Applied Soft Computing xxx (2010) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

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

A novel clustering approach: Artificial Bee Colony (ABC) algorithm

Dervis Karaboga, Celal Ozturk*

Erciyes University, Intelligent Systems Research Group, Department of Computer Engineering, Kayseri, Turkey

ARTICLE INFO

Article history:
Received 1 August 2008
Received in revised form 21 April 2009
Accepted 12 December 2009
Available online xxx

Keywords:
Classification
Clustering analysis
Artificial Bee Colony algorithm
Particle Swarm Optimization

ABSTRACT

Artificial Bee Colony (ABC) algorithm which is one of the most recently introduced optimization algorithms, simulates the intelligent foraging behavior of a honey bee swarm. Clustering analysis, used in many disciplines and applications, is an important tool and a descriptive task seeking to identify homogeneous groups of objects based on the values of their attributes. In this work, ABC is used for data clustering on benchmark problems and the performance of ABC algorithm is compared with Particle Swarm Optimization (PSO) algorithm and other nine classification techniques from the literature. Thirteen of typical test data sets from the UCI Machine Learning Repository are used to demonstrate the results of the techniques. The simulation results indicate that ABC algorithm can efficiently be used for multivariate data clustering.

1. Introduction

Clustering, which is an important tool for a variety of applications in data mining, statistical data analysis, data compression, and vector quantization, aims gathering data into clusters (or groups) such that the data in each cluster shares a high degree of similarity while being very dissimilar to data from other clusters [1–3]. The goal of clustering is to group data into clusters such that the similarities among data members within the same cluster are maximal while similarities among data members from different clusters are minimal.

Clustering algorithms are generally classified as hierarchical clustering and partitional clustering [3–5]. Hierarchical clustering groups data objects with a sequence of partitions, either from singleton clusters to a cluster including all individuals or vice versa. Hierarchical procedures can be either agglomerative or divisive: agglomerative algorithms begin with each element as a separate cluster and merge them in successively larger clusters; divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters [6,7]. Partitional procedures that we concerned in this paper, attempt to divide the data set into a set of disjoint clusters without the hierarchical structure. The most popular partitional clustering algorithms are the prototype-based clustering algorithms where each cluster is represented by the center of the cluster and the used objective function (a square-error function) is the sum of the distance from the pattern to the center [8].

The most popular class of clustering algorithms is K-means algorithm which is a center based, simple and fast algorithm [9]. However, K-means algorithm highly depends on the initial states and always converges to the nearest local optimum from the starting position of the search. In order to overcome local optima problem, the researchers from diverse fields are applying hierarchical clustering, partition-based clustering, density-based clustering, and artificial intelligence based clustering methods, such as: statistics [10], graph theory [11], expectation-maximization algorithms [12], artificial neural networks [13–16], evolutionary algorithms [17,18], swarm intelligence algorithms [19–24] and so on.

In this paper, Artificial Bee Colony (ABC) optimization algorithm, which is described by Karaboga based on the foraging behavior of honey bees for numerical optimization problems [25], is applied to classification benchmark problems (13 typical test databases). The performance of the ABC algorithm on clustering is compared with the results of the Particle Swarm Optimization (PSO) algorithm on the same data sets that are presented in [26]. ABC and PSO algorithms drop in the same class of artificial intelligence optimization algorithms, population-based algorithms and they are proposed by inspiration of swarm intelligence. Besides comparing the ABC algorithm and PSO algorithm, the performance of ABC algorithm is also compared with a wide set of classification techniques that are also given in [26]. The paper is organized as the clustering problem in Section 2, implementation of the ABC algorithm introduced in Section 3, and then experiments and results presented and discussed in Section 4. We conclude the paper in Section 5 by summarizing the observations and remarking the future works.

* Corresponding author. Tel.: +90 3524374901 32581; fax: +90 3524375784.
E-mail addresses: karaboga@erciyes.edu.tr (D. Karaboga), celal@erciyes.edu.tr (C. Ozturk).

