Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 11, No. 2, May 2020), Pages: 1-15
www.jacr.iausari.ac.ir

Solving Data Clustering Problems using Chaos Embedded Cat Swarm Optimization

Farhad Ramezani✉
Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran
ramezani.farhad@iausari.ac.ir

Received: 2017/10/27; Accepted: 2018/01/07

Abstract

In this paper, a new method is proposed for solving the data clustering problem using Cat Swarm Optimization (CSO) algorithm based on chaotic behavior. The problem of data clustering is an important section in the field of the data mining, which has always been noted by researchers and experts in data mining for its numerous applications in solving real-world problems. The CSO algorithm is one of the latest meta-heuristic algorithms, which has a simple structure and it is easy to implement. The purpose of Chaos embedded Cat Swarm Optimization (CCSO) algorithm is to replace random values by chaotic ones to offer a stable algorithm that can allow for reaching the global optima to a large extent and improve the algorithm's convergence speed. The proposed algorithm has been compared to other heuristic algorithms on standard data sets from UCI repository, and the experimental results demonstrate that the proposed algorithm yields high performance for solving the data clustering problem.

Keywords: *Data Clustering, K-means, Cat Swarm Optimization, Chaos Theory*

1. Introduction

Data mining is one of the most important steps in knowledge discovery in the process of data bases, and it is known as an important subfield in knowledge management [1]. Aided by information technology systems and database-based software, organizations are now able to store an abundance of data. Data mining is a process that allows for extraction of meaningful information from this size of data. Data Clustering is one of the important sections in the field of the data mining science, which has always been noted by researchers and experts in data mining for its numerous applications in solving real-world problems [2]. The purpose of clustering is to division available data into several groups, such that data in different groups must be as different from each other as possible, and data in one group must be very similar to each other [3]. The clustering problem has applications in solving real-world problems [4,5] such as qualitative data interpretation and data compression, process monitoring, discovery of DNA clusters, coal classification, pattern-analysis, document retrieval, and image segmentation.

Since data clustering is an NP-Hard problem [6], most algorithms presented for data clustering are unable to solve the problem in a large scale. Therefore, performing research in the area can be valuable and practical. Cat Swarm Optimization (CSO) is one of the new swarm-based optimization algorithms, and it provides better performance compared to the other algorithms. However, CSO algorithm suffers from diversity and local optima problems. In this paper, a new algorithm has been proposed for solving the data clustering problem using Chaos embedded Cat Swarm Optimization (CCSO) algorithm. The purpose of the algorithm is to replace random values by chaotic ones to offer a stable algorithm that can allow for reaching the global optima to a large extent and improve the algorithm's convergence speed.

The rest of the paper is organized as follows: In Section 2, a literature review is presented. In Section 3, the proposed algorithm has been detailed. Section 4 is dedicated to the simulation and the results obtained from the implementation of the proposed algorithm. Finally, Section 5 includes a summary and the conclusion.

2. Literature Review

Data clustering is considered as one of the important tools in data mining, machine learning, and pattern classification problems [7]. The purpose of this unsupervised operation is to divide a group of heterogeneous data into separated groups. In particular the problem is stated as follows: given N objects, allocate each object to one of K clusters and minimize the sum of squared Euclidean distances between each object and the center of the cluster belonging to every such allocated object. The clustering problem minimizing is described as in：[8, 3]

$$f\left(X,C\right)=\sum_{i=1}^{N}Min\left\{x_i - c_j{}^2 \mid j=1,\ldots,K\right\} \qquad (1)$$

Where ||xi – cj|| is the Euclidean distance between a data object xi and the cluster center cj. N and K are the number of data objects and the number of clusters, respectively. cj compute as follow：

$$c_j=\frac{1}{n_j}\sum_{x_i\in C_j}x_i,\quad i=1,\ldots,N \; and \;\; j=1,\ldots,K \qquad (2)$$

Where nj is the number of objects belonging to cluster cj. Also, the solution for data clustering problem, must meet the following criteria: (1) none of the K clusters is empty, (2) the intersections of the elements inside each of the clusters with the other clusters must be null, and (3) the sum of the numbers of elements inside the clusters must be equal to the total number of data objects.

There are many clustering algorithms in the literature. Among well-known clustering methods, center-based clustering algorithm can be mentioned [9]. The term "K-means" was first introduced by MacQueen (1967) [10], although the idea goes back to Hugo in 1956. K-means algorithm is kind of partition based clustering and is famous for its simplicity and its fast running. However, it suffers from two shortcomings. It suffers from the dependency on the initial state and convergence to local optima. In addition, the global solutions of large problems cannot be found with reasonable amount of computation effort [3]. In order to overcome the shortcomings of K-means, many heuristic approaches have been applied in the last two decades. In the following, a review of these approaches is provided.

