MMAS Algorithm using Fuzzy Rules

K.Sankar
Assistant Professor, Department of Master of Computer Applications,
KSR College of Engineering,
Tiruchengod, India

Dr. V.Venkatachalam
Principal
The KAVERY Engineering College,
Mecheri, Salem, India

ABSTRACT
The real life problems deal with imperfectly specified knowledge and some degree of imprecision, uncertainty or inconsistency is embedded in the problem specification. The well-founded theory of fuzzy sets is a special way to model the uncertainty. The rules in a fuzzy model contain a set of propositions, each of which restricts a fuzzy variable to a single fuzzy value by means of the predicate equivalency. That way, each rule covers a single fuzzy region of the fuzzy grid. The proposed system of this thesis extends this structure to provide more general fuzzy rules, covering the input space as much as possible. In order to do this, new predicates are considered and a Max-Min Ant System is proposed to learn such fuzzy rules.

Ant system is a general purpose algorithm inspired by the study of behavior of ant colonies. It is based on cooperative search paradigm that is applicable to the solution of combinatorial optimization problem. In this thesis we consider the combinatorial optimization issue of travelling salesman problem (TSP) which evaluates more generic Fuzzy rules provided by Max-Min Ant System (MMAS). The existing ant colony system (ACS) was a distributed algorithm applied to the travelling salesman problem (TSP). In ACS, a set of cooperating agents called ants cooperate to find good solutions for TSPs (but here, Ants search their path randomly). Ants cooperate using an indirect form of communication mediated by pheromone they deposit on the edges of the TSP problem in symmetric instances. However most of the TSP issues carry both symmetric and asymmetric instances.

1. INTRODUCTION
The system has shown that Ant Colony System (ACS) is competitive with other nature-inspired algorithms on some relatively simple problems. On the other hand, in the past years a lot of work has been done to define ad-hoc heuristics, to solve the Travelling Salesman Problem (TSP). In general, these ad-hoc heuristics greatly outperform, on the specific problem of the TSP, general purpose algorithmic approaches like evolutionary computation and simulated annealing (SA). Heuristic approaches to the TSP can be classified as tour constructive heuristics and tour improvement heuristics.

Tour constructive heuristics usually starts with selecting a random city from the set of cities and then incrementally builds a feasible TSP solution by adding new cities chosen according to some heuristic rule. For example, the nearest neighbor heuristic builds a tour by adding the closest node in term of distance from the last node inserted in the path. On the other hand, tour improvement heuristics start from a given tour and attempt to reduce its length by exchanging edges chosen according to some heuristic rule until a local optimum is found (i.e., until no further improvement is possible using the heuristic rule).

The most used and well-known tour improvement heuristics are 2-optimization and 3-optimization, and Lin-Kernighan [9] in which respectively two, three, and a variable number of edges are exchanged. It has been experimentally shown that, in general, tour improvement heuristics produce better quality results than tour constructive heuristics. A general approach is to use tour constructive heuristics to generate a solution and then to apply a tour improvement heuristic to locally optimize it. It has been shown recently that it is more effective to alternate an improvement heuristic with updation of the last (or of the best) solution produced, rather than iteratively executing a tour improvement heuristic starting from solutions generated randomly or by a constructive heuristic.

An example of successful application of the above alternate strategy is the work by Freisleben and Merz [7] in which a genetic algorithm is used to generate new solutions to be locally optimized by a tour improvement heuristic. ACS is a tour construction heuristic which, like genetic algorithm, each iteration produces a set of feasible solutions which are in some sense an updation of the previous best solution. It is therefore a reasonable guess that adding a tour improvement heuristic to ACS could make it competitive with the best algorithms.

The system has therefore added a tour improvement heuristic to ACS. In order to maintain ACS ability to solve both TSP and Adaptive Travelling Salesman Problem (ATSP) problems the system have decided to base the local optimization heuristic on a restricted 3-optimizationprocedure that, while inserting/removing three edges on the path, considers only 3-optimizationmoves that do not revert the order in which the cities are visited. The resulting algorithm is called ACS-3-optimization. In this way the same procedure can be applied to symmetric and asymmetric TSPs, avoiding unpredictable tour length changes. In addition, when a candidate edge (r ,s) to be removed is selected, the restricted 3-optimizationprocedure restricts the search for the other two edges to those nodes p belonging to edge (p,q) such as $\delta(r,q)<\delta(r,s)$. This project proposes an ant colony optimization algorithm for tuning generalization of fuzzy rule.

