J Glob Optim (2008) 42:279–293
DOI 10.1007/s10898-007-9242-1

# Optimization of nearest neighbor classifiers via metaheuristic algorithms for credit risk assessment

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

Received: 7 September 2007 / Accepted: 13 September 2007 / Published online: 11 October 2007

**Abstract** The classification problem consists of using some known objects, usually described by a large vector of features, to induce a model that classifies others into known classes. The present paper deals with the optimization of Nearest Neighbor Classifiers via Metaheuristic Algorithms. The Metaheuristic Algorithms used include tabu search, genetic algorithms and ant colony optimization. The performance of the proposed algorithms is tested using data from 1411 firms derived from the loan portfolio of a leading Greek Commercial Bank in order to classify the firms in different groups representing different levels of credit risk. Also, a comparison of the algorithm with other methods such as UTADIS, SVM, CART, and other classification methods is performed using these data.

**Keywords** Metaheuristic algorithms · Feature selection · Classification · Credit risk assessment

Y. Marinakis · N. Matsatsinis
Decision Support Systems Laboratory Department of Production Engineering and Management, Technical University of Crete, 73100 Chania, Greece
e-mail: marinakis@ergasya.tuc.gr

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

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

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

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

## 1 Introduction

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 objective of feature selection is to search through the space of feature subsets to identify the optimal or near-optimal one with respect to a selected 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 selected independently of the learning algorithm used to construct the classifier. If feature selection depends on the 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 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 [19]. Many algorithms are, thus, proposed to find suboptimum solutions in comparably smaller amount of time [17]. The branch and bound approaches (BB) [22], the sequential forward/backward search (SFS/SBE) [1,4,23] and the Filter approaches [3,4] search for suboptimum solutions. One of the most important filter approaches is the Kira and Rendell's Relief algorithm [18]. Stochastic algorithms, including simulated annealing (SA) [28], Scatter Search algorithms [20] and genetic algorithms (GA) [4,27] are of great interest because they often yield high accuracy and are much faster.

In this paper, three approaches for the solution of the feature selection problem are presented. These three approaches are based on three classic metaheuristic techniques, the Tabu Search [10], a Genetic Algorithm [12] and the Ant Colony Optimization [5]. In the classification phase of the proposed algorithm a number of variants of the Nearest Neighbor classification method are used [8]. The performance of the the proposed methodologies is evaluated using a credit risk assessment data sets. Credit risk evaluation is a very challenging and important problem in the domain of financial management and many classification methods have been suggested in the literature to tackle this problem. The performance of the proposed algorithms are tested using data from 1411 firms derived from the loan portfolio of a leading Greek Commercial Bank in order to classify the firms in different groups representing different levels of credit risk.

The rest of the paper is organized as follows: In the next section the performance measures used in the comparisons are presented. In Sect. 3 an analytical description of the nearest neighbor classifier is given. In the fourth section the three metaheuristic algorithms used for the solution of the feature selection problem, i.e., the Tabu Search, the Genetic Algorithm and the Ant Colony Optimization are described. In Sect. 5 the Credit Risk Assessment Problem is presented. Computational results are given in the sixth section while in the last section conclusions and future research are given.

## 2 Performance measures and cross validation

### 2.1 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 1, the estimated class is denoted by $T_1$ (True 1) if the sample is classified in its actual class and $F_2$ (False 2) if the model misclassifies the sample. On the other hand, the estimated class is denoted by $T_2$

**Table 1** Definitions of the classified and the misclassified samples

| | | Actual class | |
|---|---|---|---|
| | | 1 | 2 |
| Estimated | 1 | $T_1$ | $F_1$ |
| class | 2 | $F_2$ | $T_2$ |

(True 2) when the sample is classified correctly in class 2 and by $F_1$ (False 1) if the sample is misclassified in the class 1. In Table 1 the definitions are presented.

The results of the models are analyzed based on the root mean squared error, the accuracy rates for each group and the overall classification accuracy. The first measure is calculated from the formula:

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

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 $i$.

The second measure is the accuracy of group 1:

$$T_1\% = \frac{T_1}{T_1 + F_2} 100 \tag{2}$$

The third measure is the accuracy of group 2:

$$T_2\% = \frac{T_2}{T_2 + F_1} 100 \tag{3}$$

Finally, the overall classification accuracy is calculated by:

$$OCA = \frac{T_1 + T_2}{T_1 + F_2 + T_2 + F_1} 100 \tag{4}$$

### 2.2 Cross validation

The data set is divided in two sub-samples, 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 built from the remaining groups (the training set) and, then, the accuracy of the model is calculated using the test set. If the 10-fold validation is used, the procedure is repeated for ten times and the average accuracies are measured. In this paper, the accuracy of the proposed algorithms is measured using 10-fold cross validation.

