

Received May 6, 2019, accepted May 26, 2019, date of publication May 30, 2019, date of current version June 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2919991

Spotted Hyena Optimization Algorithm With Simulated Annealing for Feature Selection

HEMING JIA ¹, (Member, IEEE), JINDUO LI, WENLONG SONG, XIAOXU PENG, CHUNBO LANG, AND YAO LI

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 The purpose of this paper is to propose a new hybrid metaheuristic to solve the problem of feature selection. Feature selection problem is the process of finding the most relevant subset based on some criteria. A hybrid metaheuristic is a new trend in the development of optimization algorithms. In this paper, two different hybrid models are designed based on spotted hyena optimization (SHO) for feature selection problem. The SHO algorithm can find the optimal or nearly optimal feature subset in the feature space to minimize the given fitness function. In the first model, the simulated annealing algorithm (SA) is embedded in the SHO algorithm (called SHOSA-1) to enhance the optimal solution found by the SHO algorithm after each iteration. In the second model, SA enhances the final solution obtained by the SHO algorithm (called SHOSA-2). The performance of these methods is evaluated in 20 datasets in the UCI repository. The experiments show that SHOSA-1 performs better than the native algorithm and SHOSA-2. And then, SHOSA-1 is compared with six state-of-the-art optimization algorithms. The experimental results confirm that SHOSA-1 improves the classification accuracy and reduces the number of selected features compared with other wrapper-based optimization algorithms. That proves the excellent performance of SHOSA-1 in spatial search and feature attribute selection.

INDEX TERMS Hybrid optimization, spotted hyena optimization algorithm, simulated annealing, classification, SHO optimization.

I. INTRODUCTION

As a part of data mining, data preprocessing can significantly improve the performance of data mining [1]. Feature selection is a key step in data preprocessing, which can reduce processing time, process too many features and improve the accuracy of machine learning [2]. The precision of the classifier and the reduction of the number of features are the criteria for evaluating feature selection techniques [3]. Generally speaking, feature selection is divided into four steps: subset generation, subset evaluation, stop criteria, and result verification [4]. Feature selection can be broadly divided into two categories: filters and wrappers. Filters can get results faster, while wrappers can significantly improve accuracy. So wrappers are more suitable for mining algorithms [5]. Wrappers have three main factors: search techniques for

finding the best combination of features, evaluation criteria for feature subsets, and classifiers [6].

For the past decade, meta-heuristic algorithms have been one of the first choices for solving data mining and machine learning problems. The exact search generates all the solutions. If a dataset have N characteristics, an accurate search yields $2 \times N$ solutions [7]. Random search is much less computationally intensive than exact search. Random search randomly select the next set of random features [8]. Meta-heuristic algorithm is superior to precise search and random search. Although the meta-heuristic algorithm does not necessarily find the optimal solution when it runs every time, it will generate an acceptable solution within a specified time [9]. Recently, many meta-heuristic algorithms have been applied to feature selection, such as: particle swarm optimization (PSO) [10], genetic algorithm (GA) [11], flower pollination algorithm (FPA) [12], ant lion optimization (ALO) [13], butterfly optimization algorithm (BOA) [14], bat algorithm (BA) [15], memetic algorithm (MA) [16], ant

The associate editor coordinating the review of this manuscript and approving it for publication was Le Hoang Son.

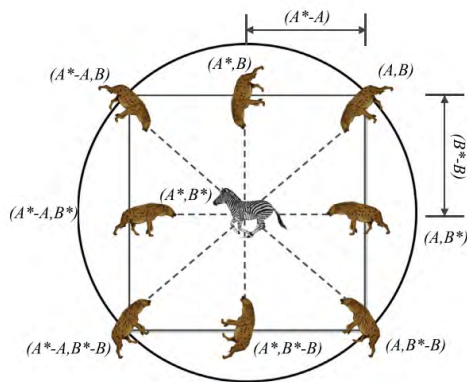


FIGURE 1. 2D position vectors of spotted hyena.

colony optimization (ACO) [17], bee colony optimization (BCO) [18], salp swarm algorithm (SSA) [19], whale optimization algorithm (WOA) [20], grey wolf optimization (GWO) [21], dragonfly algorithm (DA) [22], biogeography based optimization (BBO) [23].

When using meta-heuristic algorithm, two points must be considered: high spatial search rate and better solution [24]. Therefore, meta-heuristic algorithms are divided into two categories: exploration-oriented population algorithm and development-oriented search algorithm. There is a hybrid model called hybrid meta-heuristic that improves the performance of each algorithm [25]. The first kind of model, local search as a global search operator is used for search; In the second model, first algorithm does the search and the other algorithm strengthens the solution that the first algorithm gets.

The spotted hyena optimization (SHO) proposed by G. Dhiman and V. Kumar which is a new optimization algorithm [26], which simulates the hunting behavior of spotted hyenas. Spotted hyenas hunt prey through networks of trusted friends and the ability to recognize prey. This hunting method can find a better solution in a shorter time. SHO is similar to GWO, and GWO has received a lot of attention for its unique way of updating its three leaders [21]. SHO improved the three leaders into N leaders based on GWO, which greatly enhanced the self-adaptation of algorithm. In addition, SHO can be extended to higher dimensions. SHO tests on the dataset and compares it with several recently developed meta-heuristics by G. Dhiman. Then the algorithm is applied to the constrained engineering design problem to verify the applicability of the algorithm in practical problems. Experimental results show that the algorithm has better performance [27], [28]. SHO is proven to be superior to other optimization algorithms in optical buffer design and airfoil design problems [29]. SHO is better than other optimization algorithms in solving economic load power scheduling problem [30].

Simulated annealing algorithm (SA) is a meta-heuristic algorithm proposed by Kirkpatrick *et al.* [31]. The simulated annealing algorithm starts from a high temperature and search the global optimal solution of the target function in the space

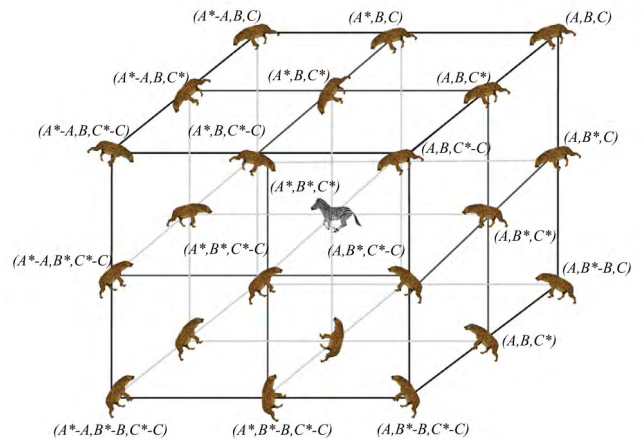


FIGURE 2. 3D position vectors and possible next locations of spotted hyena.

with the continuous decrease of temperature and the combination of probability function. SA has strong local search capability [32]. And SA has the ability to enhance the local optimal solution in the hybrid algorithm [33].

This paper presents a wrapper selection method for hybrid SHO and SA algorithms. This method is proposed to improve the utilization rate of SHO algorithm. In the first hybrid model, SA algorithm is employed as part of SHO algorithm, and SHO algorithm and SA algorithm are run once for each iteration. The second hybrid method first uses SHO algorithm to search for the optimal solution, and then uses SA algorithm to search for the current optimal solution.

The rest parts of this paper are described as follows: the second section introduces the related work of hybrid algorithm, the third section introduces the basic principles of SHO algorithm and SA algorithm, and the fourth section introduces two methods of mixed model. In section 5, experimental results are given and discussed in detail. Finally, the conclusion is given in section 6.

II. RELATED WORK

In recent years, hybrid meta-heuristic algorithms have been gradually used in the field of optimization. O. C. Martin and S. W. Otto proposed a hybrid algorithm of Markov chain and simulated annealing in 1996 [34]. The algorithm is able to solve large instances to optimality, improving upon local search methods very significantly. The hybrid meta-heuristic algorithm was proposed by I. S. Oh *et al.* in 2004, in which local search algorithm was embedded into genetic search to solve the problem. The hybrid GA showed better convergence properties compared to the classical GA [35]. A hybrid simulated annealing and genetic algorithm is proposed by P. Vasant to find the global optimal solution of nonlinear objective function and to search for the best feasible solutions to the decision variables [36]. A hybrid simulated annealing and genetic algorithm applied to signal optimization model is proposed by Z. Li and P. Schonfeld. The hybrid algorithm can quickly generate optimal or near-optimal signal timings [37].

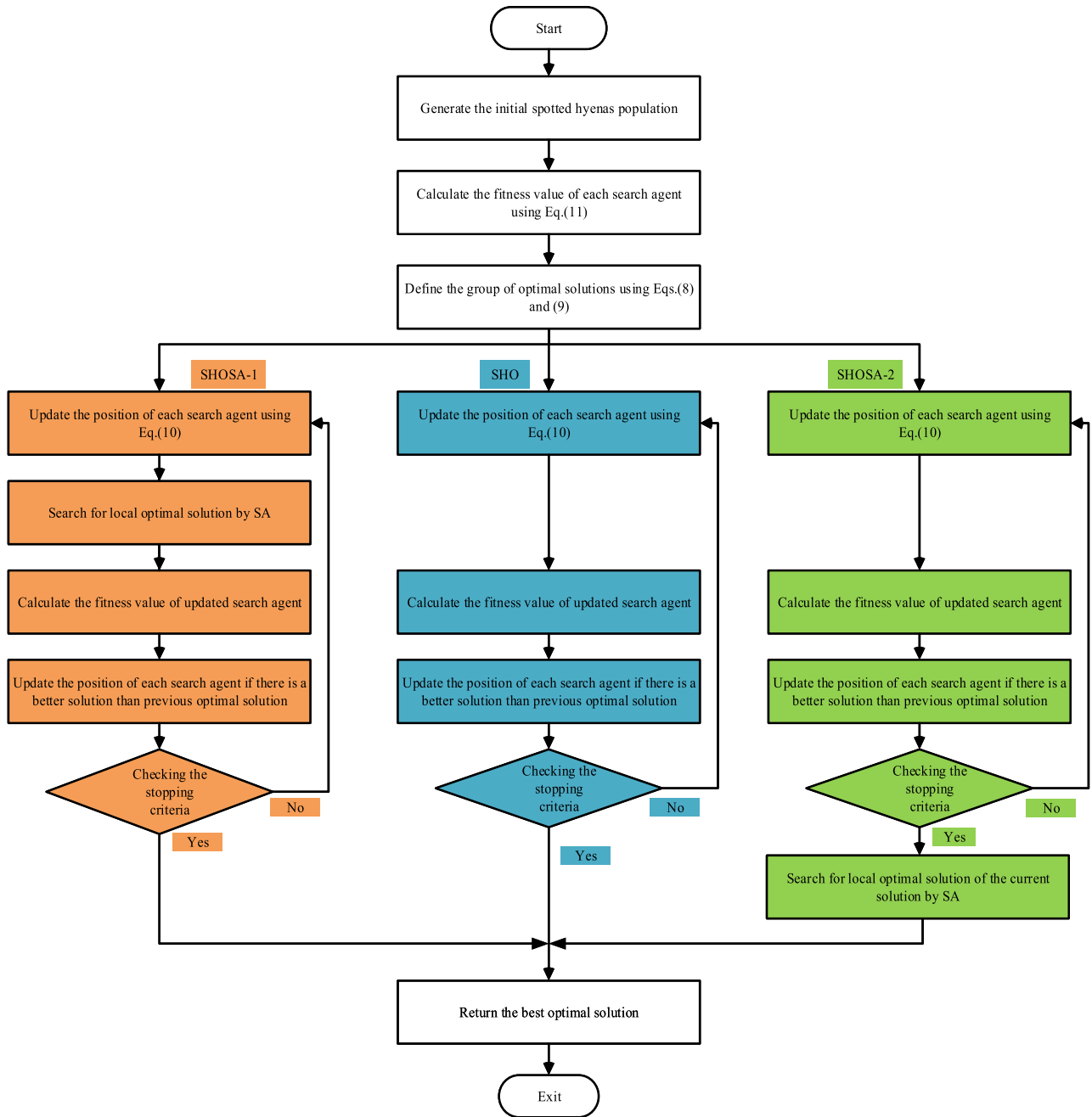


FIGURE 3. Flowchart of the SHO, SHOSA-1 and SHOSA-2.

