IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received March 20, 2019, accepted April 4, 2019, date of publication April 9, 2019, date of current version April 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2909945

A New Hybrid Seagull Optimization Algorithm for Feature Selection

HEMING JIA, (Member, IEEE), ZHIKAI XING, AND WENLONG SONG
College of Mechanical and Electrical Engineering, Northeast Forestry University, Harbin 150040, China
Corresponding author: Wenlong Song (swl@nefu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 31470714.

ABSTRACT Hybrid algorithms have attracted more and more attention in the field of optimization algorithms. In this paper, three hybrid algorithms are proposed to solve feature selection problems based on seagull optimization algorithm (SOA) and thermal exchange optimization (TEO). In the first algorithm, we take the roulette wheel to choose one of the two algorithms for located updating. Another method is to join the TEO algorithm for optimization after SOA algorithm iteration. The last method is to adopt TEO algorithm's heat exchange formula to improve the seagull attack mode of SOA algorithm, so as to improve the exploitation ability of SOA algorithm. The performance of proposed methods is evaluated on 20 standard benchmark datasets in the UCI repository and compared with three well-known hybrid optimization feature selection methods in the literature. The experimental results illustrate that the proposed algorithm has high efficiency in improving classification accuracy, ensuring the ability of hybrid SOA algorithm in feature selection and classification task information attribute selection, and reducing the CPU time.

INDEX TERMS Feature selection, hybrid optimization, seagull optimization algorithm, thermal exchange optimization.

I. INTRODUCTION

Feature Selection (FS) is a challenging task related to machine learning. Its goal is to reduce the number of features by deleting irrelevant, redundant and noisy data while maintaining acceptable classification accuracy [1]. FS for classification problems is a challenging and computationally expensive process, especially when dealing with high-dimensional data sets [2]. This is because the classification accuracy of the selected feature set must be higher than that of the feature set [3]. The goal of the FS process is to minimize the number of features, which directly reduces the size of the search space/scene and helps machine learning techniques that use only the most important features [4].The goal of FS is to find a subset of M features from a set of N features, which improves the performance of the learning algorithm. FS can be regarded as an optimization problem because it searches for optimal subsets [5]. Generally, three factors should be determined when using a wrapper feature selection model: classifier, feature subset evaluation criteria, and a searching technique to find the best combination of features [6]. Therefore, the research direction is to choose the appropriate strategy to solve the problem of selecting the best target.

In the wrapper methods, the quality of a subset is evaluated based on a classifier model [7], and the commonly used classifiers can be divided into two categories: unsupervised and supervised. The supervised models need to know the class label of each training sample in advance, which results in a better classification result than unsupervised models in most cases [8]. Afterwards, classifications were done using machine learning approaches including artificial immune system [9], [10], support vector machine (SVM) [11]–[13], K-nearest neighbors (KNN) [14]–[16], artificial neural networks (ANN) [17], [18], and case-based technique [19], [20]. The K-nearest neighbor classifier provided a simple non-parametric procedure for the assignment of a class label to the input pattern based on the class labels represented by the K-nearest training samples [21]. Kumar *et al.* a statistical test, ANOVA based on mapreduce was proposed to select the relevant features [22]. After feature selection, mapreduce based K-neatest neighbor classifier was also proposed to classify the microarray data. Ghaemi *et al.* proposed a feature selection method using forest optimization algorithm [23].

The associate editor coordinating the review of this manuscript and approving it for publication was Yanbo Chen.

This algorithm used K-NN to classify features and optimized by forest optimization algorithm. Wang *et al.* proposed to construct a classifier distance matrix and incrementally update the matrix to accelerate the calculation of the relevance criteria in evaluating the quality of the candidate features [24]. A large amount of work can be found in the literature to try to implement the hybrid optimization method to solve the FS problem, and the hybrid optimization algorithm can solve the shortcomings of the optimization algorithm [25]. For example, local optimal problem, global search ability is weak and operation time is too long.

Intelligent optimization algorithm has developed rapidly in recent years. It is divided into swarm intelligence algorithm and genetic evolution algorithm. The most classical algorithm in genetic evolution algorithm is genetic algorithm (GA), the algorithm was proposed by Goldberg DE [26]. Simulated Annealing (SA) was proposed by Kirkpatrick *et al.* [27]. Storn *et al.* proposed the Differential evolution (DE) method [28]. Swarm intelligence optimization algorithm has been studied more and more for its simplicity and strong optimization ability. Particle swarm optimization was proposed by Eberhart *et al.* [29], it was the most classical algorithm in swarm intelligence optimization algorithm. In 2008. Simon proposed the biogeography-based optimization [30], the algorithm was the study of the geographical distribution of biological organisms. In 2010, Yang and She proposed the bat algorithm [31]. The mathematical model of the algorithm was simple and solved practical engineering problems quickly. In 2012, Gandomi and Alavi proposed the krill herd algorithm [32]. This method is based on the simulation of individual grazing behavior of krill. In 2012, Yang proposed the flower pollination algorithm [33]. The method of Levy flight simulation flower pollination was introduced in this algorithm, and the randomness of the particles was strong, which optimized the search space better. In 2015, artificial bee colony algorithm proposed by Karaboga *et al.* [34], which could be optimized by imitating the behavior of bees to collect nectar. In 2016, Mirjalili *et al.* proposed the whale optimization algorithm [35]. The algorithm imitated the way which the whales hunt the food. It can be rotated and straight way to approach the prey, this method can be quickly search the best fitness in the search space. In 2017, Kaveh and Dadras introduced a new optimization algorithm based on Newton's law of cooling [36]. The algorithm mainly described that each agent was considered as a cooling object and by associating another agent as environment, a heat transferring and thermal exchange happened between them. The algorithm has simple structure, good local optimization ability and short operation time. In 2018, Dhiman and Kumar presented a novel bio-inspired algorithm called seagull optimization algorithm(SOA) for solving computationally expensive problems [37]. This algorithm has a good global search ability, it imitates the way of seagull circling over prey, and its attack will affect the local search ability of this algorithm. With the development of swarm intelligence algorithm, each algorithm was adapted to different engineering problems, so the improvement of swarm intelligence algorithm has become the research direction of some scholars.

The improved optimization algorithm is mainly divided into two types: one is to improve the core formula of the optimization algorithm by using the strategy method; the other is to combine the two optimization algorithms. The strategies commonly used by scholars are as follows opposition-based learning [38], Levy-flight [39], Gaussian mutation [40] and so on. Opposition-based learning (OBL) as a new scheme for machine intelligence was introduced by Tizhoosh [41]. The foundation of this new approach were estimates and counter-estimates, weights and opposite weights, and actions versus counter-actions. Mousavirad and Ebrahimpour-Komleh proposed a simple but efficient population-based metaheuristic algorithm called human mental search (HMS) [42]. The mental search of HMS that explored the region around each solution based on Levy flight. Chenhua proposed an improved grey wolf optimization algorithm applied to solve the function optimization problem. The Gaussian disturbance based the rules of survival of the fittest was given on the global optimum of each generation, thus the algorithm could effectively jump out of local minima [43]. The strategy method can effectively improve part of the ability of the algorithm, while the hybrid algorithm can combine the advantages of two or even more methods, so as to better improve the optimization ability of the algorithm [44]. Hybrid optimization algorithms are mainly divided into two types. One is to use a mechanism to select one of the two optimization algorithms for optimization, and use the two algorithms alternately in the iterative process for optimization. The other algorithm used the core formula of one algorithm to improve the other algorithm, and the optimization ability obtained by different improved positions was also different. Guangqian *et al.* proposed a hybrid harmony search-simulated annealing method that combines the advantages of each one of the above-mentioned metaheuristic algorithms [45]. The algorithm integrated the position updating formulas of the two optimization algorithms, and selected different position updating formulas for optimization with a certain mechanism, so as to apply to the hybrid wind/photovoltaic/biodiesel/battery system. Alsaeedan proposed hybrid algorithms for WSD that consist of a self-adaptive genetic algorithm (SAGA) and variants of ant colony optimization (ACO) algorithms: max-minant system (MMAS) and ant colony system (ACS) [46]. Aziz *et al.* examined the ability of two nature inspired algorithms namely: whale optimization algorithm (WOA) and moth-flame optimization (MFO) to determine the optimal multilevel thresholding for image segmentation [47]. The algorithm randomly selected an algorithm for optimization to solve the problem of multi-threshold image segmentation. Karishma *et al.* proposed a congestion control algorithm based on the multi-objective optimization algorithm named PSOGSA for rate optimization and regulating arrival rate of data from every child node to the parent node [48]. Daniel *et al.* propose an optimum Laplacian wavelet mask (OLWM) based fusion using hybrid cuckoo search-grey wolf optimization (HCS-GWO) for multi modal medical image fusion [49]. This algorithm fully integrates the advantages of the two optimization algorithms, so as to solve the multi-objective problem. Orhan and Abdullah proposed an effective new hybrid ant colony algorithm based on crossover and mutation mechanism for no-wait flow shop scheduling with the criterion to minimize the maximum completion time [50]. Garg and Harish presented a hybrid technique named as a PSO-GA for solving the constrained optimization problems [51]. This algorithm combined the advantages of the two optimization algorithms and proposed a new iterative updating formula. According to the above analysis, hybrid optimization algorithm can better use the advantages of different optimization algorithms to solve the various optimization problem.

This paper proposes a hybrid SOA and TEO algorithms for feature selection problem. The proposed algorithm aims to enhance the exploitation of the SOA algorithm. To enhance the exploitation, TEO algorithm is improved in hybridization method, namely SOA-TEO1, SOA-TEO2 and SOA-TEO3. In SOA-TEO1 method, we set the location update formula to randomly select SOA and TEO algorithm for optimization, so as to balance the exploitation and exploration. In SOA-TEO2 method, we firstly use SOA for location iteration, and then use TEO algorithm for final location update. In SOA-TEO3, we improve the TEO algorithm's heat exchange formula to the seagull attack formula in the SOA algorithm, so as to increase the exploitation of the SOA algorithm.

In this paper, Section 2 describes the related work. Section 3 proposes the mathematical model and principle of each basic algorithm. In Section 4, the hybrid SOA-TEO algorithm is described in detail. In Section 5, displays the experimental results and discussion. Finally, the conclusion and future work are given in Section 6.

II. RELATED WORK

In recent years, more and more attention has been paid to hybrid optimization algorithm, and it is used to solve different optimization problems. Feature selection is a common and popular optimization problem. Many scholars have studied different hybrid optimization algorithms to better solve the classification problem of feature selection [52]. Ali *et al.* presented a hybrid optimization method for the FS problem; it combines the slap swarm algorithm (SSA) with the particle swarm optimization [53]. The algorithm randomly selected an algorithm to update the particle position when updating the particle position. Mafarja and Mirjalili proposed the hybrid whale optimization algorithm with simulated annealing for feature selection [54]. This algorithm combines the advantages of the two algorithms, so as to better improve WOA's optimization ability and verify the improved algorithm's classification ability in feature selection. Zorarpac and Özel proposed a new hybrid method which combines artificial bee colony optimization technique with differential evolution algorithm [55]. The proposed hybrid method was better than pure artificial bee colony optimization, and differential evolution. Hariharan *et al.* proposed a new particle swarm optimization assisted biogeography-based algorithm for feature selection [56]. The results were compared with previous work and other meta-heuristic algorithms, which convincingly proved the effectiveness of the proposed feature selection algorithm. Wan *et al.* proposed a feature selection approach based on a modified binary coded ant colony optimization algorithm (MBACO) combined with genetic algorithm (GA) [57]. Experimental results show that the proposed method was robust, adaptive and has good performance. From above mentioned analysis, we can see that hybrid optimization algorithm applied to feature selection has a good effect.

