

Nature-inspired Intelligent Techniques for Pap Smear Diagnosis: Ant Colony Optimization for Cell Classification

Yannis Marinakis¹ and George Dounias²

¹ Department of Production Engineering and Management
Technical University of Crete, University Campus, 73100 Chania
Phone: +30-28210-37282, E-mail: marinakis@ergasya.tuc.gr

² Department of Financial & Management Engineering
University of the Aegean, 31 Fostini Str., 82100 Chios, Greece
Phone: +30- 22710-35454, E-mail: g.dounias@aegean.gr

ABSTRACT: During the last years, Nature Inspired Intelligent Techniques have been very attractive. In this paper, one of the most important Nature Inspired Intelligent Techniques, the Ant Colony Optimization (ACO), is presented for the solution of the Pap Smear Cell Classification problem. ACO 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 paths on its own. Better paths are found as the emergent result of the global cooperation among ants in the colony. This algorithm is combined with a number of nearest neighbor based classifiers. The algorithm is tested in two sets of data. The first one consists of 917 images of Pap smear cells and the second set consists of 500 images, classified carefully by cyto-technicians and doctors. Each cell is described by 20 features, and the cells fall into 7 classes but a minimal requirement is to separate normal from abnormal cells, which is a 2 class problem.

KEYWORDS: Ant Colony Optimization, Feature Selection Problem, Data Mining, Pap-Smear Classification, Nearest Neighbor based Classifiers

INTRODUCTION

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 Colnani [7]. An Ant Colony Optimization algorithm (ACO) is essentially a system based on agents which simulate the natural behavior of ants, including mechanisms of cooperation and adaptation. The ACO algorithm mimics the techniques employed by real ants to rapidly establish the shortest route from food source to their nest and vice versa (adapting to changes in the environment) without the use of visual information. Ants start searching the area surrounding their nest in a random manner and as they move, a certain amount of pheromone (pheromone is the generic name for any endogenous chemical substance secreted by an organism to incite reaction in other organisms of the same specie) is dropped on the ground, marking the path with a trail of this substance. The quantity of the laid pheromone depends upon the distance, quantity and quality of the food source. While an isolated ant that moves at random detects a laid pheromone, it is very likely that it will decide to follow its path. This ant will itself lay a certain amount of pheromone, and hence enforce the pheromone trail of that specific path. The more ants follow a given trail, the more attractive this trail becomes to be followed by other ants. However, as time passes, the pheromone starts to evaporate. The more time it takes for an ant to travel down the path and back again, the more time the pheromone has to evaporate and, thus, the path to become less prominent. A shorter path, in comparison will be visited by more ants (can be described as a loop of positive feedback in which the probability that an ant chooses a path is proportional to the number of ants that have already passed by that path) and, thus, the pheromone density remains high for a longer time. Since ants prefer to follow trails with larger amounts of pheromone, eventually all the ants converge to the shorter path.

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 paths on its own. Better paths are found as the emergent result of the global cooperation among ants in the colony. 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. 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.

In this paper, a novel approach for the solution of the Feature Subset Selection Problem based on the Ant Colony Optimization is presented. In the classification phase of the proposed algorithm, a number of variants of the Nearest Neighbor classification method are used [8]. In order to assess the efficacy of the proposed methodologies, the algorithm is used for the Pap-Smear Cell Classification task. The term "Pap-Smear" refers to samples of human cells stained by the so-called Papanicolaou method. The Papanicolaou method is a medical procedure to detect pre-cancerous cells in the uterine cervix. The performance of the proposed algorithm is tested using two data sets of images of Pap-smear cells distributed unequally on 7 different classes. Each cell is described by 20 features extracted from pictures of single human cells. The rest of the paper is organized as follows: In the next section, a detailed analysis of Ant Colony Optimization for Data Mining is presented. In section 3, an analytical description of the proposed algorithm is given. In section 4 the application of the proposed algorithm to the Pap-Smear Cell Classification task is presented while in the last section conclusions and future research are given.

