Bee Colony Optimization for Clustering Incomplete Data

Tatjana Davidović[1], Nataša Glišović[2], and Miodrag Rašković[1]

[1] Mathematical institute of Serbian academy of sciences and arts, Belgrade, Serbia
[2] State University of Novi Pazar, Novi Pazar, Serbia
tanjad@mi.sanu.ac.rs; natasaglisovic@gmail.com; goca@mi.sanu.ac.rs

Abstract. Many decision making processes include the situations when not all relevant data are available. The main issues one has to deal with when clustering incomplete data are the mechanism for filling in the missing values, the definition of a proper distance function and/or the selection of the most appropriate clustering method. It is very hard to find the method that can adequately estimate missing values in all required situations. Therefore, in the recent literature a new distance function, based on the propositional logic, that does not require to determine the values of missing data, is proposed. Exploiting this distance, we developed Bee Colony Optimization (BCO) approach for clustering incomplete data based on the p-median classification model. BCO is a population-based meta-heuristic inspired by the foraging habits of honey bees. It belongs to the class of Swarm intelligence (SI) methods. The improvement variant of BCO is implemented, the one that transforms complete solutions in order to improve their quality. The efficiency of the proposed approach is demonstrated by the comparison with some recent clustering methods.

Keywords: Data bases · Missing values · Classification of objects · Nature-inspired methods · Swarm intelligence

1 Introduction

One of the most common tasks in the everyday decision making process is clustering of the large amount of data [15]. It consists of grouping a set of m objects into K (a given number of) groups called clusters. The grouping has to be performed in such a way that objects belonging to the same cluster are similar to each other and at the same time they are different from the objects grouped in other clusters. Each object is described by the set of attributes defining different properties of that object. In fact, an object o^j, $j = 1, 2, \ldots, m$ is represented by an array $o^j = (a_1, a_2, \ldots, a_n)$ in the n-dimensional space. Coordinates a_i, $i = 1, 2, \ldots, n$ represent attributes used to characterize a given object. Attributes

In: S. Belim et al. (eds.): OPTA-SCL 2018, Omsk, Russia, published at http://ceur-ws.org

can be of different type (numerical or categorical), therefore, usually some coding is performed to properly represent their values and make easier to work with them.

Some of the important decisions that need to be made during clustering are to find the most suitable measure of similarity (the one that optimizes a given objective function), to classify objects in a certain number of clusters, sometimes even to determine the number of clusters based on properties of the given data, and, (in many cases) to resolve how to treat the data when they are incomplete. Clustering problem is known to be NP-hard [2, 9, 21] and therefore, heuristic approaches represent the most natural choice.

The methods based on the distances are often used due to their simplicity and applicability in different scenarios. They are very popular in the literature, because they can be used for any type of data, as long as the corresponding distance function is suitable for this type of data. Therefore, the problem of grouping data can be reduced to the problem of finding the distance function for this type of data. Consequently, finding the appropriate distance function has become an important area of research in the data processing [1, 25]. In certain cases the distances should be adapted to a specific domain of variables, such as categorical or time series ([7, 12]).

The number of clusters K may be given in advance, however, in some applications it may not even be known. In the cases when the most suitable number of clusters has to be discovered by the clustering algorithm we are dealing with automatic clustering. A recent survey of nature-inspired automatic clustering methods is given in [17]. From the recent literature one can conclude that the nature-inspired methods are dominating in the clustering field. Among the scarce local search exploring methods are primal-dual variable neighborhood search [14] and tabu search [23].

The special class of clustering problems include dealing with incomplete data. Most of the existing clustering algorithms assume that the object to be classified are completely described. Therefore, prior to their application the data should be preprocessed in order to determine the values of missing data. Recently, some classification algorithms that do not require to resolve the missing data problem were proposed. They include Rough Set Models [28], Bayesian Models [18] and Classifier Ensemble Models [27].

Our study is devoted to resolve the problem of missing data by using a new distance function [11] that is defined on the incomplete data. This distance is based on the propositional logic and does not presume that the missing attributes should be assigned any value. Once the problem of missing data is overcame, one can apply any clustering method to classify the given objects. Instead of using the existing (state-of-the-art) algorithms, we developed a new approach based on the Bee Colony Optimization (BCO) meta-heuristic. BCO is one of the population-based nature-inspired algorithm that mimics the behavior of honey bees during the nectar collection process. It was proposed by Lučić and Teodorović in 2001 ([20]) and evolved during the past years into simple and effective meta-heuristic method. Our implementation of BCO involves the distance func-

tion from [11] and the transformation of solutions that randomly changes the cluster representatives. The preliminary experimental evaluation is performed on 9 data sets from UCI machine learning repository [24]. Our BCO approach is compared against six existing classification algorithms from the recent literature [27]. The considered classification algorithms involve training preprocessing, and therefore, they are in a superior position with respect to our meta-heuristic approach. However, the obtained comparison results show that BCO is promising approach to this hard yet important problem in the automated decision making processes.

