A Novel Approach for Feature Selection based on the Bee Colony Optimization

Rana Forsati
Faculty of Electrical and Computer Engineering,
ShahidBeheshti University,
G. C.,
Tehran, Iran

Alireza Moayedikia
Faculty of Electrical and Computer Engineering,
ShahidBeheshti University,
G. C.,
Tehran, Iran

Andisheh Keikha
Faculty of Electrical and Computer Engineering,
ShahidBeheshti University,
G. C.,
Tehran, Iran

ABSTRACT
One of the successful methods in classification problems is feature selection. Feature selection algorithms; try to classify an instance with lower dimension, instead of huge number of required features, with higher and acceptable accuracy. In fact an instance may contain useless features which might result to misclassification. An appropriate feature selection methods tries to increase the effect of significant features while ignores insignificant subset of features. In this work feature selection is formulated as an optimization problem and a novel feature selection procedure in order to achieve to a better classification results is proposed. Experiments over a standard benchmark demonstrate that applying Bee Colony Optimization in the context of feature selection is a feasible approach and improves the classification results.

Keywords
Feature Selection, optimization, bee colony optimization

1. INTRODUCTION
Feature selection, is the technique of selecting a subset of relevant features for building robust learning models. By removing most irrelevant features from the data that these features not only makes learning harder, but also degrades generalization performance of learned models. For a given classification task, the problem of FS can be described as follows: given the original set, G, of N features, find a subset F consisting of N´ relevant features where F G and N´ N. The aim of selecting F is to maximize the classification accuracy in building learning models. It is important to select significant in the sense that the generalization performance of learning models is heavily dependent on the selected features [1,2,3,4]. So many different procedures for feature selection are proposed that classical approaches are among the most common methods. In Exhaustive Search algorithms which are one of the conventional methods of feature subset selection, all the possible subsets are evaluated and the best one among them is chosen. This guarantees the optimal solution, but the computational time is intractable when the problem size is not small [1].

Branch and Bound [5], [6] is other classical approach that uses a search tree that identifies the features being removed from the original set. It achieves a substantial reduction in the number of subset evaluations by pruning those sub trees that will never be superior to the current best solution. However, the main problem with this algorithm is its exponential time complexity. Additionally, this algorithm requires the strict assumption of monotonicity, i.e., adding new features never degrades the performance. Some other traditional feature extraction algorithms do a linear transformation of the original feature vectors [8].

Evolutionary algorithms, which are stochastic methods based on a search model, define a global function and try to optimize its value by traversing the search space. A common factor shared by the evolutionary algorithms is that they combine rules and randomness to imitate some natural phenomena [9]. Therefore, evolutionary methods can be used to perform the selection of the features which optimizes this measure of probability. These algorithms do not guarantee the correct answer, but they always generate a close estimation of it in a reasonable amount of time.

Evolutionary algorithms such as Tabu Search [7], Harmony Search [10] and Genetic Algorithms (GA) [13,14] are general high-level procedures that coordinate simple heuristics and rules to find good approximate solutions for computationally difficult combinatorial optimization problems. These methods have been previously employed to solve the problem of feature selection and results showed that these methods are suitable for achieving comparable accuracies [25-28].

[13, 14] Bjorvand takes Wroblewski's work as a foundation, but makes several variations and practical improvements both in speed and the quality of approximation and applies genetic algorithms to compute approximate reducts. To avoid wasting much processing power in a wrong search direction, he adopted a dynamic mutation rate that is proportional to the redundancy in the population, preventing all individuals from becoming equal. Also some feature selection algorithms based on swarm optimizations like PSO (Particle Swarm Optimization) [15, 16], and ant colony [17, 18] were proposed in recent years.

[15] Proposes a Binary particle swarm optimization (BPSO) which has been applied successfully to solving feature selection problems. He used two kinds of chaotic maps so-called logistic maps and tent maps are embedded in BPSO. In his paper his purpose of chaotic maps utilization is to determine the inertia weight of the BPSO. So he called his method as chaotic binary particle swarm optimization (CBPSO) to implement the feature selection, in which the K-nearest neighbor (K-NN) method with leave-one-out cross-

validation (LOOCV) serves as a classifier for evaluating classification accuracies.

