International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:4, No:4, 2010

Correlation-based Feature Selection using Ant Colony Optimization

M. Sadeghzadeh, M. Teshnehlab

***Abstract*—**Feature selection has recently been the subject of intensive research in data mining, specially for datasets with a large number of attributes. Recent work has shown that feature selection can have a positive effect on the performance of machine learning algorithms. The success of many learning algorithms in their attempts to construct models of data, hinges on the reliable identification of a small set of highly predictive attributes. The inclusion of irrelevant, redundant and noisy attributes in the model building process phase can result in poor predictive performance and increased computation. In this paper, a novel feature search procedure that utilizes the Ant Colony Optimization (ACO) is presented. The ACO is a meta-heuristic inspired by the behavior of real ants in their search for the shortest paths to food sources. It looks for optimal solutions by considering both local heuristics and previous knowledge. When applied to two different classification problems, the proposed algorithm achieved very promising results.

***Keywords*—**Ant colony optimization, Classification, Data mining, Feature selection.

I. Introduction

FINDING the best feature subset for a given problem with N number of features requires evaluating all 2^N possible subsets. The best feature subset also depends on the predictive modeling, which will be employed to predict the future unknown values of response variables of interest.

Many factors affect the success of machine learning on a given task. The quality of the data is one such factor, if information is irrelevant or redundant, or the data is noisy and unreliable, the knowledge discovery during training is more difficult.

Feature subset selection is the process of identifying and removing as much of the irrelevant and redundant information as possible. At one extreme are algorithms such as the simple nearest neighbor learner, that classifies novel examples by retrieving the nearest stored training examples, using all the available features in its distance computations. Towards the other extreme lie algorithms that explicitly try to focus on relevant features and ignore irrelevant ones. Decision trees are examples of this approach. By testing the values of certain features, decision tree algorithms attempt to divide training data into subsets containing a strong majority of one class. This necessitates the selection of a small number of highly predictive features in order to avoid over-fitting the training data. Regardless of whether a learner attempts to select features itself or ignores the issue, feature selection prior leaning can be beneficial. Reducing the dimensionality of the data reduces the size of the hypothesis space and allows algorithms to operate faster and more effectively. [1]

Algorithms for feature selection fall into two broad categories: wrapper that use the learning algorithm itself to evaluate the usefulness of features and filters that evaluate features according to heuristics based on general characteristics of the data. For application to large databases, filters have proven to be more practical than wrappers because they are much faster [2].

The filter method for feature selection operates independently of any learning algorithm and undesirable features are filtered out of the data before induction commences. Some look for consistency in the data, that is, they note when every combination of values for a feature subset is associated with a single class label [3]. Another method [4] eliminates features whose information content is subsumed by some number of the remaining features. Still other methods attempt to rank features according to a relevancy score [5].

A feature selection algorithm performs a search through the space of feature subsets and must have following four components [6]:

- A generation procedure to generate the next candidate subset for evaluation, in other words, a search method to explore the possible variable combinations of the search space such as greedy hill climbing that local changes to the current feature subset by adding or deleting a single feature from it [7,8].
- An evaluation function to evaluate the candidate subset.
- A stopping criterion to stop searching through the space of feature subsets. For example, sequential variable selection methods in Multiple Linear Regression models terminate as soon as possible when a variable is found insignificant according to the statistical test.
- A validation procedure to check whether the subset is valid.

In this paper, we will mainly be concerned with the first component, which is the search procedure. In the next section, we give a brief description of some of the available search

M. Sadeghzadeh is scientific mission of Islamic Azad University, Mahshahr branch (e-mail: sadegh_1999@yahoo.com).
M. Teshnehlab is assoc. professor of K.N.Toosi University of technology, Tehran, IRAN (e-mail: Teshnehlab@eetd.kntu.ac.ir).

procedure algorithms and their limitations. An explanation of the Ant Colony Optimization (ACO) is presented in section three. Section four describes the proposed search procedure algorithm. Experimental results are presented in section five and a conclusion is given in section six.