The paper is divided into the following sections. The methodology of clustering and the problem of missing data are described in the next section. The BCO method overview and application to the clustering problem are presented in Section 3, while the experimental evaluation is described in Section 4. The advantages and disadvantages of applied technique and some future research are given in Section 5.

2 Clustering of Incomplete Data

We consider clustering problem based on the p-median classification model that can be formulated as follows. Let us assume that we are given a set O of m objects o^j, $j = 1, 2, \ldots, m$. In addition, a distance function $D: O \times O \to \mathbb{R}^+$ is defined over the pairs of objects with a role to measure the similarity between the objects.

In order to provide the Integer Linear Programming (ILP) formulation of the considered clustering problem we define the following binary variables:

$$x_{jl} = \begin{cases} 1, \text{ if the object } o^j \text{ is assigned to cluster represented by object } o^l, \\ 0, \text{ otherwise.} \end{cases}$$

$$y_l = \begin{cases} 1, \text{ if object } o^l \text{ represents a culster,} \\ 0, \text{ otherwise.} \end{cases}$$

The ILP formulation of the clustering problem is now described as follows:

$$\min \sum_{j=1}^{m} \sum_{l=1}^{m} x_{jl} D(j,l) \tag{1}$$

$$s.t. \tag{2}$$

$$\sum_{l=1}^{m} x_{jl} = 1, \quad 1 \le j \le m, \tag{3}$$

$$x_{jl} \le y_l, \quad 1 \le j \le m, \ \ 1 \le l \le m, \tag{4}$$

$$\sum_{l=1}^{m} y_l = K, \tag{5}$$

$$x_{jl}, y_l \in \{0,1\}, \quad 1 \le j \le m, \ \ 1 \le l \le m. \tag{6}$$

The objective function (1) that should be minimized represents the sum of distances from each object to its nearest cluster representative. Every object o^j should be assigned to exactly one cluster (represented by the object o^l) as it is described by constraints (3). Constraints (4) assure that an object o^j can be assigned to a cluster only if the object o^l is selected to be a cluster representative. The total number of luster representatives is set to K by the constraint (5). The binary nature of the decision variables is described by the constraints (6).

It is very common case in real application that some values of the attributes describing objects to be clustering are missing. Values may be missing for various reasons [10]: data may not be available (often happens in medicine e.g., some very expensive or dangerous analyzes are performed only in the critical cases), errors may occur during the entering process, some data may not be considered important at the moment of entering, data acquisition from experiments may be incompetent, some values may be inconsistent with other data and they are erased, responses to a questionnaire may be incomplete, etc.

There are two main techniques to deal with the problem of incomplete data. The simplest approach is to discard samples with missing values. However, this may be applied only in cases when the number of missing values is very small. Otherwise, the result of discarding incompletely defined objects will be insufficient data for drawing any useful conclusion. Imputation is another common solution to an incomplete data problem: it consists of replacing missing values with one selected value of the considered attribute. The replacing value can be determined in various ways [19]: it can be set to zero, to a random number with some given distribution, to the average, minimum or maximum out of the existing values, to the expected value calculated from the existing ones, to a value obtained by the application of linear regression or k-nearest neighbors to the existing ones, etc. Regardless the bulk of replacement techniques, their accuracy still remains an open question. Both the above mentioned methods require significant amount of computational effort and have to be performed in the preprocessing phase of clustering process.

To overcame the missing data problem (i.e., to avoid data imputation), a new distance function (as the measure of similarity between two objects) was proposed in [11]. The most important shortcoming of the known distances (Euclidean, Manhattan, Minkowski, etc.) is that they are only applicable when we know the value of all attributes describing the objects. Therefore, the authors of [11] proposed a metric that can compare the objects for which the values of some attributes are not known. It is based on Hamming distance and propositional logic formulae. For two objects o and p from the set of n-dimensional objects the Hamming distance $d(o, p)$ represents the number of the elements on which these two vectors have different values.