In addition, two hybrid simulated annealing and genetic algorithm models were also applied in Location-Inventory-Routing problem and building optimization problem [32], [38]. A GA-GSA hybrid algorithm was proposed to improve system performance by using uncertain data [39]. A hybrid GSA-GA algorithm and a hybrid PSO-GA algorithm were proposed H. Garg to solve the constrained optimization problem [40], [41]. The results showed that the hybrid model algorithm has better performance. Some hybrid meta-heuristic algorithms have been proposed for feature selection.

J. Wu et al. introduced simulated annealing algorithm into the crossover operator of genetic algorithm, is used to simultaneously optimize the input feature subset selection, the type of kernel function and the kernel parameter setting of SVR [42]. A hybrid ant colony algorithm and genetic algorithm was applied in the categorization of text documents by Basiri and Nemati [43]. J. B. Jona and N. Nagaveni proposed a hybrid ant colony algorithm and cuckoo algorithm that applied in detect the breast cancer [44]. A hybrid whale optimization algorithm with simulated annealing algorithm was proposed by M. M. Mafarja and S. Mirjalili is applied to

TABLE 1. List of datasets used in the experiments.

S.no.	Dataset	No. of attributes	No. of objects
1	BreastCW	10	699
2	Congressional	16	435
3	ConnectBench	60	208
4	Dermatology	35	366
5	Drug_consumption	32	1888
6	Glass	10	214
7	Heart	13	270
8	Hepatitis	19	155
9	Horse-colic	27	368
10	ILPD	10	583
11	Ionosphere	34	351
12	Primary-tumor	17	339
13	Seeds	7	210
14	Soybean	35	307
15	Spambase	58	4601
16	SPECT Heart	22	267
17	SteelPF	27	1941
18	ThoracicSurgery	17	470
19	Tic-tac-toe	9	958
20	Zoo	17	101

feature selection and significantly improves the classification accuracy [33]. These algorithms all have varying degrees of improvement over native algorithms.

Despite the advantages of the above algorithms for feature selection, there is a theorem in optimization that there is No-Free-Lunch (NFL): there is no algorithm that solves all optimization problems. In other words, in the field of feature selection, there are always better algorithms that can better solve the problem of new feature selection. This prompted us to try to propose a new method for feature selection in this study.

III. THE PROPOSED ALGORITHM

A. SPOTTED HYENA OPTIMIZER (SHO)

Spotted hyenas are large carnivorous canines that live in a variety of open or dry environments. Large and medium-sized herbivores such as zebras, wildebeests, and impala are preyed upon by swarms of spotted hyenas [45]. Spotted hyenas are very intelligent social animals. Relatives and other individuals are identified by them using multiple senses. They also ranked the relationships of the same race. Individuals of high status in a population are given priority for trust [46]. Spotted hyenas have a very high success rate in group hunting because of this nature.

1) ENCIRCLING PREY

Spotted hyenas can know where their prey is and surround them. We consider the current best candidate is the spotted hyena closest to the target or prey because of search space not known a priori. The locations of other search agents are updated after the best search solution is defined.

TABLE 2. Groups of the population size and the maximum number of iterations.

No.	The population size	The number of iterations
1	10	50
2	10	100
3	10	150
4	30	50
5	30	100
6	30	150
7	50	50
8	50	100
9	50	150

The mathematical model of this behavior is described by the following equations:

$$\vec{D}_h = |\vec{B} \cdot \vec{P}_p(x) - \vec{P}(x)| \quad (1)$$

$$\vec{p}(x+1) = \vec{p}_p(x) - \vec{E} \cdot \vec{D}_h \quad (2)$$

where \vec{D}_h define the distance between spotted hyenas and their prey, X indicates the current iteration, \vec{B} and \vec{E} indicates co-efficient vectors, \vec{P}_p indicates the position vector of prey, \vec{P} indicates the position vector of spotted hyena. $||$ and \cdot is the absolute value and multiplication with vectors respectively.

The vectors B and E are computed as follows:

$$\vec{B} = 2 \cdot r \vec{d}_1 \quad (3)$$

$$\vec{E} = 2 \vec{h} \cdot r \vec{d}_2 - \vec{h} \quad (4)$$

$$\vec{h} = 5 - (Iteration * \frac{5}{MAX_{Iteration}}) \quad (5)$$

where, $Iteration = 1, 2, 3, \dots, Max_{Iteration}$.

In the process of the maximum number of iterations, \vec{h} decreases linearly from 5 to 0 for proper balancing the exploration and exploitation. This mechanism facilitates more development as the number of iterations ($Max_{Iteration}$) increases. $r \vec{d}_1$ and $r \vec{d}_2$ are random vectors in $[0, 1]$. Fig.1 [26] shows the effects of Eqs.(1) and (2) in two-dimensional environment. In this figure, Spotted hyenas can update their position based on the location of their prey by adjusting the value of vectors \vec{B} and \vec{E} . Fig.2 [26] shows possible locations to update of spotted hyenas in the 3D environment. Locations of spotted hyenas are randomly updated around the prey by Eqs.(1) and (2). This concept can also be extended to n-dimensional Spaces.

2) HUNTING

Spotted hyenas typically hunt in packs, rely on a network of trusted friends and have the ability to spot prey. In order to mathematically define spotted hyena behavior, assume that the best search agents, whichever is optimum, know the location of the prey. Other search agents towards the best search agent and according to the best solution to update

TABLE 3. The fitness comparison of the proposed approaches on the first four test datasets.

No.	BreastCW			Congressional			ConnectBench			Dermatology		
	SHO	SHOS A-1	SHOS A-2	SHO	SHOS A-1	SHOS A-2	SHO	SHOS A-1	SHOS A-2	SHO	SHOS A-1	SHOS A-2
1	0.0541	0.0356	0.1022	0.1411	0.1361	0.1496	0.0811	0.0953	0.0954	0.0173	0.0185	0.0159
2	0.0398	0.0324	0.0413	0.1398	0.1289	0.1395	0.0795	0.0885	0.0644	0.0169	0.0158	0.0142
3	0.0377	0.0315	0.0411	0.1378	0.1275	0.1395	0.0681	0.0878	0.0624	0.0166	0.0157	0.0138
4	0.0476	0.0335	0.0475	0.1354	0.1299	0.1375	0.0688	0.0865	0.0622	0.0157	0.0158	0.0139
5	0.0359	0.0304	0.0367	0.1341	0.1266	0.1368	0.0678	0.0776	0.0591	0.0153	0.0147	0.0131
6	0.0354	0.0304	0.0366	0.1339	0.1265	0.1366	0.0675	0.0765	0.0589	0.0152	0.0145	0.0129
7	0.0375	0.0342	0.0396	0.1386	0.1312	0.1484	0.0692	0.0724	0.0663	0.0184	0.0151	0.0151
8	0.0349	0.0332	0.0375	0.1365	0.1285	0.1355	0.0672	0.0712	0.0621	0.0163	0.0146	0.0143
9	0.0349	0.0329	0.0361	0.1362	0.1276	0.1321	0.0668	0.0712	0.0618	0.0162	0.0146	0.0133

TABLE 4. The computational time (sec) comparison of the proposed approaches on the first four test datasets.

No.	BreastCW			Congressional			ConnectBench			Dermatology		
	SHO	SHOS A-1	SHOS A-2	SHO	SHOS A-1	SHOS A-2	SHO	SHOS A-1	SHOS A-2	SHO	SHOS A-1	SHOS A-2
1	4.51	50.54	4.47	4.49	41.82	4.76	4.26	40.24	4.13	4.66	45.65	4.71
2	7.32	80.79	7.44	7.56	66.75	7.84	7.21	63.51	7.13	7.61	73.82	7.88
3	11.58	119.25	10.98	11.54	104.35	12.03	11.53	98.05	11.59	11.43	108.44	11.58
4	11.76	125.43	11.01	11.84	103.77	11.98	11.76	99.16	11.84	12.12	110.35	12.01
5	23.86	252.21	24.84	22.65	208.56	22.04	20.58	204.54	20.16	22.64	224.46	22.79
6	34.06	384.12	33.64	32.38	315.54	32.64	30.15	308.96	29.35	36.91	342.68	35.05
7	20.28	204.56	21.52	20.13	167.21	20.84	19.32	162.55	19.81	20.25	183.21	20.58
8	36.51	401.53	35.88	38.94	333.41	37.85	35.15	326.54	34.97	38.84	354.27	37.43
9	59.53	621.58	60.28	58.04	502.56	57.01	55.32	475.45	56.98	60.09	573.24	58.86

their location. The following equations are proposed in this mechanism:

$$\vec{D}_h = |\vec{B} \cdot \vec{P}_h - \vec{P}_k| \tag{6}$$

$$\vec{P}_k = \vec{P}_h - \vec{E} \cdot \vec{D}_h \tag{7}$$

$$\vec{C}_h = \vec{P}_k + \vec{P}_{k+1} + \dots + \vec{P}_{k+N} \tag{8}$$

where \vec{P}_h indicates the position of first best spotted hyena, \vec{P}_k defines the position of other spotted hyenas. Here, N defines the number of spotted hyenas which is computed as follows:

$$N = \text{count}_{\text{nos}}(\vec{P}_h, \vec{P}_{h+1}, \vec{P}_{h+2}, \dots, (\vec{P}_h + \vec{M})) \tag{9}$$

where \vec{M} is a random vector with a value of [0.5, 1], nos defines the number of solutions and count all candidate solutions, after addition with \vec{M} , which are far similar to the best optimal solution in search space, and \vec{C}_h is a group or cluster of N number of optimal solutions.

3) ATTACKING PREY (EXPLOITATION)

The value of the vector \vec{h} is reduced in order to mathematically model for attacking the prey. The variation in vector \vec{E} is also decreased to change the value in vector \vec{h} which can decrease from 5 to 0 over the course of iterations. The group of Spotted hyenas attack prey when $|E| < 1$. The mathematical formulation for attacking the prey

TABLE 5. Parameters of the compared algorithms.

Reference	Algorithm	Parameters	Value
[49]	ACO	Pheromone volatility	0.8
		Transfer probability constant	0.2
[50]	SSA	Controlling parameter c_1	[0,2]
[51-52]	FPA	Switch possibility	0.5
		Switch possibility	0.4
		Lévy constant λ	1.5
[53-54]	PSO	Maximum inertia weight	0.9
		Minimum inertia weight	0.4
		Learning factors c_1 and c_2	2
		Maximum velocities	+120
[55]	SCA	Minimum velocities	-120
		Dictation parameter r_1	[0,2]
		Destination parameter r_2	[0,2 π]
		Random weight r_3	[0,2]
		Switch possibility r_4	[0,1]
[56]	CS	Mutation probability value P_a	0.25
		Scale factor β	1.5

is as follows:

$$\vec{P}(x+1) = \frac{\vec{C}_h}{N} \tag{10}$$

TABLE 6. Comparison between the proposed approaches based on classification accuracy and average number of features.

Dataset	Accuracy			Attributes		
	SHO	SHOSA-1	SHOSA-2	SHO	SHOSA-1	SHOSA-2
BreastCW	0.9689	0.9726	0.9677	4.6	4.5	4.9
Congressional	0.8676	0.8725	0.8661	5.6	5.3	5.5
ConnectBench	0.9349	0.9404	0.9433	16.7	16.1	18.8
Dermatology	0.9883	0.9896	0.9907	12.2	13.5	13.3
Drug_consumption	0.7311	0.7433	0.7360	14.4	11.5	15.4
Glass	0.9866	0.9850	0.9813	2.4	1.9	2.1
Heart	0.8326	0.8385	0.8259	4.6	4.2	4.4
Hepatitis	0.7449	0.7372	0.7410	6.4	5.6	6.1
Horse-colic	0.8776	0.8793	0.8827	7.9	7.6	7.8
ILPD	0.7142	0.7134	0.7250	3.1	2.1	2.2
Ionosphere	0.9451	0.9517	0.9466	10.3	10.8	9.1
Primary-tumor	0.7692	0.7735	0.7841	7.2	6.8	7.7
Seeds	0.9400	0.9410	0.9333	3.2	2.8	2.6
Soybean	0.9870	0.9864	0.9896	4.6	4.1	4.9
Spambase	0.9294	0.9312	0.9319	28.4	29.2	28.5
SPECT Heart	0.7727	0.7830	0.7691	8.6	8.2	7.5
SteelPF	1.0000	1.0000	1.0000	6.0	6.0	6.0
ThoracicSurgery	0.8607	0.8626	0.8532	3.4	2.9	3.2
Tic-tac-toe	0.7956	0.7916	0.7942	6.1	6.2	5.7
Zoo	0.9784	0.9882	0.9803	5.6	5.4	5.1
Average	0.8812	0.8841	0.8821	8.07	7.74	8.04

TABLE 7. Comparison between the proposed approaches based on best and worst fitness measure.

