Feature Selection Using Ant Colony Optimization (ACO): A New Method and Comparative Study in the Application of Face Recognition System

Hamidreza Rashidy Kanan, Karim Faez, and Sayyed Mostafa Taheri

Image Processing and Pattern Recognition Lab, Electrical Engineering Department,
Amirkabir University of Technology (Tehran Polytechnic), Hafez Avenue, Tehran, Iran, 15914
{rashidykanan, kfaez, mostafa_taheri}@aut.ac.ir

Abstract. Feature Selection (FS) and reduction of pattern dimensionality is a most important step in pattern recognition systems. One approach in the feature selection area is employing population-based optimization algorithms such as Genetic Algorithm (GA)-based method and Ant Colony Optimization (ACO)-based method. This paper presents a novel feature selection method that is based on Ant Colony Optimization (ACO). ACO algorithm is inspired of ant's social behavior in their search for the shortest paths to food sources. Most common techniques for ACO-Based feature selection use the priori information of features. However, in the proposed algorithm, classifier performance and the length of selected feature vector are adopted as heuristic information for ACO. So, we can select the optimal feature subset without the priori information of features. This approach is easily implemented and because of using one simple classifier in it, its computational complexity is very low. Simulation results on face recognition system and ORL database show the superiority of the proposed algorithm.

Keywords: Feature Selection, Ant Colony Optimization (ACO), Genetic Algorithm, Face Recognition.

1 Introduction

Several parameters can affect the performance of pattern recognition system. Among them, feature extraction and representation of patterns can be considered as a most important. Reduction of pattern dimensionality via feature extraction and selection belongs to the most fundamental step in data processing [1].

Feature Selection (FS) is extensive and spread across many fields, including document classification, data mining, object recognition, biometrics, remote sensing and computer vision [2]. Given a feature set of size n, the FS problem is to find a minimal feature subset of size m (m<n) while retaining a suitably high accuracy in representing the original features. In real word problems FS is a must due to the abundance of noisy, irrelevant or misleading features [3].

As a simplest way, the best subset of features can be found by evaluating all the possible subsets, which is known as exhaustive search. This procedure is quite

P. Perner (Ed.): ICDM 2007, LNAI 4597, pp. 63–76, 2007.

impractical even for a moderate size feature set. Because the number of feature subset combinations with m features from a collection of n (m<n, $m \neq 0$) feature is $n!/[m!(n-m)!]$ and the total number of theses combinations is $(2^n - 2)$.

For most practical problems, an optimal solution can only be guaranteed if a monotonic criterion for evaluating features can be found, but this assumption rarely holds in the real-world [4]. As a result, we must find solutions which would be computationally feasible and represent a trade-off between solution quality and time.

Usually FS algorithms involve heuristic or random search strategies in an attempt to avoid this prohibitive complexity. However, the degree of optimally of the final feature subset is often reduced [3].

Among too many methods which are proposed for FS, population-based optimization algorithms such as Genetic Algorithm (GA)-based method and Ant Colony Optimization (ACO)-based method have attracted a lot of attention. These methods attempt to achieve better solutions by using knowledge from previous iterations.

Genetic algorithms (GA's) are optimization techniques based on the mechanics of natural selection. They used operations found in natural genetics to guide itself through the paths in the search space [5]. Because of their advantages, recently, GA's have been widely used as a tool for feature selection in pattern recognition.

Metaheuristic optimization algorithm based on ant's behavior (ACO) was represented in the early 1990s by M. Dorigo and colleagues [6]. ACO is a branch of newly developed form of artificial intelligence called Swarm Intelligence. Swarm intelligence is a field which studies "the emergent collective intelligence of groups of simple agents" [7]. In groups of insects which live in colonies, such as ants and bees, an individual can only do simple task on its own, while the colony's cooperative work is the main reason determining the intelligent behavior it shows [8].

ACO algorithm is inspired of ant's social behavior. Ants have no sight and are capable of finding the shortest route between a food source and their nest by chemical materials called pheromone that they leave when moving.

