Journal of King Saud University – Computer and Information Sciences xxx (2018) xxx–xxx

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Improved salp swarm algorithm for feature selection

Ah. E. Hegazy [a,*], M. A. Makhlouf [a], Gh. S. El-Tawel [b]

[a] Dept. of Information System, Faculty of Computers & Informatics, Suez Canal University, Ismailia, Egypt
[b] Dept. of Computer Science, Faculty of Computers & Informatics, Suez Canal University, Ismailia, Egypt

ARTICLE INFO

Article history:
Received 10 April 2018
Revised 18 May 2018
Accepted 8 June 2018
Available online xxxx

Keywords:
Feature selection
Salp swarm algorithm
Bio-inspired optimization
K-Nearest Neighbor
Classification

ABSTRACT

Salp swarm algorithm (SSA) is a recently created bio-inspired optimization algorithm presented in 2017 which is based on the swarming mechanism of salps. This paper tries to improve the structure of basic SSA to enhance solution accuracy, reliability and convergence speed. A new control parameter, inertia weight, is added to adjust the present best solution. The new method known as improved salp swarm algorithm (ISSA) is tested in feature selection task. The ISSA algorithm is consolidated with the K-nearest neighbor classifier for feature selection in which twenty-three UCI datasets are utilized to assess the performance of ISSA algorithm. The ISSA is compared with the basic SSA and four other swarm methods. The results demonstrated that the proposed method produced superior results than the other optimizers in terms of classification accuracy and feature reduction.

1. Introduction

Classification is one of the most common tasks in data mining (Yin and Gai, 2015). Because of the vast growth in the data in the world, a pre-processing procedure like feature selection becomes a challenging and fundamental task in data mining applications (Jensen, 2005). Feature selection is the process of removing the repetitive and unrelated features from a dataset to increase the performance of the learning algorithm (Liu and Motoda, 1998). The final set of features can be assessed using two distinct methods: wrapper and filter. A filter-based procedure involves the learning algorithm (e.g.classification algorithm) in the selection stage (Kohavi and John, 1997), while in the wrapper-based procedure, the feature set will be examined depending on the data properties instead of on the classifier as in a wrapper mode (Mafarja and Abdullah, 2014; Mafarja and Abdullah, 2015; Mafarja and Abdullah, 2011). In feature selection, finding the optimal subset is a decisive issue. Exhaustive search can produce all possible subsets by examining all the entire set of features (Zhong et al., 2001). This approach is impractical for the large datasets and has an extremely high computational cost because if a dataset holds M features, then there are 2^M subsets of features (Guyon and Elisseeff, 2003; Yin et al., 2016). In the previous two decades, metaheuristics have demonstrated their productivity and efficiency in solving challenging and large-scale problems in machine learning, engineering design and data mining applications (Xue et al., 2016). Nature-inspired algorithms are mostly metaheuristics inspired by nature (Talbi, 2009). There are three basic categories of these algorithms: physics-based (e.g. simulated annealing), evolutionary-based (e.g. genetic algorithms and evolutionary algorithms) and swarm-based (e.g. ant colony and bees colony) (Talbi, 2009). When using these algorithms, there are two conflicting criteria are common:intensification (exploitation of the best solutions found) and diversification (exploration of the search space) (Talbi, 2009).

As specified above, swarm techniques imitative the searching mechanism of natural animals like bat, cuckoo, ant, moth, etc. (Valdez, 2015). Lately, swarm algorithms are suggested and proved a valuable performance for feature selection tasks. This motivated us to examine the merits of the lately suggested salp swarm algorithm (SSA) (Mirjalili et al., 2017) in the domain of feature selection. SSA is an evolutionary algorithm which imitates the swarming mechanism of the salps in nature.

Previous research has demonstrated that swarm algorithms have tackled feature selection problems. In this paper, an improved version of SSA is used as a wrapper feature selection method to identify the best feature subsets. The main contributions of our work include:

[*] Corresponding author.
E-mail address: ahmedhegazy281@yahoo.com (A.E. Hegazy).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2018.06.003
1319-1578/© 2018 Production and hosting by Elsevier B.V. on behalf of King Saud University.

- We propose a binary version of SSA to solve binary problems like feature selection problems. The original SSA is developed to deal with continuous problems.
- We introduce an improved version of SSA by incorporating an inerita weight parameter to enhance both exploration and exploitation capabilities of this algorithm.
- The present work is designed to be the first to apply ISSA for feature selection problems.
- Previous research of SSA for feature selection used datasets with a small number of features (57 features) (Ahmed and Mafarja, 2018). In this paper, the maximum number of features is 856.
- The fitness function of SSA previous research for feature selection aimed to maximize the classification accuracy only. In this paper, the fitness function balances between the number of selected features and the classification accuracy.

The present research is therefore intended to make contributions to the literature on goals, motivation, and variety.

2. Related work

Feature selection is playing a significant role in any classification task. Lately, swarm algorithms are suggested and proved a valuable performance for feature selection tasks. Table 1 presents some of the earlier research algorithms for the feature selection. For more reading about metaheuristics for feature selection, interested readers are referred to see the survey papers in (Boussaïd et al., 2013; Chandrashekar and Sahin, 2014; Abdullah et al., 2017; Archana and Thanabal, 2016).

Table 1
Review of previous studies on the swarm algorithms for feature selection.

FS Method	Remarks
IBPSO (Chuang et al., 2008)	Introduced an Improved binary particle swarm optimization for feature selection in data classification. IBPSO has achieved superior results than the other optimizers
BCSA (Pereira et al., 2014)	Introduced a binary version of the cuckoo search algorithm for the feature selection task. BCS has achieved reasonable results than the other optimizers
GWO (Emary et al., 2016)	Proposed a binary version of the grey wolf optimization (GWO) for feature selection domain. Results confirm the ability of GWO to find the best feature subset
ALO (Zawbaa et al., 2015)	Introduced a feature selection method using a binary ant lion optimizer (BALO). Results confirm the ability of BALO to find the best subset
MOGA (Saroj and Jyoti, 2014)	Presented a feature selection approach using Multi-Objective Genetic Algorithm (MOGA). MOGA has achieved reasonable classification accuracy than the other optimizers
WOA (Sharawi et al., 2017)	Introduced a binary version of the whale optimization algorithm (WOA) for feature selection task. The results confirm the advantage of the WOA compared to the other optimizers
CALO (Zawbaa et al., 2016)	Proposed a chaotic version of the antlion optimizer (CALO) for feature selection domain. The performance of CALO is better than that of the other methods over the test data considered
AHSGS (Dash, 2018)	Introduced a feature selection method using harmony search and Pareto optimization approaches. The performance of AHSGS is better than that of the other methods
FAEMODE (Nayak et al., 2017)	Introduced a feature selection method using Elitism based Multi-objective Differential Evolution algorithm. The results confirm the advantage of the FAEMODE compared to the other optimizers