ANT COLONY OPTIMIZATION FOR DATA MINING

In recent years, there has been an increasing need for novel data mining methodologies that can analyze and interpret large volumes of data. Selecting the right set of features for classification is one of the most important problems in designing a good classifier. The basic feature selection problem is an optimization problem, with a performance measure for each subset of features to measure its ability to classify the samples. 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. These algorithms can be classified into two categories based on whether features are done independently of the learning algorithm used to construct the classifier. If feature selection depends on learning algorithm, the approach is referred to as a *wrapper model*. Otherwise, it is said to be a *filter model*. Filters, such as mutual information (MI), are based on the statistical tools. Wrappers assess subsets of features according to their usefulness to a given classifier. Unfortunately, finding the optimum feature subset has been proved to be NP-hard [12]. Many algorithms are, thus, proposed to find the suboptimum solutions in comparably smaller amount of time [9]. The Branch and Bound approaches (BB) [15], the Sequential Forward/Backward Search (SFS/SBE) [1], [6] and the Filters approaches [5] deterministically search for the suboptimum solutions. One of the most important of the filter approaches is the Kira and Rendell's Relief algorithm [11]. Stochastic algorithms, including Simulated Annealing (SA) [18], Scatter Search algorithms [13] and Genetic Algorithms (GA) [6] are of great interest recently because they often yield high accuracy and are much faster.

The application of ACO to classification is a research area still relatively unexplored. In fact, mining of classification rules is a search problem and ACO is very successful in global search and can cope better with attribute interaction than greedy rule induction algorithms. Furthermore, the application of ant algorithms requires minimum understanding of the problem domain. There are *two kinds* of approaches for the solution of a classification problem using Ant Colony Optimization. The *first one* is based on the discovery of classification rules. Shelokar et al. [19] proposed an ant algorithm for discovery of classification rules and an application of these rules to some process engineering problems. Parpinelli et. Al. [18] proposed an algorithm for data mining called Ant-Miner (Ant Colony-based Data Miner) which is used for extracting classification rules from data. The *second one* is based on the solution of the feature subset selection problem and then, using an independent classifier, the quality of the solution is calculated. Al-Ani [2],[3] presented a novel feature subset search procedure that utilizes the Ant Colony Optimization and used this procedure to select features for speech segment and texture classification problems. Zhang and Hu [20] proposed an algorithm which utilizes the combination of wrapper and filter models: ant colony optimization (ACO) and mutual information (MI).

THE PROPOSED ANT COLONY OPTIMIZATION ALGORITHM

In this paper, as it has already been mentioned, an algorithm for the solution of the feature selection problem based on the Ant Colony Optimization is presented. This algorithm is combined with two nearest neighbour based classifiers, the 1-Nearest Neighbor and the wk-Nearest Neighbor classifier. The steps of the proposed Ant Colony Optimization based classification algorithm are given in the following.

ENCODING

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.

INITIAL POPULATION - CALCULATION OF HEURISTIC FUNCTION

An initial population r of solutions is calculated 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 . Usually in an ACO algorithm, a heuristic value (n_i) is used in conjunction with the pheromone value to decide on the transitions to be made. This value gives an estimate of the quality of each feature with respect to its ability to improve its predictive accuracy. The n_i is calculated from the r_l best solutions ($r_l < r$) of the initial population. We would like to have an initial estimation of the most important features [7]. Thus, the features that exist in the r_l best solutions are calculated and all the features are weighted based on the times that each feature appears in the r_l best solutions. These features have greater fixed value in the n matrix.

ANT SIZE

A number of ants are used. Each ant begins from a different place in the feature vector and follows its own route. All ants start to construct solutions simultaneously. Each ant 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.

INITIAL PHEROMONE

The initial quantity of the pheromone τ_i for the feature i is calculated from the formula:

$$\tau_i = \frac{\text{ant_size}}{\text{init_opt}} \quad (1)$$

where ant_size is the initial population of ants and init_opt is the quality (accuracy) of the optimal solution of the initial population.

SELECTION OF THE FEATURES

An ant located in the feature j decides if the feature i is selected or not by the formula:

$$p_i = \frac{[\tau_i]^\alpha [n_i]^\beta}{\sum_{l=1}^M [\tau_l]^\alpha [n_l]^\beta} \quad (2)$$

where M is the number of features 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 (see section 3.6) 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.

FITNESS FUNCTION

The fitness function gives the quality of the produced member of the population. In this problem, the quality is measured with the Root Mean Squared Error (RMSE) and the overall error. Thus, for each ant the classifiers (1-Nearest Neighbor or wk- Nearest Neighbor) are called and the produced RMSE and the overall error give the fitness function.

