Ann Oper Res (2011) 188:343–358
DOI 10.1007/s10479-009-0519-2

A hybrid ACO-GRASP algorithm for clustering analysis

Yannis Marinakis · Magdalene Marinaki · Michael Doumpos · Nikolaos Matsatsinis · Constantin Zopounidis

Published online: 29 January 2009

Abstract Cluster analysis is an important tool for data exploration and it has been applied in a wide variety of fields like engineering, economics, computer sciences, life and medical sciences, earth sciences and social sciences. The typical cluster analysis consists of four steps (i.e. feature selection or extraction, clustering algorithm design or selection, cluster validation and results interpretation) with feedback pathway. These steps are closely related to each other and affect the derived clusters. In this paper, a new metaheuristic algorithm is proposed for cluster analysis. This algorithm uses an Ant Colony Optimization to feature selection step and a Greedy Randomized Adaptive Search Procedure to clustering algorithm design step. The proposed algorithm has been applied with very good results to many data sets.

Keywords Cluster analysis · Feature selection problem · Ant colony optimization · GRASP · Stochastic optimization

Y. Marinakis · N. Matsatsinis
Decision Support Systems Laboratory, Department of Production Engineering and Management, Technical University of Crete, 73100 Chania, Greece

Y. Marinakis
e-mail: marinakis@ergasya.tuc.gr

N. Matsatsinis
e-mail: nikos@ergasya.tuc.gr

M. Marinaki
Industrial Systems Control Laboratory, Department of Production Engineering and Management, Technical University of Crete, 73100 Chania, Greece
e-mail: magda@dssl.tuc.gr

M. Doumpos · C. Zopounidis (✉)
Financial Engineering Laboratory, Department of Production Engineering and Management, Technical University of Crete, 73100 Chania, Greece
e-mail: kostas@dpem.tuc.gr

M. Doumpos
e-mail: mdoumpos@dpem.tuc.gr

1 Introduction

Clustering is a very important problem that has been addressed in many contexts and by researchers in many disciplines. It embraces various scientific disciplines, from mathematics and statistics to biology and genetics, each of which uses different terms to describe the topologies formed using this analysis. Other terms more or less synonymous with clustering include unsupervised learning, numerical taxonomy, vector quantization and learning by observation.

Clustering analysis identifies clusters (groups) embedded in the data, where each cluster consists of objects that are similar to one another and dissimilar to objects in other clusters (for high quality clusters, the inter-cluster similarity is low and the intra-cluster similarity is high) (Jain et al. 1999; Mirkin 1996; Rokach and Maimon 2005; Xu and Wunsch 2005). Clustering algorithms can be classified into two categories: *hierarchical clustering* and *partitional clustering*. The most well known of the partitional methods is the classical k-means algorithm (Babu and Murty 1993; Ng 2000; Tarsitano 2003). For *hard* or *crisp clustering* each object is assigned to only one cluster while a pattern may also be allowed to belong to all clusters with a degree of membership in *fuzzy clustering*.

The typical cluster analysis consists of four steps (with a feedback pathway) which are the *feature selection* or *extraction*, the *clustering algorithm design* or *selection*, the *cluster validation* and the *results interpretation* (Xu and Wunsch 2005).

The *basic feature selection problem (FSP)* is an optimization one, where by the use of a performance measure for each subset of features its ability to classify the samples is measured. The problem is to search through the space of feature subsets to identify the optimal or near-optimal one with respect to the performance measure. In the literature many successful feature selection algorithms have been proposed (Aha and Bankert 1996; Cantu-Paz et al. 2004; Jain and Zongker 1997; Jouve and Nicoloyannis 2005; Kira and Rendell 1992; Marinakis et al. 2008)). *Feature extraction* utilizes some transformations to generate useful and novel features from the original ones.

The *clustering algorithm design* or *selection* step is usually combined with the selection of a corresponding proximity measure and the construction of a criterion function. Pattern proximity is usually measured by a distance function defined on pairs of patterns and a variety of distance measures are in use (Jain et al. 1999; Rokach and Maimon 2005). After the selection of the proximity measure, a clustering criterion function is formulated which makes the partition of clusters a well defined optimization problem. However, it should be noted that the problem is NP-hard as the clustering objective functions are highly non-linear and multi-modal functions and as a consequence it is difficult to investigate the problem in an analytical approach. Many heuristic, metaheuristic and stochastic algorithms have been developed in order to find a near optimal solution in reasonable computational time. An analytical survey of the clustering algorithms can be found in Jain et al. (1999), Rokach and Maimon (2005), Xu and Wunsch (2005). In Al-Sultan (1995), Chu and Roddick (2000), Liu et al. (2005), Sung and Jin (2000) algorithms based on Tabu Search are presented. Simulated Annealing for clustering is used in Brown and Huntley (1992), Celeux and Govaert (1992), Chu and Roddick (2000), Selim and Alsultan (1991) while in Cano et al. (2002) a clustering algorithm based on Greedy Randomized Adaptive Search Procedure (GRASP) is applied. Genetic algorithms are used in Babu and Murty (1993), Cowgill et al. (1999), Liu et al. (2004), Maulik and Bandyopadhyay (2000), Meng et al. (2000), Sheng and Liu (2006), Tseng and Yang (2000), Tseng and Yang (2001), Wu et al. (2003), Yeh and Fu (2007) while an analytical review of the use of neural networks in clustering is given in Liao and Wen (2007). Clustering algorithms based on Ant Colony Optimization are used in Azzag et al.

(2006, 2007), Chen et al. (2005), He et al. (2006), Kao and Cheng (2006), Shelokar et al. (2004), Sherafat et al. (2004), Tsang and Kwong (2006), Yang and Kamel (2006) while in Janson and Merkle (2005), Kao et al. (2007), Paterlini and Krink (2006), Shen et al. (2005a), Sun et al. (2006) clustering algorithms based on Particle Swarm Optimization are applied. Clustering algorithms based on Artificial Immune Systems are presented in Li and Tan (2006), Nasraoui et al. (2003), Younsi and Wang (2004). A hybrid technique based on combining the K-means algorithm, Nelder-Mead simplex search (a classical local descent algorithm), and particle swarm optimization, called K-NM-PSO, is proposed in Kao et al. (2007).