1568-4946/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2009.12.025

2. The Clustering problem

Clustering is the process of recognizing natural groupings or clusters in multidimensional data based on some similarity measures [6]. Distance measurement is generally used for evaluating similarities between patterns. 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 Eq. (1) is described as in [27]:

$$J(w, z) = \sum_{i=1}^{N}\sum_{j=1}^{K} w_{ij}\|x_i - z_j\|^2 \tag{1}$$

where K is the number of clusters, N the number of patterns, $x_i(i = 1, \ldots, N)$ the location of the ith pattern and $z_j(j = 1, \ldots, K)$ is the center of the jth cluster, to be found by Eq. (2):

$$z_j = \frac{1}{N_j}\sum_{i=1}^{N} w_{ij}x_i \tag{2}$$

where N_j is the number of patterns in the jth cluster, w_{ij} the association weight of pattern x_i with cluster j, which will be either 1 or 0 (if pattern i is allocated to cluster j; w_{ij} is 1, otherwise 0).

The clustering process, separating the objects into the groups (classes), is realized by unsupervised or supervised learning. In unsupervised clustering which can also be named automatic clustering, the training data does not need to specify the number of classes. However, in supervised clustering the training data does have to specify what to be learned; the number of classes. The data sets that we tackled contains the information of classes. Therefore, the optimization goal is to find the centers of the clusters by minimizing the objective function, the sum of distances of the patterns to their centers.

In this paper, the adaptation is carried out by minimizing (optimizing) the sum on all training set instances of Euclidean distance in N-dimensional space between generic instance x_j and the center of the cluster z_j. The cost function for the pattern i is given by Eq. (3), as in [26]:

$$f_i = \frac{1}{D_{\text{Train}}}\sum_{j=1}^{D_{\text{Train}}} d(x_j, p_i^{CL_{\text{known}}(x_j)}) \tag{3}$$

where D_{Train} is the number of training patterns which is used to normalize the sum that will range any distance within [0.0, 1.0] and $(p_i^{CL_{\text{known}}(x_j)})$ defines the class that instance belongs to according to database.

3. Artificial Bee Colony algorithm

Artificial Bee Colony (ABC) algorithm was proposed by Karaboga for optimizing numerical problems in [25]. The algorithm simulates the intelligent foraging behavior of honey bee swarms. It is a very simple, robust and population based stochastic optimization algorithm. The performance of the ABC algorithm is compared with those of other well-known modern heuristic algorithms such as Genetic Algorithm (GA), Differential Evolution (DE), Particle Swarm Optimization (PSO) on constrained and unconstrained problems [28–30]. The performance of ABC algorithm on training neural networks is examined by [31] tested on XOR, Decoder–Encoder and 3-Bit Parity benchmark problems and by [32] tested on pattern classification against widely used gradient-based and population-based optimization algorithms.

Pseudo-code of the ABC algorithm is:

1: Load training samples
2: Generate the initial population z_i $i = 1 \ldots SN$
3: Evaluate the fitness (f_i) of the population
4: set cycle to 1
5: **repeat**
6: **FOR** each employed bee**{**
  Produce new solution υ_i by using (6)
  Calculate the value f_i
  Apply greedy selection process**}**
7: Calculate the probability values p_i for the solutions (z_i) by (5)
8: **FOR** each onlooker bee**{**
  Select a solution z_i depending on p_i
  Produce new solution υ_i
  Calculate the value f_i
  Apply greedy selection process**}**
9: **If** there is an abandoned solution for the scout **then** replace it with a new solution which will be randomly produced by (7)
10: Memorize the best solution so far
11: cycle=cycle+1
12: **until** cycle=MCN

In ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. A bee waiting on the dance area for making a decision to choose a food source is called onlooker and one going to the food source visited by it before is named employed bee. The other kind of bee is scout bee that carries out random search for discovering new sources. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution, calculated by:

$$\text{fit}_i = \frac{1}{1 + f_i} \tag{4}$$

In the algorithm, the first half of the colony consists of employed artificial bees and the second half constitutes the onlookers. The number of the employed bees or the onlooker bees is equal to the number of solutions (the cluster centers) in the population. At the first step, the ABC generates a randomly distributed initial population $P(C = 0)$ of SN solutions (food source positions), where SN denotes the size of population. Each solution z_i where $i = 1, 2, \ldots, SN$ is a D-dimensional vector. Here, D is the number of product of input size and cluster size for each data set, i.e. the number of optimization parameters. After initialization, the population of the positions (solutions) is subjected to repeated cycles, $C = 1, 2, \ldots, MCN$, of the search processes of the employed bees, the onlooker bees and scout bees. An employed bee produces a modification on the position (solution) in her memory depending on the local information (visual information) and tests the nectar amount (fitness value) of the new source (new solution). Provided that the nectar amount of the new one is higher than that of the previous one, the bee memorizes the new position and forgets the old one. Otherwise she keeps the position of the previous one in her memory. After all employed bees complete the search process, they share the nectar information of the food sources and their position information with the onlooker bees on the dance area. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount. As in the case of the employed bee, she produces a modification on the position in her memory and checks the nectar amount of the candidate source. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one.

An artificial onlooker bee chooses a food source depending on the probability value associated with that food source, p_i, calculated by the following expression (5):

$$p_i = \frac{\text{fit}_i}{\sum_{n=1}^{SN} \text{fit}_n} \tag{5}$$

where SN is the number of food sources equal to the number of employed bees, and fit_i is the fitness of the solution given in Eq. (4) which is inversely proportional to the f_i given in Eq. (3) where f_i is the cost function of the clustering problem.

In order to produce a candidate food position from the old one in memory, the ABC uses the following expression (6):

$$v_{ij} = z_{ij} + \phi_{ij}(z_{ij} - z_{kj}) \tag{6}$$

where $k \in \{1, 2, \ldots, SN\}$ and $j \in \{1, 2, \ldots, D\}$ are randomly chosen indexes. Although k is determined randomly, it has to be different from i. $\phi_{i,j}$ is a random number between $[-1, 1]$. It controls the production of neighbor food sources around $z_{i,j}$ and represents the comparison of two food positions visible to a bee. As can be seen from (6), as the difference between the parameters of the $z_{i,j}$ and $z_{k,j}$ decreases, the perturbation on the position $z_{i,j}$ decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is adaptively reduced.

The food source of which the nectar is abandoned by the bees is replaced with a new food source by the scouts. In ABC, this is simulated by producing a position randomly and replacing it with the abandoned one. In ABC, providing that a position cannot be improved further through a predetermined number of cycles, then that food source is assumed to be abandoned. The value of predetermined number of cycles is an important control parameter of the ABC algorithm, which is called "*limit*" for abandonment. Assume that the abandoned source is z_i and $j \in \{1, 2, \ldots, D\}$, then the scout discovers a new food source to be replaced with z_i. This operation can be defined as in (7)

$$z_i^j = z_{\min}^j + \text{rand}(0, 1)(z_{\max}^j - z_{\min}^j) \tag{7}$$

After each candidate source position $v_{i,j}$ is produced and then evaluated by the artificial bee, its performance is compared with that of its old one. If the new food source has an equal or better nectar than the old source, it is replaced with the old one in the memory. Otherwise, the old one is retained in the memory. In other words, a greedy selection mechanism is employed as the selection operation between the old and the candidate one. There are three control parameters in the ABC: the number of food sources which is equal to the number of employed or onlooker bees (SN), the value of *limit*, the maximum cycle number (MCN).

In a robust search process, exploration and exploitation processes must be carried out together. In the ABC algorithm, while onlookers and employed bees carry out the exploitation process in the search space, the scouts control the exploration process. The local search performance of ABC algorithm depends on neighborhood search and greedy selection mechanisms performed by employed and onlooker bees. The global search performance of the algorithm depends on random search process performed by scouts and neighbor solution production mechanism performed by employed and onlooker bees.

4. Experimental study

In this work, 13 classification problems from the UCI database [33] which is a well-known database repository, are used to evaluate the performance of the Artificial Bee Colony algorithm. The data sets and their features: the # of patterns, the # of inputs and the # of classes are presented in Table 1. These 13 benchmark problems are chosen exactly the same as in [26], to make a reliable comparison. From the database, the first 75% of data is used in training process as a train set, and the remaining 25% of data is used in testing process as a test set. Although, some data sets' (glass, thyroid, and wine) classes are given in sequential list, they are shuffled to represent every class both in training and in testing as in [26]. The sizes of the train and test sets can be found in Table 1.