Nearest Neighbor Classifiers

Initially, the classic **1-Nearest Neighbor (1-nn)** method is used [8]. The 1-nn works as follows: In each iteration of the feature selection algorithm, a number of features are activated. For each sample of the test set its Euclidean Distance from each sample of the training set is calculated. The Euclidean Distance is calculated as follows:

$$D_{ij} = \sqrt{\sum_{l=1}^d |x_{il} - x_{jl}|^2} \quad (3)$$

where D_{ij} is the distance between the test sample x_{il} and the training sample x_{jl} , and $l = 1, \dots, d$ is the number of activated features in each iteration. With this procedure the nearest sample from the training set is calculated. Thus, each test sample is classified in the same class that its nearest sample from the training set belongs.

The previous approach may be extended to the **k-Nearest Neighbor (k-nn)** method, where we examine the k-nearest samples from the training set and, then, classify the test sample by using a voting scheme. The most common way is to choose the most representative class in the training set. Thus, the k-nn method makes a decision based on the majority class membership among the k nearest neighbors of an unknown sample. In other words every member among the k nearest has an equal percentage in the vote. However, it is natural to give more weight to those members that are closer to the test samples. This method is called **Weighted k-Nearest Neighbor (wk-nn)**. In this method, the i neighbor receives weight

$$w_i = \frac{i}{\sum_{i=1}^k i} \quad (4)$$

Thus, the following hold:

$$w_k \geq w_{k-1} \geq \dots \geq w_1 > 0 \quad (5)$$

$$w_k + w_{k-1} + \dots + w_1 = 1. \quad (6)$$

Performance Measures

In order to estimate the solution of the classifiers, a number of performance measures are calculated. In a **2-class problem** if the actual class of a sample is the N , the estimated class is denoted by T_N (True Negative) if the sample is classified in its actual class and F_P (False Positive) if the model misclassifies the sample. On the other hand, the estimated class is denoted by T_P (True Positive) when the sample is classified correctly in class P and by F_N (False Negative) if the sample is misclassified in the class N . In Table I the definitions are presented.

Estimated Class	Actual Class	
	N	P
\hat{N}	T_N	F_N
\hat{P}	F_P	T_P

Table I: Definitions of the classified and the misclassified samples.

The results of the models are analyzed based on the Root Mean Squared Error, the Error of Group 1, the Error of Group 2 and the Overall Error. The first measure is calculated from the formula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^M (y_i - \hat{y}_i)^2}{M}} \quad (7)$$

where M is the number of samples in the data set, the \hat{y}_i is the classifier model output and the y_i is the true class of the test sample i . Denoting the number of the false negative samples by F_N , the number of the true negative samples by T_N , the number of false positive samples by F_P and the number of true positive samples by T_P , we define the following relative performance measures in percentage.

The Error for Group 1:
$$F_N \% = \frac{F_N}{T_P + F_N} 100 \quad (8)$$

The Error for Group 2:
$$F_P \% = \frac{F_P}{T_N + F_P} 100 \quad (9)$$

The Overall Error is calculated by:
$$OE \% = \frac{F_N + F_P}{T_P + F_N + T_N + F_P} 100. \quad (10)$$

In a classification problem having more than two classes, the classification errors $F_N\%$, $F_P\%$ and $OE\%$ using the definitions above do not make sense. The error of a C class problem can be described using a CxC confusion matrix. The element x_{ij} in row i and column j describes the number of samples, of true class j classified as class i , i.e. all truly classified samples are placed in the diagonal and the remaining misclassified samples in the upper and lower triangular parts. The confusion matrix describes the number of errors, but an error rate is obtained by scaling each column to a norm of 100%. Table II shows the confusion matrix of a **7-class** problem where the classes 1,2, and 3 belong to class N and the classes 4, 5, 6, and 7 belong to class P and, thus, the classification errors $F_N\%$, $F_P\%$ and $OE\%$ of the 2-class problem can easily identified for the 7-class problem.

Estimated Class	Actual Class	
	1-3	4-7
1-3	T_N	F_N
4-7	F_P	T_P

Table II: Confusion matrix for a 7-class problem

UPDATE PHEROMONE

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 [7]. 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:

$$\tau_i \leftarrow \begin{cases} (1-q)\tau_i + \frac{1}{ant_opt}, & \text{if feature } i \text{ is selected} \\ (1-q)\tau_i, & \text{otherwise} \end{cases} \quad (11)$$

where ant_opt is the quality of the best ant and q is an evaporation function 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 [7].