II. The Available Search Procedure

A number of search procedure methods have been proposed in the literature. Some of the most famous ones are the stepwise, branch-and-bound, Genetic Algorithms (GA), and Evolutionary Programming (EP).

The stepwise search adds/removes a single feature to/from the current subset [9]. It considers local changes to the current feature subset. Often, a local change is simply the addition or deletion of a single feature from the subset. The stepwise which is also called the Sequential Forward Selection (SFS)/ Sequential Backward Selection (SBS) is probably the simplest search procedure and is generally sub-optimal and suffers from the so-called "nesting effect". It means that the features that were once selected/deleted cannot be later discarded/reselected. To overcome this problem, Pudil et al. [10] proposed a method to flexibly add and remove features, which they called "floating search".

The branch and bound algorithm [11] requires monotonic evaluation functions and is based on discarding subsets that do not meet a specified bound. When the size of feature set is moderate, the branch and boud algorithm may find a practicable solution. However, this method becomes impracticable for feature selection problems involving a large number of features, especially because it may need to search the entire feasible region to find the optimal solution. Also, it may not be possible to use the branch and bound algorithm in wrapper methods because of the monotonic constraint of the evaluation function, where the classification accuracy is not guaranteed to increase by including more features.

Another search procedure is based on the Genetic Algorithm (GA), which is a combinatorial search technique based on both random and probabilistic measures. Subsets of features are evaluated using a fitness function and then combined via cross-over and mutation operators to produce the next generation of subsets [12]. The GA employ a population of competing solutions, evolved over time, to converge to an optimal solution. The simplest representation is a binary representation, where each chromosome consists of fixed-length binary string with a size the number of features in the problem. Each bit in the chromosome represents either the elimination or the inclusion of the corresponding feature. Although binary representation is able to represent all possible feature subsets, it can still cause premature convergence, because it search in subspace of the total search space [13]. Even with the stratified selection scheme proposed [13], the classical binary crossover and mutation operators may produce off-springs, which belong to a different subspace than that of the parent chromosomes. This causes a loss of information gained by the GA for the current subspace. Thus binary genetic operators may cause to poor convergence since the GA cannot exploit the information properly for the strait subspaces. Another representation is a floating point representation, where each chromosome is represented as floating point arrays with a size the number of reduced features, and each gene corresponds to variable number in the feature subset [14]. According to [15, 16], the GA was able to achieve better performance than other conventional methods.

Another search procedure is based on the Evolutionary Programming (EP), which is a stochastic optimization method similar to genetic algorithms. In an application of EP, after initializing the population, all reduced features (individuals) are selected to be parents. Only mutation is used for producing same number of children from parents and survivors are chosen from all individuals (parents plus children), using a probabilistic function based on fitness. In other words, individuals with a greater fitness have a higher chance to present in the next generation. In presentation, each chromosome corresponds to one of the permutations of an array with the numbers from 1 to number of features. This means that each chromosome has a different ordered list of all features. The first section of chromosome with number of reduced features is used for calculating the fitness of the chromosome. This means the second section, i.e. remaining values correspond to unselected features [17].

We propose in this paper a subset search procedure that utilize the ant colony optimization algorithm and aims at achieving similar to better results than GA-based feature selection

III. Ant Colony Optimization

In real ant colonies, a pheromone, which is an odorous substance, is used as an indirect communication medium. When a source of food is found, ants lay some pheromone to mark the path. The quantity of the laid pheromone depends upon the distance, quantity and quality of the food source. While an isolated ant that moves at random detects a laid pheromone, it is very likely that it will decide to follow its path. This ant will itself lay a certain amount of pheromone, and hence enforce the pheromone trail of that specified path. Accordingly, the path that has been used by more ants will be more attractive to follow. In other words, the probability which an ant chooses a path increases with the number of ants that previously chose that path. This process is hence characterized by a positive feedback loop [18].

