



Honey Bees Mating Optimization algorithm for financial classification problems

Magdalene Marinaki^a, Yannis Marinakis^b, Constantin Zopounidis^{c,*}

^a Technical University of Crete, Department of Production Engineering and Management, Industrial Systems Control Laboratory, 73100 Chania, Greece

^b Technical University of Crete, Department of Production Engineering and Management, Decision Support Systems Laboratory, 73100 Chania, Greece

^c Technical University of Crete, Department of Production Engineering and Management, Financial Engineering Laboratory, University Campus, 73100 Chania, Greece

ARTICLE INFO

Article history:

Received 2 December 2008

Received in revised form 8 September 2009

Accepted 10 September 2009

Available online 17 September 2009

Keywords:

Honey Bees Mating Optimization

Metaheuristics

Feature selection problem

Financial classification problem

ABSTRACT

Nature inspired methods are approaches that are used in various fields and for the solution for a number of problems. This study uses a nature inspired method, namely Honey Bees Mating Optimization, that is based on the mating behaviour of honey bees for a financial classification problem. Financial decisions are often based on classification models which are used to assign a set of observations into predefined groups. One important step towards the development of accurate financial classification models involves the selection of the appropriate independent variables (features) which are relevant for the problem at hand. The proposed method uses for the feature selection step, the Honey Bees Mating Optimization algorithm while for the classification step, Nearest Neighbor based classifiers are used. The performance of the method is tested in a financial classification task involving credit risk assessment. The results of the proposed method are compared with the results of a particle swarm optimization algorithm, an ant colony optimization, a genetic algorithm and a tabu search algorithm.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

Several biological and natural processes have been influencing the methodologies in science and technology in an increasing manner in the past years. Feedback control processes, artificial neurons, the DNA molecule description and similar genomics matters, studies of the behaviour of natural immunological systems, and more, represent some of the very successful domains of this kind in a variety of real world applications. During the last decade, nature inspired intelligence becomes increasingly popular through the development and utilization of intelligent paradigms in advanced information systems design. Cross-disciplinary team-based thinking attempts to cross-fertilize engineering and life science understanding into advanced inter-operable systems. The methods contribute to technological advances driven by concepts from nature/biology including advances in structural genomics (intelligent drug design through imprecise data bases), mapping of genes to proteins and proteins to genes (one-to-many and many-to-one characteristics of naturally occurring organisms), modelling of complete cell structures (showing modularity and hierarchy), functional genomics (handling of hybrid sources and heterogeneous and inconsistent origins of disparate databases), self-

organization of natural systems, etc. Among the most popular nature inspired approaches, when the task is optimization within complex domains of data or information, are those methods representing successful animal and micro-organism team behaviour, such as swarm or flocking intelligence (birds flocks or fish schools inspired particle swarm optimization [1]), artificial immune systems (that mimic the biological one [2,3]), or ant colonies (ants foraging behaviours gave rise to ant colony optimization [4,5]), etc.

In the recent few years a number of swarm intelligence algorithms, based on the behaviour of the bees have been presented [6]. These algorithms are divided, mainly, in two categories according to their behaviour in the nature, the foraging behaviour and the mating behaviour. The most important approaches that simulate the foraging behaviour of the bees are the Artificial Bee Colony (ABC) algorithm proposed by Karaboga and Basturk [7,8], the Virtual Bee algorithm proposed by Yang [9], the Bee Colony Optimization algorithm proposed by Teodorovic and Dell'Orco [10], the BeeHive algorithm proposed by Wedde et al. [11], the Bee Swarm Optimization algorithm proposed by Drias et al. [12] and the Bees algorithm proposed by Pham et al. [13]. The Artificial Bee Colony algorithm [7,8] is, mainly, applied in continuous optimization problems and simulates the wagged dance behaviour that a swarm of bees perform during the foraging process of the bees. In this algorithm there are three groups of bees, the employed bees (bees that determines the food source (possible solutions) from a

* Corresponding author. Tel.: +30 28210 37236; fax: +30 28210 69410.

E-mail addresses: magda@dssl.tuc.gr (M. Marinaki), marinakis@ergasya.tuc.gr (Y. Marinakis), kostas@dpem.tuc.gr (C. Zopounidis).