III. MATERIAL AND METHODS

A. THE KNN CLASSIFIER

K-NN is widely used in almost all other fields of machine learning due to its effectiveness and robustness simplicity [58]. It is technically an 'instance based' learning approach that stores the training instances. When a new instance (x) is to be classified, a set of the K most similar training instances is retrieved and used to predict the class of the new instance. The predicted class is the most frequent class among these K nearest neighbours to x. As we describe more clearly later, we use *K-NN* to estimate the accuracy with which a collection of features.

The feature weight vector consists of real values while feature binary vector consisting of either 0 or 1 [59]. A *K-NN* classifier is used to evaluate each weight set evolved by SOA. In addition to feature weight and binary vectors, the value of K used in *K-NN* classifier is also stored in the encoding solution of SOA. Neighbors are calculated using an squared Euclidean distance defined as:

$$D(x, y) = \sum_{i=1}^{m} (x_i - y_i)^2 \tag{1}$$

where x and y are two input vectors. The m is the number of features.

B. SEAGULL OPTIMIZATION ALGORITHM

Seagulls, technically known as the seagull family, are seabirds that cover the globe. There are many kinds of seagulls, with different masses and lengths. Seagulls are omnivores that feed on insects, fish, reptiles, amphibians and earthworms. Most seagulls are covered with white feathers. Seagulls are very clever birds. They use breadcrumbs to attract fish and make the sound of rain on their feet to attract earthworms hiding underground. Seagulls can drink both fresh and salt water [37], that can be seen Fig.1.

Mathematical models of predator migration and attack are discussed. During the migration, the algorithm simulated how a group of gulls moved from one location to another. A seagull must meet the following conditions:

To avoid collisions between adjacent search agents, we use an additional variable, A, to calculate the new search

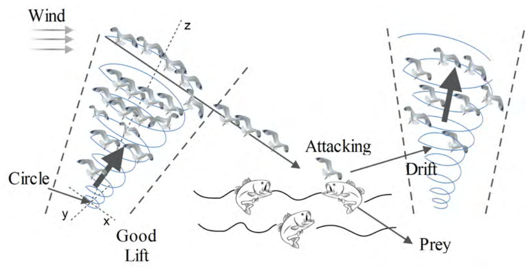

FIGURE 1. Migration and attacking behaviors of seagulls.

agent location.

$$C_s = A \times P_s \tag{2}$$

where C_s represents the position of search agent which does not collide with other search agent, P_s represents the current position of search agent, x indicates the current iteration, and A represents the movement behavior of search agent in a given search space.

$$A = f_c - (x \times (f_c / Max_{iteration})) \tag{3}$$

where f_c is introduced to control the frequency of employing variable A which is linearly decreased from f_c to 0.

After avoiding the collision between neighbours, the search agents are move towards the direction of best neighbor.

$$M_s = B \times (P_{bs}(x) - P_s(x)) \tag{4}$$

where, M_S represents the positions of search agent P_s towards the best fit search agent P_{bs}. The behavior of B is randomized which is responsible for balancing between exploration and exploitation properly. B is calculated as:

$$B = 2 \times A^2 \times rd \tag{5}$$

where, rd is a random number lies in the range of [0,1].

Lastly, the search agent can update its position with respect to best search agent by:

$$D_s = |C_s + M_s| \tag{6}$$

where, D_s represents the distance between the search agent and best fit search agent.

This development is designed to take advantage of the history and experience of the search process. When attacking prey, the spiraling action takes place in the air. This behavior in the x, y, and z planes is described as follows:

$$x' = r \times \cos(k) \tag{7}$$
$$y' = r \times \sin(k) \tag{8}$$
$$z' = r \times k \tag{9}$$
$$r = u \times e^{kv} \tag{10}$$

where r is the radius of each turn of the spiral, k is a random number in range [$0 \leq k \leq 2\pi$]. u and v are constants to define the spiral shape, and e is the base of the natural logarithm. The updated position of search agent is calculated using (6) - (9).

$$P_s(x) = (D_s \times x' \times y' \times z') + P_{bs}(x) \tag{11}$$

FIGURE 2. Hot iron objects, transferring heat to the surrounding environment.

where, P_s saves the best solution and updates the position of other search agents.

C. THERMAL EXCHANGE OPTIMIZATION

TEO is a new optimization algorithm based on Newton's law of cooling, which makes the rate of heat loss of an object directly proportional to the temperature difference between the object and its surroundings [36]. The hot iron objects transferring heat to the surrounding environment is shown in Fig. 2.

TEO's algorithm defines some agents as cooling objects, while others represent the environment. The temperature formula between the updated objects can be defined as:

$$T_i^{env} = (1 - (c_1 + c_2 \times (1 - t)) \times random) \times T_i'^{env} \tag{12}$$
$$t = \frac{l}{L} \tag{13}$$

where c_1, c_2 are the controlling variables, $T_i'^{env}$ is the previous temperature of the object, which is modified to T_i^{env}. l is the current iteration number, L is the max iteration number.

According to the previous steps and Eq.8, new temperature of each object is updated by

$$T_i^{new} = T_i^{env} + (T_i^{old} - T_i^{env}) \exp(-\beta t) \tag{14}$$
$$\beta = \frac{\cos t(object)}{\cos t(worst \quad object)} \tag{15}$$

where, the nature when an object has lower β, it exchanges the temperature slightly. The value of β for each object is evaluated according Eq.14.

To prevent the temperature of the object from falling into local optimum, set the parameter Pro. It is specified whether a component of each cooling object must be changed or not. If rand<Pro, one dimension of the i th agent is selected randomly and its value is regenerated as follows:

$$T_{i,j} = T_{i,\min} + rand \times (T_{j,\max} - T_{j,\min}) \tag{16}$$

where, $T_{i,j}$ is the j th variable of the ith agent. $T_{j,\max}$ and $T_{j,\min}$ are the lower and upper bounds of the j th variable.

IV. PROPOSED METHOD

Feature selection is a binary optimization problem, where solutions are restricted to the binary {0,1} values.

Feature selection can be considered as a multi-objective optimization problem where two contradictory objectives are to be achieved; minimal number of selected features and higher classification accuracy [60]. The smaller is the number of features in the solution and the higher the classification accuracy, the better the solution is. Each solution is evaluated according to the proposed fitness function, which depends on the KNN classifier to get the classification accuracy of the solution and on the number of selected features in the solution. We create a population of particles on N dimensions in the feature space and set the parameters of the optimization algorithm. And then, evaluate fitness using K-NN and update the best fitness. According to reference [61], it is selected as the fitness function of K-NN. Finally, terminate if termination criterion satisfied, outputting the selected subset of features.

In order to balance between the number of selected features in each solution (minimum) and the classification accuracy (maximum), the fitness function in Eq. 17 is used in both SOA algorithms to evaluate search agents.

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|N|} \tag{17}$$

where $\gamma_R(D)$ represents the classification error rate of a given classier(the K-nearest neighbor (KNN) classifier is used here). Furthermore, $|R|$ is the cardinality of the selected subset and $|N|$ is the total number of features in the dataset, α and β are two parameters corresponding to the importance of classification quality and subset length, $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$ adopted from [60].

A. THE K-FOLD CROSS VALIDATION

The K-fold cross validation is applied due to its properties being simple, easy and use all the data for training and validation. The first mechanism is to create the K-fold partition of the entire data set, repeat K times, use K1 folding for training, use left folding for training verification, and finally average the error rate of K times experiment [62].

For all target-oriented validation strategies, the procedure is comparable to the random K-fold validation (which gives a biased estimate of prediction performance): models are repeatedly trained by using the data of all except one fold and testing the model performance for the held-back data. In all of the experiments 10-fold cross validation is used to estimate the accuracy of each learned classifier. Some empirical results are reported in the following sections.

B. HYBRID SEAGULL OPTIMIZATION ALGORITHM (HSOA)

In this subsection, the Hybrid SOA is described in detail. SOA algorithm has a good global search ability, while TEO algorithm has a strong local search ability. In order to improve the local search ability of SOA algorithm, this paper proposes three hybrid optimization algorithms.

Firstly, one of the two algorithms is randomly selected for optimization in the way of roulette. The algorithm randomly selects Eq.11 or Eq.16 for the position update of this iteration, and makes use of the advantages of the two formulas

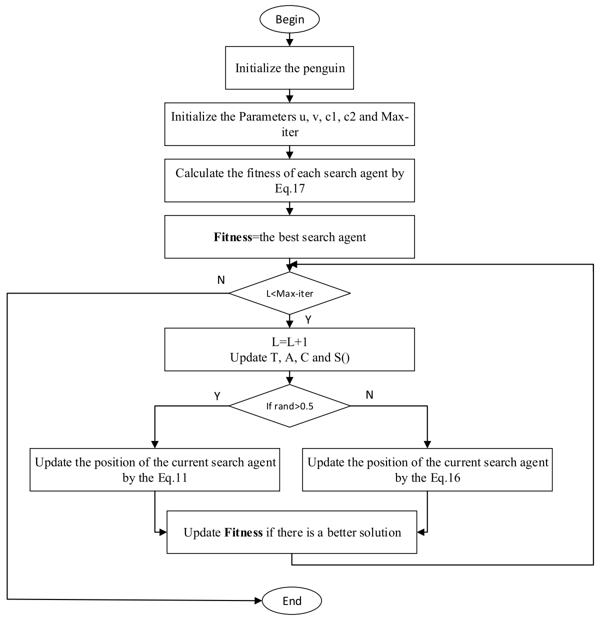

FIGURE 3. The flowchart of the SOA-TEO1.

to update the position, so as to strengthen the optimization ability. This method will randomly conduct global and local search to avoid falling into the problem of local optimization. This approach is called SOA-TEO1. The flowchart of the SOA-TEO1 can be seen from fig.3.

Secondly, this algorithm will add TEO algorithm's location update formula after the location update formula of SOA algorithm. After the iteration of SOA algorithm, TEO algorithm will enhance its local optimization ability, and this algorithm will add Eq.16 after Eq.11. This approach is called SOA-TEO2. The flowchart of the SOA-TEO2 can be seen from fig.4.

Finally, the expression in TEO algorithm is improved to the seagull attack formula in SOA algorithm to improve the local search ability of seagull algorithm. This approach is called SOA-TEO2. We apply the idea of thermal exchange in TEO algorithm to enhance the exploitation of seagulls. In the Eq.13, β exchanges the temperature slightly between objects so as to get close to the target object quickly. So, β is improved Eq.3 so that seagulls can better move towards prey. Its mathematical formula is as follows:

$$M_s = B \times (P_{bs}(x) - P_s(x)) \times \exp(-\beta t) \tag{18}$$

The flowchart of the SOA-TEO3 can be seen from fig.5.