Also another PSO-based feature selection algorithm is [19]. Which the author investigated the feature subset selection problem for the binary classification problem using logistic regression model. His approach embodies an adaptive feature selection procedure which dynamically accounts for the relevance and dependence of the features included the feature subset.

In this paper we propose Bee Colony Optimization for solving feature selection problems. BCO will be discussed in details in next sections, but as a brief description The Bee Colony Optimization (BCO) meta-heuristic uses swarm intelligence techniques. This meta-heuristic approach is nature-inspired which is to be applied for finding solutions of difficult combinatorial optimization problems. Rest of the paper is organized as follows: section 3 discusses BCO, section 4 explains the proposed algorithm, section 5 experimental result explanations and section 6 is related to conclusion and future works.

2. The BEE COLONY OPTIMIZATION

BCO has been proposed by Lucˇic´ and Teodorovic´ [20, 21]. The basic idea is to create a colony of artificial bees capable of successfully solving difficult combinatorial optimization problems. The algorithm simulates the intelligent behavior of bee swarms. An artificial bee colony behaves to some extent like and to some extent in a different way from, bee colonies found in the natural world. It is a very simple, robust and population based stochastic optimization algorithm. The BCO is model the collection and processing of nectar, the practice of which is highly organized. Each bee decides to reach the nectar source by following a nestmate who has already discovered a patch of flowers. Each hive has a so-called dance floor area on which the bees that have discovered nectar sources dance, in that way trying to convince their nestmates to follow them. If a bee decides to leave the hive to get nectar, she follows one of the bee dancers to one of the nectar areas. Upon arrival, the foraging bee takes a load of nectar and returns to the hive relinquishing the nectar to a food-storer bee. After she relinquishes the food, the bee can (a) abandon the food source and become again an uncommitted follower; (b) continue to forage at the food source without recruiting nestmates; or (c) dance and thus recruit nestmates before returning to the food source. The bee opts for one of the above alternatives with a certain probability. Within the dance area, the bee dancers 'advertise' different food sources.

The BCO is a population-based algorithm. A population of artificial bees searches for the optimal solution with every artificial bee generating one solution to the problem. The algorithm consists of two alternating phases: a forward pass and a backward pass. During each forward pass, every bee is exploring the search space and creating various partial solutions. It applies a predefined number of moves (visit certain number of nodes), which construct and/or improve the solution, yielding a new solution. During the second forward pass, bees will visit few more nodes, expand previously created partial solutions. Having obtained new partial solutions, the bees return to the nest and start the second phase, the so-called backward pass. During the backward pass, all bees share information about their solutions. In the nest, all bees participate in a decision-making process. In the nest bees exchange information about quality of the partial solutions created. Bees compare all generated partial solutions. During the backward pass, Based on the quality of the partial solutions generated, every bee decides with a certain probability whether it will advertise its solution or not. The bees with better solutions have more chances to advertise their solutions. The remaining bees have to decide whether to continue to explore their own solution in the next forward pass, or to start exploring the neighborhood of one of the solutions being advertised. Similarly, this decision is taken with a probability, so that better solutions have a higher probability of being chosen for exploration. Depending on the quality of the partial solutions generated, every bee possesses certain level of loyalty to the path leading to the previously discovered partial solution. The search process is composed of iterations. The first iteration is finished when bees create for the first time one or more feasible solutions by visiting all node. The two phases of the search algorithm, the forward and backward pass, are performed iteratively, until a stopping condition is satisfied. The possible stopping conditions could be, for example, the maximum total number of forward/backward passes, the maximum total number of forward/backward passes without the improvement of the objective function, etc.

The best discovered solution during the first iteration is saved, and then the second iteration begins. Within the second iteration, bees again incrementally construct solutions of the problem, etc. There are one or more partial solutions at the end of each iteration. The analyst-decision maker prescribes the total number of iterations.