4.1. Test problems

The problems considered in this work can be described briefly as follows. Balance data set was generated to model psychological experimental results. Each example is classified as having the balance scale tip to the right, tip to the left, or be balanced. The data set includes 4 inputs, 3 classes and there are 625 examples which is split into 469 for training and 156 for testing.

Cancer and Cancer-Int data sets are based on the "breast cancer Wisconsin - Diagnostic" and "breast cancer Wisconsin - Original" data sets, respectively. They are diagnosis of breast cancer, with 2 outputs (classify a tumor as either benign or malignant). The former one contains 569 patterns, 30 inputs and the latter one contains 699 patterns, 9 inputs.

Credit (the Australian credit card) data set is to assess applications for credit cards based on a number of attributes. There are 690 applicants in total and the output has two classes. The 14 attributes, including 6 numeric values and 8 discrete ones which have 2–14 possible values, are formed into 51 input values.

Dermatology data set contains one of the biggest number of classes; 6 of which are psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis, and pityriasis rubra pilaris. There are 366 samples, including 34 inputs.

The diabetes data set, a two class problem which is the diagnosis of diabetes (whether an individual is diabetes positive or not), has 768 patterns. We used the first 576 patterns as training set and the remaining 192 as test set. There are 8 inputs for each pattern.

For the problem of *Escherichia coli*, the original data set has 336 examples formed of eight classes, but three classes are represented with only 2, 2, 5 examples. Therefore, these 9 examples are omitted and 327 of total, first 245 of them in training and the remaining 82 examples in testing, are used. The data set contains 327 examples with 7 inputs and 5 classes.

Glass data set is the another biggest number of classes (6 classes) in the problems that we tackle. It is used to classify glass types

Table 1
Properties of the problems.

	Data	Train	Test	Input	Class
Balance	625	469	156	4	3
Cancer	569	427	142	30	2
Cancer-Int	699	524	175	9	2
Credit	690	518	172	51	2
Dermatology	366	274	92	34	6
Diabetes	768	576	192	8	2
E. coli	327	245	82	7	5
Glass	214	161	53	9	6
Heart	303	227	76	35	2
Horse	364	273	91	58	3
Iris	150	112	38	4	3
Thyroid	215	162	53	5	3
Wine	178	133	45	13	3

Please cite this article in press as: D. Karaboga, C. Ozturk, A novel clustering approach: Artificial Bee Colony (ABC) algorithm, Appl. Soft Comput. J. (2010), doi:10.1016/j.asoc.2009.12.025

as float processed building windows, non-float processed building windows, vehicle windows, containers, tableware, or head lamps. Nine inputs are based on 9 chemical measurements with one of 6 types of glass which are continuous with 70, 76, 17, 13, 9, and 29 instances of each class, respectively. Total 214 instances are split with 161 for training and 53 for testing.

Heart database that is a diagnosis of heart disease decides to whether at least one of four major vessels is reduced in diameter by more than 50% or not. It contains 76 attributes for each pattern, 35 of which are used as input values. The data is based on Cleveland Heart data from the repository with 303 patterns.

Horse data set is used to predict the fate of a horse with a colic and to classify whether the horse will survive, will die, or will be euthanized. The data set is created based on Horse Colic data with 364 patterns, each of which has 58 inputs from 27 attributes and 3 outputs.

Iris data set includes 150 objects of flowers from the Iris species: Setosa, Versicolor, Virginica. Each of 50 objects in each of three classes have 4 variables; sepal length, sepal width, petal length, and petal width.

Thyroid is the diagnosis of thyroid whether it is hyper or hypofunction. 5 inputs are used to classify 3 classes of thyroid function as being overfunction, normal function, or underfunction. The data set is based on new-thyroid data and contains 215 patterns.

Wine data which was obtained from a chemical analysis of wines were derived from three different cultivators. Therefore, the data analysis determines the three types of wines. There are 178 instances of wine samples with 13 inputs.