V. FUNCTION OPTIMIZATION EXPERIMENT

When applied to solve specific problems, different nature inspired algorithms have different optimization performance due to the difference in their search strategies and mathematical formulation. Therefore, CEC2015 standard functions [63] are used to test the improved SOA-TEO algorithm. The comparative study of the proposed algorithm and other different optimization algorithms such as WOA, SA, ABC,

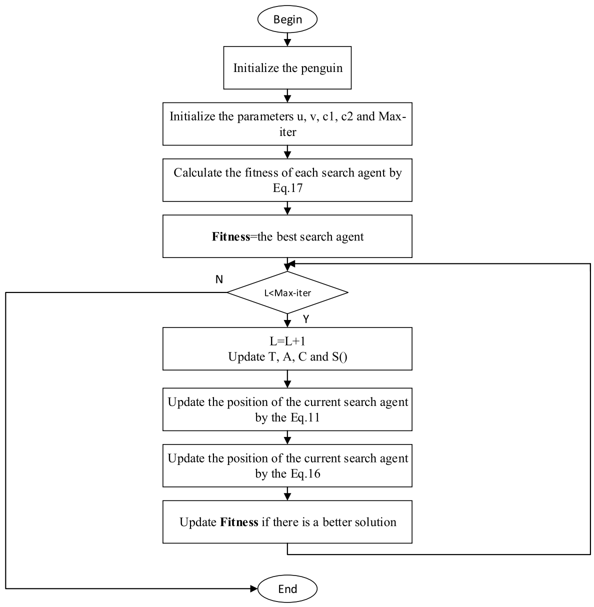

FIGURE 4. The flowchart of the SOA-TEO2.

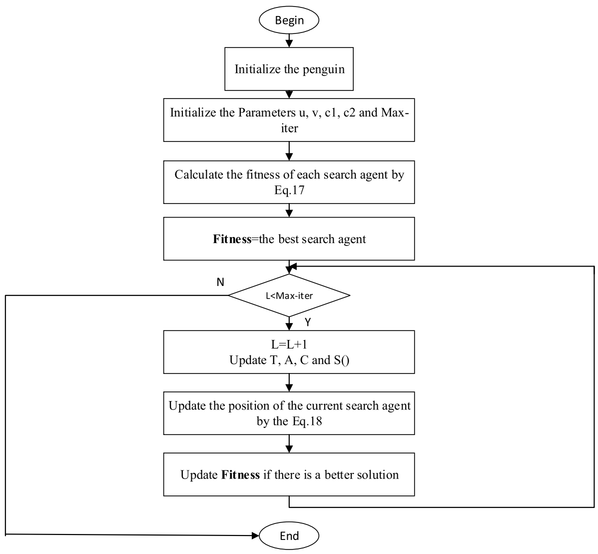

FIGURE 5. The flowchart of the SOA-TEO3.

DE, PSO, BA, GA and BBO is investigated. The parameter settings of the optimization algorithm are shown in table 1. In order to test the performance of the algorithm proposed in this paper. The detailed description of CEC 2015 benchmark test functions is presented in Table 2. To eliminate stochastic discrepancy, each experiment is independently run with each algorithm 30 times for comparisons. To obtain fair results, all the implementations are conducted under the same conditions. Population size and maximum generation are set to 30 and 500 respectively.

Table 3-5 shows the results of each optimization algorithm on the CEC2015 test function. The table records the mean and variance of each optimization algorithm running 30 times separately. In order to observe the data in table 3-5 more intuitively, we perform logarithmic operation on the data in the table, and the results are shown in fig. 6 and fig. 7. The fig.6 show the logarithmic curve of the compared algorithm's mean. The fig.7 show the logarithmic curve of the compared algorithm's std. As can be seen from the data in the table

TABLE 1. Parameters and references of the comparison algorithms.

Algorithm	Parameters	Value
SOA [37]	u	1
	v	0.001
TEO [36]	c_1	2
	c_2	2
WOA [65]	a	[0.2]
	b	1
	l	[-1,1]
SA [66]	T	0.93
ABC [67]	φ	[-1,1]
DE [68]	Crossover rate	0.3
PSO [69]	Swam size	200
	Cognitive, social	2,2
	Acceleration	0.95-0.4
	Inertial weight	
BA [70]	β	(0,1)
GA [71]	Genes code	[0,1]
BBO [72]	E	1
	I	1

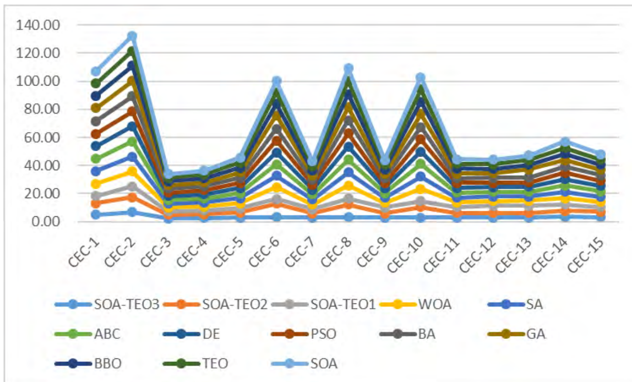

FIGURE 6. The logarithmic curve of the compared algorithm's mean.

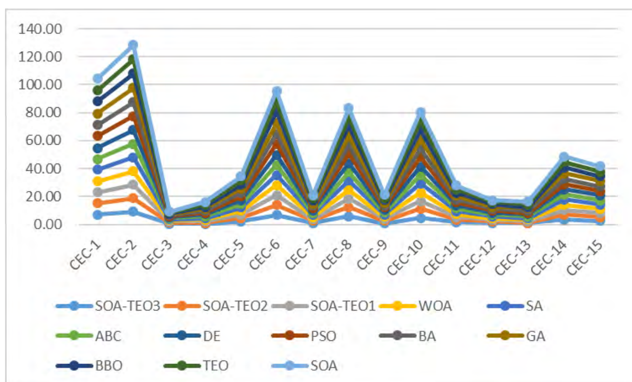

FIGURE 7. The logarithmic curve of the compared algorithm's std.

TABLE 2. CEC 2015 benchmark test functions.

No.	Functions	Related basic functions	Dim	fmin
CEC-1	Rotated Bent Cigar Function	Bent Cigar Function	30	100
CEC-2	Rotated Discus Function	Discus Function	30	200
CEC-3	Shifted and Rotated Weierstrass Function	Weierstrass Function	30	300
CEC-4	Shifted and Rotated Schwefel's Function	Schwefel's Function	30	400
CEC-5	Shifted and Rotated Katsuura Function	Katsuura Function	30	500
CEC-6	Shifted and Rotated HappyCat Function	HappyCat Function	30	600
CEC-7	Shifted and Rotated HGBat Function	HGBat Function	30	700
CEC-8	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	Griewank's Function Rosenbrock's Function	30	800
CEC-9	Shifted and Rotated Expanded Scaer's F6 Function	Expanded Scaer's F6 Function	30	900
CEC-10	Hybrid Function 1 (N = 3)	Schwefel's Function Rastrigin's Function High Conditioned Elliptic Function	30	1000
CEC-11	Hybrid Function 2 (N = 4)	Griewank's Function Weierstrass Function Rosenbrock's Function Scaer's F6 Function	30	1100
CEC-12	Hybrid Function 3 (N = 5)	Katsuura Function HappyCat Function Expanded Griewank's plus Rosenbrock's Function Schwefel's Function Ackley's Function	30	1200
CEC-13	Composition Function 1 (N = 5)	Rosenbrock's Function High Conditioned Elliptic Function Bent Cigar Function Discus Function High Conditioned Elliptic Function	30	1300
CEC-14	Composition Function 2 (N = 3)	Schwefel's Function Rastrigin's Function High Conditioned Elliptic Function	30	1400
CEC-15	Composition Function 3 (N = 5)	HGBat Function Rastrigin's Function Schwefel's Function Weierstrass Function High Conditioned Elliptic Function	30	1500

TABLE 3. The fitness value of the CEC2015 benchmark test function.

Func.	SOA-TEO1		SOA-TEO2		SOA-TEO3		WOA		SA	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
CEC-1	1.05E+05	1.54E+07	1.76E+08	1.47E+08	1.07E+05	3.53E+07	6.92E+08	7.02E+07	8.02E+08	2.86E+08
CEC-2	6.70E+06	1.70E+09	3.04E+10	5.26E+09	6.70E+07	3.53E+09	4.36E+10	4.70E+09	4.20E+10	6.82E+09
CEC-3	3.20E+02	7.12E-02	3.21E+02	1.51E-01	3.55E+02	9.57E-02	3.72E+02	1.43E-01	4.70E+02	2.10E-01
CEC-4	4.10E+02	1.75E+00	5.30E+02	9.03E+00	4.50E+02	1.29E+01	6.12E+02	1.61E+01	9.96E+02	9.70E+00
CEC-5	9.81E+02	1.50E+02	3.80E+03	3.30E+02	9.91E+02	2.40E+02	4.20E+03	3.97E+02	3.88E+03	3.52E+02
CEC-6	2.05E+03	4.75E+06	4.52E+09	2.33E+07	2.10E+03	9.03E+06	1.86E+08	1.14E+07	1.51E+08	2.71E+07
CEC-7	7.02E+02	1.24E+01	1.78E+03	4.37E+01	8.82E+02	1.40E+01	2.56E+03	2.70E+01	3.25E+03	4.99E+01
CEC-8	1.47E+03	1.03E+06	1.34E+09	3.25E+06	1.47E+04	1.05E+06	2.41E+09	1.63E+06	1.65E+09	3.78E+06
CEC-9	1.00E+03	7.30E+00	1.33E+03	4.77E+01	1.00E+04	2.45E+01	2.72E+03	3.46E+01	2.89E+03	5.15E+01
CEC-10	1.23E+03	4.82E+04	2.97E+07	3.09E+06	1.36E+04	4.14E+05	7.35E+08	8.13E+05	5.23E+08	4.49E+06
CEC-11	1.35E+03	4.52E+01	1.77E+03	7.94E+01	1.32E+04	6.39E+01	1.80E+03	1.02E+02	3.31E+03	9.80E+01
CEC-12	1.30E+03	1.03E+01	1.49E+03	1.75E+01	1.34E+05	1.22E+01	1.62E+03	1.28E+01	1.99E+03	3.25E+01
CEC-13	1.30E+03	8.67E+00	1.54E+03	1.49E+00	1.55E+05	1.86E+01	2.43E+03	3.03E+01	1.90E+03	2.48E+00
CEC-14	3.22E+03	2.32E+03	2.23E+04	4.24E+03	3.22E+04	2.92E+03	2.58E+04	4.76E+03	3.01E+04	5.46E+03
CEC-15	1.60E+03	3.36E+02	9.01E+03	6.49E+02	1.80E+03	7.10E+02	4.29E+03	1.34E+03	6.70E+03	1.05E+03

and figures, the hybrid algorithm proposed in this paper has improved its optimization ability, and the SOA-TEO3 method has the best optimization result and the best stability. In order to further observe the results of each comparison algorithm, we conduct the Wilcoxon rank-sum test. When p-values are less than 0.05, it can be determined that the results of

TABLE 4. The fitness value of the CEC2015 benchmark test function.