The distance proposed in [11] exploits the propositional logic formulae in the following way. Each object from the data can be represented as a formula α that is conjunction of the attribute values. More precisely, for $o = (o_1, o_2, \ldots, o_n)$ the corresponding formula $\alpha = o_1 \wedge o_2 \wedge \cdots \wedge o_n$. If the value of an attribute o_i is not known, it is replaced by the disjunction of all possible values, i.e.,

$o_i = o_i^1 \vee o_i^2 \vee \cdots \vee o_i^s$, where s is the number of possible values for o_i. Therefore, the corresponding object o can be represented by a set $\mathcal{A}$ of formulae obtained when the value of missing attribute o_i is substituted by each particular possible value. The set $\mathcal{A}$ consists of the following formulae

$$\alpha^1 = o_1 \wedge o_2 \wedge \cdots \wedge o_i^1 \wedge \cdots \wedge o_n;$$

$$\alpha^2 = o_1 \wedge o_2 \wedge \cdots \wedge o_i^2 \wedge \cdots \wedge o_n;$$

$$\vdots$$

$$\alpha^s = o_1 \wedge o_2 \wedge \cdots \wedge o_i^s \wedge \cdots \wedge o_n.$$

If an object o contains more attributes with unknown values, the set $\mathcal{A}$ of formulae contains combinations of all possible values for all missing attributes.

Let the objects o and p be represented by the sets of propositional formulae $\mathcal{A}$ and $\mathcal{B}$. The proposed distance $D(o, p)$ between these two sets of formulae is defined by [11]:

$$D(o,p) = D(\mathcal{A},\mathcal{B}) = \frac{\max\limits_{\alpha\in\mathcal{A}} \min\limits_{\beta\in\mathcal{B}} d(\alpha,\beta) + \max\limits_{\beta\in\mathcal{B}} \min\limits_{\alpha\in\mathcal{A}} d(\alpha,\beta)}{2} \tag{7}$$

where d is the Hamming distance. More precisely, it counts the different corresponding literals in these two formulae.

In case when the values of all attributes are given (no missing values appear), the proposed distance obviously reduces to the Hamming distance. In [11] it has been proved that $D(o, p)$ satisfies the conditions to be considered as metric. In order to evaluate its efficiency, the proposed distance function has been used within the clustering algorithm described in [5]. The experimental evaluation conducted in [11] revealed that the proposed distance outperforms Euclidian distance combined with two data imputation techniques: using average value and linear regression. The Distance $d(o, p)$ Can Be Incorporated Into Any Distance-based Clustering Method and We Use It Within the Bco Meta-heuristic Described in the Next Section.

3 Bee Colony Optimization Meta-heuristic Method

In This Section We Explain Our Implementation of the Bco Meta-heuristic for Clustering Incomplete Data. At the Beginning, an Overview of the Bco Algorithm is Presented and Then the Implementation Details Are Provided.

3.1 Overview of the BCO method

The main feature of the BCO method is that population of artificial bees searches for the optimal solution of a given optimization problem [8]. Each artificial bee is responsible for one solution of the considered problem. Its role is to make that

solution as good as possible depending on the current state of the search. The algorithm runs in iterations until a stopping condition is met. At the end of the BCO execution, the best found solution (the so called global best) is reported as the final one.

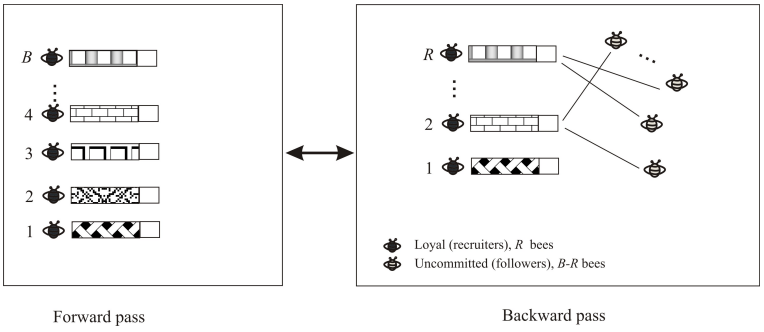

Fig. 1. Main steps of the BCO algorithm

Each iteration contains several execution steps consisting of two alternating phases: *forward pass* and *backward pass* (Fig. 1). During each *forward pass*, all bees are exploring the search space. Each bee applies a predefined number of moves, which yields a new solution. This part of the algorithm is problem dependent and should be resolved in each particular implementation of the BCO algorithm. Having obtained new solutions, the artificial bees start executing a second phase, the so-called *backward pass* where all bees share information about the quality of their solutions. The quality of each solution is defined by the current value of the objective function. When all solutions are evaluated, each bee decides with a certain probability whether it will stay *loyal* to its solution, become a *recruiter* and advertise its solution to other bees. If a bee is not loyal, it becomes an *uncommitted follower* and has to select one of the advertised solutions.

