



A novel chaotic salp swarm algorithm for global optimization and feature selection

Gehad Ismail Sayed^{1,2} · Ghada Khoriba³ · Mohamed H. Haggag³

© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Salp Swarm Algorithm (SSA) is one of the most recently proposed algorithms driven by the simulation behavior of salps. However, similar to most of the meta-heuristic algorithms, it suffered from stagnation in local optima and low convergence rate. Recently, chaos theory has been successfully applied to solve these problems. In this paper, a novel hybrid solution based on SSA and chaos theory is proposed. The proposed Chaotic Salp Swarm Algorithm (CSSA) is applied on 14 unimodal and multimodal benchmark optimization problems and 20 benchmark datasets. Ten different chaotic maps are employed to enhance the convergence rate and resulting precision. Simulation results showed that the proposed CSSA is a promising algorithm. Also, the results reveal the capability of CSSA in finding an optimal feature subset, which maximizes the classification accuracy, while minimizing the number of selected features. Moreover, the results showed that logistic chaotic map is the optimal map of the used ten, which can significantly boost the performance of original SSA.

Keywords Salp swarm algorithm · Global optimization · Feature selection · Chaos theory

1 Introduction

Recently, meta-heuristic algorithms have become surprisingly very popular. This is due to proving their superiority in solving many optimization problems. Some of the most popular meta-heuristic algorithms are Particle Swarm Optimization (PSO) [1], Artificial Bee Colony (ABC) [2] and Ant Colony Optimization [3]. Regardless of the different structure of meta-heuristic algorithms, they are divided into two main phases. These phases are exploitation and exploration phases. In exploration phase, an optimizer algorithm can efficiently discover the search space, mostly by randomization. However, during this phase, the population may face abrupt changes. In exploitation phase, an optimizer algorithm converges toward the most promising region. But, in most of the cases, there is no clear boundary between these

two phases, which makes meta-heuristic algorithms trapped into local optima. This is due to their stochastic nature and improper balancing between exploration and exploitation. Thus, many studies have been presented to improve the performance of meta-heuristic algorithms and to overcome this problem. Most of these studies used chaotic theory in their system [4–7].

Chaos is one of the most common mathematical approaches recently employed to boost the performance of meta-heuristic algorithms. It is defined as the simulation of the dynamic behavior of nonlinear systems [8]. Over the years, chaos has attracted a great attention and has been applied in different sciences such as chaos control [9], synchronization [10], optimization researches [11, 12] and so on. Each hyperchaotic system has its own dynamic behavior characteristics, which can be applied on certain applications [13]. Authors in [14] proposed a new chaotic complex model, namely cutting edge chaotic complex $L\ddot{u}$ system, where a complex nonlinear expression is added to the third equation of the $L\ddot{u}$ system. Also, the authors present complex antilag synchronization (CALS) schema. They applied their proposed CALS for two indistinguishable chaotic complex systems with different introductory qualities. The results show the effectiveness of their proposed complex system. Authors in [15] proposed

✉ Gehad Ismail Sayed
darkspot.1993@yahoo.com

¹ Faculty of Computers and Information, Cairo University, Giza, Egypt

² Scientific Research Group in Egypt, Giza, Egypt

³ Faculty of Computers and Information, Helwan University, Helwan, Egypt

another hyperchaotic system, where fractional-order is used. The experimental results show that the effectiveness of their proposed system in terms of equilibrium points, Lyapunov spectrum, and attractor forms. Moreover, the results show that their system exhibits larger Lyapunov exponents compared with other hyperchaotic systems. Authors in [16] applied the ten-term chaotic system without equilibrium on multimedia security application including image encryption and sound steganography. The results show that their system is able to encrypt 128kbit data and hide in sound files. Also, the properties of chaos is embedded with the searching mechanism of meta-heuristic algorithms and applied in different applications. Authors in [17] employed logistic chaotic map with Genetic Algorithm (GA) for encrypting the image. In this approach, the logistic chaotic map is used to encode the initial image, after that GA is applied to improve the encryption results. The simulation results reveal the capability of their approach in avoiding the frequent attacks. Authors in [5] hybridized chaos optimization algorithm (COA) and PSO. The experimental results show that embedding logistic chaotic map can significantly improve the performance of PSO in terms of convergence rate and time oscillation. Another approach based on hybridization with chaotic maps is proposed in [6]. In this approach, the authors employed ten chaotic maps with Biogeography-Based Optimization (BBO). These maps are gauss/mouse, Chebyshev, logistic, iterative, piecewise, sine, singer, circle, sinusoidal and tent chaotic maps. The simulation results show that the proposed chaotic biogeography-based optimization is able to improve the performance of original BBO for both exploration and exploitation. Authors in [18] employed piecewise chaotic map with search iteration of Harmony Search (HS). Their main objective is to minimize makespan for the permutation flow shop scheduling problem with limited buffers. The experimental results show the efficiency of their proposed chaotic harmony search algorithm compared with the original HS and other meta-heuristic algorithms in terms of solution quality and robustness. A modified version of Antlion Optimization Algorithm (ALO) based on chaotic maps, namely Chaotic Antlion Optimization Algorithm (CALO) is proposed in [19]. The authors applied CALO for feature selection problem, where five chaotic maps are employed. These maps are the singer, piecewise, logistic, tent and sinusoidal. The experimental results reveal the capability of CALO to find better feature subset than original ALO and other meta-heuristic algorithms. Authors in [7] proposed Chaotic Crow Search Algorithm (CCSA) for solving feature selection problem. In this approach, ten chaotic maps are employed. The experimental results show that the efficiency of CCSA in finding an optimal feature subset, which maximizes the classification performance, while minimizing the number of selected features.

Feature selection is an essential task for machine learning tasks. Recently, data obviously increases in both number of features and number of instances. Thus, removing irrelevant and noisy features is considered a challenging task especially for large dataset. Feature selection algorithms have proved their efficiency in removing redundant features, reducing computational cost, improving the performance of a classifier and reducing the required storage [20]. Additionally, they have been successfully applied in many applications including image retrieval [21], text categorization [22], customer relationship management [23] and diagnostic medical support systems [24, 25]. Also, feature selection can influence the clustering performance [20]. Moreover, it can further help for better data interpretation and understanding than using all features. Thus, several studies have been presented in literature applying feature selection such as in [20, 26]. From studies, it has been proved that building cluster with relevant features is more interpretable and practical in use than building a cluster with all features, which can include noise and irrelevant features. Feature selection algorithms are divided into two families; wrapper-based and filter-based algorithms. In wrapper-based algorithms, machine learning algorithms are used in evaluation, while statistical methods are used for filter-based algorithms [27]. However, it has been proven that wrapper-based algorithms provide better results compared with filter-based algorithms. However, wrapper-based algorithms are computationally expensive. In addition, filter-based algorithms such as sequential forward selection (SFS) and sequential backward selection (SBS) suffer from entrapment at local optima [27]. On the other hand, meta-heuristic algorithms can find the optimal feature subset through using their agents. Recently, meta-heuristic algorithms are applied for feature selection problem such as Moth-Flame Optimization (MFO) [28], Gray Wolf Algorithm (GWO) [29], Particle Swarm Optimization (PSO) [30] and Artificial Bee Colony (ABC) [31].

In this work, a novel hybridization approach based on Salp Swarm Algorithm (SSA) and chaos theory is proposed. SSA is one of most recently proposed algorithm. It is proposed in 2017 by Mirjalili [32]. The main inspiration of the algorithm came from the swarming behavior of salps. Salps are kind of Salpidae with transparent barrel-shaped body. The inventor of original SSA algorithm, showed that the efficiency of this algorithm compared with other well-know meta-heuristics algorithms. Although, slow convergence speed and entrapment in local optima are two possible problems, similar to other meta-heuristic algorithms. As it is previously mentioned, chaos theory is one of most common methods used to improve the performance of evolutionary algorithms. Currently, to the best of our knowledge, there is no work proposed in

literature to improve the exploitation and the exploration of SSA. The main contributions of this paper summarized as follows:

- A novel hybridization approach based on SSA and chaos theory is proposed.
- The proposed approach is applied for 14 global optimization problem
- A binary version of CSSA is proposed and applied for feature selection problem, where 20 benchmark datasets are employed.
- Ten most popular chaotic maps proposed in literature are employed and compared.
- Several evaluation criteria are used in evaluation. These criteria are; mean, standard deviation, p -values of the Wilcoxon rank sum, trajectory, search history, average fitness of whole population, convergence curves and average length of the selected feature subset.
- The performance of the proposed approach is compared with original SSA and five other well-known meta-heuristic algorithms; artificial bee colony (ABC), moth flame optimization (MFO), particle swarm optimization (PSO), chicken swarm optimization (CSO) and gray wolf optimizer (GWO)

The rest of this paper is organized as follow. Section 2 provides a full description of the original SSA algorithm concerning its nature inspiration and mathematical model. Also, the basics of chaotic concepts are illustrated in this section. The novel Chaotic Salp Swarm Algorithm (CSSA) is described in Section 3. Section 4 provides simulation results and discussions along with the used benchmark datasets. Finally, conclusions are presented in Section 5.

2 Basics and background

In this section, the basic knowledge about salp behavior and optimization algorithms based on salp behavior are proposed. In addition, the basic knowledge of chaotic maps is discussed as well.

2.1 Salp swarm algorithm (SSA)

Inspiration analysis Salp Swarm Algorithm (SSA) is one of meta-heuristic algorithms recently proposed by Mirjalili [32] in 2017. The main inspiration of this algorithm came from the swarming behavior of salps. Salps are kind of Salpidae. They have a transparent barrel-shaped body and similar tissues like jellyfishes. In addition, they move like jellyfish, as the water is pumped through the body as propulsion to move forward. Sometimes, they form a

swarm, namely salp chain. However, till now, the reason behind this behavior is not obvious. Some studies refer to this behavior when salps are searching for optimal locomotion based on rapid foraging and coordinated changes.