Func.	ABC		DE		PSO		BA		GA	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
CEC-1	5.69E+08	3.69E+07	9.45E+08	1.01E+08	6.01E+08	3.89E+08	9.20E+08	5.60E+07	2.17E+09	1.79E+08
CEC-2	6.20E+10	4.05E+09	5.04E+10	8.97E+09	6.34E+10	1.06E+10	5.88E+10	8.01E+09	7.01E+10	1.62E+10
CEC-3	5.53E+02	1.47E-01	4.67E+02	2.31E-01	4.37E+02	2.77E-01	5.85E+02	1.59E-01	3.22E+02	2.65E-01
CEC-4	6.11E+02	2.39E+01	8.43E+02	2.35E+01	8.28E+02	1.28E+01	6.73E+02	3.48E+01	5.46E+02	3.05E+01
CEC-5	4.30E+03	2.97E+02	4.53E+03	5.04E+02	4.85E+03	4.51E+02	4.24E+03	5.10E+02	4.74E+03	7.28E+02
CEC-6	2.00E+08	1.37E+07	2.54E+08	2.25E+07	2.38E+08	4.99E+07	2.21E+08	2.05E+07	3.79E+09	2.83E+07
CEC-7	3.12E+03	2.49E+01	2.81E+03	3.91E+01	3.28E+03	7.30E+01	2.65E+03	3.36E+01	1.81E+03	4.70E+01
CEC-8	2.00E+09	1.08E+06	1.34E+09	2.62E+06	2.58E+09	6.33E+06	2.08E+09	1.38E+06	2.21E+09	4.09E+06
CEC-9	2.90E+03	3.38E+01	3.01E+03	4.88E+01	2.64E+03	5.86E+01	2.99E+03	3.41E+01	1.33E+03	6.68E+01
CEC-10	6.81E+08	7.89E+05	5.84E+08	1.59E+06	4.78E+08	6.17E+06	6.20E+08	1.04E+06	2.58E+09	2.62E+06
CEC-11	2.80E+03	1.24E+02	2.97E+03	1.99E+02	2.27E+03	1.18E+02	2.95E+03	1.47E+02	2.00E+03	3.44E+02
CEC-12	2.42E+03	1.58E+01	1.93E+03	1.47E+01	1.62E+03	5.66E+01	2.72E+03	2.23E+01	1.76E+03	1.55E+01
CEC-13	2.12E+03	3.24E+01	2.37E+03	5.77E+01	1.59E+03	3.14E+00	2.55E+03	4.93E+01	5.44E+05	1.15E+02
CEC-14	3.64E+04	3.59E+03	2.84E+04	5.10E+03	4.37E+04	5.83E+03	3.58E+04	5.80E+03	2.23E+04	8.73E+03
CEC-15	7.89E+03	1.27E+03	4.55E+03	2.49E+03	5.60E+03	1.53E+03	6.68E+03	2.27E+03	1.28E+04	4.77E+03

TABLE 5. The fitness value of the CEC2015 benchmark test function.

Func.	BBO		SOA		TEO	
	Mean	Std.	Mean	Std.	Mean	Std.
CEC-1	7.85E+08	4.88E+08	4.76E+08	8.13E+07	7.46E+08	2.18E+08
CEC-2	5.89E+10	1.53E+10	3.36E+10	1.42E+10	6.51E+10	1.83E+10
CEC-3	5.85E+02	3.57E-01	3.21E+02	1.66E-01	5.34E+02	2.76E-01
CEC-4	9.29E+02	1.60E+01	5.21E+02	6.90E+01	6.29E+02	4.44E+01
CEC-5	3.93E+03	7.90E+02	3.65E+03	9.59E+02	4.56E+03	1.07E+03
CEC-6	1.42E+08	9.52E+07	1.28E+08	3.14E+07	1.50E+08	3.42E+07
CEC-7	2.59E+03	1.31E+02	1.78E+03	6.05E+01	2.02E+03	5.81E+01
CEC-8	2.53E+09	1.10E+07	1.34E+09	1.59E+06	2.02E+09	5.47E+06
CEC-9	2.59E+03	1.04E+02	1.62E+03	5.16E+01	1.79E+03	7.89E+01
CEC-10	4.48E+08	1.11E+07	4.41E+08	1.50E+06	7.40E+08	4.04E+06
CEC-11	3.20E+03	2.13E+02	1.77E+03	2.14E+02	3.46E+03	6.56E+02
CEC-12	1.69E+03	9.53E+01	1.51E+03	2.96E+01	1.75E+03	1.68E+01
CEC-13	2.90E+03	3.28E+00	1.56E+03	8.29E+01	2.48E+03	1.60E+02
CEC-14	3.63E+04	1.13E+04	2.23E+04	9.93E+03	3.57E+04	9.88E+03
CEC-15	4.15E+03	2.42E+03	4.09E+03	2.42E+03	7.68E+03	5.59E+03

SOA-ITEO3 are significantly superior to the other approach. If not, the obtained improvements are not statistically significant. Some obtained p-values are reported in Table 6. It can be seen from Wilcoxon tests in table 6 that the advantages of SOA-TEO3 algorithm are very obvious. The ISOA algorithm can not only solve single-dimensional mathematical functions but also deal with multi-dimensional mathematical functions effectively. Therefore, SOA-TEO3 algorithm has better optimization ability and can solve more complex feature selection problem.

VI. EXPERIMENTS AND RESULTS

In this section, the experimental data are simulated by matlab 2017. In order to test the optimization ability of each algorithm, 20 standard datasets are selected from the UCI data repository [64]. Table 7 records the description of each data. We use KNN classifier based on Euclidean distance matrix (where K = 3) to generate the best reduction. The maximum number of iterations is 100, and the initial population is 10. In order to prevent the randomness of the test results, each algorithm ran independently for 30 times on an Intel Core i7, 2.4GHz and 8GB of RAM. The average value and variance of the results of each algorithm are recorded.

This section firstly compares the three proposed hybrid algorithms, which are SOA-TEO1, SOA-TEO2, SOA-TEO3. In order to test the optimal values of all algorithms, all comparison algorithms are recorded in one table. And then, the proposed approaches are also compared to optimization feature selection methods including WOA-SA, ABC-DE, PSO-BBO, WOA, SA, ABC, DE, PSO, BA, GA and BBO based on the following criteria:

1) Accuracy of classification using selected features on the test data set. The mean precision and variance of 30 runs is calculated.
2) The average number of selected attributes and CPU time of each compared algorithm.
3) The p-values of the Wilcoxon ranksum test [73] of the proposed algorithm with other compared algorithm.

TABLE 6. The wilcoxon rank-sum test of the CEC2015 benchmark test function.

Function	SOA-TEO1	SOA-TEO2	WOA	SA	ABC	DE	PSO	BA	GA	BBO	SOA	TEO
CEC-1	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-2	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-3	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-4	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-5	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-6	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-7	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-8	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-9	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-10	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-11	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-12	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-13	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-14	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CEC-15	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05

TABLE 7. The datasets used in the experiments.

S.no.	Datasets	Instances	Number of classes (k)	Number of features (d)	Size of classes
1	Iris	150	3	4	50,50,50
2	Wine	178	3	13	59,71,48
3	Glass	214	6	9	29,76,70,17,13,9
4	Diabetes	768	2	8	268,500
5	Heartstatlog	270	2	13	150,120
6	Ionosphere	351	2	34	126,225
7	Sonar	208	2	60	97,111
8	Vehicle	846	3	18	199,217,218,212
9	Balancescale	625	3	4	49,288,288
10	CMC	1473	3	9	629,333,511
11	Cancer	683	2	9	444,239
12	Seeds	210	3	7	70,70,70
13	Blood	748	2	4	570,178
14	Aggregation	788	7	2	170,34,273,102,130,45,34
15	Vowel	871	6	3	72,89,172,151,207,180
16	WBC	683	2	9	444,239
17	Bupa	345	2	6	145,200
18	Jain	373	2	2	276,97
19	Thyroid	215	3	5	150,35,30
20	WDBC	569	2	30	357,212

A. COMPARISON WITH SOA-TEO1, SOA-TEO2, SOA-TEO3, SOA AND TEO

The perform of the SOA, TEO, SOA-TEO1, SOA-TEO2 and SOA-TOE3 over the two objectives (average selection size and classification accuracy) and the CPU time is compared in this section. Table 8 compares the mean and standard deviation of the original SOA and TEO with the performance of different hybrid algorithms for the 20 datasets. Table 9 show the average selection of the compared algorithms. Table 10 how the CPU time of the algorithms.

It can be seen from table 8, the three proposed hybrid optimization algorithm is very competitive in feature selection. The SOA-TEO3 have the best mean fitness value for 20 datasets. The results of SOA algorithm are generally better than those of TEO algorithm, but when multi-feature database is processed, such as Sonar and Ionosphere, its optimal value is basically the same, indicating that the local optimization ability of SOA algorithm is weak and cannot continue to optimize the search range. When database features are relatively few, such as Aggregation and WDBC, the results of TEO algorithm are better than those of SOA algorithm, indicating that TEO algorithm has better local optimization ability than SOA algorithm. Therefore, it is necessary to propose a hybrid algorithm, which can better combine the advantages of the two algorithms to solve the problem of feature selection. It can be seen from the results of the 3 comparative hybrid algorithms that their effects are better than the traditional SOA and TEO algorithms. It is verified that the hybrid algorithm can effectively solve the shortcomings of the traditional optimization algorithm. Among the three hybrid optimization

TABLE 8. Comparison of statistical results obtained using SOA, TEO, SOA-TEO1, SOA-TEO2, SOA-TEO3 feature selection algorithms for 20 datasets.

	SOA		TEO		SOA-TEO1		SOA-TEO2		SOA-TEO3	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Iris	0.6744	0.0902	0.6765	0.1039	0.8374	0.0198	0.7957	0.0142	**0.9030**	**0.0085**
Wine	0.6825	0.0854	0.6175	0.1063	0.8590	0.0169	0.7890	0.0146	**0.9094**	**0.0019**
Glass	0.7444	0.0921	0.6803	0.1022	0.8185	0.0131	0.7364	0.0132	**0.9346**	**0.0084**
Diabetes	0.7282	0.0876	0.6178	0.1045	0.8960	0.0146	0.7029	0.0062	**0.9382**	**0.0061**
Heartstatlog	0.6233	0.0921	0.6470	0.1024	0.8184	0.0105	0.7340	0.0106	**0.9399**	**0.0071**
Ionosphere	0.6105	0.0890	0.6221	0.1019	0.8325	0.0164	0.7799	0.0090	**0.9053**	**0.0019**
Sonar	0.6996	0.0873	0.6080	0.1015	0.8074	0.0159	0.7976	0.0075	**0.9397**	**0.0091**
Vehicle	0.6912	0.0891	0.6584	0.1072	0.8052	0.0192	0.7527	0.0081	**0.9171**	**0.0028**
Balancescal e	0.7279	0.0863	0.6436	0.1000	0.8911	0.0199	0.7914	0.0075	**0.9263**	**0.0085**
CMC	0.7245	0.0892	0.6716	0.1081	0.8499	0.0117	0.7634	0.0141	**0.9025**	**0.0019**
Cancer	0.7402	0.0931	0.6502	0.1035	0.8199	0.0132	0.7187	0.0115	**0.9424**	**0.0026**
Seeds	0.7207	0.0866	0.6057	0.1019	0.8136	0.0115	0.7448	0.0053	**0.9209**	**0.0083**
Blood	0.6569	0.0871	0.6697	0.1001	0.8733	0.0112	0.7398	0.0090	**0.9895**	**0.0063**
Aggregation	0.7189	0.0878	0.7537	0.1086	0.8795	0.0109	0.7451	0.0074	**0.9503**	**0.0080**
Vowel	0.7103	0.0902	0.6641	0.1012	0.8549	0.0116	0.7852	0.0107	**0.9501**	**0.0040**
WBC	0.7200	0.0918	0.6224	0.1000	0.8880	0.0133	0.7787	0.0122	**0.9249**	**0.0080**
Bupa	0.6771	0.0922	0.6931	0.1034	0.8857	0.0151	0.7439	0.0058	**0.9030**	**0.0038**
Jain	0.6783	0.0945	0.6663	0.1003	0.8281	0.0155	0.7813	0.0074	**0.9242**	**0.0024**
Thyroid	0.7000	0.0870	0.6901	0.1038	0.8959	0.0198	0.7405	0.0148	**0.9506**	**0.0047**
WDBC	0.6748	0.0867	0.7418	0.1008	0.8563	0.0111	0.7813	0.0136	**0.9455**	**0.0061**

TABLE 9. The average number of selected attributes obtained using SOA, TEO, SOA-TEO1, SOA-TEO2, SOA-TEO3 feature selection algorithms for 20 datasets.