It is obvious that bees with better solutions should have more chances to keep and advertise their solutions. Once the solution is abandoned, the corresponding bee becomes uncommitted follower and has to select one of the advertised solutions. This selection is taken with a probability, such that better advertised solutions have greater opportunities to be chosen for further exploration. Contrary to the bees in nature, artificial bees that are loyal to their solutions are also the recruiters, i.e., their solutions are advertised and would be considered by uncommitted bees.

In the basic variant of BCO there are only two parameters:

B – the number of bees involved in the search and

NC – the number of forward/backward passes in a single BCO iteration.

Algorithm 1 Pseudo-code of the BCO algorithm

```
procedure BCO
    Initialization(Problem input data, B, NC, STOP)
    while stopping criterion is not met do
        for b ← 1, B do                                  ▷ Initializing population
            Sol(b) ← SelectSolution()
        end for
        for u ← 1, NC do
            for b ← 1, B do                              ▷ Forward pass
                EvaluateMove(Sol(b))
                SelectMove(Sol(b))
            end for
            EvaluateSolutions()
            for b ← 1, B do                              ▷ Backward pass
                Loyalty(Sol(b))
            end for
            for b ← 1, B do
                if b is not loyal then
                    Recruitment(Sol(b))
                end if
            end for
        end for
        Update(x_best, f(x_best))
    end while
    Return(x_best, f(x_best))
end procedure
```

The pseudo-code of the BCO algorithm is given by the Algorithm 1. Steps in the forward pass (EvaluateMove, SelectMove, and EvaluateSolutions) are problem specific and, obviously, they differ from implementation to implementation. Therefore, there are no directions how to perform them. On the other hand, that gives the opportunity to maximally explore *a priori* knowledge about the considered problem and obtain a very powerful solution method.

Loyalty decision for each bee depends on the quality of its own solution related to solutions held by other bees. The simplest way to calculate the probability that a bee stays loyal to its current solution is to set

$$p_b = O_b, \quad b = 1, 2, \ldots, B \tag{8}$$

where:
O_b - denotes the normalized value for the objective function (or any other fitness value) of solution created by the b-th bee. The normalization is performed in such a way that $O_b \in [0, 1]$ and that larger values correspond to better solutions. More precisely, in the case of the minimization problem

$$O_b = \frac{f_{max} - f_b}{f_{max} - f_{min}}, \qquad b = 1, 2, \ldots, B.$$

Here, f_b represents the value of objective function found by bee b, while f_{max} and f_{min} correspond to the worst and the best values of objective function held by the bees, respectively.

Equation (8) and a random number generator are used for each artificial bee to decide whether it will stay loyal (and continue exploring its own solution) or it will become an uncommitted follower (and select one among the advertised solutions for further exploration). If the generated random number from $[0, 1]$ interval is smaller than the calculated probability p_b, then the bee stays loyal to its own solution. Otherwise, the bee becomes uncommitted. Some other probability functions are evaluated in [16, 22].

For each uncommitted follower it is decided which recruiter it will follow, taking into account the quality of all advertised solutions. The probability that the solution held by recruiter r would be chosen by any uncommitted bee equals:

$$p_r = \frac{O_r}{\sum_{k=1}^{R} O_k}, \quad r = 1, 2, \ldots, R \tag{9}$$

where O_k corresponds to the k-th advertised solution, and R denotes the number of recruiters. Using equation (9) and a random number generator, each uncommitted follower joins one recruiter through a roulette wheel. In [3] three other selection heuristics are considered (tournament, rank and disruptive selection), however, they will not be used in this work.

BCO has been applied to the clustering problem in [13]. The authors implemented constructive version of the BCO algorithm to cluster completely defined objects. Our implementation uses the improvement variant of BCO with a focus on clustering incomplete data.

3.2 Implementation Details

We implemented the improvement variant of the BCO algorithm, denoted by the BCOi in [8]. This means that each bee is assigned a complete feasible solution of the clustering problem and performs some transformations that should improve its quality.