3. BCFSELECT: BEECOLONYFAETURE SELECTION

In this paper a new approach to solve feature selection problem is proposed, in which the natural behaviour of the bees are simulated and modelled to solve the problem. As a brief explanation, each bee randomly selects 0 or 1 at the begin, and moves forward for d forward steps, during each forward step the bee must decide whether choose a feature or not, after d forward step is passed the backward step is started in which bees turn back to their hive and evaluate their solutions. At this point bees are divided into 2 groups of committed and uncommitted, those bees that their fitness is above a pre-specified amount are recognized as committed and the others are uncommitted. This concept is known as loyalty. To determine which bee is loyal to its solution the highest and the lowest fitness values are chosen and then their average is calculated which is called A and a number in the interval [A – 1) is randomly generated as r as the pre-specified loyalty degree, and those ants that their loyalty level to their solution is above r are considered as committed and the others that are uncommitted must follow the committed ones. At this point each uncommitted bee should choose a committed bee to follow for d steps and after d steps are taken by so called followers they are free to choose their further movements by their own. But the followers should decide which bee to follow according to roulette wheel, which is a recruiting probability. In below solution evaluation, loyalty decision and recruiting probability are discussed in greater details according to the algorithm implementations.

3.1. Solution Representation

The first question to solve feature problem by BCO is how to represent solutions. Solutions are represented in the following form:

F_1	F_2	……	F_n
1	0	……...	1

Fiis the i^{th} component of the generated solution by any bee where i, is between 1 and n, and 1 indicates the feature is selected while 0 indicates an unselected feature. n is the length of features. In bee colony approach bees must go further for s steps and then they should turn back to the hive for general fitness evaluation. The forward step is determined by the empirical studies that are described in the experimental result section for each dataset separately.

3.2. Loyalty Decision

Depending on the quality of the generated solutions, every bee possesses certain level of loyalty to their path leading to the previously discovered solution. Artificial bees that are loyal to their partial solutions, are more likely that their solutions to be advertised. The bees decide whether they stay loyal to their decision or not by the following equation:

$$P_b^{u+1} = e^{-\frac{O_{max} - O_b}{u}}, \quad b = 1,2,3,\dots B. \qquad (1)$$

Where Omax is the maximum normalized fitness of the generated solution of the bee B, Ob is the normalized value for the objective function of partial solution created by b-th bee and u is the ordinary number of forward pass (e.g., u = 1 for the first forward pass). It is worthwhile to mention that, the higher the value of u, the high the effect of the already discovered solution. Using a random number generator and equation (1), each bee decides whether remain loyal to its partial solution or become an uncommitted bee.

3.3. Recruiting Probability

Once the solution is abandoned by a bee, the bee becomes uncommitted and has to select one of the advertised solutions. This decision is taken with a probability known as recruiting probability, so that better advertised solutions have bigger opportunity to be chosen for further exploration.

$$P_b = \frac{O_b}{\sum_{k=1}^{R} O_k}, \quad b = 1,2,3,\dots,R \qquad (2)$$

Where Ok represents the normalized value of the objective function of the k-th advertised general solution and R is the number of recruiters, that each uncommitted bee choose one committed bee according to roulette wheel.

3.4. Evaluation And Stopping Condition

The quality of the solutions, produced by BCF Select, relies on the stochastic nature of the technique and the way in which the objective function is converted to a fitness function that can guide the algorithm to the desired region of the search space. As a result, designing a good fitness function is a key problem in solving problems with the BCFS method. The evaluation is straightforward since a solution represents a selected feature subset, X, and the evaluation function is clear. The fitness of a solution S is defined as:

$$\text{Fitness}(S) = (\text{correctly classified samples} / \text{total samples}) \times 100\%. \qquad (3)$$

The two phases of the search algorithm, forward and backward pass, are alternating in order to generate all required feasible solutions (one for each bee). The first iteration is finished when bees for the first time create one or more feasible solutions by visiting all the nodes. When all solutions are completed the best discovered solution during the first iteration is determined. It is used to update global best solution and an iteration of the BCO is accomplished. The two phases of the BCO are carried out iteration by iteration, until a stopping condition is satisfied. The possible stopping conditions could be, for example, the maximum total number of forward/backward passes, the maximum total number of forward/backward passes without the improvement of the objective functions, etc. At the end of each iteration, the best found solution (the so called global best) is reported as the final one. Also this point should be noted that during forward paths, when the fitness of each bee is calculated, its value is saved if it is higher than any other previous forward paths, and at the end of the iteration is represented as the highest and best generated solution. For instance in figure 1, fitness of the b-th bee is chosen as the final fitness value of the b-th bee, since it is the highest one among other three values in each constructive movement.