2. PROBLEM DOMAIN
2.1 Travelling Salesman Problem
The Travelling Salesman Problem (TSP) is a problem in combinatorial optimization studied in operations research and theoretical computer science. Given a list of cities and their pairwise distances, the task is to find a shortest possible tour that visits each city exactly once. The problem was first formulated as a mathematical problem in 1930 and is one of the most intensively studied problems in optimization. It is used as a benchmark for many optimization methods. Even

though the problem is computationally difficult, a large number of heuristics and exact methods are known, so that some instances with tens of thousands of cities can be solved.The TSP has several applications even in its purest formulation, such as planning, logistics, and the manufacture of microchips. Slightly modified, it appears as a sub-problem in many areas, such as genome sequencing. In these applications, the concept city represents, for example, customers, soldering points, or Deoxyribonucleic Acid (DNA) fragments, and the concept distance represents travelling times or cost, or a similarity measure between DNA fragments. In many applications, additional constraints such as limited resources or time windows make the problem considerably harder.

In the theory of computational complexity, the decision version of TSP belongs to the class of Nondeterministic Polynomial (NP)-complete problems. Thus, it is assumed that there is no efficient algorithm for solving TSPs. In other words, it is likely that the worst case running time for any algorithm for TSP increases exponentially with the number of cities, so even some instances with only hundreds of cities will take many years to solve exactly.

2.2 Ant Colony Optimization

The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. This algorithm is a member of ant colony algorithms family, in swarm intelligence methods, and it constitutes some metaheuristic optimizations. Initially proposed by Marco Dorigo in 1992 in his PhD thesis, the first algorithm was aiming to search for an optimal path in a graph; based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of Numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants. Ant Colony Optimization is a paradigm for designing meta heuristic algorithms for combinatorial optimization problems. The first algorithm which can be classified within this framework was presented in 1991 and, since then, many diverse variants of the basic principle have been reported in the literature. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions.

Ant colony optimization algorithms have been applied to many combinatorial optimization problems, ranging from quadratic assignment to fold protein or routing vehicles and a lot of derived methods have been adapted to dynamic problems in real variables, stochastic problems, multi-targets and parallel implementations. It has also been used to produce near-optimal solutions to the travelling salesman problem. They have an advantage over simulated annealing and genetic algorithm approaches of similar problems when the graph may change dynamically; the ant colony algorithm can be run continuously and adapt to changes in real time. This is of interest in network routing and urban transportation systems.

Ant colony optimization algorithms have been used to produce near-optimal solutions to the travelling salesman problem. The first ACO algorithm was called the Ant system (AS) and it was aimed to solve the travelling salesman problem, in which the goal is to find the shortest round-trip to link a series of cities. The general algorithm is relatively simple and based on a set of ants, each making one of the possible round-trips along the cities. At each stage, the ant chooses to move from one city to another according to some rules:

a. It must visit each city exactly once
b. A distant city has less chance of being chosen (the visibility)
c. The more intense the pheromone trail laid out on an edge between two cities, the greater the probability that that edge will be chosen
d. Having completed its journey, the ant deposits more pheromones on all edges it traversed, if the journey is short
e. After each iteration, trails of pheromones evaporate.
f. 3. Max-Min Ant System

The MAX–MIN Ant System (MMAS) algorithm achieves a strong exploitation of the search history by allowing only the best solutions to add pheromone during the pheromone trail update. Also, the use of a rather simple mechanism for limiting the strengths of the pheromone trails effectively avoids premature convergence of the search. Finally, MMAS can easily be extended by adding local search algorithms. In fact, the best performing ACO algorithms for many different combinatorial optimization problems improve the solutions generated by the ants with local search algorithms. As our empirical results show, MMAS is currently one of the best performing ACO algorithms for the TSP. One of the main ideas introduced by max–min Ant System, the utilization of pheromone trail limits to prevent premature convergence, can also be applied in a different way, which can be interpreted as a hybrid between MMAS and Ant Colony System (ACS).

MAX–MIN Ant System, which has been specifically developed to meet these requirements, differs in three key aspects from Ant System (AS):

a) To exploit the best solutions found during iteration or during the run of the algorithm, after each iteration only one single ant adds pheromone. This ant may be the one which found the best solution in the current iteration (iteration-best ant) or the one which found the best solution from the beginning of the trial (global-best ant).
b) To avoid stagnation of the search the range of possible pheromone trails on each solution component is limited to an interval [min, max].
c) Additionally, we deliberately initialize the pheromone trails to max, achieving in this way a higher exploration of solutions at the start of the algorithm.