In order to improve the efficiency of the proposed BCO, we introduced a pre-processing phase that starts with calculating distances between objects using formula (7). In such a way the distance matrix is obtained with determined values of all entries. The next step of the pre-processing phase is to sort the distance matrix (as well as corresponding objects' indices) in the non-decreasing order. This means that in the rows of sorted matrix positions of objects correspond to their distance from the object marked by the row index. More precisely, in the row s object i appears earlier then object j if $D(s, i) < D(s, j)$. The sorted matrix is used in the solution transformation process and reduces its complexity to $O(1)$, i.e., enables to perform the transformation in the constant number of steps.

Each iteration of BCO starts with the initialization of population consisting of B bees. In our implementation, for the first iteration the population is initialized by some randomly selected solutions. This means that K random objects are selected to be initial cluster representatives, called *centroids*. In order to obtain the corresponding value of the objective function, each object is assigned to the cluster represented by the nearest centroid and sum of distances between the objects and the associated centroids is calculated. The best solution in the population is identified and declared as the *global best* solution. Starting from the second iteration, $B/2$ population members are assigned the global best solution obtained from the previous iterations, while the remaining solutions are selected randomly.

An iteration of BCO, is composed of NC forward-backward passes that are implemented as follows. During the forward pass the transformation of the current solution held by each bee is performed. This transformation consists of replacing current centroids with some other objects. Namely, for each of the centroids the new object is selected randomly in the following way. Let c be the forward pass counter, i.e., $c = 1, 2, \ldots, NC$. If $c < NC/2$ a random object is selected from the closest $m/2$ ones. When $c \geq NC/2$ all m objects are considered as candidates to replace the current centroid. It is possible to select 0 as an index of the new centroid with the meaning that this centroid will not be replaced in the transformed solution. The procedure to select a new object consists of the following steps. First, a random number $k = rand(1, t)$ between 1 and t (where $t = m/2$ or $t = m$, depending on the value for c) is selected. Next, we select $k_1 = rand(0, k)$. If $k_1 = 0$, the corresponding centroid will not be changed, otherwise, the k_1-th closest object is used to replace the current centroid. This procedure is repeated for all K centroids and then the transformation of the current solution is completed. The value of the objective function is again determined in the usual way (each object is assigned to the cluster represented by the nearest centroid and sum of distances between the objects and the associated centroids is calculated).

After transforming all B solutions, BCO performs backward pass starting with the identification of the best solution held by bees that is used to update the global best solution. The remaining steps of the backward pass are implemented in standard way [8].

4 Experimental Evaluation of BCO for Clustering

The proposed BCO approach is implemented in C# under the Microsoft Visual Studio 2010 platform and run on AMD A4-6210 APU x64-based processor with 4GB RAM.

In order to evaluate the proposed BCO method for clustering, we tested it on 9 UCI Repository of Machine Learning Databases [24] for classification and compared the obtained results against the existing ones from recent literature [27]. The relevant information about the used databases is presented in Table 1. The presented 9 databases are selected for two reasons: they contain incomplete

data and they are used also in [27] enabling the comparison. As it can be seen from Table 1 the percentage of missing data in all databases is very small and this does not reflect the usual real life situations. However, the results used for comparison are obtained for the original databases, and we left the performance evaluation with respect to the amount of missing data for future work.

Table 1. Description of the databases: number of objects, attributes and classes, and percentage of missing data

Name	# objects	# attributes	attribute type	# classes	% missing
B.cancer	699	10	cat.	2	0.03
CVR	435	16	cat.	2	1.05
Dermatology	366	34	cat.+int.	6	0.02
Heart-h	294	13	cat.+int.+ real	2	0.34
Heart-c	303	13	cat.+int.+ real	5	0.08
Hepatitis	155	19	cat.+int.+ real	2	0.71
H.colic	368	27	cat.+int.+ real	2	1.78
L.cancer	32	56	int.	2	0.17
MM	961	5	int.	2	0.21