prespecified set of food sources and share this information (waggle dance) with the other bees in the hive), the onlookers bees (bees that based on the information that they take from the employed bees they search for a better food source in the neighborhood of the memorized food sources) and the scout bees (employed bees that their food source has been abandoned and they search for a new food source randomly). The Virtual Bee algorithm [9] is, also, applied in continuous optimization problems. In this algorithm, the population of the bees are associated with a memory, a food source, and then all the memories communicate between them with a waggle dance procedure. The whole procedure is similar with a genetic algorithm and it has been applied on two function optimization problems with two parameters. In the BeeHive [11] algorithm, a protocol inspired from dance language and foraging behaviour of honey bees is used. In the Bees Swarm Optimization [12], initially a bee finds an initial solution (food source) and from this solution the other solutions are produced with certain strategies. Then, every bee is assigned in a solution and when they accomplished their search, the bees communicate between them with a waggle dance strategy and the best solution will become the new reference solution. To avoid cycling the authors use a tabu list. In the Bees algorithm [13], a population of initial solutions (food sources) are randomly generated. Then, the bees are assigned to the solutions based on their fitness function. The bees return to the hive and based on their food sources, a number of bees are assigned to the same food source in order to find a better neighborhood solution. In the Bee Colony Optimization [10] algorithm, a step by step solution is produced by each forager bee and when the foragers returns to the hive a waggle dance is performed by each forager. Then the other bees, based on a probability, follow the foragers. This algorithm looks like the Ant Colony Optimization [5] algorithm but it does not use at all the concept of pheromone trails.

Contrary to the fact that there are many algorithms that are based on the foraging behaviour of the bees, the main algorithm proposed based on the marriage behaviour is the Honey Bees Mating Optimization algorithm (HBMO), that was presented [14,15]. Since then, it has been used on a number of different applications [16–18]. The Honey Bees Mating Optimization algorithm simulates the mating process of the queen of the hive. The mating process of the queen begins when the queen flights away from the nest performing the mating flight during which the drones follow the queen and mate with her in the air. The algorithm is a swarm intelligence algorithm since it uses a swarm of bees where there are three kinds of bees, the queen, the drones and the workers. There is a number of procedures that can be applied inside the swarm. In the Honey Bees Mating Optimization algorithm, the procedure of mating of the queen with the drones is described. From this point of view someone would classify the HBMO algorithm as a memetic algorithm, since we have an elitist genetic algorithm where the queen plays the role of “super-parent”. But of course this method is not a simple memetic algorithm because in this algorithm we have a number of details that differentiate the HBMO algorithm from a simple memetic algorithm. First, the queen is flying randomly in the air and, based on her speed and her energy, if she meets a drone then there is a possibility to mate with him. Even if the queen mates with the drone, she does not create directly a brood but stores the genotype (with the term “genotype” we mean some of the basic characteristics of the drones, i.e. part of the solution) of the drone in her spermatheca and the brood is created only when the mating flight has been completed. Another difference of the proposed algorithm from a memetic algorithm is that the broods are not created by using one queen and one drone but each brood uses parts of the solutions

(genotype) of the one queen and more than one drones. In our proposed algorithm except of this classic procedure that has been used from the researchers in the previous published algorithms based on Honey Bees Mating Optimization [14–18], we use also an adaptive memory procedure in order the queen to have the possibility to store from previous selected good drones (in previous mating flights) part of their solutions in order to use them in a new mating flight and to produce more fittest drones. Another difference from a classic memetic algorithm is the role of the workers. Someone could say that since the role of the workers is simply the brood care and they are only a local search phase in the algorithm, then we have one of the basic characteristics of the memetic algorithms (a genetic algorithm with a local search phase [19]). But here we have a strict parallelism of the local search phase with what happens in real life, i.e. with the foraging behaviour of the honey bees. We mean that each one of the workers, which are different honey bees, takes care one brood in order to find for him food and feed him with the “royal jelly” in order to make him fittest and if this brood is better than the queen to take her place. And thus, this algorithm combines both the mating process of the queen and one part of the foraging behaviour of the honey bees inside the hive.

This paper presents a novel approach to solve the Feature Subset Selection Problem using Honey Bees Mating Optimization Algorithm. In the classification phase of the proposed algorithm, a number of variants of the Nearest Neighbor classification method are used [20]. The algorithm is used for the credit risk assessment classification task that is a very challenging and important management science problem in the domain of financial analysis [21]. Modern finance is a broad field often involved with hard decision-making problems related to risk management. In several cases, financial decision-making problems require the assignment of the available options into predefined groups/classes. Credit risk analysis, bankruptcy prediction, and country risk assessment, among other are some typical examples [22]. In this context the development of reliable classification models is clearly of major importance to researchers and practitioners. The development of financial classification models is a complicated process, involving careful data collection and pre-processing, model development, validation and implementation. Focusing on model development, several methods have been used, including statistical methods, artificial intelligence techniques and operations research methodologies. In all cases, the quality of the data is a fundamental point. This is mainly related to the adequacy of the sample data in terms of the number of observation and the relevance of the decision attributes (i.e., independent variables) used in the analysis. During the last years the application of nature inspired methods to financial problems have been developed [23–28]. More precisely, in [24] a procedure that utilizes a genetic algorithm in order to solve the Feature Subset Selection Problem is presented and is combined with a number of Nearest Neighbor based classifiers. The genetic based classification algorithm is applied for the solution of the credit risk assessment classification problem. In [25], a memetic algorithm, which is based on the concepts of genetic algorithms and particle swarm optimization is presented. Contrary to genetic algorithms, in this algorithm the evolution of each individual of the population is performed using a particle swarm optimization algorithm. The memetic-based classification algorithm is combined with a number of nearest neighbor based classifiers and is tested in a very significant financial classification task, involving the identification of qualified audit reports. In [26] a tabu search metaheuristic combined with a number of Nearest Neighbor classifiers is applied for the solution of the credit risk assessment problem

while in [27] an ant colony optimization algorithm combined with a number of Nearest Neighbor classifiers for the solution of the same financial classification problem is presented. All the methods presented in [24–27] gave very satisfactory results and compared to other classic metaheuristic algorithms their results were in all cases better than the results of the classic metaheuristic algorithms.