## 3 Nearest neighbor classifiers

Nearest Neighbor methods are among the most popular for classification [8]. They represent the earliest general methods proposed for this problem and were heavily investigated in the fields of statistics and pattern recognition. The nearest neighbor technique is a simple and appealing method to address classification problems. In this paper, a number of variations of the nearest neighbor rules are used.

Initially, the classic *1 - Nearest Neighbor (1-nn)* method is used. The 1 - nn works as follows: In each iteration of the feature selection algorithm (Sect. 4), 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:

$$D_{ij} = \sqrt{\sum_{l=1}^{d} |x_{il} - x_{jl}|^2} \tag{5}$$

where $D_{ij}$ is the distance between the test sample $i$ and the training sample $j$, and $l = 1, \ldots, d$ 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.

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 most distant neighbors from the test sample is denoted by $i = 1$ while the nearest neighbor is denoted by $i = k$. The $i$ neighbor receives weight

$$w_i = \frac{i}{\sum_{i=1}^{k} i} \tag{6}$$

Thus, the following hold:

$$w_k \geq w_{k-1} \geq \cdots \geq w_1 > 0 \tag{7}$$

$$w_k + w_{k-1} + \cdots + w_1 = 1. \tag{8}$$

## 4 Feature subset selection problem

### 4.1 Tabu search

The first metaheuristic used in this paper for the solution of the classification problem is called *Tabu-1nn*. The Tabu-1nn uses the Tabu Search method for the solution of the feature subset selection problem and the 1- Nearest Neighbor method as a classifier.

*Tabu search (TS)* was introduced by Glover [10,11] as a general iterative metaheuristic for solving combinatorial optimization problems. Computational experience has shown that TS is a well established approximation technique, which can compete with almost all known techniques and which, by its flexibility, can beat many classic procedures. It is a form of local neighbor search. Each solution $S$ has an associated set of neighbors $N(S)$. A solution $S' \in N(S)$ can be reached from $S$ by an operation called a *move*. TS moves from a solution to its best admissible neighbor, even if this causes the objective function to deteriorate. To avoid cycling, solutions that have been recently explored are declared *forbidden or tabu* for a number of iterations. The tabu status of a solution is overridden when certain criteria (*aspiration criteria*) are satisfied. Sometimes, *intensification* and *diversification* strategies are used to improve the search. In the first case, the search is accentuated in the promising regions of the feasible domain. In the second case, an attempt is made to consider solutions in a broad area of the search space.

In the following, an analytical description of the way the Tabu Search is implemented for the solution of the feature subset selection problem is given. The algorithm starts from an initial solution, i.e., an initial choice of the activated features. The feature selection vector is represented by a 0/1-bit string where 0 shows that the feature is not activated (not included in the solution) while 1 shows that the feature is activated. Initially, only two features are, randomly, activated and the overall classification accuracy of the solution is calculated with the 1-nn classifier, as described previously. Afterwards, a neighbor solution is generated. The neighbors are generated by randomly activating or deactivating a feature. Among the neighbors, the one with the best accuracy is selected and considered as a new current solution for the next iteration. Two restrictions in the implementation of the algorithm are used. The first one is that the feature vector is not allowed to have less than two features activated while the second one is the *Tabu Moves*. With the term Tabu Moves we mean that a tabu list is maintained to avoid returning to previously visited solutions. Thus, a feature that is added or deleted in an iteration is not allowed to return to the solution for a number of iteration equal with the size of the tabu list. The aspiration criterion that is used is a mechanism that overrides the tabu status of moves, meaning that if a move leads to a sufficient good solution even if it is tabu then the restriction of the tabu list is not activated and the move is allowed.

We have, also, used an intensification and a diversification strategy. In the intensification strategy, the most promising regions of the search space are explored, i.e., the features with the less contribution in the quality of the solution are removed and the search is continued from the current solution. In the diversification strategy, the unexplored regions of the search space are explored in order to achieve a better solution. Two termination criteria are used. The first one is a prespecified maximum number of iterations and with the second termination criterion the algorithm stops if there is no improvement in the objective function value after a number of iterations.

Except of the Tabu-1nn metaheuristic the *Tabu-knn* and the *Tabu-wknn* are also used. In the Tabu-knn instead of the 1-Nearest Neighbor classifier, the k-Nearest Neighbor classifier is used while in the Tabu-wknn the weighted k-Nearest Neighbor classifier is used.