Cluster validity analysis is the assessment of a clustering procedure's output. Effective evaluation standards and criteria are used in order to find the degree of confidence for the clustering results derived from the used algorithms. There are three types of validation studies: an external assessment of validity compares the recovered structure to an a priori structure, an internal examination of validity tries to determine if the structure is intrinsically appropriate for the data while a relative test compares two structures and measures their relative merit. External indices, internal indices, and relative indices are used for these tests (Jain et al. 1999; Xu and Wunsch 2005).

In the *results interpretation* step, experts in the relevant fields interpret the data partition in order to guarantee the reliability of the extracted knowledge.

In this paper, a new hybrid metaheuristic algorithm based on an Ant Colony Optimization (ACO) (Dorigo and Stutzle 2004) algorithm for the solution of the feature selection problem and on a Greedy Randomized Adaptive Search Procedure (GRASP) (Feo and Resende 1995) for the solution of the clustering problem is proposed. In order to assess the efficacy of the proposed algorithm, this methodology is evaluated on datasets from the UCI Machine Learning Repository. Also, the method is compared with the results of eight other metaheuristic algorithms for clustering analysis that use a Tabu Search Based Algorithm (Glover 1989) a Genetic Based Algorithm (Goldberg 1989) and a Particle Swarm Optimization Algorithm (Kennedy and Eberhart 1995) and combination of them with the proposed Ant Colony Optimization algorithm for the solution of the feature selection problem (Marinakis et al. 2008) or for the clustering phase of the algorithm.

The rest of this paper is organized as follows: In the next section the proposed Hybrid ACO-GRASP Algorithm is presented and analyzed in detail. In Sect. 3, the analytical computational results for the datasets taken from the UCI Machine Learning Repository are presented while in the last section conclusions and future research are given.

2 The proposed hybrid ACO-GRASP algorithm for clustering

2.1 Introduction

The proposed algorithm (Hybrid ACO-GRASP) for the solution of the clustering problem is a two phase algorithm which combines an Ant Colony Optimization (ACO) (Dorigo and Stutzle 2004) algorithm for the solution of the feature selection problem and a Greedy Randomized Adaptive Search Procedure (GRASP) for the solution of the clustering problem. In this algorithm, the activated features are calculated by the Ant Colony Optimization algorithm (see Sect. 2.4) and the fitness (quality) of each ant is calculated by the clustering algorithm (see Sect. 2.5).

The application of ACO in the feature selection problem is a research area that is still relatively unexplored. Al-Ani (2005a, 2005b) presented a novel feature subset search procedure that utilizes Ant Colony Optimization and used this procedure to select features for

speech segment and texture classification problems. Zhang and Hu (2005) proposed an algorithm which utilizes the combination of wrapper and filter models: ant colony optimization (ACO) and mutual information (MI). In both of these approaches a combination of a wrapper method, like ant colony optimization, and a filter evaluation function, like the mutual information function, is used. In our proposed ACO-based metaheuristic algorithm, a different approach is applied that does not use the mutual information evaluation function to estimate the local importance of each feature and this makes the algorithm more flexible and less time consuming.

There is only one application, at least to our knowledge, that uses the GRASP algorithm for clustering (Cano et al. 2002), where they used a very different approach than ours. More, precisely GRASP algorithm is used for the initial solutions and for each solution k-means algorithm is used for the calculation of the clusters.

In the following, initially the clustering problem is stated, then a general description of the proposed algorithm is given while in the last two subsections each of the phases of the algorithm are presented analytically.

2.2 The clustering problem

The problem of clustering N objects (patterns) into K clusters is considered. In particular the problem is stated as follows:

Given N objects in R^n, allocate each object to one of K clusters such that the sum of squared Euclidean distances between each object and the center of its belonging cluster (which is also to be found) for every such allocated object is minimized. The clustering problem can be mathematically described as follows:

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

Subject to

$$\sum_{j=1}^{K} w_{ij} = 1, \qquad i = 1,\ldots,N \tag{2}$$

$$w_{ij} = 0 \text{ or } 1, \qquad i = 1,\ldots,N, \quad j = 1,\ldots,K \tag{3}$$

where:

- K is the number of clusters (given or unknown),
- N is the number of objects (given),
- $x_i \in R^n$, $(i = 1,\ldots,N)$ is the location of the ith pattern (given),
- $z_j \in R^n$, $(j = 1,\ldots,K)$ is the center of the jth cluster (to be found), where

$$z_j = \frac{1}{N_j}\sum_{i=1}^{N} w_{ij}x_i \tag{4}$$

  where N_j is the number of objects in the jth cluster,
- w_{ij} is the association weight of pattern x_i with cluster j, (to be found), where

$$w_{ij} = \begin{cases} 1 & \text{if pattern } i \text{ is allocated to cluster } j, \\ 0 & \text{otherwise.} \end{cases} \qquad \forall i = 1,\ldots,N,\ j = 1,\ldots,K \tag{5}$$

2.3 General description of the algorithm

Initially, as it was mentioned in Sect. 2.1, in the first phase of the algorithm a number of features are activated, using the Ant Colony Optimization Algorithm. In order to find the clustering of the samples (fitness or quality of the ACO algorithm) a GRASP algorithm is used. The clustering algorithm has the possibility to solve the clustering problem with known or unknown number of clusters. When the number of clusters is known Equation (4), denoted as *SSE*, is used in order to find the best clustering. In the case that the number of clusters are unknown two additional measures are used. The one measure is the minimization of the distance between the centers of the clusters:

$$SSC = \sum_{i}^{K} \sum_{j}^{K} (\|z_i - z_j\|)^2. \tag{6}$$

The second measure is a the minimization of a validity index (Ray and Turi 1999; Shen et al. 2005b) given by:

$$validity = \frac{SSE}{SSC}. \tag{7}$$

It should, also, be noted that the selection of the best solution of the feature selection problem cannot be performed based on the sum of squared Euclidean distances because when the features are increased (or decreased) a number of terms are added (or subtracted) in (4) and the comparison of the solutions is not possible, using only the *SSE* measure. Thus, the validity measure is used. A pseudocode of the proposed algorithm is presented in Table 1.