Dorigo et. Al [19] adopted this concept and proposed an artificial colony of ants algorithm, which was called the Ant Colony Optimization (ACO) meta-heuristic, to solve hard combinatorial optimization problems. The ACO was originally applied to solve the classical travelling salesman problem [18] where it was shown tobe an effective tool in finding good solutions. The ACO has also been successfully applied to other optimization problems including data mining, tele-communications networks, vehicle routing, etc [20, 21, 22].

In order to solve an optimization problem, a number of artificial ants are used to iteratively construct solutions. In each iteration an ant would deposit a certain amount of

pheromone proportional to the quality of the solution. At each step, every ant computes a set of feasible expansions to its current partial solution and selects one of these depending on two factors: local heuristics and prior knowledge.

For the classical Travelling Salesman Problem (TSP) [18], each artificial ant represents a simple "agent". Each agent explores the surrounding space and builds a partial solution based on local heuristics, i.e., distance to neighboring cities, and on information from previous attempts of other agents, i.e., pheromone trail or the usage of paths from previous attempts by the rest of the agents. In the first iteration, solutions of the various agents are only based on local heuristics. At the end of the iteration, "artificial pheromone" will be laid. The pheromone intensity on the various paths will be proportional to the optimality of the solutions. As the number of iterations increases, the pheromone trail will have a greater effect on the agents' solutions.

It is worth mentioning that ACO makes probabilistic decision in terms of the artificial pheromone trails and the local heuristic information. This allows ACO to explore larger number of solutions than greedy heuristics. Another characteristic of the ACO algorithm is thepheromone trail evaporation, which is a process that leads to decreasing the pheromone trail intensity over time. According to [19], pheromone evaporation helps in avoiding rapid convergence of the algorithm towards a sub-optimal region.

Please note that searching the feature space in the problem of feature selection is quite different from the other optimization problems that researchers attempted to solve using ACO. In the next section, we present our proposed ACO algorithm, and explain how it is used for searching the feature space and selecting an "appropriate" subset of features.

IV. The Proposed Search Procedure

For a given classification task, the problem of feature selection can be stated as follows: given the original set, F, of n features, find subset S, which consists of m features ($m < n, S \subset F$), such that the classification accuracy is maximized.

The feature selection representation exploited by artificial ants includes the following:

- n features that constitute the original set, $F = \{f_1, \ldots, f_n\}$.
- A number of artificial ants to search through the feature space (na ants).
- τ_i, the intensity of pheromone trail associated with feature f_i, which reflects the previous knowledge about the importance of f_i.
- For each ant j, a list that contains the selected feature subset, $S_j = \{s_1, \ldots, s_m\}$.

We propose to use a hybrid evaluation measure that is able to estimate the overall performance of subsets as well as the local importance of features. A classification algorithm is used to estimate the performance of subsets (i.e., wrapper evaluation function). On the other hand, the local importance of a given feature is measured using the correlation based evaluation function, which is a filter evaluation function.

In the first iteration, each ant will randomly choose a feature subset of m features. Only the best k subsets, $k < na$, will be used to update the pheromone trail and influence the feature subsets of the next iteration. In the second and following iterations, each ant will start with $m - p$ features that are randomly chosen from the previously selected k-best subsets, where p is an integer that ranges between 1 and m-1. In this way, the features that constitute the best k subsets will have more chance to be present in the subsets of the next iteration.

However, itwill still be possible for each ant to consider other features as well. For a given ant j, those features are the ones that achieve the best compromise between pheromone trails and local importance with respect to S_j, where S_j is the subset that consists of the features that have already been selected by ant j. The Updated Selection Measure (USM) is used for this purpose and defined as:

$$USM_i^{S_j} = \begin{cases} \dfrac{(\tau_i)^\alpha (LI_i^{S_j})^\beta}{\sum_{g \notin S_j} (\tau_g)^\alpha (LI_g^{S_j})^\beta} & if \quad i \notin S_j \\ 0 \qquad Otherwise \end{cases} \tag{1}$$