3. Methods

3.1. Salp swarm algorithm (SSA)

SSA is one of the random population-based algorithms suggested by Mirjalili et al. (2017) in 2017. SSA simulates the swarming mechanism of salps when foraging in oceans. In heavy oceans, salps usually shape a swarm known as salp chain. In SSA algorithm, the leader is the salp at the front of chain and whatever remains of salps are called followers. Like to other swarm-based techniques, the position of salps is defined in an s-dimensional search space, where s is the number of variables of a given problem. Therefore, the position of all salps is stored in a two-dimensional matrix called z. It is also assumed that there is a food source called P in the search space as the swarm's target. The mathematical model for SSA is given as follows: The leader salp can change position by using the next equation:

$$z_n^1 = \begin{cases} P_n + r_1((u_n - l_n)r_2 + l_n) & r_3 \geqslant 0 \\ P_n - r_1((u_n - l_n)r_2 + l_n) & r_3 < 0 \end{cases} \tag{1}$$

where the meanings of all symbols are shown in Table 2.

$$r_1 = 2e^{-}\left(-\frac{4a}{A}\right)^2 \tag{2}$$

The coefficient r_1 is the essential parameter in SSA because it provides a balance between exploration and exploitation capabilities.

To change the position of the followers, the next equations are utilized:

$$z_n^m = \frac{1}{2}ce^2 + v_0 e \tag{3}$$

where $m \geqslant 2, c = \frac{v_{final}}{v_0}$where $v = \frac{z - z_0}{e}$. Because the time in optimization is iteration, the conflict between iterations is equal to 1, and considering $v_0 = 0$, this equation can be declared as follows:

$$z_n^m = \frac{1}{2}\left(z_n^m + z_n^{m-1}\right) \tag{4}$$

A detail stepwise description of this algorithm is as follows.

1. Initialize the parameters of SSA such as number of salps (S), number of iterations (A), optimal salp position (Z^*) and best fitness value (f(Z^*)).
2. Initialize a population of S salp's positions randomly..
3. Evaluate the fitness of every salp.
4. Set number of iteration (a) to 0.
5. Update r1 by Eq. (2).
6. For each salp,
   (a) If $m == 1$, update the position of the leading salp by Eq. (1).
   (b) Otherwise, update the position of the follower salp by Eq. (4).
   (c) Evaluate the fitness of every salp.
   (d) Update Z^* if there is a superior solution.

Table 2
The meanings of all symbols.

Symbol	Meaning
z_n^1	leader position in nth dimension
P_n	food source position in nth dimension
u_n	upper bound of nth dimension
l_n	lower bound of nth dimension
r_1, r_2, and r_3	random variables uniformly produced in the interval of [0,1].
a	current iteration
A	maximum number of iterations
z_n^m	position of $m$ith follower salp in nth dimension
e	time
v_0	the initial speed

Please cite this article in press as: Hegazy, A.E., et al. Improved salp swarm algorithm for feature selection. Journal of King Saud University – Computer and Information Sciences (2018), https://doi.org/10.1016/j.jksuci.2018.06.003

7. Increment a to 1.
8. Repeat Steps 5 to 7 until a = A is satisfied.
9. Return the best solution Z^* and its fitness value f(Z^*).

3.2. Improved salp swarm algorithm (ISSA)

As specified by Mirjalili et al. (2017), the position of salps is updated by Eqs. (1) and (4). In SSA, the updated solution is mainly depended on the current best solution. Like PSO algorithm, an inertia weight $\omega \in$[0,1] is introduced into SSA to get the improved salp swarm algorithm (ISSA). The new parameter in the ISSA algorithm accelerates the convergence speed during the search. It also makes a balance between exploitation and exploration capabilities to first escape an enormous number of local solutions in feature selection tasks and second to achieve a precise appreciation of the optimal solution. The improved algorithm is stated by the next equations:

$$z_n^1 = \begin{cases} \omega P_n + r_1((u_n - l_n)r_2 + l_n) & r_3 \geqslant 0 \\ \omega P_n - r_1((u_n - l_n)r_2 + l_n) & r_3 < 0 \end{cases} \quad (5)$$

$$z_n^m = \frac{1}{2}(z_n^m + \omega z_n^{m-1}) \quad (6)$$

4. The Proposed approach for feature selection

4.1. Solution representation

In feature selection problems, all solutions are limited to the binary $[0, 1]$ values. For the SSA algorithm to be used with the feature selection problem, a binary version should be developed. In this work, a solution is defined as a one-dimensional vector, where the length of the vector is based on the number of features in the original dataset. Each cell in the vector has value “1” or “0”. Value “1” indicates that the corresponding feature is chosen; otherwise, the value is defined as “0”. Eq. (7) is utilized to map continuous values into binary ones.

$$Z_{mn} = \begin{cases} 1 & \text{if } X_{mn} > 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where $Z_{mn}$is the discrete form of solution vector X and X_{mn} is the continuous position of the search agent m at dimension n.

Fig. 1 presents a sample of feature subset solution for a dataset of 7 attributes. In Fig. 1, the classification task uses only the 4 chosen features (1, 4, 6 and 7) which have a value of ”1” and discards the rest of features.

4.2. Fitness function

Feature selection can be modeled as a multi-objective optimization problem where two conflicting objectives must be accomplished;

1	0	0	1	0	1	1

Fig. 1. A sample of feature subset solution.

