ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Finding Shortest Path in All Tours of a TSP Using Dynamic Candidate Set

Sonia Rajput[1], Nagendra Kumar [2]

P.G. Student, Department of Computer Engineering, Shri Ram Institute of Science & Technology, JBP, M.P, India.[1]

Assistant Professor, Department of Computer Engineering, Shri Ram Institute of Science & Technology, JBP, M.P, India.[2]

ABSTRACT: Although ant colony optimization (ACO) has successfully been applied to a wide range of optimization problems, its high time- and space-complexity prevent it to be applied to the large-scale instances. Furthermore, local search, used in ACO to increase its performance, is applied without using heuristic information stored in pheromone values. To overcome these problems, this thesis proposes new strategies including effective selection of pheromone values and heuristics, which will speed up ACO and enable it to be applied to large-scale instances.
In this paper, we introduce our new Dynamic pheromone update Rule based on Candidate set in ACO for travelling salesman problem. The TSP is to be done efficiently so that it helps in reducing the time for salespersons .There are different techniques that are proposed to solve the problems of TSP. Through this thesis we are going to present the new Algorithm that will find all the routes of salesperson as well as it will find the shortest path. A comparison of this new algorithm based TSP technique is compared with ACO and Improved ACO. A result shows that, proposed algorithm will outperform the existing techniques.

KEYWORDS: Shortest Path problem, TSP, ACO, Improved ACO, Dynamic Pheromone, Candidate set.

I. INTRODUCTION

The shortest path problem exists in variety of areas. A well known shortest path algorithm is Dijkstra's, also called "label algorithm". Shortest-path computation is a fundamental problem in computer science with applications in diverse areas such as transportation, robotics, network routing, and VLSI design. The problem is to find paths of minimum weight between pairs of nodes in edge weighted graphs.
There are two basic versions of the shortest-path problem: in the single-source shortest-path (SSSP) version, given a source node s, the goal is to find all distances between s and the other nodes of the graph; in the all-pairs shortest-path (APSP) version, the goal is to compute the distances between all pairs of nodes in the graph. While the SSSP problem can be solved very efficiently in nearly linear time by using Dijkstra algorithm [1], the APSP problem is much harder computationally.
A) Travelling Salesman Problem: Travelling salesman problem (TSP) is a combinatorial optimization problem, and is of great interest to computer scientists and mathematicians. In TSP a salesman begins from his hometown, visits each number of cities just ones and then returns to his home town. The TSP is one of the classes of NP-hard problem and it cannot be worked out optimally in a reasonable time when a large number of cities are involved.
Travelling salesman problem (TSP) is a theoretical mathematical model and has been widely applied in dynamic problems in reality. Most of the existing methods used to model TSP lack a realistic foundation, and cannot provide convenient and polytrophic operations to simulate real-world scenarios. The travelling salesman problem (TSP) [3] is a classical combinatorial optimization problem defined in an undirected graph G= {V, A, W}, where V= {1… N} is a set of nodes, $A= \{(i, j)|iÇj\}$ is a set of edges and W is a symmetrical weight matrix. The optimization objective of a TSP is to find the shortest Hamilton route on G. Owing to the usefulness in modelling a variety of real world computational problems; TSP applications have been extended to route scheduling [2], machine scheduling and sequencing [3], and some other practical computation problem [4]. Meanwhile, various methods have been proposed and successfully used in solving optimization problems in a static scenario. However, some real problems can be modeled as a TSP is not always static. When considering the situations, such as traffic control and congestion, the actual traversal route and the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

nodes needed to be visited are variable in different periods, leading to a dynamic TSP (TSP). Similar dynamic scenarios broadly exist in real-world problems, such as the scheduling of school bus for pickup and drop-off students [5] in the case of some students ask for a leave and the postman problem [6] etc.

B) Ant Colony Optimization: Ant colony optimization (ACO) is a kind of simulated evolutionary algorithm. It imitates ants' foraging process to find the shortest path, coexists with the characteristics of randomness and heuristic. It is applied successfully to solve combinatorial optimization problems, such as the TSP problem, the job-shop scheduling problem, etc. In the early 1990s, the Italy scholar Macro Dorigo proposed an ant colony algorithm by imitating ants' foraging process to find the shortest path [7]. Compared to other optimization methods, the ACO has a number of advantages which contribute to its wide use. Some of the key advantages of ACO include: (a) It does not need the calculation of derivatives.
(b) The knowledge of good solutions is retained by all ants.
(c) The ants in the colony share information.
Moreover, ACO is less sensitive to the nature of the objective function, and hence it can be used for various objective functions [8]. Then it is applied successfully to solve combinatorial optimization problems, such as the TSP problem, the job-shop scheduling problem, etc. TSP, simply speaking, is one salesman starts and ends at the same city, with all cities being visited once and only once, and the path length required as short as possible. TSP problem is a classic combinatorial optimization problem, and the study of TSP can be used for other reference. To solve TSP, the simplest method is enumerating all possible solutions, and obtain optimal path by comparing [9]. For n cities, the time complexity is $(n-1)!$, which is a large amount of engineering calculation. Because of ACO's numbers of advantages, it has played a critical role in solving TSP problem [10].
Ant colony optimization (ACO) is one of the most recent techniques for approximate optimization. The inspiring source of ACO algorithms are real ant colonies. More specifically, ACO is inspired by the ants' foraging behaviour. At the core of this behaviour is the indirect communication between the ants by means of chemical pheromone trails, which enables them to find short paths between their nest and food sources. This characteristic of real ant colonies is exploited in ACO algorithms in order to solve, for example, discrete optimization problems. Figure 1.1 a, b, c, d below shows some behaviour of ants.

