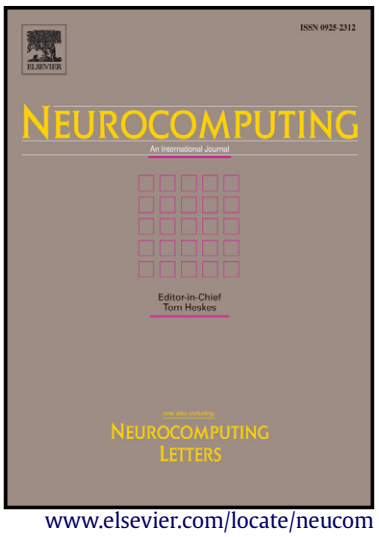

www.elsevier.com/locate/neucom

An advanced ACO algorithm for feature subset selection

Shima Kashef, Hossein Nezamabadi-pour

PII: S0925-2312(14)00860-1
DOI: http://dx.doi.org/10.1016/j.neucom.2014.06.067
Reference: NEUCOM14401

To appear in: *Neurocomputing*

Received date: 25 January 2014
Revised date: 25 April 2014
Accepted date: 23 June 2014

Cite this article as: Shima Kashef, Hossein Nezamabadi-pour, An advanced ACO algorithm for feature subset selection, *Neurocomputing*, http://dx.doi.org/10.1016/j.neucom.2014.06.067

An Advanced ACO Algorithm for Feature subset Selection

Shima Kashef and Hossein Nezamabadi-pour

Department of Electrical Engineering, Shahid Bahonar University of Kerman, P.O. Box 76619-133, Kerman, Iran

Tel & Fax: +98 341 3235900

shkashef.1988@yahoo.com; nezam@uk.ac.ir

Abstract—Feature selection is an important task for data analysis and information retrieval processing, pattern classification systems, and data mining applications. It reduces the number of features by removing noisy, irrelevant and redundant data. In this paper, a novel feature selection algorithm based on Ant Colony Optimization (ACO), called Advanced Binary ACO (ABACO), is presented. Features are treated as graph nodes to construct a graph model and are fully connected to each other. In this graph, each node has two sub-nodes, one for selecting and the other for deselecting the feature. Ant colony algorithm is used to select nodes while ants should visit all features. The use of several statistical measures is examined as the heuristic function for visibility of the edges in the graph. At the end of a tour, each ant has a binary vector with the same length as the number of features, where 1 implies selecting and 0 implies deselecting the corresponding feature. The performance of proposed algorithm is compared to the performance of Binary Genetic Algorithm (BGA), Binary Particle Swarm Optimization (BPSO), CatfishBPSO, Improved Binary Gravitational Search Algorithm (IBGSA), and some prominent ACO-based algorithms on the task of feature selection on 12 well-known UCI datasets. Simulation results verify that the algorithm provides a suitable feature subset with good classification accuracy using a smaller feature set than competing feature selection methods.

Keywords- Feature selection; Wrapper; Ant colony optimization (ACO); Binary ACO; Classification.

1. Introduction

Feature selection (FS) is generally used in machine learning, especially when the learning task involves high-dimensional datasets. The primary purpose of feature selection is to choose a subset of available features, by

eliminating features with little or no predictive information and also redundant features that are strongly correlated [1]. The availability of large amounts of data represents a challenge to classification analysis. For example, the use of many features may require the estimation of a considerable number of parameters during the classification process. Ideally, each feature used in the classification process should add an independent set of information. Often, however, features are highly correlated, and this can suggest a degree of redundancy in the available information which may have a negative impact on classification accuracy [2].

The FS approaches can generally be divided into three groups: filter, wrapper, and hybrid approaches [3]. The filter approach operates independently of any learning algorithm. These methods rank the features by some criteria and omit all features that do not achieve a sufficient score. Due to its computational efficiency, the filter methods are very popular to high-dimension data. Some popular filter methods are F-score criterion [4], mutual information [5], information gain [6] and correlation [7]. Reference [8] used the mutual information and F-score criterion to select the optimal feature recognition method. The wrapper approach involves with the predetermined learning model, selects features on measuring the learning performance of the particular learning model [7, 9]. Although wrappers may produce better results, they are expensive to run and can break down with very large numbers of features. This is due to the use of learning algorithms in the evaluation of feature subsets every time [10]. Filter and wrapper are two complementary approaches, then the hybrid approach attempts to take advantage of the filter and wrapper approaches by exploiting their complementary strengths [11, 12, 13].

Feature selection algorithms have been reviewed in [3, 14, 15]. For a large number of features, evaluating all states is computationally non-feasible requiring metaheuristic search methods. More recently, nature inspired metaheuristic algorithms have been used to select features, namely: particle swarm optimization (PSO) [16, 17, 18], genetic algorithm (GA)-based attribute reduction [19], gravitational search algorithm (GSA) [20, 21, 22]. These methods attempt to achieve better solutions by application of knowledge from previous iterations.

Ant colony optimization (ACO) [23, 24] is another promising approach to solve the combinational optimization problems and has been widely employed in feature selection [6, 25]. It was initially used for solving Traveling Salesman Problem (TSP) [26, 27] and then has been successfully applied to a large number of NP-hard problems such as Quadratic Assignment Problem (QAP), vehicle routing, system fault detecting, scheduling, etc. [28]. In recent years, some ACO-based methods for feature selecting are reported. Reference [25] introduces two terms, i.e., “update selection measure (USM)” and “local importance (LI)” in the ACO-based feature selection method. The hybrid of ACO and mutual information has been used for feature selection in the forecaster [29]. Aghdam et al. [30] proposed an ACO based feature selection algorithm for text categorization.

Basiri et al. [10] proposed an ACO algorithm for feature selection in prediction post-synaptic activity of proteins. Later, they hybridized ACO and genetic algorithms to obtain their excellent features by synthesizing them [31]. Vieira [1] presents an algorithm for feature selection based on two cooperative ant colonies, which minimizes two objectives: the number of features and the classification error. Two pheromone matrices and two different heuristics are used for these objectives. Xiong et al. [32] proposed a hybrid feature selection algorithm based on dynamic ant colony algorithm. Mutual information is taken as heuristic function. Chen et al. proposed a new rough set approach to feature selection based on ACO, which adopts mutual information based feature significance as heuristic information [33].

There are two criteria for stopping the search through the space of feature subsets in these methods. Some of the methods ask the user to predefine the number of selected features. Other methods are based on the evaluation function. In these methods, an optimal subset according to some evaluation strategy is obtained. As soon as the stopping criterion is met, searching through features is stopped [25]. Therefore, ants are only allowed to have a limited number of steps and cannot see all features. To solve this problem reference [34] for the first time suggests to use the binary form of ACO for feature selection, called BACO, in which ants visit all features one by one and are allowed to select a feature or not. Although this method could solve the problem of ACO, there is still a problem with this approach. In BACO algorithm ants traverse a constant sequence of features. Therefore, they can only judge to select or deselect the subsequent feature and are not able to track any desired sequence of unseen features. This limitation reduces the exploration of search and leads to non-optimal solutions; although the results are much better than ACO.

