Applied Soft Computing xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Efficient protocol for data clustering by fuzzy Cuckoo Optimization Algorithm

Q1 Ehsan Amiri [a,*], Shadi Mahmoudi [b]

[a] Nourabad Mamasani Branch, Islamic Azad University, Nourabad Mamasani, Iran
[b] Computer Engineering Department, Boukan Branch, Islamic Azad University, Boukan, Iran

ARTICLE INFO

Article history:
Received 25 April 2015
Received in revised form
22 November 2015
Accepted 7 December 2015
Available online xxx

Keywords:
Data clustering
Cuckoo Optimization Algorithm (COA)
Dataset
Fuzzy logic
Artificial intelligence

ABSTRACT

Data clustering is a technique for grouping similar and dissimilar data. Many clustering algorithms fail when dealing with multi-dimensional data. This paper introduces efficient methods for data clustering by Cuckoo Optimization Algorithm; called COAC and Fuzzy Cuckoo Optimization Algorithm, called FCOAC. The COA by inspire of cuckoo bird nature life tries to solve continuous problems. This algorithm clusters a large dataset to prior determined clusters numbers by this meta-heuristic algorithm and optimal the results by fuzzy logic. Firstly, the algorithm generates a random solutions equal to cuckoo population and with length dataset objects and with a cost function calculates the cost of each solution. Finally, fuzzy logic tries for the optimal solution. The performance of our algorithm is evaluated and compared with COAC, Black hole, CS, *K*-mean, PSO and GSA. The results show that our algorithm has better performance in comparison with them.

1. Introduction

Q4

Clustering is a main task in data analysis applications. Data clustering technique is introduced for grouping similar or dissimilar data in a given dataset. Grouping, decision-making, machine-learning situations, data mining, document retrieval, image segmentation and pattern classification are some important applications of clustering techniques [1]. Research in exact algorithms, heuristics and meta-heuristics for solving combinatorial optimization problems is increasingly relevant as data science grapples with larger and more complex data sets [2].

Hierarchical and partitional as mentioned in [3,4] are two categories of traditional clustering algorithms. In hierarchical clustering, the number of clusters need not be specified a prior but by partitional methods it should be determined. As a result, hierarchical methods cannot always separate overlapping clusters. Additionally, hierarchical clustering is static and points committed to a given cluster in the early stages cannot be moved between clusters [5]. While the partitional clustering divide in two categories: crisp clustering where each data point belongs to just one cluster and fuzzy clustering where every data point belongs to every cluster to some degree [6,7].

Xin-She and Deb in 2009 [8] proposed a Cuckoo Search (CS) algorithm. In 2011, Rajabioun [9] improved this algorithm and introduced a meta-heuristic algorithm called Cuckoo Optimization Algorithm (COA). COA has some priorities toward CS such as faster convergence, higher speed, high accuracy, local search ability along with general search, search with variable population (population extinction due to poor areas), ability to quickly solve optimization problems with high dimensions and so on. Fuzzy sets a theory that originally introduced by Zadeh [10,11] and it has been developed for expanded linguistic values. The linguistic values are terms which are used instead of numbers and fuzzy set theory [12]. Implementation of multi-criteria control strategies is enabling by rule based FLCs usage. When information is not complete, fuzzy logic is able to make real time decisions. Fuzzy logic systems are able to manipulate the linguistic rules in a natural way and they are particularly suitable in several context like clustering applications. This paper presents a new optimization approach to data clustering based on the COA and Fuzzy COA for decreasing clustering error rate. In fact, the dividing of clusters is dynamically regulated by a Fuzzy Logic Controller (FLC).

The rest of this paper is organized as follows: in Section 2, we have a brief discussion related with works on data clustering. Section 3, shows COA and we have a full description of it. Our work is introduced in Section 4 with some samples for more realization.

Q2 * Corresponding author. Tel.: +98 9384119312.
E-mail addresses: amireehsan@yahoo.com (E. Amiri), mahmoudi.Shadi@Yahoo.Com (S. Mahmoudi).

http://dx.doi.org/10.1016/j.asoc.2015.12.008
1568-4946/

Please cite this article in press as: E. Amiri, S. Mahmoudi, Efficient protocol for data clustering by fuzzy Cuckoo Optimization Algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.12.008

Also, the performance of our proposal is evaluated with several benchmark datasets and the results are compared with some well-known works. Finally, Section 5 summarize and conclude our work.

2. Related works

Nowadays, many scientists work on data categoring in the clusters with different manners such as meta-heuristic algorithms that they are mostly used for solving optimization problems. In [13], the authors proposed an artificial bee colony clustering algorithm to optimally partition N objects into K clusters. Fathian et al. [14] proposed applications of honey bee mating optimization in clustering (HBMK-means). In [15], authors introduce a new hybrid algorithmic nature inspired approach based on the concepts of the Honey Bees Mating Optimization Algorithm (HBMO) and of the Greedy Randomized Adaptive Search Procedure (GRASP), for optimally clustering N objects into K clusters.