Dataset	best			worst		
	SHO	SHOSA-1	SHOSA-2	SHO	SHOSA-1	SHOSA-2
BreastCW	0.0271	0.0208	0.0231	0.0469	0.0412	0.0554
Congressional	0.1082	0.1065	0.1071	0.1655	0.1487	0.1545
ConnectBench	0.0412	0.0314	0.0414	0.0985	0.1074	0.0892
Dermatology	0.0032	0.0090	0.0081	0.0303	0.0261	0.0201
Drug_consumption	0.2548	0.2208	0.2548	0.2871	0.2776	0.2860
Glass	0.0011	0.0010	0.002	0.0483	0.0410	0.0483
Heart	0.1362	0.1435	0.1427	0.2160	0.1793	0.2022
Hepatitis	0.2041	0.2321	0.2089	0.3189	0.2935	0.3083
Horse-colic	0.0946	0.0939	0.1009	0.1628	0.1628	0.1361
ILPD	0.2588	0.2690	0.2350	0.3209	0.3198	0.3028
Ionosphere	0.0309	0.0199	0.0249	0.0783	0.0815	0.0930
Primary-tumor	0.1975	0.1893	0.1975	0.2784	0.2469	0.2609
Seeds	0.0420	0.0406	0.0420	0.0891	0.0811	0.0877
Soybean	0.0012	0.0070	0.0012	0.0272	0.0400	0.0523
Spambase	0.0707	0.0695	0.0667	0.0783	0.0786	0.0794
SPECT Heart	0.1840	0.1612	0.1726	0.2780	0.2564	0.2797
SteelPF	0.0018	0.0018	0.0018	0.0018	0.0018	0.0018
ThoracicSurgery	0.1162	0.1150	0.1192	0.1704	0.1645	0.1649
Tic-tac-toe	0.1962	0.1947	0.1989	0.2319	0.2267	0.2257
Zoo	0.0013	0.0013	0.0031	0.0808	0.0232	0.0607
Average	0.0986	0.0964	0.0976	0.1505	0.1399	0.1455

where $\vec{P}(x + 1)$ updates the positions of other search agents according to the position of the best search agent and save the best solution. The SHO algorithm allows its search agents to update their positions and attack towards the prey.

4) SEARCH FOR PREY (EXPLORATION)

Spotted hyenas mainly search for prey according to the location of spotted hyena group in vector \vec{C}_h . They stay away from each other to find and attack for prey. It's good for

TABLE 8. Comparison between the proposed approaches based on mean fitness and standard deviation measure.

Dataset	mean			std		
	SHO	SHOSA-1	SHOSA-2	SHO	SHOSA-1	SHOSA-2
BreastCW	0.0359	0.0328	0.0377	0.0077	0.0065	0.0088
Congressional	0.1343	0.1298	0.1368	0.0150	0.0155	0.0151
ConnectBench	0.0672	0.0620	0.0593	0.0177	0.0132	0.0145
Dermatology	0.0151	0.0142	0.0131	0.0067	0.0060	0.0038
Drug_consumption	0.2710	0.2580	0.2665	0.0088	0.0085	0.0099
Glass	0.0157	0.0167	0.0205	0.0121	0.0150	0.0146
Heart	0.1695	0.1634	0.1760	0.0206	0.0106	0.0183
Hepatitis	0.2555	0.2631	0.2610	0.0263	0.0215	0.0320
Horse-colic	0.1239	0.1223	0.1229	0.0177	0.0125	0.0130
ILPD	0.2864	0.2864	0.2781	0.0155	0.0150	0.0197
Ionosphere	0.0575	0.0511	0.0556	0.0127	0.0178	0.0232
Primary-tumor	0.2326	0.2282	0.2190	0.0182	0.0162	0.0225
Seeds	0.0637	0.0625	0.0697	0.0120	0.0110	0.0121
Soybean	0.0103	0.0147	0.0117	0.0068	0.0095	0.0150
Spambase	0.0749	0.0733	0.0724	0.0028	0.0022	0.0038
SPECT Heart	0.2289	0.2186	0.2320	0.0239	0.0257	0.0317
SteelPF	0.0018	0.0018	0.0018	0.0000	0.0000	0.0000
ThoracicSurgery	0.1400	0.1379	0.1479	0.0166	0.0189	0.0162
Tic-tac-toe	0.2092	0.2132	0.2101	0.0098	0.0121	0.0084
Zoo	0.0248	0.0150	0.0226	0.0206	0.0112	0.0202
Average	0.1209	0.1183	0.1207	0.0136	0.0124	0.0151

spotted hyenas to leave their prey when $|E| > 1$. This mechanism allows SHO to perform global searches. Another component of SHO is \vec{B} , which makes exploration possible. In Eq.(3), the random value that provides the random weight of the prey is contained in \vec{B} vector. We assume that vector $\vec{B} > 1$ takes precedence over vector $\vec{B} < 1$ to show the randomness of the SHO algorithm. This will help global search and avoid local optimization. Random values are provided from \vec{B} vector for not only the initial iteration, but also the final iteration. This mechanism is very helpful to avoid local optimization problems especially in the final iteration. Finally, the algorithm is terminated when the termination condition is satisfied.

Here are some points to note:

- The optimal solution is saved in the iteration by the proposed algorithm.
- The proposed enveloping mechanism defines a circular solution that can be extended to higher dimensions as a hypersphere.
- The random vectors \vec{B} and \vec{E} can make the candidate solutions random.
- The proposed search method allows candidate solutions to locate possible prey locations.
- The values of the vectors \vec{E} and \vec{h} are adjusted to allow SHO to explore ($|E| \geq 1$) or exploit ($|E| \leq 1$).

5) ALGORITHM OF SHO

B. SIMULATED ANNEALING (SA)

Simulated annealing is an algorithm based on solid annealing principle proposed by Kirkpatrick et al. Simulated annealing

TABLE 9. Comparison between the proposed approaches based on average computational time(sec).

Dataset	SHO	SHOSA-1	SHOSA-2
BreastCW	23.96	250.45	24.81
Congressional	23.47	240.69	22.97
ConnectBench	20.41	216.38	20.89
Dermatology	21.94	234.33	22.75
Drug_consumption	72.34	795.02	79.51
Glass	20.05	208.84	20.50
Heart	19.69	204.48	20.42
Hepatitis	18.97	194.98	19.81
Horse-colic	19.99	205.40	20.99
ILPD	21.07	213.07	21.60
Ionosphere	22.86	211.67	21.61
Primary-tumor	21.74	208.41	21.47
Seeds	20.25	194.55	19.98
Soybean	21.24	210.23	21.72
Spambase	539.72	5800.57	586.85
SPECT Heart	20.21	195.27	21.26
SteelPF	58.39	802.82	64.06
ThoracicSurgery	22.49	225.08	24.08
Tic-tac-toe	29.04	279.25	30.79
Zoo	20.31	200.35	21.98
Average	51.91	554.59	55.40

algorithm accepts a solution worse than the current solution with a certain probability, so it is possible to jump out of the local optimal solution and reach the global optimal solution. The algorithm starts from the randomly generated initial solution, generates the best neighborhood solution so far in each

Algorithm 1 The Pseudocode of Spotted Hyena Optimizer

```

Input: the spotted hyenas population  $P_i$ 
( $i = 1, 2, \dots, n$ )
Output: the best search agent
1 procedure SHO
2 Initialize the parameters  $h, B, E, N$ 
3 Calculate the fitness of each search agent
4  $P_h$  = the best search agent
5  $C_h$  = the group or cluster of all far optimal
solutions
6 while ( number o f iterations) do
7 for each search agent do
8 Update the position of current
agent by Eq. (10)
10 end for
11 Update  $h, B, E, N$ 
12 Check if any search agent goes beyond the
given search space and then adjust it
14 Calculate the fitness of each search agent
17 Update  $P_h$  if there is a better solution than
previous optimal solution
19 Update the group  $C_h$  w.r.t  $P_h$ 
20  $x = x + 1$ 
21 end while
22 return  $P_h$ 
23 end procedure

```

iteration according to the predefined neighborhood structure, and evaluates it with the fitness function. The improving move (the neighbor is fitter than the original solution) is always accepted, whilst worse neighbor is accepted with a certain probability determined by the Boltzmann probability, $P = e^{-\theta/T}$ where θ is the difference between the fitness of the best solution ($BestSol$) and the generated neighbor ($TrialSol$). In addition, T is a parameter (called temperature) that is periodically decreased during the search according to some cooling plan. In this work, the initial temperature is set as $2 * |N|$, where $|N|$ represents the number of attributes of each dataset, and the cooling time is calculated by $T = 0.93 * T$ [33]. Algorithm 2 gives the pseudocode of SA algorithm.

IV. THE PROPOSED APPROACH**A. BINARY VERSION AND FITNESS FUNCTION**

The essence of feature selection is binary optimization problem, whose solution is limited to binary $\{0,1\}$ values. A binary version needs to be developed for the SHO algorithm used in feature selection problems. In this work, one solution is represented by a one-dimensional vector whose length is the number of attributes in the original dataset. Each value in the solution is represented by a “1” or “0”. The value “1” indicates that the property is selected, and the value “0” indicates that the property is not selected.

Feature selection can also be regarded as a multi-objective optimization problem, in which two objectives

Algorithm 2 The Pseudocode of SA Algorithm

```

1  $T_0 = 2 * |N|$  where  $|N|$  is the number of attributes
for each dataset
2  $BestSol \leftarrow S_i$ 
3  $\delta(BestSol) \leftarrow \delta(S_i)/\delta$  indicates the quality of
the solution
4 while  $T > T_0$ 
5 generate at random a new solution  $TrialSol$  in the
neighbor of  $S'_i$ 
6 calculate  $\delta(TrialSol)$ 
7 if ( $\delta(TrialSol) > \delta(BestSol)$ )
8  $S'_i \leftarrow TrialSol$ 
9  $BestSol \leftarrow TrialSol$ 
10  $\delta(S_i) \leftarrow \delta(TrialSol)$ 
11  $\delta(BestSol) \leftarrow \delta(TrialSol)$ 
12 else if ( $\delta(TrialSol) = \delta(BestSol)$ )
13 Calculate  $|TrialSol|$  and  $|BestSol|$ 
14 if ( $|TrialSol| < |BestSol|$ )
15  $S'_i \leftarrow TrialSol$ 
16  $BestSol \leftarrow TrialSol$ 
17  $\delta(S_i) \leftarrow \delta(TrialSol)$ 
18  $\delta(BestSol) \leftarrow \delta(TrialSol)$ 
19 end if
20 else // accepting the worse solution
21 Calculate  $\theta = \delta(TrialSol) - \delta(BestSol)$ 
22 Generate a random number,  $P = [0,1]$ 
23 if ( $P \leq e^{-\theta/T}$ )
24  $S'_i \leftarrow TrialSol$ 
25  $\delta(S_i) \leftarrow \delta(TrialSol)$ 
26 end if
27 end if
28  $T = 0.93 * T$  // update temperature
29 end while
30 Output  $BestSol$ 

```

are contradictory: The fewer features selected, the higher the classification accuracy. When the number of features in the solution is less and the classification accuracy is higher, the solution is better. According to the classification accuracy of the solution obtained by KNN classifier [47] and the number of features selected in the solution, a fitness function is proposed in Eq.(11).

$$Fitness = \alpha \gamma_R(D) + \beta \frac{|R|}{|N|} \quad (11)$$

where $\gamma_R(D)$ represents the given classification error rate. In addition, $|R|$ and $|N|$ are the cardinality of the selected subset and the total number of feature datasets, respectively. The parameters α and β correspond to the accuracy of classification and the importance of the number of selected features.