Where $LI_i^{S_j}$ is the local importance of feature f_i given the subset S_j. The parameters α and β control the effect of pheromone trail intensity and local feature importance respectively. $LI_i^{S_j}$ is measured using the correlation measure and defined as:

$$LI_i^{S_j} = \frac{|C_{iR}|}{\sum_{f_s \in S_j} [C_{is}]} \tag{2}$$

Where $|C_{iR}|$ is the absolute value of the correlation between feature i (f_i) and the response (class) variable R, and $|C_{is}|$ is the absolute value of the inter-correlation between feature i (f_i) and feature s (f_s) that belongs to S_j.

Below are the steps of the algorithm:

1. Initialization:
   - Set $\tau_i = cc$ and $\Delta T_i = 0, (i = 1, \ldots, n)$, where cc is a constant and $\Delta\tau_i$ is the amount of

change of pheromone trail quantity for feature f_i.

- Define the maximum number of iterations.
- Define k, where the k-best subsets will influence the subsets of the next iteration.
- Define p, where $m-p$ is the number of features that each ant will start with in the second and following iterations.

2. If in the first iteration,
   - For j=1 to na,
     - Randomly assign a subset of m features to S_j.
   - Goto step 4.
3. Select the remaining p features for each ant:
   - For $mm = m-p+1$ to m,
     - For j=1 to na,
       - Given subset S_j, Choose feature f_i that maximizes $USM_i^{S_j}$.
       - $S_j = S_j \cup \{f_i\}$.
   - Replace the duplicated subsets, if any, with randomly chosen subsets.
4. Evaluate the selected subset of each ant using a chosen classification algorithm:
   - For j=1 to na,
     - Estimate the Error (E_j) of the classification results obtained by classifying the features of S_j.
   - Sort the subsets according to their E. Update the minimum E (if achieved by any ant in this iteration), and store the corresponding subset of features.
5. Using the feature subsets of the best k ants, update the pheromone trail intensity:
   - For j=1 to k, /* update the pheromone trails */

$$\Delta\tau_i = \begin{cases} \dfrac{\max\limits_{g=1:k}(E_g) - E_j}{\max\limits_{h=1:k}(\max\limits_{g=1:k}(E_g) - E_h)} & if \quad f_i \in S_j \\ 0 & Otherwise \end{cases} \quad (3)$$

$$\tau_i = \rho.\tau_i + \Delta\tau_i \quad (4)$$

Where ρ is a constant such that $(1-\rho)$ represents the evaporation of pheromone trails.

6. If the number of iterations is less than the maximum number of iterations, or the desired E has not been achieved, initialize the subsets for next iteration and goto step 3 :
   - For j=1 to na,
     - From the features of the best k ants, randomly produce $m-p$ feature subset for ant j, to be used in the next iteration, and store it in S_j.
   - Goto step 3.

It is worth mentioning that there is little difference between the computational cost of the proposed algorithm and the GA-based search procedure. This is due to the fact that both of them evaluate the selected subsets using a "wrapper approach", which requires far more computational cost than the "filter approach" used in the proposed algorithm to evaluate the local importance of features.

V. EXPERIMENTS

In order to evaluate the effectiveness of this algorithm as a feature selector for common machine learning algorithms, experiments were performed using seven data sets from the UCI collection [23] and three additional artificial domains were borrowed from work by Langly and Sage [24] that each have 3 relevant features to which a further 27 irrelevant features have been added. The data sets and their characteristics are listed in Table 1.