ACO algorithm was firstly used in solving Traveling Salesman Problem (TSP) [9]. Then has been successfully applied to a large number of difficult problems like the Quadratic Assignment Problem (QAP) [10], routing in telecommunication networks, graph coloring problems, scheduling and etc. This method is particularly attractive for feature selection as there seems to be no heuristic that can guide search to the optimal minimal subset every time [3]. In the other hand, if features are represented as a graph, ant will discover best feature combinations as they traverse the graph.

In this paper a new modified ACO-Based feature selection algorithm has been introduced. The classifier performance and the length of selected feature vector are adopted as heuristic information for ACO. So proposed algorithm needs no priori knowledge of features. Proposed algorithm is applied to two different feature subsets that are Pseudo Zernike Moment Invariant (PZMI) and Discrete Wavelet Transform (DWT) Coefficients in the application of face recognition system and finally the classifier performance and the length of selected feature vector are considered for performance evaluation.

The rest of this paper is organized as follows. Section 2 presents a brief overview of feature selection methods. Ant Colony Optimization (ACO) and Genetic Algorithm (GA) are described in Sections 3 and 4 respectively. Section 5 explains the proposed feature selection algorithm and finally, Sections 6 and 7 attain the experimental results and conclusion.

2 An Overview of Feature Selection (FS) Approaches

Feature selection algorithms can be classified into two categories based on their evaluation procedure [11]. If an algorithm performs FS independently of any learning algorithm (i.e. it is a completely separate preprocessor), then it is a filter approach (open-loop approach). This approach is based mostly on selecting features using between-class separability criterion [11]. If the evaluation procedure is tied to the task (e.g. classification) of the learning algorithm, the FS algorithm employs the wrapper approach (closed-loop approach). This method searches through the feature subset space using the estimated accuracy from an induction algorithm as a measure of subset suitability.

The two mentioned approaches are also classified into five main methods which they are Forward Selection, Backward elimination Forward/Backward Combination, Random Choice and Instance based method.

FS methods may start with no features, all features, a selected feature set or some random feature subset. Those methods that start with an initial subset usually select these features heuristically beforehand. Features are added (Forward Selection) or removed (Backward Elimination) iteratively and in the Forward/Backward Combination method features are either iteratively added or removed or produced randomly thereafter.

The disadvantage of Forward Selection and Backward Elimination methods is that the features that were once selected/eliminated cannot be later discarded/re-selected. To overcome this problem, Pudil et al. [12] proposed a method to flexibly add and remove features. This method has been called floating search method.

In the wrapper approach the evaluation function calculates the suitability of a feature subset produced by the generation procedure and compares this with the previous best candidate, replacing it if found to be better. A Stopping criterion is tested every iteration to determine whether the FS process should continue or not.

Other famous FS approaches are based on the Genetic Algorithm (GA) [13], Simulated Annealing [3] and Ant Colony Optimization (ACO) [3, 8, 14, 15, 16].

[14] has proposed a hybrid approach for speech classification problem. This method has used combination of mutual information and ACO. [15] has used the hybrid of ACO and mutual information for selection of features in the forecaster. [16] has utilized the Fisher Discrimination Rate (FDR) as a heuristic information in the ACO-Based feature selection method which is used for selection of network intrusion features. [3] has used a ACO for finding rough set reducts. [8] has introduced a Ant-Miner which is used a difficult pheromone updating strategy and sate transition rule.

Also, some surveys of feature selection algorithms are given in [1, 17, 18].

3 Ant Colony Optimization (ACO)

In the early 1990s, ant colony optimization (ACO) was introduced by M. Dorigo and colleagues as a novel nature-inspired metaheuristic for the solution of hard combinatorial optimization (CO) problems [19]. ACO belongs to the class of metaheuristics, which are approximate algorithms used to obtain good enough solutions to hard CO problems in a reasonable amount of computation time [19].

The ability of real ants to find shortest routes is mainly due to their depositing of pheromone as they travel; each ant probabilistically prefers to follow a direction rich in this chemical. The pheromone decays over time, resulting in much less pheromone on less popular paths. Given that over time the shortest route will have the higher rate of ant traversal, this path will be reinforced and the others diminished until all ants follow the same, shortest path (the "system" has converged to a single solution) [3].

