International Journal of Computer Applications (0975 – 8887)
Volume 97– No.10, July 2014

Modified Ant Colony Optimization Algorithm for Travelling Salesman Problem

Rohit Chaturvedi
Department of Computer Science & Engineering
Indian School of Mines, Dhanbad
Dhanbad 826004, India

Haider Banka
Department of Computer Science & Engineering
Indian School of Mines, Dhanbad
Dhanbad 826004, India

ABSTRACT
Limited amount of time and computational resources in industrial domain makes Ant Colony Optimization (ACO) a useful approach to find near optimal solutions in polynomial time for Nondeterministic Polynomial time (NP) problems. For dynamically changing graphs, such as in case of network routing and urban transportation systems which are based on Travelling Salesman Problem (TSP), the ant colony algorithm is superior to simulated annealing and genetic algorithm approaches as it can be run continuously and acclimatize to changes in real time. The objective of this paper is to find a competent method which improves ACO in terms of iteration count and ability to find better solutions for TSP so that it can be used in different areas like industrial and educational, for solving NP problems more efficiently. This paper proposes a modified ant colony optimization (MACO) algorithm which uses the peculiarity of Elitist Ant System (EAS) and Ant Colony System (ACS). Using EAS property, the convergence speed is optimized by additional pheromone deposition on the arcs of the best tour and pseudorandom proportional rule and local pheromone update of ACS tunes the degree of exploration and prevents the algorithm from stagnation. The experiments done on benchmark datasets from TSPLIB manifest clearly that MACO has an upper hand in terms of performance on conventional ACO, ACO-GA and ACO-PSO.

General Terms
Population-based algorithms, pseudorandom proportional rule, pheromone update

Keywords
Ant Colony Optimization (ACO), Travelling Salesman Problem (TSP), Modified Ant Colony Optimization (MACO), Swarm Intelligence (SI).

1. INTRODUCTION
For the last 10 years, a lot of population-based algorithms [4], [5] had been proposed. Jointly these algorithms are referred to as swarm intelligence (SI) [11], [21]. As one of the competent methods of SI, ACO algorithms proposed by Dorigo [1], [8], [9] has transpired as a stochastic optimization approach, which is modeled by simulating real ant colonies. Real ants deposit a chemical substance called pheromone on the ground. A path predilection an ant makes depends on this substance: the larger the amount of pheromone deposited on a particular path, the greater the probability an ant selects that path. The ACO has been applied effectively to several combinational optimization problems, [5], [10], [6], [12], [14], [16], [17] such as traveling salesman problem (TSP), quadratic assignment problems, intrusion detection system, job-shop scheduling problem, and adaptive routing.

ACO algorithms being an apposite computational approach are used by researchers in many areas. But due to some precincts of the ACO such as long searching time and the tendency to be trapped into local optima, ACO is not taken utterly by all. In the present paper, the ACO algorithm is modified to take up the discrete combinatorial optimization problem of TSP and remove the above setbacks. The results are compared to those obtained by conventional ACO. It has been confirmed that the modified ACO is a useful tool for solving TSP, which converges quickly toward the optimal solution.

The paper is organized as follows. Section 2 describes TSP and gives a brief introduction to the Ant Colony Optimization and its variants. Section 3 presents the proposed algorithm and explains how the proposed algorithm is designed. The experimental results, data sets and parameter settings are discussed in Section 4. We conclude our work in Section 5.

2. PRELIMINARIES
2.1 Travelling Salesman Problem
Travelling salesman problem (TSP) is a problem in which a salesman tries to find the shortest route to visit each given city precisely once. This NP complex problem is one of the extensively studied combinatorial optimization problems.

Alternatively, it can be defined as, given a complete weighted graph G = (N,A) with N being the set of nodes representing the cities, and A being the set of arcs and each arc $(i,j) \epsilon A$ is assigned a value (length) d_{ij}, which is the distance between cities i and j, with $i,j \epsilon N$, find a minimum length Hamiltonian circuit of the graph, where a Hamiltonian circuit is a closed path visiting each of the $n = |N|$ nodes of G exactly once [18].