### 4.2 Genetic algorithms

The second proposed method for the solution of the feature selection problem is a genetic algorithm (GA). Genetic algorithms are search procedures based on the mechanics of natural selection and natural genetics. The first GA was developed by John H. Holland in the 1960s to allow computers to evolve solutions to difficult search and combinatorial problems, such as function optimization and machine learning [14]. Genetic algorithms offer a particularly attractive approach for problems like feature subset selection since they are generally quite effective for rapid global search of large, non-linear and poorly understood spaces. Moreover, genetic algorithms are very effective in solving large-scale problems. Genetic algorithms [12,26] mimic the evolution process in nature. GAs are based on an imitation of the biological process in which new and better populations among different species are developed during evolution. Thus, unlike most standard heuristics, GAs use information about a population of solutions, called individuals, when they search for better solutions. A GA is a stochastic iterative procedure that maintains the population size constant in each iteration, called a generation. Their basic operation is the mating of two solutions in order to form a new solution. To form a new population, a binary operator called crossover, and a unary operator, called mutation, are applied [24,25]. Crossover takes two individuals, called parents, and produces two new individuals, called offsprings, by swapping parts of the parents.

Thus, for the solution of the classification problem a metaheuristic called **Gen-1nn** is used. In this metaheuristic algorithm, a genetic algorithm is used for the solution of the feature selection problem while the 1-Nearest Neighbor is used as a classifier.

The basic phases of the proposed genetic algorithm (Gen-1nn) are the following:

*Encoding* Each individual in the population represents a candidate solution to the feature subset selection problem. Let $m$ be the total number of features. The individual (chromosome) is represented by a binary vector of dimension $m$. If a bit is equal to 1 it means that the corresponding feature is selected (activated); otherwise the feature is not selected. This is the simplest and most straightforward representation scheme.

*Initial population* The initial population is generated at random. Thus, in order to explore subsets of different numbers of features, the number of 1 s for each individual is generated at random. Only different individuals are allowed. Thus, in the initial population there are not individuals with the same characteristics. With this way we ensure the diversity of the initial population.

*Fitness function* The fitness of the individual is based on the overall classification accuracy of the evolved subset of features to predict class values for unseen cases. The fitness function gives the quality of the produced member of the population. In this problem the quality is measured with the RMSE and the OCA. Thus, for each individual the 1-nn classifier is called and the produced RMSE and OCA give the fitness function.

*Selection of the parents* The selection mechanism is responsible for selecting the parent chromosome from the population and forming the mating pool. The selection mechanism emulates the survival of- the-fittest mechanism in nature. It is expected that a fitter chromosome has a higher chance of surviving on the subsequent evolution. In this work, we are using the roulette wheel selection [21] which is one of the most common and easy to implement selection mechanisms. Basically, it works as follows: each individual in the population is associated with a sector in a virtual wheel. According to the fitness value of the individual, the sector will have a larger area when the corresponding individual has a better fitness value while a lower fitness value will lead to a smaller sector.

*Crossover operator* In a specific percentage of the individuals (80% in our case) a crossover phase is applied. The most classic crossover operator, the 1-point crossover, is used in the crossover phase of the algorithm. In the 1-point crossover, the two parents are separated in two parts in a randomly determined point. Then, the two offsprings take the 1-part from the one parent and the second part of the other parent. Afterwards for each offspring its fitness function is calculated.

For example:

$$\begin{array}{lr|l} \text{parent 1}: & 1001 & 10101 \\ \text{parent 2}: & 1111 & 00000 \end{array}$$

The two parents are separated between the fourth and the fifth feature and the two produced offsprings are the following:

$$\begin{array}{lr|l} \text{Offspring 1}: & 1001 & 00000 \\ \text{Offspring 2}: & 1111 & 10101 \end{array}$$

*Mutation operator* In a specific percentage of the offsprings (25% in our case) a mutation phase is applied. Mutation operates on a single string and generally changes a bit at random. Afterwards for each offspring its fitness function is calculated.

For example:

| | |
|---|---|
| Before mutation: | 111**1**10101 |
| After mutation: | 111**0**10101 |

*Next generation - Survival of the fittest* In the next generation, the fittest individual from the whole population survives. With the term whole population we mean the initial population and the offsprings from both mutation and crossover phases. Thus, the population is sorted based on the fitness function of the individuals and in the next generation the fittest individuals survive. It must be mentioned that the size of the population of each generation is equal to the initial size of the population.

*Stopping criteria* There are two stopping criteria for the genetic algorithm. The one is the maximum number of generations and the other is the genetic convergence, which means that whenever the solutions of the genetic algorithm converge to one solution the genetic algorithm stops.