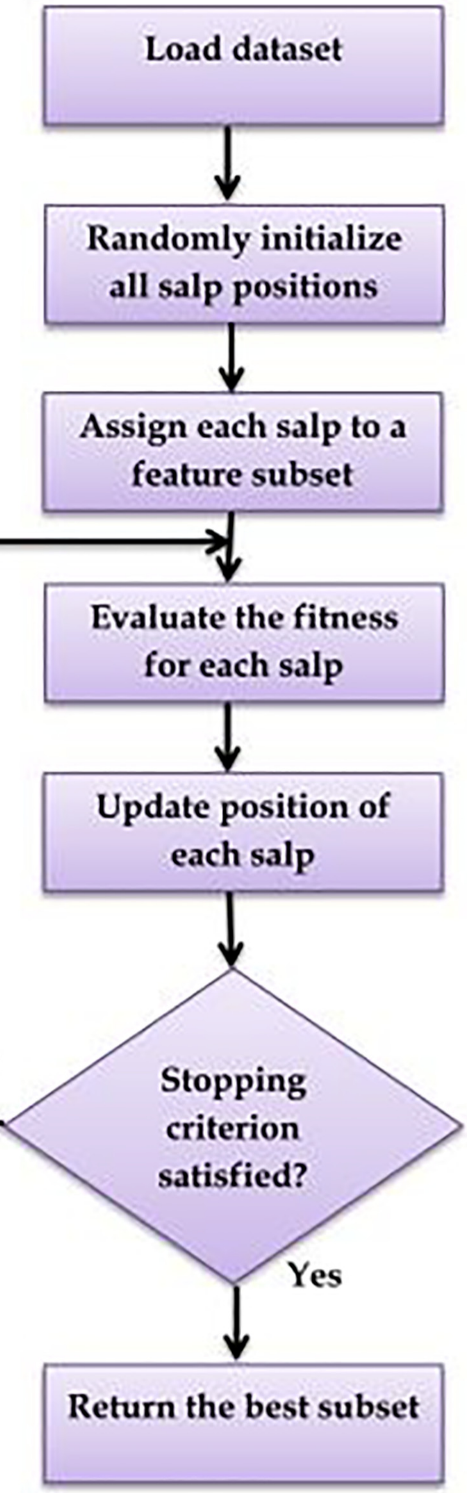
Load dataset
Randomly initialize all salp positions
Assign each salp to a feature subset
Evaluate the fitness for each salp
Update position of each salp
Stopping criterion satisfied?
No
Yes
Return the best subset

Fig. 3. ISSA feature selection flowchart.

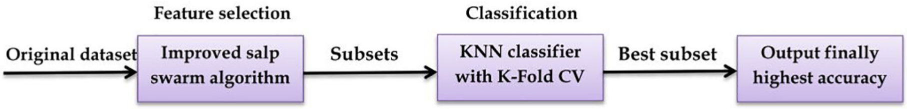
Feature selection
Classification
Original dataset
Improved salp swarm algorithm
Subsets
KNN classifier with K-Fold CV
Best subset
Output finally highest accuracy

Fig. 2. Two stages of feature selection.

Please cite this article in press as: Hegazy, A.E., et al. Improved salp swarm algorithm for feature selection. Journal of King Saud University – Computer and Information Sciences (2018), https://doi.org/10.1016/j.jksuci.2018.06.003

minimum number of selected features and maximum classification accuracy. The optimal solution is the solution which has a small number of selected features and high classification accuracy. KNN classification accuracy is used as a fitness function to assess the performance of all search agents. In order to balance between the number of selected features in each solution (minimum) and the classification accuracy (maximum), the fitness function in Eq. (8) is utilized in both ISSA and all algorithms to assess search agents.

$$Fitness = \rho Err(D) + \varphi \frac{|F|}{|T|} \tag{8}$$

Where Err(D) is the classifier error rate of the identified subset, ρ and φ are constants to control the classification accuracy and the feature reduction, $|F|$ is the size of identified feature subset, $|T|$ is the total number of features. ρ in [0,1] and $\varphi = (1 - \rho)$, In this work $\rho = 0.9$.

4.3. System architecture

We used ISSA as a wrapper feature selection mode by applying the KNN classifier as a fitness function. Note the proposed algorithm can be utilized with any other classifier. The K-nearest neighbor (KNN) is a simple and very common classifier. In this work, the KNN is utilized as a classification algorithm to assess the quality of the selected subset of features. As appeared in Fig. 2, the proposed model involves two principal stages; features selection and classification. First, the model takes the original dataset as input, then ISSA algorithm is gotten to pick valuable features, then the resulted features are utilized to feed KNN classifier. Finally, the results are assessed. The flowchart of ISSA for feature selection is shown in Fig. 3.

5. Experimental results and discussions

5.1. Datasets description

To assess the effectiveness of the proposed ISSA approach, twenty-three UCI (UCI Machine Learning Repository, YYYY) benchmark datasets were utilized. The datasets are displayed in Table 3.

Table 4
Parameters of all algorithms.

Parameter	Value	Meaning
ρ	0.9	Fitness function constant
NIters	50	Maximum iterations
NAgents	10	Population size
NRuns	20	No of runs
Problem Dimension		No of features in the dataset
Search Domain		Binary vector [0, 1]
K	5	K-value in kNN
ω	0.7	Inertia factor of PSO and ISSA
c1, c2	1.4	PSO learning factors
Cross_val	0.9	GA Crossover Fraction
Mut_val	0.1	GA Mutation Fraction

A wrapper approach-based on the KNN classifier (where K = 5 (Zawbaa et al., 2015)) is utilized to produce the optimal subset. K-fold (here, K = 10) cross-validation technique is adapted on the data set to produce training and test set. The training set is utilized to build optimal feature set and test set is utilized to estimate classification accuracy to evaluate the effectiveness of the method. The datasets that are picked hold various numbers of attributes, extending from 9 to 856. Most of them have a vast number of attributes, so they are suitable for the feature selection process. The datasets are likewise various as far as the number of classes (from 2 to 16) and instances (from 32 to 5000). In addition, we chose a set of high dimensional datasets to prove the performance of algorithms in large feature spaces. The experiments were executed in MATLAB-R2015a on a pc with windows 8, AMD A10 CPU 2.30 GHz and 4 GB memory. Table 4 presents all algorithms parameters. For a fair comparison, the best choice of the value of ω in ISSA is 0.7 as in PSO. All the parameters are defined based on domain specific-knowledge as ρ and φ parameters in the fitness function, or by trial and error such as the rest of parameters. PSO parameters values are chosen based on the common settings in Xue et al. (2016).

5.2. Performance metrics

The proposed method is compared with SSA, PSO, GA, ALO and GWO using the following metrics:

Table 3
Datasets details.

Category Type	DS ID	Dataset	# Features	# Samples	# Classes
Small <20	D1	Wine	13	178	3
	D2	Hepatitis	19	155	2
	D3	Vehicle	18	846	4
	D4	Zoo	16	101	7
	D5	Heart	13	270	2
	D6	Breastcancer	9	699	2
Medium [20–100]	D7	Ionosphere	34	351	2
	D8	Lung Cancer	56	32	3
	D9	Dermatology	34	366	6
	D10	Sonar	60	208	2
	D11	BreastEW	30	569	2
	D12	Soybean Small	35	47	4
	D13	Movementlibras	90	360	15
	D14	Parkinsons	22	195	2
	D15	Spambase	57	4601	2
	D16	Waveform	40	5000	3
Large >100	D17	Arrhythmia	279	452	16
	D18	Multiple Features	649	2000	10
	D19	Semeion	256	1593	10
	D20	Clean	166	476	2
	D21	CNAE	856	1080	9
	D22	DNA	180	2000	3
	D23	Hillvalley	101	606	2

Please cite this article in press as: Hegazy, A.E., et al. Improved salp swarm algorithm for feature selection. Journal of King Saud University – Computer and Information Sciences (2018), https://doi.org/10.1016/j.jksuci.2018.06.003

- Classification accuracy: it is gained by computing the average accuracy from 20 runs using the selected features on the test dataset.
- Fitness values: they are gotten from each approach as stated. The mean, min (best), max(worst) and std fitness values are compared.
- Average selection size: is the average number of selected features.
- Average execution time: is the run time for a given optimizer in seconds.