2.2 Ant Colony Optimization Algorithms
In all existing ACO algorithms for the TSP, τ_{ij} refers to the attractiveness of visiting city j directly after city i as in all of them, the pheromone trails are related with the arcs. The heuristic information is defined as $\eta_{ij} = 1/ d_{ij}$, that is, the attractiveness of going from city i directly to city j is inversely proportional to the distance between the two cities [13], [15], [18], [19], [20].

2.2.1 Ant System (AS)
Ant-density, Ant-quantity, and Ant-cycle were the three different versions of AS proposed at the beginning. In the ant-density and ant-quantity versions, pheromone was updated straight away after a move from one city to adjacent city by the ants whereas in the ant-cycle version the pheromone update was done after the construction of the tour by the ants and the amount of pheromone deposited by each ant was directly proportional to the quality of the tour, that is the

shorter tour paths got more pheromone deposition then the longer ones. At present, ant-cycle is called AS as the two other variants were discarded because of their inferior performance [4].

2.2.1.1 Tour Construction

At first, m ants are put on arbitrarily chosen cities. At each construction step, ant k applies a probabilistic action choice rule, called random proportional rule, to decide the next city to be visited. The probability with which ant k, currently at city i, selects to go to city j is given by:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{ij}]^\alpha [\eta_{ij}]^\beta}, \quad if j \in N_i^k$$

where α and β are two parameters which decide the respective impact of the pheromone trail and the heuristic information, and N_i^k is the possible vicinity of ant k when being at city i, that is, the set of cities that ant k has not visited yet (the probability of selecting a city outside N_i^k is 0).

2.2.1.2 Update of Pheromone Trails

After the tour construction is completed by all the ants, the pheromone trails are updated. This is done by first lowering the pheromone value on all arcs by a constant factor, and then adding pheromone on the arcs that the ants have crossed in their tours. Pheromone evaporation is implemented by:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij}, \quad \forall (i,j) \epsilon L$$

where $0 < \rho \leq 1$ is the pheromone evaporation rate. After evaporation, all ants deposit pheromone on the arcs they have crossed in their tour:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^k, \quad \forall (i,j) \in L$$

where $\Delta\tau_{ij}^k$ is the amount of pheromone ant k deposits on the arcs it has visited. It is defined as follows:

$$\Delta\tau_{ij}^k = \begin{cases} 1/C^k, & if\ arc\ (i,j)\ belongs\ to\ T^k \\ 0, & otherwise \end{cases}$$

where C^k, the length of the tour T^k built by the k^{th} ant, is computed as the sum of the lengths of the arcs belonging to T^k.

2.2.2 Elitist Ant System (EAS)

It was the first improvement on the AS, the thought is to supply strong additional enrichment to the arcs belonging to the best tour found since the start of the algorithm; this tour is denoted as T^{bs} (best-so-far tour).

2.2.2.1 Update of Pheromone Trails

The additional enrichment of tour T^{bs} is attained by adding a quantity e/C^{bs} to its arcs, where e is a parameter that defines the weight given to the best-so-far tour T^{bs}, and C^{bs} is its length [18]. Hence, the equation for the pheromone deposit becomes:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^k + e\Delta\tau_{ij}^{bs}$$

where $\Delta\tau_{ij}^k$ is the amount of pheromone ant k deposits on the arcs it has visited and $\Delta\tau_{ij}^{bs}$ is defined as follows:

$$\Delta\tau_{ij}^{bs} = \begin{cases} 1/C^{bs}, & if\ arc(i,j)\ belongs\ to\ T^{bs} \\ 0, & otherwise \end{cases}$$

2.2.3 Ant Colony System (ACS)

There are three major differences between ACS and AS. First, it utilizes the exploration understanding gathered by the ants more strongly than AS through the use of a more belligerent action choice rule. Second, only the arcs of the best-so-far tour are privileged with pheromone evaporation and pheromone deposit processes. Third, to increase the exploration of surrogate paths, some pheromone is removed from the arc (i,j) used by the ant to move from city i to city j [5], [7].