	Attributes	Instances	SOA	TEO	SOA-TEO1	SOA-TEO2	SOA-TEO3
Iris	4	150	2.9239	2.7116	2.9445	2.5931	**1.5927**
Wine	13	178	9.4580	12.5810	5.1881	4.1701	**3.8156**
Glass	9	214	7.2145	6.8069	5.1863	4.6945	**4.4245**
Diabetes	8	768	7.1241	7.0664	4.8794	3.6874	**2.8095**
Heartstatlog	13	270	9.9949	12.4321	10.0784	8.6588	**2.6960**
Ionosphere	34	351	24.5768	27.6793	13.0112	12.8244	**7.5890**
Sonar	60	208	47.2785	54.6374	31.2908	30.7422	**27.1744**
Vehicle	18	846	16.1150	13.3472	8.0550	7.3564	**6.0793**
Balancescale	4	625	2.9373	3.5268	2.7010	2.5465	**1.9172**
CMC	9	1473	8.8224	7.6045	6.5604	5.6514	**4.0961**
Cancer	9	683	8.9938	7.5147	4.6305	3.5144	**2.4272**
Seeds	7	210	5.5197	6.6153	5.6678	4.7477	**2.2566**
Blood	4	748	3.9925	3.3376	2.6463	2.9995	**1.7527**
Aggregation	2	788	1.4951	1.5060	1.3179	**1.1904**	1.6366
Vowel	3	871	2.3640	2.1964	2.4085	2.1801	**1.4358**
WBC	9	683	6.0148	6.7142	5.5952	**2.1395**	3.0838
Bupa	6	345	5.8479	4.7311	4.5965	5.9151	**2.0365**
Jain	2	373	1.9338	1.5622	1.3365	**1.1719**	1.2016
Thyroid	5	215	4.2364	3.8835	2.6121	2.6749	**1.6226**
WDBC	30	569	25.4813	29.7266	10.6186	11.6530	**9.1142**
Total	249	-	202	216	131	122	**89**

algorithms, SOA-TEO 3 has the best results and SOA-TEO 1 has the worst results. It shows that SOA-TEO 3 algorithm can better increase the exploitation of SOA algorithm.

As can be seen from the results in table 9, the SOA-TEO 3 algorithm has the fewest selected feature attributes and is obviously superior to other comparison algorithms. When SOA-TEO2 algorithm has few characteristic properties, such as Aggregation and Jain, its results are relatively excellent. The feature attributes of the hybrid algorithm are all less than those of the original SOA and TEO algorithm, which indicates that the hybrid algorithm can overcome the shortcomings of the original algorithm, so as to better improve the optimization ability of the algorithm. From the sum of the attributes in the 20 databases, the SOA-TEO 3 algorithm has the least characteristic attributes, indicating that the SOA-TEO3 algorithm has the best performance. According to the CPU time in table 10, SOA-TEO3 algorithm has the shortest CPU time, followed by SOA-TEO1 algorithm

TABLE 10. CPU time obtained using SOA, TEO, SOA-TEO1, SOA-TEO2, SOA-TEO3 feature selection algorithms for 20 datasets.

	SOA	TEO	SOA-TEO1	SOA-TEO2	SOA-TEO3
Iris	3.6561	2.1716	3.1395	4.8871	**1.5935**
Wine	5.7029	3.1672	5.0460	6.5380	**2.1337**
Glass	5.7900	4.0088	5.3966	5.2070	**2.0249**
Diabetes	4.4958	3.9650	5.5575	4.8629	**2.1224**
Heartstatlog	2.0176	1.0962	3.4932	4.5410	**0.9949**
Ionosphere	4.7840	3.1208	4.8375	3.1080	**2.1184**
Sonar	4.8710	3.0448	4.1914	5.2556	**1.3089**
Vehicle	3.8835	2.1133	4.8985	3.5966	**2.1220**
Balancescale	5.0097	3.0463	6.5051	7.0209	**1.5263**
CMC	3.6354	2.4688	4.2964	4.6206	**1.1946**
Cancer	5.4580	4.0285	5.9266	4.4106	**2.4039**
Seeds	6.2071	4.2873	6.6122	6.8312	**1.0587**
Blood	3.9970	2.5904	3.2581	4.3679	**1.5874**
Aggregation	4.4122	3.7274	6.7104	7.9908	**1.9315**
Vowel	4.2558	2.2790	5.0950	5.4262	**1.1442**
WBC	3.2560	2.5157	4.7402	3.3465	**0.9939**
Bupa	2.2228	1.1895	2.5014	2.0151	**0.8235**
Jain	3.7815	2.0472	3.6807	3.3423	**1.3796**
Thyroid	3.3313	2.8703	4.6906	4.5049	**1.1821**
WDBC	10.6959	5.7537	11.2447	8.9266	**1.3009**

TABLE 11. Comparison of statistical results obtained using WOA-SA, ABC-DE, PSO-BBO, SOA-TEO3 feature selection algorithms for 20 datasets.

	WOA-SA		ABC-DE		PSO-BBO		SOA-TEO3	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Iris	0.8695	0.0193	0.8269	0.0125	0.7518	0.0091	**0.9030**	**0.0085**
Wine	0.8224	0.0133	0.8342	0.0173	0.7737	0.0135	**0.9094**	**0.0019**
Glass	0.8703	0.0168	0.8896	0.0171	0.7147	0.0180	**0.9346**	**0.0084**
Diabetes	0.8086	0.0122	0.8846	0.0176	0.7769	0.0097	**0.9382**	**0.0061**
Heartstatlog	0.8386	0.0107	0.8032	0.0103	0.7981	0.0150	**0.9399**	**0.0071**
Ionosphere	0.8181	0.0190	0.8910	0.0144	0.7873	0.0180	**0.9053**	**0.0019**
Sonar	0.8812	0.0188	0.8096	0.0182	0.7958	0.0158	**0.9397**	**0.0091**
Vehicle	0.8689	0.0172	0.8897	0.0122	0.7716	0.0096	**0.9171**	**0.0028**
Balancescale	0.8424	0.0103	0.8093	0.0192	0.7415	0.0082	**0.9263**	**0.0085**
CMC	0.8409	0.0170	0.8432	0.0188	0.7906	0.0121	**0.9025**	**0.0019**
Cancer	0.8252	0.0175	0.8286	0.0181	0.7077	0.0138	**0.9424**	**0.0026**
Seeds	0.8984	0.0139	0.8861	0.0117	0.7882	0.0116	**0.9209**	**0.0083**
Blood	0.8343	0.0180	0.8730	0.0150	0.7582	0.0170	**0.9895**	**0.0063**
Aggregation	0.8165	0.0160	0.8565	0.0148	0.7958	0.0097	**0.9503**	**0.0080**
Vowel	0.8195	0.0115	0.8938	0.0148	0.7479	0.0117	**0.9501**	**0.0040**
WBC	0.8493	0.0152	0.8063	0.0122	0.7720	0.0117	**0.9249**	**0.0080**
Bupa	0.8095	0.0180	0.8523	0.0191	0.7411	0.0154	**0.9030**	**0.0038**
Jain	0.8664	0.0135	0.8185	0.0162	0.7942	0.0137	**0.9242**	**0.0024**
Thyroid	0.8697	0.0108	0.8656	0.0198	0.7604	0.0179	**0.9506**	**0.0047**
WDBC	0.8610	0.0106	0.8665	0.0107	0.7645	0.0126	**0.9455**	**0.0061**

and SOA-TEO2. Besides, The CPU time of SOA algorithm is slower than that of TEO algorithm. SOA-TEO3 absorbs the short operation time of TEO algorithm while retaining the optimization ability, and its classification accuracy of the database is also better than other algorithms.

B. COMPARISON WITH THE HYBRID OPTIMIZATION ALGORITHMS

In this experiment, for further showing the merits of SOA-TEO3 method, comparison is performed with other hybrid optimization feature selection algorithms, such as hybrid whale optimization algorithm with simulated annealing for feature selection (WOA-SA) [54], hybrid approach of differential evolution and artificial bee colony for feature selection (ABC-DE) [55] and hybrid PSO assisted biogeography-based optimization for emotion and stress recognition from speech signal (PSO-BBO) [56]. Table 11-13 compares the mean and standard deviation of SOA-TEO3 with the performance of hybrid optimization feature selection algorithms for the 20 datasets. Fig.8 show the average number

TABLE 12. Comparison of statistical results obtained using WOA, SA, ABC, DE feature selection algorithms for 20 datasets.

	WOA		SA		ABC		DE	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Iris	0.7672	0.0106	0.6827	0.0861	0.7151	0.0856	0.6723	0.0882
Wine	0.7381	0.0113	0.6847	0.0923	0.7225	0.0853	0.6622	0.0907
Glass	0.7975	0.0164	0.6544	0.0950	0.7409	0.0867	0.6835	0.0915
Diabetes	0.7601	0.0103	0.7365	0.0892	0.7368	0.0867	0.6948	0.0924
Heartstatlog	0.7775	0.0166	0.6519	0.0912	0.6644	0.0858	0.7229	0.0893
Ionosphere	0.7277	0.0171	0.6832	0.0895	0.6767	0.0923	0.7294	0.0910
Sonar	0.7620	0.0175	0.7077	0.0918	0.7447	0.0925	0.6502	0.0941
Vehicle	0.7340	0.0085	0.6843	0.0857	0.6575	0.0862	0.6639	0.0885
Balancescale	0.7108	0.0160	0.6834	0.0861	0.6839	0.0913	0.7052	0.0903
CMC	0.7151	0.0088	0.6988	0.0873	0.7070	0.0930	0.6618	0.0867
Cancer	0.7176	0.0150	0.7452	0.0947	0.7368	0.0892	0.6797	0.0903
Seeds	0.7750	0.0132	0.7260	0.0942	0.6615	0.0852	0.6864	0.0883
Blood	0.7236	0.0099	0.6935	0.0934	0.6975	0.0949	0.6718	0.0922
Aggregation	0.7555	0.0106	0.6917	0.0899	0.6610	0.0916	0.6831	0.0923
Vowel	0.7550	0.0165	0.7056	0.0870	0.7479	0.0117	0.7339	0.0935
WBC	0.7025	0.0090	0.7155	0.0941	0.7720	0.0117	0.6661	0.0851
Bupa	0.7399	0.0105	0.7228	0.0878	0.7411	0.0154	0.7009	0.0950
Jain	0.7749	0.0153	0.7198	0.0879	0.7942	0.0137	0.7475	0.0897
Thyroid	0.7770	0.0142	0.7406	0.0935	0.7604	0.0179	0.7076	0.0875
WDBC	0.7610	0.0106	0.7254	0.0872	0.7645	0.0126	0.7223	0.0917

TABLE 13. Comparison of statistical results obtained using PSO, BA, GA, BBO feature selection algorithms for 20 datasets.