5.3. Numerical results and discussion

The best, mean and worst fitness values acquired by all optimizers at the end of 20 runs are presented in Tables 5–7 respectively. For each dataset, the best values are appeared in boldface. As indicated by Table 5, we notice that the performance of the proposed ISSA algorithm is superior to standard SSA and other optimizers for 13 datasets out of 23 datasets (D1, D2, D4, D6, D7, D9, D11, D13, D14, D19, D20, D22 and D23), despite PSO achieves the best results in 5 datasets, SSA in 3 datasets, ALO in 3 datasets and GA in 2 datasets. PSO is observed to be the second best performing on the 23 datasets. GWO for all cases acquires the worst results. Table 6 indicates that the ISSA can find better mean fitness values in 11 out of 23 datasets (D2, D4, D6, D7, D9, D11, D13, D14, D19, D22 and D23). SSA is observed to be the second best optimizer since it acquires the better values in 7 out of 23 datasets. As appeared in Table 7, the ISSA gets the best values in 12 out of 23 datasets (D2, D4, D6, D10, D11, D13, D14, D16, D17, D20, D22 and D23), despite SSA and ALO get the best results than ISSA for five datasets.

Table 5
Statistical Best fitness values of 20 runs for all optimizers.

DS	ISSA	SSA	PSO	GA	ALO	GWO
Wine	**0.0000**	0.0040	0.0160	**0.0000**	0.0310	0.0410
Hepatitis	**0.0684**	0.0770	0.0855	0.0862	0.1290	0.1954
Vehicle	0.2130	0.1902	0.1896	**0.1723**	0.2923	0.2535
Zoo	**0.0000**	0.0300	0.0610	0.1180	0.1772	0.1006
Heart	0.1000	0.0890	**0.0780**	0.1450	0.1513	0.1293
Breastcancer	**0.0090**	0.0340	0.0340	0.0220	0.0399	0.0395
Ionosphere	**0.0770**	0.0940	0.0940	0.1200	0.1166	0.1412
Lung cancer	0.0295	0.0294	**0.0001**	0.0312	0.1887	0.253
Dermatlogy	**0.0000**	0.0167	**0.0000**	0.0174	0.0185	0.0221
Sonar	0.1160	0.0710	**0.0480**	0.2900	0.1652	0.2132
BreastEW	**0.0160**	0.0210	0.0260	0.0470	0.0548	0.0652
SoybeanSmall	0.0207	0.0345	0.0276	0.0355	**0.0017**	0.1943
Movementlibras	**0.0840**	0.2700	0.2760	0.3110	0.3125	0.2700
Parkinsons	**0.0280**	0.0410	0.0480	0.0690	0.0932	0.1033
Spambase	0.2010	**0.1920**	**0.1920**	0.2640	0.2814	0.2794
Waveform	0.2050	**0.1870**	0.2050	0.2320	0.2589	0.2596
Arrhythmia	0.0310	**0.0290**	0.0390	0.0810	0.0363	0.0353
Multiple Features	0.0820	0.1600	0.1800	0.1430	**0.0432**	0.0441
Semeion	**0.0090**	0.0360	0.0780	0.1410	0.0954	0.0824
Clean	**0.0100**	0.0400	0.0400	0.0400	0.0144	0.0170
CNAE	0.1250	0.1740	0.1740	0.2500	**0.1119**	0.1410
DNA	**0.0900**	0.1240	0.1460	0.1350	0.1125	0.1421
Hillvalley	**0.1940**	0.2100	0.2190	0.2220	0.3987	0.3891
Average	**0.0743**	0.0936	0.0981	0.1292	0.1359	0.1483

Table 6
Statistical mean fitness values of 20 runs for all optimizers.

DS	ISSA	SSA	GA	PSO	ALO	GWO
Wine	0.0100	**0.0070**	0.0230	0.0540	0.0372	0.0430
Hepatitis	**0.1100**	0.1200	0.1600	0.1500	0.1293	0.2195
Vehicle	0.2300	**0.2100**	0.2300	**0.2100**	0.3019	0.2945
Zoo	**0.0590**	0.0790	0.1160	0.2000	0.1875	0.1027
Heart	0.1200	**0.1160**	0.1250	0.1850	0.2026	0.1301
Breastcancer	**0.0190**	0.0390	0.0380	0.0370	0.0425	0.0415
Ionosphere	**0.0970**	0.1180	0.1270	0.1760	0.1199	0.1415
Lung cancer	0.0900	**0.0700**	0.1500	0.1724	0.1902	0.2844
Dermatlogy	**0.0030**	0.0300	0.0900	0.0425	0.0214	0.0226
Sonar	0.1450	**0.1370**	0.1690	0.3190	0.1769	0.2256
BreastEW	**0.0240**	0.0320	0.0420	0.0540	0.0564	0.0654
SoybeanSmall	0.0300	0.0400	0.0600	0.0400	**0.0019**	0.2913
Movementlibras	**0.2270**	0.2900	0.2960	0.3330	0.3889	0.2784
Parkinsons	**0.0320**	0.0550	0.0840	0.0950	0.0936	0.1242
Spambase	0.2390	**0.2180**	0.2210	0.2760	0.2904	0.2840
Waveform	0.2090	**0.2080**	0.2180	0.2400	0.2681	0.2648
Arrhythmia	0.0440	0.0460	0.0600	0.0900	0.0374	**0.0363**
Multiple Features	0.1220	0.2110	0.2440	0.2650	0.0499	**0.0448**
Semeion	**0.0610**	0.0840	0.1170	0.1520	0.1012	0.0885
Clean	0.0360	0.0560	0.0640	0.0920	**0.0161**	0.0184
CNAE	0.2090	0.2510	0.2670	0.3270	**0.1662**	0.200
DNA	**0.1190**	0.1550	0.1780	0.1910	0.1371	0.1452
Hillvalley	**0.2150**	0.2250	0.2450	0.2630	0.4052	0.4088
Average	**0.1065**	0.1216	0.1445	0.1723	0.1488	0.1633

Please cite this article in press as: Hegazy, A.E., et al. Improved salp swarm algorithm for feature selection. Journal of King Saud University – Computer and Information Sciences (2018), https://doi.org/10.1016/j.jksuci.2018.06.003

Table 7
Statistical worst fitness values of 20 runs for all optimizers.

