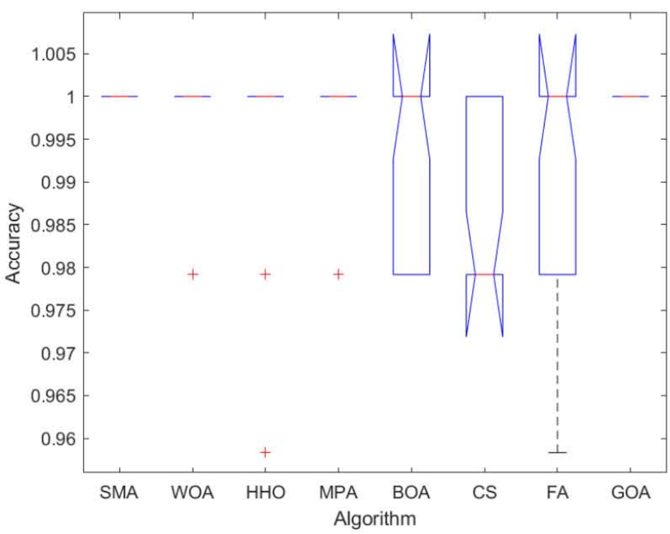


(n) QSAR

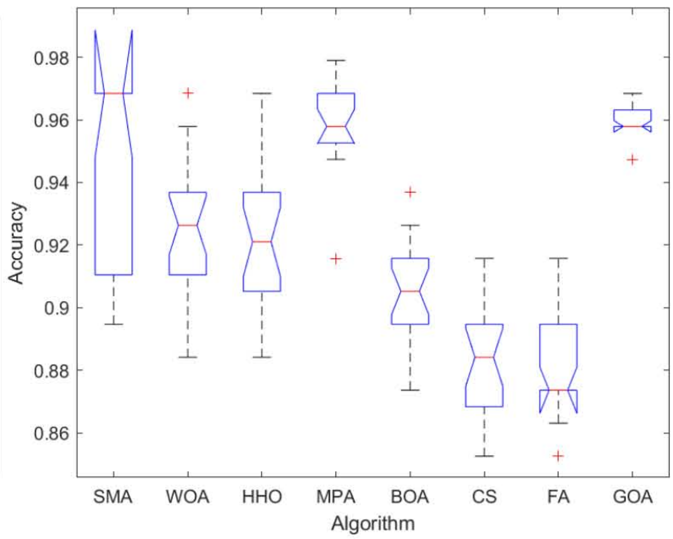


(o) Semeion

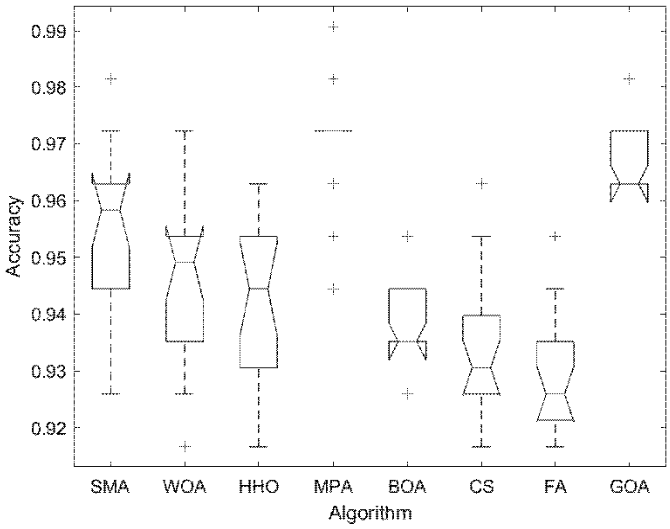
Fig. 3 Convergence curves of eight natural heuristic algorithms on fifteen datasets.