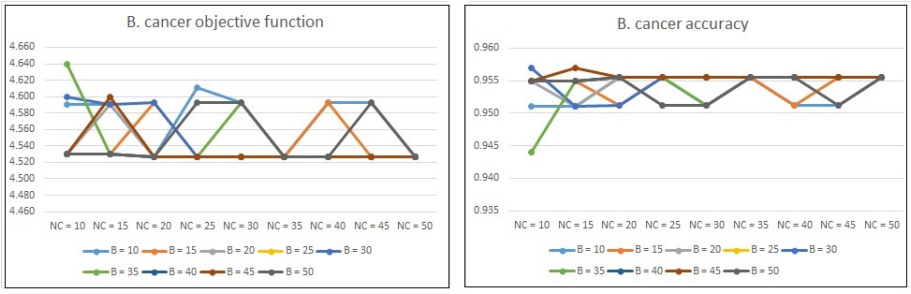

Fig. 2. Parameter tuning of BCO parameters on B.cancer database

In order to determine the best combination of values for BCO parameters, we tested the combinations of following values: $B \in \{10, 15, 20, 25, 30, 35, 40, 45, 50\}$; $NC \in \{10, 15, 20, 25, 30, 35, 40, 45, 50\}$. Stopping criterion is defined as the maximum number of iterations and is set to 200. The number of repetitions is set to 100. It is important to note that BCO showed excellent stability, the same results were obtained in all 100 executions. The obtained results are presented in Figs. 2-5. The two graphics related to the same database represent the dependence of the objective function value and classification accuracy on the values of

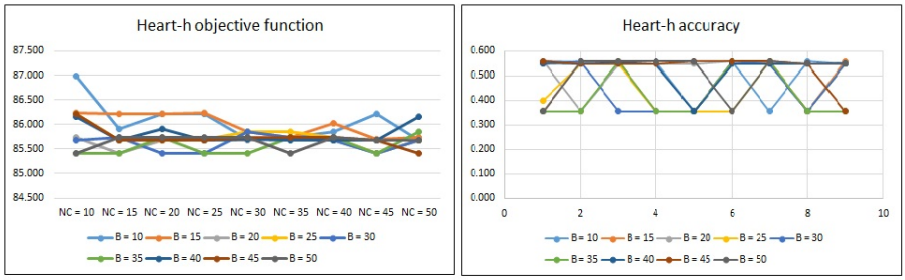

Fig. 3. Parameter tuning of BCO parameters on Heart-h database

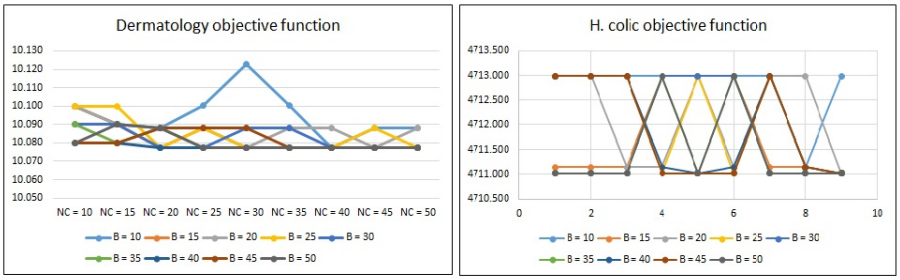

Fig. 4. Parameter tuning of BCO parameters on Dermatology and H.colic databases

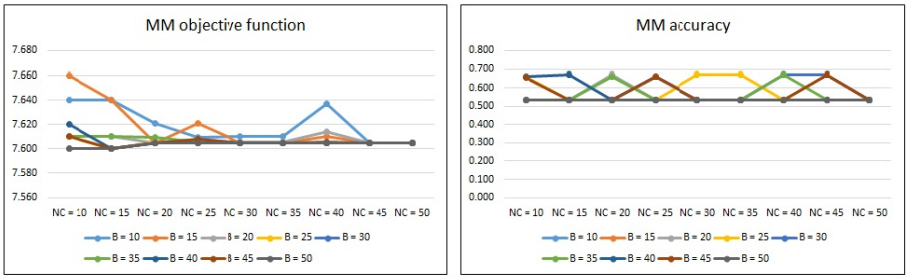

Fig. 5. Parameter tuning of BCO results on MM database

BCO parameters B and NC. For some databases the values of parameters do not have any influence (CVR, Heart-C, Hepatitis, L.cancer), and therefore, the corresponding graphs are omitted. For some other the success rate does not depend on the parameters' values, however, the objective function value varies when the parameters' values are changing (Dermatology and H.colic). Therefore, for these two databases we presented only graphs related to the objective function values. The results for remaining databases are sensitive to the change of the values for BCO parameters regarding both criteria: objective function value and accuracy. Based on this results and having in mind that the main criterion for the quality of clustering method is the minimal value of the objective function, we selected $B = 40$ and $NC = 35$ as the best combination for parameters' values. The results obtained for this combination of parameters' values are compared with the ones reported in [27]. It should be noted that this combination does not guarantee the best possible results for each of the databases, it is selected as the one that provide the least degradation in majority of the databases. For the illustration, we report also the best results, obtained with the combinations of parameters customized for each database.