3.1 Fuzzy Rule

Human beings make decisions based on rules. Although, we may not be aware of it, all the decisions we make are all based on computer like if-then statements. If the weather is fine, then we may decide to go out. If the forecast says the weather will be bad today, but fine tomorrow, then we make a decision not to go today, and postpone it till tomorrow. Rules associate ideas and relate one event to another. Fuzzy machines, which always tend to mimic the behavior of man, work the same way. However, the decision and the means of choosing that decision are replaced by fuzzy sets and the rules are replaced by fuzzy rules. Fuzzy rules also operate using a series of if-then statements. For instance, if X then A, if y then b, where A and B are all sets of X and Y. Fuzzy rules define fuzzy patches, which is the key idea in fuzzy logic.

Fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing. Fuzzy provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions.

A fuzzy rule is defined as a conditional statement in the form:

IF x is A, THEN y is B

Where x and y are linguistic variables; A and B are linguistic values determined by fuzzy sets on the universe of discourse X and Y, respectively. A fuzzy set is a class of objects with a continuum of grades of membership. Such a set is characterized by a membership (characteristic) function which assigns to each object a grade of membership ranging between zero and one. The notions of inclusion, union, intersection, complement, relation, convexity, etc., are extended to such sets, and various properties of these notions in the context of fuzzy sets are established.

The proposed Max-Min Ant System is an improved version of basic ant system applied to both symmetric and asymmetric instances of the Travelling Salesman Problem. The improved Max-Min ant system is augmented with a local search procedure, and produce better efficiency compared to ACS. In MMAS, the ant search is regarded as a combination of consequent values selected from every rule. A pheromone matrix among all candidate consequent values is constructed. Searching for the best one among all combinations of rule consequent values is based mainly on the pheromone matrix. The obtained fuzzy rules of TSP from Max-Min ant system provide two benefits. They are lower number of rules and their accuracy, improves with the increase in generalization being introduced and which provides an optimum solution.

4. MAX-MIN ANT FUZZY ON TSP – EXPERIMENTAL EVALUATION

The task in the TSP is to find a tour of minimum length through a given set of cities. A TSP can be represented by a complete weighted directed graph G=(V,A,d) with the set of nodes (cities) V={1,2,…n}, the set of paths A, $(i,j) \in V * V$, and the weight function d: A → IN, associating a positive integer weight dij with every path (i, j), interpretable as the distance between nodes i and j. The goal is to find a shortest cycle that visits each vertex in the graph exactly once. In the context of TSPs we will refer to such a cycle also as tour. In the symmetric TSP the distances between nodes are independent of the direction, i.e. for every pair of nodes holds dij = dji . In the more general asymmetric TSP atleast for one pair of node holds $dij \neq dji$. The TSP is a NP-hard optimization problem which has many applications.

Among the algorithmic approaches for the solution of TSPs is elaborate Branch and CUT algorithms that are able to solve very large instances of hundreds up to few thousand cities to optimality. Despite of the success of complete optimization algorithms still much interest is put on local search heuristics for the approximate solution of TSPs. Nowadays even more important is the fact that the TSP has become a standard test bed for algorithmic ideas. Apart from basic local search procedures, most metaheuristics algorithms are applied to TSP.

New algorithmic approaches should therefore be tested on this standard problem and a good performance on TSPs is taken as an indicator of the promise a new approach holds. This is one of the main reasons why the system applies Max-Min ant system to this problem. Another reason is that Ant system has been proposed at hand of the application to TSP and so we can compare Max-Min Ant system directly to Ant system. Furthermore, the system investigates the behavior of Max-Min Ant system and hope that some findings for the application to TSPs also carry over to other combinatorial optimization problems. The proposed scheme applied Max-Min Ant system to symmetric and asymmetric TSP. For all the problems we used, the optimal solution value is known and the results are presented most often as excess over the known optimal solution.

The performance of ant system can be enhanced by allowing only the best ant, update the trails in every cycle. Yet a disadvantage of this is the early stagnation of the search that makes further tour improvement impossible. When stagnation occurs, the trails on few paths grow so high that the ants will always construct the corresponding tour again and again. In MMAS the system only allow the best ant to update the trails. To alleviate the problem concerning early stagnation, the system introduces explicit maximum and minimum trial stagnation on the paths, hence the name MAX-MIN Ant System. The maximum and minimum trail limits are chosen in a problem dependent way depending on the average Path length .This way the influence of the trail intensities is limited.