In general, an ACO algorithm can be applied to any combinatorial problem as far as it is possible to define:

- Appropriate problem representation. The problem must be described as a graph with a set of nodes and edges between nodes.
- Heuristic desirability (η) of edges. A suitable heuristic measure of the "goodness" of paths from one node to every other connected node in the graph.
- Construction of feasible solutions. A mechanism must be in place whereby possible solutions are efficiently created.
- Pheromone updating rule. A suitable method of updating the pheromone levels on edges is required with a corresponding evaporation rule. Typical methods involve selecting the n best ants and updating the paths they chose.
- Probabilistic transition rule. The rule that determines the probability of an ant traversing from one node in the graph to the next.

3.1 ACO for Feature Selection

The feature selection task may be reformulated into an ACO-suitable problem. ACO requires a problem to be represented as a graph. Here nodes represent features, with the edges between them denoting the choice of the next feature. The search for the optimal feature subset is then an ant traversal through the graph where a minimum number of nodes are visited that satisfies the traversal stopping criterion. Figure 1 illustrates this setup. The ant is currently at node ***a*** and has a choice of which feature to add next to its path (dotted lines). It chooses feature ***b*** next based on the transition rule, then ***c*** and then ***d***. Upon arrival at ***d***, the current subset *{a; b; c; d}* is determined to satisfy the traversal stopping criterion (e.g. a suitably high classification accuracy has been achieved with this subset). The ant terminates its traversal and outputs this feature subset as a candidate for data reduction.

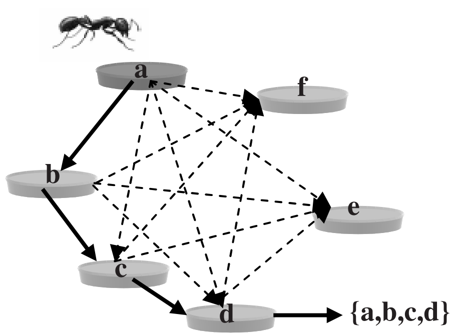

Fig. 1. ACO problem representation for FS

A suitable heuristic desirability of traversing between features could be any subset evaluation function - for example, an entropy-based measure [19], rough set dependency measure [20] or the Fisher Discrimination Rate (FDR)[16]. The heuristic desirability of traversal and edge pheromone levels are combined to form the so-called probabilistic transition rule, denoting the probability of an ant at feature i choosing to travel to feature j at time t:

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{l \in J_i^k} [\tau_{il}(t)]^{\alpha}.[\eta_{il}]^{\beta}} & if\ \ j \in J_i^k \\ 0 & otherwise \end{cases} \tag{1}$$

Where k is the number of ants, η_{ij} is the heuristic desirability of choosing feature **j** when at feature **i** (η_{ij} is optional but often needed for achieving a high algorithm performance [21]), J_i^k is the set of neighbor nodes of node **i** which have not yet been visited by the ant **k.** $\alpha \succ 0, \beta \succ 0$ are two parameters that determine the relative importance of the pheromone value and heuristic information (the choice of α, β is determined experimentally) and $\tau_{ij}(t)$ is the amount of virtual pheromone on edge (i,j).

The overall process of ACO feature selection can be seen in figure 2. The process begins by generating a number of ants, k, which are then placed randomly on the graph (i.e. each ant starts with one random feature). Alternatively, the number of ants to place on the graph may be set equal to the number of features within the data; each ant starts path construction at a different feature. From these initial positions, they traverse edges probabilistically until a traversal stopping criterion is satisfied. The resulting subsets are gathered and then evaluated. If an optimal subset has been found or the algorithm has executed a certain number of times, then the process halts and outputs the best feature subset encountered. If neither condition holds, then the pheromone is updated, a new set of ants are created and the process iterates once more.