2.4 Ant colony optimization for the feature subset selection problem

Feature selection is used as the first step of the clustering task in order to reduce the dimension of problem, decrease noise and improve the speed of the algorithm by the elimination of irrelevant or redundant features. In this paper, Ant Colony Optimization (Dorigo and Stutzle 2004) for the solution of the feature selection problem is used. The Ant Colony Optimization (ACO) metaheuristic is a relatively new technique for solving combinatorial optimization problems (COPs). Based strongly on the Ant System (AS) metaheuristic developed by Dorigo, Maniezzo and Colorni (Dorigo et al. 1996; Dorigo and Stutzle 2004), ant colony optimization is derived from the foraging behaviour of real ants in nature. The main idea of ACO is to model the problem as the search for a minimum cost path in a graph. Artificial ants walk through this graph, looking for good paths. Each ant has a rather simple behaviour so that it will typically only find rather poor-quality

Table 1 Hybrid ACO-GRASP for clustering

```
do while stopping criteria not satisfied
    call ACO algorithm(Solution)
    call GRASP algorithm(Solution)
    if Solution is better than Best_Solution_Found then
        Best_Solution_Found ⟵ Solution
    endif
enddo
return Best_Solution_Found
```

Table 2 Ant colony optimization

Initialization
Creation of the initial population of Ants
Calculation of heuristic function n_i for each feature i
Calculation of the initial pheromone τ_i for each feature i
Selection of the maximum number of generations
Main Phase
Do until the maximum number of generations has been reached:
    Each ant begins its own tour from a different feature
    For each ant in the population:
        Do while stopping criteria are not satisfied:
            Select the next feature based on the pheromone
                and the heuristic function
            Calculate the fitness function of each ant
                using the GRASP algorithm for clustering
            Apply a Local Search Procedure
        Enddo
    EndFor
    Update the pheromone based on the solution of the best ant in the population
    (Elitist Strategy for ACO)
Enddo
Return the best ant (the best solution).

paths on its own. Better paths are found as the emergent result of the global cooperation among ants in the colony (Dorigo et al. 1996).

An ACO algorithm consists of a number of cycles (iterations) of solution construction. During each iteration a number of ants (which is a parameter) construct complete solutions using heuristic information and the collected experiences of previous groups of ants (Dorigo and Stutzle 2004). These collected experiences are represented by a digital analogue of trail pheromone which is deposited on the constituent elements of a solution. Small quantities are deposited during the construction phase while larger amounts are deposited at the end of each iteration in proportion to solution quality. Pheromone can be deposited on the components and/or the connections used in a solution depending on the problem (Dorigo and Stutzle 2004). A pseudocode of the proposed Ant Colony Optimization algorithm is presented in Table 2.

In the proposed algorithm, every candidate feature in ACO is mapped into a binary ant where the bit 1 denotes that the corresponding feature is selected and the bit 0 denotes that the feature is not selected. An initial population r of solutions is formed in order to find an initial local optimum solution to use it in the calculation of the heuristic function n_i of the feature i. The n_i is calculated from the r_1 best solutions ($r_1 < r$) of the initial population. We would like to have an initial estimation of the most important features (Dorigo and Stutzle 2004). Thus, the features that exist in the r_1 best solutions are identified and all the features are weighted based on the times that each feature appears in the r_1 best solutions. These features have greater fixed value in the $[n_i]$ matrix, where $[\cdot]$ denotes the i element of the matrix n.

In the algorithm, a number of ants are used that start to construct solutions simultaneously. Each ant begins from a different place in the feature vector and follows its own route. Also, it has the possibility to visit all features and built solutions completely. Each ant is used for a number of generations starting always from the same feature and choosing in each generation different features based on the quantity of pheromone that exists in each feature. The initial quantity of the pheromone τ_i for the feature i is calculated from the formula (Dorigo

and Stutzle 2004):

$$\tau_i = \frac{ant\_size}{init\_opt} \tag{8}$$

where *ant_size* is the initial population of ants and *init_opt* is the quality of the optimum solution of the initial population.

An ant located in the feature j decides if the feature i is selected or not by the formula (Dorigo and Stutzle 2004):

$$p_i = \frac{[\tau_i]^\alpha [n_i]^\beta}{\sum_{l=1}^{m} [\tau_l]^\alpha [n_l]^\beta} \tag{9}$$

where m is the number of features, $[\cdot]$ denotes the i element of the matrices τ, n and α, β are two empirically selected parameters. If $\alpha = 0$ the features that are selected in the initial solutions are more likely to be selected and if $\beta = 0$ only pheromone is used without any heuristic information. Afterwards, the fitness of each ant is calculated using the GRASP algorithm for clustering and each ant chooses the next feature that will visit based on the previous formula.

In the proposed algorithm, another restriction is added. This restriction prunes the ability of each ant to create a path with all features activated. This is done because if all ants find a solution with all the features the result will be the same solutions for all ants. Off course, for each ant the optimal solution for all changes of the features is kept. When all ants have completed their paths a simple local search is applied in each ant in order to optimize the solutions. The local phase is very simple, features not activated in the current solution are now activated and vice versa in order to find a better solution.

When all ants have constructed their first solution, the pheromone trails are updated. A number of different approaches have been proposed for the pheromone update solutions (Dorigo and Stutzle 2004). In the proposed algorithm, only the best ant leaves pheromone in its own features (this strategy is called *Elitist Strategy for ACO*). Thus, the pheromone quantity of each feature becomes (Dorigo and Stutzle 2004):

$$\tau_i \leftarrow \begin{cases} (1-q)\tau_i + \frac{1}{ant\_opt}, & \text{if feature } i \text{ is selected} \\ (1-q)\tau_i, & \text{otherwise} \end{cases} \tag{10}$$

where *ant_opt* is the quality of the best ant and q is an evaporation parameter that is used in order not to have a continuous increase of the pheromone values in each feature. The parameter q is used to avoid unlimited accumulation of the pheromone trails and it enables the algorithm to forget bad decisions previously taken (Dorigo and Stutzle 2004).