Mathematical model of SSA The population of SSA is divided into two groups. These two groups are leader and followers. The leader is the one that taking the position at the front of the chain, while the remainder of salps is known as followers. Let, dim is the number of variables or number of dimensions for a given problem. y is defined as the position of a salp. F denotes the food source, which is the target of the swarm in the search space. The leader updates his position according to the following equation:

$$y_i^1 = \begin{cases} F_i + r_1((ub_i - lb_i)r_2 + lb_i), & r_3 \geq 0 \\ F_i - r_1((ub_i - lb_i)r_2 + lb_i), & r_3 < 0 \end{cases} \quad (1)$$

where y_i^1 denotes as first salp's position in $i - th$ dimension, known as leader, lb_i , ub_i are the lower boundary and upper boundary at $i - th$ dimension, respectively, F_i is the food's position in $i - th$ dimension and r_1 , r_2 and r_3 are random numbers. r_1 is responsible for balancing between exploitation and exploration. It is mathematically defined as follows:

$$r_1 = 2e^{-\left(\frac{4t}{T}\right)^2} \quad (2)$$

where t is the current iteration number, and T is the maximum number of iterations. r_2 and r_3 are two random numbers uniformly generated in range $[0, 1]$. r_3 is responsible for indicating whether the next position should to be toward negative infinity or positive infinity. The followers update their position according to Newton's law of motion using the following equation:

$$y_i^j = \frac{1}{2}\alpha l^2 + \beta_0 l \quad (3)$$

where y_i^j shows the position of $j - th$ follower in $i - th$ dimension, $i \geq 2$, β_0 is the initial speed, $\alpha = \frac{\beta_{final}}{\beta_0}$, $\beta = \frac{y - y_0}{t}$ and l is time. As the time is called an iteration in the optimization process, the discrepancy within iteration equals to one. Let $\beta_0 = 0$, the updating position of followers in $i - th$ dimension can be represented as follows:

$$y_i^j = \frac{1}{2} \left(y_i^j + y_i^{j-1} \right) \quad (4)$$

2.2 Chaotic maps

Most meta-heuristic algorithms have randomness parameters. These parameters are using probability distribution, usually Gaussian or uniform distributions. Recently, chaos theory used to enhance these parameters [33]. It has the same characteristics as randomness, with better dynamical and statistical characteristics [34]. Chaos is known as a phenomenon. Any change of the initial condition of chaos may lead to non-linear change for the future behavior. Three main properties are describing the chaos; (1) quasi-stochastic, (2) ergodicity and (3) sensitivity to initial conditions. Mixing these characteristics can guarantee the diversity of the generated solutions. Thus, this diversity can be enough to reach every mode of the multi-modal problem [35]. Quasi-stochastic is defined as the ability to replace random variables with values of chaotic maps. Ergodic property refers to the ability of chaotic variables to search non-repeatedly all states within a certain range. Finally, sensitivity to initial condition property is defined as any small change, in the initial starting points, may lead to different behavior. Combining all these properties can significantly boost the performance of meta-heuristic optimization algorithms [36].

In this study, ten distinguished non-invertible maps with different characteristics are used. Table 1 shows these maps, where o_s denotes the s -th number in the chaotic sequence and s is defined as the index of the chaotic sequence o . The other parameters including c , d , and μ are the control parameters. These parameters are used to determine the chaotic behavior of the dynamic system. From this table, it can be observed that chaotic maps have a determinate form without including random factors. In this work, the initial point (o_0) for all adopted chaotic maps is initially set to 0.7. This value is same value used for the same ten chaotic maps in [37].

3 The novel chaotic salp swarm algorithm

In this section, a novel chaotic salp swarm algorithm (CSSA) is proposed where chaotic maps are employed to replace random variables with chaotic variables. The original has mainly three main parameters which affect its performance. These parameters are r_1 , r_2 and r_3 . As it is shown in (2), r_1 is linearly decreased through the iterations, while r_3 is responsible for determining whether the next position should be toward negative or positive infinity. As it can be seen from (1), r_2 and r_1 are the two main parameters influencing the updating position of a salp. Thus, they significantly impact on balancing between exploration and exploitation. Exploration concentrates on finding new better solutions by investigating the search space on a

large scale, while exploitation concentrates on exploiting the data in the local region [7]. An algorithm should properly balance these two components to approximate the global optima. In this study, chaotic maps are employed to adjust a r_2 parameter of SSA. Equation (5) shows the updating of r_2 parameter according to the chaotic map. Equation (5) shows the updated position of a salp according to the chaotic map, where o^t is the obtained value of chaotic map at t -th iteration.

$$r_2 = o_s \quad (5)$$

$$y_i^t = \begin{cases} F_i + r_1((ub_i - lb_i)o^t + lb_i), & r_3 \geq 0 \\ F_i - r_1((ub_i - lb_i)o^t + lb_i), & r_3 < 0 \end{cases} \quad (6)$$

Such embedding chaotic maps into updating position of salp can improve the performance and the convergence rate of SSA as it will be demonstrated later in the following section. The mathematical formula of each used chaotic maps is previously defined at Table 1. In this study, the proposed CCSA is applied not only on 14 unimodal and multimodal benchmark optimization problems but also on feature selection problem. In CSSA feature selection algorithm, the solution pool is restricted to discrete binary form, where positions of salps are restricted to $\{0, 1\}$ [38]. Let a salp position (solution in the search space) y expressed by a i -th-dimensional variable $y = [x_1, x_2, \dots, x_{dim}]$, where dim is the maximum number of dimensions. When the value of variable equals to 1, then it means that the corresponding feature is selected and when the value equals to 0, then it means that the corresponding feature is not selected. For example, the feature representation for data have 5 features can be as $[1, 0, 0, 1, 1]$ or $[0, 0, 1, 1, 1]$ and so on. From previous two examples, it can be observed that each solution has different features and each solution has different length. Equation (7) shows how each agent transfers from continues to discrete binary space, where B is random number in range $[0, 1]$.

$$y_i^t = \begin{cases} 1 & \text{if } (s(y_i^t)) \geq B \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where

$$s(y_i^t) = \frac{1}{1 + e^{10(y_i^t - 0.5)}} \quad (8)$$

Next, the detailed description of the proposed chaotic version of SSA algorithm is presented as follows:-

3.1 Parameters initialization

In the beginning, CSSA starts with randomly initialized salps positions. Then, it initially sets the initial parameters.

Table 1 Definition of the ten adapted chaotic maps

No.	Name	Definition	Range
CSSA1	Chebyshev	$o_{s+1} = \cos(s \cos^{-1}(o_s))$	$(-1, 1)$
CSSA2	Circle	$o_{s+1} = \text{mod}(o_s + d - (\frac{c}{2\pi}) \sin(2\pi o_s), 1)$, $c = 0.5$ and $d = 0.2$	$(0, 1)$
CSSA3	Guass/mouse	$o_{s+1} = \begin{cases} 1, & o_s = 0 \\ \frac{1}{\text{mod}(o_s, 1)}, & \text{otherwise} \end{cases}$	$(0, 1)$
CSSA4	Iterative	$o_{s+1} = \sin\left(\frac{c\pi}{o_s}\right)$, $c = 0.7$	$(-1, 1)$
CSSA5	Logistic	$o_{s+1} = c o_s (1 - o_s)$, $c = 4$	$(0, 1)$
CSSA6	Piecewise	$o_{s+1} = \begin{cases} \frac{o_s}{l}, & 0 \leq o_s < l \\ \frac{o_s - l}{0.5 - l}, & l \leq o_s < 0.5 \\ \frac{1 - l - o_s}{0.5 - l}, & 0.5 \leq o_s < 1 - l \\ \frac{1 - o_s}{l}, & 1 - l \leq o_s < 1 \end{cases}$, $l = 0.4$	$(0, 1)$
CSSA7	Sine	$o_{s+1} = \frac{c}{4} \sin(\pi o_s)$, $c = 4$	$(0, 1)$
CSSA8	Singer	$o_{s+1} = \mu(7.86o_s - 23.31o_s^2 + 28.75o_s^3 - 13.302875o_s^4)$, $\mu = 1.07$	$(0, 1)$
CSSA9	Sinusoidal	$o_{s+1} = c o_s^2 \sin(\pi o_s)$, $c = 2.3$	$(0, 1)$
CSSA10	Tent	$o_{s+1} = \begin{cases} \frac{o_s}{0.7}, & o_s < 0.7 \\ \frac{10}{3}(1 - o_s), & o_s \geq 0.7 \end{cases}$	$(0, 1)$

The lower boundary and upper boundary are initially set respect to the used benchmark function, while for the used benchmark dataset, it initially set the lower boundary to 0 and the upper boundary to 1 for given data. The maximum number of iterations is set for global optimization problem to 500, while 30 for feature selection problem. Finally, the population size is set to 50 for global optimization problem and 20 for feature selection problem. The values of population size and number of iterations are set low for feature selection problem, due to the complexity of the search space.

3.2 Fitness function

Fitness function is used to evaluate each solution (salp position). All the used global benchmark problems are minimization problems. Thus, the solution with the minimum fitness value $\min(f(X))$ is selected as the best solution obtained so far. While for feature selection problem, the optimal solution is the one, which maximizes the classification accuracy, while minimizing the number of selected features. Equation (9) shows the used fitness function for feature selection problem, where these two objectives are combined into one by setting a weight factor. In this equation, Acc is the classification accuracy obtained from K-nearest neighbor (KNN) classifier, where k equals to 3 with mean absolute distance. KNN is one of supervised machine learning algorithms. It classifies the new instance based on distance from the new instance to the training instances [38]. In this work, KNN classifier is used to

indicate the goodness of the selected features subset. The selection of K and distance method are based on trial and error method. w_f is the weight factor, which used to control the importance of the number of selected features and the classification accuracy. In this study, our goal is to maximize the classification accuracy in the first place then to minimize the number of selected features. Thus, w_f is set to 0.8. L_f denotes the length of selected features subset, while L_t denotes the total number of features for a given dataset.

$$Fn_t = \text{maximize} \left(w_f \times Acc + (1 - w_f) \times \left(1 - \frac{L_f}{L_t} \right) \right) \quad (9)$$

3.3 Positions updating

After evaluating the fitness function of each salp and selecting the best salp position. The best salp position updates its position according to (6), (7), (3) and (4).

3.4 Termination criteria

The process of evaluating each salp and updating the position of best salp will repeat again and again until it reaches to the maximum number of iterations or the optimal solution is early found. In our case, the optimization process terminates when we the maximum number of iteration is met; 50 for global optimization problem and 30 for feature selection problem.

The pseudo code of the proposed CSSA is defined at Algorithm 1.