APPLICATION EXAMPLE

PAP-SMEAR CELL CLASSIFICATION

The metaheuristic algorithm presented in the previous sections is used for the classification of cells in cervical smears. Cervical smears are cytology specimens taken from the uterine cervix. A specimen is taken from the uterine cervix using a small brush, a cotton stick or wooden stick and transferred onto a thin, rectangular glass plate (slide). The specimen is stained using the Papanicolaou method which is a medical procedure to find pre-cancerous cells in the uterine cervix. This makes it possible to see characteristics of cells more clearly in a microscope. The purpose of the smear screening is to diagnose premalignant cell changes before they progress to cancer [4],[10],[14],[16]. The specimens most often contain cells from the *columnar epithelium* and the *squamous epithelium*. The columnar epithelium is located in the upper part of the cervix and the squamous epithelium in the lower part. Between these two is the *metaplastic epithelium*, also called the transformation zone or the *squamo-columnar junction*. It takes a skilled cyto-technician to differentiate between the different kinds of cells and it is a time consuming job as every glass slide can contain up to 300,000 cells. In the *squamous* epithelium there are 4 layers of cells: the basal, the parabasal, the intermediate, and finally the superficial layer. The *columnar* epithelium only contains a single layer of cells containing columnar cells and reserve cells. The *metaplastic* epithelium consists of reserve cells from the columnar epithelium. In *dysplastic* cells, the genetic information is somehow changed, and the cell will not divide as it should. This is a precancerous cell. Depending on which kind of cell that divides incorrectly, it is given diagnoses like *dysplasia* (i.e. disordered growth) and *carcinoma in situ*. The dysplastic cells are divided into *mild*, *moderate* and *severe* dysplastic.

DATA AND PARAMETER DESCRIPTION

The classification task is performed using 2 databases, an old one with 500 cells and a new one with 917 cells, built by the Herlev University Hospital. The images were prepared and analyzed by the staff at the hospital using a commercial software package CHAMP (Dimac Imaging – www.dimac-imaging.com) for segmenting the images. The cells were selected, not to collect a natural distribution, but to make a good collection of the important classes (7 classes). For both databases, 20 features are extracted (Table IV). The Pap-smear data fall into 7 classes (Table III), but a minimal requirement is to separate normal from abnormal, which is a 2-class problem.

Old Data	New Data
The old data contains 500 cells with the following distribution:	The new data contains 917 cells with the following distribution:
1. Normal - Columnar epithelial, 50 cells.	1. Normal - Super_cial squamous epithelial, 74 cells.
2. Normal - Parabasal squamous epithelial, 50 cells.	2. Normal - Intermediate squamous epithelial, 70 cells
3. Normal - Intermediate squamous epithelial, 50 cells.	3. Normal - Columnar epithelial, 98 cells.
4. Normal - Super_cial squamous epithelial, 50 cells.	4. Abnormal - Mild squamous non-keratinizing dysplasia, 182 cells.
5. Abnormal - Mild squamous non-keratinizing dysplasia, 100 cells.	5. Abnormal - Moderate squamous non-keratinizing dysplasia, 146 cells.
6. Abnormal - Moderate squamous non-keratinizing dysplasia, 100 cells.	6. Abnormal - Severe squamous non-keratinizing dysplasia, 197 cells.
7. Abnormal - Severe squamous non-keratinizing dysplasia, 100 cells.	7. Abnormal - Squamous cell carcinoma in situ intermediate, 150 cells.

Table III: Description of classes.

Feature number	Feature	Feature number	Feature
1	Nucleus area	11	Cytoplasm longest diameter
2	Cytoplasm area	12	Cytoplasm elongation
3	N/C ratio (Size of nucleus relative to cell size)	13	Cytoplasm roundness
4	Nucleus brightness	14	Nucleus perimeter
5	Cytoplasm brightness	15	Cytoplasm perimeter
6	Nucleus shortest diameter	16	Nucleus position
7	Nucleus longest diameter	17	Maxima in nucleus
8	Nucleus elongation	18	Minima in nucleus
9	Nucleus roundness	19	Maxima in cytoplasm
10	Cytoplasm shortest diameter	20	Minima in cytoplasm

Table IV: Features describing each cell.

It should be noted that from a medical point of view, it is worse to misclassify an abnormal cell as normal, than oppositely. Since we are looking for abnormal cells this is called a positive finding, while a normal cell is called a negative finding. An abnormal cell misclassified as normal is called a false negative finding, and it is important that a classifier minimizes the number of false negative. By analogy, a normal cell misclassified as abnormal is called false positive.