2.5 Greedy randomized adaptive search procedure for the clustering problem

As it was mentioned earlier in the clustering phase of the proposed algorithm a Greedy Randomized Adaptive Search Procedure (GRASP) (Feo and Resende 1995; Marinakis et al. 2005a, 2005b; Resende and Ribeiro 2003) is used. GRASP is an iterative two phase search algorithm which has gained considerable popularity in combinatorial optimization. Each iteration consists of two phases, a construction phase and a local search phase. In the construction phase, a randomized greedy function is used to built up an initial solution. This randomized technique provides a feasible solution within each iteration. This solution is then exposed for improvement attempts in the local search phase. The final result is simply

Table 3 GRASP

```
do while stopping criteria not satisfied
Greedy Solution Phase
    Choose the number of clusters
    Choose the centers of the clusters randomly
    Find the distances of each sample for all centers of the clusters
    Order all the samples from smallest to the largest distance
    Construct the RCL by using the first D samples of the ordering
    Select randomly a sample for the RCL list
    Calculate the new centers of the clusters
    Readjust the RCL list
    if all samples have been clustered then
        Calculate the validation measures
    endif
Local Search Phase
    do while an improvement in the solution is occured
        Calculate the distance of each sample for all the centers of the clusters
        Assign a sample in a better cluster if possible
        Calculate the new centers of the clusters
        Calculate the validation measures
    enddo
    if Solution is better than Best_Solution_Found then
        Best_Solution_Found ⟵ Solution
    endif
enddo
return Best_Solution_Found
```

the best solution found over all iterations. In the first phase, a randomized greedy technique provides feasible solutions incorporating both greedy and random characteristics. This phase can be described as a process which stepwise adds one element at a time to the partial (incomplete) solution. The choice of the next element to be added is determined by ordering all elements in a candidate list with respect to a greedy function. The heuristic is adaptive because the benefits associated with every element are updated during each iteration of the construction phase to reflect the changes brought on by the selection of the previous element. The probabilistic component of a GRASP is characterized by randomly choosing one of the best candidates in the list but not necessarily the top candidate. The greedy algorithm is a simple one pass procedure for solving the clustering problem. In the second phase, a local search is initialized from the solution of the first phase, and the final result is simply the best solution found over all searches (cf. multi-start local search). A pseudocode of the proposed GRASP algorithm is presented in Table 3.

In the following the way the GRASP algorithm is applied for the solution of the clustering problem is analyzed in detail. An initially solution is constructed step by step (by the term initially solution we mean an initial clustering of the samples in the clusters) and, then, this solution is exposed for development in the local search phase of the algorithm. The first problem that we have to face was the selection of the number of the clusters. Thus, the algorithm works with two different ways.

If the number of clusters is known a priori, then a number of samples equal to the number of clusters are selected randomly as the initial clusters. In this case, as the iterations of GRASP increased the number of clusters do not change. In each iteration, different samples (equal to the number of clusters) are selected as initial clusters. Afterwards, the Restricted Candidate List (RCL) is created. The RCL parameter determines the level of greediness or randomness in the construction. In our implementation, the best promising candidate

samples are selected to create the RCL. The samples in the list are ordered taking into account the distance of each sample from all centers of the clusters and the ordering is from the smallest to the largest distance. From this list, the first D samples (D is a parameter of the problem) are selected in order to form the final Restricted Candidate List. This type of RCL is a cardinality based RCL. The candidate sample for inclusion in the solution is selected randomly from the RCL using a random number generator. Finally, the RCL is readjusted in every iteration by recalculated all the distances based on the new centers and replacing the sample which has been included in the sample by another sample that does not belong to the RCL, namely the $(D + m)$th sample where m is the number of the current iteration. When all the samples have been assigned to clusters three measures are calculated (the best solution is calculated based on the sum of squared Euclidean distances between each object and the center of its belonging cluster, see Sect. 2.3) and a local search strategy is applied in order to improve the solution. The local search works as follows: For each sample the probability of its reassignment in a different cluster is examined by calculating the distance of the sample from the centers. If a sample is reassigned to a different cluster the new centers are calculated. The local search phase stops when in an iteration no sample is reassigned.

If the number of clusters is unknown then initially a number of samples are selected randomly as the initial clusters. Now, as the iterations of GRASP increased the number of clusters changes and cannot become less than two. In each iteration a different number of clusters can be found. The creation of the initial solutions and the local search phase work as in the previous case. The only difference compared to the previous case concerns the use of the validity measure in order to choose the best solution because as we have different number of clusters in each iteration the sum of squared Euclidean distances varies significantly for each solution.

3 Computational results

3.1 Data and parameter description

The performance of the proposed methodology is tested on 12 benchmark instances taken from the UCI Machine Learning Repository. The datasets were chosen to include a wide range of domains and their characteristics are given in Table 4. The data varies in term of the number of observation from very small samples (Zoo with 101 observations) up to larger data sets (Spambase with 4601 observations). Also, there are data sets with two, three, four and seven clusters. In one case (Breast Cancer Wisconsin) the data set is appeared with different size of observations because in this data set there is a number of missing values. The problem of missing values was faced with two different ways. In the first way where all the observations are used we took the mean values of all the observations in the corresponding feature while in the second way where we have less values in the observations we did not take into account the observations that they had missing values. Some data sets involve only numerical features, and the remaining include both numerical and categorical features. For each data set, Table 4 reports the total number of features and the number of categorical features in parentheses.

The parameter settings for the Hybrid ACO-GRASP metaheuristic are:

- The number of ants used is equal to the number of features because in the initial iteration each ant begins from a different feature.
- The number of iterations that each ant constructs a different solution, based on the pheromone trails, is equal to 20.

Table 4 Data sets characteristics

Data sets	Observations	Features	Clusters
Australian Credit (AC)	690	14 (8)	2
Breast Cancer Wisconsin 1 (BCW1)	699	9	2
Breast Cancer Wisconsin 2 (BCW2)	683	9	2
Heart Disease (HD)	270	13 (7)	2
Hepatitis 1 (Hep1)	155	19 (13)	2
Ionosphere (Ion)	351	34	2
Spambase (spam)	4601	57	2
Iris	150	4	3
Wine	178	13	3
Zoo	101	17	7
Vehicle (Veh)	848	18	4
Segment (Seg)	2310	19	7

– $q = 0.5$.
– The size of RCL varies between 30 and 150.
– The number of GRASP's iterations is equal to 100.