Algorithm 1 Chaotic salp swarm algorithm

```

1: Set the initial value of maximum number of iterations
    $tMax$ , upper boundary  $ub$ , and lower boundary  $lb$ ,
   population size  $n$ , number of dimensions  $dim$  and
   define the fitness function  $F_n$ .
2: Randomly initialize the positions of salps  $y_i^t$  corre-
   sponding to  $ub$  and  $lb$ , where  $i = 1, 2, \dots, n$  and
    $t = 1, 2, \dots, tMax$ .
3: Set  $t := 1$ . {Counter initialization}.
4: repeat
5:   Evaluate the fitness function of each salp position
      $F_n(y_i^t)$ .
6:   Set the best salp position to  $y_{best}$ 
7:   Update  $r_1$  according to (2)
8:   Get the value of chaotic map  $o_t$ 
9:   Update  $r_2$  according to (5)
10:  for ( $i = 1 : i \leq n$ ) do
11:    if  $i == 1$  then
12:      Update the position of the leading salp
        according to (6)
13:    else
14:      Update the position of the follower salp
        according to (4)
15:    end if
16:    Update the salp position according to (7)
17:  end for
18:  Check the feasibility of  $y_i^t$ 
19:  Evaluate the new position of crow  $y_i^t$ 
20:  if  $F_n(y_i^t)$  is better than  $F_n(y_{best})$  then
21:    Update the position of the best salp.
22:  end if
23:  Set  $t = t + 1$ . {Iteration counter increasing}.
24: until ( $t < tMax$ ). {Termination criteria satisfied}.
25: Produce the best salp position  $y_{best}$ .

```

4 Simulation results and discussions

In this section, the proposed CSSA algorithm is applied for two problems; global optimization problem and feature selection problem. Section 4.1 discusses the performance of the proposed CSSA for solving the global optimization problem. Moreover, in this section, a comparison between CSSA with different chaotic maps is presented. Section 4.2 compares the performance of the proposed CSSA as feature selection algorithm with other meta-heuristic feature

selection algorithms. All the conducted experiments are performed on the same PC with the same specification including Core i3 and RAM 2 GB on OS Windows 7. Moreover, all the experiments are reported on average 30 independent runs.

4.1 CSSA for global optimization problem

This subsection aims to evaluate the performance of the proposed CSSA, to compare CSSA with different chaotic maps and to select the optimal chaotic map. Several measurements are used in this study. These measurements are mean, standard deviation (SD) and p -values of the Wilcoxon rank sum. Also, the search history, the trajectory, the average fitness of all group and the convergence curve are used.

Due to the stochastic behavior of meta-heuristic optimization algorithms, there is a need to employ a set of test function to prove the superiority and efficiency of an algorithm [39]. In this work, several experiments are conducted to evaluate the performance of the proposed CSSA, where two benchmark categories are employed. The mathematical definition and characteristic of the benchmark functions are presented in Tables 12, 13, 14 and 15 in the Appendix section. In these tables, lb denotes the lower boundary of search space, while ub denotes the upper boundary of the search space, opt is defined for the optimal value of a function, and finally, dim is the number of dimensions. For more information of these function, the authors refer to [40–42]. The first category, namely unimodal benchmark functions have no local optima and one global optima. This category is used to evaluate the exploitation of the algorithm. The second category known as multimodal benchmark functions has multiple local optima and multi-global optima, despite unimodal function. The properties of this category are used to evaluate the exploration of the algorithm. Also, it is used to evaluate the capability of the algorithm in avoiding the local optima.

Table 2 compares the performance of the original SSA with different versions of CSSA for unimodal benchmark functions in terms of mean and standard deviation. The notations of CSSA1 till CSSA10 are previously defined in Table 1. In Table 2, the best results are underlined. As it can be observed, CSSA with logistic chaotic map obtained the best mean values for three out of six benchmark functions. These functions are $F1$, $F2$ and $F4$. Also, it can be observed that CSSA with Gauss, piecewise and chebychev are in second place. Additionally, it can be seen that the proposed CSSA algorithms have high stability compared with original SSA. This is due to the stochastic behavior of

Table 2 Comparison of statistical results obtained using SSA and different version of CSSA for unimodal benchmark functions

F	Mean	Std	Mean	Std	Mean	Std	Mean	Std
	SSA		CSSA1		CSSA2		CSSA3	
F1	5.36E-04	0.0015	1.87E-09	6.26E-10	1.85E-09	3.15E-10	2.07E-09	4.11E-10
F2	0.1737	0.4272	5.52E-06	1.46E-06	4.24E-06	1.01E-06	6.28E-06	1.70E-06
F3	4.23E-04	0.001	7.05E-10	5.03E-10	7.30E-10	5.01E-10	9.49E-10	5.68E-10
F4	0.0459	0.1425	1.21E-05	1.47E-06	1.31E-05	3.21E-06	1.33E-05	3.05E-06
F5	679.1383	803.8195	8.9486	0.0164	8.9472	0.0197	<u>8.9354</u>	<u>0.0173</u>
F6	28.1148	20.3592	2.05E-10	6.67E-11	2.23E-10	8.14E-11	2.34E-10	3.93E-11
F7	0.0258	0.0131	<u>9.06E-05</u>	<u>9.27E-05</u>	9.82E-05	7.37E-05	1.90E-04	1.54E-04
	CSSA4		CSSA5		CSSA6		CSSA7	
F1	1.69E-09	4.38E-10	<u>1.40E-09</u>	<u>3.31E-10</u>	1.81E-09	4.61E-10	2.45E-09	9.35E-10
F2	5.38E-06	1.29E-06	<u>4.14E-06</u>	<u>1.68E-06</u>	5.22E-06	1.42E-06	5.27E-06	1.29E-06
F3	6.13E-10	2.91E-10	9.01E-10	4.87E-10	8.87E-10	6.93E-10	<u>5.40E-10</u>	<u>1.94E-10</u>
F4	1.30E-05	2.84E-06	<u>1.08E-05</u>	<u>1.55E-06</u>	1.25E-05	3.17E-06	1.19E-05	2.35E-06
F5	8.954	0.0152	8.9428	0.0192	8.9425	0.074	8.9436	0.0135
F6	1.79E-10	5.65E-11	1.83E-10	5.02E-11	<u>1.77E-10</u>	<u>3.89E-11</u>	2.18E-10	6.50E-11
F7	2.52E-04	2.03E-04	9.30E-05	1.94E-04	2.47E-04	1.59E-04	1.94E-04	1.58E-04
	CSSA8		CSSA9		CSSA10			
F1	1.67E-09	6.77E-10	1.46E-09	7.72E-10	1.82E-09	4.82E-10		
F2	4.68E-06	1.62E-06	5.43E-06	2.24E-06	5.53E-06	1.84E-06		
F3	7.37E-10	3.93E-10	8.69E-10	9.76E-10	1.07E-09	7.09E-10		
F4	1.24E-05	3.49E-06	1.22E-05	5.57E-06	1.24E-05	3.00E-06		
F5	8.9517	0.0191	8.9785	0.016	8.9483	0.0122		
F6	2.04E-10	8.04E-11	1.93E-10	9.63E-11	1.81E-10	3.87E-11		
F7	1.89E-04	1.57E-04	1.81E-04	1.67E-04	5.11E-04	6.48E-04		

The underlined entries used to highlight the best obtained results

SSA meta-heuristic algorithm, where random variables are included. Moreover, it can be seen that all versions of CSSA obtained better results than original SSA. Table 3 shows p -values obtained from Wilcoxon's rank sum test for the pair-wise comparison of the best value obtained from whole iterations with a 5% significance level. This test is used to determine whether the proposed CSSA algorithms provide a significant improvement compared to the original SSA. As it can be observed, all CSSA versions are statistically significant. From these two tables, it can be concluded that embedding chaotic variables during optimization process of SSA can significantly boost the performance of original SSA in terms of exploitation and stability.

Table 4 compares mean and standard deviation values of SSA with the proposed CSSA algorithms for multimodal benchmark functions. As it can be observed, again, CSSA with logistic chaotic map obtained the best results for most of the multimodal functions. As it obtained the best mean values for four functions out of seven functions;

$F9$, $F10$, $F12$ and $F13$. While CSSA with Gauss chaotic map obtained the optimal results for $F13$ and $F14$. Also, the results from embedding different chaotic maps in optimization process of SSA obtained better results than SSA. Moreover, it can be observed the high stability of the proposed CSSA algorithms compared with SSA. Table (5) shows the p -values of Wilcoxon's rank-sum test. As it can be seen, most of the obtained values are less than 0.05, which means that the proposed CSSA algorithms are statistically significant. From these two tables, it can be concluded that embedding the characteristics of chaotic maps able to obviously avoid local optima and obtain global optima with a small number of iterations. This is due to a proper balance between exploitation and exploration when employing chaotic variables during optimization process of SSA. Based on all the produced results, we select logistic chaotic map as the most appropriate map. Thus, in next, this map will be further evaluated in more details.

Table 3 Comparison of p -values of the Wilcoxon ranksum test obtained using SSA and different version of CSSA over unimodal benchmark functions

F	SSA vs. CSSA1	SSA vs. CSSA2	SSA vs. CSSA3	SSA vs. CSSA4	SSA vs. CSSA5
F1	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F2	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F3	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F4	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F5	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F6	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F7	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
	SSA vs. CSSA6	SSA vs. CSSA7	SSA vs. CSSA8	SSA vs. CSSA9	SSA vs. CSSA10
F1	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F2	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F3	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F4	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F5	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F6	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F7	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05

Table 4 Comparison of statistical results obtained using SSA and different version of CSSA for multimodal benchmark functions

F	Mean	Std	Mean	Std	Mean	Std	Mean	Std
	SSA		CSSA1		CSSA2		CSSA3	
F8	- 2.45E+03	372.4443	- 2.65E+03	205.4819	- 2.54E+03	269.9168	- 2.68E+03	373.0626
F9	28.5552	14.9136	3.30E-10	1.84E-10	2.86E-10	1.14E-10	3.36E-10	1.78E-10
F10	1.3758	1.3597	9.73E-06	2.01E-06	1.04E-05	2.93E-06	8.82E-06	1.37E-06
F11	0.1981	0.1348	<u>1.64E-09</u>	<u>8.19E-10</u>	2.06E-09	1.08E-09	1.69E-09	5.19E-10
F12	0.7009	1.033	0.1596	0.0574	0.2147	0.0751	0.15236	0.0566
F13	0.0048	0.0083	0.0016	3.79E-04	0.0055	0.0092	<u>0.0015</u>	<u>4.02E-04</u>
F14	- 4.5898	1.0662	- 4.869	1.66E-01	- 5.2736	1.5971	<u>- 9.8918</u>	<u>1.87E-01</u>
	CSSA4		CSSA5		CSSA6		CSSA7	
F8	- 2.53E+03	418.136	- 2.51E+03	335.2845	- 2.29E+03	353.0461	<u>- 2.71E+03</u>	<u>639.7074</u>
F9	2.49E-10	1.18E-10	<u>1.30E-10</u>	<u>2.26E-10</u>	2.98E-10	1.35E-10	2.54E-10	8.00E-11
F10	7.92E-06	1.82E-06	<u>7.03E-06</u>	<u>2.41E-06</u>	8.09E-06	2.27E-06	7.20E-06	1.80E-06
F11	1.78E-09	8.11E-10	2.48E-09	1.48E-09	2.46E-09	2.30E-09	2.13E-09	5.86E-10
F12	0.2588	0.1223	<u>0.1462</u>	<u>0.0627</u>	0.1734	0.0623	0.2069	0.0507
F13	0.0022	0.0013	<u>0.0015</u>	<u>0.0011</u>	<u>0.0015</u>	<u>3.49E-04</u>	0.0031	0.0017
F14	- 7.546	1.4474	- 9.8451	1.4758	- 6.9001	2.7323	- 7.7459	1.888
	CSSA8		CSSA9		CSSA10			
F8	- 2.47E+03	331.5414	- 2.24E+03	241.6747	- 2.48E+03	602.5672		
F9	1.63E-10	8.12E-11	28.8593	20.454	2.83E-10	2.05E-10		
F10	9.61E-06	5.73E-06	8.84E-06	2.75E-06	9.16E-06	2.83E-06		
F11	4.44E-09	3.77E-09	0.1372	0.3136	2.53E-09	1.68E-09		
F12	0.2511	0.1382	0.4466	0.1802	0.2282	0.103		
F13	0.0026	0.0013	0.004	0.0052	0.003	0.003		
F14	- 8.0451	1.852	- 4.7983	1.5991	- 6.7176	1.3963		