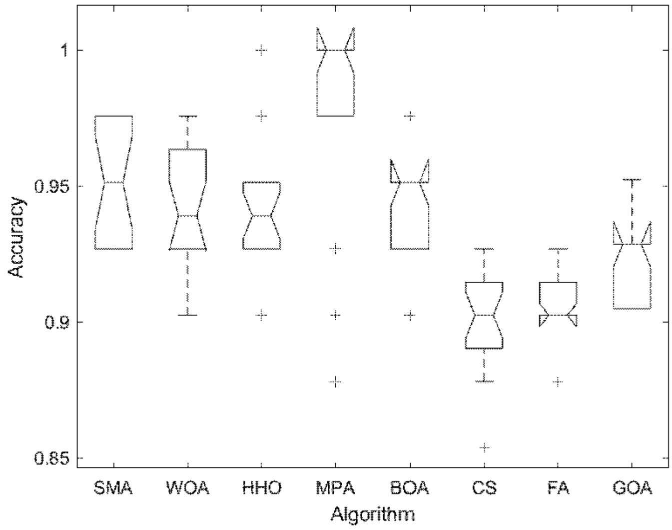
(a) Algerian



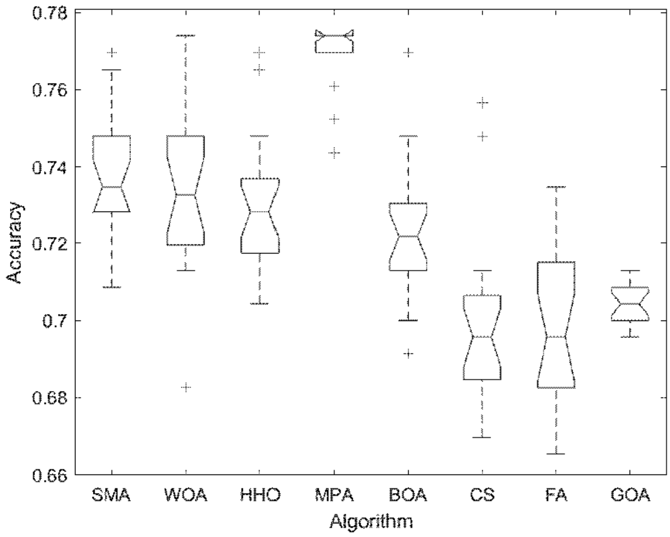
(b) Clean1



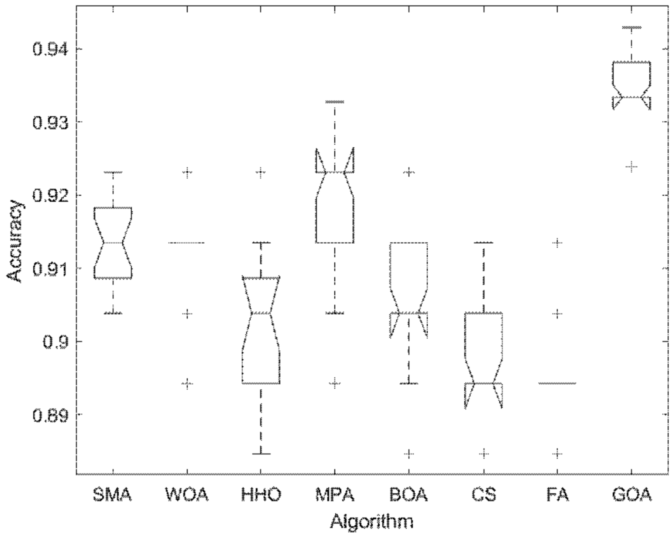
(c) Climate



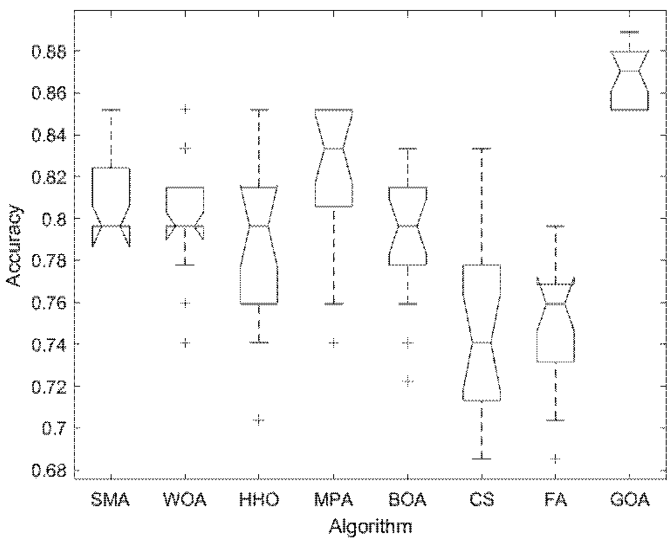
(d) Connectionist



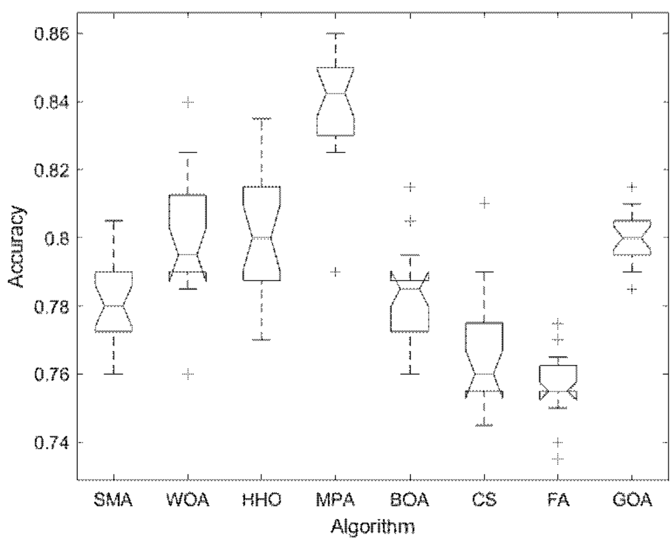