	PSO		BA		GA		BBO	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Iris	0.7644	0.0102	0.6633	0.0938	0.6826	0.0922	0.7003	0.0895
Wine	0.7252	0.0184	0.7252	0.0908	0.7164	0.0907	0.7280	0.0941
Glass	0.7059	0.0155	0.7440	0.0942	0.6647	0.0947	0.7311	0.0918
Diabetes	0.7655	0.0158	0.6901	0.0936	0.6670	0.0941	0.7700	0.0885
Heartstatlog	0.7850	0.0151	0.6898	0.0853	0.7098	0.0874	0.7249	0.0901
Ionosphere	0.7055	0.0128	0.6902	0.0938	0.7354	0.0904	0.7696	0.0859
Sonar	0.7546	0.0164	0.6752	0.0855	0.6504	0.0901	0.7434	0.0868
Vehicle	0.7921	0.0172	0.6984	0.0925	0.6721	0.0862	0.7850	0.0948
Balancescale	0.7178	0.0132	0.7150	0.0882	0.7041	0.0854	0.7374	0.0857
CMC	0.7212	0.0133	0.7457	0.0896	0.6807	0.0859	0.7960	0.0858
Cancer	0.7673	0.0198	0.7467	0.0942	0.7408	0.0907	0.7753	0.0859
Seeds	0.7978	0.0193	0.6929	0.0864	0.6714	0.0886	0.7860	0.0882
Blood	0.7136	0.0117	0.6689	0.0944	0.6620	0.0931	0.7759	0.0933
Aggregation	0.7265	0.0192	0.7456	0.0919	0.7347	0.0924	0.7486	0.0895
Vowel	0.7084	0.0132	0.7043	0.0879	0.6934	0.0929	0.7511	0.0900
WBC	0.7719	0.0114	0.6871	0.0893	0.6371	0.0935	0.7160	0.0851
Bupa	0.8771	0.0118	0.7373	0.0915	0.8247	0.0894	0.7133	0.0946
Jain	0.8509	0.0175	0.7358	0.0870	0.8388	0.0938	0.7287	0.0861
Thyroid	0.8123	0.0111	0.7488	0.0873	0.8651	0.0856	0.7012	0.0909
WDBC	0.8587	0.0183	0.7224	0.0926	0.8591	0.0941	0.7919	0.0916

of selected by compared algorithms. Table 5 show the CPU time of the algorithms.

It can be seen from table 11-13 that the hybrid optimization algorithm is superior to the classical optimization algorithm, indicating that the hybrid optimization algorithm can effectively improve the feature selection ability of the original algorithm. Table 11 shows the comparison between SOA-TOE3 and three classical hybrid optimization algorithms. It can be seen from the table that SOA-TEO3 has the optimal value and the most accurate classification of features in the datasets. In Fig.8, the closer the curve is to the X-axis, the smaller the feature attributes in the datasets searched by the optimization algorithm. The curve of SOA-TEO3 algorithm is closest to the X-axis, indicating that its characteristic attribute is the smallest in the comparison algorithm and its classification accuracy to the datasets is the highest. In figure 9, the closer the curve is to the X-axis, the shorter the CPU time of the optimization algorithm is. It can be seen

TABLE 14. The calculated p-values from the wilcoxon test for the SOA-TEO3 versus other optimizers.

	SOA-TEO1	SOA-TEO2	WOA-SA	ABC-DE	PSO-BBO	WOA
Iris	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Wine	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Glass	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Diabetes	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Heartstatlog	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Ionosphere	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Sonar	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Vehicle	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Balancescale	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CMC	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Cancer	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Seeds	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Blood	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Aggregation	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Vowel	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
WBC	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Bupa	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Jain	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Thyroid	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
WDBC	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05

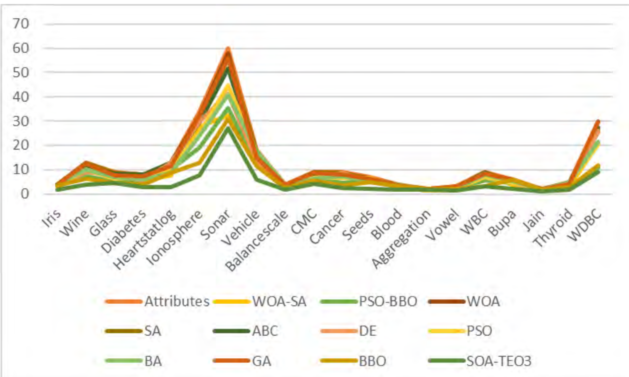

FIGURE 8. The average number of selected by compared algorithms.

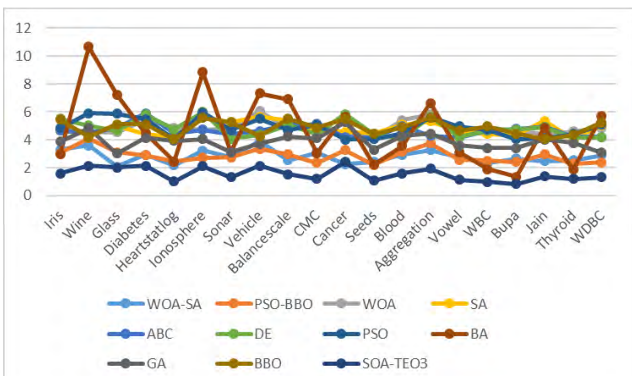

FIGURE 9. The CPU time of compared algorithms.

from the figure that SOA-TEO3 algorithm has the shortest CPU time and is obviously superior to other comparison algorithms. Above all, SOA-TEO3 algorithm can not only ensure the accuracy of feature selection, but also reduce CPU time. It has an excellent ability to solve the problem of feature classification in the datasets.

C. STABILITY AND STATISTICAL ANALYSIS

Based on the natural optimization algorithm, the results of each run are not the same. Therefore, in order to analyze the stability of the proposed algorithm based on SOA-TEO3, we use the value of standard deviation (STD). The STD can be intuitive to the operation stability of the algorithm, and the lower the value of the algorithm, the stronger the robustness of the algorithm. Table 8, 11, 12, 13 shows the STD values of each algorithm after 30 runs. It can be seen from the table that the stability of SOA-TEO3 algorithm is the strongest, especially when dealing with the multi-attribute datasets, its stability is obviously better than other comparison algorithms, indicating that proposed algorithm has a good ability, and can find the optimal attribute of datasets better, more accurately and more stable.

We statistically analyze the experimental results to better observe the differences between algorithms. We use Wilcoxon rank sum test [51], a nonparametric statistical test that checks whether one of two independent samples is larger than the other. We calculate the p-value of fitness of SOA-TEO3 algorithm and other compared algorithm in this paper. The experimental statistical results are shown in table 14 and table 15. If the p-value of two algorithms is greater than 0.05, there is no significant difference between the two algorithms. On the other hand, a p-value less than 0.05 means that there is a significant difference between the two algorithms at the significance level of 5%. It can be seen from the table 14 and table 15 that the SOA-TEO3 algorithm

TABLE 15. The calculated p-values from the wilcoxon test for the SOA-TEO3 versus other optimizers.

	SA	ABC	DE	PSO	BA	GA	BBO
Iris	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Wine	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Glass	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Diabetes	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Heartstatlog	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Ionosphere	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Sonar	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Vehicle	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Balancescale	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
CMC	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Cancer	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Seeds	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Blood	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Aggregation	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Vowel	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
WBC	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Bupa	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Jain	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
Thyroid	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05
WDBC	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05	P<0.05

is obviously better than the comparison algorithm in the statistical sense.

D. COMPARISON WITH THE NOVEL FEATURE SELECTION METHOD

In this section, some commonly used feature selection approaches such as a new feature selection (NFS) [74], the unsupervised feature selection (UFS) [75] and mutual information maximization (DRJMIM) [76] are utilized to validate the effectiveness of the proposed method. The fig.10 show the average number of selected by novel feature selection methods. The fig.11 show the CPU time of the compared feature selection methods.

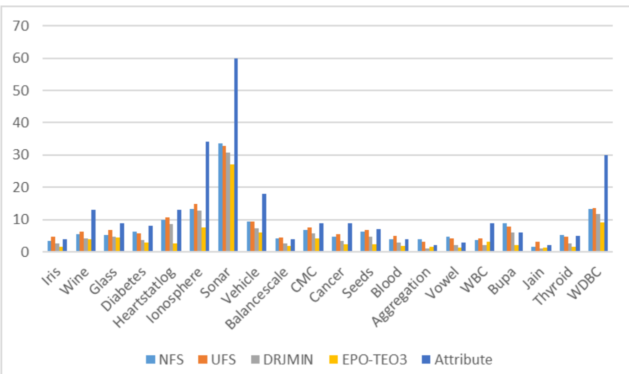

FIGURE 10. The average number of selected by novel feature selection methods.

As can be seen from figure 10, the number of features can be effectively reduced by all the comparison algorithms. The SOA-TEO3 algorithm has the best effect, and

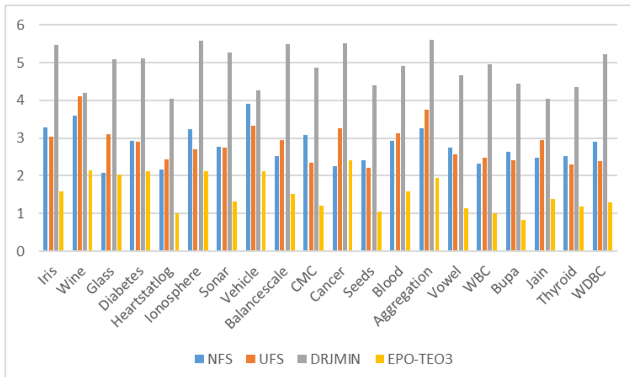

FIGURE 11. The CPU time of compared feature selection methods.

the number of features is the least in the comparison algorithm. As can be seen from figure 11, the SOA-TEO3 algorithm has the shortest CPU time, which can guarantee the accuracy and reduce the operation time at the same time. In order to better test each comparison algorithm, the ability of a classifier to discriminate between '0' and '1' is evaluated using receiver operating characteristic (ROC) curve analysis [77]. The ROC curve shows the trade-off between '0' and '1', and demonstrates that the closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate the classifier. The area under the curve (AUC) is the evaluation criteria for the classifier [78]. Fig.12 shows the ROC curve for the compared feature selection methods. Fig.13 shows the radar chart of the AUC value.