The algorithm was implemented in Fortran 90 and was compiled using the Lahey f95 compiler on a Centrino Mobile Intel Pentium M 750 at 1.86 GHz, running Suse Linux 9.1.

3.2 Results of the proposed algorithm

The objective of the computational experiments is to show the performance of the proposed algorithm in searching for a reduced set of features with high clustering of the data. The purpose of feature variable selection is to find the smallest set of features that can result in satisfactory predictive performance. Because of the curse of dimensionality, it is often necessary and beneficial to limit the number of input features in order to have a good predictive and less computationally intensive model. In general there are $2^{number\ of\ features} - 1$ possible feature combinations and, thus, in our cases the problem with the fewest number of feature combinations is the Iris (namely $2^4 - 1$), while the most difficult problem is the Spambase where the number of feature combinations is $2^{57} - 1$.

The results of the proposed algorithm are given in Table 5. After the selection of the final parameters, 10 different runs with the selected parameters were performed for each of the datasets. In Table 5, the results of the best run of the proposed algorithm are given in the second column, while in the third column the average results of the 10 runs of the algorithm are presented, and in the last column the variance of the corrected clustered samples is presented. As it can be seen, the proposed algorithm has achieved a very good performance concerning the clustered samples. Also, taking into account the average and the variance, it should be noted that there are no significant differences in the obtained results.

A comparison with the classic k-means and other metaheuristic approaches for the solution of the clustering problem is presented in Table 6. In this table, eight other algorithms are used for the solution of the feature subset selection problem and for the clustering problem. In the first one in both phases (feature selection phase and clustering phase) an Ant Colony Optimization algorithm is used (columns 4 and 5 of Table 6 of the first group) while in the second one a Particle Swarm Optimization in both phases (feature selection phase and clustering phase) algorithm is used (columns 6 and 7 of Table 6 of the first group).

Table 5 Results of the algorithm

Instance	Corrected clustered		
	Best	Average	Variance
BCW2	662(96.92%)	659.7(96.58%)	2.9
Hep1	134(86.45%)	132.2(85.29%)	1.73
AC	603(87.39%)	601.2(87.13%)	2.62
BCW1	676(96.70%)	673.8(96.39%)	3.51
Ion	291(82.90%)	288.3(82.13%)	2.45
spam	3993(86.78%)	3991.1(86.74%)	4.32
HD	232(85.92%)	230.1(85.22%)	2.76
Iris	145(96.67%)	143.3(95.53%)	2.67
Wine	176(98.87%)	175.3(98.48%)	0.45
Zoo	95(94.05%)	93.9(92.97%)	1.43
Veh	804(94.81%)	802.1(94.58%)	2.54
Seg	2108(91.25%)	2105.8(91.16%)	6.17

Subsequently, in the second group of algorithms and columns 2 and 3 of Table 6 a genetic algorithm (Goldberg 1989) is used in the first phase of the algorithm while a Greedy Randomized Adaptive Search Procedure is used in the second phase of the algorithm. In the second group and in columns 4 and 5 of Table 6 a Tabu Search Algorithm (Glover 1989) is used in the first phase and a GRASP algorithm is used in the second phase. The classic k-means algorithm is used for the clustering problem using all features (columns 6 and 7 of Table 6 of the second group). Finally, in the third group of algorithms and columns 2 and 3 of Table 6 the proposed ACO is used in the first phase of the algorithm while a Genetic Algorithm is used in the second phase of the algorithm. In the third group and in columns 4 and 5 of Table 6 the proposed ACO Algorithm is used in the first phase and a Tabu Search algorithm is used in the second phase. Finally, a genetic algorithm is used in both phases of the problem (columns 6 and 7 of Table 6 of the third group).

The Tabu Search (Glover 1989, 1990) is running for 1000 iterations and with size of the Tabu List equal to 10. The Genetic algorithm (Goldberg 1989; Holland 1975) is running for 20 generations, having a population size equal to 500, and using a single 1-point crossover operator with probability equal to 0.8 and a mutation operator with a probability equal to 0.25. In the clustering phase of these two algorithms the GRASP algorithm is used. For the Particle Swarm Optimization (Kennedy and Eberhart 1995) algorithm: the number of swarms is set equal to 1, the number of particles is set equal to 50, the number of generations is set equal to 50 and the coefficients are $c_1 = 2$, $c_2 = 2$, $w_{\max} = 0.9$ and $w_{\min} = 0.01$.

From this table it can be observed that the Hybrid ACO-GRASP algorithm performs better than the other algorithms in all instances. For this method the percentage of the correct clustered samples varies between 82.90% to 98.87%, while for the Genetic-GRASP algorithm the percentage of the correct clustered samples varies between 75.78% to 98.31%, for the Tabu-GRASP algorithm the percentage of the correct clustered samples varies between 74.92% to 97.75%, for the ACO-Genetic the percentage of the correct clustered samples varies between 76.92% to 97.75%, for the ACO-Tabu the percentage of the correct clustered samples varies between 75.49% to 97.75%, for the ACO the percentage of the correct clustered samples varies between 73.50% to 97.75%, for the Genetic the percentage of the correct clustered samples varies between 73.21% to 97.75%, for the PSO the percentage of the correct clustered samples varies between 74.35% to 97.75% and for the k-means the