Besides the Gen-1nn two other variations are used, the **Gen-knn** where instead of the 1-Nearest Neighbor classifier in the fitness function the k-Nearest Neighbor classifier is used and the **Gen-wknn** where the weighted k-Nearest Neighbor classifier is used. In both Gen-knn and Gen-wknn, the value of k is changed dynamically depending on the number of generation. Each generation uses different k. The reason that k does not have a constant value is that we would like to ensure the diversity of the individuals in each generation. Thus, the genetic convergence will not be achieved too soon and this will probably lead to a better solution.

### 4.3 Ant colony optimization

The last proposed algorithm for the solution of the feature selection problem is based on ant colony optimization. The ant colony optimization (ACO) metaheuristic is a relatively new technique for solving combinatorial optimization problems (COPs). Based strongly on the ant system (AS) metaheuristic developed by Dorigo, Maniezzo and Colorni [5], ant colony optimization is derived from the foraging behaviour of real ants in nature.

The main idea of ACO is to model the problem as the search for a minimum cost path in a graph. Artificial ants walk through this graph, looking for good paths. Each ant has a rather simple behaviour so that it will typically only find rather poor-quality 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.

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. Zhang and Hu [31] proposed an algorithm which utilizes the combination of wrapper and filter models: ant colony optimization (ACO) and mutual information (MI).

The steps of the proposed algorithm *(ACO-1nn)*, ACO for the solution of the feature selection problem and 1-Nearest Neighbor as a classifier, are 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 formed in order to find an initial local optimum solution to use it in the calculation of the heuristic function $n_i$ of the feature $i$. The $n_i$ is calculated from the $r_1$ best solutions ($r_1 < r$) of the initial population. We would like to have an initial estimation of the most important features [5]. Thus, the features that exist in the $r_1$ best solutions are identified and all the features are weighted based on the times that each feature appears in the $r_1$ best solutions. These features have greater fixed value in the $[n_i]$ matrix, where $[\cdot]$ denotes the $i$ element of the matrix $n$.

*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 $\tau_i$ for the feature $i$ is calculated from the formula:

$$\tau_i = \frac{ant\_size}{init\_opt} \tag{9}$$

where $ant\_size$ is the initial population of ants and $init\_opt$ is the quality (overall classification accuracy) of the optimum 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:

$$p_i = \frac{[\tau_i]^\alpha [n_i]^\beta}{\sum_{l=1}^{m} [\tau_l]^\alpha [n_l]^\beta} \tag{10}$$

where $m$ is the number of features, $[\cdot]$ denotes the $i$ element of the matrices $\tau$, $n$ and $\alpha$, $\beta$ 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 below) 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 RMSE and the overall classification accuracy. Thus, for each ant the 1-nn classifier is called and the produced RMSE and the accuracy give the fitness function.

*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 [5]. 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:

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

where *ant_opt* is the quality of the best ant and $q$ is an evaporation parameter that is used in order not to have a continuous increase of the pheromone values in each feature. The parameter $q$ is used to avoid unlimited accumulation of the pheromone trails and it enables the algorithm to forget bad decisions previously taken [5].

Except of the ACO-1nn, the *ACO-wknn* is used where instead of the 1-Nearest Neighbor classifier in the fitness function, the weighted k-Nearest Neighbor classifier. In ACO-wknn the value of k is changed dynamically depending on the number of iteration. Each generation uses different k. The reason that k does not have a constant value is that we would like to ensure the diversity of the ants in each iteration.

## 5 Credit risk assessment

The metaheuristic algorithms presented in the previous section are implemented to a credit risk assessment problem. Credit risk assessment [7] is a major issue for financial institutions (e.g., banks), for firms that grant credit to their customers, as well as for institutional and individual investors. On a daily basis credit/financial analysts have to investigate an enormous volume of financial and non-financial data of firms, estimate the corresponding credit risk and finally make crucial decisions regarding the financing of firms. Corporate credit risk assessment decisions involve two major issues: the determination of the probability of default and the estimation of potential future benefits and losses for credit granting. The former issue is addressed by classifying the firms seeking credit into homogeneous groups representing different levels of credit risk.

Considerable attention has been devoted in this field from the theoretical and academic points of view during the last three decades. Increasing information about credit risk and rapid advances in computer technology have improved modeling techniques for both consumer and commercial credit. Financial and operational researchers have tried to relate the characteristics of a firm (financial ratios and strategic variables) to its credit risk. According to this relationship the components of credit risk are identified, and decision models are developed to assess credit risk and the corresponding creditworthiness of firms as accurately as possible. Decisions regarding credit risk assessment concern the evaluation of the firms¢ financial and non-financial characteristics in order to make “optimal” decisions which incorporate a tradeoff between the potential risk of loss and the probability of profits from granting credit. Actually, credit-granting decisions are usually realized by credit and financial analysts as sorting (classifying) the firms seeking financing from banks or credit institutions into categories according to their creditworthiness (i.e., creditworthy and insolvent firms). One of the key decisions financial institutions have to make is to decide whether or not to grant a loan to a customer. This decision basically boils down to a binary classification problem which aims at distinguishing firms of low credit risk from firms of high credit risk.