DS	ISSA	SSA	GA	PSO	ALO	GWO
Wine	0.0340	**0.0170**	0.0670	0.0680	0.0487	0.0533
Hepatitis	**0.1282**	0.1539	0.1966	0.1975	0.1358	0.3954
Vehicle	0.2478	**0.2280**	0.2478	0.2315	0.3542	0.3125
Zoo	**0.1470**	0.1820	0.1820	0.3240	0.1985	0.1587
Heart	0.1560	0.1560	0.200	0.2330	0.2125	**0.1459**
Breastcancer	**0.0260**	0.0430	0.0430	0.0470	0.0435	0.0456
Ionosphere	0.1450	0.1620	0.1710	0.2390	**0.1359**	0.1547
Lung cancer	0.1471	**0.1177**	0.2059	0.2145	0.2215	0.2954
Dermatlogy	0.0509	0.0678	0.0509	0.0687	**0.0255**	0.0298
Sonar	**0.1590**	0.2000	0.2290	0.3620	0.1894	0.2365
BreastEW	**0.0320**	0.0420	0.0530	0.0580	0.0623	0.0755
SoybeanSmall	0.0552	0.0552	0.0552	0.0553	**0.0022**	0.2956
Movementlibras	**0.2840**	0.3030	0.3120	0.3710	0.4125	0.3215
Parkinsons	**0.0410**	0.0690	0.1310	0.1240	0.1123	0.1354
Spambase	0.2640	**0.2370**	0.2670	0.2850	0.3025	0.2954
Waveform	**0.2160**	0.2170	0.2340	0.2500	0.2987	0.2789
Arrhythmia	**0.0510**	0.0570	0.0760	0.1030	0.0540	0.0514
Multiple Features	0.1430	0.2800	0.3200	0.3470	**0.0521**	0.0542
Semeion	0.1200	**0.1170**	0.1380	0.1650	0.1235	0.1264
Clean	**0.0500**	0.0800	0.0900	0.1500	0.1095	0.1099
CNAE	0.3640	0.3200	0.3200	0.5420	**0.2987**	0.3542
DNA	**0.1460**	0.2020	0.2140	0.2700	0.1478	0.1524
Hillvalley	**0.2440**	0.2540	0.2760	0.2970	0.4987	0.4879
Average	**0.1414**	0.1548	0.1774	0.2175	0.1757	0.1985

When Tables 5–7 are studied together, the proposed method ISSA gives superior results compared to the others in terms of fitness value. Note that the numbers in the last row in each Table demonstrates the fitness value averaged over all the datasets. In Tables 5–7, ISSA gains the best average result with values 0.0743, 0.1065 and 0.1414 respectively. From these Tables, one can note that the ISSA outperformed the other algorithms especially in large datasets [D17-D23].

In terms of the feature reduction, Table 8 presents the average feature size gained by all optimizers. Results presented in this Table indicate that the ISSA picks a minimal number of features in 17 out of 23 datasets. ISSA gets superior results than other optimizers especially in huge datasets [D17-D23]. Table 9 shows the classification accuracy gained by all optimizers. We can observe that the performance of ISSA is better than the other algorithms in 15 out of 23 datasets and the classification performance of ISSA is much obvious in large datasets where it achieves better results in large datasets. Moreover, the classification accuracy of ISSA is better than utilizing all features in a dataset. That is because of the ability of the ISSA to explore the feature space in a much intelligent way. This result confirms the capability of the ISSA approach to examine the feature space efficiently and identify the best subset that gives higher classification accuracy.

We can observe from Tables 8 and 9 that ISSA obtains much better performance for most datasets. For example, for the Hepatitis data set, the ISSA algorithm used only 3.40 features and obtains the best accuracy with value 0.9123. For the waveform dataset, ISSA used only 19.78 features; but, regardless of this fact, its accuracy was compatible with the other algorithms (0.7690 against 0.7660).

Table 8
Average selection size of 20 runs for all optimizers.

DS	ISSA	SSA	GA	PSO	ALO	GWO
Wine	**4.50**	5.07	4.80	6.60	5.00	7.00
Hepatitis	**3.40**	4.50	5.10	8.30	5.70	8.20
Vehicle	**8.40**	**8.40**	12.30	11.05	**8.40**	11.90
Zoo	5.40	7.90	11.25	6.95	6.30	**5.20**
Heart	6.20	6.20	6.35	8.40	**5.40**	6.30
Breastcancer	**4.02**	5.90	5.30	5.40	6.10	5.37
Ionosphere	**11.07**	16.08	16.47	16.78	13.60	21.04
Lung cancer	20.45	22.06	**20.42**	27.80	25.47	28.04
Dermatlogy	**17.39**	21.27	20.48	25.70	23.47	24.03
Sonar	18.89	**17.49**	29.78	31.70	32.11	29.94
BreastEW	**12.91**	18.04	15.25	15.47	13.74	15.05
SoybeanSmall	**9.04**	13.26	15.23	14.79	21.49	20.79
Movementlibras	**23.78**	35.48	35.02	41.04	39.48	42.49
Parkinsons	5.51	6.79	7.25	9.48	11.23	**5.28**
Spambase	**23.78**	29.48	31.47	29.06	29.45	32.45
Waveform	**19.78**	23.71	29.80	30.00	29.45	32.54
Arrhythmia	**95.15**	109.48	125.40	119.95	113.27	186.00
Multiple Features	**125.47**	169.00	260.00	312.00	295.48	304.75
Semeion	**113.48**	222.00	226.60	223.11	218.40	218.20
Clean	59.60	61.60	66.80	62.50	**58.20**	93.20
CNAE	**216.23**	225.20	327.14	573.14	514.45	522.05
DNA	**59.8**	65.20	118.45	112.6667	139.80	89.80
Hillvalley	**19.40**	26.20	44.00	52.57	45.64	57.40

Table 9
Average classification accuracy (%) of 20 runs for all optimizers.