Genetic algorithms (GAs) [16–18] are another meta-heuristic methods. Bai [19] implemented a Master–Slave parallel genetic algorithm (PGA) with a Marsili and Giada log-likelihood fitness function to identify clusters within stock correlation matrices. Also, utilized the Nvidia Compute Unified Device Architecture (CUDA) programming model to implement the PGA and visualize the results using minimal spanning trees (MSTs). In [20], researcher proposed a new parameter estimation approach for the mixture normal distribution. The developed model estimates parameters of the mixture normal distribution by maximizing the log likelihood function using genetic algorithm (GA).

Ng et al. [21], presented a tabu search [22,23] based clustering algorithm to extend the K-means paradigm in order to categorical domains and domains with both numeric and categorical values. By using tabu search based techniques, their algorithm can explore the solution space beyond the local optimality aims to find a global solution of the fuzzy clustering problem.

Ant colony optimization (ACO) firstly proposed by Dorigo [24,25]. Shelokar et al. [26] presented an ant colony optimization methodology for optimally clustering N objects into K clusters. In [5], the authors proposed a novel algorithm called ant colony optimization with different favor (ACODF) for data clustering. Li et al. [27] proposed a two-stage framework for gene selection so that the modified ant system and improved ant colony system are used by the fuzzy logic control.

Lee and Geem [28] described a new Harmony Search (HS) meta-heuristic algorithm-based approach for engineering optimization problems with continuous design variables. Mahdavi and Abolhassani [29] proposed a novel Harmony K-means Algorithm (HKA) that deals with document clustering based on Harmony Search (HS) optimization method. Ni et al. [30] improved the PSO with a ring topology.

Hatamlou [31] proposed a new algorithm for data clustering by Black hole phenomena. The particle swarm optimization (PSO) algorithm was developed based on the swarm behavior, such as fish and bird schooling in nature [32]. The gravitational search algorithm (GSA) was constructed based on the law of gravity and the notion of mass interactions. In the GSA algorithm, the searcher agents are a collection of masses that interact with each other based on the Newtonian gravity and the laws of motion [33].

Saida et al. [34] presented a new algorithm for data clustering based on the cuckoo search optimization called CS. Cuckoo search is generic and robust for many optimization problems and it has attractive features like easy implementation, stable convergence characteristic and good computational efficiency.

Pei Honga et al. used of the general GGA representation and operators to reduce redundancy in the chromosome representation for attribute clustering. They compared the efficiency of the proposed approach with that of an existing approach [35].

Ozturk et al. [36] improved version of the discrete binary artificial colony algorithm (DisABC) and applied to the dynamic clustering problem. The performance analysis and performance comparisons of the algorithms have been tested on benchmark problems in terms of the index quality, obtained number of cluster and correct classification percentage (CCP) by applying the static algorithms such as K-means and FCM in addition to the evolutionary some well-known computation based algorithms.

3. Cuckoo Optimization Algorithm

The Cuckoo Optimization Algorithm is based on the obligate brood parasitic behavior of some cuckoo species with the levy flight behavior instead of isotropic simple randomized hiking. These birds called brood parasite because of they even make nests and lays their eggs in the other host bird's nest. The cuckoo just should find a nest with the most similar eggs with its egg. It throws out one of the host bird eggs and lays its egg. Sometimes, the host bird detected cuckoo's egg and this time host bird throw out that.

The cuckoos look for the most suitable area to lay their eggs to maximize the survival rate [37]. when after the cuckoo chicks grow, they make a group and their society. Each group has its area. The best area is destination for other groups and they migrate to this area. Each group resides in an area nearest to the current best area. The egg laying radius calculates by considering the number of eggs each cuckoo will lay and also destination of each cuckoo has to current best area. Then, it lays eggs in some random nests inside its radius area. This process continues until the best position with maximum profit value is obtained and most of the cuckoos population is gathered around the same position [37].

The habitat array uses for keeping input variable values that these variables have floating point values. Eq. (1) shows the habitat array:

$$\text{habitat} = [x_1, x_2, x_3, \ldots, x_{N_{var}}] \quad (1)$$

The profit of a habitat is obtained by evaluation of profit f_p function at a habitat and Eq. (2) shows it:

$$\text{Profit} = f_p(\text{habitat}) = f_p(x_1, x_2, x_3, \ldots, x_{N_{var}}) \quad (2)$$

After that, for beginning the optimization, the algorithm generates a habitat matrix with $N_{var} * N_{pop}$ size and in each of these habits lay random egg number. In nature, each cuckoo lays between 5 and 20 eggs. These numbers are used as lower and upper limits for each cuckoo in each iteration. They also lay eggs within a maximum distance from their habits. This maximum range is called Egg Laying Radius (ELR). Eq. (3) shows this formula in which $\propto$ is an integer value for regularizing the maximum value of ELR, var_{hi} and var_{low} use in order to high limit and low limit.

$$ELR = \propto \times \frac{\text{The number of current cuckoo's eggs}}{\text{Total number of eggs}} \times (var_{hi} - var_{low}) \quad (3)$$

when the cuckoo's become mature, they stay in their groups and habits till the egg-laying time arrives. They migrate to new habits with more similarity of egg to the host birds. After that, the cuckoo groups are formed in a different area, the best society with maximum profit identifies and marks in a goal area for migrate other cuckoos.