Table 6 Comparisons of the algorithm with other algorithms

Instance	ACO-GRASP		ACO		PSO	
	Sel. feat.	Correct clustered	Sel. feat.	Correct clustered	Sel. feat.	Correct clustered
BCW2	5	662(96.92%)	5	662(96.92%)	5	662(96.92%)
Hep1	9	134(86.45%)	9	133(85.80%)	10	132(85.16%)
AC	8	603(87.39%)	8	601(87.10%)	8	602(87.24%)
BCW1	5	676(96.70%)	8	674(96.42%)	8	674(96.42%)
Ion	2	291(82.90%)	16	258(73.50%)	12	261(74.35%)
spam	56	3993(86.78%)	41	3967(86.22%)	37	3960(86.06%)
HD	9	232(85.92%)	9	227(84.07%)	9	227(84.07%)
Iris	3	145(96.67%)	3	145(96.67%)	3	145(96.67%)
Wine	7	176(98.87%)	7	174(97.75%)	7	174(97.75%)
Zoo	6	95(94.05%)	8	89(88.11%)	11	88(87.12%)
Veh	5	804(94.81%)	11	751(88.56%)	13	754(88.91%)
Seg	7	2108(91.25%)	12	2018(87.35%)	15	2026(87.70%)
Instance	Genetic-GRASP		Tabu-GRASP		*k*-Means	
	Sel. feat.	Correct clustered	Sel. feat.	Correct clustered	Sel feat.	Correct clustered
BCW2	5	662(96.92%)	6	661(96.77%)	9	654(95.74%)
Hep1	9	134(86.45%)	10	132(85.16%)	19	121(78.06%)
AC	8	602(87.24%)	9	599(86.81%)	14	580(84.05%)
BCW1	5	676(96.70%)	8	674(96.42%)	9	672(96.13%)
Ion	17	266(75.78%)	4	263(74.92%)	34	248(70.65%)
spam	56	3938(85.59%)	34	3810(82.80%)	57	3958(86.02%)
HD	7	231(85.55%)	9	227(84.07%)	13	220(81.48%)
Iris	4	145(96.67%)	3	145(96.67%)	4	144(96%)
Wine	7	175(98.31%)	7	174(97.75%)	13	172(96.92%)
Zoo	10	90(89.10%)	9	91(90.09%)	17	82(81.18%)
Veh	6	733(86.43%)	8	723(85.25%)	18	718(84.66%)
Seg	11	2044(88.48%)	13	2031(87.92%)	19	1954(84.58%)
Instance	ACO-Genetic		ACO-Tabu		Genetic	
	Sel. feat.	Correct clustered	Sel. feat.	Correct clustered	Sel feat.	Correct clustered
BCW2	5	662(96.92%)	5	662(96.92%)	7	658(96.33%)
Hep1	9	134(86.45%)	9	134(86.45%)	8	132(85.16%)
AC	9	600(86.95%)	9	599(86.81%)	10	597(86.52%)
BCW1	5	675(96.56%)	8	674(96.42%)	7	672(96.13%)
Ion	11	270(76.92%)	9	265(75.49%)	12	257(73.21%)
spam	38	3971(86.30%)	35	3960(86.06%)	41	3944(85.70%)
HD	11	229(84.18%)	10	228(84.44%)	8	224(82.96%)
Iris	3	145(96.67%)	3	145(96.67%)	3	145(96.67%)
Wine	7	174(97.75%)	7	174(97.75%)	7	174(97.75%)
Zoo	7	92(91.08%)	9	90(89.10%)	12	84(83.16%)
Veh	6	773(91.15%)	8	769(90.68%)	10	731(86.20%)
Seg	9	2078(89.95%)	7	2069(89.56%)	8	2018(87.35%)

percentage of the correct clustered samples varies between 70.65% to 96.92%. It should be mentioned that in some instances the differences in the results between the Hybrid ACO-GRASP algorithm and the other algorithms are very significant.

Mainly, for the four data sets that have the largest number of features and the largest number of clusters compared to the other data sets, i.e. in the Ionosphere data set the percentage of the Hybrid ACO-GRASP algorithm is 82.90%, for the ACO-Genetic algorithm is 76.92%, for the Genetic-GRASP algorithm is 75.78%, for the ACO-Tabu algorithm is 75.49%, for the Tabu-GRASP algorithm is 74.92%, for the PSO is 74.35%, for the ACO is 73.50%, for the Genetic is 73.21% and for the k-means is 70.65%, in the Spambase data set the percentage of the Hybrid ACO-GRASP algorithm is 86.78%, for the ACO-Genetic is 76.30%, for the ACO is 86.22%, for the PSO is 86.06%, for the ACO-Tabu is 86.06%, for the k-means is 86.02%, for the Genetic is 85.70%, for the Genetic-GRASP algorithm is 85.59% and for the Tabu-GRASP algorithm is 82.80%, in the Vehicle data set the percentage of the Hybrid ACO-GRASP algorithm is 94.81%, for the ACO-Genetic is 91.15%, for the ACO-Tabu is 90.68%, for the PSO is 88.91%, for the ACO is 88.56%, for the Genetic-GRASP algorithm is 86.43%, for the Genetic is 86.20%, for the Tabu-GRASP algorithm is 85.25% and for the k-means is 84.66% and in the Segment data set the percentage of the Hybrid ACO-GRASP algorithm is 91.25%, for the ACO-Genetic is 89.95%, for the ACO-Tabu is 89.56%, for the Genetic-GRASP algorithm is 88.48%, for the Tabu-GRASP algorithm is 87.92%, for the PSO is 87.70%, for the ACO is 87.35%, for the Genetic is 87.35% and for the k-means is 84.58%. These results prove the significance of the solution of the feature selection problem in the clustering algorithm as when a more sophisticated method (ACO) for the solution of this problem was used the performance of the clustering algorithm was improved. An important observation is the improvement that gives in the results the combination of ACO for feature selection and GRASP for clustering. We use the proposed ACO algorithm in the feature selection phase of the algorithm and in one case a Genetic algorithm in the clustering phase and in another case a Tabu Search algorithm in the clustering phase. From this table it can be seen that the results of these algorithms are inferior compared to the results of the proposed algorithm. This proves that each phase of the proposed algorithm is necessary and equally important for the efficient performance of the algorithm.

It should, also, be mentioned that the algorithm was tested with two options: with known and unknown number of clusters. In case that when the number of clusters was unknown and thus in each iteration of the algorithm different initial values of clusters were selected the algorithm always converged to the optimal number of clusters and with the same results as in the case that the number of clusters was known.