**Table 2** List of financial ratios

| | Financial ratios | Description |
|---|---|---|
| 1 | EBIT/TA | Earnings before interest and taxes/total assets |
| 2 | NI/NW | Net income/net worth |
| 3 | SALES/TA | Sales/total assets |
| 4 | GP/TA | Gross profit/total assets |
| 5 | NI/WC | Net income/working capital |
| 6 | NW/TD | Net worth/total debt |
| 7 | TD/TA | Total debt/total assets |
| 8 | LTD/(LTD+NW) | Long-term debt/(long-term debt+net worth) |
| 9 | CA/CL | Current assets/current liabilities |
| 10 | QA/CL | Quick assets/current liabilities |
| 11 | CASH/CL | Cash/current liabilities |
| 12 | CL/NW | Current liabilities/net worth |
| 13 | CF/TA | Cash flow/total assets |
| 14 | TD/WC | Total debt/working capital |
| 15 | WC/TA | Working Capital/total assets |
| 16 | I/CA | Inventories/Current assets |

## 6 Computational results

### 6.1 Data and parameter description

The data used in this paper are derived from the loan portfolio of the Commercial Bank of Greece, one of the leading Greek commercial banks. Overall 1411 firms are considered from different business sectors. The firms have been classified by the bank's loan officers in two classes: default and non-default. The default class includes 218 firms (class 2) whereas the non-default class consists of 1193 firms (class 1). Thus, the credit risk assessment classification task will be to discriminate between these two groups of firms. On the basis of the available financial data of the firms 16 financial ratios (features) are used as adequate measures of corporate credit risk (Table 2). It should also be noticed that according to the international financial literature the selected ratios cover all aspects of the corporate financial performance, including profitability, solvency and managerial performance. 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. The analysis is based on 10-fold cross validation.

Concerning the specifications and the parameters needed for the proposed algorithms:

- The parameter settings for the *tabu based metaheuristic* are:
  - The maximum number of iterations: 1000.
  - The size of the Tabu List: 10.
- The parameter settings for the *genetic based metaheuristic* are:
  - Population size: 1000.
  - Number of generations: 50.
  - Crossover probability: 0.8.

**Table 3** Number of features selected by the algorithms and RMSE

| Method | RMSE | Folds | | | | | | | | | | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Tabu-1nn | 0.175 | 8 | 8 | 10 | 8 | 8 | 7 | 8 | 9 | 9 | 9 | 8.4 |
| Tabu-3nn | 0.211 | 7 | 10 | 9 | 10 | 9 | 11 | 7 | 9 | 7 | 9 | 8.8 |
| Tabu-5nn | 0.208 | 12 | 9 | 8 | 11 | 8 | 6 | 10 | 7 | 10 | 8 | 8.9 |
| Tabu-8nn | 0.208 | 7 | 8 | 7 | 10 | 8 | 6 | 10 | 7 | 9 | 5 | 7.7 |
| Tabu-10nn | 0.213 | 7 | 8 | 7 | 9 | 8 | 9 | 6 | 8 | 8 | 4 | 7.4 |
| Tabu-w3nn | 0.221 | 7 | 9 | 8 | 11 | 7 | 6 | 9 | 7 | 12 | 6 | 8.2 |
| Tabu-w5nn | 0.204 | 7 | 8 | 9 | 10 | 9 | 6 | 9 | 7 | 12 | 5 | 8.2 |
| Tabu-w8nn | 0.210 | 7 | 6 | 8 | 10 | 8 | 6 | 10 | 7 | 10 | 8 | 8 |
| Tabu-w10nn | 0.205 | 7 | 8 | 7 | 10 | 8 | 8 | 10 | 8 | 10 | 8 | 8.4 |
| Gen-1nn | 0.153 | 10 | 9 | 9 | 8 | 6 | 6 | 9 | 9 | 10 | 7 | 8.3 |
| Gen-knn | 0.176 | 8 | 7 | 8 | 10 | 7 | 7 | 9 | 10 | 8 | 8 | 8.2 |
| Gen-wknn | 0.179 | 10 | 8 | 6 | 10 | 8 | 6 | 7 | 7 | 8 | 9 | 7.9 |
| ACO-1nn | 0.181 | 7 | 7 | 8 | 10 | 6 | 9 | 8 | 6 | 9 | 9 | 7.9 |
| ACO-wknn | 0.156 | 11 | 9 | 4 | 4 | 5 | 8 | 6 | 8 | 10 | 7 | 7.2 |

  - Mutation probability: 0.25.
- The parameter settings for the *ACO based metaheuristic* are:
  - Number of ants: 16 (equal to the number of features, because in the initial iteration each ant begins from a different feature).
  - Number of iterations that each ant constructs a different solution, based on the pheromone trails: 50.
  - $q = 0.5$.