TABLE 1
DISCRETE CLASS DATA SETS

Data set	instances	attributes	# of classes
Contact-lenses	24	4	3
Diabetes	768	8	2
Glass	214	10	7
Heart-statlog	270	13	2
Iris	150	4	3
Vehicle	946	18	4
Zoo	101	17	7
B1	120	30 (27 irrelevant)	2
B2	120	30 (27 irrelevant)	2
B3	120	30 (27 irrelevant)	2

B1 = $x_1 x_2 x_3$

B2 = $x_1 x_2 \vee x_1 x_3 \vee x_2 x_3$

B3 = $x_1 x_2 x_3 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3$

Two machine learning algorithms representing two diverse approaches to learning were used in the experiments, a probabilistic learner (Naive Bayes), and an instance-based learner (KNN). Naïve Bayes is a bayesian classifier that assumes independence between features in order to simplify the calculation of conditional probabilities. KNN is an instance based learner that classifies a new instance by comparing it

with a data base of instances seen during training. The class of the most similar instance is assigned to the new instance.

We use from forward feature selection (FFS), backward feature selection (BFS), paudil's floating feature selection (PFFS), and branch and bound feature selection (BBFS) which criterion of evaluating features is 1-nearest neighbor leave-one-out classification performance and 60 percent of data were used to train and 40 percent to test them.

We then compare the classification error rate for these feature selection method with ant colony optimization feature selection (ACOFS). The parameters of ACOFS algorithm described in the previous section are assigned the following values:

- $\alpha = \beta = 1$, which basically makes the trail intensity and local measure equally important.
- The number of ants, $na = 200$, and the maximum number of iterations is 20.
- k =50. Thus, only the best $na/4$ ants are used to update the pheromone trails and affect the feature subsets of the next iteration.
- if $m \prec 9$ then $p = \lfloor m/2 \rfloor$, otherwise $m - p = \max(m-4, round(0.6 \times m))$, where p is the number of the remaining features that need to be selected in each iteration. It can be seen that p will be equal to 4 if $m \geq 9$. The rational behind this is that evaluating the importance of features locally becomes less reliable as the number of selected features increases. In addition, this will reduce the computational cost especially for large values of m.
- The initial value of trail intensity $cc = 1$, and the trail evaporation is 0.25, i.e. $\rho = 0.75$.
- The Error of a k-nearest neighbor classifier trained with randomly chosen 60 percent of data is used to evaluate the performance of the selected subsets in each iteration.

The selected features in each method showed in Table 2.

The classification error rate were calculated for each algorithm- data set combination before and after feature selection, and showed in Tables 3 and 4. In the last column, bold numbers show the error rate with ACOFS ratio to error rate without feature selection not increased. All the programs for feature selection developed in Matlab software.

TABLE II
THE SELECTED FEATURES

Dataset	FFS	BFS	PFFS	BBFS	ACOFS
Contact-lenses	4,3	3,4	3,4	3,4	3,4
Diabetes	1,2,8	1,2,3,5,8	1,2,8	2,6	2,6,8
Glass	1,3,5	1,3,9	1,3,5	1,3	1,3,5
Heart-statlog	8,9,10,13	2,3,6,10, 12,13	2,3,6,10, 12,13	5,10	3,6,7,9
Iris	3,4	3,4	3,4	3,4	3,4
Vehicle	1,3,5,6, 10,12,13, 15	1,2,3,6,8 , 10,11,14, 15,17,18	1,3,5,6, 8,14,17, 18	10,12	1,2,3, 5,8,9, 17,18
Zoo	3,5,10, 11,13,14, 15	4,5,7,9, 10,12,13 , 14	5,10,11, 13,14,15	3,5	4,5,7, 10,13 14,15
B1	1,2,3	1,2,3	1,2,3	29,30	1,2,3
B2	1,2,3	1,2,3	1,2,3	2,3	1,2,3
B3	1,2,3	1,2,3	1,2,3	29,30	1,2,15

TABLE III
THE CLASSIFICATION ERROR RATE (NAIVE BAYES)