A genetic algorithm has been proposed for solving the clustering problem [11]. They defined a primary mutation operator specific to clustering called distance-based mutation. The results have demonstrated that this method has optimized the output result of K-means. Niknam et al [12] introduced an evolutionary algorithm based on combination of PSO and SA for solving the data clustering problem. SA algorithm is used for searching the space around the global solution, and the mutation operator is used for increasing information exchange between particles so that the algorithm is not trapped in local optima. Rana et al [13] have presented a hierarchical clustering algorithm based on combination of K-means algorithm and PSO algorithm. In this algorithm, the initial search process begins with PSO algorithm due to its quick convergence into the optimum solution; then, the PSO algorithm results are optimized by K-means algorithm so that a better solution is obtained. Yang et al [14] presented a hybrid genetic algorithm (HGA) clustering method, which establishes a balance between population variety and convergence speed by utilizing the tabu list and aspiration criterion. There are several CSO implementations for data clustering [15, 16].

In [15], two major changes have been applied in CSO algorithm. In their proposed algorithm, mixture rate has been ignored, so that the cats run both Seeking and Tracing methods. Moreover, the value of CDC is assumed to be 100%, so that all the cat dimensions change in seeking method. The performance of the CSO algorithm show that it can obtains more accurate results in terms of classification errors in comparison to K-means and PSO. Yugal Kumar and G. Sahoo [16] have proposed a modified version of CSO algorithm, called IL-ICSO, for solving the data clustering problem. For improvement of population diversity and prevention of CSO algorithm from getting trapped in local optima, Opposition-based learning and Cauchy mutation have been used. The performance of the OL-ICSO algorithm measured on several datasets and the simulation results shows that the algorithm provides better results in comparison to other algorithms. Also, a number of other heuristic intelligent algorithms have also been presented for solving the data clustering problem in TS [17], ACO [18], HSA [3,19], HBMO [20], and [21-28].

There are some shortcomings associated with these algorithms, such as dependency on the initial state, trapping in local optima, quality of solutions, and boundary level constraints. In order to overcome these shortcomings, a lot of researches have been going on in this direction. For example, heuristic algorithms like K-means can be used for improvement of initialization, cluster center specification, and initial population quality. Moreover, different mutation operators or combination of global search algorithms with local search algorithms such as TS, SA, and GELS can be used for preventing from getting trapped in local optima and increasing population diversity. The efficiency of chaotic functions has been proven in some of the research performed on different problems [29, 30]. In this paper, a chaotic function has been used instead of a random number generation function for solving initialization problems, initial population quality improvement, creation of more diversity in the population, and prevention from getting trapped in local optima.

3. Proposed Algorithm

Optimization algorithms inspired by the nature have demonstrated considerable success as intelligent optimization methods along with classic methods. Cat Swarm Optimization algorithm [31] has been inspired by PSO [32] and ACO [33] algorithms. The CSO algorithm is one of the latest meta-heuristic algorithms, which has a simple structure and it is easy to implement. Despite high capabilities in solving a variety of optimization problems, Cat Swarm Optimization algorithm is weak in the completely searching the solution space. Due to high convergence speed, the algorithm tends to move toward the local optima, which is also called premature convergence [34].

Random algorithms inspired by natural behavior need a series of random values generated by random functions. The chaos process is a bounded random-like deterministic behavior without duplicated values that does not converge either. In this research, Cat Swarm Optimization algorithm based on chaotic behavior (CCSO) has been used for solving the data clustering problem. In CCSO algorithm, the chaotic number generator has replaced the random number function to improve convergence speed as well as make the search more accurate for calculation of the global optima solution.

3.1 Cat Structure

For solving the data clustering problem, each cat's position represents one solution for the current problem, and the index of each array column represents its corresponding cluster center. In CCSO algorithm, the cat's current position is stated as a two-dimensional vector, and represented by X. In this method, each solution of the problem is defined as follows:

$$\begin{aligned} &X = (c_{11},\ldots,c_{1d},c_{21},\ldots,c_{2d},\ldots,c_{k1},\ldots,c_{kd}) \\ &X = (X_{ij}) \qquad (3) \\ &1 \le i \le k;\ 1 \le j \le d \end{aligned}$$

The numbers of the vector's rows and columns have been defined as k × d, where k is the number of the clusters, and d is the number of each object's dimensions (features). The lower and upper bounds for each dimension of the solution are specified by the relation L^(X_ij)<X_ij<U^(X_ij). Figure 1 shows an instance of the cat's current position.

$$\mathbf{X} = \begin{array}{|c|c|c|c|c|} \hline C_{11} & C_{21} & C_{31} & C_{41} & C_{51} \\ \hline C_{12} & C_{22} & C_{32} & C_{42} & C_{52} \\ \hline C_{13} & C_{23} & C_{33} & C_{43} & C_{53} \\ \hline \end{array}_{3\times 5}$$

Figure 1. An instance of the cat's current position with 5 clusters

As shown in the Figure 1, the number of the clusters has been assumed as five, and the number of each object's features as three. For example, C43 represents the third feature from the center of the fourth cluster among the available clusters.