### 6.2 Results of the proposed algorithms

As it has already, been mentioned 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 credit risk assessment problem analysed in this paper, there are $2^{16}$ possible feature combinations. The objective of the computational experiments is to show the performance of the proposed algorithms in searching for a reduced set of features with high accuracy.

Table 3 shows the number of features selected by the algorithms in each run of the 10-fold cross validation process. The results indicate that the algorithms selected almost has of the features to build the final models. The algorithms that select the smallest number of features is the ACO-wknn with 7.2 (on average) and the Tabu-10nn with 7.4. Three others classifiers select less than eight features on average, the Tabu-8nn (7.7), the Gen-wknn (7.9) and the ACO-1nn (7.9).

Table 4 reports the frequency with which each feature is included in the optimal solutions of all algorithms (number and percentage, in parentheses), thus providing an indication on

**Table 4** Selection frequencies of the features in the optimal feature sets

| Features | Frequency | | Features | Frequency | |
|---|---|---|---|---|---|
| 1 | 77 | (55.00) | 9 | 56 | (40.00) |
| 2 | 99 | (70.71) | 10 | 60 | (42.85) |
| 3 | 102 | (72.85) | 11 | 74 | (52.85) |
| 4 | 65 | (46.42) | 12 | 42 | (30.00) |
| 5 | 79 | (56.42) | 13 | 76 | (54.28) |
| 6 | 48 | (34.28) | 14 | 33 | (23.57) |
| 7 | 80 | (57.14) | 15 | 86 | (61.42) |
| 8 | 85 | (60.71) | 16 | 73 | (52.14) |

**Table 5** Classification results (accuracy rates)

| Methods | Group 1 | Group 2 | ACA | OCA |
|---|---|---|---|---|
| Tabu-1nn | 0.9839 | 0.8898 | 0.9369 | 0.9688 |
| Tabu-3nn | 0.9805 | 0.8167 | 0.8986 | 0.9539 |
| Tabu-5nn | 0.9822 | 0.8176 | 0.8999 | 0.9553 |
| Tabu-8nn | 0.9780 | 0.8383 | 0.9081 | 0.9553 |
| Tabu-10nn | 0.9797 | 0.8170 | 0.8983 | 0.9532 |
| Tabu-w3nn | 0.9730 | 0.8298 | 0.9014 | 0.9497 |
| Tabu-w5nn | 0.9822 | 0.8251 | 0.9036 | 0.9568 |
| Tabu-w8nn | 0.9797 | 0.8261 | 0.9029 | 0.9546 |
| Tabu-w10nn | 0.9823 | 0.8258 | 0.9040 | 0.9568 |
| Gen-1nn | 0.9907 | 0.8987 | 0.9447 | 0.9759 |
| Gen-knn | 0.9899 | 0.8516 | 0.9207 | 0.9674 |
| Gen-wknn | 0.9873 | 0.8671 | 0.9272 | 0.9660 |
| ACO-1nn | 0.9899 | 0.8986 | 0.9442 | 0.9752 |
| ACO-wknn | 0.9915 | 0.8341 | 0.9128 | 0.9660 |
| UTADIS | 0.9480 | 0.9673 | 0.9576 | 0.9511 |
| SVM (linear) | 0.9497 | 0.9262 | 0.9379 | 0.9461 |
| SVM (rbf) | 0.9355 | 0.9026 | 0.9190 | 0.9305 |
| CART | 0.9296 | 0.9537 | 0.9416 | 0.9333 |
| Discriminant analysis | 0.9296 | 0.8840 | 0.9068 | 0.9227 |
| Logistic regression | 0.9388 | 0.9121 | 0.9254 | 0.9348 |
| PNN | 0.9288 | 0.9305 | 0.9296 | 0.9291 |
| 3 Nearest neighbors | 0.9179 | 0.8983 | 0.9081 | 0.9149 |

the importance of the features. The results show that the most important feature is sales/total assets (feature 3) which is selected 102 times, i.e., in 72.85% of all cases. The second most important feature is net income/net worth (feature 2) that is selected 99 times, i.e., in 70.71% of all cases. The two less important features is total debt/working capital (feature 14) that is selected only 33 times and current liabilities/net worth (feature 12) which is selected only 42 times (in the 30% of all solutions).