4.2. Algorithms and settings

The Particle Swarm Optimization algorithm is a population-based and swarm intelligence based evolutionary algorithm for problem solving. In the PSO algorithm which simulates the social behavior of a flock of birds flying to resources, the particles iteratively evaluate the fitness of the candidate solutions and remember the location which is the best. The parameters of PSO algorithm are (as in [26]): $n = 50$, $T_{\max} = 1000$, $v_{\max} = 0.05$, $v_{\min} = -0.05$, $c_1 = 2.0$, $c_2 = 2.0$, $w_{\max} = 0.9$, $w_{\min} = 0.4$. In order to make a fair comparison, the values of colony size and maximum cycle number of the ABC algorithm are chosen same as or less than the values of swarm size and maximum iteration number used in PSO case, respectively. Such as we selected the *colony size* 20, maximum cycle/generation number (*MCN*) 1000, and *limit* value 1000. Thus, total evaluation # of ABC algorithm is 20,000 where it is 50,000 for PSO algorithm. We observed that in all runs of the algorithms the results do not differ much, so that the experiments are cut after 5 runs since they have the same results.

In [26], besides the PSO algorithm other classification techniques that drop into groups of Bayesian, based on functions, lazy, meta-techniques, tree-based, and rule-based techniques are given. For each of those groups, the selected techniques are: the Bayes Net [34] from the Bayesian; the MultiLayer Perceptron Artificial Neural Network (MLP) [35] and the Radial Basis Function Artificial Neural Network (RBF) [36] from the function-based; the KStar [37] from the lazy; the Bagging [38] and the MultiBoostAB [39] from the meta-techniques; the Naive Bayes Tree (NBTree) [40] from the tree-based ones; the Ripple Down Rule (Ridor) [41] from the rule-based ones; and for the others the Voting Feature Interval (VFI) [42], respectively.

Table 2
Classification error percentages on test data sets.

	ABC	PSO [26]
Balance	15.38	25.47
Cancer	2.81	5.80
Cancer-Int	0	2.87
Credit	13.37	22.96
Dermatology	5.43	5.76
Diabetes	22.39	22.50
E. coli	13.41	14.63
Glass	41.50	39.05
Heart	14.47	17.46
Horse	38.26	40.98
Iris	0	2.63
Thyroid	3.77	5.55
Wine	0	2.22

4.3. Results and discussion

For each problem, we report the Classification Error Percentage (CEP) which is the percentage of incorrectly classified patterns of the test data sets. We classified each pattern by assigning it to the class whose center is closest, using the Euclidean distances, to the center of the clusters. This assigned output (class) is compared with the desired output and if they are not exactly the same, the pattern is separated as incorrectly classified. It is calculated for all test data and the total incorrectly classified pattern number is percentaged to the size of test data set, which is given by Eq. (8).

$$\mathrm{CEP} = 100 \times \frac{\#\ \text{of misclassified examples}}{\text{size of test data set}} \tag{8}$$

As described above, the data is given in two pieces: the training set (the first 75%) and the test set (the last 25%). The results of the algorithms ABC and PSO for the problems are given in Table 2 where classification error percentages (CEP values) are presented. ABC algorithm outperforms PSO algorithm in 12 problems, whereas PSO algorithm's result is better than that of ABC algorithm only for one problem (the glass problem) in terms of classification error. Moreover, the average classification error percentages for all problems are 13.13% for ABC and 15.99% for PSO.

Table 3
Average classification error percentages and ranking of the techniques given in [26] and the ABC algorithm on each problem.