DS	ISSA	SSA	GA	PSO	ALO	GWO	FULL
Wine	**0.9780**	0.9550	0.9570	0.9230	0.9420	0.9120	0.9310
Hepatitis	**0.9123**	0.8939	0.8750	0.8605	0.8825	0.8479	0.8462
Vehicle	**0.7398**	0.7352	0.7163	0.6695	0.6814	0.6172	0.6489
Zoo	0.8720	0.7980	0.8540	0.8240	0.8050	**0.8750**	0.8314
Heart	0.8130	0.8130	**0.8240**	0.8220	0.8020	0.8070	0.8105
Breastcancer	**0.9570**	0.9550	0.9550	0.9510	0.9500	0.9530	0.9530
Ionosphere	**0.8530**	0.8360	0.8240	0.8480	0.8430	0.8190	0.8168
Lung cancer	0.5978	**0.6023**	0.4820	0.5627	0.5056	0.5014	0.4156
Dermatlogy	**0.9825**	0.9623	0.9645	0.9071	0.9322	0.9488	0.8158
Sonar	0.7340	**0.7370**	0.7170	0.7230	0.7140	0.7140	0.7019
BreastEW	**0.9610**	0.9420	0.9350	0.9490	0.9420	0.9490	0.9311
SoybeanSmall	0.9708	**0.9736**	0.9438	0.8648	0.9098	0.9205	0.875
Movementlibras	**0.6938**	0.6899	0.6902	0.6466	0.6597	0.6866	0.6436
Parkinsons	0.8529	0.8429	0.8492	**0.8653**	0.8367	0.8367	0.8087
Spambase	0.8814	0.8735	0.8229	0.8735	0.8804	**0.8839**	0.7807
Waveform	0.7660	**0.7690**	0.7620	0.7620	**0.7690**	0.7650	0.7569
Arrhythmia	**0.6600**	0.6378	0.5802	0.5707	0.5462	0.5641	0.5629
Multiple Features	**0.9611**	0.9468	0.9286	0.9359	0.9255	0.9106	0.9265
Semeion	**0.9899**	0.9595	0.9578	0.9314	0.9807	0.9732	0.9537
Clean	**0.8098**	**0.8098**	0.7648	0.7784	**0.8098**	0.7953	0.7601
CNAE	**0.8912**	0.8514	0.8246	0.8147	0.7962	0.8407	0.8083
DNA	**0.8611**	0.8429	0.7954	0.7979	0.7666	0.8425	0.7856
Hillvalley	**0.6319**	0.6319	0.5627	0.5507	0.5709	0.5544	0.5411

We additionally evaluate running time of ISSA with other algorithms. As proven in Table 10, ISSA needs between 17 s to 540 s relying on the dataset size to find the most effective subset. ISSA achieves the least running time in 9 out of 23 datasets, while GA gets the best values in 7 out of the datasets. In average, GA gets the best running time over all the datasets. ISSA is observed to be the second most effective performing on the 23 datasets.

Tables 11 and 12 present specific feature reduction examples for the Hepatits dataset, that has 19 input features, and for the Wine dataset, that has 13 input features. From the Hepatitis dataset, we can notice that ISSA proposes that only five features are suitable sufficient to classify data. For the Wine dataset, ISSA proposes that only six features will satisfy the same classification accuracy as if we examine all the features. In the majority of the tests executed, on average, about 75% of the features identified by ISSA are in common with the features identified by other optimizers.

For evaluating the stability of all optimizers on the feature space, Fig. 4 presents the standard deviation of fitness value. It is noticeable to remark that ISSA gets standard deviation values less than the other algorithms in 12 out of 23 datasets. Consequently, we can conclude that the ISSA always attain the optimal solution regardless of the initial solution and the random numbers of optimization which confirms the robustness of the ISSA optimizer. For more explanation, Figs. 5–7 present the reduction size, classification accuracy and computational time averaged over all the datasets. From the preceding Figures, we can observe that ISSA suggests a very good overall performance over all tested datasets in terms of feature reduction and classification accuracy. Moreover, ISSA has comparable computational time with GA which is the quickest optimizer.

Based on the results stated above, we can observe that ISSA produced the maximum classification accuracy and the least number of chosen features in comparison with the other optimizers on

Table 10
Average execution time (sec) of 20 runs for all optimizers.

DS	ISSA	SSA	GA	PSO	ALO	GWO
Wine	21.13	20.25	**19.60**	28.13	26.27	32.45
Hepatitis	**23.98**	24.16	27.20	29.36	29.35	28.15
Vehicle	40.39	40.39	**40.10**	42.32	52.03	54.74
Zoo	**17.39**	18.29	19.20	21.79	24.65	21.44
Heart	25.97	**24.23**	30.30	31.74	35.23	38.12
Breastcancer	**21.19**	21.70	21.70	23.40	27.14	23.78
Ionosphere	32.48	36.90	36.20	**30.27**	32.74	37.46
Lung cancer	31.78	**31.23**	33.70	33.48	36.48	39.73
Dermatlogy	**52.77**	54.03	55.20	61.24	57.81	64.48
Sonar	61.43	62.36	65.20	66.74	**58.48**	68.16
BreastEW	45.39	46.89	44.20	50.75	**41.48**	49.12
SoybeanSmall	43.16	42.35	**40.20**	45.48	47.23	45.19
Movementlibras	65.78	65.82	64.80	69.48	68.92	**64.78**
Parkinsons	54.30	57.48	55.30	61.49	**54.27**	62.48
Spambase	**195.73**	199.18	198.20	210.27	236.69	253.74
Waveform	293.45	292.78	**288.50**	290.37	301.07	305.41
Arrhythmia	122.19	130.78	**108.60**	125.75	242.78	233.18
Multiple Features	**475.84**	488.19	489.60	498.38	530.78	515.37
Semeion	**222.18**	245.78	281.20	298.64	315.45	305.47
Clean	**223.47**	225.73	250.20	249.48	287.35	277.73
CNAE	**540.78**	542.79	546.50	591.74	588.66	566.91
DNA	481.29	482.69	**461.30**	470.91	505.45	513.48
Hillvalley	82.78	83.79	**80.30**	97.58	115.47	105.38

Table 11
An example of the features selected for all optimizers using the Hepatits dataset.

Algorithm	All features	No. of selected	Features Indices	Reduction (%)
ISSA	19	5	1, 5, 10, 11, 18	26.31
SSA	19	5	5, 8, 9, 12, 19	26.31
GA	19	6	1, 6, 10, 11, 12, 18	31.57
PSO	19	5	1, 5, 11, 14, 18	26.31
ALO	19	5	5, 10, 12, 17, 19	26.31
GWO	19	6	1, 5, 13, 14, 17, 18	31.57

Table 12
An example of the features selected for all optimizers using the Wine dataset.