In our previous work [35], we have proposed a new ACO-based FS algorithm, called ABACO. This algorithm is an advanced version of binary ant colony optimization, which attempts to solve the problems of ACO and BACO algorithms by combination of these two. It gives ants the ability of a comprehensive view of features, and helps them to select the most salient features. In this paper, we intend to extend this algorithm by adding heuristic desirability. At each step, ants should be able to visit all unseen features, but they can judge to select a feature or not. It means there are two roads ending to each feature; one for selecting and the other for deselecting that feature. This is the difference between the proposed algorithm and the traditional ACO algorithms, which led to better results compared to them and other metaheuristic algorithms like BGA and BPSO. K-nearest neighbor (k-NN) classifier performance is regarded as evaluation criteria of the feature subset, and then feature pheromone is computed and updated according to the evaluation results. Experimental results show that this algorithm has high classification accuracy and can effectively reduce the number of features.

The paper is organized as follows. Section 2 summarizes ant colony algorithm and its binary form. Our proposed ABACO is discussed elaborately in Section 3. Section 4 presents the results of our experimental studies. Finally, Section 5 concludes the paper with a brief summary and a few remarks.

2. Ant colony optimization

Ant colony optimization is a metaheuristic algorithm which was inspired by the foraging behavior of real ants. ACO was introduced by Dorigo and his colleagues for the solution of hard combinational optimization (CO) problems in the early 1990s [23]. When a food source is found, ants lay some pheromone to mark the path. The quantity of the laid pheromone depends on the distance, quantity and quality of the food source. While an isolated ant moves essentially at random, an ant encounter the previously laid trail can detect it and decide with high probability to follow it, thus reinforcing the trail with its own pheromone. The process is thus characterized by a positive feedback loop, where the more are the ants following a trail, the more that trail becomes attractive for being followed. This indirect communication between the ants via pheromone trails enables them to find the shortest path between the food source and their nest [24].

Artificial ants, also referred as agents, imitate their natural counterparts and find the optimal solutions to the problems. They lay pheromone on edges of the graph and they choose their path with respect to probabilities that depend on pheromone trails that have been previously laid by other ants. These pheromone trails progressively decrease by evaporation.

Artificial ants also have some extra features that are not found in real ants. Each ant contains an internal memory, which is used to store their previous actions, and they may have some characteristics such as local search, to improve the quality of computed paths. Based on the problem for which the algorithm is designed, daemon actions may be introduced into the algorithm to speed up convergence. An example of such actions is to deposit additional pheromone on the states of the global best solutions. Finally, the movement of ants is guided by two factors: the pheromone value and the problem-specific local heuristic [24].

2.1. Ant colony optimization for feature selection

As mentioned earlier, given the original set of size n, feature selection problem is to find a minimal subset of salient features of size p ($p<n$), such that the classification accuracy is maximized. The optimization capability of ACO can be used to select features. Each of the original features is treated as a graph node to construct graph G,

and then search feature subset based on this graph. Nodes are fully connected to allow any feature to be selected next.

ACO algorithms are stochastic algorithms that make probabilistic decision in terms of the artificial pheromone trails (i.e. history of previous successful moves) and the local heuristic information (expressing desirability of the move/ visibility of the edge). These two factors are combined to form the so-called probabilistic transition rule:

$$P_{ij}^{k}(t)=\begin{cases}\dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{l}\tau_{il}^{\alpha}\eta_{il}^{\beta}} & If\ l\ and\ j\ are\ admissible\ nodes\\ 0 & otherwise\end{cases} \quad (1)$$

where $P_{ij}^{k}(t)$ denotes the transition probability from feature (node) i to feature (node) j for the k-th ant at time step t, τ_{ij} is the amount of pheromone trail on edge (i,j) at time t; η_{ij} is the heuristic desirability or visibility of edge (i,j); α and β are two parameters that control the relative importance of the pheromone value versus the heuristic information.

The solution space is initially empty and is expanded by adding a solution component at every probabilistic decision. The transition probability used by ACO is a balance between pheromone intensity (i.e. history of previous successful moves), τ_{ij}, and heuristic information (expressing desirability of the move), η_{ij}. This effectively balances the exploitation-exploration trade-off. The best balance between exploitation and exploration is achieved through proper selection of the parameters α and β. If $\alpha = 0$, no pheromone information is used, i.e. previous search experience is neglected. If $\beta = 0$, the attractiveness (or visibility) of moves is neglected.

At the end of every iteration, ants that have found good solutions are made to mark their path by depositing pheromones on the edges chosen by them. The optimization algorithm may be designed such that either the iteration best ant or global best ant or both of them deposit pheromone. After all ants have completed their solutions, pheromone evaporation on all edges triggered. It helps in avoiding rapid convergence of the algorithm toward a sub-optimal region. The pheromone content of path (i,j) at a time instance t+1 is given by (2):

$$\tau_{ij}(new)=(1-\rho)\tau_{ij}(t)+\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t)+\Delta\tau_{ij}^{g}(t) \quad (2)$$

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{F^{k}} & \text{if the } k\text{-}th \text{ ant traverse arc } (i,j) \text{ in } T_k \\ 0 & otherwise\end{cases} \quad (3)$$

where $\rho\epsilon(0,1]$ is the evaporation rate, m is the number of ants, $\Delta\tau_{ij}^{k}(t)$ and $\Delta\tau_{ij}^{g}(t)$ are respectively, the amount of pheromone laid on edge (i,j) by the k-th ant, and the amount of pheromone to be deposited by the global best ant g up to the time instance t, over the edge (i,j). Q is a constant, and F^{k} is the cost value of the solution found by k-th ant in its current tour T_k (in this paper classification error rate is considered as cost function). The less its classification error rate, the stronger the intensity of pheromone it is allowed to deposit in each iteration.

The proposed algorithm uses the max-min ant system, i.e., only the global best ant is allowed to update the pheromone trails, and the value of pheromone on each road is confined to $[\tau_{min},\tau_{max}]$. Therefore, Eq. (2) is modified as follows:

$$\tau_{ij}(new)=\left[(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}^{g}(t)\right]_{\tau\min}^{\tau\max} \quad (4)$$

The iterative process continues till the stopping criterion is reached. The stopping criterion may be either a number of iterations or a solution of desired quality [36].

2.2. **Binary ant colony optimization**

Touring ant colony optimization (TACO), was initially designed by [37], for handling continuous variables (which are decoded as binary strings) in the optimization problems, and was later used for different problems such as digital filter design [38, 39]. In this algorithm, each solution is represented with a string of binary bits. Artificial ants search for the value of each bit in the string. In other words, they try to decide whether the value of a bit is 0 or 1. At the decision stage for the value of a bit, ants only use the pheromone information. After all ants in the colony have produced their solutions and the pheromone amount belonging to each solution has been calculated, the pheromone of sub-paths (edges) between the bits are updated. This is carried out by evaporating the previous pheromone amounts and depositing the new pheromone amounts on the paths. The concept of TACO algorithm is shown in Fig. 1.