The COA uses K-mean clustering technique for grouping cuckoos, after that the mean profit value determines for each group. Then the group with maximum profit finds and spots for new cuckoo's destination. This is obvious in this immigration some cuckoos have a deviation. Fig. 1 depicts this immigration. As this figure shows, after that goal area is determined, the cuckoos in another

Please cite this article in press as: E. Amiri, S. Mahmoudi, Efficient protocol for data clustering by fuzzy Cuckoo Optimization Algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.12.008

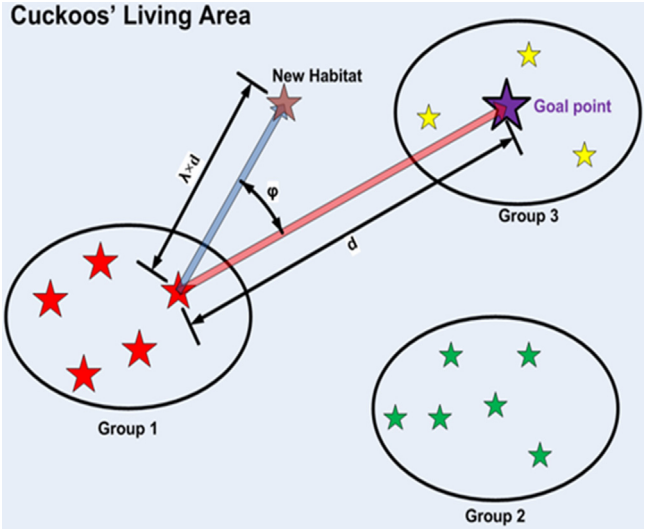

Fig. 1. Cuckoo immigration to habit area.

area out of goal area interest to fly to the goal area. Each cuckoo has to fly equal with d, sometimes, the cuckoo has a deviation equal with Q radios and flies equal with $a * d$ and make a new group. In Fig. 1, small circles are cuckoos.

Parameters a and Q help cuckoos to find new areas and for each cuckoo are defined as follows:

$$a \sim U[0, 1]$$
$$Q \sim U[-w, w] \quad (4)$$

The parameter a is a random value between 0 and 1. The w is a parameter constrains the deviation from goal habitat. To global maximum profit the amount $\Pi/6$ seems necessary for w. Algorithm 1 shows general COA steps proposed in [9,37].

Algorithm 1 Main steps in the proposed COA.

1 Initialize cuckoo habitat with some random points on the profit function (accuracy)
2 Dedicate some eggs to each cuckoo
3 Define ELR for each cuckoo:
$ELR = \propto \times \frac{\text{the number of current cuckoo's eggs}}{\text{total number of eggs}} \times (var_{hi} - var_{low})$
4. Let cuckoos lay eggs inside their corresponding ELR
5 Kill those eggs that are recognized by host birds
6 Let eggs hatch and the chicks grow
7 Evaluate the habitat of each newly grown cuckoo
8 Limit cuckoos maximum number in the environment and kill those who live in worst habitat
9 Cluster cuckoos and find best group and select goal habitat
9.1 clustering with K = means method:
$J = \sum_{j=1}^{MaxIter} \sum_{i=1}^{cuckoopop} ||cuckoo_i^j - c_j||^2$
10 Let new cuckoo population immigrate toward goal habitat
11 If stop condition is satisfied, if not, go to 2

4. Proposed data clustering method

In this section, we describe our proposed algorithm for data clustering with COAC and FCOAC. In the next subsection, we present our method and the performance of that is shown in the last.

4.1. Algorithm details for COAC and FCOAC

This algorithm tries to solve a clustering problem with COA and fuzzy COA where the aim is to find the optimal assignment of N objects with M attributes to one of the K clusters that in each cluster, the sum of squared Euclidean distances between the each object and the center of the belonging cluster is minimized. In this work, at the start, the algorithm generates R cuckoo's agents in the range of minimum and maximum number of cuckoos to build solutions. Each agent has an empty solution string S of length N where each string element corresponds to one of the test samples. In solution string S, an element assigned value shows the cluster number to which the test sample is assigned in S and is a value between 1 and K. For each solution string calculates a cost with a cost function and finally the cluster with minimum cost is selected as best clustering. It is obvious that in different iterations, the population of cuckoos is changing but controlled. These different populations and solutions powers the algorithm to find the best solution.

Algorithm 2 shows FCOAC steps.

Table 1
Input sample dataset.

Object number	Attributes									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	24	2	3	3	1	1	2	3	0	1
2	45	1	3	0	1	1	3	4	0	1
3	43	2	3	7	1	1	3	4	0	1
4	42	3	2	9	1	1	3	3	0	1
5	36	3	3	8	1	1	3	2	0	1
6	19	4	4	0	1	1	3	3	0	1
7	38	2	3	6	1	1	3	2	0	1
8	21	3	3	1	1	0	3	2	0	1

Table 2
Created solutions for 3 cuckoos from the sample dataset with 8 objects.

S_1	S1	3	3	1	3	2	1	2	2
S_2	S2	3	3	3	1	2	2	3	2
S_3	S3	3	1	2	1	1	3	2	2

Algorithm 2 Main steps in the proposed FCOAC.