The underlined entries used to highlight the best obtained results

Table 5 Comparison of p -values of the Wilcoxon ranksum test obtained using SSA and different version of CSSA over multimodal benchmark functions

F	SSA vs. CSSA1	SSA vs. CSSA2	SSA vs. CSSA3	SSA vs. CSSA4	SSA vs. CSSA5
F8	< 0.05	0.7337	< 0.05	0.9097	0.9698
F9	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F10	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F11	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F12	0.2147	5.32E-01	4.55E-01	< 0.05	7.54E-01
F13	< 0.05	1.62E-01	1.41E-01	< 0.05	< 0.05
F14	< 0.05	8.21E-01	< 0.05	< 0.05	< 0.05
	SSA vs. CSSA6	SSA vs. CSSA7	SSA vs. CSSA8	SSA vs. CSSA9	SSA vs. CSSA10
F8	0.273	< 0.05	0.7913	< 0.05	0.5708
F9	< 0.05	< 0.05	< 0.05	0.6232	< 0.05
F10	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F11	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
F12	< 0.05	< 0.05	< 0.05	0.9097	0.1545
F13	< 0.05	< 0.05	< 0.05	0.1041	< 0.05
F14	9.85E-01	< 0.05	< 0.05	0.5476	< 0.05

Figure 1a shows the mathematical representation of three unimodal benchmark functions including $F1$, $F3$ and $F7$ and three multimodal benchmark functions including $F9$, $F13$ and $F14$. Also, Fig. 1 the search history, the trajectory of the first salp in the first dimension and the average fitness of all group of CSSA with logistic chaotic map. As it can be observed from Fig. 1b, the salps positions are scattered around the optimal solution in the search space during the course of iterations. Also, from Fig. 1c, it can be observed that salp individuals faced abrupt changes of their trajectories in the initial iterations. After that, these abrupt changes are gradually decreased during the optimization process. Based on this behavior, a population-based algorithm can guarantee to converge to some point and search locally in a search space [43]. Figure 1d shows another evidence of the significant improvement and fast convergence rate of salps individuals. As it can be observed, average fitness values are gradually decreased. From this behavior, it can be concluded the improvement of the fitness value of all individuals through iterations. This is due to a proper balance between exploration and exploitation phases. In Fig. 1e, the convergence curves of original SSA and the proposed CSSA is drawn. As it can be observed, CSSA converges faster than original SSA. Also, the proposed CSSA obtains better results than SSA. This can be an evidence of the superiority of the proposed CSSA in terms of the stability and the performance.

4.2 CSSA for feature selection

In this subsection, the performance of CSSA as feature selection algorithm is evaluated and compared with other

meta-heuristic feature selection algorithms. All these algorithms are applied to twenty benchmark datasets with different types. The used datasets are collected from UCI machine learning repository [44]. Table 6 shows a brief description of each used dataset. From this table, it can be observed that some of the adopted datasets contain missing values in their records. In this work, the median method is used to solve this problem by replacing the missing value with the median value of known values of a feature given class. Equation (10) shows the mathematical representation of the median method, where $Q_{i,j}$ is the missing value at i - th iteration given j - th feature for class C . However, in case of missing categorical values, the most repeated value is replaced with the missing value.

$$\overline{Q}_{i,j} = \text{median}_{i:Q_{i,j} \in C_r} Q_{i,j} \quad (10)$$

Also, four evaluation criteria are used in this work. These criteria are mean, standard deviation (SD), average feature subset length, p -values of the Wilcoxon rank sum test and convergence curve.

Table 8 shows the statistical results of the proposed CSSA with logistic chaotic map and five other meta-heuristic algorithms, previously proposed in the literature as feature selection algorithms. The optimal results are underlined. These related algorithms are artificial bee colony (ABC) [45], moth flame optimization (MFO) [28], particle swarm optimization (PSO) [46], chicken swarm optimization (CSO) [47], gray wolf optimizer (GWO) [48], bird swarm algorithm [49], crow search algorithm [7], whale optimization algorithm [50] and sine cosine

algorithm [51]. In all the experiments, we used the same fitness function with same population size, number of dimensions, search boundary and the maximum number of iteration. Moreover, all the algorithms start with same

initial population. The reason behind this, we would like to make a fair comparison. Table 7 shows the rest of parameters set for the competitor algorithms. In this table, the best results are underlined. As it can be observed from

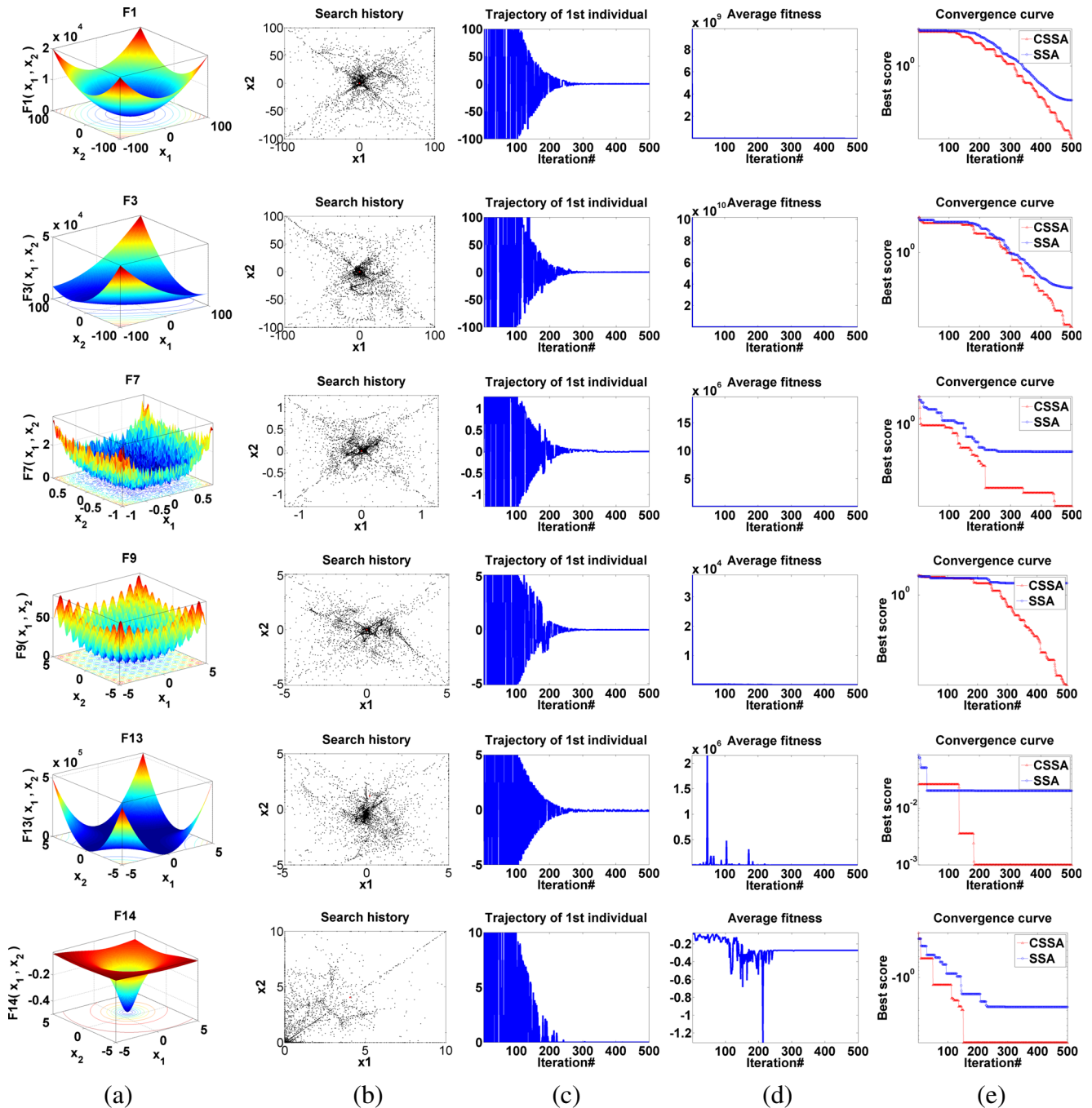


Fig. 1 **a** Graphical representations of benchmark mathematical functions, **b** Search history of CSSA, **c** trajectory of CSSA in first

dimension, **d** the average fitness of all group of CSSA, and **e** the convergence curve of CSSA and SSA

Table 6 Datasets description

ID	Dataset	No. of attributes	No. of instances	Missing values	Type
D1	Waveform	21	5000	No	Physical
D2	Zoo	18	101	No	Life
D3	Credit approval	15	690	Yes	Business
D4	Cylinder bands	40	512	Yes	Physical
D5	Mice protein expression	82	1080	Yes	Clinical
D6	Parkinson's disease detection	23	197	No	Clinical
D7	Thoracic surgery	17	470	No	Clinical
D8	Indian liver patient	10	583	No	Clinical
D9	Germen credit	24	1000	No	Business
D10	Letter recognition	16	50000	No	Computer
D11	Wisconsin diagnosis breast cancer	32	596	No	Clinical
D12	Cardiotocography	23	2126	No	Clinical
D13	Hepatitis	19	155	Yes	Clinical
D14	Ionosphere	351	34	No	Physical
D15	Sonar	60	208	No	Physical
D16	Arrhythmia	279	452	Yes	Life
D17	Lung cancer	56	32	Yes	Clinical
D18	Statlog heart	13	270	No	Clinical
D19	Glass identification	10	214	No	Physical
D20	Chess	36	3196	No	Game