The rest of the paper is organized as follows: the next section provides a short description of the feature selection problem. In Section 3, a detailed analysis of the proposed algorithm is presented. Section 4 describes the applications context using the aforementioned financial data sets and the experimental settings, whereas Section 5 presents the obtained computational results. The last section concludes the paper and discusses some future research directions.

2. Feature selection problem

Recently, there has been an increasing need for novel data-mining methodologies that can analyze and interpret large volumes of data. The proper selection of the right set of features for classification is one of the most important problems in designing a good classifier. Feature selection is widely used as the first stage of classification task to reduce the dimension of problem, decrease noise, improve speed by the elimination of irrelevant or redundant features. 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 subset, 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 [29]. Many algorithms are, thus, proposed to find suboptimum solutions in comparably smaller amount of time [30]. The Branch and Bound approaches (BB) [31], the Sequential Forward/Backward Search (SFS/SBE) [32–34] and the filter approaches [33,35] search for suboptimum solutions. One of the most important filter approaches is the Kira and Rendell's Relief algorithm [36]. Stochastic algorithms, including Simulated Annealing (SA) [37], Scatter Search algorithms [38], Ant Colony Optimization [39–43] and Genetic algorithms (GA) [33,44] are of great interest because they often yield high accuracy and are much faster.

3. The Honey Bees Mating Optimization algorithm

In this paper, as it has already been mentioned, an algorithm for the solution to the feature selection problem based on the Honey Bees Mating Optimization is presented. This algorithm is combined with three Nearest Neighbor-based classifiers, the 1-Nearest Neighbor, the k-Nearest Neighbor and the Weighted k (wk)-Nearest Neighbor classifier. A pseudocode of the proposed Honey Bees Mating Optimization based classification algorithm is presented in the following and then an analytical description of the steps of the algorithm is given.

Initialization

Generate the initial population randomly

Selection of the best bee as the queen

Select maximum number of mating flights (M)

do while $i \leq M$

 Initialize queen's spermatheca, energy and speed.

 Select α

do while $energy > 0$ and *spermatheca is not full*

 Select a drone

if the drone passes the probabilistic condition

 Add sperm of the drone in the spermatheca

endif

$Speed(t+1) = \alpha \times Speed(t)$

$energy(t+1) = \alpha \times energy(t)$

enddo

do $j = 1, Size\ of\ Spermatheca$

 Select a sperm from the spermatheca

 Generate a brood by using a crossover operator

 between the queen's genotype and the

 selected sperm

 Select, randomly, a worker

 Use the selected worker to improve the brood's

 fitness

if the brood's fitness is better than the queen's

 fitness

Replace the queen with the brood

else

if the brood's fitness is better than

 one of the drone's fitness

Replace the drone with the brood

endif

endif

enddo

enddo

return The Queen (Best Solution Found)

3.1. Nearest Neighbor classifiers

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, a number of features are activated. For each sample of the test set, the Euclidean Distance from each sample in the training set is calculated. The Euclidean Distance is calculated as follows:

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

where D_{ij} is the distance between the test sample $i = 1, \dots, M_{\text{test}}$ (M_{test} is the number of test samples) and the training sample $j = 1, \dots, M_{\text{train}}$ (M_{train} is the number of training samples), 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 to which 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 Neighbors has an equal percentage in the vote.

However, it is natural to attach more weight to those members that are closer to the test samples. This method is called the Weighted k Nearest Neighbor (wk-nn). In this method, the most distant neighbor from the test sample is denoted by $i = 1$ while the Nearest Neighbor is denoted by $i = k$. The i_{th} neighbor receives the weight

$$w_i = \frac{i}{\sum_{i=1}^k i} \tag{2}$$

Thus, the following holds:

$$\begin{aligned} w_k \geq w_{k-1} \geq \dots \geq w_1 > 0 \\ w_k + w_{k-1} + \dots + w_1 = 1. \end{aligned} \tag{3,4}$$