1 Initialize cuckoo habitat with some random points on the profit function (accuracy)
2 Dedicate some eggs to each cuckoo
3 Define ELR for each cuckoo:
$ELR = \propto \times \frac{\text{the number of current cuckoo's eggs}}{\text{total number of eggs}} \times (var_{hi} - var_{low})$
4 Let cuckoos lay eggs inside their corresponding ELR
5 Kill those eggs that are recognized by host birds
6 Let eggs hatch and the chicks grow
7 Evaluate the habitat of each newly grown cuckoo
8 Limit cuckoos maximum number in the environment and kill those who live in worst habitat
9 Cluster cuckoos and find best group and select goal habitat
9.1 clustering with K = means method:
$J = \sum_{j=1}^{MaxIter} \sum_{i=1}^{cuckoopop} ||cuckoo_i^j - c_j||^2$
10 Let new cuckoo population immigrate toward goal habitat
11 Call FUZZY system for calculating Costs
$fitness(l_h) = \sum_{i=1}^{l_h} \sum_{j=1}^{M} \frac{\sqrt{\left(InputMatrix_{(ClusterMatrix)_{i',j}} - InputMatrix_{(ClusterMatrix)_{r',j}}\right)^2}}{M}$
12 If stop condition is satisfied, if not, go to 2

Let us consider the purpose of illustration, Table 1 shows a dataset with $N = 8$ objects and $M = 10$ attributes that we want to category these in $K = 3$ clusters. We called this table as *InputMatrix*. At the start, R cuckoo's agents is created that they are initial cuckoo's population. COA for each of initial cuckoo's population makes a solution string of random numbers that the length of this solution is equal with the number of objects. Eq. (5) is used for generating these random numbers.

$$S_i = K - 1 * Rand(1, N) + 1 \quad (5)$$

By this equation, N random number generates for each solution string. Each number should be between 1 and K.

Table 2 depicts the solutions that had been generated for $R = 3$ cuckoo's agents. For instance, S_3 is given as below in which the first

Table 3
Divide the solution to K categories based on random cluster numbers in the solution S_3.

K_1	K_2	K_3
2	3	1
4	7	6
5	8	–

Table 4
Calculated fitness value for dataset objects by S_3.

Object Number	Attributes										Fitness
	1	2	3	4	5	6	7	8	9	10	
1	24	2	3	3	1	1	2	3	0	1	0
2	45	1	3	0	1	1	3	4	0	1	0.5058
3	43	2	3	7	1	1	3	4	0	1	0.6900
4	42	3	2	9	1	1	3	3	0	1	0
5	36	3	3	8	1	1	3	2	0	1	0.8344
6	19	4	4	0	1	1	3	3	0	1	0.7906
7	38	2	3	6	1	1	3	2	0	1	0
8	21	3	3	1	1	0	3	2	0	1	2.2228

element signed to cluster label 3 (K_3), the second element is 1 and assigned to cluster label 1 (K_1) and so on.

S_3	3	1	2	1	1	3	2	2

when the solutions have been generated, Eq. (6) for each solution calculates a fitness. The elements in string S_3 is categorized in K groups based on the allocated cluster label. We kept elements index place in these K different arrays. As is clear from Table 3, for cluster label K_1, we have taken 2, 4 and 5 that they are the index number of elements place in S_3 (places 2, 4 and 5 of S_3 are taken 1) and for cluster label K_2, elements 3, 7 and 8 are the places of 2 and then 1 and 6 are taken into cluster label K_3. This table, called as *ClusterMatrix*.

Then, the algorithm calculates the cluster size. The cluster size is the count of elements in each category called l_i that index i has a value between 1 and K. The element count for the category K_1 is $l_1 = 3$, for category K_2 also is $l_2 = 3$ and $l_3 = 2$ for category K_3. One of the elements should be selected as cluster center in each category. The algorithm generates K random numbers between 1 and l_i value for each category. Let us assume these random numbers are 1, 3 and 2 in order for categories K_1, K_2 and K_3. With adapting these number by categories in Table 3, it is obvious that index 1 of category K_1 has the value of 2, index 3 of K_2 has 8 value and the last has 6 value. The cluster centers is determined with these random numbers and the squared Euclidean distance is calculated for each cluster toward to the cluster centers.

The last column in Table 4 illustrates the calculated fitness values for S_3 string by Eq. (6). In this equation l_h is the number of elements in cluster h and r is the cluster center index. In this table, some objects have zero fitness that they are the cluster centers.

$$fitness(l_h) = \sum_{i=1}^{l_h} \sum_{j=1}^{M} \frac{\sqrt{\left(InputMatrix_{(ClusterMatrix)_r j} - InputMatrix_{(ClusterMatrix)_{r'} j}\right)^2}}{M} \quad (6)$$

Next, it calculates the sum of all fitness values and takes for cost value by Eq. (7) for COAC algorithm. The last column in Table 5

Table 5
Calculated cost value for S_3.

S_3	3	1	2	1	1	3	2	2	5.0436

Table 6
Calculated cost value for all solutions.

S_1	3	3	1	3	2	1	2	2	8.8633
S_2	3	3	3	1	2	2	3	2	6.5451
S_3	3	1	2	1	1	3	2	2	5.0436

Table 7
Cluster numbers for S_2 solution.