Each solution is evaluated by the fitness function during the iteration. Both SHO and SA algorithms use this fitness function to evaluate the search agents in order to balance the higher classification accuracy and the minimum

TABLE 10. Comparison between the proposed SHOSA-1 and other algorithms based on the classification accuracy.

Dataset	ACO	CS	FPA	PSO	SCA	SSA	SHOSA-1
BreastCW	0.8584	0.9503	0.9649	0.9326	0.9637	0.9557	0.9726
Congressional	0.7688	0.8349	0.8518	0.8119	0.8495	0.8528	0.8725
ConnectBench	0.6827	0.7692	0.8538	0.7885	0.8606	0.8183	0.9404
Dermatology	0.6501	0.7978	0.9716	0.8077	0.9590	0.8716	0.9896
Drug_consumption	0.6691	0.7041	0.7075	0.6740	0.6883	0.7021	0.7433
Glass	0.4836	0.6953	0.9869	0.7972	0.9822	0.9860	0.9850
Heart	0.6761	0.7556	0.7993	0.6370	0.7704	0.7830	0.8385
Hepatitis	0.5337	0.5513	0.6628	0.5436	0.6462	0.5769	0.7372
Horse-colic	0.6625	0.6987	0.7573	0.6153	0.7533	0.7567	0.8793
ILPD	0.6093	0.6637	0.6938	0.6428	0.6966	0.6777	0.7134
Ionosphere	0.8272	0.8693	0.8949	0.8375	0.8938	0.8722	0.9517
Primary-tumor	0.6057	0.7376	0.7388	0.6329	0.7324	0.7265	0.7735
Seeds	0.7714	0.8914	0.9305	0.8162	0.9324	0.9162	0.9410
Soybean	0.9686	0.9779	0.9851	0.9747	0.9786	0.9857	0.9864
Spambase	0.7528	0.7610	0.9032	0.8178	0.8970	0.7970	0.9312
SPECT Heart	0.5699	0.6649	0.7160	0.5830	0.7096	0.6649	0.7830
SteelPF	0.9488	0.9998	0.7356	0.5938	0.7129	1.0000	1.0000
ThoracicSurgery	0.7251	0.7736	0.8353	0.7579	0.8247	0.7940	0.8626
Tic-tac-toe	0.5995	0.7290	0.7843	0.6116	0.7568	0.7626	0.7916
Zoo	0.7517	0.9137	0.9725	0.8353	0.9647	0.9216	0.9882
Average	0.7058	0.7870	0.8373	0.7356	0.8286	0.8211	0.8841

TABLE 11. Comparison between the proposed SHOSA-1 and other algorithms based on average number of features.

Dataset	ACO	CS	FPA	PSO	SCA	SSA	SHOSA-1
BreastCW	2.2	6.4	5.3	3.9	6.1	6.5	4.5
Congressional	6.4	5.8	5.3	7.9	6.5	5.9	5.3
ConnectBench	24.0	34.2	34.3	27.3	27.7	43.5	16.1
Dermatology	11.5	21.4	24.4	15.5	20.2	32.1	13.5
Drug_consumption	14.1	14.4	20.8	13.9	15.3	19.1	11.5
Glass	3.3	7.8	4.6	5.0	4.6	10.0	1.9
Heart	6.9	5.2	6.1	5.3	5.5	2.6	4.2
Hepatitis	10.5	5.6	9.2	10.3	7.6	10.3	5.6
Horse-colic	14.5	10.0	10.8	13.8	11.7	7.6	7.6
ILPD	4.3	4.2	5.2	4.1	3.7	3.6	2.1
Ionosphere	17.4	14.2	15.2	16.3	12.9	16.7	10.8
Primary-tumor	7.4	10.8	10.4	8.6	7.8	11.5	6.8
Seeds	2.3	4.0	3.3	3.1	3.0	6.4	2.8
Soybean	19.9	14.2	16.7	17.3	13.3	23.9	4.1
Spambase	24.9	34.6	41.9	25.3	31.7	55.0	29.2
SPECT Heart	11.6	10.6	13.2	12.2	10.1	12.1	8.2
SteelPF	20.3	10.4	19.9	18.0	16.2	6.0	6.0
ThoracicSurgery	7.5	8.5	8.6	9.1	6.2	8.6	2.9
Tic-tac-toe	3.0	6.3	6.6	3.6	5.6	9.0	6.2
Zoo	6.3	10.2	9.3	7.6	9.5	13.3	5.4
Average	10.92	11.94	13.56	11.41	11.26	15.19	7.74

number of selected features in each solution, $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$ [13].

B. HYBRID SHO-SA METHODS

SHO algorithm is an optimization algorithm that has achieved good results in many optimization problems in recent years. We regard the solution obtained by SHO algorithm as

the initial state, and then use SA algorithm to optimize the solution. Two hybrid models of the two algorithms are considered in this method. On one hand, SA algorithm is embedded in SHO algorithm to find the best solution and replace the original solution in the neighborhood of the randomly selected solution and the known optimal solution. In this method, SA algorithm is the operator of

TABLE 12. Comparison between the proposed SHOSA-1 and other algorithms based on best fitness measure.

Dataset	ACO	CS	FPA	PSO	SCA	SSA	SHOSA-1
BreastCW	0.0939	0.0356	0.0350	0.0225	0.0343	0.0378	0.0208
Congressional	0.1965	0.1201	0.1123	0.1207	0.1337	0.1324	0.1065
ConnectBench	0.3154	0.1485	0.0987	0.0609	0.0708	0.1298	0.0314
Dermatology	0.3696	0.1471	0.0241	0.0215	0.0320	0.1233	0.0090
Drug_consumption	0.4013	0.2901	0.2845	0.2692	0.2968	0.2840	0.2208
Glass	0.4342	0.2486	0.0020	0.0020	0.0040	0.0100	0.0010
Heart	0.2975	0.2078	0.1598	0.1647	0.1940	0.1785	0.1435
Hepatitis	0.4806	0.3966	0.2596	0.2972	0.2829	0.3902	0.2321
Horse-colic	0.3466	0.2406	0.1621	0.1694	0.1812	0.2076	0.0939
ILPD	0.4407	0.2825	0.2722	0.2689	0.2757	0.2893	0.2690
Ionosphere	0.1697	0.0945	0.0883	0.0661	0.0883	0.0827	0.0199
Primary-tumor	0.3822	0.2284	0.2342	0.2272	0.2202	0.2551	0.1893
Seeds	0.1050	0.0717	0.0420	0.0514	0.0543	0.0746	0.0406
Soybean	0.0143	0.0152	0.0041	0.0032	0.0082	0.0085	0.0070
Spambase	0.2907	0.2335	0.0948	0.0981	0.0894	0.2054	0.0695
SPECT Heart	0.4396	0.2972	0.2555	0.2033	0.2353	0.2743	0.1612
SteelPF	0.0068	0.0024	0.2134	0.0652	0.2519	0.0018	0.0018
ThoracicSurgery	0.2869	0.1849	0.1211	0.1367	0.1331	0.1481	0.1150
Tic-tac-toe	0.3674	0.2393	0.1835	0.1793	0.2143	0.2270	0.1947
Zoo	0.1131	0.0445	0.0050	0.0044	0.0056	0.0469	0.0013
Average	0.2776	0.1765	0.1326	0.1216	0.1403	0.1554	0.0964

TABLE 13. Comparison between the proposed SHOSA-1 and other algorithms based on worst fitness measure.

Dataset	ACO	CS	FPA	PSO	SCA	SSA	SHOSA-1
BreastCW	0.3577	0.0791	0.0469	0.0384	0.0496	0.0582	0.0412
Congressional	0.3601	0.2317	0.1707	0.1649	0.1811	0.1993	0.1487
ConnectBench	0.4921	0.2821	0.2138	0.1662	0.1772	0.2151	0.1074
Dermatology	0.5439	0.2914	0.0404	0.0546	0.0537	0.1507	0.0261
Drug_consumption	0.4388	0.3462	0.3090	0.2937	0.3127	0.3165	0.2776
Glass	0.9162	0.4224	0.0523	0.0308	0.0595	0.0470	0.0410
Heart	0.5238	0.3285	0.2698	0.2177	0.2845	0.2437	0.1793
Hepatitis	0.6888	0.5362	0.3971	0.3718	0.4093	0.4580	0.2935
Horse-colic	0.5166	0.3825	0.3073	0.2285	0.3275	0.3066	0.1628
ILPD	0.5905	0.3649	0.3366	0.3096	0.3254	0.3458	0.3198
Ionosphere	0.2959	0.1792	0.1327	0.1824	0.1389	0.1666	0.0815
Primary-tumor	0.6508	0.4124	0.2988	0.2726	0.3069	0.3052	0.2469
Seeds	0.5439	0.1566	0.0986	0.0877	0.0891	0.1123	0.0811
Soybean	0.1126	0.0354	0.0482	0.0301	0.0626	0.0482	0.0400
Spambase	0.3307	0.2605	0.1155	0.1227	0.1264	0.2239	0.0786
SPECT Heart	0.8471	0.3745	0.3241	0.3315	0.3411	0.3979	0.2564
SteelPF	0.5175	0.0044	0.2979	0.2789	0.3552	0.0018	0.0018
ThoracicSurgery	0.4304	0.3313	0.1977	0.1826	0.2205	0.2742	0.1645
Tic-tac-toe	0.6700	0.3052	0.2434	0.2319	0.2763	0.2704	0.2267
Zoo	0.6211	0.1803	0.0839	0.1008	0.1014	0.2029	0.0232
Average	0.5224	0.2752	0.1992	0.1849	0.2099	0.2172	0.1399

SHO algorithm. This method is called SHOSA-1. On the other hand, SA algorithm enhances the final solution found by SHO algorithm. This method is called SHOSA-2. The flowchart of the SHO, SHOSA-1 and SHOSA-2 is summarized in fig.3.

C. COMPUTATIONAL COMPLEXITY

1) TIME COMPLEXITY

In the population initialization phase, SHO has the computational complexity $O(N)$. For the main iteration loop, the computational complexity is $O(dim)$.

TABLE 14. Comparison between the proposed SHOSA-1 and other algorithms based on mean fitness measure.

Dataset	ACO	CS	FPA	PSO	SCA	SSA	SHOSA-1
BreastCW	0.1783	0.0563	0.0402	0.0317	0.0426	0.0511	0.0328
Congressional	0.2914	0.1674	0.1502	0.1411	0.1533	0.1497	0.1298
ConnectBench	0.3977	0.2342	0.1504	0.1098	0.1427	0.1872	0.0620
Dermatology	0.4336	0.2065	0.0353	0.0338	0.0465	0.1366	0.0142
Drug_consumption	0.4153	0.2977	0.2965	0.2837	0.3137	0.3013	0.2580
Glass	0.6432	0.3096	0.0176	0.0135	0.0222	0.0239	0.0167
Heart	0.4081	0.2463	0.2038	0.1943	0.2319	0.2170	0.1634
Hepatitis	0.5840	0.4472	0.3386	0.3236	0.3543	0.4243	0.2631
Horse-colic	0.4244	0.3020	0.2442	0.2040	0.2485	0.2437	0.1223
ILPD	0.4894	0.3374	0.3089	0.2853	0.3045	0.3230	0.2864
Ionosphere	0.2204	0.1337	0.1087	0.0978	0.1091	0.1316	0.0511
Primary-tumor	0.4934	0.2661	0.2647	0.2540	0.2696	0.2776	0.2282
Seeds	0.2870	0.1132	0.0735	0.0700	0.0712	0.0921	0.0625
Soybean	0.0461	0.0260	0.0197	0.0161	0.0251	0.0212	0.0147
Spambase	0.3114	0.2427	0.1032	0.1075	0.1075	0.2106	0.0733
SPECT Heart	0.5388	0.3366	0.2872	0.2661	0.2921	0.3372	0.2186
SteelPF	0.0710	0.0034	0.2678	0.2063	0.2892	0.0018	0.0018
ThoracicSurgery	0.3461	0.2294	0.1684	0.1609	0.1774	0.2093	0.1379
Tic-tac-toe	0.4998	0.2753	0.2208	0.2120	0.2470	0.2450	0.2132
Zoo	0.3125	0.0918	0.0330	0.0434	0.0409	0.0860	0.0150
Average	0.3696	0.2161	0.1666	0.1527	0.1745	0.1835	0.1183

TABLE 15. Comparison between the proposed SHOSA-1 and other algorithms based on standard deviation of the fitness measure.