(e) Diabetic



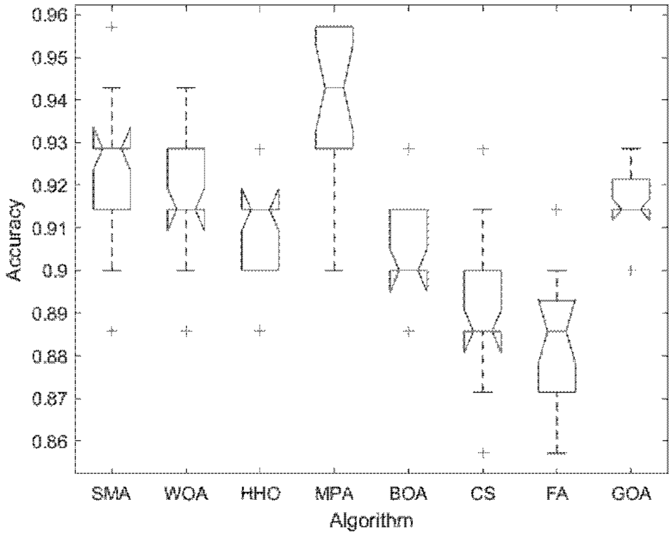
(f) Forest



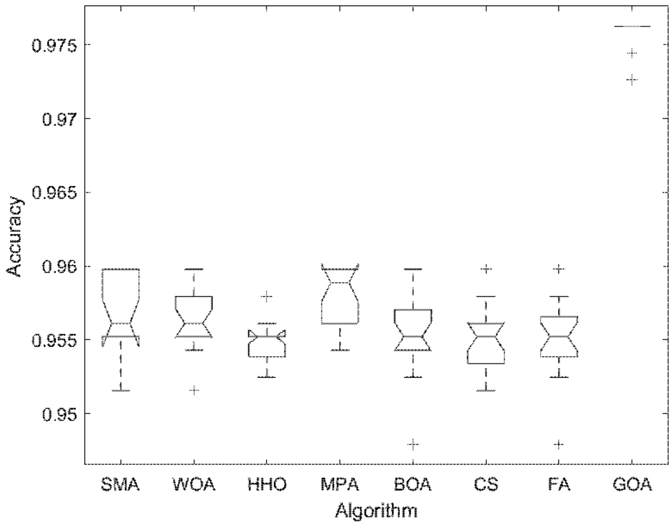
(g) Heart



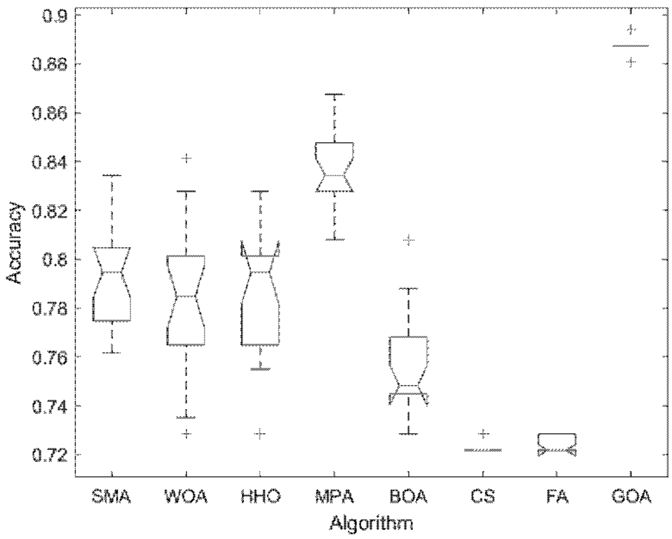
(h) Im



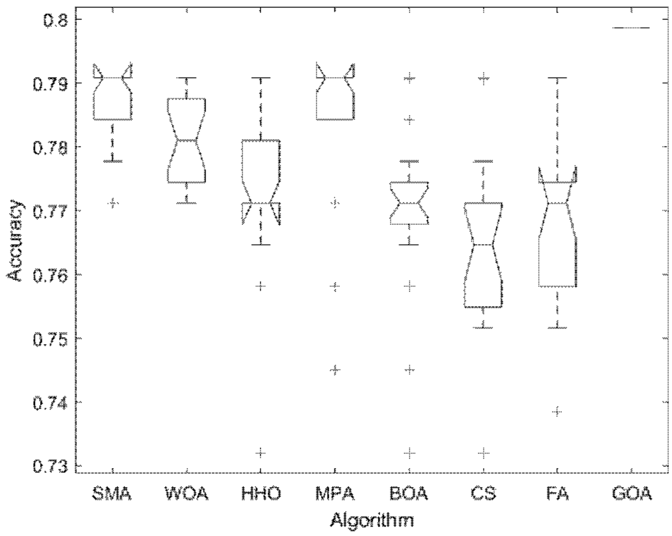
(i) Ionosphere



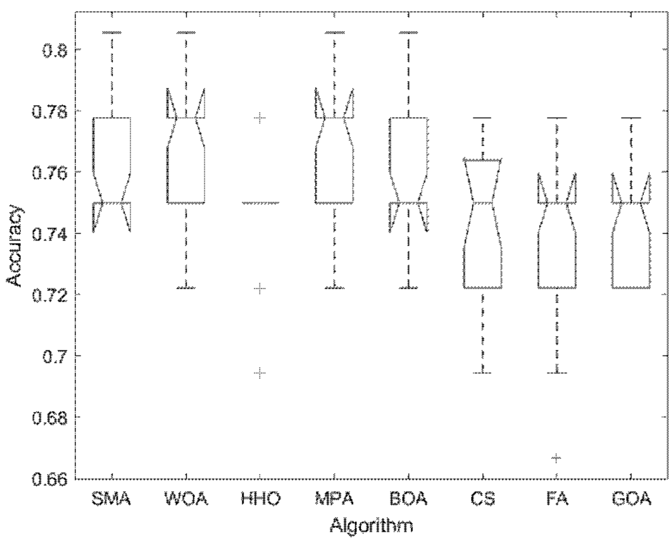