Table 7 Meta-heuristic algorithms parameters settings

Algorithm	Parameter	Value
PSO	An inertial weight	1
	A inertia weight damping ratio	0.9
	Personal learning coefficient	1.5
	Global learning coefficient	2.0
ABC	A number of colony size	10
	A number of food source	5
	A number of limit trials	5
CSO	A number of chicken updated	10
	The percent of roosters population size	0.15
	The percent of hens population size	0.7
	The percent of mother hens population size	0.05
MFO	a	− 1
	b	1
BSA	The frequency of birds flight behaviors	10
	A Cognitive accelerated coefficient	1.5
	A Social accelerated coefficient	1.5
	The indirect and direct parameters	1
CSA	Awareness probability	0.1
	Flight length	2
SCA	a	− 1
WOA	a	2
	a	− 1
GWO	b	2
	a	2

Table 8 Comparison of statistical results obtained using CSSA and most recent and popular meta-heuristics feature selection algorithms for 20 Datasets

	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
	D1		D2		D3		D4		D5	
CSSA	<u>0.7664</u>	<u>0.0029</u>	0.8815	0.0081	0.8291	0.0066	<u>0.7237</u>	<u>0.0127</u>	<u>0.9827</u>	<u>0.0134</u>
SSA	0.7571	0.0057	0.8737	0.0091	0.8082	0.0088	0.7235	0.0119	0.9494	0.0077
ABC	0.7395	0.005	0.8816	0.0101	0.8159	0.0046	0.6968	0.0119	0.8895	0.005
MFO	0.7455	0.0055	0.8746	0.0056	0.8147	0.0053	0.7195	0.0088	0.927	0.003
CSO	0.7291	0.0037	0.875	0.0062	0.8018	0.007	0.7098	0.0092	0.9216	0.0054
GWO	0.7615	0.0047	0.8825	0.0097	0.8206	0.0081	0.6993	0.0073	0.8959	0.006
PSO	0.7658	0.0081	0.8746	0.0075	0.8288	0.0061	0.7064	0.0133	0.9252	0.0076
BSA	0.7595	0.0046	0.884	0.0126	0.8253	0.0098	0.7145	0.0093	0.9136	0.0176
CSA	0.7645	0.0025	0.8813	0.0051	<u>0.8292</u>	<u>0.0032</u>	0.7208	0.0054	0.886	0.0044
SCA	0.7331	0.0049	0.873	0.0058	0.8043	0.01	0.7029	0.0091	0.9082	0.0039
WOA	0.7612	0.0108	<u>0.8868</u>	<u>0.0081</u>	0.8178	0.012	0.7235	0.0206	0.9744	0.0132
	D6		D7		D8		D9		D10	
CSSA	0.8982	0.0047	0.9711	0.0061	<u>0.7276</u>	<u>0.0026</u>	<u>0.7639</u>	<u>0.0017</u>	0.8184	<u>0.0063</u>
SSA	0.8821	4.90E-03	0.9207	0.0118	0.7123	0.0053	0.7309	0.007	0.7967	0.0107
ABC	0.8487	0.0094	0.9414	0.0172	0.7124	0.0137	0.687	0.0157	0.7885	0.016
MFO	0.8677	0.0124	0.918	0.0142	0.7157	0.0032	0.7112	0.0102	0.7861	0.0033
CSO	0.8426	0.0101	0.9269	0.0124	0.6927	0.0036	0.716	0.011	0.7525	0.0088
GWO	0.8345	0.0063	0.9168	0.0067	0.6848	0.0113	0.6872	0.0059	0.8159	0.0072
PSO	0.8514	0.02	0.9311	0.0093	0.7087	0.0085	0.6915	0.0074	0.7860	0.0072
BSA	0.8735	0.0212	0.9186	0.0217	0.7104	0.0151	0.7254	0.0232	0.7554	0.0252
CSA	0.8247	0.0108	0.9004	0.0065	0.6669	0.0084	0.6678	0.0055	0.8076	0.0086
SCA	0.8327	0.0096	0.8985	0.0067	0.6963	0.0096	0.6954	0.0045	0.7833	0.0024
WOA	<u>0.899</u>	<u>0.0174</u>	<u>0.9733</u>	<u>0.0149</u>	0.7276	0.0056	0.7635	0.0049	0.7658	0.0178
	D11		D12		D13		D14		D15	
CSSA	0.8847	0.0016	<u>0.8904</u>	<u>0.0019</u>	1.0162	0	<u>0.9278</u>	<u>0.0022</u>	<u>0.9894</u>	<u>0.0012</u>
SSA	0.8802	0.0113	0.884	0.0115	0.9667	0.0197	0.8912	0.0085	0.955	0.0027
ABC	0.8661	0.0094	0.8649	0.0057	0.8843	0.0099	0.8453	0.0163	0.9603	0.0156
MFO	0.8773	0.0075	0.8714	0.0071	0.9274	0.0172	0.8841	0.0079	0.955	0.0027
CSO	0.8781	0.0015	0.8769	0.0063	0.8984	0.032	0.8917	0.0078	0.9512	0.0176
GWO	0.8559	0.0056	0.8484	0.0031	0.8984	0.0117	0.8526	0.0065	0.9368	0.0027
PSO	0.8588	0.0107	0.8527	0.0126	0.9003	0.0126	0.8594	0.0092	0.9478	0.0075
BSA	0.8583	0.0109	0.8702	0.0079	0.9705	0.0389	0.8848	0.0163	0.9387	0.0014
CSA	0.8395	0.0077	0.8419	0.0052	0.8585	0.0059	0.835	0.0082	0.9186	0.0027
SCA	0.8604	0.0116	0.8557	0.0038	0.9141	0.014	0.8638	0.0051	0.944	0.0075
WOA	<u>0.8879</u>	<u>0.0019</u>	0.8877	0.0051	<u>1.0171</u>	<u>0.0043</u>	0.9186	0.0063	0.9875	0.0027
	D16		D17		D18		D19		D20	
CSSA	<u>1.0172</u>	<u>0</u>	<u>1.0499</u>	<u>0</u>	<u>0.7446</u>	<u>2.02E-02</u>	0.7038	0.0025	0.8463	0.0024
SSA	0.8513	0.0163	0.9142	0.0318	0.7225	0.0027	0.6705	0.0083	0.8292	0.0123
ABC	0.7525	0.0096	0.6961	0.0264	0.7167	0.0087	0.6876	0.0048	0.8099	0.0087
MFO	0.8054	0.0086	0.7921	0.0443	0.7198	0.0083	0.6995	0.002	0.8147	0.0031
CSO	0.8009	0.0061	0.8204	0.0271	0.7142	0.0132	0.6842	0.0075	0.8044	0.0521
GWO	0.762	0.0072	0.7121	0.0194	0.7117	0.0033	<u>0.7104</u>	<u>0.0124</u>	0.8175	0.0616
PSO	0.7782	0.0187	0.7649	0.04	0.7208	0.0021	0.6795	0.0092	0.8513	0.0286
BSA	0.8546	0.0287	0.9483	0.0999	0.7242	0.0103	0.6962	0.0089	0.7949	0.0173
CSA	0.7503	0.0037	0.6782	0.0245	0.6992	0.0079	0.6825	0.0132	0.8315	0.016
SCA	0.7841	0.0105	0.7451	0.0145	0.7292	0.0057	0.693	0.0137	0.8434	0.0903
WOA	1.0172	0	1.0499	0	0.7198	0.0064	0.692	0.0139	<u>0.8594</u>	<u>0.0034</u>

The underlined entries used to highlight the best obtained results

Table 9 The best controller parameters for two case studies

		k_p	k_i	L	k_d	m	J
Case Study 1	Angle	167	5	0.37	38	0.29	1.0054
	Control						
	Position	59	19	0.23	38	0.18	
Case Study 2	Angle	202	18	0.15	31	0.11	1.4364
	Control						
	Position	122	3	0.36	28	0.2	
	Control						

Table 8, the proposed CSSA feature selection algorithm is very competitive algorithm. CSSA obtains the highest mean fitness value for 12 benchmark datasets out of 20, with WOA in the second place. WOA obtains the top results for 6 benchmark datasets out of 20. However, it can be seen that embedding chaos theory in the searching mechanism of SSA achieves the higher mean fitness values with higher stability compared with the original SSA and the other meta-heuristic algorithms. Also, it can be observed that in all cases, CSSA can find better feature subset, which maximizes the classification accuracy, while minimizing the length of the selected feature subset.

Moreover, it can be observed that ABC and PSO in most cases obtain the worst results. The reason behind this can be referred to (1) improper balancing between exploitation and exploration and (2) improper setting of parameters values. Also, as we mentioned before, meta-heuristic algorithms are stochastic. Thus they have random parameters, which have a great influence on their performance. These random parameters can lead agents such as bee and moth to be stuck at local optima with high probability. Therefore, these algorithms such as PSO and ABC need to be tuned first to get better results. Several studies have been proposed in literature to improve the performance of these algorithms such as in [52–54]. Authors in [55] proposed a new initialization method and new updating mechanism of PSO. They applied their modified PSO based their new initialization and updating mechanism on feature selection problem. The results showed that their proposed algorithm can overcome the drawbacks of original PSO. Moreover, the results showed that the modified PSO obtain better feature subset compared with original PSO, which maximizing the classification accuracy while minimizing the number of selected features. Authors in [56] proposed a modified version of PSO, where an adaptive factor, namely f_a is adopted to avoid the local optima and prevent the premature convergence. The authors applied

their modified particle swarm optimization (MPSO) on fractional-order proportional integral derivative (FOPID) controller. Table (9) shows the best controller parameters tuned from MPSO for two control design case studies. The description for both cases are described in [56]. Also, the authors compared their algorithm with the original PSO. The results prove the stability and robustness of the MPSO based FOPID controller regarding tracking errors for both smooth and a non smooth cart trajectory.