The pheromone on each edge is updated according to the following formula:

$$\tau_{ij}(t+1) = (1-\rho).\tau_{ij}(t) + \rho.\Delta\tau_{ij}(t) \tag{2}$$

Where:

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{n} (\gamma'(S^k) \big/ |S^k|) \tag{3}$$

This is the case if the edge (i;j) has been traversed; $\Delta\tau_{ij}(t)$ is 0 otherwise. The value $0 \le \rho \le 1$ is decay constant used to simulate the evaporation of the pheromone, S^k is the feature subset found by ant **k**. The pheromone is updated

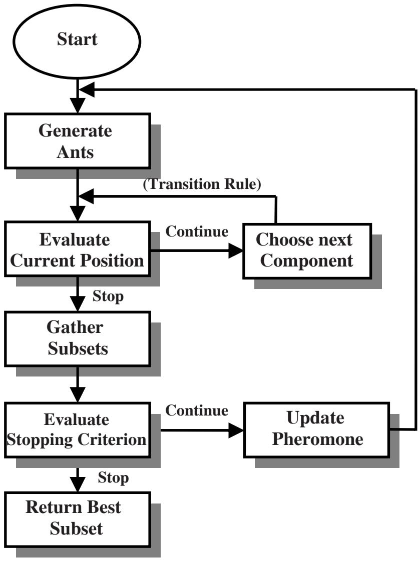

Fig. 2. ACO-based feature selection overview

according to both the measure of the "goodness" of the ant's feature subset γ' and the size of the subset itself. By this definition, all ants update the pheromone.

4 Genetic Algorithm (GA)

The GA's is a stochastic global search method that mimics the metaphor of natural biological evolution [5]. These algorithms are general purpose optimization algorithms with a probabilistic component that provide a means to search poorly understood, irregular spaces.

GA's work with a population of points rather than a single point. Each ''point'' is a vector in hyperspace representing one potential (or candidate) solution to the optimization problem. A population is, thus, just an ensemble or set of hyperspace vectors. Each vector is called a chromosome in the population. The number of elements in each vector (chromosome) depends on the number of parameters in the optimization problem and the way to represent the problem. How to represent the problem as a string of elements is one of the critical factors in successfully applying a GA (or other evolutionary algorithm) to a problem.

A typical series of operations carried out when implementing a GA paradigm is:

- ❖ Initialize the population;
- ❖ Calculate fitness for each chromosome in population;
- ❖ Reproduce selected chromosomes to form a new population;
- ❖ Perform crossover and mutation on the population;
- ❖ Loop to second step until some condition is met.

Initialization of the population is commonly done by seeding the population with random values. The fitness value is proportional to the performance measurement of the function being optimized. The calculation of fitness values is conceptually simple. It can, however, be quite complex to implement in a way that optimizes the efficiency of the GA's search of the problem space. It is this fitness that guides the search of the problem space.

After fitness calculation, the next step is reproduction. Reproduction comprises forming a new population, usually with the same total number of chromosomes, by selecting from members of the current population using a stochastic process that is weighted by each of their fitness values. The higher the fitness, the more likely it is that the chromosome will be selected for the new generation. One commonly used way is a ''roulette wheel'' procedure that assigns a portion of a roulette wheel to each population member where the size of the portion is proportional to the fitness value. This procedure is often combined with the elitist strategy, which ensures that the chromosome with the highest fitness is always copied into the next generation.

The next operation is called crossover. To many evolutionary computation practitioners, crossover is what distinguishes a GA from other evolutionary computation paradigms. Crossover is the process of exchanging portions of the strings of two ''parent'' chromosomes. An overall probability is assigned to the crossover process, which is the probability that given two parents, the crossover process will occur. This probability is often in the range of 0.65–0.80. The final operation in the typical GA procedure is mutation. Mutation consists of changing an element's value at random, often with a constant probability for each element in the population. The probability of mutation can vary widely according to the application and the preference of the person exercising the GA. However, values of between 0.001 and 0.01 are not unusual for mutation probability.