4. EXPERIMENTAL RESULTS

In this section we present the experimental evidences and results that was made on several standard datasets, and the comparisons that were made with other relevant works, done by other authors. Section 4-1 discusses the nature of the chosen datasets and its next section is related to the results of the experiments.

4.1 Dataset Description

The data sets in this study were obtained from the UCI Repository [23]. Table1 illustrates the format of the six classification problems. If the number of features is between 4 and 19, the sample groups can be considered small; these datasets include the Iris, Heart, Breast, Glass, Vowel and Vehicle data sets. If the number of features is between 20 and 49, the sample test groups are medium scale problems; these include the Ionosphere problems. If the number of features is greater than 50, the test problems are large scale problems. Also in this algorithm 1-NN classifier along with LOOCV is used in order to assess the accuracy of the generated solutions.

4.2. Comparisons and Discussions

In the previous subsection the structure of datasets were explained. Now it this section we compare our algorithm with other related works. Table 2 shows that Bee Colony can perform better than other algorithms like MLP-based FS method (MLPFS) [1], artificial neural net input gain measurement approximation (ANNIGMA) [12], and hybrid genetic algorithm for FS (HGAFS) [13].

The table 3 illustrates comparisons of Bee Colony with HS-based algorithms, and other methods like GA [24] and PSO [11].

Table 1.a brief overview of the datasets

Dataset	Size	Number of Classes	Number of features	Number of Bees	NC	Iteration	Classification Method
Vowel	990	11	10	20	2	80	1-NN
Vehicle	846	4	18	20	3	80	1-NN
Ionosphere	351	2	34	20	6	80	1-NN
Breast	699	2	9	20	3	80	1-NN
Iris	150	3	4	20	1	80	1-NN
Heart	270	2	14	20	2	80	1-NN

Table 2.comparisons among Bee colony and ANNIGMA [12], HGAFS [13], MLPFS [1]

Dataset		ANNIGMA	HGAFS	MLPFS	Bee Colony
Ionosphere	No. Features	9.00	6.00	32.00	15
	Accuracy (%)	90.20	92.76	90.60	93.16

Table 3.a brief comparison between Bee Colony and GA [24], PSO [24] and [22]

Dataset	Unreduced	HHS	VHS	GA	PSO	BeeColony
Heart	76.67	79.26	79.26	79.26	70.37	80.74(6)
Ionosphere	87.83	89.57	86.09	82.61	86.96	93.16(15)
Iris	96	96	96	96	96	96(4)

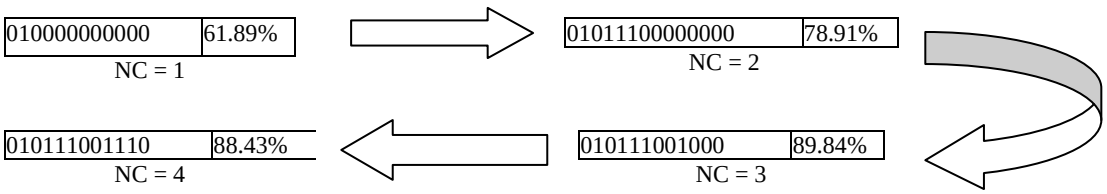

Figure1: Choosing the best fitness of the b-th bee

5. CONCLUSION

According to the experimental results and analyses, we drew a number of conclusions and comparisons. It should be noted that the experimental results and analyses from which we draw our conclusions were based on various standard data sets covering a large spectrum of problem sizes.

1. SFFS is the best sequential search algorithm, and Bee Colony was successful to outperform it, in most cases, but not all situations.

2. Rough set reduction algorithms were among the most powerful procedures for feature selection problems, but could not do better than bee colony approach.

Finally, the proposed algorithm is worthwhile to be considered as one of the best methods for feature selection problems, but any other change can be done to improve its performance.

6. ACKNOWLEDGEMENT

This work has been supported by the Grantnumber 600/1817 from the vice presidency of research and technology of ShahidBeheshti University, G. C.

7. REFERENCES

[1] E. Gasca, J.S. Sanchez, R. Alonso, Eliminating redundancy and irrelevance using a new MLP-based feature selection method, Pattern Recognition 39 (2006) 313–315.