### 6.3 Comparative analysis

In order to evaluate the performance of the proposed metaheuristic algorithms comparisons, were performed with other multicriteria decision analysis methods and machine learning techniques, namely UTADIS [6,16], support vector machines (SVM) [13,30], logistic regression (LR) [9,13], 3 nearest-neighbor algorithm [8,9,13], probabilistic neural networks (PNN) [9,13,29], classification and regression trees (CART) [2] and discriminant analysis [9,15]. These techniques are very popular among financial researchers and practitioners for developing financial classification models.

Table 5 shows the accuracy rates for each group, the overall classification accuracy (OCA), as well as the average classification accuracy (ACA), which is calculated by $ACA = \frac{T_1\%+T_2\%}{2}$ (the best results are denoted in bold). We observe that the proposed methods give very good results in the accuracy of group 1, as for all methods the accuracy is between 0.9730 and 0.9915. The overall accuracy of the proposed methods is between 0.9497 and 0.9752. From Table 6, it is observed that the best of all the proposed algorithms is the Gen-1nn because it has the best value in the OCA (0.9752), the second best value in the ACA (0.9447), the second best value in the accuracy of group 1 (0.9907), and it is in the seventh place in the accuracy for group 2 (0.8987). The second best algorithm is ACO-1nn. ACO-1nn is ranked in the second place concerning the OCA (0.9752), in the third place concerning the ACA (0.9442), in the fourth place concerning the accuracy for group 1 (0.9899) and in the eighth place concerning the accuracy for group 2 (0.8986). From the other algorithms used in the comparisons, the only competitive with the proposed algorithms is the UTADIS, which is ranked first in accuracy for group 2 and in ACA, but it is in the 16th place in the accuracy of group 1 and in the 14th place in the OCA. The results show the higher efficiency of the proposed methods when the 1-Nearest Neighbor method is used as a classifier. This can be observed if we see the values of the most important performance measure, the overall classification accuracy. The Gen-1nn, ACO-1nn and Tabu-1nn outperform all the other proposed algorithms and from all the algorithms that are used for the comparisons. It should, also, be noted that the ACO-1nn algorithm, except of its very good performance based on the accuracy, it uses only 7.9 features (on average) during the cross validation procedure (see Table 3). Based on these characteristics of the ACO-1nn algorithm, we can say that this algorithm has the higher performance of all the proposed algorithms.

## 7 Conclusions and future research

In this paper, three different metaheuristic algorithms, tabu search, genetic algorithms and ant colony aptimization were proposed for solving the feature subset selection problem. Three different classifiers were used for the classification problem, based on the nearest neighbor classification rule (the 1-nearest neighbor, the k-nearest neighbor and the wk-nearest neighbor). The performance of the proposed algorithms was tested using credit rating data in order to classify the firms in different groups representing different levels of credit risk. Also, a comparison with other classification methods was performed using these data.

The results of the analysis show that such metaheuristic algorithms can be successfully applied in the feature selection problem, leading to models that provide good classification results with a small number of features. *ACO-1nn* was found to provide the best results in terms of accuracy rates using almost half of the available features. Concerning the CPU time needed for the three algorithms to solve the feature selection problem, the lower computational

**Table 6** The best results for each evaluation criterion

| Methods | Group 1 | Methods | Group 2 |
|---|---|---|---|
| ACO-wknn | 0.9915 | UTADIS | 0.9673 |
| Gen-1nn | 0.9907 | CART | 0.9537 |
| Gen-knn | 0.9899 | PNN | 0.9305 |
| ACO-1nn | 0.9899 | SVM (linear) | 0.9262 |
| Gen-wknn | 0.9873 | Logistic regression | 0.9121 |
| Method | ACA | Method | OCA |
| UTADIS | 0.9576 | Gen-1nn | 0.9759 |
| Gen-1nn | 0.9447 | ACO-1nn | 0.9752 |
| ACO-1nn | 0.9442 | Tabu-1nn | 0.9688 |
| CART | 0.9416 | Gen-knn | 0.9674 |
| SVM (linear) | 0.9379 | ACO-wknn | 0.9660 |

time was achieved with the Tabu Search metaheuristic. The other two methods have similar computational times.

Future research will focus on the use of different machine learning classifiers (SVM, neural networks, etc.). It would also be worth investigating the use of other popular metaheuristic algorithms such as simulated annealing, the greedy randomized adaptive search procedure and particle swarm optimization.