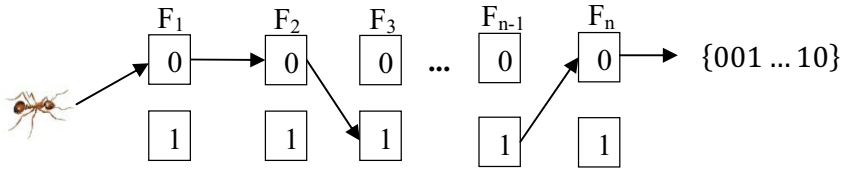

Fig. 1. The concept of touring ant colony optimization algorithm

Using TACO for feature selection problem was first introduced by [34]. They proposed three modifications of TACO called elitism TACO (ETACO), rank-based TACO (RTACO) and binary ACO (BACO), where the latter led to the best results. In this approach, only the global best ant is admissible to deposit pheromone and max-min ant system strategy is utilized for pheromone updating, i.e. the pheromone trail is limited to the interval $[\tau_{min}, \tau_{max}]$, where τ_{min} and τ_{max} are arbitrary positive real number satisfied $\tau_{min} < \tau_{max}$. Later, [40] employed this type of ACO for feature selection called ACOFS, using F-score criterion as the heuristic value (visibility of edges) but with a different pheromone updating strategy. Here, nodes represent features, where each node contains two sub-nodes; 0 and 1. First, all ants are at the beginning of the bit string and then pass through these sub-nodes. The ant's tour ends when it passes the last feature. If an ant chooses sub-node 1 (or 0) of the *i*-th feature, it means this feature is selected (or deselected) by that ant. Assume that the probability of being preferred of the sub-path between 0 and 1 (0→1) at a stage is calculated. Then the following equation is used:

$$P_{01}(t) = \frac{\tau_{01}}{\tau_{01} + \tau_{00}} \qquad (5)$$

where P_{01} is the probability associated with the sub-path (0→1), and τ_{00} and τ_{01} are the artificial pheromones of the sub-paths (0→0 and 0→1).

Although BACO is able to find near-optimal solution in a short time, it suffers a big problem, which is the limited view of ants toward features. At any time each ant can only observe its next feature and is not able to see other features. In the proposed algorithm, we try to solve this problem by combining conventional ACO and BACO.

3. Proposed feature selection algorithm

This section describes the proposed approach. It is a combination of ACO and BACO algorithms to remove their limitations. In other words, there is no need to predefine the number of features to be selected (limitation of ACO) and this task is assigned to the algorithm to select feature subsets with arbitrary numbers. Beside, nodes are fully connected and ants are able to observe all features simultaneously (contrary to BACO algorithm) and they can decide to select a feature or not (similar to BACO).

Like ACO-based FS approach, the problem is defined as a fully connected graph where nodes represent features, with the edges between them denoting the choice of the next features. Similar to BACO algorithm, there are two sub-nodes assigned to each feature in the graph, one for selecting and the other for deselecting the corresponding feature. Fig. 2 illustrates this structure. Based on the pheromone values ants decide their next edge. In each iteration, all ants should visit all features, but can decide whether to select a feature or not. If an ant chooses sub-node 1 (or 0) of feature F_i, it means the feature is selected (or deselected) by that ant. Note that, ants are only allowed to select one of the sub-nodes of each feature, sub-node 1 or 0. For the next step, this ant can see all the unvisited features. Again, there are two roads ending to each feature. Based on the pheromone values and heuristic information of these edges, the ant chooses its next road and the process continues until the ant visits all features. A similar process is repeated for other ants. At the end of each iteration, each ant has a solution path in the form of a binary vector with the same length as the number of the features, where 1 means selecting and 0 means deselecting the corresponding feature. For updating the pheromone of the roads, only the best ant, i.e. the ant with the smallest classification error of the classifier, is allowed to deposit pheromone on the edges it has traversed. This process continues for all iterations and at last, the best feature subset with the least classification error of the classifier is suggested as the best result.

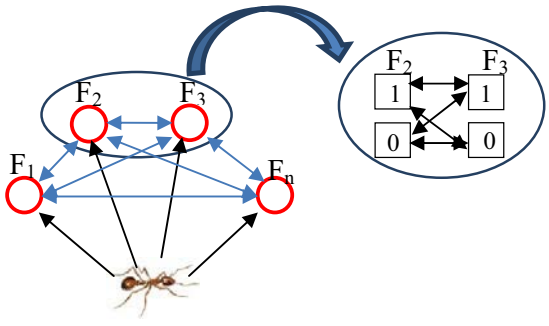

Fig. 2. ABACO algorithm representation.

With this idea, the FS problem can be modeled as TSP, where ants start their tours from a random node and end it when all nodes are visited. The main point of the proposed algorithm is that, by considering two sub-nodes with the values of 0 and 1 for each node, there is no need to predetermine the number of the features to be selected. Beside, as each node contains two sub-nodes, there are four edges between each two nodes instead of one edge. These edges are illustrated in Fig. 3 for features *i* and *j*. Expressions above each edge show the pheromone intensity and heuristic information of that edge. For example, $\tau_{i1,j0}$ and $\eta_{i1,j0}$ are respectively, the pheromone value and heuristic information of the edge which connects sub-node 1 of feature *i* to sub-node 0 of feature *j*. In the following, the methods are described.

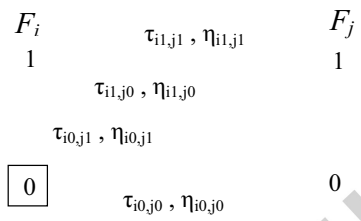

Fig.3. Edges between two nodes (features) in ABACO algorithm.

The search for the optimal feature subset is the goal of the ants traverse through the graph. Suppose an ant is currently at node $F_{i,x}$ ($i=1,2,\ldots,n$ and $x=0,1$) and has to choose one path connecting $F_{j,y}$ ($j \in addmissible\,nodes$ and $y=0,1$) to pass through. A probabilistic function of transition, denoting the probability of an ant at node $F_{i,x}$ to choose the path to reach $F_{j,y}$, is designed by combining the heuristic desirability (visibility of edge (i,j)) and pheromone density of the edge. The probability of the ant k at sub-node $F_{i,x}$ to choose the edge (i,j) at time t is

$$P_{ix,jy}^{k}(t)=\begin{cases}\dfrac{\tau_{ix,jy}^{\alpha}\eta_{ix,jy}^{\beta}}{\sum_{l}\tau_{ix,l0}^{\alpha}\eta_{ix,l0}^{\beta}+\sum_{l}\tau_{ix,l1}^{\alpha}\eta_{ix,l1}^{\beta}} & \text{if } j,l\in admissible\,nodes\\ 0 & otherwise\end{cases} \tag{6}$$

Here, $\tau_{ix,jy}$ is the pheromone on edge (ix,jy) between sub-nodes $F_{i,x}$ and $F_{j,y}$ at time t, which reflects the potential tend for ants to follow this edge. $\eta_{ix,jy}$ is the heuristic information reflecting the desirability of choosing edge (ix,jy). α and β are two parameters that determine the relative importance of the pheromone value and the heuristic information.

3.1. **Heuristic information measurement**

Adding heuristic desirability/visibility to the edges, enhances the ability of exploiting the search space and makes it easier to find the optimum solution. A heuristic value, η, for each feature generally represents the

attractiveness of the features, and can be any subset evaluation function like an entropy based measure or rough set dependency measure [41]. In the proposed algorithm, we tried four methods to determine the heuristic value including F-score and three different approaches based on correlation.