Dataset	The classification Error rate Without and with feature selection					
	Without	FFS	BFS	PFFS	BBFS	ACOFS
Contact-lenses	0.333	0.333	0.333	0.333	0.333	**0.333**
Diabetes	0.238	0.287	0.313	0.287	0.244	**0.212**
Glass	0.205	0.157	0.169	0.157	0.181	**0.157**
Heart-statlog	0.120	0.176	0.111	0.111	0.306	0.204
Iris	0.050	0.017	0.017	0.017	0.017	**0.017**
Vehicle	0.390	0.414	0.363	0.402	0.548	0.393
Zoo	0.103	0.077	0.128	0.080	0.256	**0.077**
B1	0.125	0.125	0.125	0.125	0.125	**0.125**
B2	0.083	0	0	0	0.313	**0**
B3	0.250	0.250	0.250	0.250	0.250	**0.250**

Experiments showed improvement in some natural domains but not in others, apparently because some of the UCI data sets contain correlated features but few irrelevant ones. Feature selection significantly improves the performance of Naïve bayes and k-nn classifiers in most of natural domains and artificial domains. In artificial domains ACOFS has successfully removed the27 irrelevant attributes from the first two boolean domains (B1& B2). As expected, ACOFS is not effective on the third of the boolean domains (B3). Due to the high degree of feature interaction in this domain, none of the relevant features in isolation can be distinguished from the irrelevant ones. Also, in most cases ACOFS is better than other feature selection methods.

TABLE IV
THE CLASSIFICATION ERROR RATE (k-NN CLASSIFIER)

Dataset	The classification Error rate Without and with feature selection					
	Without	FFS	BFS	PFFS	BBFS	ACOFS
Contact-lenses	0.333	0.111	0.111	0.111	0.111	**0.111**
Diabetes	0.280	0.261	0.257	0.261	0.254	**0.251**
Glass	0.024	0.024	0.036	0.024	0.036	**0.024**
Heart-statlog	0.315	0.222	0.176	0.176	0.296	**0.167**
Iris	0.017	0.067	0.067	0.067	0.067	0.067
Vehicle	0.348	0.333	0.292	0.274	0.467	0.289
Zoo	0.410	0.026	0.026	0.026	0.333	**0.026**
B1	0.104	0	0	0	0.125	**0**
B2	0.208	0	0	0	0.271	**0**
B3	0.229	0	0	0	0.250	0.188

VI. CONCLUSION

In this paper, we presented a novel feature selection search procedure based on the Ant Colony Optimization meta-heuristic. The proposed algorithm utilizes both local importance of features and overall performance of subsets to search through the feature space for optimal solutions. When used to select features in presented datasets, the proposed algorithm outperformed other feature selection methods (FFS,BFS,PFFS,BBFS). Results show this algorithm selects a good subset of features that are useful to common machine learning algorithms by improving their accuracy and making their results easier to understand specially in data sets with irrelevant or redundant features. It is obvious that in feature selection, improvement in correct classification rate depends on the correlation between features, and hence depends on data set. Therefore in data sets with uncorrelated features and without irrelevant features, feature selection may be result decreasing of correct classification rate. Other advantage of proposed algorithm is that it scales only with the number of features. Therefore, does not require extra computation cost if the number of the data points in a dataset increases. Proposed evaluation function cannot identify strongly interacting features such as in a parity problem. Extension of proposed algorithm to deal with feature interactions will be explored. One approach might be to model higher order dependencies, that is, correlation between pairs of features and the class.

REFERENCES

[1] Hall, M.A. Correlation-based Feature selection for Machine Learning, *Ph.D. Thesis, Department of Computer Science.* Hamilton, New Zeland: The University of Waikato, 1999.

[2] Kohavi, R., and John, G. H, : The Wrapper Approach, *Feature Selection for Knowledge Discovery and Data Mining*, H. Liu and H. Motoda Eds., Kluwer Academic Publishers, 1998, 33-50.

[3] Almuallim, *H.* and Dietterich, T. G., Learning with many irrelevant features, *Proceedings of the Ninth National Conference on Artificial Intelligence,* AAAI Press, 1991, 547-552.

[4] Koller, D. and Sahami, M. , Toward optimal feature selection, *Proceedings of the Thirteenth International Conference on Machine Learning*, Morgan Kaufmann, 1996, 248-292.