As can be seen from figure 12 and figure 13, the ROC curve of SOA-TEO3 is better than other comparison algorithms,

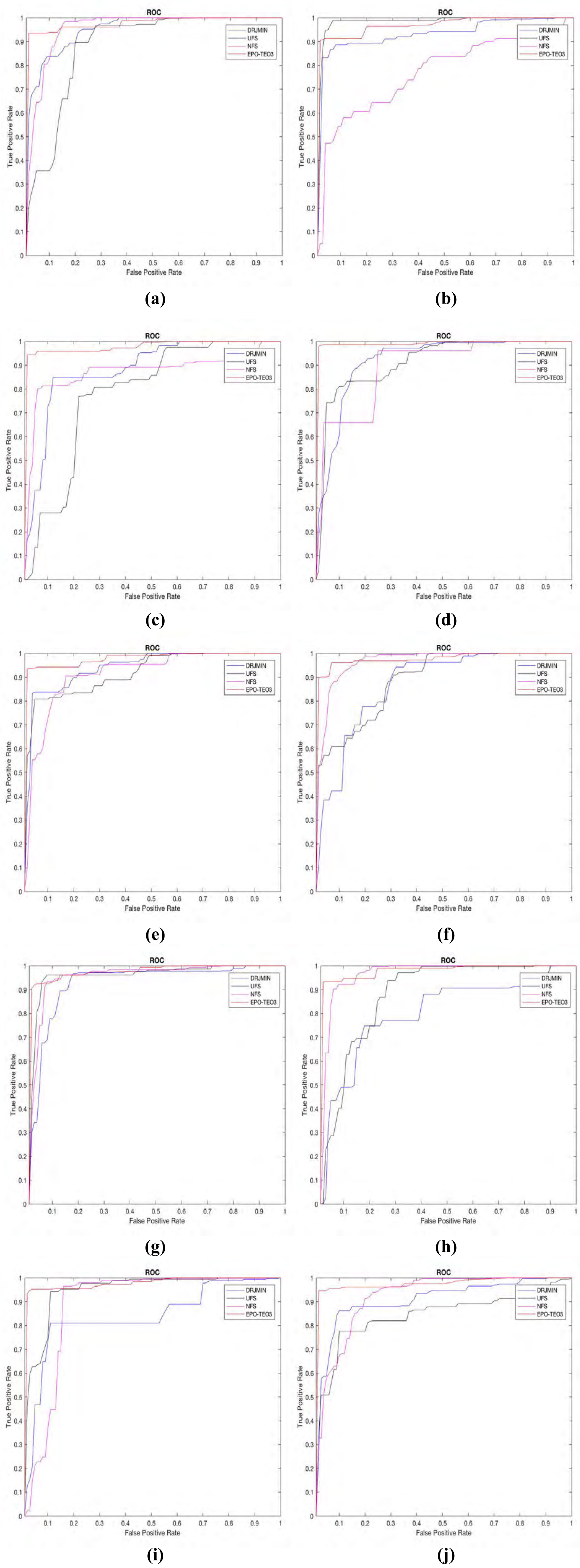

FIGURE 12. The ROC curve of the compared feature selection. (a) Iris, (b) Wine, (c) Glass, (d) Diabetes, (e) Heartstatlog, (f) Ionosphere, (g) Sonar, (h) Vehicle, (i) Balancescale, (j) CMC, (k) Cancer. (l) Seeds, (m) Blood, (n) Aggregation, (o) Vowel, (p) WBC, (q) Bupa, (r) Jain, (s) Thyroid, (t) WDBC.

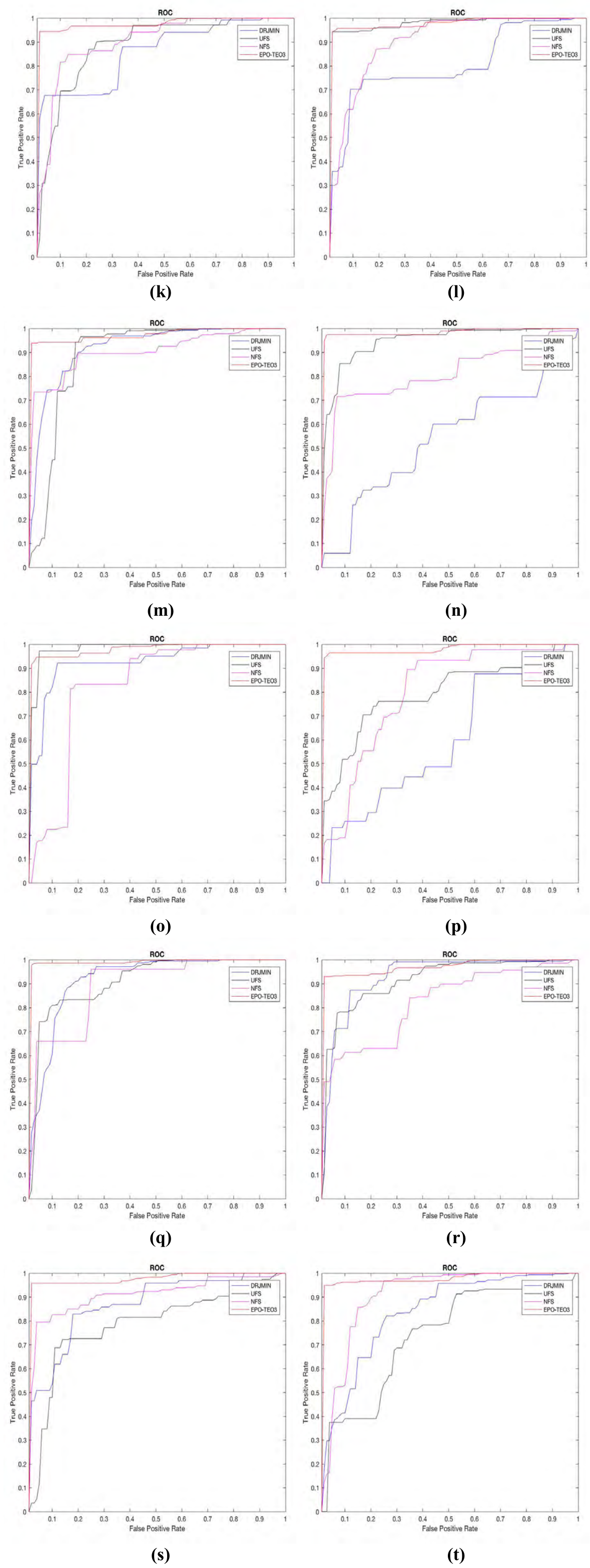

FIGURE 12. *(Continued.)* The ROC curve of the compared feature selection. (a) Iris, (b) Wine, (c) Glass, (d) Diabetes, (e) Heartstatlog, (f) Ionosphere, (g) Sonar, (h) Vehicle, (i) Balancescale, (j) CMC, (k) Cancer. (l) Seeds, (m) Blood, (n) Aggregation, (o) Vowel, (p) WBC, (q) Bupa, (r) Jain, (s) Thyroid, (t) WDBC.

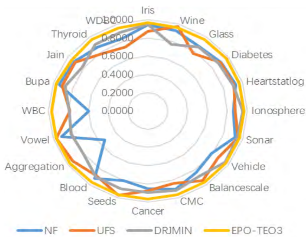

FIGURE 13. The radar chart of the AUC value.

and the AUC value is also better than other comparison algorithms, indicating that SOA-TEO3 has a good effect on data classification.

VII. CONCLUSIONS

In this paper, a novel hybridization approach is proposed based on SOA and TEO. This paper proposes three hybrid methods, the first is to randomly select an algorithm for location update in the iteration process, the second is to add TEO algorithm's location update formula after SOA algorithm iteration, and the last is to use TEO algorithm's heat exchange formula to improve the predation mode of SOA algorithm. Through experimental verification and analysis, it is found that the SOA-TEO3 fusion method is superior to the other two fusion methods, and effectively improves the local optimization ability of the SOA algorithm.

The performances of the proposed approaches are assessed and compared against three hybrid optimization feature selection methods including WOA-SA, ABC-DE and PSO-BBO. At the same time, the approach method is compared with the feature selection methods of some classical optimization algorithms. The criteria of two evaluation methods: classification accuracy and average selection size are reported. We find the SOA-TEO3 algorithm can enhance the exploitation of the SOA and reduce the CPU time. In the comparison of hybrid optimization algorithm, the proposed method also has strong competitiveness in the classification of feature selection. The experimental results show that SOA-TEO3 algorithm can balance exploitation and exploration effectively and has good robustness.

The algorithm proposed in this paper does not carry out feature selection for the actual data set. In the future, we will study this algorithm and apply it to more fields to better solve the problem of feature selection. The improved algorithm has better ability in feature selection, but the results are relatively complex. We will study simpler optimization algorithm and continue to study to improve the accuracy of feature selection.

REFERENCES

[1] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. M. Al-Zoubi, and S. Mirjalili, "Binary grasshopper optimisation algorithm approaches for feature selection problems," *Expert Syst. Appl.*, vol. 117, pp. 267–286, Mar. 2019.

[2] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, nos. 1–4, pp. 131–156, 1997.

[3] Z. Peng, X. Hu, P. Li, and X. Wu, "Online feature selection for high-dimensional class-imbalanced data," *Knowl.-Based Syst.*, vol. 136, pp. 187–199, Nov. 2017.

[4] A. Zarshenas and K. Suzuki, "Binary coordinate ascent: An efficient optimization technique for feature subset selection for machine learning," *Knowl.-Based Syst.*, vol. 110, pp. 191–201, Oct. 2016.

[5] X. Zhang *et al.*, "Feature selection in mixed data: A method using a novel fuzzy rough set-based information entropy," *Pattern Recognit.*, vol. 56, pp. 1–15, 2016.

[6] K. Zheng and X. Wang, "Feature selection method with joint maximal information entropy between features and class," *Pattern Recognit.*, vol. 77, pp. 20–29, May 2018.

[7] G. I. Sayed, G. Khoriba, and M. H. Haggag, "A novel chaotic salp swarm algorithm for global optimization and feature selection," *Appl. Intell.*, vol. 48, no. 10, pp. 3462–3481, 2018.

[8] M. Wang, C. Wu, L. Wang, D. Xiang, and X. Huang, "A feature selection approach for hyperspectral image based on modified ant lion optimizer," *Knowl.-Based Syst.*, vol. 168, pp. 39–48, Mar. 2019. doi: 10.1016/j.knosys.2018.12.031.

[9] L. Sharmila and U. Sakthi, "An artificial immune system-based algorithm for abnormal pattern in medical domain," *J. Supercomput.*, vol. 4, pp. 1–15, Apr. 2018.

[10] M. B. Cole, O. Galárraga, I. B. Wilson, B. Wright, and A. N. Trivedi, "At federally funded health centers, medicaid expansion was associated with improved quality of care," *Health Affairs*, vol. 36, no. 1, pp. 40–48, 2017.

[11] B. Manavalan and J. Lee, "SVMQA: Support–vector-machine-based protein single-model quality assessment," *Bioinformatics*, vol. 33, no. 16, pp. 2496–2503, 2017.

[12] Y. Liu, J.-W. Bi, and Z.-P. Fan, "A method for multi-class sentiment classification based on an improved one-vs-one (OVO) strategy and the support vector machine (SVM) algorithm," *Inf. Sci.*, vols. 394–395, pp. 38–52, Jul. 2017.

[13] Y. Pan, Z. Xiao, X. Wang, and D. Yang, "A multiple support vector machine approach to stock index forecasting with mixed frequency sampling?" *Knowl.-Based Syst.*, vol. 122, pp. 90–102, Apr. 2017.

[14] G. Bhattacharya, K. Ghosh, and A. S. Chowdhury, "Granger causality driven AHP for feature weighted kNN," *Pattern Recognit.*, vol. 66, pp. 425–436, Jun. 2017.

[15] F. Shaukat, G. Raja, A. Gooya, and A. F. Frangi, "Fully automatic detection of lung nodules in CT images using a hybrid feature set," *Med. Phys.*, vol. 44, no. 7, pp. 3615–3629, 2017.

[16] E. Viegas, A. O. Santin, A. França, R. Jasinski, V. A. Pedroni, and L. S. Oliveira, "Towards an energy-efficient anomaly-based intrusion detection engine for embedded systems," *IEEE Trans. Comput.*, vol. 66, no. 1, pp. 163–177, Jan. 2017.

[17] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta, "Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science," *Nature Commun.*, vol. 9, Jun. 2018, Art. no. 2383.

[18] F. Ascione, N. Bianco, C. De Stasio, G. M. Mauro, and G. P. Vanoli, "Artificial neural networks to predict energy performance and retrofit scenarios for any member of a building category: A novel approach," *Energy*, vol. 118, pp. 999–1017, Jan. 2017.

[19] O. E. Flouty *et al.*, "Infantile cranial fasciitis: Case-based review and operative technique," *Child's Nervous Syst.*, vol. 33, no. 6, pp. 899–908, 2017.