PARAMETERS OF THE PROPOSED ALGORITHM

The data set is divided in two categories the training set and the test set. To test the efficiency of the proposed methods a s-fold validation procedure is utilized. Initially, the data set is divided in s disjoint groups containing approximately M/s samples each, where M is the number of the samples in the data set (for example if a 10-fold validation is used the data set is divided in 10 equal size disjoint groups). Next, each of these groups is systematically removed from the data set, a model is building from the remaining groups (the training set) and, then, the accuracy of the model is calculated using the hidden group (the test set). If the 10-fold validation is used, the procedure is repeated for ten times having different group hidden, and, then, the average accuracies are measured. In this paper, the accuracy of the proposed algorithm is measured using the 10-fold cross validation procedure.

In the proposed algorithms we tested different k's. The first choice is the 2-fold validation where the set is divided in two equal parts. The other choices are the 3-fold (66% training set - 33% test set), 4-fold (75% - 25%), 5-fold (80% - 20%), 10-fold (90% - 10%) and the last is 20-fold (95% - 5%). 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.

As it has already been mentioned two approaches that use different classifiers, the 1-nn and the wk-nn, are used. The first approach is called the **ACO-1nn**, and the other is called **ACO-wknn**. In ACO-wknn the value of k is changed dynamically depending on the number of iteration.

The parameter settings for the ACO based metaheuristic are:

- The number of ants used is equal to the number of features (20) 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 50.
- $q=0.5$.

RESULTS OF 2-CLASS PROBLEM

Table V shows the errors (the Root Mean Square Error, the Error for group 1 and for group 2, $F_N\%$ and $F_P\%$, and the Overall Error) for the proposed algorithms in the 2-class problem for both data sets. We observe that the proposed algorithms give, for example, in the 2-fold Cross Validation, where the data set is divided in two equal parts and the half samples belongs to the training set while the other samples belongs to the test set, very good results as the RMSE is between 0,218828 - 0,249235 for the new data set and 0,107967 for the old data set. In this case, in the new data set although ACO – 1nn has the best performance based on the RMSE, the $F_P\%$ and the $OE\%$, the value of $F_N\%$ is not as good as in the other method, meaning is equal to 2,669306 in ACO-1nn while for ACO – wknn is equal to 1,776465. As the value of s in the s-fold Cross Validation is increasing the results are improved taking into account the values of the four performance measures. For example, the best solution in the 10-fold Cross Validation has in the new data set the RMSE equal to 0,126588 and in the old data set has the RMSE equal to 0 while in the 20-fold Cross Validation in the new data set the RMSE has been improved and is equal to 0,014744 and in the old data set in both methods all the performance measures are equal to 0. From this Table it can be observed that both methods give very good results and these results are almost identical for all methods. However, a higher performance is observed for the ACO-1nn. It should, also, be noted that the value of the most significant performance measure, the $F_N\%$, is always smaller than 2,669306 and especially in 10-fold and 20-fold Cross Validation is smaller than 1 and for a number of methods is equal to 0 in the new data set while in the old data is smaller than 1 in all methods independent of the selected cross validation.

	NEW DATA				OLD DATA			
	RMSE	$F_N\%$	$F_P\%$	$OE\%$	RMSE	$F_N\%$	$F_P\%$	$OE\%$
<i>2-fold Cross Validation (50% Training Set – 50% Test Set)</i>								
ACO – 1nn	0,218828	2,669306	10,7438	4,798737	0,107967	1	1,5	1,2

ACO – wknn	0,249235	1,776465	18,59504	6,216285	0,107967	0,666667	2	1,2
<i>3-fold Cross Validation (66% Training Set – 33% Test Set)</i>								
ACO – lnn	0,19486	1,925926	9,089506	3,816565	0,088144	0,666667	1,002563	0,799606
ACO – wknn	0,207896	1,777778	11,57922	4,362299	0,098828	0,333333	1,997588	0,999206
<i>4-fold Cross Validation (75% Training Set – 25% Test Set)</i>								
ACO – lnn	0,188547	1,776028	8,674863	3,597874	0,031623	0	1	0,4
ACO – wknn	0,20535	2,222457	9,918033	4,251946	0,031623	0,333333	0,5	0,4
<i>5-fold Cross Validation (80% Training Set – 20% Test Set)</i>								
ACO – lnn	0,1711	1,481481	7,040816	2,945474	0,048284	0,333333	1	0,6
ACO – wknn	0,183286	1,185185	9,532313	3,382038	0	0	0	0
<i>10-fold Cross Validation (90% Training Set – 10% Test Set)</i>								
ACO – lnn	0,126588	0,739684	4,933333	1,853798	0	0	0	0
ACO – wknn	0,159971	0,445566	8,7	2,618251	0	0	0	0
<i>20-fold Cross Validation (95% Training Set – 5% Test Set)</i>								
ACO – lnn	0,014744	0,147059	0,416667	0,217391	0	0	0	0
ACO – wknn	0,069403	0	4,102564	1,086957	0	0	0	0