Algorithm	All features	No. of selected	Features Indices	Reduction (%)
ISSA	13	6	1, 3, 6, 10, 11, 12	46.15
SSA	13	7	1, 3, 7, 8, 9, 11, 12	53.84
GA	13	5	6, 7, 9, 10, 11	38.46
PSO	13	4	6, 10, 11, 12	30.76
ALO	13	7	1, 2, 3, 8, 9, 11, 12	53.84
GWO	13	6	3, 7, 8, 10, 11, 12	46.15

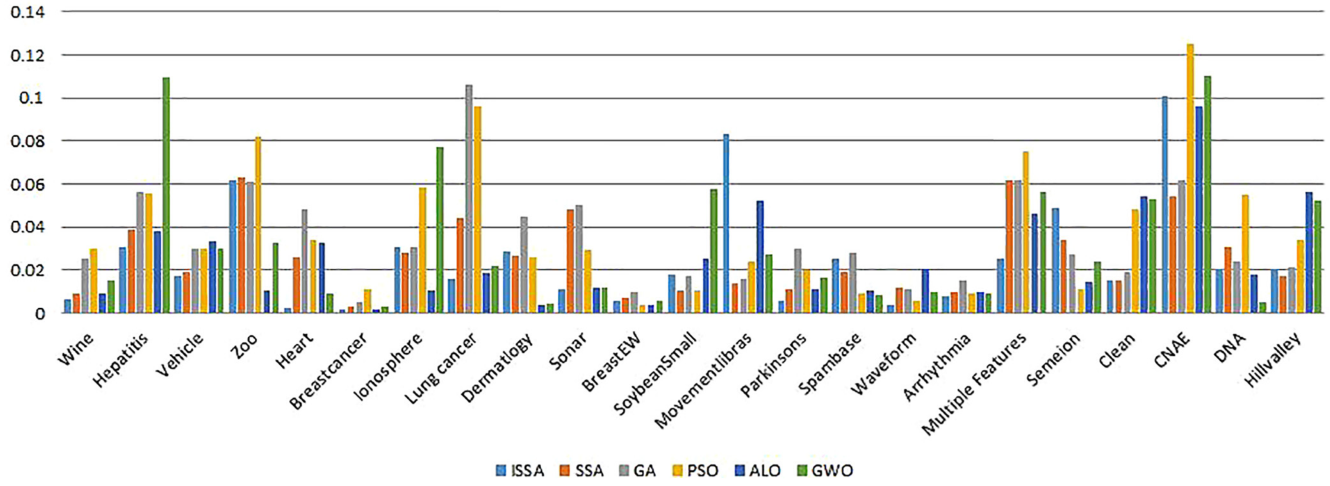

Fig. 4. STD fitness values acquired of all optimizers averaged over all the datasets.

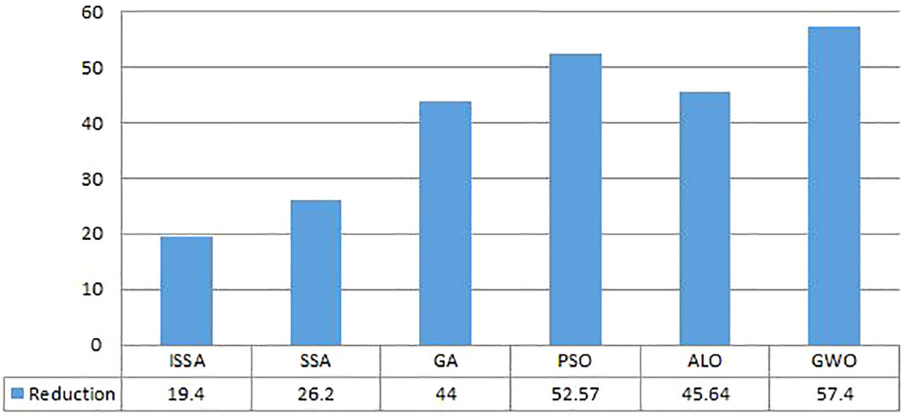

Fig. 5. Average feature reduction ratio of all optimizers averaged over all the datasets.

most datasets. We can notice that GA has been the quickest algorithm followed by ISSA and then SSA. Thus, if we study the best trade-off amongst classification accuracy, feature reduction and computational time, the ISSA is the best choice. According to these results, the ISSA is the suitable optimizer. Such an enhancement of the results got here from embedding inertia weight parameter in the searching procedure of SSA. This aids the algorithm to enhance the solution accuracy and the capability of selecting the feature

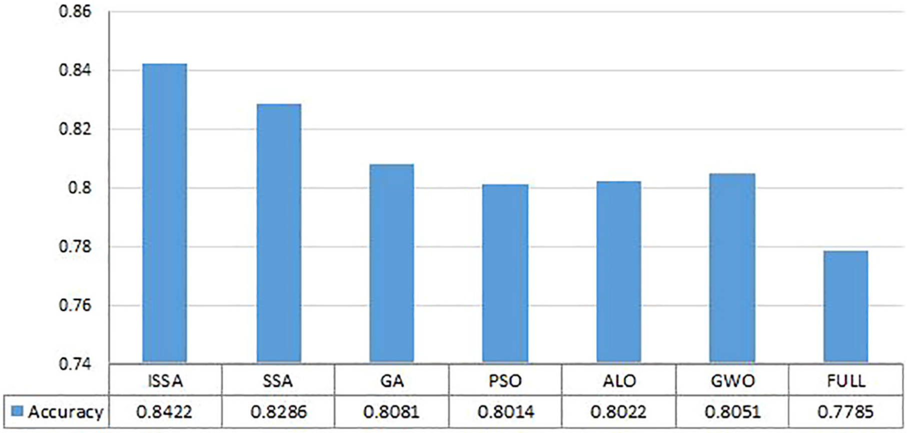

Fig. 6. Average classification accuracy of all optimizers averaged over all the datasets.