3.2. Fitness function

The fitness function measures the quality of the produced members of the population. In this problem, the quality is measured with the overall classification accuracy (see below for a complete description). Thus, for each bee the classifiers (1-Nearest Neighbor, k-Nearest Neighbor or wk-Nearest Neighbor) are called and the produced overall classification accuracy (OCA) gives the fitness function. In the fitness function we would like to maximize the OCA. The accuracy of a C class problem can be described using a $C \times C$ confusion matrix. The element c_{ij} in row i and column j describes the number of samples of true class j classified as class i , i.e., all correctly classified samples are placed in the diagonal and the remaining misclassified cases in the upper and lower triangular parts. Thus, the overall classification accuracy (OCA) can be easily obtained as:

$$OCA = 100 \times \frac{\sum_{i=1}^C c_{ii}}{\sum_{i=1}^C \sum_{j=1}^C c_{ij}} \tag{5}$$

3.3. HBMO algorithm

Initially, we have to choose the population of the honey bees that will configure the initial hive. Each bee is randomly placed in the d -dimensional space as a candidate solution (in the feature selection problem d corresponds to the number of activated features). One of the key issues in designing a successful algorithm for feature selection problem is to find a suitable mapping between feature selection problem solutions and bees in Honey Bees Mating Optimization algorithm. Every candidate feature in HBMO is mapped into a binary bee where the bit 1 denotes that the corresponding feature is selected and the bit 0 denotes that the feature is not selected. Afterwards the fitness of each individual is calculated as described previously and the best member of the initial population of bees is selected as the queen of the hive while all the other members of the population are the drones.

Before the process of mating begins, the user has to define a number that corresponds to the queen's size of spermatheca. This number corresponds to the maximum number of mating of the queen in a single mating flight. Each time the queen successfully mates with a drone the genotype of the drone is stored and a

variable is increased by one until the size of spermatheca is reached. Another two parameters have to be defined, the number of queens and the number of broods that will be born by all queens. In this implementation of Honey Bees Mating Optimization (HBMO) algorithm, the number of queens is set equal to one, because in the real life only one queen will survive in a hive, and the number of broods is set equal to the number corresponding to the queen's spermatheca size. Then, we are ready to begin the mating flight of the queen. At the start of the flight, the queen is initialized with some energy content (initially, the speed and the energy of the queen are generated at random) and returns to her nest when the energy is within some threshold from zero to full spermatheca [16]. A drone mates with a queen probabilistically using the following annealing function [14,15]:

$$Prob(D) = e^{(-\Delta(f))/(Speed(t))} \tag{6}$$

where $Prob(D)$ is the probability of adding the sperm of drone D to the spermatheca of the queen (that is, the probability of a successful mating), $\Delta(f)$ is the absolute difference between the fitness of D and the fitness of the queen and $Speed(t)$ is the speed of the queen at time t . The probability of mating is high when the queen is still at the beginning of her mating flight, therefore her speed is high, or when the fitness of the drone is as good as the queen's. After each transition in space, the queen's speed and energy decays according to the following equations:

$$Speed(t + 1) = \alpha \times Speed(t) \tag{7}$$

$$energy(t + 1) = \alpha \times energy(t) \tag{8}$$

where α is a factor $\in (0, 1)$ and is the amount of speed and energy reduction after each transition and each step. A number of mating flights are realized. If the mating is successful (i.e., the drone passes the probabilistic decision rule), the drone's sperm is stored in the queen's spermatheca. By crossovering the drone's genotypes with the queen's, a new brood (trial solution) is formed which later can be improved, employing workers to conduct local search. One of the major differences of the Honey Bees Mating Optimization algorithm from the classic evolutionary algorithms is that since the queen stores a number of different drone's sperm in her spermatheca she can use parts of the genotype of different drones to create a new solution which gives the possibility to have more fittest broods. In the crossover phase, we use a crossover procedure which initially identifies the common characteristics of the queen and the drone and, then, inherits them to the broods. This crossover operator is a kind of adaptive memory procedure. Initially, the adaptive memory has been proposed by Rochat and Taillard [45] as a part of a tabu search metaheuristic for the solution of the Vehicle Routing Problem. This procedure stores characteristics of good solutions. Each time a new good solution is found the adaptive memory is updated. In our case, in the first generation the adaptive memory is empty. In order to add a solution or a part of a solution in the adaptive memory there are a number of possibilities:

- (1) The candidate for the adaptive memory solution is a previous best solution (queen) and its fitness function is at most 10% worst than the value of the current best solution.
- (2) The candidate for the adaptive memory solution is a member of the population (drone) and its fitness function is at most 10% worst than the value of the current best solution.
- (3) A sequence of activated features is common for the queen and for a number of drones.