Table V: Results of the algorithms in the 2-class problem

The selection of a set of appropriate input feature variables is an important issue in building a good classifier. 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 a classifier in order to have a good predictive and less computationally intensive model. In the Pap-smear cell classification problem analysed in this paper, there are 2^{20} possible feature combinations. The objective of the computational experiments is to show the performance of the proposed algorithm in searching for a reduced set of features with low RMSE. In Table VI, the average number of features that each algorithm selects for both data bases is presented. It can be seen that in all cases the number of features selected are fewer than the total number of features used in order to describe a cell. For the new data set, the minimum average number of features used is equal to 8,5 observed in ACO-lnn in 4-fold cross validation while the maximum average number of features used is equal to 11 observed in ACO-wknn in 4-fold cross validation and ACO-lnn in 3-fold cross validation. For the old data set, the minimum average number of features used is equal to 7,75 observed in ACO-lnn in 4-fold cross validation while the maximum average number of features used is equal to 14,5 observed in ACO-wknn in 2-fold cross validation.

	<i>2-fold</i>	<i>3-fold</i>	<i>4-fold</i>	<i>5-fold</i>	<i>10-fold</i>	<i>20-fold</i>
NEW DATA						
ACO – lnn	10	11	8,5	9,8	9,7	9,7
ACO – wknn	10	10,33	11	10,8	10,3	9,4
OLD DATA						
ACO – lnn	10	9	7,75	11,4	10,5	8,9
ACO – wknn	14,5	11,67	11,25	11,4	10,4	11,25

Table VI: Results of the algorithms (average number of features used) for the 2-class problem.

RESULTS OF 7-CLASS PROBLEM

Table VII shows the errors (the Root Mean Square Error and the Overall Error) for the proposed algorithms in the 7-class problem for both data sets (as it has already been mentioned the $F_N\%$ and the $F_P\%$ do not make sense in the 7-class problem). We observe that the proposed algorithms give, for example, in the 2-fold Cross Validation, where the data set is divided in two equal parts and the half samples belongs to the training set while the other samples belongs to the test set, very good results as the RMSE is between 1,020786 – 1,030028 for the new data set and between 0,898933 – 0,914948 for the old data set. As the value of s in the s -fold Cross Validation is increasing the results are improved taking into account the values of the four performance measures. For example, the best solution in the 10-fold Cross Validation has in the new data set the RMSE equal to 0,77879 and in the old data set has the RMSE equal to 0,524292 while in the 20-fold Cross Validation in the new data set the RMSE has been improved and is equal to 0,638976 and in the old data set is equal to 0,366199. From this Table it can be observed that all methods give very good results and these results are almost identical for all methods. However, a higher performance is observed for the ACO-lnn.

	NEW DATA		OLD DATA	
	RMSE	OE%	RMSE	OE%
<i>2-fold Cross Validation (50% Training Set – 50% Test Set)</i>				
ACO – 1nn	1,030028	5,780556	0,898933	1,6
ACO – wknn	1,020786	6,652729	0,914948	1,8
<i>3-fold Cross Validation (66% Training Set – 33% Test Set)</i>				
ACO – 1nn	0,910197	4,362299	0,819103	1,39961
ACO – wknn	1,060677	5,234473	0,807685	1,398408
<i>4-fold Cross Validation (75% Training Set – 25% Test Set)</i>				
ACO – 1nn	0,893888	4,687678	0,704141	0,4
ACO – wknn	0,968416	5,779381	0,710849	0,8
<i>5-fold Cross Validation (80% Training Set – 20% Test Set)</i>				
ACO – 1nn	0,887065	4,362675	0,710722	1,2
ACO – wknn	0,911689	4,909123	0,664113	0,4
<i>10-fold Cross Validation (90% Training Set – 10% Test Set)</i>				
ACO – 1nn	0,77879	2,943144	0,524292	0
ACO – wknn	0,812723	3,823459	0,540925	0
<i>20-fold Cross Validation (95% Training Set – 5% Test Set)</i>				
ACO – 1nn	0,638976	2,070048	0,366199	0
ACO – wknn	0,70262	2,181159	0,393758	0