	ABC	PSO	BayesNet	MlpAnn	RBF	KStar	Bagging	MultiBoost	NBTree	Ridor	VFI
Balance	15.38(4)	25.47(9)	19.74(5)	9.29(1)	33.61(10)	10.25(2)	14.77(3)	24.20(8)	19.74(5)	20.63(7)	38.85(11)
Cancer	2.81(2)	5.80 (6)	4.19 (4)	2.93(3)	20.27(11)	2.44(1)	4.47(5)	5.59(6)	7.69(10)	6.36(8)	7.34(9)
Credit	13.37(5)	22.96(10)	12.13(2)	13.81(6)	43.29(11)	19.18(9)	10.68(1)	12.71(4)	16.18(7)	12.65(3)	16.47(8)
Cancer-Int	0.00(1)	2.87 (2)	3.42(3)	5.25(7)	8.17(11)	4.57(5)	3.93(4)	5.14(6)	5.71(9)	5.48(8)	5.71 (9)
Dermatology	5.43(6)	5.76(7)	1.08(1)	3.26(3)	34.66(10)	4.66(5)	3.47(4)	53.26(11)	1.08(1)	7.92(9)	7.60(8)
Diabetes	22.39(1)	22.50(2)	25.52(3)	29.16(7)	39.16(11)	34.05(9)	26.87(5)	27.08(6)	25.52(3)	29.31(8)	34.37(10)
E. coli	13.41(1)	14.63(3)	17.07(5)	13.53(2)	24.38(10)	18.29(8)	15.36(4)	31.70(11)	20.73(9)	17.07(5)	17.07(5)
Glass	41.50(9)	39.05(7)	29.62(5)	28.51(4)	44.44(10)	17.58(1)	25.36(3)	53.70(11)	24.07(2)	31.66(6)	41.11(8)
Heart	14.47(1)	17.46(2)	18.42(3)	19.46(6)	45.25(11)	26.70(10)	20.25(7)	18.42(3)	22.36(8)	22.89(9)	18.42(3)
Horse	38.26(7)	40.98(10)	30.76(2)	32.19(5)	38.46(8)	35.71(6)	30.32(1)	38.46(8)	31.86(3)	31.86(3)	41.75(11)
Iris	0.00(1)	2.63(7)	2.63(7)	0.00(1)	9.99(11)	0.52(5)	0.26(4)	2.63(7)	2.63(7)	0.52(5)	0.00(1)
Thyroid	3.77(2)	5.55(3)	6.66(5)	1.85(1)	5.55(3)	13.32(10)	14.62(11)	7.40(6)	11.11(8)	8.51(7)	11.11(8)
Wine	0.00(1)	2.22(4)	0.00(1)	1.33(3)	2.88(7)	3.99(8)	2.66(6)	17.77(11)	2.22(4)	5.10(9)	5.77(10)

Please cite this article in press as: D. Karaboga, C. Ozturk, A novel clustering approach: Artificial Bee Colony (ABC) algorithm, Appl. Soft Comput. J. (2010), doi:10.1016/j.asoc.2009.12.025

Table 4
Average classification error percentages and general ranking of the techniques on all problems.

	ABC	PSO	BayesNet	MlpAnn	RBF	KStar	Bagging	MultiBoost	NBTree	Ridor	VFI
Average	13.13	15.99	13.17	12.35	26.93	14.71	13.30	22.92	14.68	15.38	18.89
Rank	2	8	3	1	11	5	4	10	6	7	9

Table 5
The sum of ranking of the techniques and general ranking based on the total ranking.

	ABC	PSO	BayesNet	MlpAnn	RBF	KStar	Bagging	MultiBoost	NBTree	Ridor	VFI
Total	41	72	46	49	124	79	58	98	76	87	101
Rank	1	5	2	3	11	7	4	9	6	8	10

In Table 3, the classification error percentages of ABC algorithm and 10 techniques that are given in [26] are presented, and the rankings of the techniques on each problem are also given in the parenthesis. At a glance, one can easily see that the ABC algorithm gets the best solution in 6 of the problems and the second solutions in 2 of the problems. To be able to make a good comparison of the all algorithms, Tables 4 and 5 are reported. The former one shows the average classification errors of all problems and the general ranking based on the average values and the latter one is the sum of the algorithms' rankings of each problem and arranges the totals from minimum value to maximum value. The execution times of the techniques are not considered, since execution times range less than 1 min on a PC with 2.6 GHz Core 2 Duo processor and 2.0 GB-RAM.