Dataset	ACO	CS	FPA	PSO	SCA	SSA	SHOSA-1
BreastCW	0.0783	0.0092	0.0041	0.0054	0.0055	0.0063	0.0065
Congressional	0.0485	0.0278	0.0185	0.0148	0.0131	0.0191	0.0155
ConnectBench	0.0618	0.0353	0.0320	0.0269	0.0376	0.0246	0.0132
Dermatology	0.0519	0.0309	0.0060	0.0108	0.0083	0.0108	0.0060
Drug_consumption	0.0121	0.0143	0.0087	0.0078	0.0074	0.0107	0.0085
Glass	0.1088	0.0454	0.0153	0.0091	0.0180	0.0125	0.0150
Heart	0.0670	0.0310	0.0336	0.0167	0.0292	0.0222	0.0106
Hepatitis	0.0520	0.0377	0.0434	0.0220	0.0392	0.0251	0.0215
Horse-colic	0.0434	0.0303	0.0481	0.0177	0.0414	0.0292	0.0125
ILPD	0.0334	0.0214	0.0220	0.0135	0.0169	0.0153	0.0150
Ionosphere	0.0319	0.0249	0.0141	0.0338	0.0166	0.0246	0.0178
Primary-tumor	0.0799	0.0395	0.0220	0.0167	0.0287	0.0167	0.0162
Seeds	0.1235	0.0231	0.0164	0.0124	0.0118	0.0138	0.0110
Soybean	0.0206	0.0068	0.0120	0.0071	0.0153	0.0105	0.0095
Spambase	0.0138	0.0121	0.0068	0.0083	0.0100	0.0052	0.0022
SPECT Heart	0.0769	0.0270	0.0236	0.0403	0.0299	0.0399	0.0257
SteelPF	0.0998	0.0007	0.0252	0.0643	0.0302	0.0000	0.0000
ThoracicSurgery	0.0395	0.0425	0.0280	0.0129	0.0246	0.0367	0.0189
Tic-tac-toe	0.0584	0.0188	0.0185	0.0170	0.0161	0.0154	0.0121
Zoo	0.1218	0.0400	0.0292	0.0273	0.0297	0.0484	0.0112
Average	0.0612	0.0259	0.0214	0.0192	0.0215	0.0194	0.0124

The update process terminates within the maximum number of iterations, so the total computational complexity of SHO is $O(N \cdot dim \cdot Maxiter)$.

In SHOSA-1, SA is nested in each iteration of SHO. The computational complexity of SHOSA-1 is

$$O((N \cdot dim + O(SA)) \cdot Maxiter).$$

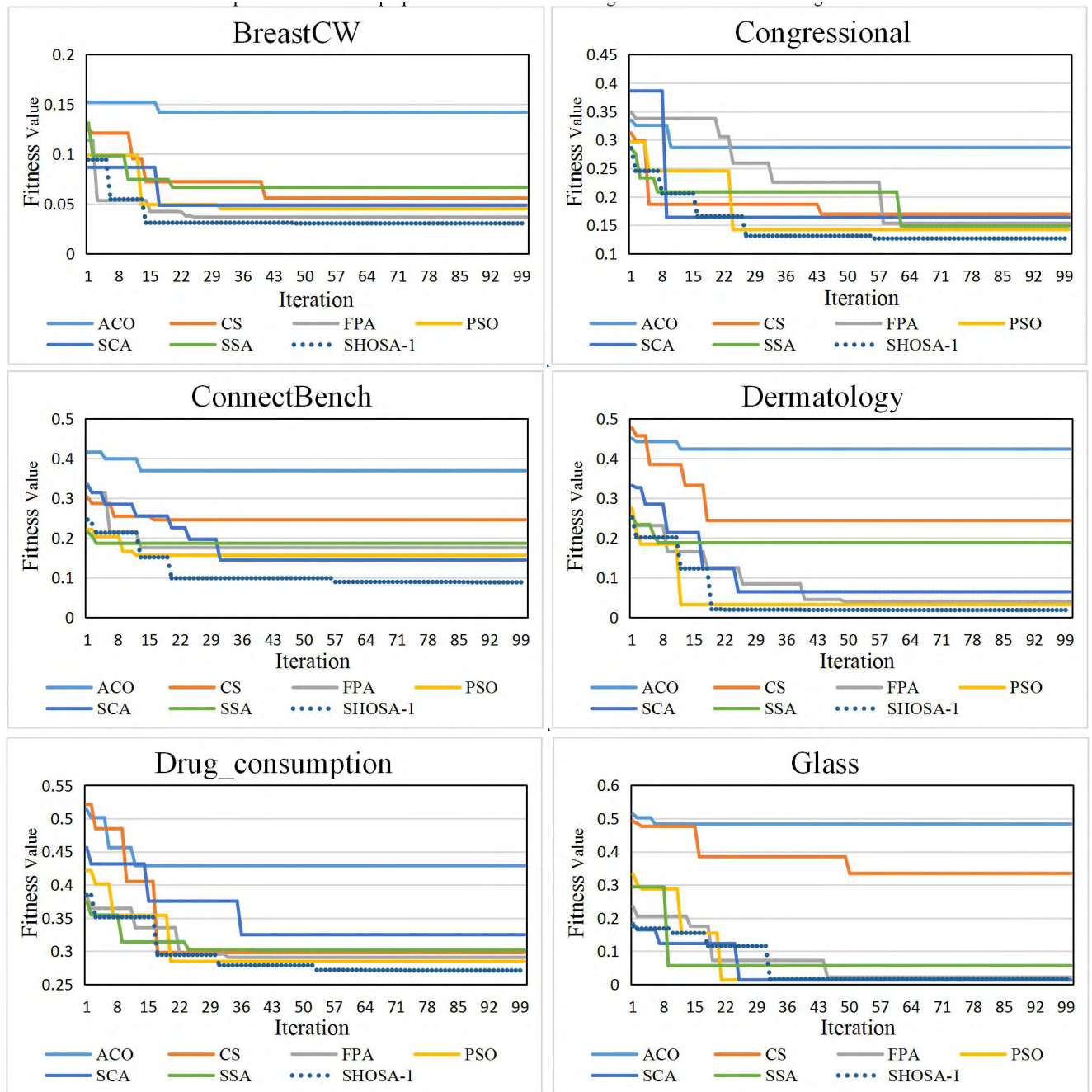
where $O(SA)$ represents the Computational complexity of SA.

In SHOSA-2, SA algorithm enhances the final solution found by SHO algorithm. The computational complexity of SHOSA-2 is $O(N \cdot dim \cdot Maxiter)$.

2) SPACE COMPLEXITY

The space complexity of SHO algorithm is the maximum amount of space used at any one time which is considered during its initialization process.

TABLE 16. Comparison between the proposed SHOSA-1 and other algorithms based on the convergence curve.



Thus, the total space complexity of SHO, SHOSA-1 and SHOSA-2 is $O(n \cdot dim)$.

V. EXPERIMENTS

A. DATASETS

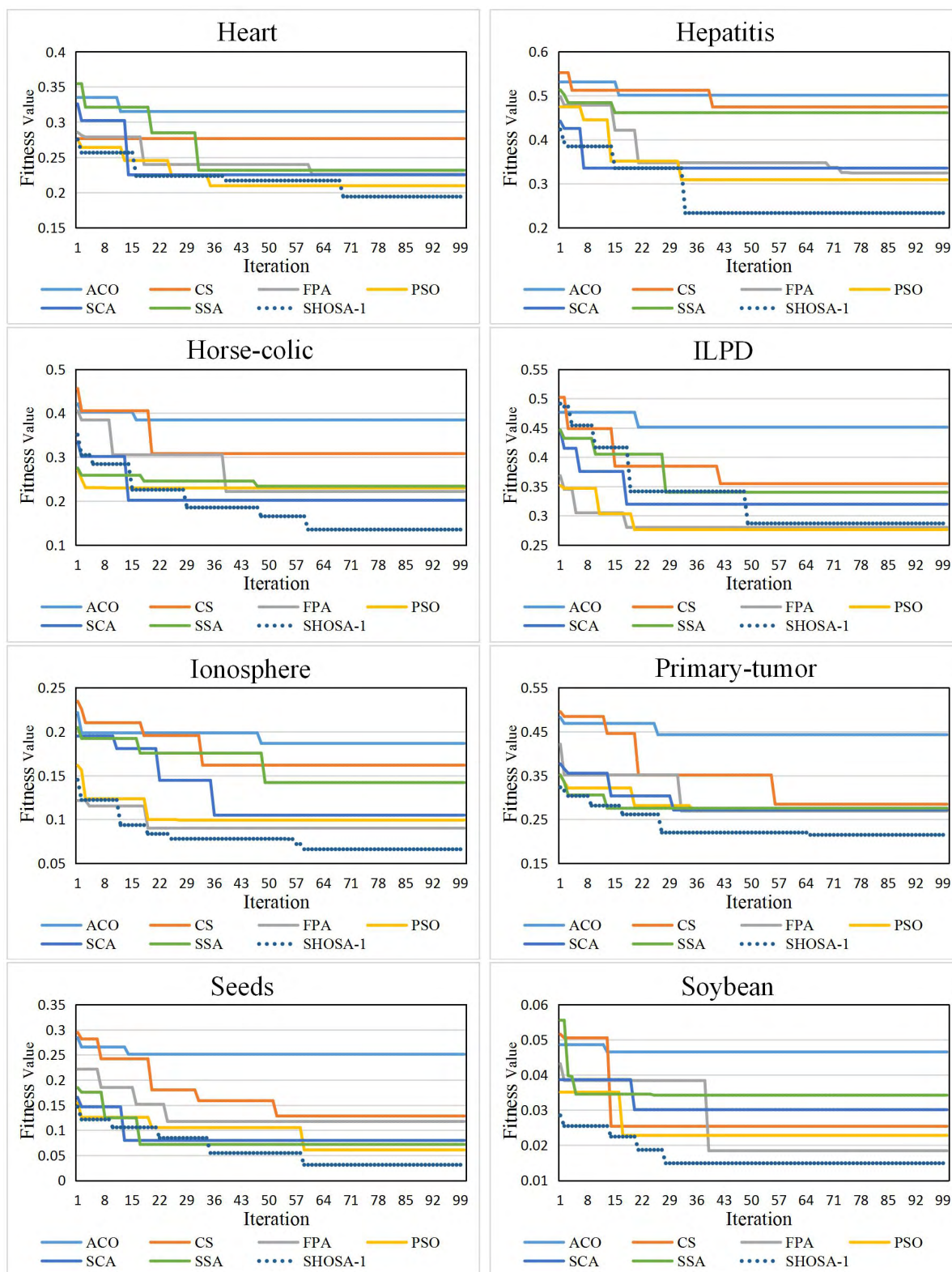
In this section, the implementation of the proposed three algorithms is done using MATLAB. In order to evaluate the performance of the proposed algorithms, the proposed algorithms are implemented on twenty FS benchmark datasets

from the UCI data repository [48]. The number of attributes and instances of each datasets are presented in Table 1. These datasets contain many attributes and instances that can be effectively tested against the proposed binary approach.

B. PARAMETER SETTINGS

In preliminary experiment, the population size and the maximum number of iterations were selected by testing the first

TABLE 16. (Continued.) Comparison between the proposed SHOSA-1 and other algorithms based on the convergence curve.



four datasets. For this experiments, the population size is selected as 10, 30 and 50, and the maximum number of iterations is selected as 50, 100 and 150. SHO, SHOSA-1

and SHOSA-2 are selected as the test algorithm in the initial experiment. Table 2 shows groups of the population size and the maximum number of iterations.

TABLE 16. (Continued.) Comparison between the proposed SHOSA-1 and other algorithms based on the convergence curve.