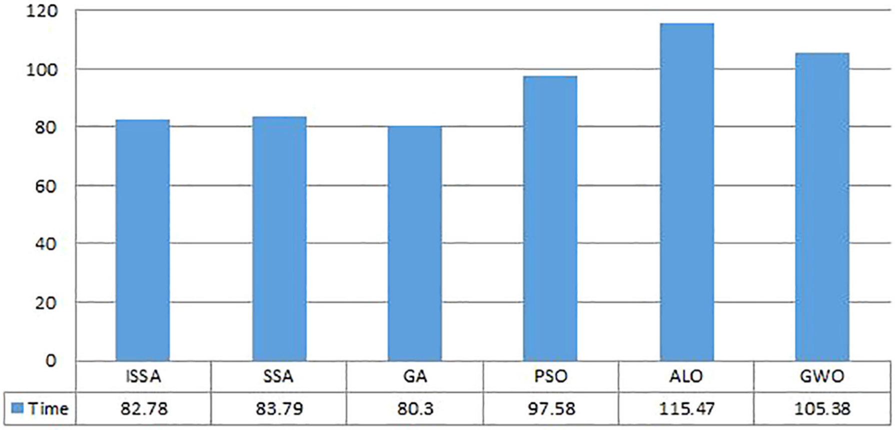

Fig. 7. Average execution time of all optimizers averaged over all the datasets.

subset better than the other optimizers. Optimization results verified that ISSA is an effective search algorithm since it has a simple concept and has the ability to find global solutions. ISSA has only a few control parameters and has many applications in real-world engineering optimization problems.

6. Conclusion

In this work, we have introduced an improved binary version of the salp swarm algorithm for feature selection tasks in wrapper approach. The performance of the ISSA algorithm is evaluated on distinct small, medium, and large size datasets, and compared with the basic SSA and four other nature-inspired algorithms which are PSO, GA, ALO and GWO. The proposed optimizer has outperformed the other optimizers in classification accuracy, being the second fastest optimizer and the one that has selected the minimal number of features. The results demonstrated that the integration of the inerita weight parameter in the SSA algorithm (ISSA) achieve better performance than other algorithms.

References

Abdullah, A.S., Ramya, C., Priyadharsini, V., Reshma, C., Selvakumar, S., 2017. A survey on evolutionary techniques for feature selection. Conference on Emerging Devices and Smart Systems (ICEDSS), pp. 58–62.

Ahmed, S., Mafarja, M., 2018. Feature selection using salp swarm algorithm with chaos. 2nd International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence (ISMSI2018).

Archana, S.R., Thanabal, M.S., 2016. Optimization algorithms for feature selection in classification: a survey. Int. J. Innovative Res. Comput. Commun. Eng. 4 (2).

Boussaïd, I., Lepagnot, J., Siarry, P., 2013. A survey on optimization metaheuristics. Inf. Sci. 237, 82–117.

Chandrashekar, G., Sahin, F., 2014. A survey on feature selection methods. Comput. Electr. Eng. 40 (1), 16–28.

Chuang, L.Y., Chang, H.W., Tu, C.J., Yang, C.H., 2008. Improved binary PSO for feature selection using gene expression data. Comput. Biol. Chem. 32 (1), 29–38.

Dash, R., 2018. An adaptive harmony search approach for gene selection and classification of high dimensional medical data. J. King Saud Univ. – Comput. Inf. Sci.

Emary, E., Zawbaa, H.M., Hassanien, A.E., 2016. Binary grey wolf optimization approaches for feature selection. Neurocomputing 172, 371–381.

Guyon, I., Elisseeff, A., 2003. An introduction to variable and feature selection. J. Mach. Learn. Res. 3, 1157–1182.

Jensen, R., 2005. Combining Rough and Fuzzy Sets for Feature Selection. School of Informatics, University of Edinburgh.

Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artif. Intell. 97 (1), 273–324.

Liu, H., Motoda, H., 1998. Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, Boston (p. 214).

Mafarja, M., Abdullah, S., 2011. Modified great deluge for attribute reduction in rough set theory. Fuzzy Systems and Knowledge Discovery (FSKD), Eighth International Conference, pp. 1464–1469.

Mafarja, M., Abdullah, S., 2014. Fuzzy Modified Great Deluge Algorithm for Attribute Reduction. Recent Advances on Soft Computing and Data Mining, Springer, pp. 195–203.

Mafarja, M., Abdullah, S., 2015. A fuzzy record-to-record travel algorithm for solving rough set attribute reduction. Int. J. Syst. Sci. 46 (3), 503–512.

Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M., 2017. Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems. Adv. Eng. Softw. 114, 163–191.

Nayak, S.K., Rout, P.K., Jagadev, A.K., Swarnkar, T., 2017. Elitism based multi-objective differential evolution for feature selection: a filter approach with an efficient redundancy measure. J. King Saud Univ. – Comput. Inf. Sci.

Please cite this article in press as: Hegazy, A.E., et al. Improved salp swarm algorithm for feature selection. Journal of King Saud University – Computer and Information Sciences (2018), https://doi.org/10.1016/j.jksuci.2018.06.003

Pereira, L.A.M., Rodrigues, D., Almeida, T.N.S., Ramos, C.C.O., Souza, A.N., Yang, X.-S., Papa, J.P., 2014. A binary cuckoo search and its application for feature selection. Cuckoo Search Firefly Alg., Stud. Comput. Intell. 516, 141–154.

Saroj, Jyoti, 2014. Multi-objective genetic algorithm approach to feature subset optimization. IEEE International Advance Computing Conference (IACC), pp. 544–548.

Sharawi, M., Zawbaa, H.M., Emary, E., 2017. Feature selection approach based on whale optimization algorithm. Ninth International Conference on Advanced Computational Intelligence (ICACI).

Talbi, E.G., 2009. Metaheuristics: from design to implementation, Wiley Online Library, p. 593.

UCI Machine Learning Repository: Data Sets. [Online]. Available:http://archive.ics.uci.edu/ml/index.php.

Valdez, F., 2015. Bio-Inspired Optimization Methods. Springer Handbook of Computational Intelligence, pp. 1533–1538.

Xue, B., Zhang, M., Browne, W.N., Yao, X., 2016. A survey on evolutionary computation approaches to feature selection. IEEE Trans. Evol. Comput. 20 (4), 606–626.

Yin, H., Gai, K., 2015. An empirical study on preprocessing high-dimensional class-imbalanced data for classification. 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, pp. 1314–1319.

Yin, H., Gai, K., Wang, Z., 2016. A classification algorithm based on ensemble feature selections for imbalanced-class dataset. 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), pp. 245–249.

Zawbaa, H.M., Emary, E., Parv, B., 2015. Feature selection based on antlion optimization algorithm. Third World Conference on Complex Systems (WCCS), pp. 1–7.

Zawbaa, H.M., Emary, E., Grosan, C., 2016. Feature selection via chaotic antlion optimization. PLOS ONE 11 (3), e0150652.

Zhong, N., Dong, J., Ohsuga, S., 2001. Using rough sets with heuristics for feature selection. J. Intell. Inf. Syst. 16 (3), 199–214.

Please cite this article in press as: Hegazy, A.E., et al. Improved salp swarm algorithm for feature selection. Journal of King Saud University – Computer and Information Sciences (2018), https://doi.org/10.1016/j.jksuci.2018.06.003