3.1.1. Method 1

The first three methods are based on correlation. Correlation is one of the most common and useful statistics that describes the degree of relationship between two variables. A number of criteria have been proposed in statistics to estimate correlation. In this work, ABACO uses the best known *Pearson product-moment correlation coefficient* to measure correlation between different features of a given training set. The correlation coefficient r_{ij} between two features *i* and *j* is:

$$r_{ij} = \frac{\sum_{h}(x_i - \overline{x_i})(x_j - \overline{x_j})}{\sqrt{\sum_{h}(x_i - \overline{x_i})^2}\sqrt{\sum_{h}(x_j - \overline{x_j})^2}} \qquad (7)$$

where x_i and x_j are the value of features *i* and *j*, respectively. The variables $\overline{x_i}$ and $\overline{x_j}$ represent the mean values of x_i and x_j, averaged over *h* samples. If the features *i* and *j* are completely correlated, i.e., exact linear dependency exist, then r_{ij} would be 1 or -1. If *i* and *j* are completely uncorrelated then r_{ij} would be 0 [7].

Method 1 employs the idea of min redundancy, which tries to select the most distinct features. For convenience, the correlation between features *i* and *j* is assumed to be high. In this case, the two features are highly similar. Therefore, to describe the whole set, one of these features is enough. Hence, if one of them is selected, the probability to select/deselect the other feature can be described as *1- rij/ rij*, that has a low/high value. Again, if the first feature is not selected, presence of the other feature is not necessary too, because they are similar. So, the probability to select/deselect the other feature can be described as*1- rij /rij*, that has a low/high value. Eq.8 illustrates the above statements.

$$\begin{aligned} \eta_{i0,j0} &= |r_{ij}| \\ \eta_{i0,j1} &= 1 - |r_{ij}| \\ \eta_{i1,j0} &= |r_{ij}| \\ \eta_{i1,j1} &= 1 - |r_{ij}| \end{aligned} \qquad (8)$$

3.1.2. Method 2

In this method, the idea of max-relevance and min-redundancy is used. Max-relevance is one of the most popular approaches to realize max-dependency in feature selection, i.e. selecting the features with the highest relevance to the target class *c*. Relevance is usually characterized in terms of correlation or mutual information [42]. As Eq. (9) shows, geometric mean of the two criteria (max-relevance and min-redundancy) is considered to calculate the heuristic information of edges. Here, $cls\text{-}cor_j$, is the correlation between feature *j* and class labels over all samples. The more this value is close to 1 for a feature, the higher this feature is correlated to the class labels and thus the feature is more important.

$$\begin{aligned}
\eta_{i0,j0} &= \sqrt{(|r_{ij}|)(1-|cls_cor_j|)} \\
\eta_{i0,j1} &= \sqrt{(1-|r_{ij}|)(|cls_cor_j|)} \\
\eta_{i1,j0} &= \sqrt{(|r_{ij}|)(1-|cls_cor_j|)} \\
\eta_{i1,j1} &= \sqrt{(1-|r_{ij}|)(|cls_cor_j|)}
\end{aligned} \tag{9}$$

3.1.3. Method 3

The idea behind this method is to use feature-feature correlation, if feature *i* is selected, and class-feature correlation is used in the case of deselecting feature *i*.

$$\begin{aligned}
\eta_{i0,j0} &= 1-|cls_cor_j| \\
\eta_{i0,j1} &= |cls_cor_j| \\
\eta_{i1,j0} &= |r_{ij}| \\
\eta_{i1,j1} &= 1-|r_{ij}|
\end{aligned} \tag{10}$$

3.1.4. Method 4

This method is based on F-score which is a measurement to evaluate the discrimination ability of feature *i*. Eq. (11) defines the F-score of the *i*-th feature. The numerator specifies the discrimination among the categories of the target variable, and the denominator indicates the discrimination within each category. A larger F-score implies to a greater likelihood that this feature is discriminative [43].

$$F\_score_i = \frac{\sum_{k=1}^{c}(\overline{x_i^k}-\overline{x_i})^2}{\sum_{k=1}^{c}\left[\frac{1}{N_i^k-1}\sum_{j=1}^{N_i^k}(x_{ij}^k-\overline{x_i^k})^2\right]} (i=1,2,...,n) \tag{11}$$

where c is the number of classes and n is the number of features; N_i^k is the number of samples of the feature i in class k, ($k = 1,2,...,c$, $i = 1,2,...,n$), x_{ij}^k is the j-th training sample for the feature i in class k, (j = 1,2,…, N_i^k), $\overline{x_i}$ is the mean value of feature i of all classes and $\overline{x_i^k}$ is the mean of feature i of the samples in class k. Heuristic information of edges using this criterion is as follows:

$$\begin{aligned}
&\eta_{i0,j0} = (\xi/n)\sum_{k=1}^{n} F_score_k \\
&\eta_{i0,j1} = F_score_j \\
&\eta_{i1,j0} = (\xi/n)\sum_{k=1}^{n} F_score_k \\
&\eta_{i1,j1} = F_score_j
\end{aligned} \qquad (12)$$

here n is the total number of features and $\xi \in (0,1)$ is a constant. This method is similar to the heuristic information used in ACOFS algorithm [40]. Hereafter, ABACO with heuristic desirability is called $ABACO_H$.

3.2. **Pheromone updating**

After all ants have completed their solutions, pheromone evaporation on all edges is triggered. As mentioned earlier, in this algorithm only the global best ant is allowed to participate in pheromone updating of edges according to Eq. 4.

3.3. **Proposed algorithm**

The steps of ABACO can be described by the flowchart shown in Fig. 4, which are described in more details as follows.

Step 1: Initialize the parameters of ACO, including the number of ants m, the maximum number of iteration $Imax$, the tunable parameters α, β and ρ, the initial pheromone level τ_0, the pheromone trail interval $[\tau_{min}, \tau_{max}]$ and the heuristic information, η of all edges using one of the methods described in the sub-section 3-1.

Step 2: Construct candidate solutions from a randomly selected node and select one edge from the rest *2(n-1)* roads ending to *(n-1)* features, follow the proposed probabilistic transition rule, Eq. 6. Choosing sub-node 1 of a feature means selecting, and sub-node 0 means deselecting that feature. Stay in this step while all features (nodes) have been visited by each ant.

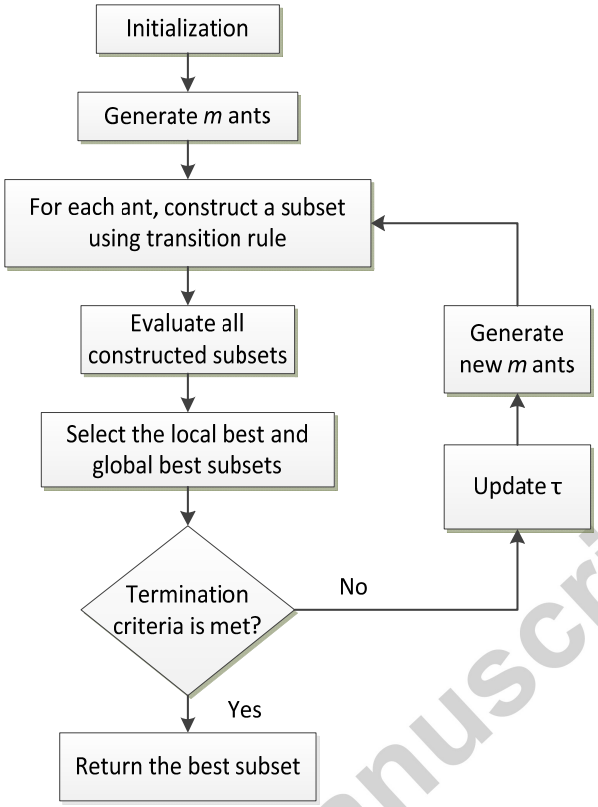

Fig. 4. The structure of the proposed ACO-based FS approach ($ABACO_H$).