Authors in [57] applied their modified version of PSO on multilevel thresholding problem. They used two-dimensional (2D) Kullback–Leibler(K–L) divergence as the fitness function to find the optimal thresholds. Figure 2 summarized the mean fitness values obtained for number of segmentation levels equal to 3 and 5 in terms of Boundary Displacement Error (BDE), Probability Rand Index (PRI), Global Consistency Error (GCE) and Variation of Information (VOI). The authors adopted Berkeley Segmentation Dataset and Benchmark (BSDS300) in their experiments. The simulation results showed that their

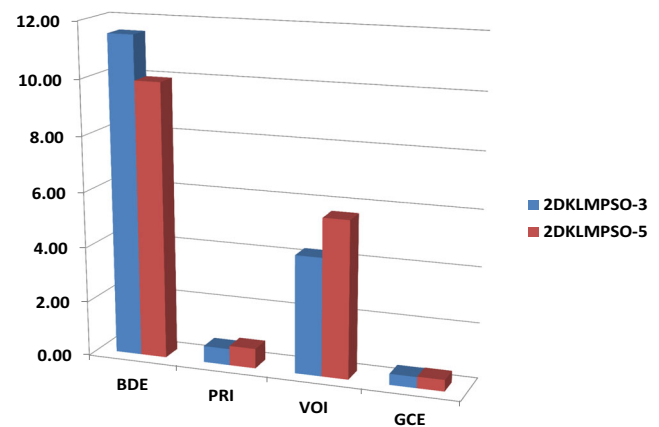
**Fig. 2** Average performance indices of 2DKLMPSO over BSDS300 with 3-level and 5-level segmentation

Table 10 CSSA vs other meta-heuristics feature selection algorithms for 20 Datasets in terms of average feature vector length

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
CSSA	0.4750	0.4689	<u>0.3386</u>	0.5194	0.3324	<u>0.1968</u>	0.3437	0.3815	0.3087	0.3391
SSA	0.4967	0.4978	0.3743	0.5657	0.3524	0.2683	0.4083	0.4889	0.3594	0.3667
ABC	0.6254	0.5000	0.4089	0.5954	0.5646	0.5121	0.3745	0.3700	0.4819	0.4422
MFO	0.5476	0.5250	0.4511	0.5148	0.4138	0.5167	0.4647	<u>0.3300</u>	0.3639	0.3933
CSO	0.6190	0.4833	0.4067	0.4204	0.3404	0.3697	0.4510	0.3367	0.3750	<u>0.3633</u>
GWO	0.4619	0.4792	0.4733	0.5407	0.5458	0.5348	0.4196	0.4633	0.3875	0.4344
PSO	0.4397	0.5000	0.3911	0.4583	0.3738	0.3727	0.4451	0.3633	0.5653	0.4267
BSA	0.5238	<u>0.4067</u>	0.3467	0.5213	0.3404	0.5364	0.4961	0.3567	<u>0.3042</u>	0.3967
CSA	<u>0.4302</u>	0.5500	0.4400	<u>0.3426</u>	0.6150	0.5909	0.6196	0.5000	0.5569	0.6000
SCA	0.6190	0.4604	0.5156	0.5815	0.4842	0.5470	0.4882	0.3400	0.5097	0.3900
WOA	0.4857	0.4125	0.3644	0.4056	<u>0.1150</u>	0.2348	<u>0.2510</u>	0.4100	0.3742	0.3900
	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20
CSSA	<u>0.2748</u>	0.3238	0.3877	0.3702	0.2196	0.4682	0.3374	<u>0.0234</u>	0.5708	0.3095
SSA	0.2943	0.3373	<u>0.3860</u>	0.3535	0.2220	0.5214	0.3738	0.0438	0.5750	0.3190
ABC	0.3621	0.4121	0.5317	0.5029	0.4567	0.6030	0.6164	0.0400	0.5630	0.3792
MFO	0.4803	0.4061	0.4850	0.4059	0.3867	0.4178	0.5287	0.0383	0.5296	0.3875
CSO	0.3576	0.3606	0.4067	<u>0.3520</u>	0.4267	0.3941	0.3661	0.0547	0.4926	0.3917
GWO	0.4394	0.4182	0.5433	0.4657	0.4033	0.5652	0.6895	0.0208	0.4815	0.3917
PSO	0.4545	0.3545	0.4583	0.4569	0.4000	0.4807	0.6140	0.0393	0.3889	<u>0.2500</u>
BSA	0.4606	0.4318	0.5133	0.2735	0.4667	<u>0.2674</u>	0.2158	0.0239	<u>0.3370</u>	0.4708
CSA	0.6106	0.5455	0.6050	0.5863	0.6000	0.6474	0.6140	0.0377	0.5556	0.4417
SCA	0.4652	0.4636	0.4117	0.4235	<u>0.2102</u>	0.5756	<u>0.2148</u>	0.0641	0.5593	0.3917
WOA	0.2985	0.2652	0.3917	0.4186	0.2167	0.5333	0.4910	0.0463	0.5519	0.2542

The underlined entries used to highlight the best obtained results

Table 11 CSSA vs other meta-heuristics feature selection algorithms for 20 Datasets in terms of p -values of the Wilcoxon ranksum test

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10
CSSA vs. SSA	0.6905	0.212	< 0.05	0.8235	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
CSSA vs. ABC	< 0.05	0.8497	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.1508	< 0.05	< 0.05
CSSA vs. MFO	< 0.05	0.0597	0.2778	0.6905	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
CSSA vs. CSO	< 0.05	0.1009	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
CSSA vs. GWO	0.8951	0.5689	0.6905	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
CSSA vs. PSO	0.1508	0.1173	0.0952	0.0952	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
CSSA vs. BSA	0.6905	0.6485	< 0.05	0.2222	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	0.0952
CSSA vs. CSA	0.4206	0.8466	< 0.05	0.5476	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
CSSA vs. SCA	< 0.05	0.0196	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
CSSA vs. WOA	0.8413	0.1444	0.8413	0.5253	0.4413	0.1349	0.8932	0.9524	0.2857	0.8413
	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20
CSSA vs. SSA	0.254	0.5476	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
CSSA vs. ABC	< 0.05	< 0.05	< 0.05	< 0.05	0.3333	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
CSSA vs. MFO	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
CSSA vs. CSO	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
CSSA vs. GWO	< 0.05	< 0.05	< 0.05	< 0.05	< 0.06	< 0.05	< 0.05	< 0.05	0.5952	< 0.05
CSSA vs. PSO	< 0.05	< 0.05	< 0.05	< 0.05	< 0.07	< 0.05	< 0.05	< 0.05	< 0.05	0.5239
CSSA vs. BSA	< 0.05	< 0.05	< 0.05	< 0.05	< 0.08	< 0.05	< 0.05	< 0.05	0.2511	0.4564
CSSA vs. CSA	< 0.05	< 0.05	< 0.05	< 0.05	< 0.09	< 0.05	< 0.05	< 0.05	< 0.05	< 0.05
CSSA vs. SCA	< 0.05	< 0.05	< 0.05	< 0.05	< 0.10	< 0.05	< 0.05	< 0.05	0.4603	0.8672
CSSA vs. WOA	< 0.05	0.3095	0.8965	< 0.05	0.9921	0.9824	0.8845	< 0.05	0.254	0.6667

Fig. 3 Performance comparison on D1 to D20

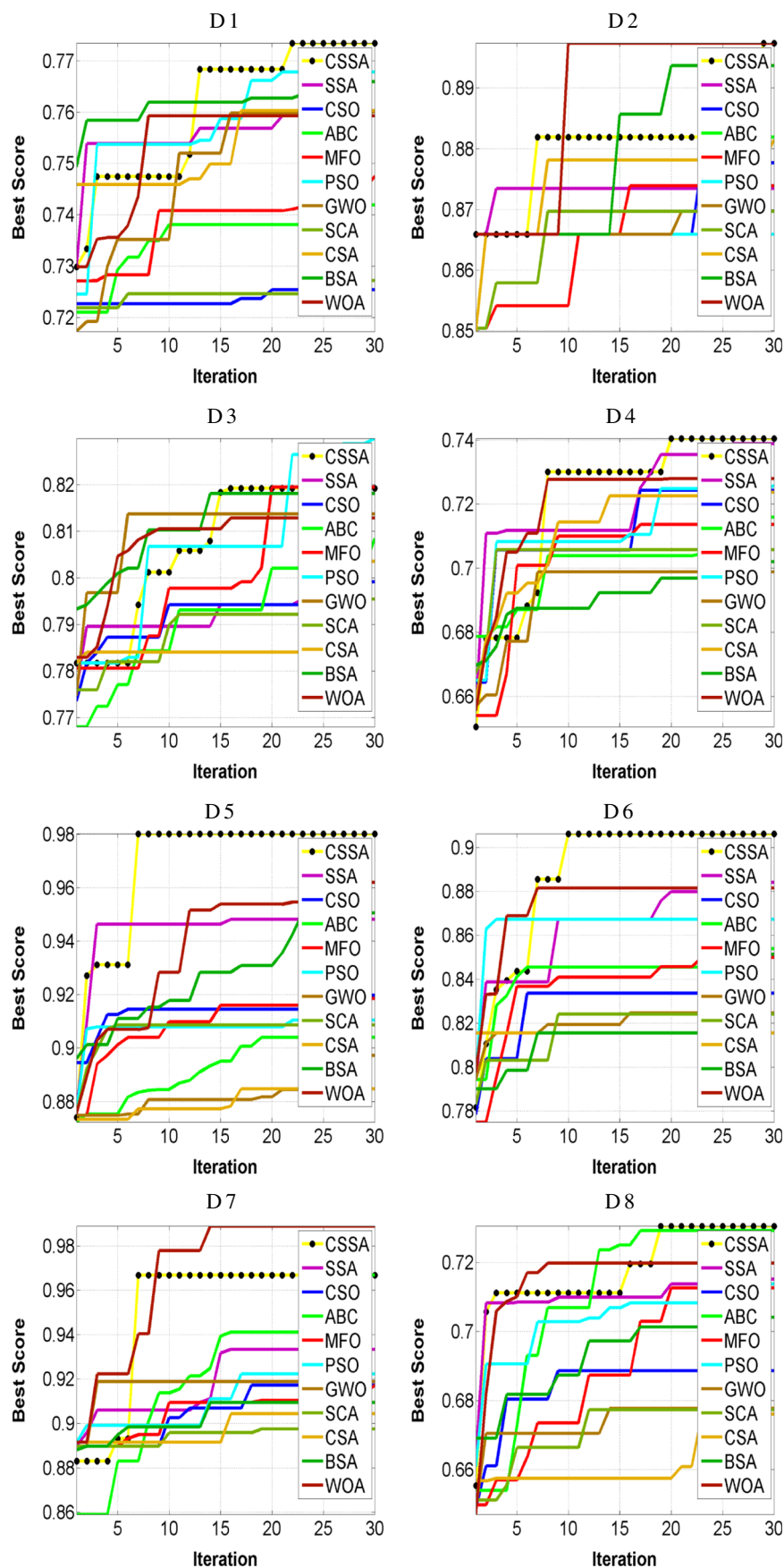


Fig. 3 (continued)