(j) Page



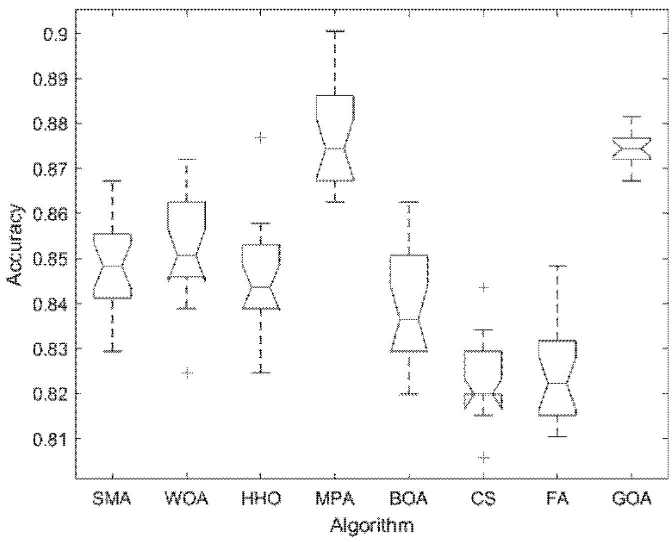
(k) Parkinson



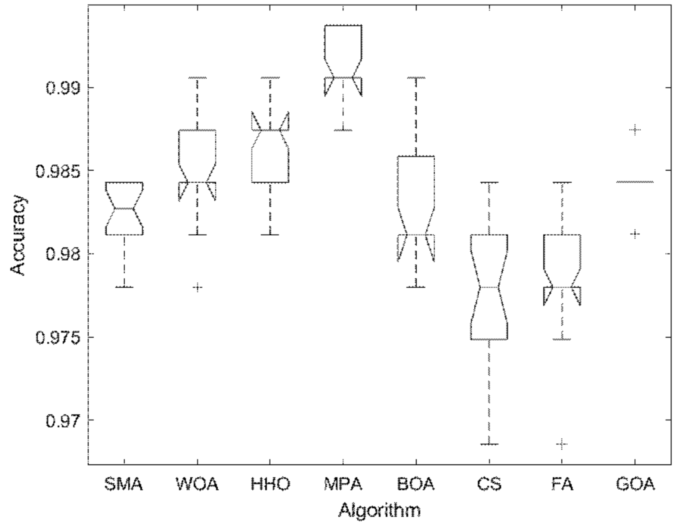
(l) Pima



(m) Planning



(n) QSAR



(o) Semeion

Fig. 4 Box-plot of classification accuracy for eight natural heuristic algorithms.