Step 3: Evaluate the candidate feature subsets using the trained classifier by testing the classification accuracy on the training set.

Step 4: After evaporation of the laid pheromone, find the ant with the best feature subset. It is the subset with the best classification accuracy. Only permit this ant to deposit pheromone according to the scheme explained in the previous section. Keep the pheromone values of edges in the interval [τ_{min} , τ_{max}]. If the maximum number of iteration, *Imax* is not reached go back to step 2; otherwise, go to the next step.

Step 5: search for the global best subset which produces the highest classification accuracy among all local best solutions.

4. Experimental studies

A series of experiments are conducted to show the effectiveness of the proposed feature selection algorithm. All experiments were performed on a laptop with 2.40 GHz CPU and 4 Gb of RAM using Matlab. For

experimental studies, we have considered twelve datasets from the UCI (University of California, Irvine) machine learning repository [45], including Abalon, Glass, Iris, Letter, Shuttle, Spambase, Tae, Vehicle, Waveform, Wine, Wisconsin and Yeast. These datasets have been the subject of many studies in machine learning, covering examples of small, medium and high -dimensional datasets [16, 44]. The characteristics of these datasets, summarized in Table 1, show a considerable diversity in the number of features, classes, and samples. To validate the results obtained by the proposed algorithm, it is compared with binary particle swarm optimization (BPSO) and CatfishBPSO [17], binary genetic algorithm (BGA), improved binary gravitational search (IBGSA) [21], ant colony optimization with/without heuristic information (ACO_H/ACO), binary ant colony optimization (BACO) [34], a modified binary ACO based feature selection algorithm presented in [40], which is denoted as ACOFS and ABACO [35], and the results obtained are reported.

Table 1

Characteristics of different benchmark data sets.

Database name	Number of classes	Number of features	Number of samples
Abalone	11	8	3842
Glass	6	9	214
Iris	3	4	150
Letter	26	16	20000
Shuttle	7	9	58000
Spambase	2	57	4601
Tae	3	5	151
Vehicle	4	18	846
Waveform	3	21	5000
Wine	3	13	178
Wisconsin	2	9	683
Yeast	9	8	1484

4.1. **Evaluation functions**

The features selected by the proposed algorithms are evaluated with the well-known metrics precision, recall, accuracy and feature-reduction. Precision is defined as the ratio of correctly assigned category C samples to the total number of samples classified as category C as in Eq. 13. Recall is the ratio of correctly assigned category C samples to the total number of samples actually in category C as in Eq. 14 [36]. Let TP_i, FP_i, TN_i, and FN_i indicate a number of samples as follows:

TP_i – the number of test samples correctly classified under ith category (C_i)

FP_i – the number of test samples incorrectly classified under C_i

TN_i – the number of test samples correctly classified under other categories

FN_i – the number of test samples incorrectly classified under other categories

$$precision_i = \frac{TP_i}{TP_i + FP_i} \tag{13}$$

$$recall_i = \frac{TP_i}{TP_i + FN_i} \tag{14}$$

In this paper, classification accuracy (CA) is used to define the quality function of a solution, which is the percentage of samples correctly classified and evaluated as in (15).

$$Accuracy = \frac{number \text{ of samples correctly classified}}{\text{total number of samples taken for experimentation}} \tag{15}$$

Another parameter which is used for comparison is the average feature reduction F_r, to investigate the rate of feature reduction:

$$F_r = \frac{n - p}{n} \tag{16}$$

where n is the total number of features and p is the number of selected features by the FS algorithm. F_r is the average feature reduction. The more it is close to 1, the more features are reduced, and the classifier complexity is less. The following section describes the implementation results.

4.2. **Parameter setting**

Population size for all algorithms and the maximum iterations are set to 50. During each experimentation, 60% of samples were chosen randomly for training. Remaining 40% of samples were used for testing. Results are averaged over 20 independent runs in each data set and by every algorithm. The selected features of each method are classified using k-Nearest Neighbor *(k-NN, k=1)* and fitness function is defined as the classification accuracy.

For ACO-based algorithms $ABACO_H$, ABACO, BACO, ACO_H and ACO, the evaporation coefficient, ρ is 0.049, the minimum and the maximum pheromone intensity of each edge is set to 0.1 and 6, respectively. Also the initial pheromone intensity (τ_0) of each edge is set to 0.1. Since α and β in ACO are two parameters that determine the relative importance of the pheromone and the heuristic information, we first fix the value of α as 1 and set the value of β in the range of [0.2,1]. The best parameters that are obtained are as follows: α =1, β =0.5. Parameters of ACOFS are set according to [40] (α =1, β =0.5, $\rho = 0.049$, $\tau_0 = 1$).

For GA-based FS method, we choose one point cross-over with probability of 0.9. The mutation probability is set to 0.01. Finally parameters of BPSO and CatfishBPSO are set according to [17]: $w = 1$ and $c_1 = c_2 = 2$. In IBGSA, the parameters, as reported in [21], are set to $k_1 = 1$ and $k_2 = 500$, and Hamming distance is used for distance calculation.

4.3. **Expeimental results and analysis**

To specify the best method for the heuristic information of edges, the average classification accuracies of the proposed algorithm over 20 independent runs on the tested datasets are given in Table 2. The numbers in the last row of the table, show the average classification accuracy over the datasets. According to this table, method 4

could achieve the best result among other proposed methods. Therefore, this method is chosen for the heuristic information of paths.

Table 2

Classification accuracy of each method on the tested data sets. The results are averaged over 20 independent runs. The number below each column of the table, shows the average classification accuracy over the data sets.

Dataset	Method 1	Method 2	Method 3	Method 4
	Mean(±Std)	**Mean(**±Std)	**Mean(**±Std**)**	**Mean(**±Std**)**
Abalone	0.242(±0.0066)	0.244(±0.0069)	0.243(±0.0045)	0.244(±0.0044)
Glass	0.742(±0.0425)	0.749(±0.0345)	0.736(±0.0365)	0.758(±0.0410)
Iris	0.968(±0.0157)	0.971(±0.0186)	0.978(±0.0165)	0.976(±0.0148)
Letter	0.855(±0.0093)	0.862(±0.0096)	0.858(±0.0113)	0.861(±0.0129)
Shuttle	0.998(±0.0006)	0.998(±0.0008)	0.998(±0.0007)	0.998(±0.0008)
Spambase	0.918(±0.0032)	0.922(±0.0032)	0.921(±0.0007)	0.923(±0.0047)
Tae	0.567(±0.0303)	0.553(±0.0298)	0.557(±0.0463)	0.583(±0.0506)
Vehicle	0.745(±0.0151)	0.75(±0.0166)	0.75(±0.0151)	0.753(±0.0126)
Waveform	0.792(±0.0062)	0.79(±0.0047)	0.793(±0.0042)	0.798(±0.0056)
Wine	0.974(±0.0138)	0.964(±0.0097)	0.962(±0.0120)	0.969(±0.0144)
Wisconsin	0.975(±0.0061)	0.975(±0.0065)	0.976(±0.0073)	0.977(±0.0065)
Yeast	0.518(±0.0104)	0.52(±0.0142)	0.51(±0.0124)	0.524(±0.0152)
Average	0.774	0.775	0.773	**0.780**

To show the utility of the proposed algorithm, we compare the algorithm with IBGSA [21], CatfishBPSO [17], BGA, BPSO, ACO and two new ACO-based FS algorithms [34, 40], which are reported to be very strong algorithms in FS. Table 3 shows the mean of classification accuracy (CA) results of every algorithm for each dataset. Comparison of the average precision and recall and the amount of F_r of the competing algorithms on the datasets are displayed in Table 4. We can conclude from these tables that the proposed ABACO algorithm can obtain, in some cases, better classification accuracy using a smaller feature set, compared to other algorithms.