As the system use as lower limit Tmin , the probability that a specific path is chosen may become very small, but will be still greater then zero. The trail limits alleviate the problem associated with the early stagnation of search especially for long runs, that leading to a higher degree of exploration. The trail strength in MMAS is initialization to Tmax for all paths. After each iteration, evaporation will reduce the trail strength by a factor p. Only the trail on paths participating in the best tours is allowed to increased their intensities or maintain them at a high level. Thus the trail strength on bad paths decreases slowly and only good paths can maintain a high level of trail strength and will therefore is selected more often by the ants. The performance of MMAS improves considerably over Ant System.

Despite of using maximum and minimum trail limits, long runs of the MMAS still can show stagnation behavior. If the mean 0.05-branching factor approaches very low values only few new tours are built, leading to very limited exploration of possible better. To avoid this, the system added the trail-smoothing mechanism. In case of stagnation of the search as indicated by mean branching factor, the system adjust the trail intensities according to a portion ally update, the trail intensity in increased proportional to the different between Tmax and the current trail intensity Tij(t) on the path (I,j)

Increase~ Tmax - Tij(t)

As an advantage of the proportional update, the system does not completely forget the trails learned so far. Its overall effect is that by increasing the trail intensities, the probability distribution for the selection of the exploration of new tours is higher. The system call this approach smoothing of the trail as the differences between high and low trail intensities become less pronounced, i.e. smoothing. With the smoothing approach the solution quality for longer runs increased significantly.

Resources	Latency/ runtime	Force Directed	List scheduling				Proposed MMAS (Average over 5 runs)				Existing ACO			
			IM	ID	LWID	SN	IM	ID	LWID	SN	IM	ID	LWID	SN
1a, 1fm, 1m, 3i, 3o	8/32	8	8	8	9	8	8	8	8	8	6	6	6	6
2a, 1fm, 2m	11/22	11	11	11	13	13	11	11	11	11	9	9	9	9
1a, 1fm, 1m	27/24000	28	28	31	31	28	27.2	27.2	27	27.2	25.8	25.8	25	25
2a, 2m, 3i, 3o	13/232	19	19	19	19	18	17.2	17.2	17	17.8	15.5	15.5	15	15.2
1a, 1fm, 1m, 3i, 3o	14/11560	19	19	21	21	21	16.2	16.4	16.2	17	14.8	14.6	14.8	14
2a, 2m, 1fm, 3i, 3o	-	18	19	20	18	18	17.4	18.2	17.6	17.6	15.6	16.8	15.4	15.4
2a, 2m, 1fm, 3i, 3o	-	23	23	23	23	23	21.2	21.2	21.2	21.2	19.8	19.8	19.8	19.8

Table 1: Evaluation Results

MAX-MIN Ant System algorithm

while time limit not reached do

for a = 0 to m - 1 do

{construction process of ant a}

C0 ← ϕ

for e = 0 to |E| - 1 do

Choose place p randomly from set P! places suitable for event e, according to probabilities probep for event e and place p

Ce ← Ce-1 ∪ {ep}

end for

C ← solution after applying local search algorithm to C|E|-1

Citeration best ← best of Cand Citeration best

end for

Cglobal best ← best of Citeration best and Cglobal best

global best or local best pheromone update (according to γ) for T using

Cglobal best, Tmin, and Tmax

end while

Where

|E| → a number of events in the set E (provided in the input file for each instance)

p - a place from the set P; p ∈ P

e → an event from the set E; e ∈ E, e < |E|

T → the pheromone matrix

C - Complete assignment of events into places, C: E → P

Tmin → minimal pheromone level

Tmax → maximal pheromone level

The basic mode of operation of the MAX-MIN Ant System is as follows. At each iteration of the algorithm, each of the ants constructs a complete assignment C of events into paths.

Following a pre-ordered list of events, the ants choose the path for the given event probabilistically, guided by stigmergic information. This information is in the form of a matrix of pheromone values. TSP problem specific knowledge (Fuzzy rule information) is also used by the algorithm. The path for an event is chosen only from the ones that are suitable for the given event, placing the event that will not violate any hard constraint. If, at some point of time during the construction of the assignment, there is no such a place available, a list of paths are extended by one, and the event is placed in one of the paths of this additional paths. This of course results in an infeasible solution as number of timeslots used from now on exceeds route number. This also means that pheromone matrix has to be extended as well.