2.2.3.1 Tour Construction

In ACS, pseudorandom proportional rule is used by an ant k in selecting city j as the city to move next when being at city i. This rule is defined as:

$$j = \begin{cases} argmax_{l \in N_i^k} \left\{ \tau_{il} [\eta_{ij}]^\beta \right\}, & if\ q \leq q_o \\ J, & otherwise \end{cases}$$

where q is a random variable homogeneously strewn in [0,1], q_o ($0 \leq q_0 \leq 1$) is a parameter, and J is a random variable chosen according to the probability distribution equation (with α = 1). The ant makes the best possible move with probability q_0 as stipulated by the erudite pheromone trails and the heuristic information (in this case, the ant is utilizing the learned knowledge); while with probability ($1-q_0$) it performs a prejudiced exploration of the arcs. Alteration in the parameter q_0 allows intonation of the extent of investigation and the choice of whether to contemplate the search of the system around the best-so-far solution or to explore other tours.

2.2.3.2 Global Pheromone Trail Update

Only the best-so-far ant is permitted to add pheromone after each iteration in ACS. Hence, the update in ACS is enacted by the following equation:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs}, \quad \forall (i,j) \in T^{bs}$$

where $\Delta\tau_{ij}^{bs}$ = $1/C^{bs}$. In ACS, it is noteworthy that the pheromone trail update, that is both evaporation and new pheromone deposit, only appertains to the arcs of T^{bs}, not to all the arcs as in AS. This is major difference between ACS and AS because by using this method the computational complexity of the pheromone update at each iteration is reduced from $O(n^2)$ to $O(n)$, where n is the size of the instance being solved.

2.2.3.3 Local Pheromone Trail Update

In ACS, a local pheromone update rule used by the ants in addition to the global pheromone trail updating rule, that they apply immediately after having crossed an arc (i,j) at the time of tour construction:

$$\tau_{ij} \leftarrow (1-\xi)\tau_{ij} + \xi\tau_o$$

where ξ, $0 < \xi < 1$, and τ_o are two parameters. The value of τ_o is set equal to the initial value for the pheromone trails. On the basis of experiments done, a good value for ξ was found to be

0.1, while a good value for τ_o was found to be $1/nC^{nn}$, where n is the number of cities in the TSP instance and C^{nn} is the length of a nearest-neighbor tour. The consequence of the local updating rule is that each time an ant uses an arc (i,j) its pheromone trail τ_{ij} is abridged, so that the arc becomes less pleasing for the subsequent ants. This sanctions an increase in the discovery of arcs that have not been visited yet and, in practice, has the outcome that the algorithm does not show a stagnation behavior.

3. PROPOSED WORK

ACO qualities like very good search capability for optimization problems are overshadowed by its drawback like taking too much time to converge and trapping in local optima in order to find an optimal solution for TSP problems [23]. After the modification of the conventional ACO algorithm in the proposed work, these problems are minimized.

3.1 Modified ACO Algorithm

Conventional ACO algorithm give the elementary structure for the design of MACO algorithm (refer Fig. 1). Properties of Elitist Ant System and Ant Colony System have been used to overcome the problems faced by conventional ACO algorithm. There are three main modifications in this algorithm. Firstly, pseudorandom proportional rule is used to select the next node an ant should travel which is used in Ant Colony System. Second, in order to remove the stagnation behavior of the algorithm and to increase its exploration for the arcs that have not been visited yet, a local pheromone update is done. Third, the best tour found from the start of the execution of algorithm is provided with additional pheromone deposition so as to increase its convergence speed which is used in Elitist Ant System.