Table VII: Results of the algorithms in the 7-class problem

In the 7-class problem, the same objective with the one of the 2-class problem concerning the number of features selected (i.e., to have a high performance of the proposed algorithm in searching for a reduced set of features with low RMSE) is tested. In Table VIII, the average number of features that each algorithm selects for both data bases are presented. It can be seen that in all cases the number of features selected are fewer than the total number of features used in order to describe a cell. For the new data set, the minimum average number of features used is equal to 9 observed in ACO-1nn in 2-fold cross validation and in ACO-wknn in 5-fold cross validation while the maximum average number of features used is equal to 10,9 observed in ACO-wknn in 10-fold cross validation. For the old data set, the minimum average number of features used is equal to 8,25 observed in ACO-1nn in 20-fold cross validation while the maximum average number of features used is equal to 11,5 observed in ACO-1nn in 2-fold cross validation.

	<i>2-fold</i>	<i>3-fold</i>	<i>4-fold</i>	<i>5-fold</i>	<i>10-fold</i>	<i>20-fold</i>
NEW DATA						
ACO – 1nn	9	10,33	10	9,2	9,2	9,7
ACO – wknn	10,5	9,67	10	9	10,9	9,4
OLD DATA						
ACO – 1nn	11,5	10,33	9,75	10,6	10,4	8,25
ACO – wknn	13	10,33	12	11,2	9,33	9,2

Table VIII: Results of the algorithms (average number of features used) for the 7-class problem.

In Table IX, the times that each feature was selected in the optimal solutions of all algorithms are presented. The five most important features that are used in the algorithms for the two Datasets and for both 2-class and 7-class problems are typed with bold letters. As it can be seen, the three most important features are the fifth feature (Cytoplasm brightness) as it was selected totally 247 times, i.e. in the 70,00% in all solutions, the fourth feature (Nucleus brightness) as it was selected totally 233 times, i.e. in the 66,2% in all solutions and the third feature (N/C ratio (Size of nucleus relative to cell size)) as it was selected totally 226 times, i.e. in the 64,2% in all solutions. The five less important features that are used in the algorithms for the two Datasets and for both 2-class and 7-class problems are typed with italic letters. As it can be seen, the three less important features are the ninth feature (Nucleus roundness) as it was selected totally only 100 times, i.e. in the 28% in all solutions, the eighth feature (Nucleus elongation) as it was selected totally 116 times, i.e. in the 32,95% in all solutions and the twelfth feature (Cytoplasm elongation) as it was selected 132 times, i.e. in the 37,5% in all solutions.

Features	<i>2-class problem</i>				<i>7-class problem</i>			
	NEW DATA		OLD DATA		NEW DATA		OLD DATA	
	Times Selected	Average (%)	Times Selected	Average (%)	Times Selected	Average (%)	Times Selected	Average (%)
1	52	59,09091	51	57,95455	60	68,18182	50	56,818
2	48	54,54545	41	46,59091	48	54,54545	44	50
3	58	65,90909	47	53,40909	65	73,86364	56	63,636
4	63	71,59091	48	54,54545	58	65,90909	64	72,727
5	64	72,72727	53	60,22727	63	71,59091	67	76,136
6	43	48,86364	36	40,90909	35	39,77273	43	48,864
7	57	64,77273	46	52,27273	53	60,22727	48	54,545
8	26	29,54545	45	51,13636	24	27,27273	21	23,864
9	21	23,86364	31	35,22727	23	26,13636	25	28,409
10	38	43,18182	41	46,59091	41	46,59091	42	47,727
11	40	45,45455	40	45,45455	39	44,31818	55	62,5
12	36	40,90909	38	43,18182	24	27,27273	34	38,636
13	33	37,5	43	48,86364	34	38,63636	29	32,955
14	45	51,13636	35	39,77273	49	55,68182	30	34,091
15	46	52,27273	55	62,5	44	50	40	45,455
16	41	46,59091	70	79,54545	32	36,36364	68	77,273
17	36	40,90909	41	46,59091	43	48,86364	32	36,364
18	41	46,59091	52	59,09091	39	44,31818	30	34,091
19	39	44,31818	47	53,40909	30	34,09091	40	45,455
20	40	45,45455	53	60,22727	49	55,68182	35	39,773

Table IX: Results of the algorithms (times each feature was selected).