Once all the ants have constructed their assignment of events into paths, a local search routine is used to further improve the solutions. Finally the best solution of all iteration is compared to the global best solution found so far. Only the better of the two is kept as the new global best. If the differences between extreme pheromone values were too large, all ants would almost always generate the same solutions, which would mean algorithm stagnation. The MAX-MIN Ant System introduces upper and lower limits on the pheromone values max and min respectively that prevent this. The maximal difference between the extreme levels of pheromone may be controlled, and thus the search intensification versus diversification may be balanced. The pheromone table T is updated either by the best assignment of a given iteration (i.e. local best), or by the global best assignment. We probabilistically choose which one to use. The local best update is chosen with probability proportional to its quality compared to the quality of the global best solution, and also the exploration rate.

5. TSP MAX_MIN ANT FUZZY PERFORMANCE EVALUATION

The proposed MMAS algorithm was applied to the identification of the analytical function using equally distributed fuzzy partitions with triangular membership functions for all input and output fuzzy domains and with five sets each of the TSP. The experiment consisted on the identification of the system with randomly generated training sets with three different sizes (10, 20 and 50 examples), and the subsequent run of the proposed MMAS algorithm over the identified fuzzy model. In this respect, the original Ant System version was implemented, which applies a random proportional rule for selecting each step and whose pheromone deposit mechanism is run once the solution is completed.

Two measures were used in order to analyze both the generality and accuracy results of the method. On the one hand, the number of rules describing the model together with the averaged complexity of the premises was considered for evaluating its generality. On the other hand, the normalized mean square error between the model and the system output was obtained for accuracy evaluation. The experiment was run 10 times for each training set size and the average results were obtained. It can be observed that the generalization capability of the MMAS algorithm is better than the one provided by the Ant Colony Optimization system. It shows the average generality of the fuzzy models expressed as the average number of rules describing them and, the average complexity of the premises in the rules.

(Resource Labels: a=alu, fm=faster multiplier, m=multiplier, i=input, o=output)

(Heuristic Labels: IM=Instruction Mobility ID=Instruction Depth, LWID=Latency Weighted Instruction Depth, SN=Successor Number)

Again, the MMAS algorithm provides fuzzy models described with a lower number of rules when compared with the initial fuzzy models and these rules had also have a low complexity. In addition, practically no case needed the maximum number of cycles, NCmax, to obtain the best solution and, in average; about 8 cycles were enough which proves the solidarity of the MMAS system compared to the Ant colony optimization system. Table 1 summarizes the experiment results.

Compared with the variety of list scheduling and the force-directed scheduling method, the MMAS algorithm generates better results consistently over all testing cases. For some of the testing samples, it provides significant improvement on the schedule latency. The biggest saving achieved is 23%. This is obtained when LWID is used as the local heuristic for our algorithm and also as the heuristic for constructing the priority list for the traditional list scheduler.

Though the results of force-directed scheduler are generally superior to that of the list scheduler, our algorithm achieves even better results. On average, comparing with the force-directed approach, our algorithm provides a 6.2% performance enhancement for the testing cases, while performance improvement for individual test sample can be as much as 14.7%. Finally, compared with the optimal scheduling results computed by using the integer linear programming model, the results generated by the proposed algorithm are much closer to the optimal than those provided by the list scheduling heuristics and the force directed approach. The MMAS algorithm improves the average schedule latency by 44% comparing with the list scheduling heuristics.

The performance of traditional list scheduler heavily depends on the input. This is echoed by the data in Table 1. Meantime, it is easy to observe that the proposed algorithm is much less sensitive to the choice of different local heuristics and input applications. This is evidenced by the fact that the standard deviation of the results achieved by the new algorithm is much smaller than that of the traditional list scheduler. Based on the data shown in Table 1, the average standard deviation for list scheduler over all and different heuristic choices is 0.8128, while that for the MMAS algorithm is only 0.1673. In other words, user can expect to achieve much more stable scheduling results on different application regardless the choice of local heuristic. This is a great attribute desired in practice.

6. CONCLUSION

In this work, MMAS algorithm has been proposed to increase the generality of the fuzzy rules by searching for its structure to be maximal. With this aim, the method searches good descriptions by means of compound rules of fuzzy models initially expressed with conventional single rules. The construction graph allows representing each solution as a sequential addition of labels to the premises in the antecedent, and from the cooperative behavior of the ants good combinations of compound rules emerge.