4 Conclusions and future research

In this paper a new metaheuristic algorithm, the Hybrid ACO-GRASP, is proposed for solving the Clustering Problem. This algorithm is a two phase algorithm which combines an Ant Colony Optimization (ACO) algorithm for the solution of the feature selection problem and a Greedy Randomized Adaptive Search Procedure (GRASP) for the solution of the clustering problem. A number of metaheuristic algorithms and the classic k-means were also used for comparison purposes. The performance of the proposed algorithms is tested using various benchmark datasets from UCI Machine Learning Repository. The objective of the computational experiments, the desire to show the high performance of the proposed algorithms, was achieved as the algorithms gave very efficient results. The significance of the solution of the clustering problem by the proposed algorithm is proved by the fact that the

percentage of the correct clustered samples is very high and in some instances is larger than 96%. Also, the focus in the significance of the solution of the feature selection problem is proved by the fact that the instances with the largest number of features gave better results when the ACO algorithm was used. Future research is intended to be focused in using different algorithms (e.g. Particle Swarm Optimization) both to the feature selection phase and to the clustering algorithm phase.

References

Aha, D. W., & Bankert, R. L. (1996). A comparative evaluation of sequential feature selection algorithms. In D. Fisher & J.-H. Lenx (Eds.), *Artificial intelligence and statistics*. New York: Springer.

Al-Ani, A. (2005a). Feature subset selection using ant colony optimization. *International Journal of Computational Intelligence*, *2*(1), 53–58.

Al-Ani, A. (2005b). Ant colony optimization for feature subset selection. *Transactions on Engineering, Computing and Technology*, *4*, 35–38.

Al-Sultan, K. (1995). A tabu search approach to the clustering problem. *Pattern Recognition*, *28*(9), 1443–1451.

Azzag, H., Guinot, C., Venturini, G. (2006). Data and text mining with hierarchical clustering ants. In A. Abraham, C. Grosan & V. Ramos (Eds.), *Swarm intelligence in data mining* (pp. 153–190).

Azzag, H., Venturini, G., Oliver, A., & Gu, C. (2007). A hierarchical ant based clustering algorithm and its use in three real-world applications. *European Journal of Operational Research*, *179*, 906–922.

Babu, G., & Murty, M. (1993). A near-optimal initial seed value selection in *k*-means algorithm using a genetic algorithm. *Pattern Recognition Letters*, *14*(10), 763–769.

Brown, D., & Huntley, C. (1992). A practical application of simulated annealing to clustering. *Pattern Recognition*, *25*(4), 401–412.

Cano, J. R., Cordón, O., Herrera, F., & Sánchez, L. (2002). A GRASP algorithm for clustering. In F. J. Garijo, J. C. Riquelme, & M. Toro (Eds.), *LNAI: Vol. 2527. IBERAMIA 2002* (pp. 214–223). Berlin: Springer.

Cantu-Paz, E., Newsam, S., & Kamath, C. (2004). Feature selection in scientific application. In *Proceedings of the 2004 ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 788–793).

Celeux, G., & Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, *14*, 315–332.

Chen, L., Tu, L., & Chen, H. (2005). A novel ant clustering algorithm with digraph. In L. Wang, K. Chen, & Y. S. Ong (Eds.), *LNCS: Vol. 3611. ICNC 2005* (pp. 1218–1228). Berlin: Springer.

Chu, S., & Roddick, J. (2000). A clustering algorithm using the tabu search approach with simulated annealing, In N. Ebecken & C. Brebbia (Eds.), *Data mining II-proceedings of second international conference on data mining methods and databases* (pp. 515–523), Cambridge, U.K.

Cowgill, M., Harvey, R., & Watson, L. (1999). A genetic algorithm approach to cluster analysis. *Computers and Mathematics with Applications*, *37*, 99–108.

Dorigo, M., & Stutzle, T. (2004). *Ant colony optimization*. Cambridge: Bradford Book.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, *26*(1), 29–41.

Feo, T. A., & Resende, M. G. C. (1995). Greedy randomized adaptive search procedure. *Journal of Global Optimization*, *6*, 109–133.

Glover, F. (1989). Tabu search I. *ORSA Journal on Computing*, *1*(3), 190–206.

Glover, F. (1990). Tabu search II. *ORSA Journal on Computing*, *2*(1), 4–32.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Massachusetts: Addison-Wesley.

He, Y., Hui, S. C., & Sim, Y. (2006). A novel ant-based clustering approach for document clustering. In H. T. Ng et al. (Eds.), *LNCS: Vol. 4182. AIRS 2006* (pp. 537–544). Berlin: Springer.

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.

Jain, A., & Zongker, D. (1997). Feature selection: evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 153–158.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, *31*(3), 264–323.

Janson, S., & Merkle, D. (2005). A new multi-objective particle swarm optimization algorithm using clustering applied to automated docking. In M. J. Blesa et al. (Eds.), *LNCS: Vol. 3636. HM 2005* (pp. 128–141). Berlin: Springer.

Jouve, P.-E., & Nicoloyannis, N. (2005). A filter feature selection method for clustering. In M.-S. Hacid et al. (Eds.), *LNAI: Vol. 3488. ISMIS 2005* (pp. 583–593). Berlin: Springer.

Kao, Y., & Cheng, K. (2006). An ACO-based clustering algorithm. In M. Dorigo et al. (Eds.), *LNCS: Vol. 4150. ANTS 2006* (pp. 340–347). Berlin: Springer.

Kao, Y.-T., Zahara, E., & Kao, I.-W. (2007). A hybridized approach to data clustering. *Expert Systems with Applications*. doi:10.1016/j.eswa.2007.01.028.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of 1995 IEEE international conference on neural networks* (Vol. 4, pp. 1942–1948).

Kira, K., & Rendell, L. (1992). A practical approach to feature selection. In *Proceedings of the ninth international conference on machine learning*, Aberdeen, Scotland (pp. 249–256).

Li, Z., & Tan, H.-Z. (2006). A combinational clustering method based on artificial immune system and support vector machine. In B. Gabrys, R. J. Howlett, & L. C. Jain (Eds.), *LNAI: Vol. 4251. KES 2006, Part I* (pp. 153–162). Berlin: Springer.

Liao, S.-H., & Wen, C.-H. (2007). Artificial neural networks classification and clustering of methodologies and applications – literature analysis from 1995 to 2005. *Expert Systems with Applications*, *32*, 1–11.

Liu, Y., Chen, K., Liao, X., & Zhang, W. (2004). A genetic clustering method for intrusion detection. *Pattern Recognition*, *37*, 927–942.