K_1	K_2	K_3
4	5	1
–	6	2
–	8	3
–	–	7

Table 8
The cost values after local search procedure.

S_1	3	3	1	3	2	1	**3**	2	9.5341
S_2	3	3	**1**	1	2	2	3	2	5.9319
S_3	3	1	2	1	**2**	3	2	2	4.5723

shows the cost values and Table 6 shows these calculated costs for all given solutions in Table 2.

$$CostValue = \sum_{i=1}^{l_h} fitness(i) \quad (7)$$

As Table 6 shows, S_3 has the best cost and is a candidate solution. Many of meta-heuristic algorithms use some form of local search procedure for improving solution are discovered. In this proposal, the local search procedure is applied on all solutions in each iteration. In this work, after the costs are calculated, for more optimization and by this reason that with more probably the cluster with bigger member has bigger error rate, we generate random cluster numbers for H% of that cluster that have bigger cluster size and calculates the cost again for this new solution. In our illustrative example, Table 7 shows the clusters for S_2 solution and with $H = 20\%$, we generate 1 random number (20% of 8). The meaning of this random number is that we have permission to change one of the elements of bigger cluster size (here K_3). Then again, it generates another random number between 1 and the cluster size (K_3 has 4 member then the cluster size is 4) and is the index place of one element K_3.

We assume that this number is 3. Then the algorithm should decide and change the value of this element by another cluster number. The algorithm generates a new random number between 1 and 3 and different with the current cluster number for this element of the solution. Assumes that this random cluster number is 1 and the third element of S_2 has been changed to 1. These operations have been done without changing the cluster centers. Other operations for calculating cost are like before. Finally, the current cost changes to 5.9319 although previous it was 6.5451. Sometimes, this cost has grown like S_1. Table 8 illustrates the final costs after the local search procedure and again S_3 is the best solution. We have highlighted changed bit in Table 8.

Finally, the fitness values are calculated, the fuzzy system determines a fuzzy value for each string. Fig. 2 shows our fuzzy system. We used of Mamdani fuzzy model and have 243 different rules. In our fuzzy system the input values are the same calculated fitness values and the result of this system is the desired cost. Figs. 3 and 4 depict the membership functions for input values and output values. We have used of linguistic variables *very Low* (*VL*), *Low* (*L*), *Medium* (*M*), *High* (*H*) and *Very High* (*VH*) for all input values and *very bad* (*VB*), *Bad* (*B*), *Medium* (*M*), *Good* (*G*) and *Very Good* (*VG*) for output values.

Please cite this article in press as: E. Amiri, S. Mahmoudi, Efficient protocol for data clustering by fuzzy Cuckoo Optimization Algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.12.008

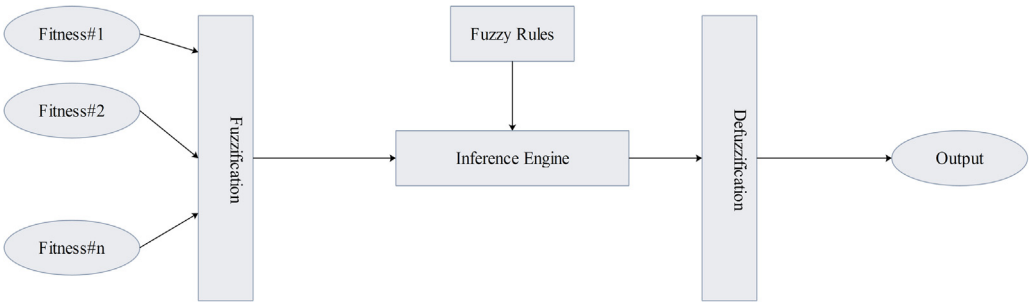

Fig. 2. Embedded fuzzy system model.

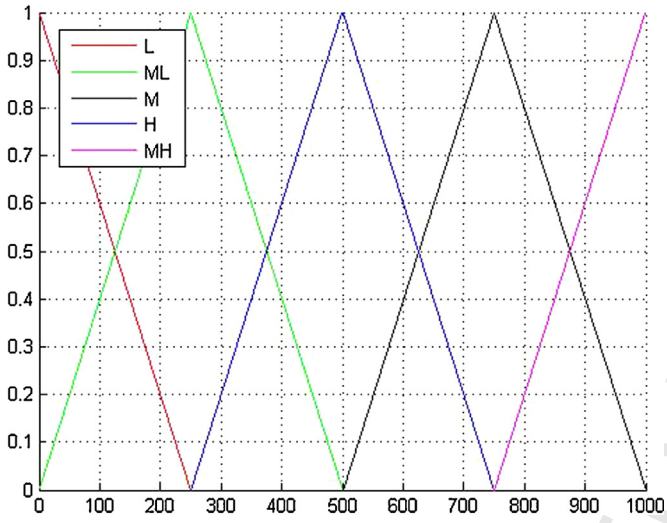

Fig. 3. Membership function for input values.

Table 9
The cost values by FCOAC.