[5] Kira, K. and Rendell, L. A practical approach to feature selection, *Proceedings of the Ninth International Conferene on Machine Learning,* Morgan Kaufmann, 1992, 249-256.

[6] Dash, M., Liu, H. , Consistency-based search in feature selection, *Artificial Intelligence 151,* Elsevier Pub., 2003, 155-176.

[7] Deng, K. and Moore, A., On the Greediness of Feature Selection Algorithms, *International Conference of Machine Learning (ICML'98),* Van den Herik H. J. and Weijters Eds T. , University Maastricht, The Netherlands, 1998.

[8] Skalak, D., Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms, *Eleventh International MachineLearning Conference,* Morgan Kaufmann, New Brunswick, NJ, 1994, 293-301.

[9] J. Klitter. Feature set search algorithms. In C. H. Chen, editor, *Pattern Recognition and signal Processing.* Sijhoff and Noordhoff, the Netherlands, 1978.

[10] P. Paudil, J. Novovicova, and J. Klittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15: 1994, 1119-1125.

[11] P.M. Narendra and K. Fukunaga, A branch and bound algorithm for feature subset selection. *IEEE Transaction on Computers*, C-26, 1977, 917-922.

[12] J. Yang and V. Honavar, Feature subset selection using a genetic algorithm, *IEEE Transactions on Intelligent Systems*, 13, 1998, 44-49.

[13] Leardi, R., Boggia, R. and Terrile, M., Genetic Algorithms as a Strategy for Feature Selection, *Journal of Chemometrics, vol. 6*, 1992, 267-281

[14] M. Sadeghzadeh and M.H. Shenasa, Correlation Based Feature Selection Using Genetic Algorithms, *Proceedings of the Third International Conference on Modeling, Simulation and Applied Optimization, Sharjah, U.A.E. ,* 2009.

[15] M. Gletsos, S.G. Mougiakakou, G.K. Matsopoulos, K.S. Nikita, A. S. Nikita, and D. Kelekis. A Computer-Aided Diagnostic System to Characterized CT Focal Liver Lesions: Design and Optimization of a Neural Network Classifier, *IEEE Transactions on Information Technology in Biomedicine, 7*, 2003, 153-162.

[16] I.-S. Oh, J.-S. Lee, and B.-R. Moon, Hybrid Genetic Algorithms for Feature Selection, *IEEE Transactions on Pattern Analysis and Machine Inteligence,* 26, 2004, 1424-1437.

[17] M. Sadeghzadeh and M.H. Shenasa, Correlation Based Feature Selection Using Evolutionary Programming, *Proceedings of the Twelfth IASTED International Conference on Artificial Intelligence and Soft Computing, Palma de Mallorca, Spain* , 2008, 185-190.

[18] M. Dorigo, V. Maniezzo, and A. Colorni. Ant Systems: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B,* 26: 1996, 29-41.

[19] T. Stutzle and M. Dorigo, The Ant Colony Optimization Metaheuristic: Algorithm, Applications, and Advances. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, Kluwer Academic Publishers, Norwell, MA, 2002.

[20] G. Di Caro and M. Dorigo, AntNet:Distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9: 1998, 317-365.

[21] R. S. Parpinelli: H. S. Lopes, A. A. Freitas, Data mining with an ant colony optimization algorithm, *IEEE Transactions on Evolutionary Computation*,6, 2002, 321-332.

[22] G. Di Caro and M. Dorigo. AntNet: Distributed stigmergetic control for communications networks. *Journal of Artificial Inteligence Research*, 9, 1998, 317-365.

[23] Blake, C., Keogh, E., and Merz C. J., *UCI Repository of Machine Learning Data Bases*, University of California, Department of Information and Computer Science,Irvine,CA.,1998. [http://www.ics.uci.edu/~mlean/MLRepository.html]

[24] Langky, P. and Sage, S., Scaling to domains with irrelevant features, *R. Greiner (ed) Computational Learning Theory and Natural Learning Systems*,MIT Press, 1994.