3.2 Fitness Function

For calculation of each cat's fitness, its current position is applied in the fitness function, and the result is stored in the variable Cost. In CCSO algorithm, the cat's fitness value is equal to the sum of the minimum distances between each cluster center and the instance of that cluster, calculated using Equation (4).

$$Cost = \sum_{n=1}^{N} min\left\{ \sqrt{\sum_{j=1}^{d}\left(X_{C_{ij}} - O_{nj}\right)^{2}} \quad \middle| \quad i = 1,\ldots,k \right\} \qquad (4)$$

Where N and Onj, denote the total number of available instances and the jth feature of the nth object in the test data set, respectively.

3.3 Seeking Mode

Four necessary factors have been defined for modeling the seeking mode: Seeking memory pool (SMP), seeking range of the selected dimension (SRD), counts of dimension to change (CDC), and self-position considering (SPC) [31]. In this step, first SMP copies are made from the kth cats, and each of them is called a candidate point in the search space. If the SPC value equals True, the cat's current position is also considered as a candidate. Next, for each copy, CDC dimensions of the cat's current position are selected, and randomly plus or minus SRD value from their values. For example, consider that the CDC value is 50% and the SRD value is 20%. In view of the CDC value and the number of the cat's current position's dimensions, which equals 5, a number of the dimensions are randomly selected, and their current values are randomly plus or minus from the SRD value. Then, the fitness values of all candidate points are calculated, and each one's selection probability will be calculated using Equation (5).

$$P_i = \frac{|FS_i - FS_b|}{FS_{max} - FS_{min}} \quad , \quad where\, 0 < i < j \tag{5}$$

If all cats' fitness values are exactly equal, selection probabilities of all candidate points will be equal to 1. It should be noted that if the fitness function is aimed at finding the minimum solution, FSb = FSmax; otherwise, FSb = FSmin. Finally, the candidate point with the highest selection probability (best fitness) is selected as the kth cat's new position. Figure 2 has displayed Movement of the cat's position using the seeking mode.

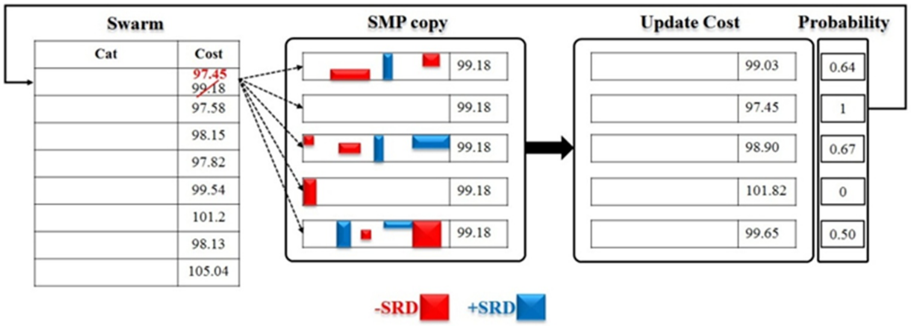


Figure 2. Movement of the cat's position using the seeking mode.

3.4 Tracking Mode

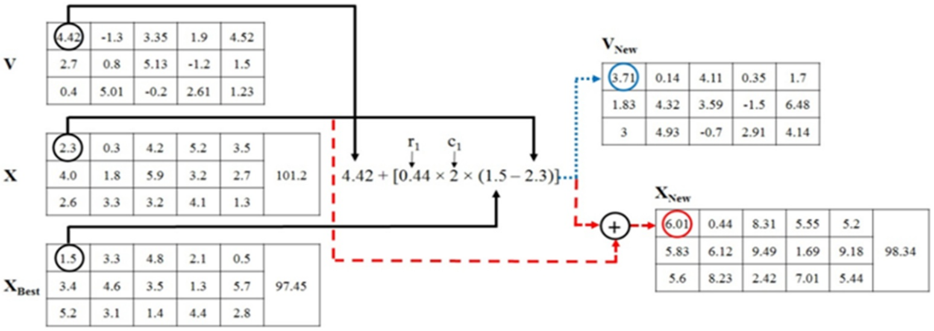
In the tracing mode, the cat's velocity vector is first updated based on the current position and the best cat's position using Equation (6).

$$v_{k,d} = v_{k,d} + r_1 \times c_1 \times \left(x_{best,d} - x_{k,d} \right), \\ d = 1, 2, \ldots, M \tag{6}$$

In this equation, xbest,d is the position of the cat with the best fitness value, xk,d is the position of the kth cat, c1 is a constant, and r1 is a random value in the range [0,1].

In this algorithm it's considered that cl = 2. After updating the cat velocity vector, it is examined whether each dimension's new velocity is within speed limits.