4.1 GA for Feature Selection

Several approaches exist for using GAs for feature subset selection. The two main methods that have been widely used in the past are as follow. First is due to [13], of finding an optimal binary vector in which each bit corresponds to a feature (Binary Vector Optimization method (BVO)). A '1' or '0' suggests that the feature is selected or dropped, respectively. The aim is to find the binary vector with the smallest number of 1's such that the classifier performance is maximized. This criterion is often modified to reduce the dimensionality of the feature vector at the same time [21]. The second and more refined technique [22]) uses an m-ary vector to assign weights to features instead of abruptly dropping or including them as in the binary case. This gives a better search resolution in the multidimensional space [23].

5 Proposed Feature Selection Algorithm

The main steps of proposed algorithm are as follows:

1) Initialization
   - Determine the population of ants (p).
   - Set the intensity of pheromone trial associated with any feature.
   - Determine the maximum of allowed iterations (k)

2) Generation ants and evaluation of each ants
    - Any ant (A_i , i=1:p) randomly is assigned to one feature and it should visit all features and build solutions completely. In this step, the evaluation criterion is Mean Square Error (MSE) of the classifier. If any ant could not decrease the MSE of the classifier in three successive steps, it finished its work and exit.
3) Evaluation of the selected subset of each ant
    - In this step the importance of the selected subset of each ant is evaluated through classifier performance. Then the subsets according to their MSE are sorted and some of them are selected according to ACS and AS_{rank} algorithms.
4) Check the stop criterion
    - If the number of iterations is more than the maximum allowed iteration exit, otherwise continue.
5) Pheromone updating
    - For features which are selected in the step 3 pheromone intensity are updated.
6) Go to 2 and continue

6 Experimental Results

To show the utility of proposed feature selection algorithm and to compare with GA-Based approach two sets of experiments were carried out. For experimental studies we have considered ORL gray scale face image database. This database contains 400 facial images from 40 individuals in different states. So, the number of classes in our experiments is 40. The total number of images in each class is 10.

Figure 3 shows some samples images of this database.

Fig. 3. Some samples of ORL database

Two different sets of features were extracted from each face image which they are Pseudo Zernike Moment Invariant (PZMI) and Discrete Wavelet Transform (DWT)

Coefficient. Then proposed ACO-based and GA-based feature selection methods are applied to each feature set and finally, the length of selected feature vector and classifier performance are considered for evaluating the proposed algorithm.

The details of experiments are as follows:

6.1 Feature Extraction

After preprocessing (histogram equalization) of facial images, we extract the PZMI and DWT coefficients as a feature vector.

In the PZMI feature extraction step, the PZMI of orders 1 to 20 and their repetitions are extracted from any face image. I.e. for any **n** (order of PZMI), we extract one feature vector of order **n** and all repetitions **m** $(m \leq n)$. For example if we choose $n = n_0$, we have one feature vector which has $(n_0 + 1)$ elements [24, 25].

In the DWT feature extraction step, Discrete Wavelet Transform is applied to any face images. Since the face images are not continuous, we used Haar wavelet which is also discrete. We applied pyramid algorithm to each preprocessed image for decomposing it into 3 resolution levels. Then we used the approximation of images at level 3 and converted them into vectors by concatenating the columns. Dimensions of ORL database images is 92×112, so after decomposing them, the length of wavelet feature for each image is 168 [26].

For scale invariancy of extracted features (PZMI and DWT Coefficients), we normalized them.

6.2 Feature Selection

After the extraction of PZMI and DWT Coefficients, ACO and GA are used to select the optimal feature sets.

We consider our system as a block diagram that is shown in Figure 4.

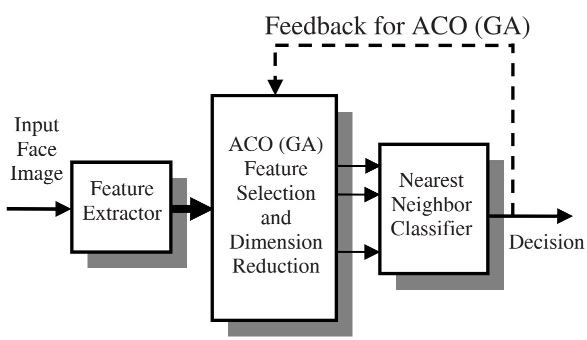

Fig. 4. Block Diagram of proposed feature selection scheme

For GA-based feature selector, we set the length of chromosomes to **L** which L=20 for PZMI features and L=168 for DWT Coefficients features. Each gene

$g_i (i = 1,2,...,L)$ corresponds to a specific order of PZMI or specific DWT Coefficient feature component.