More analytically, in this crossover operator, the points are selected randomly from the adaptive memory, from the selected drones and from the queen. Thus, initially two crossover operator

numbers are selected (Cr_1 and Cr_2) that control the fraction of the parameters that are selected for the adaptive memory, the selected drones and the queen. If there are common parts in the solutions (queen, drones and adaptive memory) then these common parts are inherited to the brood, else the Cr_1 and Cr_2 values are compared with the output of a random number generator, $rand_i(0, 1)$. If the random number is less or equal to the Cr_1 the corresponding value is inherited from the queen, if the random number is between the Cr_1 and the Cr_2 then the corresponding value is inherited, randomly, from the one of the elite solutions that are in the adaptive memory, otherwise it is selected, randomly, from the solutions of the drones that are stored in spermatheca. Thus, if the solution of the brood is denoted by $b_i(t)$ (t is the iteration number), the solution of the queen is denoted by $q_i(t)$, the solution in the adaptive memory is denoted by $ad_i(t)$ and the solution of the drone by $d_i(t)$:

$$b_i(t) = \begin{cases} q_i(t), & \text{if } rand_i(0, 1) \leq Cr_1 \\ ad_i(t), & \text{if } Cr_1 < rand_i(0, 1) \leq Cr_2 \\ d_i(t), & \text{otherwise.} \end{cases} \quad (9)$$

In each iteration the adaptive memory is updated based on the best solution. In real life, the role of the workers is restricted to brood care and for this reason the workers are not separate members of the population but they are used as local search procedures in order to improve the broods produced by the mating flight of the queen. Each of the workers have different capabilities and the choice of two different workers may produce different solutions. Each of the worker have the possibility to activate or deactivate a number of different features. Each of the brood will choose, randomly, one worker to feed it with royal jelly (local search phase) having as a result the possibility of replacing the queen if the solution of the brood is better than the solution of the current queen. If the brood fails to replace the queen, then in the next mating flight of the queen this brood will be one of the drones. The number of drones remains constant in each iteration and the population of drones consists of the fittest drones of all generations.

4. Application

The nature inspired algorithm is applied to a financial classification problem which is related to credit risk assessment. The data, taken from Doumpos and Pasiouras [46] involve 1330 firm-year observations for UK non-financial firms, over the period 1999–2001. The sample observations are classified into five risk groups according to their level of likelihood of default, measured on the basis of their QuiScore, a credit rating assigned by Qui Credit Assessment Ltd. In particular, on the basis of their QuiScore, the firms are classified into five risk groups as follows:

- Secure group, consisting of firms for which failure is very unusual and normally occurs only as a result of exceptional market changes.
- Stable group, consisting of firms for which failure is rare and will only come about if there are major company or market changes.
- Normal group, consisting of firms that do not fail, as well as some that fail.
- Unstable (caution) group, consisting of firms that have a significant risk of failure.
- High-risk group, consisting of firms which are unlikely to be able to continue their operation unless significant remedial action is undertaken.

Such multi-group credit risk rating systems are widely used in practice, mainly by banking institutions. According to Treacy and Carey [47] the internal reporting process in banking institutions is

Table 1
List of financial ratios.

Feature number	Financial ratios
1	Current ratio
2	Quick ratio
3	Shareholders liquidity ratio
4	Solvency ratio
5	Asset cover
6	Annual change in fixed assets
7	Annual change in current assets
8	Annual change in stock
9	Annual change in debtors
10	Annual change in total assets
11	Annual change in current liabilities
12	Annual change creditors
13	Annual change in loans/overdraft
14	Annual change in long-term liabilities
15	Net profit margin
16	Return on total assets
17	Interest cover
18	Stock turnover
19	Debtors turnover
20	Debtors collection (days)
21	Creditors payment (days)
22	Fixed assets turnover
23	Salaries/turnover
24	Gross profit margin
25	Earning Before Interest and Taxes (EBIT) margin
26	Earning Before Interest and Taxes, Depreciation and Amortization (EBITDA) margin

facilitated by credit risk management systems that are based on a multi-group scheme for rating risk. Internal reports prepared in this context are more informative to senior management and they provide a better representation of the actual risk exposure for the banking institution.

The evaluation criteria used in the analysis involve ratios based on the financial statements of the firms. It should be noted that credit scoring has non-financial aspects as well but obtaining objective, comprehensive, and reliable non-financial information is a cumbersome process, as such data are rarely available [46]. Therefore, within the scope of this current study the analysis was based only on financial information. A total of 26 financial ratios (Table 1) and annual changes (features) were initially considered based on data availability and previous studies on credit assessment and bankruptcy prediction.

5. Computational results

The algorithms were implemented in Fortran 90 (Lahey f95 compiler) on a Centrino Mobile Intel Pentium M750/1.86 GHz, running Suse Linux 9.1. To test the efficiency of the proposed method, the *10-fold cross-validation procedure* is utilized. Initially, the data set is divided into 10 disjoint groups containing approximately $M/10$ samples each, where M is the number of the samples in the data set. Next, each of these groups is systematically removed from the data set, a model is built from the remaining parts (the training set) and, then, the accuracy of the model is calculated using the excluding parts (the test set).