[20] M. Ali, "iCBLS: An interactive case-based learning system for medical education," *Int. J. Med. Inform.*, vol. 109, pp. 55–69, Jan. 2018.

[21] Y. Chen and Y. Hao, "A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction," *Expert Syst. Appl.*, vol. 80, pp. 340–355, Sep. 2017.

[22] M. Kumar, N. K. Rath, A. Swain, and S. K. Rath, "Feature selection and classification of microarray data using mapreduce based ANOVA and K-nearest neighbor," *Procedia Comput. Sci.*, vol. 54, pp. 301–310, Jun. 2015.

[23] M. Ghaemi and M.-R. Feizi-Derakhshi, "Feature selection using Forest Optimization Algorithm," *Pattern Recognit.*, vol. 60, pp. 121–129, Dec. 2016.

[24] A. Wang, N. An, G. Chen, L. Li, and G. Alterovitz, "Accelerating wrapper-based feature selection with K-nearest-neighbor," *Knowl.-Based Syst.*, vol. 83, pp. 81–91, Jul. 2015.

[25] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *J. Comput. Sci.*, vol. 25, pp. 152–160, Mar. 2017.

[26] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, nos. 2–3, pp. 95–99, 1988.

[27] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[28] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[29] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci. (MHS)*, Oct. 1995, pp. 39–43.
[30] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008.
[31] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperat. Strategies Optim.*, vol. 284, pp. 65–74, 2010.
[32] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, pp. 4831–4845, May 2012.
[33] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Proc. Int. Conf. Unconventional Comput. Natural Comput.* Berlin, Germany: Springer, Sep. 2012, pp. 240–249.
[34] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in *Foundations of Fuzzy Logic and Soft Computing* (Lecture Notes in Computer Science), vol. 4529. Cancún, Mexico: IFSA, 2007.
[35] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
[36] A. Kaveh and A. Dadras, "A novel meta-heuristic optimization algorithm: Thermal exchange optimization," *Adv. Eng. Softw.*, vol. 110, pp. 69–84, Aug. 2017.
[37] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," *Knowl.-Based Syst.*, vol. 165, pp. 169–196, Feb. 2018.
[38] A. H. Niknamfar, S. T. A. Niaki, and S. A. A. Niaki, "Opposition-based learning for competitive hub location: A bi-objective biogeography-based optimization algorithm," *Knowl.-Based Syst.*, vol. 128, pp. 1–19, Jul. 2017.
[39] S. Amirsadri, S. J. Mousavirad, and H. Ebrahimpour-Komleh, "A levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training," *Neural Comput. Appl.*, vol. 30, no. 12, pp. 3707–3720, 2017.
[40] L. Xu *et al.*, "Enhanced moth-flame optimization based on cultural learning and Gaussian mutation," *J. Bionic Eng.*, vol. 15, no. 4, pp. 751–763, 2018.
[41] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *Proc. Int. Conf. Comput. Intell. Modeling, Control Automat., Int. Conf. Intell. Agents, Web Technol. Internet Commerce*, Nov. 2005, pp. 695–701.
[42] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search: A new population-based metaheuristic optimization algorithm," *Appl. Intell.*, vol. 47, no. 3, pp. 850–887, 2017.
[43] T.-M. Li, J. Lehtinen, R. Ramamoorthi, W. Jakob, and F. Durand, "Anisotropic Gaussian mutations for metropolis light transport through Hessian-Hamiltonian dynamics," *ACM Trans. Graph.*, vol. 34, no. 6, p. 209, 2015.
[44] Y. Marinakis, A. Migdalas, and A. Sifaleras, "A hybrid particle swarm optimization—Variable neighborhood search algorithm for constrained shortest path problems," *Eur. J. Oper. Res.*, vol. 261, no. 3, pp. 819–834, 2017.
[45] D. Guangqian, K. Bekhrad, P. Azarikhah, and A. Maleki, "A hybrid algorithm based optimization on modeling of grid independent biodiesel-based hybrid solar/wind systems," *Renew. Energy*, vol. 122, pp. 551–560, Jul. 2018.
[46] W. Alsaeedan, M. El Bachir Menai, and S. Al-Ahmadi, "A hybrid genetic-ant colony optimization algorithm for the word sense disambiguation problem," *Inf. Sci.*, vol. 417, pp. 20–38, Nov. 2017.
[47] M. A. El Aziz, A. A. Ewees, and A. E. Hassanien, "Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation," *Expert Syst. Appl.*, vol. 83, pp. 242–256, Oct. 2017.
[48] K. Singh, K. Singh, L. H. Son, and A. Aziz, "Congestion control in wireless sensor networks by hybrid multi-objective optimization algorithm," *Comput. Netw.*, vol. 138, pp. 90–107, Jun. 2018.
[49] E. Daniel, J. Anitha, and J. Gnanaraj, "Optimum Laplacian wavelet mask based medical image using hybrid cuckoo search—Grey wolf optimization algorithm," *Knowl.-Based Syst.*, vol. 131, pp. 58–69, Sep. 2017.
[50] O. Engin and A. Güçlü, "A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems," *Appl. Soft Comput.*, vol. 72, pp. 166–176, Nov. 2018.
[51] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Appl. Math. Comput.*, vol. 274, pp. 292–305, Feb. 2016.
[52] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Appl. Soft. Comput.*, vol. 43, pp. 117–130, Jun. 2016.
[53] R. A. Ibrahim, A. A. Ewees, D. Oliva, M. A. Elaziz, and S. Lu, "Improved salp swarm algorithm based on particle swarm optimization for feature selection," *J. Ambient Intell. Humanized Comput.*, pp. 1–15, Sep. 2018. [Online]. Available: https://link.springer.com/article/10.1007/s12652-018-1031-9. doi: 10.1007/s12652-018-1031-9.
[54] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neuro Comput.*, vol. 260, pp. 302–312, Oct. 2017.
[55] E. Zorarpacı and S. A. Özel, "A hybrid approach of differential evolution and artificial bee colony for feature selection," *Expert Syst. Appl.*, vol. 62, pp. 91–103, Nov. 2016.
[56] C. K. Yogesh *et al.*, "A new hybrid PSO assisted biogeography-based optimization for emotion and stress recognition from speech signal," *Expert Syst. Appl.*, vol. 69, pp. 149–158, Mar. 2017.
[57] Y. Wan, M. Wang, Z. Ye, and X. Lai, "A feature selection method based on modified binary coded ant colony optimization algorithm," *Appl. Soft Comput.*, vol. 49, pp. 248–258, Dec. 2016.
[58] M. A. Tahir, A. Bouridane, and F. Kurugollu, "Simultaneous feature selection and feature weighting using Hybrid tabu Search/K-nearest neighbor classifier," *Pattern Recognit. Lett.*, vol. 28, no. 4, pp. 438–446, 2007.
[59] H. K. Chantar and D. W. Corne, "Feature subset selection for Arabic document categorization using BPSO-KNN," in *Proc. 3rd World Congr. Nature Biol. Inspired Comput.*, Oct. 2011, pp. 546–551.
[60] F. Rohart, B. Gautier, A. Singh, and K.-A. Lê Cao, "mixOmics: An R package for 'omics feature selection and multiple data integration," *PLoS Comput. Biol.*, vol. 13, no. 11, 2017, Art. no. e1005752.
[61] A. Rashno, B. Nazari, S. Sadri, and M. Saraee, "Effective pixel classification of Mars images based on ant colony optimization feature selection and extreme learning machine," *Neurocomputing*, vol. 226, pp. 66–79, Feb. 2017.
[62] G. Jiang and W. Wang, "Error estimation based on variance analysis of k-fold cross-validation," *Pattern Recognit.*, vol. 69, pp. 94–106, Sep. 2017.
[63] N. Awad, M. Z. Ali, and R. G. Reynolds, "A differential evolution algorithm with success-based parameter adaptation for CEC2015 learning-based optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 1098–1105.
[64] C. L. Blake and C. J. Merz. (1998). *UCI Repository of Machine Learning Databases*. Accessed: Jun. 1, 2016. [Online]. Available: http://www.ics.uci.edu/?mlearn/
[65] G. I. Sayed, A. Darwish, and A. E. Hassanien, "A new chaotic whale optimization algorithm for features selection," *J. Classification*, vol. 35, no. 2, pp. 300–344, 2018.
[66] M. Battistutta, A. Schaerf, and T. Urli, "Feature-based tuning of single-stage simulated annealing for examination timetabling," *Ann. Oper. Res.*, vol. 252, no. 2, pp. 239–254, 2017.
[67] E. Hancer, B. Xue, D. Karaboga, and M. Zhang, "A binary ABC algorithm based on advanced similarity scheme for feature selection," *Appl. Soft Comput.*, vol. 36, pp. 334–348, Nov. 2015.
[68] V. K. Ojha, V. Snášel, and A. Abraham, "Multiobjective programming for type-2 hierarchical fuzzy inference trees," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 2, pp. 915–936, Apr. 2018.
[69] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary PSO with mutation operator for feature selection using decision tree applied to spam detection," *Knowl.-Based Syst.*, vol. 64, pp. 22–31, Jul. 2014.
[70] S. Kashi, A. Minuchehr, N. Poursalehi, and A. Zolfaghari, "Bat algorithm for the fuel arrangement optimization of reactor core," *Ann. Nucl. Energy*, vol. 64, pp. 144–151, Feb. 2014.
[71] S. A. Kazarlis, S. E. Papadakis, J. B. Theocharis, and V. Petridis, "Microgenetic algorithms as generalized hill-climbing operators for GA optimization," *IEEE Trans. Evol. Comput.*, vol. 5, no. 3, pp. 204–217, Jun. 2001.
[72] S. Yazdani, J. Shanbehzadeh, and E. Aminian, "Feature subset selection using constrained binary/integer biogeography-based optimization," *ISA Trans.*, vol. 52, no. 3, pp. 383–390, May 2013.
[73] F. C. Lam and M. T. Longnecker, "A modified Wilcoxon rank sum test for paired data," *Biometrika*, vol. 70, no. 2, pp. 510–513, 1983.
[74] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018.

[75] W. Gao, L. Hu, P. Zhang, and J. He, "Feature selection considering the composition of feature relevancy," *Pattern Recognit. Lett.*, vol. 112, pp. 70–74, Sep. 2018.
[76] T. Chang *et al.*, "Robust unsupervised feature selection via dual self-representation and manifold regularization," *Knowl.-Based Syst.*, vol. 145, pp. 109–120, Apr. 2018.
[77] W. Mumtaz, M. N. b Mohamad Saad, N. Kamel, S. S. A. Ali, and A. S. Malik, "An EEG-based functional connectivity measure for automatic detection of alcohol use disorder," *Artif. Intell. Med.*, vol. 84, pp. 79–89, Jan. 2018.
[78] K. Ren *et al.*, "A robust AUC maximization framework with simultaneous outlier detection and feature selection for positive-unlabeled classification," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

HEMING JIA received the Ph.D. degree in system engineering from Harbin Engineering University, China, in 2012. He is currently an Associate Professor with Northeast Forestry University. His research interests include nonlinear control theory and application, image segmentation, and swarm optimization algorithm.

ZHIKAI XING was born in Daqing, China, in 1993. He is currently pursuing the M.S. degree in control engineering from Northeast Forestry University, China. His research interests include image segmentation and intelligent optimization algorithm.

WENLONG SONG was received the Ph.D. degree in mechanical design and theory from Northeast Forestry University, China, in 2008, where he is currently a Professor. His research interests include image segmentation and swarm intelligence algorithm.