1. **I**mport dataset from **TSPLIB**
2. **F**ind the no. of cities in dataset and store it in a variable *Ncity*
3. **M**ake no. of ants equal to ***N**city* and store it in ***N**ants*
4. **F**ind the coordinates of ***N**city* and store it in ***x**city* and *ycity*
5. **F**ind distance between cities and store it in ***d**city*
6. **F**ind the inverse of distance between cities and store it in ***v**is*
7. **I**nitialize the pheromone matrix of size ***N**city* * ***N**city* to 0.1 and store it in ***p**hmone*
8. **S**et maximum iteration denoted by ***m**axit* to 400 and factors $\boldsymbol{\alpha} = 1$, $\boldsymbol{\beta} = 5$ and evaporation coefficient $\boldsymbol{\rho} = 0.1$
9. **S**et elitist ant factor ***e*** equal to ***N**city* and best distance denoted by ***d**best* to 9999999
10. **I**nitialize the tour matrix of size ***N**ants* * ***N**city* and denoted by ***t**our* to random courses
11. **for *i**t*** = 1 **to *m**axit*** **do**
12. **for *i**a*** = 1 **to** ***N**ants*** **do**
13. **for *i**q*** = 2 **to** ***N**city*-1 **do**
14. **S**tore the starting city of each ant in ***st***
15. **S**tore the next cities which are not yet visited for each ant in variable ***n**xt*
16. **C**alculate the probabilities from ***st*** to ***n**xt* and store it in variable ***p**rob*
17. **C**hoose the next city according to the **P**seudorandom **P**roportional **R**ule used in ACS and store it in variable ***n**ewcity*
18. **M**odify the tour matrix of ach ant denoted by ***t**our*
19. **end for**
20. **end for**
21. **I**nitialize a temporary pheromone matrix denoted by ***p**htemp* and of size ***N**city* * ***N**city* to 0
22. **for *i**c*** = 1 **to *N**ants*** **do**
23. **I**nitialize a distance matrix denoted by ***d**ist* and which stores the tour distance of each ant to 0
24. **for *i**d*** = 1 **to *N**city*** **do**
25. **C**alculate the distance travelled by each ant upto current city and store it in ***d**ist*
26. **U**pdate ***p**htemp* for each ant acoording to the value of ***d**ist*
27. **end for**
28. **end for**
29. **C**alculate the minimum length of a tour travelled by elitist ant and store it in ***d**best*
30. **I**nitialize a temporary pheromone matrix denoted by ***p**h1* and of size ***N**city* * ***N**city* to 0
31. **U**pdate ***p**h* according to ***d**best* and multiply elitist factor ***e*** to ***p**h* and store the product in ***p**h*
32. **E**vaporate pheromone from ***p**hmone* using evaporation coefficient $\boldsymbol{\rho}$
33. **A**dd ***p**htemp* and ***p**h* to ***p**hmone* and store the sum in ***p**hmone*
34. **end for**
35. **P**lot**T**he**G**raphs**O**f**O**verall**B**est**A**nt()

A TSP instance is given as the coordinates of a number of n points. Number of cities is calculated from the TSP instance and stored in a variable (Ncity). Number of ants is taken equal to the number of cities and stored in a variable (Nants). The x and y coordinates of the cities are stored in two arrays (xcity and ycity) and all intercity distances are precompiled and stored in a distance matrix (dcity) with n^2 entries. The heuristic information η_{ij} is inversely proportional to the distance between cities i and j, a straightforward choice being $\eta_{ij} = 1/dcity_{ij}$ and this information is stored in visibility matrix (vis). Pheromone matrix (phmone) is a square matrix of size equal to number of cities and is initialized to 0.1. The pheromone trails τ_{ij} in the TSP refer to the desirability of visiting city j directly after i. Maximum iteration (maxit) is set to 400. Parameters α, β, ρ and e are set to 1, 5, 0.1 and number of cities (Ncity) respectively. The global best tour distance is stored in a variable (dbest). A square tour matrix (tour) of size Nants * Ncity is initialized to random tours which gives the tour information of each ant after every iteration.

When constructing a tour, an ant located on city i chooses the next city j according to the pseudorandom proportional rule. Then a local pheromone deposition is done according to the tours travelled by the ants. At last, additional pheromone is deposited at the arcs of the best tour and pheromone is evaporated according to the pheromone evaporation rate. This process is repeated for maximum iteration which is set to 400.

4. EXPERIMENTS AND RESULTS

HP Compaq dc7800p Convertible Minitower machine with 2.33 GHz Intel(R) Core(TM)2 Duo CPU and 1GB of memory using Microsoft Windows XP Professional was used for the pragmatic analysis of the algorithm. The program was written in MATLAB using MATLAB R2008a. The algorithm was tested using several TSP problems which were taken from the

TSPLIB website [3] so as to proof the dominance of the MACO algorithm on existing ACO algorithms. We compared our proposed algorithm results with those of the conventional ACO, ACO-GA and ACO-PSO algorithms in the aspects of algorithm convergence and solution quality, that is the number of iterations taken by the algorithm and the propinquity to the best fitness value. The results of ACO, ACO-GA and ACO-PSO for benchmark datasets were taken from a paper [22].