If $g_i = 1$, this means we select this order (this feature component) as one of optimal orders (optimal components). Otherwise, $g_i = 0$ means discard it. Because most of orders (feature components) may be selected, the probability of every bit being equal to 1 is set to 0.8 when the initial population of chromosomes is creating. Its purpose is to speed up the convergence.

Given a chromosome q the fitness function F(q) is defined as:

$$F(q) = \frac{1}{\sum_{x \in \Omega} \delta(x,q)} \tag{4}$$

Here Ω is the training image set for GAs and $\delta(x,q)$ is defined as:

$$\delta(x,q) = \begin{cases} 1, & \textit{if } x \textit{ is classified correctly} \\ 0, & \textit{if } x \textit{ is misclassified} \end{cases} \tag{5}$$

For simplicity, we have used the nearest neighbor classifier and the aim is to find a binary vector with the smallest number of 1's such that the classifier performance is maximized. In order to select the individuals for the next generation, GA's roulette wheel selection method was used.

Further Genetic Algorithms parameters are summarized in Table 1.

Table 1. GA-Based Feature Selection Parameters

	PZMI Features	DWT Coefficients Features
Population size	50	50
Number Of Generation	25	25
Chromosome length	20	168
Probability of crossover	0.7	0.7
Probability of mutation	0.003	0.003

For ACO-Based feature selector, we use same primary features which are utilized in the GA-Based feature selector.

In this step, we have applied proposed algorithm to the extracted features in the formats of ACS and AS_{rank} with the same parameters.

Various parameters for leading to better convergence are tested and the best parameters that are obtained by simulations are as follows:

α=1, β=0.1, ρ=0.2, the initial pheromone intensity of each feature is equal to 1, the number of Ant in every iteration p=50 and the maximum number of iterations k=25. These values are chosen to justify the comparison with GA. Selected features of each method are classified using nearest neighbor classifier and the obtained MSE is considered for performance evaluation.

6.2.1 Comparison of ACO-Based and GA-Based Methods

The results of this step are summarized in Tables 2 and 3.

Table 2 gives, for each method, the best MSE and the average of execution time.

Table 2. MSE and execution time of Three Different Methods

Method	MSE		Time (s)	
	PZMI	DWT	PZMI	DWT
GA	3.5%	2%	1080	1560
ACO (ACS)	3%	1%	780	1320
ACO (AS_{rank})	1.5%	0.25%	300	960

Both ACO-Based methods (ACS and AS_{rank}) produce much lower classification errors and execution times than GA-Based method. ACS and AS_{rank} algorithms have comparable performance. The ACS method is faster than AS_{rank} method however it has lower performance. Also, Table 3 gives the optimal selected features for each method. Both ACO-Based and GA-Based methods significantly reduce the number of original features. But ACO-Based method (ACS and AS_{rank}) chooses fewer features.

Table 3. Selected Features of Three Different Methods

Method	Selected Features		Number of Selected Features	
	PZMI (Order)	DWT (Component)	PZMI	DWT
GA	2, 4, 8, 10, 12, 13	1, 4, 5, 6, 12, 14, 15, 17, 21, 26, 29, 32, 35, 37, 40, 44, 47, 49, 52, 53, 57, 58, 62, 64, 69, 72, 74, 79, 84, 88, 91, 93, 98, 100, 107, 111, 117, 125, 136, 137, 139, 145, 149, 155, 161, 167, 168	55	47
ACO (ACS)	4, 6, 9, 12, 13	4, 5, 20, 25, 29, 30, 37, 42, 44, 49, 58, 62, 68, 70, 73, 93, 94, 95, 96, 100, 102, 109, 112, 113, 114, 118, 120, 121, 125, 132, 138, 139, 141, 147, 149, 152, 156, 157, 158, 159, 163, 168	49	42
ACO (AS_{rank})	6, 8, 10, 14	2, 6, 18, 21, 22, 42, 49, 57, 58, 73, 75, 83, 93, 95, 96, 100, 116, 118, 122, 125, 136, 138, 144, 147, 149, 153, 157, 158, 160, 167	42	30