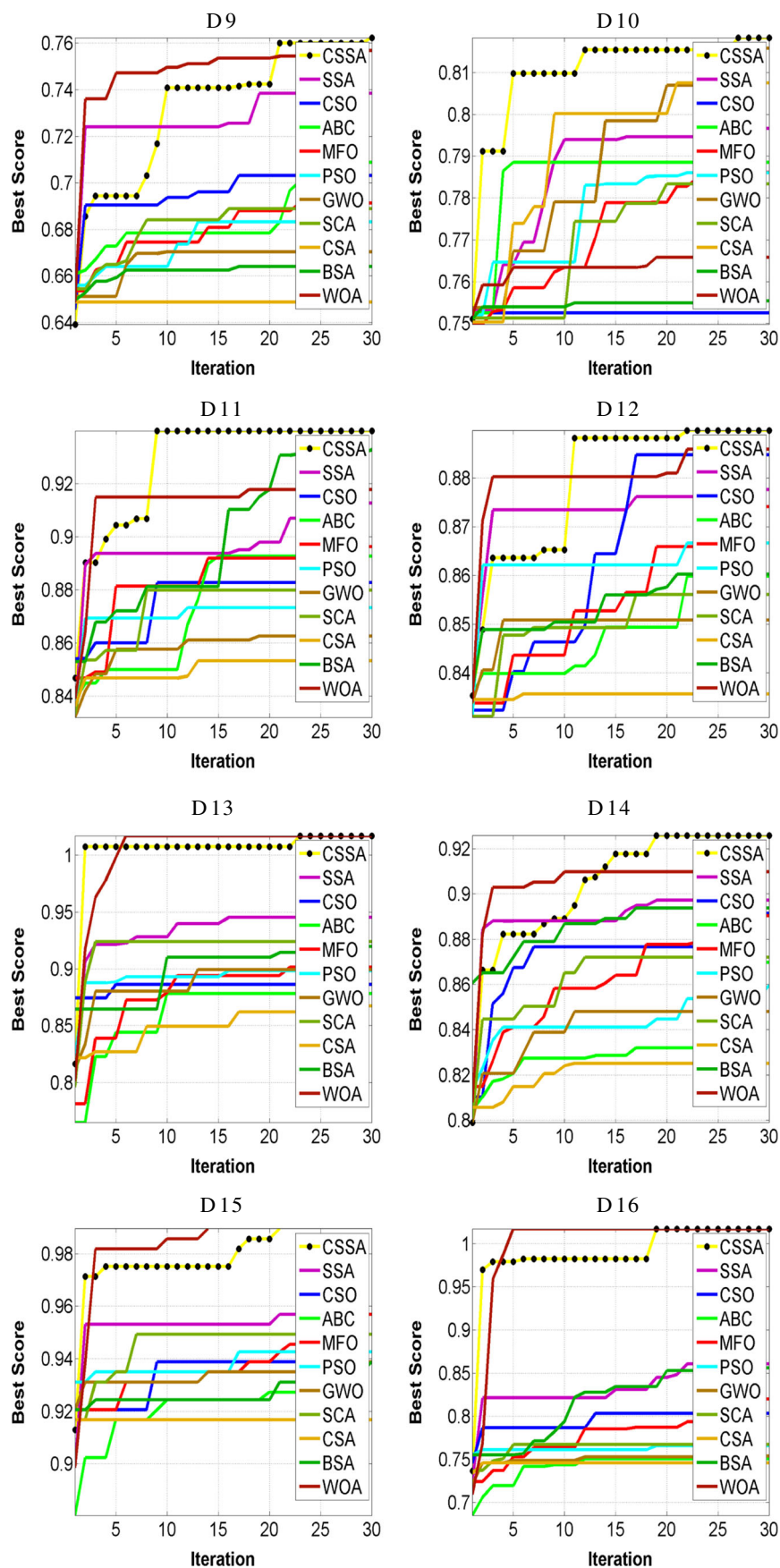
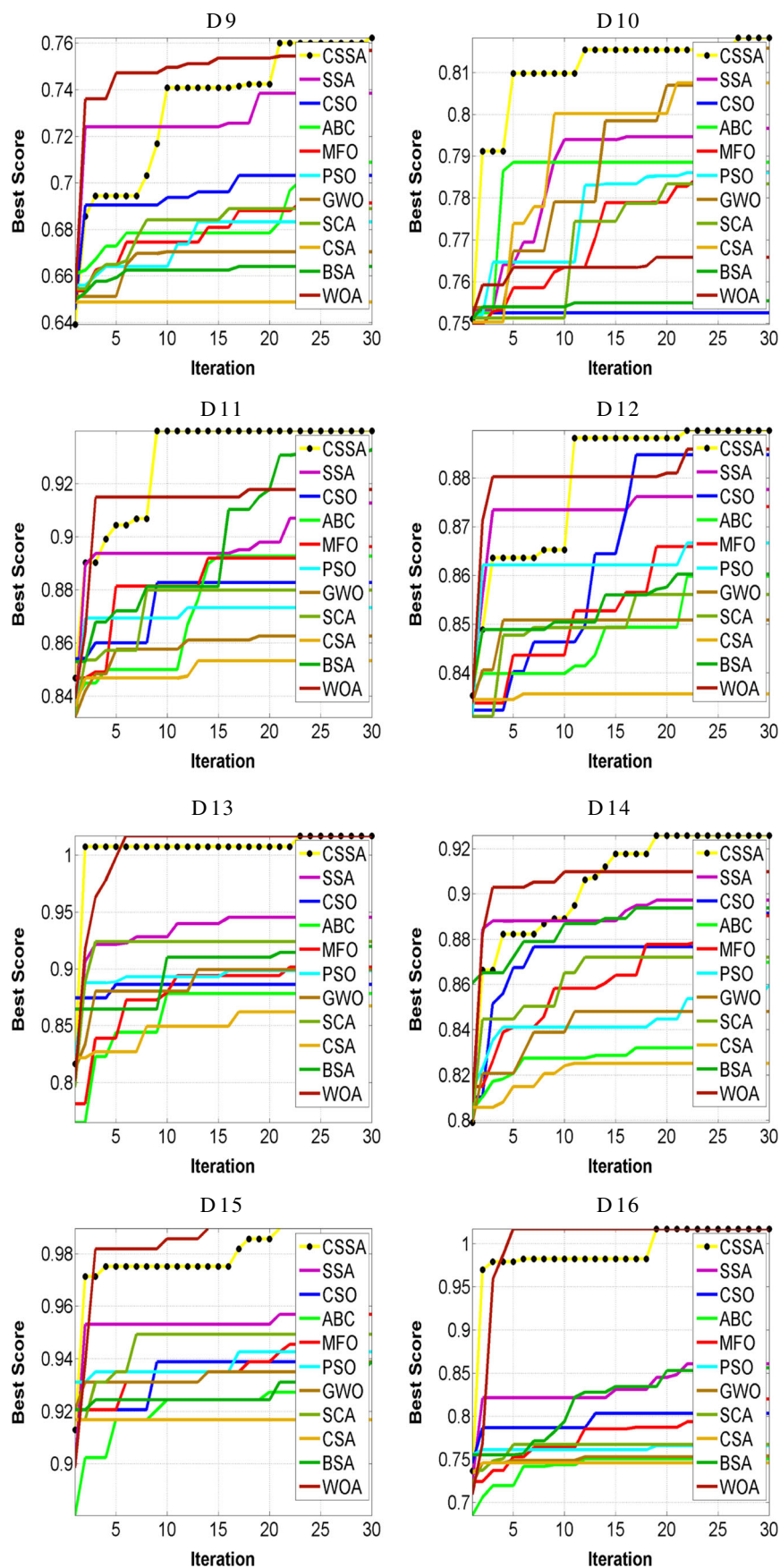


Fig. 3 (continued)



hybrid 2D K-L divergence with MPSO (2DKLMPSO) can efficiently overcome the premature convergence of PSO and can reduce the time complexity of multilevel thresholding segmentation. In this work, the authors used the default parameters setting of all competitor algorithms, as we are mainly focused on improving the performance of only SSA.

Table 10 compares the length of the selected feature subset on average 30 independent runs. Equation (11) shows the mathematical formulation of average feature subset length (ASS), where y_{best} is the best position obtained so far at i -th iteration and L is the number of features in the original dataset.

$$ASS = \frac{1}{tMax} \sum_{i=1}^{tMax} \frac{length(y_{best}^i)}{L} \quad (11)$$

As it can be observed from Table 10, CSSA yields the minimum feature subset on four cases, however in the rest of cases, it takes place on second or third. From the obtained results at Table 10, it can be observed that the proposed CSSA is very competitive algorithm compared with other popular and most recent feature selection algorithm. Table 11 shows p -values of the Wilcoxon rank sum test of CSSA with other meta-heuristic feature selection algorithms for twenty benchmark datasets. As it can be observed, the proposed CSSA is statistically significant compared with the other algorithms. This is because most of the obtained results are less than 0.05. Thus, it can be considered a sufficient evidence against the null hypothesis.

For further evaluation of the proposed CSSA with logistic chaotic map, the graphical representation of the convergence curve for single run is drawn for the twenty benchmark datasets as shown in Fig. 3. In this figure, the proposed CSSA is marked with black circles on the yellow line. As it can be observed, the proposed CSSA is superior. As it obtained the highest fitness value for most of cases. In case of the rest of cases, WOA In addition, it can be observed that the proposed CSSA converges faster than the other algorithms toward the optimal results. Moreover, it can be obviously noticed that the proposed CSSA always obtains better results compare with the original SSA. Such improvement of the performance came from embedding

chaotic sequence in optimization process of SSA, which highly impact on the performance of SSA.

5 Conclusion

In this paper, a novel hybridization approach is proposed based on SSA and chaos theory. The effectiveness of ten chaotic maps is analyzed and compared regarding enhancing the performance of the original SSA. Fourteen benchmark global optimization functions and twenty benchmark datasets were employed. The experimental results show that the proposed chaotic version of SSA can significantly boost the performance of SSA in terms of both exploitation and exploration. Moreover, the results suggest that logistic chaotic map is the optimal map. Another finding of this paper was that embedding chaotic maps in optimization process of SSA can find optimal feature subset, which maximizes the classification accuracy, while minimizes the number of selected features. Also, the performance of CSSA as feature selection algorithm is compared with ten other meta-heuristic feature selection algorithms. These algorithms are SSA, ABC, PSO, GWO, MFO, CSO, BSA, CSA, WOA and SCA. The results show the superiority of the proposed CSSA algorithm.

In the future, more complex science and real-world problems will be considered for evaluating CSSA. Additionally, more chaotic maps are also worth employed to CSSA.