The methods evaluated in [27] represent the classification algorithms that work in two stages. The first one involves training on the set of already classified data while the second stage is devoted to the evaluation of the resulting trained classification method. Therefore, those methods already have some knowledge about the data to be processed. Our clustering method is based on a metaheuristic approach that does not involve any training or learning and it could be considered to be in the inferior position with respect to the other method used for comparison.

In [27] six methods were compared: Selective Neural Network Ensemble (SNNE), Multi-Granulation Ensemble Method (MGNE) proposed in [26], Neural Network Ensemble method without any assumption about distribution (NNNE) from [6], the method (Bag1) obtained when the mean value of the corresponding attribute is assigned to the missing entries, and then conduct the bagging algorithm [4], the method conducting bagging algorithm after the removal of the samples with missing values (Bag2) and NN method that conduct a single classifier on the data remaining after removing the samples with missing values. We compared our BCO approach with all these methods with respect to the results about classification accuracy reported in [27]. The comparison results are presented in Table 2.

The first column of Table 2 contains the database name, the next 6 columns present the accuracy of the methods evaluated in [27]. The remaining two columns contain the results related to our BCO. Accuracy obtained for the selected combination of values for BCO parameters ($B = 40$, $NC = 35$) is reported in the eighth column, while in the last column the best results (for customized combination of parameter values is shown. The improved values are underlined. We cannot estimate if the comparison is fair enough, since no time measurement units are provided in [27]. Our running times were in milliseconds and therefore,

Table 2. Results of the comparison: classification accuracy of the tested methods

Databases	SNNE	MGNE	NNNE	Bag1	Bag2	NN	BCO	best BCO
B.cancer	0.935	0.938	0.936	0.938	0.939	0.65	**0.956**	**0.957**
CVR	0.942	0.945	0.942	**0.965**	0.964	0.513	0.875	0.875
Dermatology	**0.886**	0.879	0.861	0.849	0.844	0.277	0.874	0.874
Heart-h	**0.816**	0.806	0.806	0.812	0.76	0.656	0.548	0.558
Heart-c	0.526	0.519	0.516	0.52	0.524	0.437	**0.671**	**0.671**
Hepatitis	0.676	0.676	0.682	0.663	0.681	0.551	**0.895**	**0.895**
H.colic	0.735	0.723	0.701	0.778	0.633	0.583	**0.923**	**0.923**
L.cancer	0.524	0.522	0.498	0.503	0.517	0.347	**0.719**	**0.719**
MM	0.834	0.836	0.801	0.829	**0.84**	0.504	0.536	0.672
bf Average	0.764	0.760	0.749	0.762	0.745	0.502	**0.777**	**0.794**

they are not reported here. As can be seen from this table, our BCO shows slightly better performance on five (out of nine) databases and on average.

The presented results are preliminary, there is still room for the improvements of the BCO generated results and we plan realize them as the future work. For example, the solution modification scheme could be changed in the sense that learning from previously visited solutions is included and the parameter values may be tuned with finer granulation.

5 Conclusion

The clustering problem in large databases containing incomplete data is considered in this paper. Instead of deleting objects with missing attributes or imputing missing values, we used the recently proposed new metric function able to determine distance between objects with missing attributes. This distance function is based on Hamming distance and propositional logic and can be incorporated into any clustering method. We used it within Bee Colony Optimization framework and implemented a new population-based approach to the clustering problem. The proposed implementation is tested on large databases available on the Internet and compared with some recent clustering methods developed for classifying incomplete data. Preliminary experimental evaluation shows the potential of our approach.

The directions of future work may include some modification of the proposed approach. For example, theoretical aspect of meta-heuristic methods can be taken into account, new transformation rules could be examined. The second direction should include performance evaluation of the proposed approach with respect to the amount of missing data. In addition, more extensive experimental evaluation should be performed and the proposed approach need to be compared with variety of the existing meta-heuristic clustering methods.

Acknowledgement. This work has been supported by the Serbian Ministry of Science, Grant nos. OI174033 and III044006.