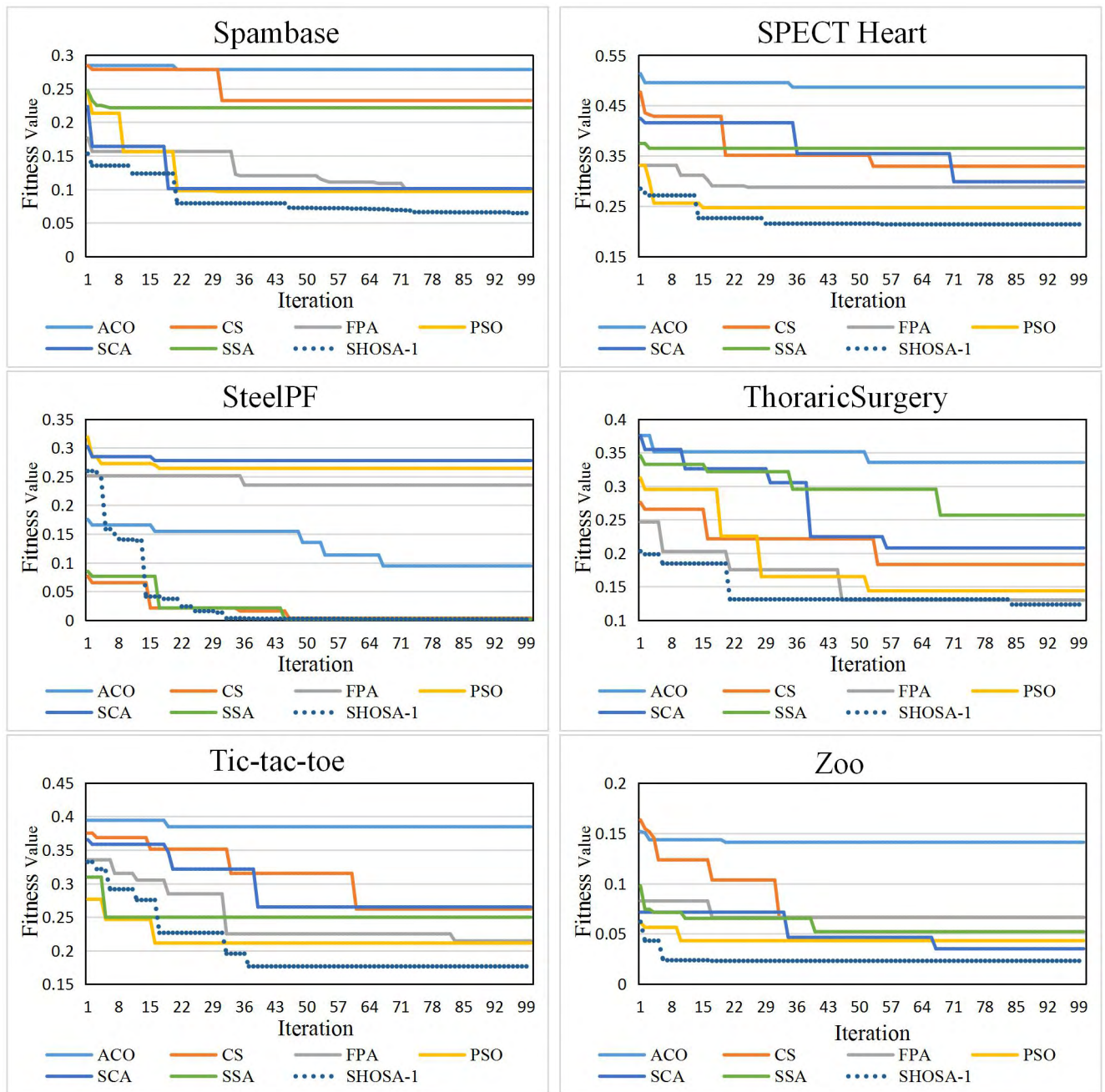


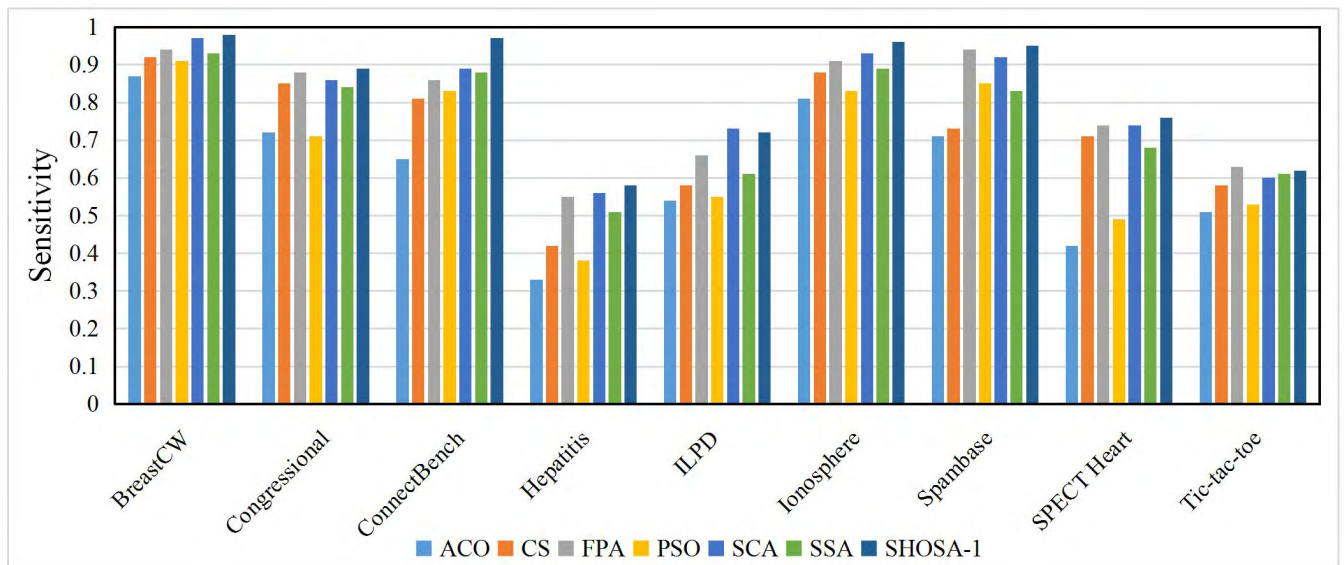
Table 3 shows the fitness of different algorithms on different datasets. Table 4 shows the running times of different algorithms in different datasets. From Table 3 and Table 4, the fifth group of parameters can achieve a relatively ideal fitness without taking much time. Therefore, the population size is 30 and the maximum number of iterations is 100. In addition, the exit condition of SA algorithm is the number of iterations. The particle number of SA is 10, and the maximum iteration number is 30.

Since the results obtained by running the optimization algorithm are accidental, we run all the algorithms separately

for 30 times. In addition, each algorithm runs on an i5 machine with a 3GHz CPU and 4GB of RAM.

In order to evaluate the effectiveness of the algorithm, each dataset is divided by cross-validation. The KNN classifier based on Euclidean distance matrix (where $K = 5$ [13]) is used to generate the best reduction. In the k-fold cross-validation, the k-1 fold is used for training and validation, and the remaining fold is used for testing. This process is repeated M times. Each optimization algorithm was evaluated $K \cdot M$ times for each dataset. The datasets are divided into training datas, verification datas and test datas. The training datas are

TABLE 17. Comparison between the proposed SHOSA-1 and other algorithms based on sensitivity.



used to train the classifier in the optimization process, and the validation datas are used to evaluate the performance of the classifier in the optimization process. The test datas are used to evaluate the selected feature data of the classifier. In addition, some classical or latest optimization algorithms are selected for comparison with the proposed algorithm. The parameters of ant colony optimization (ACO), salp swarm algorithm (SSA), flower pollination algorithm (FPA), particle swarm optimization(PSO), sine cosine algorithm (SCA) and cuckoo search (CS) were set in table 5 according to the references [49]–[56]. The reason for choosing these optimization algorithms are they are all applied to feature selection and both are tested on UCI similar datasets [12], [57]–[61].

C. EVALUATION CRITERIA

The following criteria are used to evaluate each optimization algorithm when it is run:

Average classification accuracy: the index of average classifier accuracy of selected feature set is described. The classification accuracy in this study is calculated as:

$$Mean = \frac{1}{M} \sum_{i=1}^M Accuracy(i) \tag{12}$$

where M is the number of times and $Accuracy(i)$ is classification accuracy in the optimization algorithm i^{th} run.

Average selection size: represents the average value of selected features for M times. Shown by the following:

$$Mean = \frac{1}{M} \sum_{i=1}^M Size(i) \tag{13}$$

where $Size(i)$ represents the number of features are selected in the optimization algorithm i^{th} run, M is represents the number of algorithm runs.

Best fitness: Represents the best fitness obtained by running the M times optimization algorithm.

Worst fitness: Represents the worst fitness obtained by running the M times optimization algorithm.

Average fitness: Represents the fitness average value of the solution obtained after M computations of the optimization algorithm, which can be expressed as:

$$Mean = \frac{1}{M} \sum_{i=1}^M Fitness(i) \tag{14}$$

where $Fitness(i)$ represents fitness in the optimization algorithm i^{th} run, M is represents the number of algorithm runs.

Statistical standard deviation (std): Represents the variation of the optimal solution obtained after M times of execution of the optimization algorithm, represent as:

$$Std = \sqrt{\frac{1}{M-1} \sum (Fitness(i) - Mean)^2} \tag{15}$$

where $Fitness(i)$ represents the fitness in the i^{th} run, M is represents the number of algorithm runs.

Average running time: represents the average time of each run of the algorithm in seconds. Calculate the following:

$$Mean = \frac{1}{M} \sum_{i=1}^M Runtime(i) \tag{16}$$

where $Runtime(i)$ represents the time in the i^{th} run. M is represents the number of algorithm runs.

Sensitivity and specificity: Sensitivity represents the proportion of the paired pairs in all positive examples, which measures the ability of the classifier to identify positive examples. Specificity represents the proportion of pairs in all negative examples, which measures the ability of the classifier to

TABLE 18. Comparison between the proposed SHOSA-1 and other algorithms based on specificity.

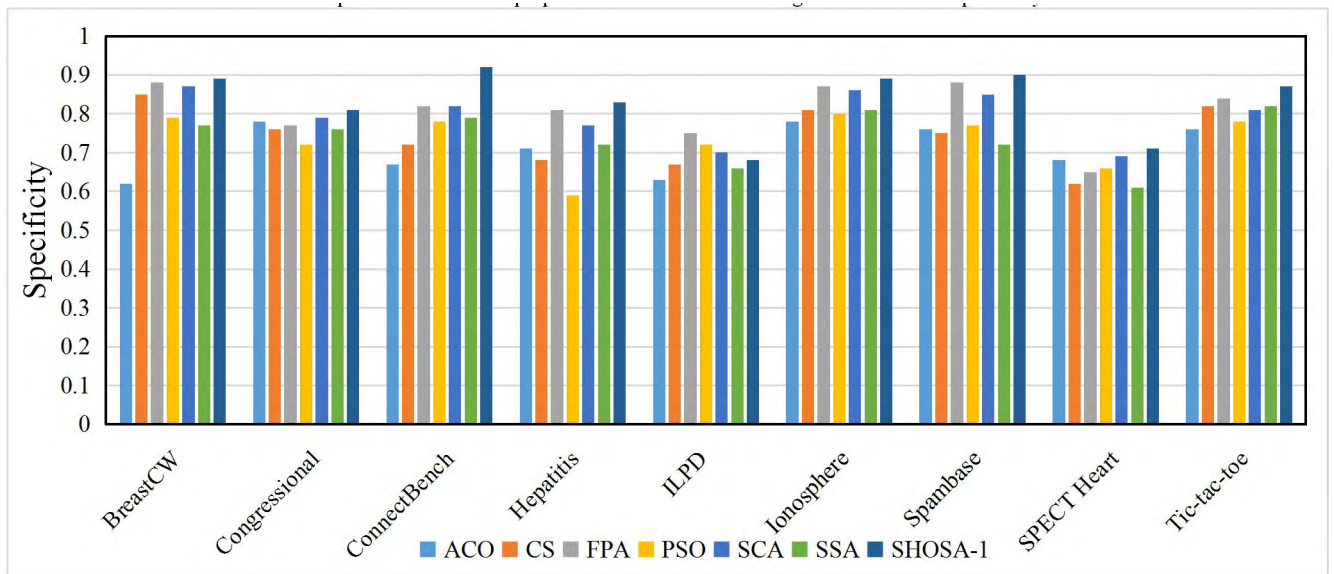


TABLE 19. Comparison between the proposed SHOSA-1 and other algorithms based on the average running time (sec).