## References

1. Aha, D.W., Bankert, R.L.: A comparative evaluation of sequential feature selection algorithms. In: Fisher, D., Lenx, J.-H. (eds.), Artificial Intelligence and Statistics, Springer-Verlag, New York (1996)
2. Breiman, L., Friedman, J., Olshen, R., Stone, C.J.: Classification and Regression Trees. Chapman and Hall, New York (1984)
3. Cantu-Paz, E.: Feature Subset Selection, Class Separability, and Genetic Algorithms. Genetic and Evolutionary Computation Conference, pp. 959–970 (2004)
4. Cantu-Paz, E., Newsam, S., Kamath, C.: Feature selection in scientific application. In: Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 788–793 (2004)
5. Dorigo, M., Stutzle, T.: Ant Colony Optimization, A Bradford Book. The MIT Press, Cambridge (2004)
6. Doumpos, M., Zopounidis, C.: Multicriteria Decision Aid Classification Methods Kluwer Academic Publishers, Dordrecht (2002)
7. Doumpos, M., Kosmidou, K., Baourakis, G., Zopounidis, C.: Credit risk assessment using a multicriteria hierarchical discrimination approach: a comparative analysis. Eur. J. Oper. Res. **138**, 392–412 (2002)
8. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification and Scene Analysis, 2nd edn. John Wiley and Sons, New York (2001)
9. Giudici, P.: Applied Data Mining: Statistical Methods for Business and Industry. John Wiley and Sons, Chichester (2003)
10. Glover, F.: Tabu search I. ORSA J. Comput. **1**(3), 190–206 (1989)
11. Glover, F.: Tabu search II. ORSA J. Comput. **2**(1), 4–32 (1990)
12. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company, INC, Massachussets (1989)
13. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning; Data mining, Inference, and Prediction. Springer Series in Statistics, Springer-Verlag, New York (2001)

14. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI (1975)
15. Huberty, C.J.: Applied Discriminant Analysis. John Wiley, New York (1994)
16. Jacquet-Lagrèze, E.: An application of the UTA discriminant model for the evaluation of R&D projects. In: Pardalos, P.M., Siskos, Y., Zopounidis, C. (eds.) Advances in Multicriteria Analysis, pp. 203–211. Kluwer Academic Publishers, Dordrecht (1995)
17. Jain, A., Zongker, D.: Feature selection: evaluation, application, and small sample performance. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**, 153–158 (1997)
18. Kira, K., Rendell, L.: A practical approach to feature selection. In:Proceedings of the Ninth International Conference on Machine Learning, pp. 249–256. Aberdeen, Scotland (1992)
19. Kohavi, R., John, G.: Wrappers for feature subset selection. Artif Intelli **97**, 273–324 (1997)
20. Lopez, F.G., Torres, M.G., Batista, B.M., Perez, J.A.M., Moreno-Vega, J.M.: Solving feature subset selection problem by a parallel scatter search. Eur. J. Oper. Res. **169**, 477–489 (2006)
21. Marinakis, Y., Migdalas, A., Pardalos, P.M.: A hybrid genetic-GRASP algorithm using langrangean relaxation for the traveling salesman problem. J. Combi. Optim. **10**, 311–326 (2005)
22. Narendra, P.M., Fukunaga, K.: A branch and bound algorithm for feature subset selection. IEEE Trans. Comput. **26**(9), 917–922 (1977)
23. Pudil, P., Novovicova, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognit. Lett. **15**, 1119–1125 (1994)
24. Reeves, C.R.: Genetic algorithms. In: Reeves, C.R. (ed.) Modern Heuristic Techniques for Combinatorial Problems, pp. 151–196. McGraw - Hill, London (1995)
25. Reeves, C.R.: Genetic algorithms. In: Glover, F., Kochenberger, G.A. (eds.) Handbooks of Metaheuristics, pp. 55–82. Kluwer Academic Publishers, Dordrecht (2003)
26. Rego, C., Glover, F. : Local search and metaheuristics. In: Gutin, G., Punnen, A. (eds.) The Traveling Salesman Problem and its Variations, pp. 309–367. Kluwer Academic Publishers, Dordrecht (2002)
27. Siedlecki, W., Sklansky, J.: A note on genetic algorithms for large-scale feature selection. Pattern Recognition Lett. **10**, 335–347 (1989)
28. Siedlecki, W., Sklansky, J.: On automatic feature selection. Int. J. Pattern Recogn. Artif. Intell. **2**(2), 197–220 (1988)
29. Specht, D.: Probabilistic neural networks. Neural Networks **3**, 109–118 (1990)
30. Vapnik, V.N.: Statistical Learning Theory. Wiley, New York (1998)
31. Zhang, C., Hu, H.: Ant colony optimization combining with mutual information for feature selection in support vector machines. In: Zhang, S., Jarvis, R. (eds.) AI 2005, pp. 918–921. LNAI 3809 (2005)