S_1	3	3	1	3	2	1	2	2	8.4796
S_2	3	3	3	1	2	2	3	2	5.5845
S_3	3	1	2	1	1	3	2	2	4.4678

The last column in Table 9 shows the fuzzy cost values for all given solutions in Table 2 and was calculated by Eq. (8). The w is a calculated fuzzy amount for fitness value. As Table 9 shows, S_3 has the best cost and is a candidate solution. This algorithm repeats several times and in each time, the best solution compares with the current candidate solution and the better cost selects for the new candidate solutions and saves.

$$CostValue(l_h) = w(l_h) * fitness(l_h) \quad (8)$$

Table 10
Benchmark dataset description.

Dataset	Number of clusters	Number of attributes	Number of data objects
Iris	3	4	150 (50, 50, 50)
Wine	3	13	178 (59, 71, 48)
Seeds	3	7	210 (70, 70, 70)
CMC	3	9	1473 (629, 334, 510)
UKM	4	5	259 (63, 88, 83, 24)
Vowel	6	3	871 (72, 89, 172, 151, 207, 180)
Glass	6	9	214 (70, 76, 17, 13, 9, 29)

4.2. Performance evaluation

We implemented our proposal by MATLAB tool and evaluated the performance of this by use seven benchmark dataset (Iris, Wine, Seeds, Contraceptive Method Choice (CMC), User Knowledge Modeling (UKM), Vowel and Glass) that have varying complexity. They are available in the repository of the machine learning databases [38,39]. These datasets are given in Table 10.

Several runs were performed to find the algorithmic parameters for finding the best solution and performance. Table 11 shows that initial algorithmic parameters that used for simulation of our algorithm.

$$ER = \frac{Number\ of\ misplaced\ data}{Total\ number\ of\ data\ \ in\ dataset} * 100 \quad (9)$$

We compared our results with some well-known algorithms including Black hole [31], Cuckoo Search (CS) [34], K-means

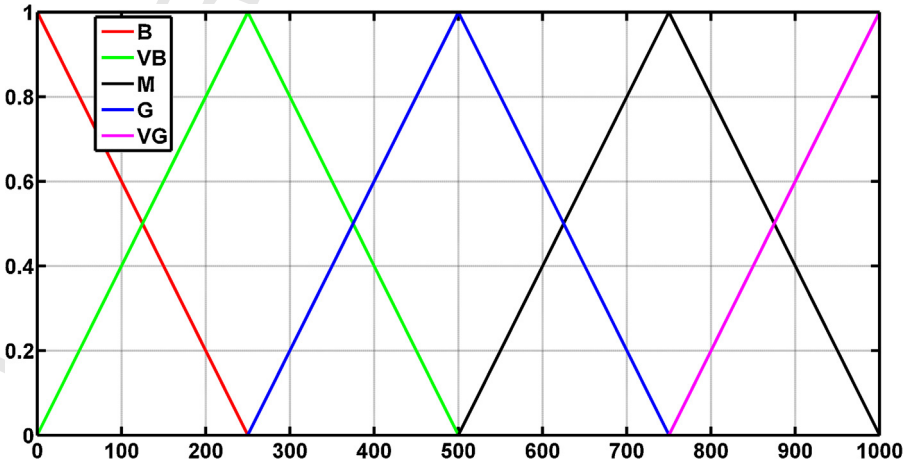

Fig. 4. Membership function for output value.

Please cite this article in press as: E. Amiri, S. Mahmoudi, Efficient protocol for data clustering by fuzzy Cuckoo Optimization Algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.12.008

Table 11
The initial parameters for running the simulator.

Variable name	Value	Description
numCuckooS	10	Minimum number of cuckoo in each habit
maxNumOfCuckoos	20	Maximum number of cuckoo in each habit
MinNumberOfEggs	2	Minimum number of eggs for laying by each cuckoo
maxNumberOfEggs	4	Maximum number of eggs for laying by each cuckoo
maxIter	200	Maximum number of each run iteration
LSP	20%	Local search probability

Table 12
The comparison table for our algorithms in 10 runs on different datasets.

Algorithm	Min-Cost	Avg-Cost	Max-Cost	ER (%)
(a) The results for COAC				
Iris	1.8429	1.8869	3.0229	11.5333
Wine	240.3569	248.0417	550.8276	9.4382
Seeds	5.4613	3.3368	3.2680	13.6667
CMC	0.5681	0.5787	0.7676	11.2403
UKM	3.3124	3.3498	5.3593	14.0476
Vowel	558.3689	582.4493	715.8966	14.1102
Glass	50.0535	53.1587	74.2647	35.2134
(b) The results for FCOAC				
Iris	1.7345	1.6567	2.4341	9.8101
Wine	238.4567	231.0465	548.6783	6.1824
Seeds	5.1287	3.1262	3.0630	10.1334
CMC	0.4961	0.3784	0.6826	10.26
UKM	3.0134	3.1634	4.2395	13.9867
Vowel	557.3098	580.2345	713.6643	12.13
Glass	49.1934	51.9231	68.2312	33.35

Table 13
The comparison table for all algorithms in 10 runs on different datasets.