.Figure 3 shows how the velocity vector and the cats' current position are updated



V	4.42	-1.3	3.35	1.9	4.52
	2.7	0.8	5.13	-1.2	1.5
	0.4	5.01	-0.2	2.61	1.23

X	2.3	0.3	4.2	5.2	3.5	101.2
	4.0	1.8	5.9	3.2	2.7	
	2.6	3.3	3.2	4.1	1.3	

X_{Best}	1.5	3.3	4.8	2.1	0.5	97.45
	3.4	4.6	3.5	1.3	5.7	
	5.2	3.1	1.4	4.4	2.8	

V_{New}	3.71	0.14	4.11	0.35	1.7
	1.83	4.32	3.59	-1.5	6.48
	3	4.93	-0.7	2.91	4.14

X_{New}	6.01	0.44	8.31	5.55	5.2	98.34
	5.83	6.12	9.49	1.69	9.18	
	5.6	8.23	2.42	7.01	5.44	

Figure 3. Updating the velocity vector and the cat's position using the tracing mode.

As shown in Figure 3, once the velocity vector is updated, the cat's position is updated using the velocity vector and based on Equation (7).

$$x_{k,d} = x_{k,d} + v_{k,d} \qquad (7)$$

3.5 Chaotic Behavior

It is very important in simulation of complicated phenomena to generate uniform random numbers, as well as used for different applications. Chaos is a phenomenon that occurs in definable nonlinear systems highly sensitive to initial conditions, and it is also a bounded deterministic pseudo-random behavior without duplicated values that does not converge either. The problem with random search algorithms is the premature convergence. In CCSO algorithm, it has been sought to improve the algorithm's convergence speed by replacing random values by chaotic ones. Irreversible one-dimensional patterns are simple systems which are also capable of creating chaotic behavior. Some of the well-known types of one-dimensional patterns are Logistic map, Tent map, ICMIC map, and Sinusoidal map [30]. In this algorithm, the logistic function in Equation (8) is used for generation of chaotic numbers.

$$x_{k+1} = ax_k\left(1 - x_k\right) \qquad (8)$$

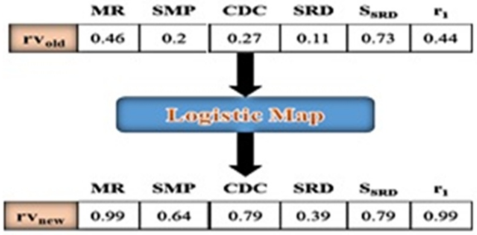
In this equation, xk is the kth repetition of a chaotic number. Clearly, x∈ (0, 1), and under the initial conditions, x_0∉ {0.0, 0.25, 0.5, 0.75, 1.0} is generated. Also, 0 < a ≤ 4, and it is usually considered that a=4.

In CCSO algorithm, all values generated randomly in Cat Swarm Optimization algorithm are saved in a separate array to be updated using Logistic Map function. Table 1 shows these chaotic variables and their applications.

Table 1. Chaotic variables and their applications.

MR	This variable is used for specification of the number of cats performing the seeking mode and tracing mode.
SMP	This variable is used for specification of the number of copies made of the cat current position in the tracing mode. In CCSO algorithm, a maximum value has been assumed for cats' copies so that the SMP value is in the range (0, 1).
CDC	This variable specifies how many of the cats dimensions could change. A maximum value has also been assumed for this variable, so that the CDC value is in the range (0, 1).
SRD	The SRD variable has the range (0, 1), and specifies what percentage is added to or subtracted from the cat's current dimension (by S_{SRD}).
S_{SRD}	If the S_{SRD} value is in the range (0, 0.5], the SRD variable has a negative coefficient, and a positive one otherwise.
r_1	A chaotic value in the range (0, 1) used in Equal 6. This variable is a chaotic coefficient that specifies by what percentage the current cat in relies on the best cat's knowledge, and by what percentage it relies on its own knowledge.

Figure 4 has displayed how these variables are updated. In Figure 4, Logistic Map function is called once for each of the rv array elements, and its output replaces the old value of the corresponding elements.



	MR	SMP	CDC	SRD	S_{SRD}	r_1
rv_{old}	0.46	0.2	0.27	0.11	0.73	0.44

	MR	SMP	CDC	SRD	S_{SRD}	r_1
rv_{new}	0.99	0.64	0.79	0.39	0.79	0.99

Figure 4. Updating of the chaotic variables.

The pseudo-code of CCSO algorithm for solving the data clustering problem has been displayed in Algorithm 1.

Algorithm 1. The CCSO algorithm

```
Initialize the swarm of Cats
do
  evaluate (swarm)
  best_cat = find_x_best (swarm)
  for each cat c in swarm do
    if (check_mr_flag (c) == SEEKING_MODE) then
      smp = chaos (smp)
      spc = chaos (spc)
      neighborhood = copy (c, smp × Max_SMP, spc)
      for each cat c' in neighborhood do
       cdc = chaos (cdc)
        for i = 1 to cdc × Max_CDC do
          srd = chaos (srd)
          s_srd = chaos (s_srd)
          c' = change_dimension (c', srd, s_srd)
        end_for
      end_for
      evaluate (neighborhood)
      c = pick_cat (neighborhood, Max_Probability)
    elseif  // tracing_mode
      $r_1$ = chaos ($r_1$)
      c = update_velocity (c, $r_1$, $c_1$)
      c = update_position (c)
    end_if
  end_for
  foreach cat c in swarm do
    mr = chaos (mr)
    c = set_flags (c, mr)
  end_for
while (a stop criterion is not satisfied)
```

4. Experimental Results

In this section, the results of performing CCSO algorithm over the standard data set of the data clustering problem are presented. For comparison of the performance of the designed algorithm to those of other algorithms, four standard data sets (Iris, Wine, Vowel and Contraceptive Method Choice (CMC)) of UCI repository have been used. Table 2 has displayed the properties of the standard data sets used in the simulation.