Table 1.The best fitness (best total distance) attained by each algorithm compared to the optimal solution

Methods \ TSP Problems		bays29	berlin52	kroA100	ch150
ACO	best fitness	2020	7548	22939	7109
	accuracy (%)	100	99.92	91.10	92.21
	no. of iterations	523	802	1533	4346
ACO-GA	best fitness	2020	7544	22274	6718
	accuracy (%)	100	99.97	95.33	97.08
	no. of iterations	518	6500	4500	7500
ACO-PSO	best fitness	2020	7544	22238	6689
	accuracy (%)	100	99.97	95.50	97.53
	no. of iterations	422	2000	3500	4000
Modified ACO	best fitness	**2022**	**7544.37**	**21494.03**	**6580.87**
	accuracy (%)	**99.90**	**99.97**	**99.00**	**99.19**
	no. of iterations	**31**	**149**	**335**	**321**
Optimal Solution		2020	7542	21282	6528

We used the problem of 29, 52, 100, 150 node data from TSPLIB [3] in the experiments performed on the MACO algorithm. Table 1 show the outcome which has been generated by 10 pragmatic experiments for Modified ACO algorithm with reference to four TSP problems as given in the table. Over the maximum number of 400 iterations, the global best total distance or best fitness of MACO is displayed in the table and compared with conventional ACO, ACO-GA and ACO-PSO. For bays29 and berlin52, it can be discerned that there is very slight difference in the fitness values of the algorithms. The data of kroA100 is more complex than ch150, hence even after having less number of nodes, its accuracy rate is less and iteration count is more for all the algorithms. It can be seen in table that MACO performs way beyond other algorithms for kroA100 and ch150 in terms of accuracy and number of iteration runs. These results construe that when the nodes and complexity of the graph increases, MACO performance is way beyond other ACO algorithms. This clearly elucidates the dominance of the proposed method over the existing ones.

5. CONCLUSION

This paper proffer how an improved ant colony algorithm can solve traveling salesman problem competently and thus help in taking up other NP complex problems without any hitch. The improvements centralize on pseudorandom proportional rule and local pheromone update used in Ant Colony System and additional pheromone deposition to the arcs belonging to the best tour used in Elitist Ant System. As delineated from the experimental results, the proposed system is more successful than the conventional ACO, ACO-GA and ACO-PSO algorithms in terms of convergence speed and the ability to find better solutions hence can be used in place of these algorithms for finding optimal solution for NP problems in polynomial time.

Even though traveling salesman problem (TSP) is used here, this algorithm can be applied to other optimization problems which occur in industrial and educational environments where computational resources and time are limited.

6. REFERENCES

[1] Colorni, A.; Dorigo, M.; Maniezzo, V. Distributed optimization by ant colonies. Proceedings of the 1st European Conference on Artificial Life; Varela, F., Bourgine, P., Eds.; Elsevier: Paris, France, 1991; pp 134-142.

[2] Reinelt, G. The Traveling Salesman: Computational Solutions for TSP Applications, Lecture Notes in Computer Science, Berlin: Springer, 1994, vol. 840.

[3] TSPLIB - A Travelling Salesman Problem Library. http://www.iwr.uni-heidelberg.de/groups/comopt/ software/TSPLIB95/.

[4] Dorigo, M., Maniezzo, V., and Colorni, A., The Ant System: Optimization by a Colony of Cooperating Agents, IEEE Trans. Systems, Man Cybernetics, Part B, 1996, vol. 26, no. 1, pp. 29–41.

[5] Dorigo, M., and Gambardella, L.M., Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, IEEE Trans. Evolutionary Computation, 1997, vol. 1, no. 1, pp. 53–66.

[6] Baue, A.; Bullnheimer, B.; Hartl, R. F.; et al. An ant colony optimization approach for the single machine total tardiness problem. CEC 99, Proceedings of the 1999 Congress on Evolutionary Computation;1999; Vol. 2, pp 1450-1456.