Algorithm	Error rate				
	Iris (%)	Wine (%)	Vowel (%)	Glass (%)	CMC (%)
FCOAC	9.81	6.18	12.13	33.35	10.26
COAC	11.53	9.44	14.11	35.21	11.24
Black hole	10.02	28.47	41.65	36.51	54.39
CS	10.01	27.07	4245	35.41	51.21
K-means	13.42	31.14	43.57	38.44	54.48
GSA	10.04	29.15	42.26	41.39	57.68
PSO	10.06	28.79	42.39	41.20	51.50

Table 14
The calculated clusters centers for Iris dataset.

Algorithm	Calculated center			
	Att.1	Att.2	Att.3	Att.4
Center1	5.9065	2.9500	4.1174	1.3761
Center2	5.8289	3.1013	3.6342	1.1132
Center3	5.7786	3.0964	3.5071	1.1393

[40], Particle Swarm Optimization (PSO) [32,41] and Gravitational Search Algorithm (GSA) [33]. For getting better results, we experiment the algorithms 10 times and each experiment time consist of 200 iterations and finally the average the results have been calculated. In this work we had calculated the minimum, average and maximum costs between all runs on each dataset. Also, the average error rates (ER) had been calculated. Eq. (9) has been defined for calculating error rate and is the present of the average of data that they are misplaced clustering.

Table 12 illustrates the results of our algorithms on different datasets. The ER results of our proposal is compared with Black hole, ACO, PSO, GSA, CS and *K*-means and have been discussed in Table 13. As is obvious from Table 13 on Iris dataset, our proposal ER is 9.81 and 11.53. Also, on Wine dataset, our results are 6.18 and 9.24 and are better than other algorithms results and in Vowel, Glass and CMC datasets are also better than others.

Also, Tables 14–16 show the calculated clusters centers for *Iris*, *Wine* and *Vowel* datasets in order for best found solution.

Table 15
The calculated clusters centers for Wine dataset.

Algorithm	Calculated center		
	Center1	Center2	Center3
Att.1	12.9988	13.0624	12.8807
Att.2	2.2562	2.4694	2.1665
Att.3	2.3660	2.4022	2.2965
Att.4	18.8420	19.8118	19.6279
Att.5	97.7600	101.6471	98.2791
Att.6	2.2258	2.3519	2.2635
Att.7	1.8570	2.1207	2.0488
Att.8	0.3712	0.3481	0.3781
Att.9	1.4910	1.5856	1.7174
Att.10	5.2250	5.0016	4.9756
Att.11	0.9368	0.9612	0.9740
Att.12	2.5304	2.6646	2.6016
Att.13	771.8600	766.9176	678.2791

Table 16
The calculated clusters centers for Vowel dataset.

Algorithm	Calculated center		
	Att.1	Att.2	Att.3
Center1	482.4	1520.5	2598.6
Center2	462.5	1571.8	2568.1
Center3	478.2	1547.2	2577.3
Center4	459.4	1451.4	2522.6
Center5	479.5	1542.9	2568.3
Center6	463.3	1401.3	2541.4

5. Conclusion

Data clustering is a method for categorizing similar or dissimilar data. Nowadays, many researches are working in this field. A novel algorithm with Fuzzy Cuckoo Optimization Algorithm (FCOA) for data clustering has been proposed in this paper. Our algorithm can be used when the number of clusters is determined. We experimented performance of our proposal on seven different benchmark algorithms and was compared the results with Black hole, CS, *K*-means, GSA and PSO. For getting better performance, we ran 10 times with 200 iterations. The COA is useful for solving data clustering problem. In addition it is simple for implement. The preliminary computational experience is very encouraging in terms of the quality of the solution found and the average number of function evaluations. As a future works, we plan to use of the proposed method with a different algorithm.

References

[1] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comp. Surv. (CSUR) 31 (1999) 264–323.
[2] D. Hendricks, D. Cieslakiewicz, D. Wilcox, T. Gebbie, An unsupervised parallel genetic cluster algorithm for graphics processing units, 2014, arXiv preprint arXiv:1403.4099.
[3] A.K. Jain, R.C. Dubes, Algorithms for Clustering Data, Prentice-Hall Inc., 1988.
[4] G. Karypis, E.-H. Han, V. Kumar, Chameleon hierarchical clustering using dynamic modeling, Computer 32 (1999) 68–75.
[5] C.-F. Tsai, C.-W. Tsai, H.-C. Wu, T. Yang, ACODF: a novel data clustering approach for data mining in large databases, J. Syst. Softw. 73 (2004) 133–145.
[6] M. Blatt, S. Wiseman, E. Domany, Superparamagnetic clustering of data, Phys. Rev. Lett. 76 (1996) 3251.
[7] H. Frigui, R. Krishnapuram, A robust competitive clustering algorithm with applications in computer vision, IEEE Trans. Pattern Anal. Mach. Intell. 21 (1999) 450–465.
[8] Y. Xin-She, S. Deb, Cuckoo search via Lévy flights, in: World Congress on Nature & Biologically Inspired Computing, 2009, NaBIC 2009, 2009, pp. 210–214.