V. Conclusion

Based on the wrapper feature selection method, this paper compares the grasshopper optimization algorithm (GOA) with seven natural heuristic algorithms, that is SMA, WOA, HHO, MPA, BOA, CS, FA. The convergence curves and box plots of accuracy values of 8 natural heuristic algorithms on 15 datasets were presented. The comprehensive performance indicators are compared. Combining various natural heuristic algorithms proposed, the results are evaluated based on the mean and variation of fitness values, the number of selected features and accuracy. Through algorithm comparison, this paper finds that GOA obtains the highest average fitness value on most datasets. In the number of selected features, GOA wins with absolute advantage, and also holds an advantage in accuracy. The mean and standard deviation data of fitness, selected feature quantity data, accuracy, convergence curve and box plot of accuracy values obtained from 8 algorithms through 15 datasets have great reference value for subsequent research.

References

[1] S. Arora and P. Anand, "Binary Butterfly Optimization Approaches for Feature Selection," *Expert Systems with Applications*, vol. 116, pp. 147-160, 2019.

[2] G. Brown, A. Pocock, M. J. Zhao and M. Lujan, "Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection," *Journal of Machine Learning Research*, vol. 13 no. 1, pp. 27-66, 2012.

[3] M. Tubishat, M. A. M. Abushariah, N. Idris and I. Aljarah, "Improved Whale Optimization Algorithm for Feature Selection in Arabic Sentiment Analysis," *Applied Intelligence*, vol. 49, no. 5, pp. 1688-1707, 2019.

[4] X. Zhang, J. Zhou,Y. Lin and X. Sun, "Deep Learning Based Feature Selection for Hyperspectral Image Classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 6, pp. 2500-2511, 2016.

[5] Y. Li, N. Dong and X. Li, "An Adaptive Feature Selection Approach for Machine Learning Classification," *Journal of Intelligent & Fuzzy Systems*, vol. 35, no. 3, pp. 2675-2686, 2018.

[6] J. Priyadarshini, M. Premalatha, R. Čep, M. Jayasudha and K. Kalita, "Analyzing Physics-Inspired Metaheuristic Algorithms in Feature Selection with K-Nearest-Neighbor," *Applied Sciences*, vol. 13, no. 2, pp. 906, 2023.

[7] S. Fakhraei, H. Soltanian-Zadeh and F. Fotouhi, "Bias and Stability of Single Variable Classifiers for Feature Ranking and Selection," *Expert Systems with Applications*, vol. 41, no. 15, pp. 6945-6958, 2014.

[8] J. D. Li, and H. Liu, "Challenges of Feature Selection for Big Data Analytics," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 9-15, 2017.

[9] N. Zou, W. Zou and Q. Zhao, "Feature Selection Based on Genetic Algorithm and Support Vector Machine," *Arabian Journal for Science and Engineering*, vol. 42, no. 7, pp. 3019-3033, 2017.

[10] X. Chu, Q. Zhao and Y. Liu, "Feature Selection for Hyperspectral Remote Sensing Image Classification by Improved Genetic Algorithm and Classifier Ensemble," *International Journal of Remote Sensing*, vol. 39, no. 6, pp. 1788-1809, 2018.

[11] S. Saremi, S. Mirjalili and A. Lewis, "Grasshopper Optimisation Algorithm: Theory and Application," *Advances in Engineering Software*, vol. 105, pp. 30-47, 2017.

[12] S. Arora and P. Anand, "Chaotic Grasshopper Optimization Algorithm for Global Optimization," *Neural Computing and Applications*, vol. 31, no. 8, pp. 4385-4405, 2019.

[13] Y. Li and L. Gu, "Grasshopper Optimization Algorithm Based on Curve Adaptive and Simulated Annealing," *Application Research of Computers*, vol. 36, no. 12, pp. 3637-3643, 2019.

[14] H. Fan, "Short-Term Electricity Load Forecasting Method Based on Firefly Algorithm," *Electric Power Industry in China*, vol. 54, no. 3, pp. 141-148, 2021.

[15] A. Gandomi, X. S. Yang and A. Alavi, "Erratum to: Cuckoo Search Algorithm: A Metaheuristic Approach to Solve Structural Optimization Problems," *Engineering with Computers*, vol. 29, no. 2, pp. 245, 2013.

[16] L. Wang, T. Shen and C. Li, "A New K-Nearest Neighbor Feature Selection Method for High-Dimensional Data," *Neurocomputing*, vol. 74, no. 12-13, pp. 2195-2203, 2011.