4.1 Experimental Conditions

The results of performing of the algorithm over each of these four standard data sets are stated below. For evaluation of CCSO algorithm, the obtained results have been examined and compared to results of other algorithms such as OL-ICSO [16], HBMO [20], GA algorithm [21], PSO algorithm [23], ACO algorithm [6,12,24], SA algorithm [22], TS algorithm [17], and K-means [12,22,24,25]. The CCSO algorithm has been implemented in C# 2012 programming language, and the experiments have been performed on a computer with an Intel(R) Core(TM) i7-4770HQ CPU @ 2.20GHz processor and a 16GB RAM. The values of the main variables have been specified based on the complexity of different test data sets.

Table 2. Standard data sets used in the simulation.

Data Set	Characteristics	Attribute Characteristics	# of Instances	# of Attributes	Area	Date
Iris	Multivariate	Real	150	4	Life	1936
Wine	Multivariate	Integer, Real	178	13	Physical	1991
Vowel	Multivariate	Integer	871	3	Life	1977
Contraceptive Method Choice	Multivariate	Categorical, Integer	1473	9	Life	1987

Table 3 shows the value ranges for each of the main variables in CCSO algorithm. CCSO algorithm has been executed 10 times over each of the test data sets. The best, the worst, and the average solutions have been obtained, and the standard deviation for each of them has also been calculated.

Table 3. Main variables in CCSO algorithm.

Values	Variable
# of Cats	40-100
MR	(0,1)
SMP	(0,1)
Max_{SMP}	5-20
SPC	1
CDC	(0,1)
Max_{CDC}	6-20
SRD	(0,1)
S_{SRD}	(0,1)
r_1	(0,1)
c_1	2
# of Iteration	100
# of Runs for each test	10

4.2 Evaluation

For examining the CCSO algorithm and proving its performance against other algorithms, the results obtained from execution of the algorithm have been compared to those of other algorithms, and presented in this section. The performance of the algorithms is evaluated and compared using four criteria: Sum of intra-cluster distances as a cost function, the convergence speed, population diversity, and error rate (ER). Error rate is the percentage of misplaced data objects, as shown in the following equation:

$$ER = \frac{number\ of\ \ misplaced\ objects}{total\ umber\ of\ \ objects\ within\ dataset} \times 100 \qquad (9)$$

Table 4 shows the results obtained from execution of CCSO algorithm over Iris data set. In this table, Cbest states the best solution, Caverage states the average of the best solutions, Cworst represents the worst solution, and Standard deviation is the standard deviation in different executions of the algorithms experimented with. Based on Table

4, the proposed algorithm converges in all executions into the optimal value 96.5872. The standard deviation of the evaluation function for CCSO clustering algorithm is zero, which suggests the high precision and reliability of the proposed algorithm. In other words, this algorithm provides a better optimum value and standard deviation than those of other methods.

Table 4. Comparison of results obtained from algorithms over Iris data set.

Method	Function Value			Standard deviation
	C_{best}	$C_{average}$	C_{worst}	
CCSO	96.5872	96.5872	96.5872	0
CSO	96.97	97.86	98.91	0.71
OL-ICSO	96.31	96.79	97.36	0.12
HBMO	96.752047	96.95316	97.757625	0.531
GA	113.986503	125.197025	139.778272	14.563
PSO	96.8942	97.2328	97.8973	0.347168
ACO	97.100777	97.171546	97.808466	0.367
SA	97.4573	99.957	102.01	2.018
TS	97.365977	97.868008	98.569485	0.53
K-means	97.333	106.05	120.45	14.6311

Table 5 shows the results obtained from execution of CCSO algorithm over Wine data set. Compared to the other algorithms, the best value selected by CCSO clustering algorithm for Wine data set is optimum and equals 16292.2765. The results shown in Table 5 suggest that the worst result obtained from the CCSO algorithm is more optimal than those obtained by the other methods.

Table 5. Comparison of results obtained from algorithms over Wine data set.