Tables 2 and 3 show that using proposed method, we can achieve 99.75% and 98.5% recognition rate only with 30 and 42 selected features for DWT coefficients and PZMI features respectively.

Since feature selection is typically done in an off-line manner, the execution time of a specific algorithm is of much less importance than its ultimate classification performance. So, we can say that the AS_{rank} method gives better results. Finally, selected features of each method are classified and the obtained recognition rates are shown in Figures 5 and 6.

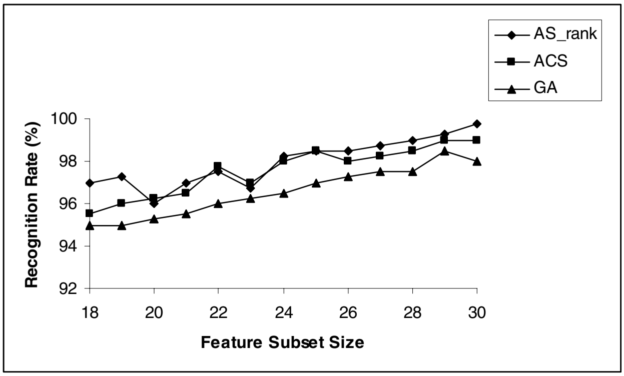

Fig. 5. Recognition Rate of DWT Feature Subsets Obtained Using AS_{rank} , ACS and GA

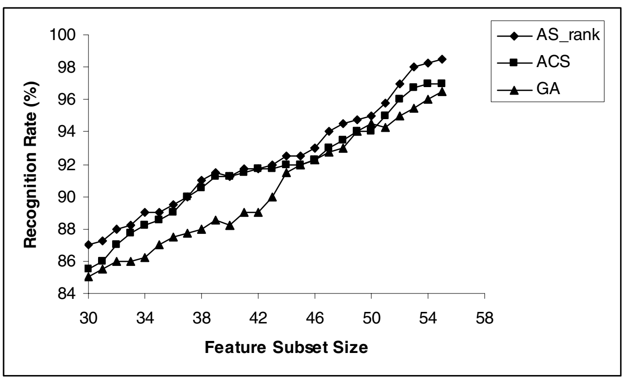

Fig. 6. Recognition Rate of PZMI Feature Subsets Obtained Using AS_{rank} , ACS and GA

It can be seen that the performance of both ACS and AS_{rank} was found to be much better than that of GA-Based method and proposed ACO-Based algorithm was able to achieve better performance than GA-Based algorithm in most of the cases.

7 Conclusion

In this paper a novel ACO-Based feature selection algorithm is presented. In the proposed algorithm, the classifier performance and the length of selected feature vector are adopted as heuristic information for ACO. So, we can select the optimal feature subset without the priori knowledge of features. Proposed approach is simulated in the ACS and AS_{rank} algorithm formats. To show the utility of proposed algorithm and to compare with GA-Based approach two sets of experiments were carried out on two different sets of features that they are PZMI and DWT coefficients. Simulation results on face recognition system and ORL database show that the proposed ACO-Based method outperforms GA-Based method since, it achieved better performance with the lower number of features and execution time.

Acknowledgment

This research was supported by the Iran Telecommunication Research Center (ITRC).