[2] P. Pudil, J. Novovicova, J. Kittler, Floating search methods in feature selection, Pattern Recognition Letters 15 (11) (1994) 1119–1125.

[3] R. Setiono, H. Liu, Neural network feature selector, IEEE Transactions on Neural Networks vol. 8 (1997).

[4] A. Verikas, M. Bacauskiene, Feature selection with neural networks, Pattern Recognition Letters 23 (2002) 1323–1335.

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

[6] H. Liu, Lei Tu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering 17 (4) (2005) 491–502.

[7] M. Das, H. Liu, Feature selection for clustering, Proceedings of Pacific-asia Conference on Knowledge Discovery and Data Mining (2000) 110–121.

[8] M.A. Hall, Correlation-based feature selection for discrete and numeric class machine learning, in: Proceedings of the 17th International Conference on Machine Learning, 2000.

[9] Y. Lei, H. Liu, Feature selection for high-dimensional data: a fast correlation- based filter solution, in: Proceedings of the 20th International Conference on Machine Learning (ICML), 2003.

[10] K. Michalak, H. Kwasnicka, Correlation-based feature selection strategy in neural classification, in: Proceedings of the 6th International Conference on Intelligent Systems Design and Applications (ISDA), 2006.

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

[12] C. Hsu, H. Huang, D. Schuschel, The ANNIGMA-wrapper approach to fast feature selection for neural nets, IEEE Transactions on Systems Man, and Cybernetics—Part B:Cybernetics32(2)(2002)207–212.

[13] J. Huang, Y. Cai, X. Xu, A hybrid genetic algorithm for feature selection wrapper based on mutual information, Pattern Recognition Letters 28 (2007) 1825–1844.

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

[15]H. Zhang, G. Sun, Feature selection using Tabu search method, Pattern Recognition 35 (2002) 701-711.

[16]D.A. Bell, H. Wang, A formalism for relevance and its application in feature subset selection, Machine Learning 41 (2004) 175–195.

[17] E. Parzen, ARARMA models for time series analysis and forecasting, Journal of Forecasting 1 (1982) 67–87.

[18] A.A. Albrecht, Stochastic local search for the feature set problem, with applications to microarray data, Applied Mathematics and Computation 183 (2006) 1148–1164.

[19] S.F. Cotter, K. Kreutz-Delgado, B.D. Rao, Backward sequential elimination for sparse vector selection, Signal Processing 81 (2001) 1849–1864.

[20]P. Lucic, D. Teodorovic, Bee System: Modeling combinatorial optimization transportation engineering problems by swarm intelligence. In preprints of the TRISTAN IV Triennial symposium on Transportation Analysis. Sao Miguel, Azores Island, (2001) 441-445.

[21]P. Lucic, D. Teodorovic, Transportation modeling: an artificial approach. In proceedings of the 14th IEEE International Conference on Tools with Artificial intelligence. Washington DC, (2002) 216-223.

[22] R. Diao, Q. Shen, Two New Approaches to Feature Selection with Harmony Search, WCCI 2010 IEEE World Congress on Computational Intelligence, 2010, Spain.

[23] P. Murphy, D. Aha, UCI repository of machine learning databases, 1995, URL http://www.sgi.com/Technology/mlc/db.

[24] J. Wroblewski, Finding minimal reducts using genetic algorithm, Proceedings of the second annual join conference on information science, (1995) 186–189.

[25] R. Forsati, A. Moayedikia, B. Safarkhani, Heuristic approach to solve feature selection problem, DICTAP, 2011, pp. 707-717.

[26] R. Forsati, M. Shamsfard, P. Mojtahedpour, An efficient meta heuristic algorithm for pos-tagging, Fifth International Multi-Conference on Computing in the Global Information Technology (ICCGI, pp.93-98, 20-25, 2010.

[27]R. Forsati, M. Mahdavi, M. Kangavari, B.Safarkhani, Web page clustering using Harmony Search optimization, Canadian Conference on Electrical and Computer Engineering, CCECE 2008, , pp.001601-001604, 2008.

[28] R. Forsati, M. Shamsfard, Cooperation of evolutionary and statistical PoS-tagging, Proceedings of the 2012 CSI International Symposium on Artificial Intelligence and Signal Processing (AISP), Shirza University , Iran, 2012.