In Table 5, another comparison has been made according to the sum of the ranks available in Table 3 for each algorithm. The lower sum of ranks for an algorithm shows the better average results in total cases against the others. Although this quantity is of lower accuracy degree for reporting results in some cases, it is common in Nonparametric Statistics. As can be seen in this Table, $ABACO_H$, gets the first rank. Both ABACO and ACOFS achieve the second rank. IBGSA and CatfishBPSO ranked third and fourth and BACO, ACO_H, GA, BPSO and ACO get the fifth to ninth rankings, respectively.

In practical engineering issues, solving the problem consumes the main time, and the time spent by the metaheuristic algorithm operators is negligible. As a wrapper method is employed in this special problem, classifier takes a lot of time. Table 6 shows the average time of algorithms over each dataset. The last row of the table, shows the average latency of each method over the datasets. According to this table, the delay of ABACO is not much different than other algorithms. The results obtained by $ABACO_H$ confirm that the proposed algorithm can be thought as a worthwhile method for feature selection.

Table 3

Classification accuracy for the tested datasets. The average of results over 20 independent runs is reported. The number in brackets in each table slot shows the ranking of each algorithm.

Dataset	$ABACO_H$	ABACO	ACOFS	BACO	ACO_H	ACO	BGA	BPSO	IBGSA	CatfishBPSO
Abalone	**0.244** (1)	0.241 (3)	0.243 (2)	0.243 (2)	0.241 (3)	0.238 (5)	0.241 (3)	0.241 (3)	0.241 (3)	0.240 (4)
Glass	**0.758** (1)	0.747 (3)	0.750 (2)	0.743 (4)	0.728 (8)	0.713 (9)	0.733 (7)	0.734 (6)	0.738 (5)	0.733 (7)
Iris	0.976 (2)	0.974 (3)	**0.977** (1)	0.967 (6)	0.971 (4)	0.963 (9)	0.967 (7)	0.965 (8)	0.969 (5)	0.968 (6)
Letter	**0.861** (1)	0.856 (4)	0.859 (3)	0.859 (3)	0.856 (4)	0.806 (7)	0.837 (5)	0.825 (6)	0.86 (2)	0.856 (4)
Shuttle	**0.998** (1)	**0.998** (1)	**0.998** (1)	**0.998** (1)	**0.998** (1)	**0.998** (1)	0.997 (2)	**0.998** (1)	**0.998** (1)	**0.998** (1)
Spambase	0.923 (2)	0.921 (4)	0.922 (3)	0.919 (5)	0.913 (6)	0.901 (8)	0.906 (7)	0.9 (9)	0.922 (3)	**0.924** (1)
Tae	**0.583** (1)	0.573 (3)	0.569 (4)	0.558 (8)	0.559 (7)	0.578 (2)	0.556 (9)	0.567 (5)	0.556 (9)	0.565 (6)
Vehicle	**0.753** (1)	**0.753** (1)	0.749 (2)	0.749 (2)	0.739 (5)	0.718 (8)	0.737 (6)	0.722 (7)	0.745 (4)	0.746 (3)
Waveform	**0.798** (1)	0.795 (4)	0.797 (2)	0.793 (6)	0.796 (3)	0.768 (10)	0.776 (9)	0.792 (7)	0.794 (5)	0.79 (8)
Wine	0.969 (2)	0.969 (2)	0.964 (3)	0.964 (3)	0.958 (4)	0.938 (7)	0.957 (5)	0.941 (6)	**0.978** (1)	0.969 (2)
Wisconsin	0.976 (2)	0.976 (2)	0.974 (4)	0.975 (3)	0.974 (4)	0.968 (5)	0.975 (3)	0.968 (5)	**0.977** (1)	0.976 (2)
Yeast	0.524 (2)	**0.529** (1)	0.516 (4)	0.515 (5)	0.512 (7)	0.509 (8)	0.513 (6)	0.507 (9)	0.515 (5)	0.522 (3)

At this point, it should be mentioned that although there is no universal metaheuristic algorithm that can get the best results on the entire available benchmarks, the results obtained by $ABACO_H$ confirm that the proposed algorithm also can be thought as a worthwhile method for feature selection.

4.4. **Discussion**

This section briefly explains the reason that the performance of ABACO was better than other algorithms. Here, are some salient characteristics of ABACO.

The first one is that ABACO permits ants to explore all features, but in most of ACO-based FS algorithms, searching among features continues until the stopping criterion is met. Therefore, ants do not have the opportunity to observe all features.

Table 4

Comparison of performance (precision, recall and Fr) of the algorithms on 12 data sets.

	Metrics	Abalone	Glass	Iris	Letter	Shuttle	Spambase	Tae	Vehicle	Waveform	Wine	Wisconsin	Yeast	Sum
$ABACO_H$	Precision	0.222	0.743	0.978	0.867	0.912	0.92	0.589	0.755	0.799	0.972	0.973	0.529	**9.259**
	Recall	0.223	0.736	0.975	0.862	0.927	0.92	0.585	0.757	0.799	0.973	0.976	0.521	**9.254**
	Fr	0.337	0.299	0.437	0.343	0.541	0.431	0.357	0.436	0.585	0.484	0.384	0.063	4.697
ABACO	Precision	0.221	0.693	0.974	0.863	0.912	0.919	0.585	0.755	0.797	0.97	0.973	0.504	9.166
	Recall	0.219	0.676	0.974	0.861	0.907	0.916	0.576	0.762	0.797	0.973	0.976	0.51	9.147
	Fr	0.386	0.361	0.375	0.335	0.453	0.439	0.389	0.428	0.272	0.549	0.348	0.076	4.411