The proposal have shown that MMAS is an interesting novel approach to parallel stochastic optimization of the TSP. MMAS has been shown to compare favorably with previous attempts to apply other heuristic algorithms like genetic algorithms, evolutionary programming, and simulated annealing. Nevertheless, competition on the TSP is very tough, and a combination of a constructive method which generates good starting solution with local search which takes these solutions to a local optimum seems to be the best strategy. The system has shown that MMAS is also a very good constructive heuristic to provide such starting solutions for local optimizers.

The MMAS model presents an exponential pheromone deposition approach to improve the performance of classical ant system algorithm which employs uniform deposition rule. A simplified analysis using differential equations is carried out to study the stability of basic ant system dynamics with both exponential and constant deposition rules. A roadmap of connected cities, where the shortest path between two specified cities are to be found out, is taken as a platform to compare max-min ant system model (an improved and popular model of ant system algorithm) with exponential and constant deposition rules. Extensive simulations are performed to find the best parameter settings for non-uniform deposition approach and experiments with these parameter settings revealed that the above approach outstripped the traditional one by a large extent in terms of both solution quality and convergence time.

7. REFERENCES

[1] S. Abe and M. S. Lan, "Fuzzy rules extraction directly from numerical data for function approximation," IEEE Trans. Syst. Man Cybern., vol. 25. no. 1. pp. 119-129, 2005.

[2] J.Casillas, O. Cordon, and F. Herrera, "COR: A methodology to improve ad hoc data-driven linguistic rule learning methods by inducing cooperation among rules", IEEE Trans. Syst. Man Cybern. Part B-Cyber., vol. 32, no. 4, pp. 526-537, 2002.

[3] J. L. Castro, J. J. Castro-Schez, and J. M. Zurita, "Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems", Fuzzy Sets Syst., vol. 101, pp. 331-342, 1999.

[4] M. Dorigo, A. Colorni, and V. Maniezzo, "The ant system: Optimization by a colony of cooperating agents", IEEE Trans. Syst. Man Cyber. B, vol. 26. no, pp. 29-41, 1996.

[5] M.Dorigo and L. M. Gambardella, "Ant colony systems: A cooperative learning approach to the travelling salesman problem", IEEE Trans. Evol. Comput., vol. 1. pp. 53-66, 1997.

[6] M. Dorigo and T. Stuzle, Ant Colony Optimization. MIT Press, 2004.

[7] Freisleben, B. Merz, "A genetic local search algorithm for solving symmetric and asymmetric travelling salesman problems", May 1996. pp. 616—621

[8] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms", IEEE Trans. Fuzzy Syst., vol. 3, no. 3, pp. 260-270, 2005.

[9] Lin, Shen, Kernighan, B. W. "An Effective Heuristic Algorithm for the Travelling-Salesman Problem", Vol. 21, No. 2. (1973), pp. 498-516.

[10] Pablo Carmona, Juan Luis Castro, "Using Ant Colony Optimization for Learning Maximal Structure Fuzzy Rules", 2005 IEEE International Conference on Euzzy Systems.

[11] A. Piegat, "Fuzzy Modeling and Control", Heidelberg, Germany: Physical-Verlag, 2001.

[12] T. Stutzle and H. H. Hoos, "MAX-MIN ant system", Futur Gener. Compu. Syst., Vol. 16. no. 8. pp. 889-914, 2000.

[13] L.X. Wang and J. M. Mendel, "Generating fuzzy rules by learning from examples", IEEE Trans. Syst. Man Cyber., vol. 22, no. 6, pp. 1415-1427, 1992.

8. AUTHORS PROFILE

K.Sankar is a Research Scholar at the Anna University Coimbatore. He is now working as a Senior Lecturer at KSR College of Engineering, Tiruchengode. His Research interests are in the field of Data Mining and Optimization Techniques.

Dr.V.Venkatachalam is a principal of The Kavery Engineering College. He received his B.E in Electronics and Communication at Coimbatore Institute of Technology Coimbatore. He obtained his M.S degree in Software systems from Birla Institute of Technology Pilani. He did his M.Tech in Computer Science at Regional Engineering College (REC) Trichi. He obtained his P.hD degree in Computer Science and Engineering from Anna University Chennai. He has published 3 papers in International Journal and 20 papers in International & National conferences.