Liu, Y., Liu, Y., Wang, L., & Chen, K. (2005). A hybrid tabu search based clustering algorithm. In R. Khosla et al. (Eds.), *LNAI: Vol. 3682. KES 2005* (pp. 186–192). Berlin: Springer.

Marinakis, Y., Migdalas, A., & Pardalos, P. M. (2005a). Expanding neighborhood GRASP for the traveling salesman problem. *Computational Optimization and Applications*, *32*, 231–257.

Marinakis, Y., Migdalas, A., & Pardalos, P. M. (2005b). A hybrid genetic-GRASP algorithm using Langrangean relaxation for the traveling salesman problem. *Journal of Combinatorial Optimization*, *10*, 311–326.

Marinakis, Y., Marinaki, M., Doumpos, M., Matsatsinis, N., & Zopounidis, C. (2008). Optimization of nearest neighbor classifiers via metaheuristic algorithms for credit risk assessment. *Journal of Global Optimization*, *42*, 279–293.

Maulik, U., & Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern Recognition*, *33*, 1455–1465.

Meng, L., Wu, Q. H., & Yong, Z. Z. (2000). A faster genetic clustering algorithm. In S. Cagnoni et al. (Eds.), *LNCS: Vol. 1803. EvoWorkshops 2000* (pp. 22–33). Berlin: Springer.

Mirkin, B. (1996). *Mathematical classification and clustering*. Dordrecht: Kluwer.

Nasraoui, O., Gonzalez, F., Cardona, C., Rojas, C., & Dasgupta, D. (2003). A scalable artificial immune system model for dynamic unsupervised learning. In E. Cantú-Paz et al. (Eds.), *LNCS: Vol. 2723. GECCO 2003* (pp. 219–230). Berlin: Springer.

Ng, M. K. (2000). A note on constrained k-means algorithms. *Pattern Recognition*, *33*, 515–519.

Paterlini, S., & Krink, T. (2006). Differential evolution and particle swarm optimization in partitional clustering. *Computational Statistics and Data Analysis*, *50*, 1220–1247.

Ray, S., & Turi, R. H. (1999). Determination of number of clusters in K-means clustering and application in colour image segmentation. In *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques (ICAPRDT99)*, Calcutta, India.

Resende, M. G. C., & Ribeiro, C. C. (2003). Greedy randomized adaptive search procedures. In F. Glover & G. A. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 219–249). Boston: Kluwer.

Rokach, L., & Maimon, O. (2005). Clustering methods. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 321–352). New York: Springer.

Selim, S., & Alsultan, K. (1991). A simulated annealing algorithm for the clustering problems. *Pattern Recognition*, *24*(10), 1003–1008.

Shelokar, P. S., Jayaraman, V. K., & Kulkarni, B. D. (2004). An ant colony approach for clustering. *Analytica Chimica Acta*, *509*, 187–195.

Shen, H.-Y., Peng, X.-Q., Wang, J.-N., & Hu, Z.-K. (2005a). A mountain clustering based on improved PSO algorithm. In L. Wang, K. Chen, & Y. S. Ong (Eds.), *LNCS: Vol. 3612. ICNC 2005* (pp. 477–481). Berlin: Springer.

Shen, J., Chang, S. I., Lee, E. S., Deng, Y., & Brown, S. J. (2005b). Determination of cluster number in clustering microarray data. *Applied Mathematics and Computation*, *169*, 1172–1185.

Sheng, W., & Liu, X. (2006). A genetic k-medoids clustering algorithm. *Journal of Heuristics*, *12*, 447–466.

Sherafat, V., Nunes de Castro, L., & Hruschka, E. R. (2004). TermitAnt: an ant clustering algorithm improved by ideas from termite colonies. In N. R. Pal et al. (Eds.), *LNCS: Vol. 3316. ICONIP 2004* (pp. 1088–1093). Berlin: Springer.

Sun, J., Xu, W., & Ye, B. (2006). Quantum-behaved particle swarm optimization clustering algorithm. In X. Li, O. R. Zaiane, & Z. Li (Eds.), *LNAI: Vol. 4093. ADMA 2006* (pp. 340–347). Berlin: Springer.

Sung, C., & Jin, H. (2000). A tabu-search-based heuristic for clustering. *Pattern Recognition*, *33*, 849–858.

Tarsitano, A. (2003). A computational study of several relocation methods for k-means algorithms. *Pattern Recognition*, *36*, 2955–2966.

Tsang, C.-H., & Kwong, S. (2006). Ant colony clustering and feature extraction for anomaly intrusion detection. *Studies in Computational Intelligence (SCI)*, *34*, 101–123.

Tseng, L., & Yang, S. (2000). A genetic clustering algorithm for data with non-spherical-shape clusters. *Pattern Recognition*, *33*, 1251–1259.

Tseng, L., & Yang, S. (2001). A genetic approach to the automatic clustering problem. *Pattern Recognition*, *34*, 415–424.

Wu, F.-X., Zhang, W. J., & Kusalik, A. J. (2003). A genetic k-means clustering algorithm applied to gene expression data. In Y. Xiang & B. Chaib-draa (Eds.), *LNAI: Vol. 2671. AI 2003* (pp. 520–526). Berlin: Springer.

Xu, R., & Wunsch, D., II (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*(3), 645–678.

Yang, Y., & Kamel, M. S. (2006). An aggregated clustering approach using multi-ant colonies algorithms. *Pattern Recognition*, *39*, 1278–1289.

Yeh, J.-Y., & Fu, J. C. (2007). A hierarchical genetic algorithm for segmentation of multi-spectral human-brain MRI. *Expert Systems with Applications*. doi:10.1016/j.eswa.2006.12.012.

Younsi, R., & Wang, W. (2004). A new artificial immune system algorithm for clustering. In Z. R. Yang et al. (Eds.), *LNCS: Vol. 3177. IDEAL 2004* (pp. 58–64). Berlin: Springer.

Zhang, C., & Hu, H. (2005). Ant colony optimization combining with mutual information for feature selection in support vector machines. In S. Zhang & R. Jarvis (Eds.), *LNAI: Vol. 3809. AI 2005* (pp. 918–921). Berlin: Springer.