As has already been mentioned, three approaches that use different classifiers, the 1-nn, k-nn and the wk-nn, are used. In HBMO based classifier the value of k is changed dynamically depending on the number of iterations. Each generation uses different k . The reason why k does not have a constant value is that we would like to ensure the diversity of the bees in each iteration of HBMO, respectively. The determination of k is done by using a random number generator with a uniform distribution (0, 1) in each iteration. Then, the produced number is converted to an

integer k (e.g., if the produced number is in the interval 0.2–0.3, then $k = 3$).

The parameters of the proposed algorithm were selected after thorough testing. A number of different alternative values were tested and the ones selected are those that gave the best computational results concerning both the quality of the solution and the computational time needed to achieve this solution. Thus, the selected parameters are given in the following:

- (1) Number of queens equal to 1.
- (2) Number of drones equal to 200.
- (3) Number of mating flights (M) equal to 50.
- (4) Size of queen's spermatheca equal to 50.
- (5) Number of broods equal to 50.
- (6) $\alpha = 0.9$.
- (7) Number of different workers ($w = 7$).

For comparison purposes, four other metaheuristic algorithms were used, a tabu based metaheuristic, a genetic based metaheuristic, an ant colony optimization and a particle swarm optimization based algorithm. For more details of how these methods are applied for the solution of this problem please see [24–27]. These methods use in the classification phase the 1-nn, the k-nn and the wk-nn classifiers. In the genetic, aco and pso based metaheuristics with the k-nn and wk-nn classifier the value of k is changed dynamically depending on the number of iteration. Each generation uses different k . The maximum number of iterations for the tabu based metaheuristic is equal to 1000 and the size of the tabu list is equal to 10. The parameter settings for the genetic based metaheuristic are:

- (1) Population size equal to 200.
- (2) Number of generations equal to 50.
- (3) Probability of crossover equal to 0.8.
- (4) Probability of mutation equal to 0.25.

The parameter settings for the ant colony based metaheuristic are:

- (1) The number of ants used is set equal to the number of features (26 for the credit risk assessment).
- (2) The number of iterations that each ant constructs a different solution, based on the pheromone trails, is set equal to 50.
- (3) $q = 0.5$.

The parameter settings for the PSO based metaheuristic are:

- (1) The number of swarms is set equal to 1.
- (2) The number of particles is set equal to 50.
- (3) The number of generations is set equal to 50.
- (4) The coefficients are $c_1 = 2, c_2 = 2$.
- (5) $w_{max} = 0.9$ and $w_{min} = 0.01$.

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 analyzed in this paper, there are $2^{26} - 1$ 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 high accuracy.

Table 2
Classification results.

Methods	Overall classification accuracy (%)	Average no. of features
HBMO 1-nn	75.21	10.2
HBMO k-nn	74.18	10.3
HBMO wk-nn	73.52	10.6
ACO 1-nn	72.18	11.2
ACO k-nn	72.85	12
ACO wk-nn	71.72	11.8
PSO 1-nn	72.10	11.3
PSO k-nn	73.30	10.4
PSO wk-nn	71.05	11.7
Gen 1-nn	68.12	12.4
Gen k-nn	69.84	13
Gen wk-nn	69.02	11.2
Tabu 1-nn	67.74	12.2
Tabu 5-nn	66.09	12
Tabu w5-nn	67.36	12.4
1-nn	55.78	26

Table 3
Selection frequencies of the features in the optimal feature sets.

Features	Frequency	Relative frequency	Features	Frequency	Relative frequency
1	17	(56.66)	14	8	(26.67)
2	19	(63.33)	15	18	(60)
3	14	(46.66)	16	13	(43.33)
4	27	(90)	17	10	(33.33)
5	15	(50)	18	14	(46.67)
6	11	(36.67)	19	10	(33.33)
7	11	(36.67)	20	5	(16.67)
8	14	(46.67)	21	8	(26.67)
9	5	(16.67)	22	13	(43.33)
10	11	(36.67)	23	7	(23.33)
11	6	(20)	24	8	(26.67)
12	2	(6.67)	25	18	(60)
13	8	(26.67)	26	19	(63.33)

Table 2 presents the classification results for the optimal solution of the proposed algorithm, the HBMO based metaheuristic, for the financial classification problem. The results of the algorithms used for the comparisons are also shown. In the credit risk assessment problem, which involves 5 classes, the nature inspired metaheuristic provides considerably better results compared to the other methods in terms of OCA. HBMO 1-nn provides the best results on the evaluation criterion followed by HBMO k-nn. In terms of the number of features used in the final model, HBMO 1-nn also provides the best results with an average (over the 10 folds of the cross-validation analysis) of 10.2 features. The significance of the proposed algorithm is demonstrated by the fact that when the classification problem is solved using 1-nn classifier and without solving the feature selection problem the overall classification accuracy is only 55.78%. The selection frequencies of the features (Table 3) indicate that the solvency ratio, EBITDA margin, EBIT margin, net profit margin and the quick ratio are the most important variables. It should be noted that the combination of all the features of the proposed Honey Bees Mating Optimization algorithm gave the ability in the proposed algorithm to perform better than the other algorithms used in the comparisons.