Method	Function Value			Standard deviation
	C_{best}	$C_{average}$	C_{worst}	
CCSO	16292.2765	16293.1916	16293.5369	0.5433
CSO	16352.0357	16410.917	16532.8521	49.816
OL-ICSO	16309.54	16348.46	16568.87	36.24
HBMO	16357.28438	16357.28438	16357.28438	0
GA	16530.53381	16530.53381	16530.53381	0
PSO	16345.9670	16417.4725	16562.3180	85.4974
ACO	16530.53381	16530.53381	16530.53381	0
SA	16473.4825	17521.094	18083.251	753.084
TS	16666.22699	16785.45928	16837.5356	52.073
K-means	16555.68	18061	18563.12	793.213

The results obtained from CMC data set in Table 6 demonstrate that CCSO has achieved the global optimal value 5625.4719, while the best results presented by CSO, OL-ICSO, HBMO, GA, PSO, ACO, SA, TS, and K-means algorithms are 5693.6537, 5628.63, 5699.2670, 5705.6301, 5700.9853, 5701.9230, 5849.0380, 5885.0621, and 5842.20, respectively. The standard deviation of the evaluation function for the algorithm is 1.5691, much smaller than the numbers obtained by the other algorithms.

Table 6. Comparison of results obtained from algorithms over Contraceptive Method Choice.

Method	Function Value			Standard deviation
	C_{best}	$C_{average}$	C_{worst}	
CCSO	5625.4719	5627.6039	5632.8431	1.5691
CSO	5693.6537	5765.0173	5891.3916	36.8431
OL-ICSO	5628.63	5741.16	5892.24	31.47
HBMO	5699.2670	5713.9800	5725.3500	12.6900
GA	5705.6301	5756.5984	5812.6480	50.3694
PSO	5700.9853	5820.9647	5923.2490	46.959690
ACO	5701.9230	5819.1347	5912.4300	45.63470
SA	5849.0380	5893.4823	5966.9470	50.867200
TS	5885.0621	5993.5942	5999.8053	40.84568
K-means	5842.20	5893.60	5934.43	47.16

Table 7 displays the results obtained from Vowel data set. As displayed in the table, CCSO algorithm has presented better results than those of the other methods. Therefore, the proposed algorithm is more reliable and applicable than available algorithms.

Table 7. Comparison of results obtained from algorithms over Vowel data set.

Method	Function Value			Standard deviation
	C_{best}	$C_{average}$	C_{worst}	
CCSO	148964.35	148967.2563	148979.1736	8.2841
CSO	148971.5194	151902.2073	157504.7951	2619.23
OL-ICSO	149201.632	161431.0431	165804.671	2746.0416
HBMO	149513.735	159153.498	165991.6520	3105.5445
GA	148976.0152	151999.8251	158121.1834	2813.4692
PSO	149395.602	159458.1438	165939.8260	3485.3816
ACO	149370.4700	161566.2810	165986.4200	2847.08594
SA	149468.268	162108.5381	165996.4280	2846.23516
TS	149422.26	159242.89	161236.81	916
K-means	148964.35	148967.2563	148979.1736	8.2841

Figure 5 shows the convergence speed of the best solutions obtained from the two CSO and CCSO algorithms. As shown in the figure, the CCSO algorithm is obtained the best solution of CSO algorithm in a smaller number of generations, and has improved it in the later generations. Another criteria for comparison of the two algorithms, is their population diversity in different generations .

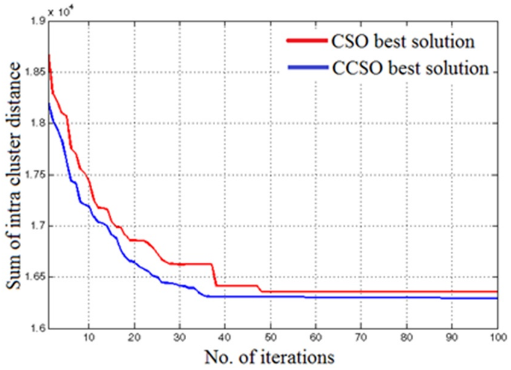


Figure 5. Convergence of sum of intra cluster distance parameter for wine dataset.

The best, worst, and intermediate solutions obtained by CSO and CCSO algorithms are presented in Figure 6. Higher difference between the best and worst solutions of CCSO algorithm and the distribution of the intermediate solutions between these two parameters demonstrates higher diversity in the algorithm. Furthermore, CCSO algorithm has overcome the optimal solutions of the other algorithms in the 37th generation.