Appendix A: List of benchmark functions

Table 12 Definition of unimodal benchmark functions

Function	Definition
$F1$	$f_1(x) = \sum_{i=1}^n x_i^2$
$F2$	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $
$F3$	$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$
$F4$	$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$
$F5$	$f_5(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
$F6$	$f_6(x) = \sum_{i=1}^n i x_i^2$
$F7$	$f_7(x) = \sum_{i=1}^n i x_i^4 + U(0, 1)$

Table 13 Definition of multimodal benchmark functions

Function	Definition
F_8	$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$
F_9	$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10n]$
F_{10}	$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + \epsilon$
F_{11}	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
F_{12}	$f_{12}(x) = \frac{\pi}{n} \{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_i + 1)] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, 10, 100, 4) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$
F_{13}	$f_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$
F_{14}	$f_{21}(x) = -\sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$

Table 14 Properties of unimodal benchmark functions, lb denotes lower bound, ub denotes upper bound, dim denotes dimensions, opt denotes optimum point

No.	lb	ub	opt	dim
F_1	-100	100	0	10
F_2	-10	10	0	10
F_3	-100	100	0	10
F_4	-100	100	0	10
F_5	-30	30	0	10
F_6	-5.12	5.12	0	10
F_7	-1.28	1.28	0	10

Table 15 Properties of multimodal benchmark functions, lb denotes lower bound, ub denotes upper bound, dim denotes dimensions, opt denotes optimum point

No.	lb	ub	opt	dim
F_8	-500	500	-418.9829 * 50	10
F_9	-5.12	5.12	0	10
F_{10}	-32	32	0	10
F_{11}	-600	600	0	10
F_{12}	-50	50	0	10
F_{13}	-5	5	0	4
F_{14}	0	10	-10.1532	4

References

- Kennedy J, Eberhart R (1995) Particle swarm optimization. In: IEEE international conference on neural networks, vol 4, pp 1942–1948
- Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *J Glob Optim* 39:459–530
- Colnari A, Dorigo M, Maniezzo V (1991) Distributed optimization by ant colonies. In: Proceedings of the first european conference on artificial life, pp 134–176
- Hong Y, Angelo A, Beltran J, Paglinawan A (2016) A chaos-enhanced particle swarm optimization with adaptive parameters and its application in maximum power point tracking. *Math Probl Eng* 2016:19
- Li B, Jiang W (1998) Optimizing complex functions by chaos search. *Journal of Cybernetics and Systems* 29:409–419
- Saremi S, Mirjalili S, Lewis A (2014) Biogeography-based optimization with chaos. *Neural Comput & Applic* 25(5):1077–1097
- Sayed G, Hassanien A, Azar A (2017) Feature selection via a novel chaotic crow search algorithm. *Neural Comput & Applic*:1–18. <https://doi.org/10.1007/s00521-017-2988-6>
- Coelho L, Mariani V (2008) Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. *Expert System with Application* 34(3):1905–1918
- Zhang L, Zhang CJ (2008) Hopf bifurcation analysis of some hyperchaotic systems with time-delay controllers. *Kybernetika* 44(1):35–42
- Zhu ZL, Li SP, Yu H (2008) A new approach to generalized chaos synchronization based on the stability of the error system. *Kybernetika* 44(4):492–500
- Yang DX, Li G, Cheng GD (2007) On the efficiency of chaos optimization algorithms for global optimization. *Chaos Solitons Fractals* 34:1366–1375
- Tavazoei MS, Haeri M (2007) Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. *Appl Math Comput* 187:1076–1085
- Lassoued A, Boubaker O (2016) On new chaotic and hyperchaotic systems: a literature survey. *Nonlinear Analysis: Modelling and Control* 21(6):770–789
- Mahmoud E, Abood F (2017) A new nonlinear chaotic complex model and its complex antilag synchronization. *Complexity* 2017:1–13
- Lassoued A, Boubaker O (2017) Dynamic analysis and circuit design of a novel hyperchaotic system with fractional-order terms. *Complexity* 2017:1–10
- Wang X, Akgul A, Kacar S, Pham V (2017) Multimedia security application of a ten-term chaotic system without equilibrium. *Complexity* 2017:1–10

17. Abdullah A, Enayatifa R, Lee M (2012) A hybrid genetic algorithm and chaotic function model for image encryption. *Journal of Electronics and Communication* 66(1):806–816
18. Pan Q, Wang L, Gao L (2011) A chaotic harmony search algorithm for the flow shop scheduling problem with limited buffers. *Appl Soft Comput* 11:5270–5280
19. Zawbaa H, Emary E, Grosan C (2016) Feature selection via chaotic antlion optimization. *PloS ONE* 11(3):1–21. e0150652. <https://doi.org/10.1371/journal.pone.0150652>
20. Chen CH (2014) A hybrid intelligent model of analyzing clinical breast cancer data using clustering techniques with feature selection. *Appl Soft Comput* 20:4–14
21. Rui Y, Huang TS, Chang S (1999) Image retrieval: current techniques, promising directions and open issues. *J Vis Commun Image Represent* 10:39–62
22. Yang Y, Pederson JO (1997) A comparative study on feature selection in text categorization. In: *Proceedings of the fourteenth international conference on machine learning*, pp 412–420
23. Ng K, Liu H (2000) Customer retention via data mining. *AI Review* 14:569–590
24. Ben-Dor A (2000) Tissue classification with gene expression profiles. *J Comput Biol* 7:559–583
25. Golub TR (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286:531–537
26. Yu Z (2014) Hybrid clustering solution selection strategy. *Pattern Recogn* 47:3362–3375
27. Guyon I, Elisseeff A (2003) An introduction to variable and attribute selection. *Machine Learning Research* 3:1157–1182
28. Sayed G, Hassanien A (2017) Moth-flame swarm optimization with neutrosophic sets for automatic mitosis detection in breast cancer histology images. *Appl Intell* 47(2):397–408
29. Emary E, Zawbaa H, Hassanien A (2016) Binary gray wolf optimization approaches for feature selection. *Neurocomputing* 172:371–381
30. Lin S, Shih-Chieh Ying K, Lee Z (2008) Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications* 35(4):1817–1824
31. Schiezero M, Pedrini H (2013) Data feature selection based on artificial bee colony algorithm. *EURASIP Journal on Image and Video Processing* 2013(1):1–8
32. Seyedali M, Gandomi A, Mirjalili S, Saremi S, Faris H, Mirjalili S (2017) Salp swarm a bio-inspired optimizer for engineering design problems. *Adv Eng Softw* 114:163–191
33. Gandomia AH, Yangb XH (2014) Chaotic bat algorithm. *Journal of Computational Science* 5:224–232
34. Tavazoei MS, Haeri M (2007) An optimization algorithm based on chaotic behavior and fractal nature. *J Comput Appl Math* 2016(2):1070–1081
35. Wanga GG, Guo L, Gandomi AH, Hao GH, Wangb H (2014) Chaotic krill herd algorithm. *Inf Sci* 274:17–34
36. Zhang Q, Li Z, Zhou CJ, Wei XP (2013) Bayesian network structure learning based on the chaotic particle swarm optimization algorithm. *Genet Mol Res* 12(4):4468–4479
37. Saremi S, Mirjalili S, Lewis A (2014) Biogeography-based optimization with chaos. *Neural Comput & Applic* 25(5):1077–1097
38. Emary E, Zawbaa H, Hassanien A (2016) Binary gray wolf optimization approaches for feature selection. *Neurocomputing* 172:371–381
39. Sayed G, Darwish A, Hassanien A (2017) Quantum multiverse optimization algorithm for optimization problems. *Neural Comput & Applic*:1–18. <https://doi.org/10.1007/s00521-017-3228-9>
40. Digalakis J, Margaritis K (2001) On benchmarking functions for genetic algorithms. *Int J Comput Math* 77:481–506
41. Yang XS (2010) *Test problems in optimization*. Wiley, UK
42. Molga M, Smutnicki aC (2005) *Test functions for optimization needs*. Unpublished paper
43. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse imizer: a nature-inspired algorithm for global optimization. *Neural Comput & Applic* 27(2):1–19
44. Bache K, Lichman M Uci machine learning repository. <http://archive.ics.uci.edu/ml>. Retrieved November 12, 2017
45. Subanya B, Rajalaxmi RR (2014) Feature selection using artificial bee colony for cardiovascular disease classification. In: *International conference on electronics and communication systems (ICECS)*, pp 1–6
46. Wang X, Yang J, Teng X, Xia W, Jensen R (2007) Feature selection based on rough sets and particle swarm optimization. *Pattern Recogn Lett* 28(4):459–471
47. Hafez AI, Zawbaa HM, Emary E, Mahmoud HA, Hassanien AE (2015) An innovative approach for feature selection based on chicken swarm optimization. In: *The 7th international conference of soft computing and pattern recognition (soCPaR)*. Fukuoka, Japan, pp 19–24
48. Emary E, Zawbaa HM, Grosan C, Hassanien A (2015) Feature subset selection approach by gray-wolf optimization. Springer International Publishing, Cham, pp 1–13
49. Meng X, Gao X, Lu L, Liu Y, Zhang H (2016) A new bio-inspired optimisation algorithm: bird swarm algorithm. *Journal of Experimental & Theoretical Artificial Intelligence* 28(4):673–687
50. Canayaz M, Demir M (2017) Feature selection with the whale optimization algorithm and artificial neural network. In: *International artificial intelligence and data processing symposium (IDAP)*, pp 1–5
51. Hafez A, Zawbaa H, Emary E, Hassanien A (2016) Sine cosine optimization algorithm for feature selection. In: *International symposium on INnovations in intelligent systems and applications (INISTA)*, pp 1–5
52. Mahanipour A, Nezamabadi-pour H (2017) Improved pso-based feature construction algorithm using feature selection methods. In: *The 2nd conference on swarm intelligence and evolutionary computation (CSIEC)*, pp 1–5
53. Peng A, new bio-inspired optimisation algorithm A (2017) Bird swarm algorithm. *J Nanoelectron Optoelectron* 12(4):404–408
54. Yao B, Yan Q, Zhang M, Yang Y (2017) Improved artificial bee colony algorithm for vehicle routing problem with time windows. *PLoS ONE* 12(9):1–18
55. Xue B, Zhang M, Browne WN (2014) Particle swarm optimisation for feature selection in classification novel initialisation and updating mechanisms. *Applied Soft Computing* 18:261–276, 05
56. Mehdi H, Boubaker O (2017) The inverted pendulum in control theory and robotics: from theory to new innovations. Chapter Stabilization and tracking control of the inverted pendulum on a cart via a modified PSO fractional order PID controller. Tunisia, pp 93–115
57. Zhao X, Turk M, Li W, Lien K, Wang G (2016) A multi-level image thresholding segmentation algorithm based on two-dimensional k–l divergence and modified particle swarm optimization. *Appl Soft Comput* 48:151–159