References

1. Aggarwal, C.C.: Towards systematic design of distance functions for data mining applications. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 9–18. ACM (2003)
2. Aloise, D., Deshpande, A., Hansen, P., Popat, P.: NP-hardness of Euclidean sum-of-squares clustering. Machine learning 75(2), 245–248 (2009)
3. Alzaqebah, M. Abdullah, S.: Hybrid bee colony optimization for examination timetabling problems. Computers & Operations Research 54, 142–154 (2015)
4. Breiman, L.: Bagging predictors. Machine learning 24(2), 123–140 (1996)
5. Chan, E.Y., Ching, W.K., Ng, M.K., Huang, J.Z.: An optimization algorithm for clustering using weighted dissimilarity measures. Pattern recognition 37(5), 943–952 (2004)
6. Chen, H., Du, Y., Jiang, K.: Classification of incomplete data using classifier ensembles. In: International Conference on Systems and Informatics (ICSAI). pp. 2229–2232. IEEE (2012)
7. Das, G., Mannila, H.: Context-based similarity measures for categorical databases. In: European Conference on Principles of Data Mining and Knowledge Discovery. pp. 201–210. Springer (2000)
8. Davidović, T., Teodorović, D., Šelmić, M.: Bee colony optimization Part I: The algorithm overview. Yugoslav Journal of Operational Research 25(1), 33–56 (2015)
9. Falkenauer, E.: Genetic Algorithms and Grouping Problems. Wiley, New York (1998)
10. Gautam, C., Ravi, V.: Evolving clustering based data imputation. In: 2014 International Conference on Circuit, Power and Computing Technologies (ICCPCT). pp. 1763–1769. IEEE (2014)
11. Glišović, N., Rašković, M.: Optimization for classifying the patients using the logic measures for missing data. Scientific Publications of the State University of Novi Pazar Series A: Applied Mathematics, Informatics and mechanics 9(1), 91–101 (2017)
12. Gunopulos, D., Das, G.: Time series similarity measures and time series indexing. In: Acm Sigmod Record. vol. 30, P. 624. ACM (2001)
13. Haghighat, A.T., Forsati, R.: Data clustering using bee colony optimization. In: The Seventh International Multi-Conference on Computing in the Global Information Technology. pp. 189–194 (2012)
14. Hansen, P., Brimberg, J., Urošević, D., Mladenović, N.: Solving large p-median clustering problems by primal–dual variable neighborhood search. Data Mining and Knowledge Discovery 19(3), 351–375 (2009)
15. Jain, A.K.: Data clustering: 50 years beyond K-means. Pattern recognition letters 31(8), 651–666 (2010)
16. Jakšić Krüger, T.: Development, implementation, and theoretical analysis of the Bee Colony Optimization (BCO) meta-heuristic method. Ph.D. thesis, Faculty of Techincal Sciences, University of Novi Sad (2017)
17. Jose-Garcia, A., Gómez-Flores, W.: Automatic clustering using nature-inspired metaheuristics: A survey. Applied Soft Computing 41, 192–213 (2016)

18. Lin,H.-C. Su, C.-T.: A selective bayes classifier with meta-heuristics for incomplete data. Neurocomputing 106, 95–102 (2013)
19. Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data. Wiley Series in Probability and Statistics. John Wiley & Sons (2002)
20. Lučić, P., Teodorović, D.: Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. In: Preprints of the TRISTAN IV Triennial Symposium on Transportation Analysis. pp. 441–445 (2001)
21. Mahajan, M., Nimbhorkar, P., Varadarajan, K.: The planar k-means problem is NP-hard. In: International Workshop on Algorithms and Computation. pp. 274–285. Springer (2009)
22. Maksimović, P., Davidović, T.: Parameter calibration in the bee colony optimization algorithm. In: XI Balcan Conference on Operational Research (BALCOR 2013). pp. 263–272 (2013)
23. Sung, C.S., Jin, H.W.: A tabu-search-based heuristic for clustering. Pattern Recognition 33(5), 849–858 (2000)
24. UCI Repository of machine learning databases for classification
25. Wang, F., Sun, J.: Survey on distance metric learning and dimensionality reduction in data mining. Data Mining and Knowledge Discovery 29(2), 534–564 (2015)
26. Yan, Y-T., Zhang, Y-P., Zhang, Y-W.: Multi-granulation ensemble classification for incomplete data. In: International Conference on Rough Sets and Knowledge Technology. pp. 343–351. Springer (2014)
27. Yan, Y-T., Zhang, Y-P., Zhang, Y-W., Du, X-Q.: A selective neural network ensemble classification for incomplete data. International Journal of Machine Learning and Cybernetics 8(5), 1513–1524 (2017)
28. Zhang, Q., Xie, Q., Wang, G.: A survey on rough set theory and its applications. CAAI Transactions on Intelligence Technology 1(4), 323–333 (2016)