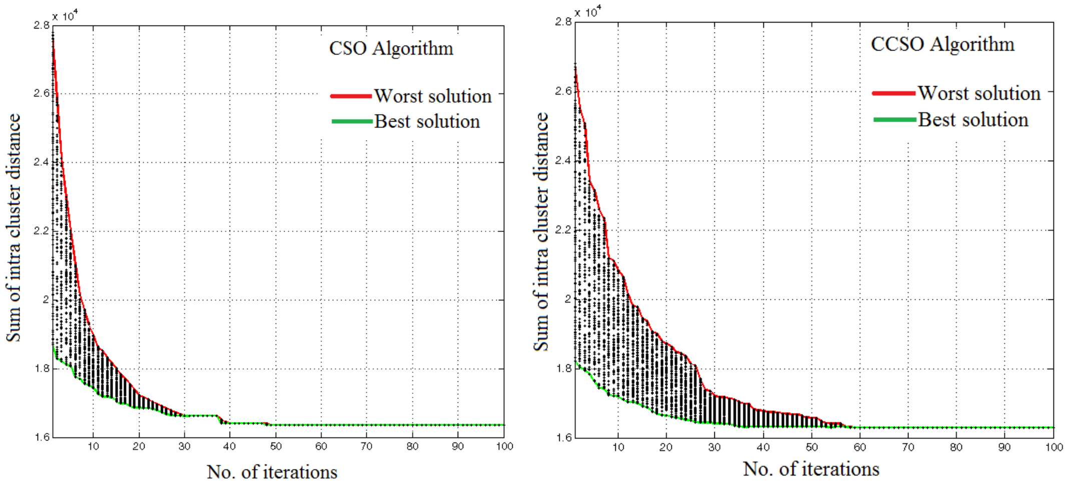


Figure 6. Population diversity of CSO and CCSO algorithms for wine dataset.

The error rates obtained by CSO and CCSO algorithms are displayed in Table 8. As observed in the table, CCSO algorithm has provided lower error rates in the all datasets. Furthermore, the times used by the two algorithms for different datasets are almost similar to each other.

Table 8. The error rate of clustering algorithms on the test datasets.

Data Set	CSO	CCSO
Iris Data Set	10.05	10.01
Wine Data Set	28.93	28.37
Vowel Data Set	42.18	41.62
Contraceptive Method Choice	54.47	53.95

5. Conclusion

Simulating optimization algorithms inspired by the nature for solving complex problems has been an interesting research area for several decades. In this paper, a new method is proposed for solving the data clustering problem using chaos embedded Cat Swarm Optimization algorithm. The CSO algorithm is one of the latest meta-heuristic algorithms, which has a simple structure and it is easy to implement. However, CSO algorithm suffers from diversity and local optima problems. In the proposed algorithm, in order to enhance the population diversity and convergence speed, as well as prevent premature convergence to local optima of the CSO algorithm, a chaotic function called Logistic Map has been used instead of the random number function. The proposed algorithm is compared to other innovative algorithms over four benchmark datasets.

The results of the experiments show that the CCSO algorithm can successfully be applied to clustering for the purpose of classification. In future research, the proposed algorithm can also be utilized for many different areas of applications. In addition, the application of other chaotic functions in CCSO may be effective.

References

[1] Silwattananusarn, T. and Tuamsuk, K., “Data mining and its applications for knowledge management: a literature review from 2007 to 2012”, ***International Journal of Data Mining & Knowledge Management Process,*** Vol. 2, (2012), 13-24.

[2] Fathian, M., Amiri, B. and Maroosi, A., “Application of honey-bee mating optimization algorithm on clustering”, ***Applied Mathematics and Computation,*** Vol. 190, (2007), 1502-1513.

[3] Amiri, B., Hossain, L. and Mosavi, S. E., “Application of harmony search algorithm on clustering”, In: Proceedings of the World Congress on Engineering and Computer Science, San Francisco, USA, (2010), 20-22.

[4] Ronen, M., Shabtai, Y. and Guterman H., “Hybrid model building methodology using unsupervised fuzzy clustering and supervised neural networks”, ***Biotechnology and bioengineering,*** Vol. 77, (2002), 420-429.

[5] Jain, A. K., Murty, M. N. and Flynn, P. J., “Data clustering: a review”, ***ACM computing surveys (CSUR),*** Vol. 31, (1999), 264-323.

[6] Shelokar, P. S., Jayaraman, V. K. and Kulkarni, B. D, “An ant colony approach for clustering”, ***Analytica Chimica Acta,*** Vol. 509, (2004), 187-195.

[7] Garey, M. R., Johnson, D. and Witsenhausen, H., “The complexity of the generalized Lloyd-max problem (corresp.)”, ***IEEE Transactions on Information Theory,*** Vol. 28, (1982), 255-6.

[8] Marinakis, Y., Marinaki, M., Doumpos, M., Matsatsinis, N. and Zopounidis, C., “A hybrid stochastic genetic–GRASP algorithm for clustering analysis”, ***Operational Research,*** Vol. 8, (2009), 33-46.

[9] Forgey, E., “Cluster analysis of multivariate data: Efficiency vs. interpretability of classification”, ***Biometrics,*** Vol. 21, (1965), 768-769.

[10] MacQueen, J., “Some methods for classification and analysis of multivariate observations”, In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1, (1967), 281-297.

[11] Maulik, U. and Bandyopadhyay, S., “Genetic algorithm-based clustering technique”, Pattern Recognition, Vol. 33, (2000), 1455-1465.

[12] Niknam, T., Amiri, B., Olamaei, J. and Arefi, A., “An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering”, ***Journal of Zhejiang University Science A,*** Vol.

10, (2009), 512-519.