Dataset	ACO	CS	FPA	PSO	SCA	SSA	SHOSA-1
BreastCW	73.01	24.19	22.18	21.76	21.82	24.21	250.45
Congressional	69.80	22.47	20.93	21.39	20.88	20.89	240.69
ConnectBench	61.75	20.15	18.78	19.07	18.66	19.43	216.38
Dermatology	67.02	22.36	20.77	21.15	20.74	22.28	234.33
Drug_consumption	235.52	75.83	72.55	68.68	69.94	92.55	795.02
Glass	58.25	21.12	18.94	19.04	18.87	19.27	208.84
Heart	55.44	17.75	18.67	18.78	18.45	18.72	204.48
Hepatitis	53.18	19.26	18.19	18.43	17.94	18.06	194.98
Horse-colic	57.20	20.25	19.09	19.29	19.37	19.35	205.40
ILPD	60.32	21.98	20.04	20.15	20.44	20.88	213.07
Ionosphere	59.91	20.99	19.89	19.91	19.79	20.57	211.67
Primary-tumor	60.47	21.68	19.47	19.71	19.42	19.93	208.41
Seeds	54.87	19.58	18.33	18.25	18.12	18.56	194.55
Soybean	59.07	20.97	19.68	19.79	19.55	20.51	210.23
Spambase	1927.01	568.47	520.98	522.13	517.07	836.79	5800.57
SPECT Heart	54.23	19.50	18.30	18.36	18.18	18.38	195.27
SteelPF	226.99	78.74	71.07	78.20	75.09	89.41	802.82
ThoracicSurgery	60.05	21.75	20.54	20.75	20.69	20.89	225.08
Tic-tac-toe	77.02	27.81	25.48	24.57	25.03	37.08	279.25
Zoo	54.66	19.76	18.24	18.47	18.38	18.74	200.35

identify negative examples. Calculate the following:

$$Sensitivity = \frac{TN}{TN + FP} \tag{17}$$

$$specificity = \frac{TP}{TP + FN} \tag{18}$$

True positives (TP): the number of positive examples that are correctly divided. False positives (FP): the number of positive examples that are incorrectly divided. False negatives (FN): the number of negative examples that are

incorrectly divided. True negatives (TN): the number of negative examples that are correctly divided.

Wilcoxon rank sum test: It's a nonparametric test. It is a test in which the sample rank is substituted for the sample value. This statistical test returns a p-value parameter to verify that the difference between the two algorithms is significant [62].

D. COMPARISON OF SHO, SHOSA-1 AND SHOSA-2

Table 6 summarizes the results of the proposed method in terms of classification accuracy and selection size. It is

TABLE 20. Comparison between the proposed SHOSA-1 and other algorithms based on standard deviation of the running time.

Dataset	ACO	CS	FPA	PSO	SCA	SSA	SHOSA-1
BreastCW	0.9351	0.8062	1.2554	0.6015	0.4676	1.0041	11.3585
Congressional	1.7352	0.4216	0.2096	0.3145	0.2568	0.3429	4.3356
ConnectBench	0.8789	0.2008	0.2989	0.5539	0.4134	0.3061	6.8332
Dermatology	1.8847	0.2515	0.3202	0.8944	0.4278	0.3917	1.3252
Drug_consumption	7.3944	1.9244	1.0621	5.2755	1.3033	6.7062	10.4339
Glass	2.2057	1.4304	0.3161	0.5297	0.3977	0.4258	1.4404
Heart	0.4953	0.2361	0.2445	0.3726	0.3440	0.4047	1.3305
Hepatitis	0.1498	0.1837	0.4390	0.4874	0.3598	0.4003	3.4250
Horse-colic	1.3958	0.1264	0.2957	0.4042	0.3506	0.2779	0.9993
ILPD	0.4171	1.0995	0.2581	0.3779	0.6274	1.1047	0.8744
Ionosphere	1.5218	0.1563	0.2949	0.3448	0.3175	0.4398	0.6529
Primary-tumor	1.3131	0.5091	0.2968	0.6127	0.5059	0.6010	1.7174
Seeds	0.4488	0.2413	0.3466	0.4312	0.3720	0.2914	0.4944
Soybean	1.0506	0.2603	0.3889	0.2809	0.3397	0.4061	0.6123
Spambase	28.9649	11.1962	9.0196	45.7244	10.4247	11.0041	77.1605
SPECT Heart	0.6028	0.3321	0.4532	0.3262	0.4691	0.3377	1.3741
SteelPF	2.4433	4.8566	1.6942	4.4099	0.9633	0.5573	4.8990
ThoracicSurgery	0.6543	0.9920	0.4479	0.2874	0.3672	0.5573	2.4088
Tic-tac-toe	1.4294	0.9532	0.8456	1.3190	0.5089	0.3566	2.2074
Zoo	0.8510	0.5300	0.3534	0.4508	0.3379	0.5038	0.8205
Average	2.8386	1.3354	0.9420	3.1999	0.97773	1.3210	6.7352

TABLE 21. Three large data sets were compared with the running results of the truncated version.

	Drug_consumption	Drug_consumption (truncated)	Spambase	Spambase (truncated)	SteelPF	SteelPF (truncated)
No. of Attributes	32	32	58	58	27	27
No. of Objects	1888	500	4601	200	1941	500
Accuracy	0.7433	0.7551	0.9312	0.9288	1	1
Attributes	11.5	11.9	29.2	27.6	6	6
Best Fitness	0.2208	0.2171	0.0695	0.7005	0.0018	0.0018
Worst Fitness	0.2776	0.2642	0.0786	0.7926	0.0018	0.0018
Mean Fitness	0.2580	0.2463	0.0733	0.07533	0.0018	0.0018
STD(fitness)	0.0085	0.0077	0.0022	0.0035	0	0
Time	795.02	221.34	5800.57	247.51	802.82	206.76

obvious that SHOSA-1 can be classified more accurately in more datasets (eleven out of twenty datasets). In addition, the proposed SHOSA-1 and SHOSA-2 algorithms are more accurate than the original algorithm SHO in most datasets. In selecting feature number size, the SHOSA-1 method can select fewer features on more datasets (thirteen out of twenty datasets). The method SHOSA-1 is in competition with the native algorithm SHO. In terms of classification accuracy and the number of selected features, SHOSA-1 shows the best effect on the average of 20 datasets. This proves that the proposed method is effective. In addition, the accuracy of the three proposed algorithms on the STEELPF dataset reach 1, which shows the excellent performance of the proposed algorithms.

Method SHOSA-1 also performs better in fitness (best, worst, mean and standard deviation) than the native algorithm SHO and method SHOSA-2 in more datasets which are shown in Tables 7-8. It is shown as follows: SHOSA-1 performs well in 15 datasets and 13 datasets in the best and

worst fitness; SHOSA-1 performs well in 12 datasets in the mean fitness; SHOSA-1 performs well in 13 datasets in the standard deviation measure. The method SHOSA-2 competes with the native algorithm SHO. It is obvious that method SHOSA-1 has better effect and higher stability than method SHOSA-2 and the native algorithm SHO.

Table 9 shows the average running time of the different methods. The native algorithm SHO and the method SHOSA-2 run at approximately the same time, but the method SHOSA-1 takes more time to run. This is due to the SA algorithm and fitness function run times.

E. COMPARISON WITH THE DIFFERENT OPTIMIZATION ALGORITHM

In the last section, SHOSA-1 method is superior to other proposed methods in classification accuracy and average selection size. In this section, the best performing method, the SHOSA-1 method, is compared with various approaches to solving the feature selection problem.

TABLE 22. P-values of the Wilcoxon test of the proposed SHOSA-1 vs. other algorithms (the data of $p > 0.05$ has been bolded, “+” indicates significant difference).

Test Dataset	ACO		CS		FPA		PSO		SCA		SSA		SHO		SHOSA-2	
	p	h	p	h	p	h	p	h	p	h	p	h	p	h	p	h
BreastCW	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1
Congressional	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1
ConnectBench	<0.05	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	0.2238	0
Dermatology	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1
Drug_consumption	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1
Glass	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1
Heart	<0.05+	1	<0.05	1	0.1745	0	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	0.3346	0
Hepatitis	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1
Horse-colic	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1
ILPD	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05+	1	<0.05+	1	0.1438	0
Ionosphere	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1
Primary-tumor	0.1952	0	<0.05+	1	0.1552	0	<0.05+	1	0.1355	0	0.0917	0	0.1943	0	0.1978	0
Seeds	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1
Soybean	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1
Spambase	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1
SPECT Heart	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1
SteelPF	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	0.1233	0	0.1452	0
ThoracicSurgery	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	<0.05+	1
Tic-tac-toe	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	<0.05	1	<0.05+	1	0.2377	0
Zoo	<0.05+	1	<0.05	1	<0.05	1	<0.05+	1	<0.05+	1	0.1586	0	0.1484	0	0.1826	0

Table 10 shows the classification accuracy of ACO, CS, FPA, PSO, SCA, SSA and SHOSA-1. According to the results in Table 7, the classification accuracy of SHOSA-1 is far superior to all other algorithms on almost all datasets. On the dataset Glass, FPA is slightly better than SHOSA-1. Both SSA and SHOSA-1 achieve the highest accuracy on the dataset SteelPF.

Table 11 shows the average value of selected features of SHOSA-1 and other algorithms. On fourteen datasets, SHOSA-1 solutions can choose fewer features. In addition, ACO algorithm can choose fewer features on five datasets.

Table 12-15 shows the fitness(best, worst, mean and standard deviation) obtained by running different algorithms. SHOSA-1 have better and more stable fitness than other algorithms. In addition, the fitness of PSO is better than other algorithms. FPA algorithm have relatively stable performance in standard deviation fitness measure.

Table 16 shows the convergence curves of different algorithms. From the results, we can see that SHOSA-1 has good convergence results in most data sets, but not the fastest convergence.

Sensitivity and specificity are statistical measures, which are suitable for binary classification test. Therefore, sensitivity and specificity were analyzed only on 9 binary classification datasets. Table 17 and Table 18 show the sensitivity and specificity of SHOSA-1 and the comparison algorithm. Obviously, SHOSA-1 showed the highest sensitivity and the highest specificity in the eight datasets.

Table 19 shows the average running time of SHOSA-1 and other algorithms. The algorithm with the shortest time is SCA, followed by FPA. SHOSA-1 takes more time in computing speed than other algorithms. However, the proposed SHOSA-1 has demonstrated highly competitive classification

accuracy and the number of selected features. Therefore, SHOSA-1 performs well overall compared to other algorithms.

Table 20 shows the standard deviation of the running time of the proposed SHOSA-1 algorithm and other algorithms. The results show that the FPA takes the most stable time to run in seven datasets and the CS takes the most stable time to run in six datasets. SHOSA-1 algorithm runs long time and unstable on Drug_consumption, Spambase, and SteelPF datasets, so SHOSA-1 is not recommended for large datasets.

Table 21 shows the three large data sets were compared with the running results of the truncated version. The results show that the truncated data set can still maintain better accuracy, selected feature number and fitness. This proves the good performance of SHOSA-1.

Table 22 is Wilcoxon test of the proposed SHOSA-1 with other algorithms that shows the results of the Wilcoxon’s rank sum test evaluated by P values. When P value is less than 0.05, there is a significant difference between the two algorithms in processing the same dataset. The results show that the proposed algorithm SHOSA-1 is significantly different from the comparison algorithm.

To sum up, the experiments in this section show the excellent ability of SHOSA-1 in terms of accuracy and number of selected features. This shows that SHOSA-1 algorithm has better search ability than other algorithms in global search and local search.

VI. CONCLUSIONS, LIMITATIONS AND FUTURE WORK

In this paper, a hybrid SHO and SA is proposed to solve the problem of feature selection in packaging mode. Embedding SA in SHO and adding SA after SHO are two hybrid methods for machine learning feature selection. A fitness function

is used to evaluate the feature selection problem, including the classification accuracy and the number of features. The performance of these algorithms are evaluated on 20 standard datasets in UCI repository. The evaluated performance includes classification accuracy, average number of features, best and worst and mean fitness measure, standard deviation measure and average computational time. The proposed SHOSA-1 algorithm is applied to these datasets and compared with other algorithms including ACO, FPA, PSO, SSA, CS, SCA. The results show that the average accuracy of the SHOSA-1 algorithm on these 20 data sets is increased by 9.82%, the average reduction of 4.64 features and the average fitness improvement of 9.22%. The SHOSA-1 is significantly different from other algorithms by Wilcoxon's rank sum test. SHOSA-1 is not recommended for large datasets due to its long running time.