ACOFS	Precision	0.225	0.723	0.978	0.864	0.958	0.921	0.574	0.749	0.798	0.965	0.969	0.509	9.233
	Recall	0.223	0.71	0.976	0.861	0.934	0.918	0.569	0.752	0.798	0.97	0.974	0.496	9.181
	Fr	0.331	0.365	0.319	0.361	0.522	0.447	0.356	0.426	0.272	0.483	0.367	0.102	4.351
BACO	Precision	0.223	0.73	0.967	0.865	0.944	0.917	0.568	0.749	0.794	0.965	0.973	0.509	9.204
	Recall	0.222	0.695	0.967	0.863	0.93	0.917	0.562	0.754	0.794	0.972	0.974	0.507	9.157
	Fr	0.387	0.347	0.459	0.332	0.565	0.465	0.363	0.439	0.261	0.494	0.387	0.089	4.588
ACO_H	Precision	0.221	0.677	0.971	0.861	0.933	0.91	0.568	0.739	0.796	0.959	0.971	0.476	9.082
	Recall	0.219	0.682	0.971	0.857	0.921	0.91	0.573	0.745	0.796	0.964	0.973	0.485	9.096
	Fr	0.344	0.24	0.313	0.313	0.4	0.461	0.41	0.397	0.238	0.478	0.328	0.113	4.035
ACO	Precision	0.222	0.648	0.963	0.812	0.931	0.898	0.589	0.716	0.768	0.941	0.964	0.464	8.916
	Recall	0.219	0.654	0.962	0.808	0.914	0.896	0.58	0.721	0.768	0.943	0.966	0.476	8.907
	Fr	0.388	0.372	0.313	0.313	0.472	0.461	0.41	0.431	0.371	0.504	0.389	0.156	4.58
BGA	Precision	0.225	0.719	0.969	0.843	0.925	0.903	0.567	0.734	0.777	0.958	0.971	0.48	9.071
	Recall	0.222	0.682	0.968	0.839	0.894	0.901	0.556	0.74	0.777	0.964	0.975	0.476	8.994
	Fr	0.373	0.339	0.388	0.341	0.561	0.454	0.441	0.454	0.324	0.49	0.368	0.141	4.674
BPSO	Precision	0.221	0.708	0.965	0.863	0.97	0.896	0.578	0.718	0.795	0.944	0.963	0.466	9.087
	Recall	0.217	0.673	0.965	0.858	0.91	0.895	0.571	0.725	0.795	0.946	0.966	0.464	8.985
	Fr	0.438	0.367	0.45	0.348	0.531	0.477	0.38	0.478	0.283	0.531	0.406	0.206	**4.895**
NBGSA	Precision	0.212	0.672	0.961	0.84	0.911	0.906	0.548	0.728	0.773	0.93	0.966	0.472	8.919
	Recall	0.212	0.668	0.962	0.835	0.928	0.905	0.546	0.732	0.773	0.935	0.964	0.474	8.934
	Fr	0.542	0.315	0.372	0.42	0.381	0.465	0.283	0.413	0.51	0.341	0.386	0.129	4.557
catfish BPSO	Precision	0.22	0.706	0.97	0.865	0.867	0.928	0.579	0.749	0.792	0.972	0.976	0.496	9.12
	Recall	0.221	0.699	0.969	0.862	0.935	0.926	0.568	0.754	0.793	0.974	0.977	0.491	9.169
	Fr	0.369	0.359	0.3	0.34	0.527	0.443	0.44	0.451	0.305	0.513	0.392	0.134	4.573

Table 5

The sum of the relative obtained ranks on the 12 number of data sets for each of the algorithms.

$ABACO_H$	ABACO	ACOFS	BACO	ACO_H	ACO	GA	BPSO	IBGSA	catfish BPSO
17 (1)	31 (2)	31 (2)	48 (5)	56 (6)	81 (9)	69 (7)	70 (8)	44 (3)	47 (4)

The second characteristic is the new search technique used in ABACO. Unlike other ACO-based FS algorithms that ants should select every feature they visit, in ABACO algorithm ants have the authority to select or deselect visiting features. This search technique is common between BACO, ACOFS and ABACO.

Third, the advantage of ABACO to BACO and ACOFS arises from the comprehensive view of ants to features in ABACO compared to the limited view in BACO and ACOFS. Although in BACO and ACOFS, every ant could visit all features, and had the authority to select or deselect the next feature, but at any time each ant could only observe its next feature and could not see other features. Since in ABACO, ants can see all of the unvisited features simultaneously, they can decide better about the next feature, and are not forced to select or deselect a predefined feature.

Finally, compared to the initial version of ABACO introduced in [35], adding heuristic desirability increases the exploration of search and guide ants to more salient features.

Table 6

The average time of algorithms on each dataset. The last row of the table, shows the average latency of each method over the datasets.

Dataset	$ABACO_H$	ABACO	ACOFS	BACO	ACO_H	ACO	BGA	BPSO	IBGSA	CatfishBPSO
Iris	1.13	1.12	1.11	0.87	1.14	1.15	0.85	0.87	0.85	0.88
Glass	1.60	1.57	1.65	1.10	1.41	1.44	1.06	1.15	1.05	1.25

Vehicle	8.29	7.10	6.51	4.50	5.20	7.06	4.56	5.22	5.11	6.58
Wine	2.08	1.83	1.56	1.02	1.94	1.74	1.31	1.11	1.53	1.17
Abalone	8.07	7.69	7.94	5.22	7.57	6.56	6.27	6.08	6.94	6.58
Letter	16.40	15.77	10.14	10.70	9.99	13.04	11.99	8.08	8.41	11.29
Shuttle	20.93	22.84	19.20	15.20	23.94	28.74	19.95	16.99	17.64	25.94
Spambase	169.89	161.36	128.62	109.64	112.72	106.77	75.26	60.08	62.20	80.37
Tae	1.13	0.83	1.10	0.76	1.10	0.79	0.85	0.85	0.82	1.28
Waveform	91.84	67.06	88.52	75.94	73.78	66.67	49.37	50.45	56.14	81.41
Wisconsin	2.97	2.10	3.73	2.08	2.98	2.33	2.77	2.36	2.25	2.91
Yeast	4.03	2.97	3.42	3.35	4.81	3.38	3.44	3.33	2.61	4.82
mean	27.36	24.35	22.79	19.19	20.54	19.97	14.80	13.04	13.79	18.70

5. Conclusion

Feature selection is an important task which can significantly affect the performance of classification and recognition. In this paper, we present a new feature selection technique based on Ant Colony Optimization (ACO) by combining two models of ACO. The proposed algorithm has a strong search capability in the problem space and can effectively find the minimal feature subset. This algorithm is compared with some powerful algorithms, including IBGSA, CatfishBPSO, ACOFS, BACO, ACO with and without heuristic desirability, BGA and BPSO.

In order to evaluate the performance of these approaches, experiments were performed using twelve datasets from the UCI machine learning repository. The experimental results confirm our algorithm and provide obvious evidences, allowing us to conclude that our method achieves a better feature set in terms of classification accuracy and number of selected features. Further investigation on the parameters values and testing the ABACO model with other heuristic functions are an area of future research.

References

[1] S.M. Vieira, J.M.C. Sousa, T.A. Runkler, Two cooperative ant colonies for feature selection using fuzzy models, Expert Systems with Applications 37 (2010) 2714–2723.

[2] M. Pal and G.M. Foody, Feature Selection for Classification of Hyperspectral Data by SVM, IEEE Transactions on Geoscience and Remote Sensing, 48 (5) (2010).

[3] H. Liu and L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transaction on Knowledge and Data Engineering, 17 (4) (2005) 491-502.

[4] S. Ding, Feature selection based F-score and ACO algorithm in support vector machine, 2nd International Symposium on Knowledge Acquisition and Modeling, 2009.

[5] L.T. Vinh, S. Lee, Y.-T. Park and B.J. d'Auriol, A novel feature selection method based on normalized mutual information, Appl Intell 37 (2010) 100-120.

[6] M.H. Aghdam, N. Ghasem-Aghaee and M.E. Basiri, Text feature selection using ant colony optimization, Expert Systems with Applications, 36 (2009) 6843-6853.

[7] M. Kabir, Shahjahan, K. Murase, A new local search based hybrid genetic algorithm for feature selection, Neurocomputing, 74 (2011) 2914–2928.

[8] D. Hu, P.Ronhvde, Z. Nussinov, Replica inference approach to unsupervised multiscale image segmentation, Physical Review E 85, 016101 (2012).