[7] Cassillas, J., Cordon, O., and Herrera, F., Learning Fuzzy Rules Using Ant Colony Optimization Algorithms, Proc. of ANTS2000—From Ant Colonies to Artificial Ants: A Series of Int. Workshops on Ant Algorithms, Brussels, 2000, pp. 13–21.

[8] Bonabeau, E.; Dorigo, M.; Theraulaz, G. Inspiration for optimization from social insect behavior.Nature2000, 406, 39-42.

[9] Krieger, M. J.; Billeter, J. B.; Keller, L. Ant-like task allocation and recruitment in cooperative robots.Nature2000, 406, 992-995.

[10] Wang, Y.; Xie, J. Y. Ant colony optimization for multicast routing in Circuits and Systems. IEEE: The 2000 IEEE Asia-Pacific Conference; 2000; pp 54-57.

[11] J. Kennedy, R. C. Eberhart, and Y. Shi, Swarm Intelligence. Morgan Kaufmann, 2001.

[12] Silva, A. R. M.; Ramalho, G. L. Ant system for the set covering problem Systems.2001 IEEE International Conference on Man and Cybernetics;2001; Vol. 5, pp 3129-3133.

[13] Dorigo, M. and Stutzle, T. The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances, in Handbook of Metaheuristics, Glover, F. and Kochenberger, G., Eds., Norwell: Kluwer, 2002.

[14] Xiong, W. Q.; Wei, P. A kind of ant colony algorithm for function optimization. Machine Learning and

Cybernetics. International Conference on Proceedings; 2002; Vol. 1, pp 552-555.

[15] Gutiahr, W.J., A Converging ACO Algorithm for Stochastic Combinatorial Optimization, Lecture Notes in Computer Science (Proc. of SAFA-2003 (Stochastic Algorithms: Foundations and Applications)), Berlin: Springer, 2003, no. 2827, pp. 10–25.

[16] Li, Y. M.; Xu, Z. B. An ant colony optimization heuristic for solving maximum independent set problems.Fifth International Conference on Computational Intelligence and Multimedia Applications; 2003; pp 206-211.

[17] Gomez, J. F.; Khodr, H. M.; DeOliveira, P. M.; et al. Ant Colony System Algorithm for the Planning of Primary Distribution Circuits Power Systems. IEEE Transactions on Power Systems2004, 19, 996-1004.

[18] Dorigo, M. and Stutzle, T., Ant Colony Optimization, Bradford Book, 2004.

[19] Gueret, C., Monmarche, N., and Slimane, M., Ants Can Play Music, Lecture Notes in Computer Science (Proc. of IV Int. Workshop on Ant Colony Optimization and Swarm Intelligence ANTS-2004), Berlin: Springer, 2004, no. 3172, pp. 310–317.

[20] Shtovba, S.D., Ant Algorithms: Theory and Applications, Programming and Computer Software, Vol. 31, No. 4, 2005, pp. 167-178.

[21] A. P. Engelbrecht, Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, 2006.

[22] Nonsiri, Sarayut, and Siriporn Supratid. "Modifying ant colony optimization." Soft Computing in Industrial Applications, 2008. SMCia'08. IEEE Conference on. IEEE, 2008.

[23] Hlaing, Z.C.S.S., and Khine, M.A., Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm, International Journal of Information and Education Technology, Vol. 1, No. 5, December 2011.

7. AUTHORS

Haider Banka is presently working as associate professor in the department of Computer Science & Engineering at Indian School of Mines, Dhanbad, India. He received his Ph.D. in Engineering from Jadavpur University, in 2008. During 2003–2004, he was a lecturer in Engineering College, Durgapur, India. He received his M.Sc. and M.Tech. degree in Computer Science from University of Calcutta, India, in 2001 and 2003, respectively. He has published many research papers in International and National Journals including various Trans. Of IEEE, Springer, Elsevier etc. He serves as a reviewer of several international journals. His research areas are Soft Computing, Bioinformatics, Design & Analysis of Algorithms, Data Mining, Pattern Recognition, Combinatorial Optimization etc.

Rohit Chaturvedi has completed his M.Tech. in Computer Application, department of Computer Science & Engineering from Indian School of Mines, Dhanbad. His research interest is mainly on Ant Colony Optimization, Design & Analysis of Algorithms, Pattern Recognition, and Combinatorial Optimization etc.