References

1. kml, L., Kittler, J.: Feature set search algorithms. In: Chen, C.H. (ed.) Pattern Recognition and Signal Processing, Sijhoff and Noordhoff, the Netherlands (1978)
2. Ani, A.A.: An Ant Colony Optimization Based Approach for Feature Selection. In: Proceeding of AIML Conference (2005)
3. Jensen, R.: Combining rough and fuzzy sets for feature selection. Ph.D. Thesis, University of Edinburgh (2005)
4. Kohavi, R.: Feature Subset Selection as search with Probabilistic Estimates. AAAI Fall Symposium On Relevance (1994)
5. Srinivas, M., Patnik, L.M.: Genetic Algorithms: A Survey. IEEE Computer Society Press, Los Alamitos (1994)
6. Dorigo, M., Caro, G.D.: Ant Colony Optimization: A New Meta-heuristic. In: Proceeding of the Congress on Evolutionary Computing (1999)
7. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)
8. Liu, B., Abbass, H.A., McKay, B.: Classification Rule Discovery with Ant Colony Optimization. IEEE Computational Intelligence 3(1) (2004)
9. Dorigo, M., Maniezzo, V., Colorni, A.: The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B 26(1), 29–41 (1996)
10. Maniezzo, V., Colorni, A.: The Ant System Applied to the Quadratic Assignment Problem. Knowledge and Data Engineering 11(5), 769–778 (1999)
11. Duda, R.O., Hart, P.E.: Pattern Recognition and Scene Analysis. Wiley, Chichester (1973)
12. Pudil, P., Novovicova, J., Kittler, J.: Floating search methods in feature selection. Pattern Recognition Letters 15, 1119–1125 (1994)
13. Siedlecki, W., Sklansky, J.: A note on genetic algorithms for large-scale feature selection. Pattern Recognition Letters 10(5), 335–347 (1989)

14. Ani, A.A.: Ant Colony Optimization for Feature Subset Selection. Transactions On Engineering, Computing And Technology 4 (2005)
15. Zhang, C.K., Hu, H.: Feature Selection Using The Hybrid Of Ant Colony Optimization and Mutual Information For The Forecaster. In: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics (2005)
16. Gao, H.H., Yang, H.H., And Wang, X.Y.: Ant Colony Optimization Based Network Intrusion Feature Selection And Detection. In: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics (2005)
17. Bins, J.: Feature Selection of Huge Feature Sets in the Context of Computer Vision. Ph.D. Dissertation, Computer Science Department, Colorado State University (2000)
18. Siedlecki, W., Sklansky, J.: On Automatic Feature Selection. International Journal of Pattern Recognition and Artificial Intelligence 2(2), 197–220 (1988)
19. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. Theoretical Computer Science 344, 243–278 (2005)
20. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishing, Dordrecht (1991)
21. Yang, J., Honavar, V.: Feature Subset Selection Using a Genetic Algorithm. IEEE Intelligent Systems 13, 44–49 (1998)
22. Punch, W.F., Goodman, E.D., Pei, L.C.S.M., Hovland, P., Enbody, R.: Further research on Feature Selection and Classification using Genetic Algorithms. In: Proc. Int. Conf. Genetic Algorithms, pp. 557–564 (1993)
23. Raymer, M., Punch, W., Goodman, E., Kuhn, L., Jain, A.K.: Dimensionality Reduction Using Genetic Algorithms. IEEE Transactions on Evolutionary Computing 4, 164–171 (2000)
24. Rashidy Kanan, H., Faez, K., Ezoji, M.: Face Recognition: An Optimized Localization Approach and Selected PZMI Feature Vector Using SVM Classifier. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCS, vol. 4113, pp. 690–696. Springer, Heidelberg (2006)
25. Rashidy Kanan, H., Faez, K., Ezoji, M.: An Efficient Face Recognition System Using a New Optimized Localization Method. In: ICPR'2006. Proceeding of the 18th International Conference on Pattern Recognition (2006)
26. Rashidy Kanan, H., Faez, K.: ZMI and Wavelet Transform Features and SVM Classifier in the Optimized Face Recognition system. In: ISSPIT 2005. Proceeding of the 5th IEEE International Symposium on Signal Processing and Information Technology, pp. 295–300. IEEE Computer Society Press, Los Alamitos (2005)