Please cite this article in press as: E. Amiri, S. Mahmoudi, Efficient protocol for data clustering by fuzzy Cuckoo Optimization Algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.12.008

[9] R. Rajabioun, Cuckoo optimization algorithm, Appl. Soft Comp. 11 (2011) 5508–5518.
[10] L.A. Zadeh, Fuzzy sets, Inform. Control 8 (1965) 338–353.
[11] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning — I, Inform. Sci. 8 (1975) 199–249.
[12] E. Amiri, A. Harounabadi, S. Mirabedini, Nodes clustering using fuzzy logic to optimize energy consumption in Mobile Ad hoc Networks (MANET), Manage. Sci. Lett. 2 (2012) 3031–3040.
[13] C. Zhang, D. Ouyang, J. Ning, An artificial bee colony approach for clustering, Expert Syst. Appl. 37 (2010) 4761–4767.
[14] M. Fathian, B. Amiri, A. Maroosi, Application of honey-bee mating optimization algorithm on clustering, Appl. Math. Computat. 190 (2007) 1502–1513.
[15] Y. Marinakis, M. Marinaki, N. Matsatsinis, A hybrid clustering algorithm based on honey bees mating optimization and greedy randomized adaptive search procedure, in: Learning and Intelligent Optimization, Springer, 2008, pp. 138–152.
[16] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading Menlo Park, 1989.
[17] K. Krishna, M.N. Murty, Genetic *K*-means algorithm, IEEE Trans. Syst. Man Cybern. Part B: Cybernetics 29 (1999) 433–439.
[18] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, U Michigan Press, 1975.
[19] A. Bai, Multiobjective clustering using support vector machine: application to microarray cancer data, in: Intelligent Computing, Networking, and Informatics, Springer, 2014, pp. 1209–1215.
[20] J.-Y. Shin, J.-H. Heo, C. Jeong, T. Lee, Meta-heuristic maximum likelihood parameter estimation of the mixture normal distribution for hydro-meteorological variables, Stochast. Environ. Res. Risk Assess. 28 (2014) 347–358.
[21] M.K. Ng, J.C. Wong, Clustering categorical data sets using tabu search techniques, Pattern Recogn. 35 (2002) 2783–2790.
[22] F. Glover, Future paths for integer programming and links to artificial intelligence, Comp. Opera. Res. 13 (1986) 533–549.
[23] K.S. Al-Sultan, A tabu search approach to the clustering problem, Pattern Recogn. 28 (1995) 1443–1451.
[24] M. Dorigo, G. Di Caro, L.M. Gambardella, Ant algorithms for discrete optimization, Artif. Life 5 (1999) 137–172.
[25] M. Dorigo, C. Blum, Ant colony optimization theory: a survey, Theor. Comp. Sci. 344 (2005) 243–278.
[26] P. Shelokar, V.K. Jayaraman, B.D. Kulkarni, An ant colony approach for clustering, Anal. Chim. Acta 509 (2004) 187–195.
[27] Y. Li, G. Wang, H. Chen, L. Shi, L. Qin, An ant colony optimization based dimension reduction method for high-dimensional datasets, J. Bionic Eng. 10 (2013) 231–241.
[28] K.S. Lee, Z.W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Comp. Methods Appl. Mech. Eng. 194 (2005) 3902–3933.
[29] M. Mahdavi, H. Abolhassani, Harmony *K*-means algorithm for document clustering, Data Mining Knowledge Discov. 18 (2009) 370–391.
[30] J. Ni, L. Li, F. Qiao, Q. Wu, A novel memetic algorithm and its application to data clustering, Memetic Comp. 5 (2013) 65–78.
[31] A. Hatamlou, Black hole: a new heuristic optimization approach for data clustering, Inform. Sci. 222 (2013) 175–184.
[32] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization, Swarm Intell. 1 (2007) 33–57.
[33] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inform. Sci. 179 (2009) 2232–2248.
[34] I.B. Saida, K. Nadjet, B. Omar, A new algorithm for data clustering based on cuckoo search optimization, in: Genetic and Evolutionary Computing, Springer, 2014, pp. 55–64.
[35] T.-P. Hong, C.-H. Chen, F.-S. Lin, Using group genetic algorithm to improve performance of attribute clustering, Appl. Soft Computat. 29 (2015) 371–378.
[36] C. Ozturk, E. Hancer, D. Karaboga, Dynamic clustering with improved binary artificial bee colony algorithm, Appl. Soft Computat. 28 (2015) 69–80.
[37] S. Mahmoudi, R. Rajabioun, S. Lotfi, Binary Cuckoo Optimization Algorithm, 2013.
[38] UCI Repository of Machine Learning Databases retrieved from the World Wide Web, 1998.
[39] S.K. Pal, D.D. Majumder, Fuzzy sets and decision making approaches in vowel and speaker recognition, IEEE Trans. Syst. Man Cybern. 7 (1977) 625–629.
[40] A.K. Jain, Data clustering: 50 years beyond *K*-means, Pattern Recogn. Lett. 31 (2010) 651–666.
[41] B.-Y. Qu, J.J. Liang, P.N. Suganthan, Niching particle swarm optimization with local search for multi-modal optimization, Inform. Sci. 197 (2012) 131–143.

Please cite this article in press as: E. Amiri, S. Mahmoudi, Efficient protocol for data clustering by fuzzy Cuckoo Optimization Algorithm, Appl. Soft Comput. J. (2015), http://dx.doi.org/10.1016/j.asoc.2015.12.008