The MLP artificial neural network technique is best, ABC is the second best, and BayesNet is the third best technique when mean CEP values from Table 4 are considered. However, even if the results in the table are comparable, we believe that it may cause some significant points to be disregarded since the distribution of the error rates are not proportional. Furthermore, while the error rate difference is around 5% in some problems, it is more than 30% in some other cases. Therefore, the general ranking of the techniques in Table 5 is realized by calculating the sum of the ranks of each problem from Table 3. From this ranking, the first three degree is ABC algorithm as first, BayesNet technique as second, and MLP artificial neural network technique as third. Test error rates (classification error) and rankings from the tables show that clustering with the ABC algorithm offers superior generalization capability. We can claim that by looking at the good performance of ABC algorithm, it can be used for clustering of classification problems studied in this paper.

5. Conclusion

In this work, Artificial Bee Colony algorithm, which is a new, simple and robust optimization technique, is used in clustering of the benchmark classification problems for classification purpose. Clustering is an important classification technique that gathers data into classes (or clusters) such that the data in each cluster shares a high degree of similarity while being very dissimilar from data of other clusters. The performance of the ABC algorithm is compared with Particle Swarm Optimization algorithm and other nine techniques which are widely used by the researchers. The results of the experiments show that the Artificial Bee Colony algorithm can successfully be applied to clustering for the purpose of classification. There are several issues remaining as the scopes for future studies such as using different algorithms in clustering and comparing the results of ABC algorithm to the result of those algorithms.

References

[1] A. Jain, R. Dubes, Algorithms for Clustering Data, Prentice-Hall, Englewood Cliffs, NJ, 1998.
[2] M. Sarkar, B. Yegnanafayana, D. Khemani, A clustering algorithm using an evolutionary programming-based approach, Pattern Recogn. Lett. 18 (1997) 975–986.
[3] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Academic Press, 2001.
[4] H. Frigui, R. Krishnapuram, A robust competitive clustering algorithm with applications in computer vision, IEEE Trans. Pattern Anal. Mach. Intell. 21 (May (5)) (1999) 450–465.
[5] Y. Leung, J. Zhang, Z. Xu, Clustering by scale-space filtering, IEEE Trans. Pattern Anal. Mach. Intell. 22 (December (12)) (2000) 1396–1410.
[6] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surveys 31 (3) (1999) 264–323.
[7] L. Rokach, O. Maimon, Clustering methods, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, Springer, New York, 2005, pp. 321–352.
[8] B. Mirkin, Mathematical Classification and Clustering, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.
[9] J. MacQueen, Some methods for classification and analysis of multivariate observations, pp. 281–297, in: Proc. 5th Berkeley Symp. Math. Stat. Probability, 1967.
[10] E.W. Forgy, Cluster analysis of multivariate data: efficiency versus interpretability of classification, Biometrics 21 (3) (1965) 768–769.
[11] C.T. Zahn, Graph-theoretical methods for detecting and describing gestalt clusters, IEEE Trans. Comput. C-20 (January (1)) (1971) 68–86.
[12] T. Mitchell, Machine Learning, McGraw-Hill, New York, 1997.
[13] J. Mao, A.K. Jain, Artificial neural networks for feature extraction and multivariate data projection, IEEE Trans. Neural Netw. 6 (March (2)) (1995) 296–317.
[14] S.H. Liao, C.H. Wen, Artificial neural networks classification and clustering of methodologies and applications—literature analysis from 1995 to 2005, Expert Sys. Appl. 32 (2007) 1–11.
[15] N.R. Pal, J.C. Bezdek, E.C.K. Tsao, Generalized clustering networks and Kohonen's self-organizing scheme, IEEE Trans. Neural Netw. 4 (July (4)) (1993) 549–557.
[16] T. Kohonen, Self-Organizing Maps, vol. 30, Springer–Verlag, Berlin, Germany, 1995.
[17] E. Falkenauer, Genetic Algorithms and Grouping Problems, Wiley, Chichester, UK, 1998.
[18] S. Paterlini, T. Minerva, Evolutionary approaches for cluster analysis, in: A. Bonarini, F. Masulli, G. Pasi (Eds.), Soft Computing Applications, Springer–Verlag, Germany, 2003, pp. 167–178.
[19] C.H. Tsang, S. Kwong, Ant colony clustering and feature extraction for anomaly intrusion detection, Stud. Comput. Intell. 34 (2006) 101–123.
[20] R. Younsi, W. Wang, A new artificial immune system algorithm for clustering, in: Z.R. Yang, et al. (Eds.), IDEAL 2004, LNCS 3177, Springer, Berlin, 2004, pp. 58–64.
[21] P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni, An ant colony approach for clustering, Anal. Chim. Acta 509 (2004) 187–195.
[22] S. Paterlini, T. Krink, Differential evolution and particle swarm optimisation in partitional clustering, Comput. Stat. Data Anal. 50 (2006) 1220–1247.
[23] Y. Kao, K. Cheng, An ACO-based clustering algorithm, in: M. Dorigo, et al. (Eds.), ANTS, LNCS 4150, Springer, Berlin, 2006, pp. 340–347.
[24] M. Omran, A. Engelbrecht, A. Salman, Particle swarm optimization method for image clustering, Int. J. Pattern Recogn. Artif. Intell. 19 (3) (2005) 297–322.
[25] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
[26] I. De Falco, A. Della Cioppa, E. Tarantino, Facing classification problems with Particle Swarm Optimization, Appl. Soft Comput. 7 (3) (2007) 652–658.
[27] Y. Marinakis, M. Marinaki, M. Doumpos, N. Matsatsinis, C. Zopounidis, A hybrid stochastic genetic—GRASP algorithm for clustering analysis, Oper. Res. Int. J. (ORIJ) 8 (1) (2008) 33–46.
[28] B. Basturk, D. Karaboga, An Artificial Bee Colony (ABC) algorithm for numeric function optimization, in: IEEE Swarm Intelligence Symposium, Indiana, USA, 2006.
[29] D. Karaboga, B. Basturk, Artificial Bee Colony (ABC) optimization algorithm for solving constrained optimization problems, LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, vol. 4529, Springer–Verlag, 2007, pp. 789–798.