6. Conclusions and future research

An important issue in building a good classifier is the selection of a set of appropriate input feature variables. The Honey Bees Mating Optimization algorithm has been proposed in this study for solving this Feature Subset Selection Problem. Three different classifiers

were used for the classification problem, based on the nearest neighbor classification rule. The performance of the proposed algorithm was tested using financial data involving credit risk assessment. The obtained results indicate the high performance of the proposed algorithm in searching for a reduced set of features with high accuracy. HBMO 1-*nn* was found to provide the best results in terms of accuracy rates using less than the half of the available features. Future research will focus on the use of different machine learning classifiers (SVM, neural networks, etc.).

References

- [1] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of 1995 IEEE International Conference on Neural Networks* 4 (1995) 1942–1948.
- [2] D. Dasgupta (Ed.), *Artificial Immune Systems and their Application*, Springer, Heidelberg, 1998.
- [3] L.N. De Castro, J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*, Springer, Heidelberg, 2002.
- [4] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation* 1 (1) (1997) 53–66.
- [5] M. Dorigo, T. Stutzle, *Ant Colony Optimization*, A Bradford Book, The MIT Press Cambridge, Massachusetts, London, England, 2004.
- [6] A. Baykasoglu, L. Ozbakir, P. Tapkan, Artificial bee colony algorithm and its application to generalized assignment problem, in: F.T.S. Chan, M.K. Tiwari (Eds.), *Swarm Intelligence, Focus on Ant and Particle Swarm Optimization*, I-Tech Education and Publishing, 2007, pp. 113–144.
- [7] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization* 39 (2007) 459–471.
- [8] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing* 8 (2008) 687–697.
- [9] X.S. Yang, Engineering optimizations via nature-inspired virtual bee algorithms, in: J.M. Yang, J.R. Alvarez (Eds.), *IWINAC 2005, LNCS 3562*, Springer-Verlag, Berlin/Heidelberg, 2005, pp. 317–323.
- [10] D. Teodorovic, M. Dell'Orco, Bee colony optimization—a cooperative learning approach to complex transportation problems, *Advanced OR and AI Methods in Transportation* (2005) 51–60.
- [11] H.F. Wedde, M. Farooq, Y. Zhang, BeeHive: an efficient fault-tolerant routing algorithm inspired by honey bee behavior, in: M. Dorigo (Ed.), *Ant Colony Optimization and Swarm Intelligence LNCS 3172*, Springer, Berlin, 2004, pp. 83–94.
- [12] H. Drias, S. Sadeg, S. Yahi, Cooperative bees swarm for solving the maximum weighted satisfiability problem, *IWAAN International Work Conference on Artificial and Natural Neural Networks LNCS 3512* (2005) 318–325.
- [13] D.T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, M. Zaidi, The bees algorithm—a novel tool for complex optimisation problems, in: *IPROMS 2006 Proceeding 2nd International Virtual Conference on Intelligent Production Machines and Systems*, Elsevier, Oxford, 2006.
- [14] H.A. Abbass, A monogenous MBO approach to satisfiability, in: *Proceeding of the International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA'2001*, Las Vegas, NV, USA, 2001.
- [15] H.A. Abbass, Marriage in honey-bee optimization (MBO): a haplometrosis polygynous swarming approach, in: *The Congress on Evolutionary Computation (CEC2001)*, Seoul, Korea, May, (2001), pp. 207–214.
- [16] A. Afshar, O.B. Haddad, M.A. Marino, B.J. Adams, Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation, *Journal of the Franklin Institute* 344 (2007) 452–462.
- [17] M. Fathian, B. Amiri, A. Maroosi, Application of honey bee mating optimization algorithm on clustering, *Applied Mathematics and Computation* 190 (2007) 1502–1513.
- [18] O.B. Haddad, A. Afshar, M.A. Marino, Honey-bees mating optimization (HBMO) algorithm: a new heuristic approach for water resources optimization, *Water Resources Management* 20 (2006) 661–680.
- [19] P. Moscato, C. Cotta, A gentle introduction to memetic algorithms, in: F. Glover, G.A. Kochenberger (Eds.), *Handbooks of Metaheuristics*, Kluwer Academic Publishers, Dordrecht, 2003, pp. 105–144.
- [20] R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification and Scene Analysis*, second edition, John Wiley and Sons, New York, 2001.
- [21] M. Doumplos, K. Kosmidou, G. Baourakis, C. Zopounidis, Credit risk assessment using a multicriteria hierarchical discrimination approach: A comparative analysis, *European Journal of Operational Research* 138 (2002) 392–412.