CONCLUSIONS

In this paper, a classification algorithm is proposed for solving the Pap-smear cell classification problem. Different classifiers are used for the classification problem, based on the Nearest Neighbor classification rule (the 1-Nearest Neighbor and the wk-Nearest Neighbor) and the Ant Colony Optimization method is used for the Feature Selection Problem. The performance of the proposed algorithm is tested using two data sets of Pap-smear cells. The obtained results indicate the high performance of the proposed algorithm in searching for a reduced set of features (in almost all cases less than 50% of all features are used) with high accuracy and in achieving excellent classification of Pap-smear cells both in 2 classes and in 7 classes. Future research is intended to be focused in using different than the Nearest Neighbor classifiers and different algorithms for the Feature Selection Problem.

REFERENCES

- [1] Aha D.W. and Bankert R.L., 1996, "A Comparative evaluation of sequential feature selection algorithms". In Artificial Intelligence and Statistics, Fisher D. and Lenx J.-H. (Eds.), Springer-Verlag, New York.
- [2] Al-Ani A., 2005a, "Feature subset selection using ant colony optimization". International Journal of Computational Intelligence, 2(1), pp. 53-58.
- [3] Al-Ani A., 2005b, "Ant colony optimization for feature subset selection". Transactions on Engineering, Computing and Technology, 4, pp. 35-38.
- [4] Byriel, J., 1999, "Neuro-fuzzy classification of cells in cervical smears". Master's Thesis, Technical University of Denmark: Oersted-DTU, Automation.
- [5] Cantu-Paz E., 2004, "Feature subset selection, class separability, and genetic algorithms". Genetic and Evolutionary Computation Conference, pp. 959-970.
- [6] Cantu-Paz E., Newsam S. and Kamath C., 2004, "Feature selection in scientific application". Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 788-793.

- [7] Dorigo M. and Stützle T., 2004, *Ant Colony Optimization*. A Bradford Book, The MIT Press Cambridge, Massachusetts, London, England.
- [8] Duda R.O. and Hart P. E., 1973, *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York,.
- [9] Jain A. and Zongker D., 1997, "Feature selection: Evaluation, application, and small sample performance". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, pp. 153-158.
- [10] Jantzen, J., Norup, J., Dounias, G. and Bjerregaard B., 2006, "Pap-smear benchmark data for pattern classification". (submitted).
- [11] Kira K. and Rendell L., 1992, "A practical approach to feature selection". *Proceedings of the Ninth International Conference on Machine Learning*, Aberdeen, Scotland, pp. 249-256.
- [12] Kohavi R. and John G., 1997, "Wrappers for feature subset selection". *Artificial Intelligence*, 97, pp. 273-324.
- [13] Lopez F.G., Torres M.G., Batista B.M., Perez J.A.M. and Moreno-Vega. J.M., 2006, "Solving feature subset selection problem by a parallel scatter search". *European Journal of Operational Research*, 169, pp. 477-489.
- [14] Martin, E., 2003, "Pap-smear classification", Master's Thesis, Technical University of Denmark: Oersted-DTU, Automation.
- [15] Narendra P.M. and Fukunaga K., 1977, "A branch and bound algorithm for feature subset selection". *IEEE Transactions on Computers*, 26(9), pp. 917-922.
- [16] Norup, J., 2005, "Classification of pap-smear data by transductive neuro-fuzzy methods", Master's Thesis, Technical University of Denmark: Oersted-DTU, Automation.
- [17] Parpinelli R.S., Lopes, H.S., and Freitas, A.A., 2002, "An ant colony algorithm for classification rule discovery". In *Data mining: A heuristic approach*, Abbas H., Sarker R. and Newton C. (Eds.), London, UK: Idea group publishing, pp. 191-208.
- [18] Siedlecki W. and Sklansky J., 1988, "On automatic feature selection". *International Journal of Pattern Recognition and Artificial Intelligence*, 2(2), pp. 197-220.
- [19] Shelokar P.S., Jayaraman V.K. and Kulkarni B.D., 2004, "An ant colony classifier system: application to some process engineering problems". *Computers and Chemical Engineering*, 28, pp. 1577-1584.
- [20] Zhang C. and Hu H., 2005, "Ant colony optimization combining with mutual information for feature selection in support vector machines". In *AI 2005, LNAI 3809*, Zhang S. and R. Jarvis (Eds.), pp. 918-921.