[13] Rana, S., Jasola, S. and Kumar, R., “A hybrid sequential approach for data clustering using K-Means and particle swarm optimization algorithm”, ***International Journal of Engineering, Science and Technology,*** Vol. 2, (2010), 167-176.

[14] Yong-Guo, L., Ke-Fei, C. and Xue-Ming, L., “A hybrid genetic based clustering algorithm”, In: Proceedings of the 3rd International Conference on Machine Learning and Cybernetics 2004, Shanghai, (2004), 1677-1682.

[15] Santosa, B. and Ningrum, M. K., “Cat swarm optimization for clustering”, InSoft Computing and Pattern Recognition, 2009. SOCPAR'09. International Conference of, Malacca, Malaysia, (2009), 54-59.

[16] Kumar, Y. and Sahoo, G., “An Improved Cat Swarm Optimization Algorithm Based on Opposition-Based Learning and Cauchy Operator for Clustering”, ***Journal of Information Processing Systems,*** DOI: 10.3745/JIPS.02.0022.

[17] Ng, M. K. and Wong, J. C., “Clustering categorical data sets using tabu search techniques”, Pattern Recognition, Vol. 35, (2002), 2783-2790.

[18] Liu, X.Y. and Fu, H., “An effective clustering algorithm with ant colony”, ***Journal of Computers,*** Vol. 5, (2010), 598-605.

[19] Moh’d Alia, O., Al-Betar, M. A., Mandava, R. and Khader, A, T., “Data clustering using harmony search algorithm”, In: Swarm, Evolutionary, and Memetic Computing: Springer Berlin Heidelberg, (2011), 79-88.

[20] Fathian, M. and Amiri, B., “A honeybee-mating approach for cluster analysis”, ***The International Journal of Advanced Manufacturing Technology 2008,*** Vol. 38, (2008), 809-821.

[21] Krishna, K. and Murty, M. N., “Genetic K-means algorithm”, ***In Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 1999,*** Vol. 29, (1999), 433-439.

[22] Sung, C. S. and Jin, H. W., “A tabu-search-based heuristic for clustering”, Pattern Recognition, Vol. 33, (2000), 849-858.

[23] Kao, Y. T., Zahara, E. and Kao, I. W., “A hybridized approach to data clustering”, ***Expert Systems with Applications,*** Vol. 34, (2008), 1754-1762.

[24] Firouzi, B., Sadeghi, M. S. and Niknam, T., “A new hybrid algorithm based on PSO, SA, and K-means for cluster analysis”, ***International journal of innovative computing, information and control,*** Vol. 6, (2010), 3177-92.

[25] Niknam, T. and Amiri, B., “An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis”, ***Applied Soft Computing,*** Vol. 10, (2010), 183-97.

[26] Moshiri, B., Eslambolchi, P. and HoseinNezhad, R., "Fuzzy Clustering Approach Using Data Fusion Theory and its Application to Automatic Isolated Word Recognition", ***IJE TRANSACTIONS B: Applications,*** Vol. 16, No. 4, (2003), 329-336.

[27] Pakhira, M. K., "A Fast K-means Algorithm using Cluster Shifting to Produce Compact and Separate Clusters (RESEARCH NOTE)", ***IJE TRANSACTIONS A: Basics,*** Vol. 28, No. 1, (2015), 35-43.

[28] Mohammadkhanloo, M., and Bashiri, M., "A Clustering Based Locaion-allocation Problem Considering Transportation Costs and Statistical Properties (Research Note)", ***IJE TRANSACTIONS C: Aspects,*** Vol. 26, No. 6, (2013), 597-604.

[29] Yang, D., Li, G. and Cheng, G., “On the efficiency of chaos optimization algorithms for global optimization”, ***Chaos, Solitons & Fractals,*** Vol. 34, (2007), 1366-1375.

[30] Talatahari, S., Azar, B. F., Sheikholeslami, R. and Gandomi, A. H., “Imperialist competitive algorithm combined with chaos for global optimization”, ***Communications in Nonlinear Science and Numerical Simulation,*** Vol. 17, (2012), 1312-1319.

[31] Chu, S. C., Tsai, P. W. and Pan, J. S., "Cat swarm optimization", In: PRICAI 2006: Trends in artificial intelligence: Springer Berlin Heidelberg, (2006), 854-858.

[32] Eberhart, R.C. and Kennedy, J., "A new optimizer using particle swarm theory", In Proceedings of the sixth international symposium on micro machine and human science, Vol. 1, (1995), 39-43.

[33] Dorigo, M. and Gambardella, L. M., "Ant colony system: a cooperative learning approach to the traveling salesman problem", ***Evolutionary Computation, IEEE Transactions,*** Vol. 1, (1997), 53-66.

[34] Klein, C. E., Coelho, L. D. S., Sant'Anna, Â. M., Freire, R. Z. and Mariani, V. C., "Improved Cat Swarm Optimization Approach Applied to Reliability-Redundancy Problem", European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning 2014, Bruges, Belgium, (2014), 159-164.