- [22] M. Doumplos, C. Zopounidis, P.M. Pardalos, Multicriteria sorting methodology: Application to financial decision problems, *Parallel Algorithms and Applications* 15 (1–2) (2000) 113–129.
- [23] A. Brabazon, M. O'Neill, *Biologically Inspired Algorithms for Financial Modelling*, Springer-Verlag, Berlin, 2006.
- [24] M. Marinaki, Y. Marinakis, C. Zopounidis, Application of a genetic algorithm for the credit risk assessment problem, *Foundations of Computing and Decision Sciences* 32 (2) (2007) 139–152.
- [25] M. Marinaki, Y. Marinakis, C. Zopounidis, A memetic algorithm for the identification of firms receiving qualified audit reports, *Journal of Computational Optimization in Economics and Finance* 1 (1) (2008) 59–71.
- [26] Y. Marinakis, M. Marinaki, M. Doumplos, N. Matsatsinis, C. Zopounidis, Optimization of nearest neighbor classifiers via metaheuristic algorithms for credit risk assessment, *Journal of Global Optimization* 42 (2008) 279–293.
- [27] Y. Marinakis, M. Marinaki, C. Zopounidis, Application of ant colony optimization to credit risk assessment, *New Mathematics and Natural Computation* 4 (1) (2008) 107–122.
- [28] L. Yu, S. Wang, K.K. Lai, L. Zhou, *Bio-Inspired Credit Risk Analysis Computational Intelligence with Support Vector Machines*, Springer-Verlag, Berlin, 2008.
- [29] R. Kohavi, G. John, Wrappers for feature subset selection, *Artificial Intelligence* 97 (1997) 273–324.
- [30] A. Jain, D. Zongker, Feature selection: evaluation, application, and small sample performance, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1997) 153–158.
- [31] P.M. Narendra, K. Fukunaga, A branch and bound algorithm for feature subset selection, *IEEE Transactions on Computers* 26 (9) (1977) 917–922.
- [32] D.W. Aha, R.L. Bankert, A comparative evaluation of sequential feature selection algorithms, in: D. Fisher, J.-H. Lenx (Eds.), *Artificial Intelligence and Statistics*, Springer-Verlag, New York, 1996.
- [33] E. Cantu-Paz, S. Newsam, C. Kamath, Feature selection in scientific application, in: *Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004, pp. 788–793.
- [34] P. Pudil, J. Novovicova, J. Kittler, Floating search methods in feature selection, *Pattern Recognition Letters* 15 (1994) 1119–1125.
- [35] E. Cantu-Paz, Feature subset selection, class separability, and genetic algorithms, in: *Genetic and Evolutionary Computation Conference*, 2004, 959–970.
- [36] K. Kira, L. Rendell, A practical approach to feature selection, in: *Proceedings of the Ninth International Conference on Machine Learning*, Aberdeen, Scotland, (1992), pp. 249–256.
- [37] W. Siedlecki, J. Sklansky, On automatic feature selection, *International Journal of Pattern Recognition and Artificial Intelligence* 2 (2) (1988) 197–220.
- [38] F.G. Lopez, M.G. Torres, B.M. Batista, J.A.M. Perez, J.M. Moreno-Vega, Solving feature subset selection problem by a parallel scatter search, *European Journal of Operational Research* 169 (2006) 477–489.
- [39] A. Al-Ani, Feature subset selection using ant colony optimization, *International Journal of Computational Intelligence* 2 (1) (2005) 53–58.
- [40] A. Al-Ani, Ant colony optimization for feature subset selection, *Transactions on Engineering Computing and Technology* 4 (2005) 35–38.
- [41] R.S. Parpinelli, H.S. Lopes, A.A. Freitas, An ant colony algorithm for classification rule discovery, in: H. Abbas, R. Sarker, C. Newton (Eds.), *Data Mining: A Heuristic Approach*, Idea Group Publishing, London, UK, 2002, pp. 191–208.
- [42] P.S. Shelokar, V.K. Jayaraman, B.D. Kulkarni, An ant colony classifier system: application to some process engineering problems, *Computers and Chemical Engineering* 28 (2004) 1577–1584.
- [43] C. Zhang, H. Hu, Ant colony optimization combining with mutual information for feature selection in support vector machines, in: S. Zhang, R. Jarvis (Eds.), *AI 2005. LNAI 3809*, 2005, 918–921.
- [44] W. Siedlecki, J. Sklansky, A note on genetic algorithms for large-scale feature selection, *Pattern Recognition Letters* 10 (1989) 335–347.
- [45] Y. Rochat, E.D. Taillard, Probabilistic diversification and intensification in local search for vehicle routing, *Journal of Heuristics* 1 (1995) 147–167.
- [46] M. Doumplos, F. Pasiouras, Developing and testing models for replicating credit ratings: a multicriteria approach, *Computational Economics* 25 (2005) 327–341.
- [47] W.F. Treacy, M. Cavey, Credit risk rating systems at large US banks, *Journal of Banking and Finance* 24 (2000) 167–201.