For future studies, SHOSA-1 can be applied to various other public datasets, real-world problems and it can also be employed with more classifiers like artificial neural networks and support vector machines to extend the current approach. In addition, SHO algorithm should be combined with other meta-heuristic algorithms. For example, hybrid SHO algorithm and GA algorithm are used to solve optimization problems. SHOSA-1 algorithm also can be extended to higher dimensions and be applied in practical fields such as medicine in future work. SHOSA-1 will help optimize the feature selection process of high-dimensional biomedical data, better mine the function of biological data set in the field of disease diagnosis, and improve the efficiency of disease diagnosis.

REFERENCES

- [1] S. F. Crone, S. Lessmann, and R. Stahlbock, "The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing," *Eur. J. Oper. Res.*, vol. 173, no. 3, pp. 781–800, Sep. 2006.
- [2] J. Jose, "A survey on feature selection techniques," *Int. J. Eng. Res. Gen. Sci.*, vol. 2, no. 5, pp. 44–46, Aug. 2014.
- [3] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.
- [4] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, nos. 1–4, pp. 131–156, 1997.
- [5] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.
- [6] 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.
- [7] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.
- [8] C. Lai, M. J. T. Reinders, and L. Wessels, "Random subspace method for multivariate feature selection," *Pattern Recognit. Lett.*, vol. 27, no. 10, pp. 1067–1076, 2006.
- [9] E. G. Talbi, *Metaheuristics: From Design to Implementation*, vol. 42, no. 4. Hoboken, NJ, USA: Wiley, Nov. 2012, pp. 414–415.
- [10] X. Wang, J. Yang, and X. Teng, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognit. Lett.*, vol. 28, no. 4, pp. 459–471, 2007.
- [11] S. Oreski and G. Oreski, "Genetic algorithm-based heuristic for feature selection in credit risk assessment," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2052–2064, 2014.
- [12] S. A.-F. Sayed, E. Nabil, and A. Badr, "A binary clonal flower pollination algorithm for feature selection," *Pattern Recognit. Lett.*, vol. 77, pp. 21–27, Jul. 2016.
- [13] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary ant lion approaches for feature selection," *Neurocomputing*, vol. 213, pp. 54–65, Nov. 2016.
- [14] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," *Expert Syst. Appl.*, vol. 116, pp. 147–160, Feb. 2019.
- [15] D. Rodrigues, L. A. M. Pereira, R. Y. M. Nakamura, K. A. P. Costa, X.-S. Yang, A. N. Souza, and J. P. Papa, "A wrapper approach for feature selection based on bat algorithm and optimum-path forest," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2250–2258, Apr. 2014.
- [16] J. Lee and D.-W. Kim, "Memetic feature selection algorithm for multi-label classification," *Inf. Sci.*, vol. 293, pp. 80–96, Feb. 2015.
- [17] S. Tabakhi and P. Moradi, "Relevance–redundancy feature selection based on ant colony optimization," *Pattern Recognit.*, vol. 48, no. 9, pp. 2798–2811, Sep. 2015.
- [18] 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.
- [19] H. Faris, M. M. Mafarja, A. A. Heidari, I. Aljarah, A. M. Al-Zoubi, S. Mirjalili, and H. Fujita, "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems," *Knowl.-Based Syst.*, vol. 154, pp. 43–67, Aug. 2018.
- [20] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018.
- [21] Q. Tu, X. Chen, and X. Liu, "Multi-strategy ensemble grey wolf optimizer and its application to feature selection," *Appl. Soft Comput.*, vol. 76, pp. 16–30, Mar. 2019.
- [22] M. Mafarja, I. Aljarah, A. A. Heidari, H. Faris, P. Fournier-Viger, X. Li, and S. Mirjalili, "Binary dragonfly optimization for feature selection using time-varying transfer functions," *Knowl.-Based Syst.*, vol. 161, pp. 185–204, Dec. 2018.
- [23] H. Garg, "An efficient biogeography based optimization algorithm for solving reliability optimization problems," *Swarm Evol. Comput.*, vol. 24, pp. 1–10, Oct. 2015.
- [24] E. G. Talbi, "Metaheuristics: From design to implementation," *J. Art. Rev.*, vol. 42, no. 4, pp. 414–415, Nov. 2012.
- [25] E.-G. Talbi, "A taxonomy of hybrid metaheuristics," *J. Heuristics*, vol. 8, no. 5, pp. 541–564, 2002.
- [26] G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications," *Adv. Eng. Softw.*, vol. 114, pp. 48–70, Dec. 2017.
- [27] G. Dhiman and V. Kumar, "Multi-objective spotted hyena optimizer: A Multi-objective optimization algorithm for engineering problems," *Knowl. Based Syst.*, vol. 150, pp. 175–197, Jun. 2018.
- [28] G. Dhiman and A. Kaur, "Spotted hyena optimizer for solving engineering design problems," in *Proc. Int. Conf. Mach. Learn. Data Sci.*, Dec. 2017, pp. 114–119.
- [29] G. Dhiman and A. Kaur, "Optimizing the design of airfoil and optical buffer problems using spotted hyena optimizer," *Designs*, vol. 2, no. 3, pp. 28–43, Aug. 2018.
- [30] G. Dhiman, S. Guo, and S. Kaur, "ED-SHO: A framework for solving nonlinear economic load power dispatch problem using spotted hyena optimizer," *Mod. Phys. Lett. A*, vol. 33, no. 40, Dec. 2018, Art. no. 1850239.
- [31] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Sci.*, vol. 220, no. 4598, pp. 671–680, May 1983.
- [32] Y. Li, H. Guo, L. Wang, and J. Fu, "A hybrid genetic-simulated annealing algorithm for the location-inventory-routing problem considering returns under e-supply chain environment," *Sci. World J.*, vol. 2013, no. 10, Nov. 2013, Art. no. 125893.
- [33] 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.
- [34] O. C. Martin and S. W. Otto, "Combining simulated annealing with local search heuristics," *Ann. Oper. Res.*, vol. 63, no. 1, pp. 57–75, Feb. 1996.
- [35] I.-S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1424–1437, Nov. 2004.
- [36] P. Vasant, "Hybrid simulated annealing and genetic algorithms for industrial production management problems," *Int. J. Comput. Methods*, vol. 7, no. 2, pp. 279–297, 2010.
- [37] S. Publicover, C. V. Harper, and C. Barratt, "Hybrid simulated annealing and genetic algorithm for optimizing arterial signal timings under over-saturated traffic conditions," *J. Adv. Transp.*, vol. 49, no. 1, pp. 153–170, 2015.

- [38] L. Junghans and N. Darde, "Hybrid single objective genetic algorithm coupled with the simulated annealing optimization method for building optimization," *Energy Buildings*, vol. 86, pp. 651–662, Jan. 2015.
- [39] H. Garg, "A hybrid GA-GSA algorithm for optimizing the performance of an industrial system by utilizing uncertain data," in *Handbook of Research on Artificial Intelligence Techniques and Algorithms*. 2015.
- [40] H. Garg, "A hybrid GSA-GA algorithm for constrained optimization problems," *Inf. Sci.*, vol. 478, pp. 499–523, Apr. 2019.
- [41] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Appl. Math. Comput.*, vol. 274, pp. 292–305, Feb. 2016.
- [42] J. Wu, Z. Lu, and L. Jin, "A novel hybrid genetic algorithm and Simulated Annealing for feature selection and kernel optimization in support vector regression," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, Aug. 2012, pp. 401–406.
- [43] M. E. Basiri and S. Nemati, "A novel hybrid ACO-GA algorithm for text feature selection," in *Proc. 11th Conf. Congr. Evol. Comput.*, May 2009, pp. 2561–2568.
- [44] J. B. Jona and N. Nagaveni, "Ant-cuckoo colony optimization for feature selection in digital mammogram," *Pakistan J. Biol. Sci.*, vol. 17, no. 2, pp. 266–271, 2014.
- [45] J. E. Smith and K. E. Holekamp, "Spotted hyenas," in *Encyclopedia of Animal Behavior*. 2019, pp. 190–208.
- [46] A. Ilany, A. S. Booms, and K. E. Holekamp, "Topological effects of network structure on long-term social network dynamics in a wild mammal," *Ecol. Lett.*, vol. 18, no. 7, pp. 687–695, Jul. 2015.
- [47] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statist.*, vol. 46, no. 3, pp. 175–185, 1992.
- [48] C. Blake. (1998). *UCI Repository of Machine Learning Databases*. [Online]. Available: <http://www.ics.uci.edu/?mlearn/MLRepository.html>
- [49] Y. Wu, M. Gong, W. Ma, and S. Wang, "High-order graph matching based on ant colony optimization," *Neurocomputing*, vol. 328, pp. 97–104, Feb. 2019.
- [50] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.
- [51] X. S. Yang, "Flower pollination algorithm for global optimization," in *Proc. UCNC*, vol. 7445, Dec. 2012, pp. 240–249.
- [52] L. Shen, C. Fan, and X. Huang, "Multi-level image thresholding using modified flower pollination algorithm," *IEEE Access*, vol. 6, pp. 30508–30519, May 2018.
- [53] B. Akay, "A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding," *Appl. Soft Comput.*, vol. 13, no. 6, pp. 3066–3091, 2013.
- [54] Y. Liu, C. Mu, W. Kou, and J. Liu, "Modified particle swarm optimization-based multilevel thresholding for image segmentation," *Soft Comput.*, vol. 19, no. 5, pp. 1311–1327, 2015.
- [55] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
- [56] A. S. Joshi, O. Kulkarni, G. M. Kakandikar, and V. M. Nandedkar, "Cuckoo search optimization—A review," *Mater. Today, Proc.*, vol. 4, no. 8, pp. 7262–7269, 2017.
- [57] H. Ghimatgar, K. Kazemi, M. S. Helfroush, and A. Aarabi, "An improved feature selection algorithm based on graph clustering and ant colony optimization," *Knowl. Based Syst.*, vol. 159, pp. 270–285, Nov. 2018.
- [58] L. A. M. Pereira et al., "A binary cuckoo search and its application for feature selection," in *Cuckoo Search and Firefly Algorithm*. 2014, pp. 141–154.
- [59] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, and S. Wang, "An improved particle swarm optimization for feature selection," *J. Bionic Eng.*, vol. 8, no. 2, pp. 191–200, Jun. 2011.
- [60] A. I. Hafez, H. M. Zawbaa, E. Emary, and A. E. Hassanien, "Sine cosine optimization algorithm for feature selection," in *Proc. Int. Symp. Innov. Intell. Syst. Appl.*, Aug. 2016, pp. 1–5.
- [61] A. E. Hegazy, M. A. Makhlof, and G. S. El-Tawel, "Improved salp swarm algorithm for feature selection," *J. King Saud Univ. Comput. Inf. Sci.*, to be published. doi: [10.1016/j.jksuci.2018.06.003](https://doi.org/10.1016/j.jksuci.2018.06.003).
- [62] S. García, J. Derrac, J. Luengo, C. J. Carmona, and F. Herrera, "Evolutionary selection of hyperrectangles in nested generalized exemplar learning," *Appl. Soft Comput.*, vol. 11, no. 3, pp. 3032–3045, Apr. 2011.



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.



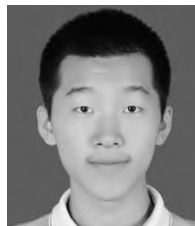
JINDUO LI was born in Jilin, China, in 1996. He is currently pursuing the M.S. degree in control engineering with Northeast Forestry University, China. His research interests include feature selection and swarm intelligence algorithm.



WENLONG SONG 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.



XIAOXU PENG was born in Harbin, China, in 1995. He is currently pursuing the M.S. degree in control theory and control engineering with Northeast Forestry University, China. His research interests include image segmentation and swarm intelligence algorithm.



CHUNBO LANG was born in Shenyang, China, in 1998. He is currently pursuing the B.S. degree in automation with Northeast Forestry University, China. His research interests include image segmentation and swarm intelligence algorithm.



YAO LI was born in Yichun, China, in 1997. She is currently pursuing the B.S. degree in automation with Northeast Forestry University, China. Her research interests include image segmentation and swarm intelligence optimization.

...