Please cite this article in press as: D. Karaboga, C. Ozturk, A novel clustering approach: Artificial Bee Colony (ABC) algorithm, Appl. Soft Comput. J. (2010), doi:10.1016/j.asoc.2009.12.025

[30] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 171–459.
[31] D. Karaboga, B. Basturk, C. Ozturk, Artificial Bee Colony (ABC) optimization algorithm for training feed-forward neural networks, LNCS: Modeling Decisions for Artificial Intelligence, MDAI, vol. 4617, Springer–Verlag, 2007, pp. 318–329.
[32] D. Karaboga, C. Ozturk, Neural networks training by Artificial Bee Colony Algorithm on pattern classification, Neural Netw. World 19 (3) (2009) 279–292.
[33] C.L. Blake, C.J. Merz, University of California at Irvine Repository of Machine Learning Databases, 1998, http://www.ics.uci.edu/ mlearn/MLRepository.html.
[34] F. Jensen, An Introduction to Bayesian Networks, UCL Press/Springer–Verlag, 1996.
[35] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representation by back-propagation errors, Nature 323 (1986) 533–536.
[36] M.H. Hassoun, Fundamentals of Artificial Neural Networks, The MIT Press, Cambridge, MA, 1995.
[37] J.G. Cleary, L.E. Trigg, K*: an instance-based learner using an entropic distance measure, pp. 108–114, in: Proceedings of the 12th International Conference on Machine Learning, 1995.
[38] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[39] G.I. Webb, Multiboosting: a technique for combining boosting and wagging, Mach. Learn. 40 (2) (2000) 159–196.
[40] R. Kohavi, Scaling up the accuracy of naive-Bayes classifiers: a decision tree hybrid, in: E. Simoudis, J.W. Han, U. Fayyad (Eds.), Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, 1996, pp. 202–207.
[41] P. Compton, R. Jansen, Knowledge in context: a strategy for expert system maintenance, in: C.J., Barter, M.J., Brooks (Eds.), Proceedings of Artificial Intelligence, vol. 406, LNAI, Berlin, Springer–Verlag, Adelaide, Australia, 1988, pp. 292–306.
[42] G. Demiroz, A. Guvenir, Classification by voting feature intervals, pp. 85–92, in: Proceedings of the Seventh European Conference on Machine Learning, 1997.