[9] R. Kohavi, G. John, Wrappers for feature selection, Artificial Intelligence, 97 (1-2) (1997) 273-324.

[10] M.E. Basiri, N. Ghasem-Aghaee, M.H. Aghdam, Using ant colony optimization-based selected features for predicting post-synaptic activity in proteins, EvoBIO in: Lecture Notes in Computer Science, 4973 (2008) 12-23, Italy.

[11] P. Bermejo, J.A. Gámez and J.M. Puerta, A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets, Pattern Recognition Letters, 32 (2001) 701–711.

[12] J. Huang, Y. Cai and X. Xu, "A hybrid genetic algorithm for feature selection wrapper based on mutual information," Pattern Recognition Letters Vol. 28, pp. 1825-1844, 2007.

[13] R.K. Sivagaminathan, S. Ramakrishnan, A hybrid approach for feature subset selection using neural networks and ant colony optimization, Expert Systems with Applications, 33 (2007) 49-60.

[14] I. Guyon and A. Elisseeff, An introduction to variable and feature selection, Journal of Machine Learning Research, 3 (2003) 1157-1182.

[15] M. Dash and H. Liu, Feature selection for classification, Intelligent Data Analysis, 1 (1997) 131-156.

[16] L.Y. Chuang, C.H. Yang and J.C. Li, Chaotic maps based on binary particle swarm optimization for feature selection, Applied Soft Computing, 11 (2011) 239-248.

[17] L.Y. Chuang, S.W. Tsai, C.H. Yang, Improved binary particle swarm optimization using catfish effect for feature selection, Expert Systems with Applications, 38 (2011) 12699–12707.

[18] X.Wang, J. Yang, X.Teng, W.Xia, R. Jensen "Feature selection based on rough sets and particle swarm optimization." Pattern Recognition Letters, 28 (4) (2007)459-471.

[19] I. oh, J.S.Lee and B.R. Moon, Hybrid genetic algorithm for feature selection, IEEE Transaction on Pattern Analysis and Machine Intelligence, 26 (11) (2004) 1424-1437.

[20] S. Sarafrazi, and H. Nezamabadi-pour, Facing the classification of binary problems with a GSA-SVM hybrid system, Mathematical and Computer Modelling, 57 (1-2) (2013) 270-278.

[21] E. Rashedi, and H. Nezamabadi-pour, Feature subset selection using improved binary gravitational search algorithm, Journal of Intelligent and Fuzzy Systems, 2013. DOI: 10.3233/IFS-130807.

[22] E. Rashedi, H.Nezamabadi-pour and S.Saryazdi, A simultaneous feature adaptation and feature selection method for content-based image retrieval systems, Knowledge based System, 39 (2013) 85-94.

[23] M. Dorigo, G.D. Caro, Ant colony optimization: A new meta-heuristic, in: Proceedings of IEEE Congress on Evolutionary Computing, 1999.

[24] M. Dorigo, V. Maniezzo and A. Colorni, The ant system: optimization by a colony of cooperative agents, IEEE Transaction on System, Man, and Cybernetics, 26 (1) (1996) 1-13.

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

[26] M. Dorigo and L.M.Gambardella, Ant colonies for the traveling salesman problem, BioSystems, 43 (1997) 73-81.

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

[28] C. Blum, Ant colony optimization: introduction and recent trends, Physics of Life Review, 2 (2005) 353-373.

[29] C.K. Zhang and H. Hu, Feature selection using the hybrid of ant colony optimization and mutual information for the forecaster, in: Proceeding of the 4th International Conference on Machine Learning and Cybernetics, 2005, pp. 1728-1732.

[30] M.H. Aghdam, N. Ghasem-Aghaee and M.E. Basiri, Application of ant colony optimization for feature selection in text categorization, in: Proceeding of 5th IEEE Congress on Evolutionary Computation, Hong Kong, 2008.

[31] S. Nemati, M.E. Basiri, N. Ghasem-Aghayee, M.H. Aghdam, A novel ACO-GA hybrid algorithm for feature selection in protein function prediction, Expert Systems with Applications 36 (2009) 12086-12094.

[32] S.H. Xiong, J.Y. Wang, H. Lin, Hybrid feature selection algorithm based on dynamic weighted ant colony algorithm, in: Proceedings of the 9th International Conference on Machine Learning and Cybernetics, Qingdao,2010.

[33] Y.Chen, D. Miao, R. Wang, A rough set approach to feature selection based on ant colony optimization, Pattern Recognition Letters 31 (2010) 226-233.

[34] H. Touhidi, H.Nezamabadi-pour, and S.Saryazdi, Feature selection using binary ant algorithm, Frist Joint Congress on Fuzzy and Intelligent Systems, Mashhad, Iran, Aug. 2007 (In Farsi).

[35] S. Kashef and H. Nezamabadi-pour, A new feature selection algorithm based on binary ant colony optimization, 5th Conference on Information and Knowledge Technology, IKT, Shiraz, Iran, 2013.

[36] M. Janaki Meena, K.R. Chandran, A. Karthik, A. Vijay Samuel, An enhanced ACO algorithm to select features for text categorization and its parallelization, Expert Systems with Applications 39 (2012) 5861–5871.

[37] T. Hiroyasu, M. Miki, Y. One, Y. Minami, Ant colony for continuous functions, The Science and Engineering, Doshisha University, 2000.

[38] N. Karaboga, A. Kalinli, D. Karaboga, Designing digital IIR filters using ant colony optimization algorithm, Engineering Application of Artificial Intelligence, 17(2004) 301-309.

[39] L. Ozbakir, A. Baykasoglu, S. Kulluk and H. Yapici, TACO-miner: An ant colony based algorithm for rule extraction from trained neural networks, Expert Systems with Applications, 36 (2009) 12295-12305.

[40] B. Chen, L. Chen, Y. Chen, Efficient ant colony optimization for image feature selection, Signal Processing 93 (2013) 1566-1576.

[41] R. Jensen, Combining rough and fuzzy sets for feature selection, phd thesis, University of Edinburgh, 2005.

[42] H. Peng, F. Long, C. Ding, Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence, 27 (8) (2005).

[43] C.L. Huang, ACO-based hybrid classification system with feature subset selection and model parameters optimization, Neurocomputing 73 (2009) 438-448.

[44] C.T. Su, H.C. Lin, Applying electromagnetism-like mechanism for feature selection, Information Science, 181 (2011) 972-986.

[45] UCI Machine Learning Repository. Center for Machine Learning and Intelligent Systems. http://archieve.ics.uci.edu/ml/datasets.html>

Shima Kashef received her B.Sc. and M.Sc. degrees in Electrical Engineering from Shahid Bahonar University of Kerman, Iran, in 2011 and 2014, respectively. Her research interests include pattern recognition and evolutionary computation.

Hossein Nezamabadi-pour received his B.Sc. degree in Electrical Engineering from Shahid Bahonar University of Kerman in 1998, and his M.Sc. and Ph.D. degrees in Electrical Engineering from Tarbait Moderres University, Tehran, Iran, in 2000 and 2004, respectively. In 2004, he joined the Department of Electrical Engineering at Shahid Bahonar University of Kerman, Kerman, Iran, as an assistant Professor, and was promoted to full Professor in 2012. Dr. Nezamabadi-pour is the author and co-author of more than 300 peer reviewed journal and conference papers. His interests include image processing, pattern recognition, soft computing, and evolutionary computation.