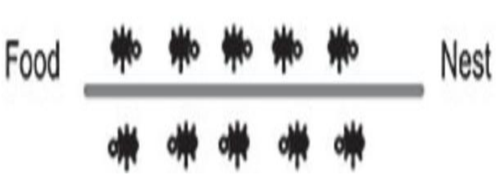
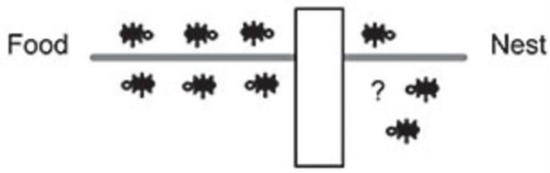
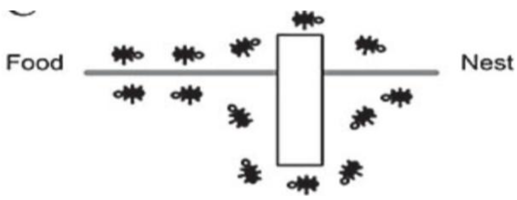


Figure: 1. a) Ants in a Pheromone Trail between nest and food.



b) An Obstacle interrupts the trail.



c) Ants find two paths to go around the obstacle.

d) A New Pheromone Trail is found along the shorter path.

Depending on the point of view, ACO algorithms may belong to different classes of approximate algorithms. Seen from the artificial intelligence (AI) perspective, ACO algorithms are one of the most successful strands of swarm intelligence. The goal of swarm intelligence is the design of intelligent multi-agent systems by taking inspiration from the collective behaviour of social insects such as ants, termites, bees, wasps, and other animal societies such as flocks of birds or fish schools. Examples of "swarm intelligent" algorithms other than ACO are those for clustering and data mining inspired by ants' cemetery building behaviour, those for dynamic task allocation inspired by the behaviour of wasp colonies, and particle swarm optimization. Seen from the operations research (OR) perspective, ACO algorithms belong to the class of metaheuristics. The term metaheuristics, first introduced in, derives from the composition of two

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Greek words. Heuristic derives from the verb heuriskein which means “to find”, while the suffix meta means “beyond, in an upper level”. Before this term was widely adopted, metaheuristics were often called modern heuristics. In addition to ACO, other algorithms such as evolutionary computation iterated local search, simulated annealing, and tabu search, are often regarded as metaheuristics.

C) Ant System for the TSP: In the TSP is given a completely connected, undirected graph G = (V, E) with edge-weights. The nodes V of this graph represent the cities, and the edge weights represent the distances between the cities. The goal is to find a closed path in G that contains each node exactly once (henceforth called a tour) and whose length is minimal. Thus, the search space S consists of all tours in G. The objective function value f (s) of a tour s ∈ S is defined as the sum of the edge-weights of the edges that are in s. The TSP can be modelled in many different ways as a discrete optimization problem. The most common model consists of a binary decision variable Xe for each edge in G. If in a solution Xe = 1, then edge e is part of the tour that is defined by the solution.

Concerning the AS approach, the edges of the given TSP graph can be considered solution components, i.e., for each $e_{i,j}$ is introduced a pheromone value $\tau_{i,j}$. The task of each ant consists in the construction of a feasible TSP solution, i.e., a feasible tour. In other words, the notion of task of an ant changes from “choosing a path from the nest to the food source” to “constructing a feasible solution to the tackled optimization problem”. Note that with this change of task, the notions of nest and food source lose their meaning. Each ant constructs a solution as follows. First, one of the nodes of the TSP graph is randomly chosen as start node. Then, the ant builds a tour in the TSP graph by moving in each construction step from its current node (i.e., the city in which she is located) to another node which she has not visited yet. At each step the traversed edge is added to the solution under construction. When no unvisited nodes are left the ant closes the tour by moving from her current node to the node in which she started the solution construction. This way of constructing a solution implies that an ant has a memory T to store the already visited nodes.

II. RELATED WORK

Travelling Salesman Problem (TSP) is a well known NP-hard problem in combinatorial optimization which was first studied in the 1930s by Karl Menger in Vienna and Harvard. Dynamic Travelling Salesman Problem is an extension of the traditional TSP which involves dynamic-cost allocation between the cities. This may include, but is not limited to, inclusion and exclusion of certain cities at any given time. It was proposed by Psaraftis [11] in 1988. DTSP is widely applicable in real-time scenarios than TSP but is arguably more difficult to solve. Usually, the algorithms used to solve Static TSP turn out to be inefficient for DTSP.

Yan Cao and Jun Zhang presented Rs-ACS algorithm to solve Travelling Salesman Problem using Pheromone matrix update rule [12]. The proposed model contains two environmental changes, weight change and dimension change. In dimension change some nodes are removed temporarily from the solution space because the salesman did not want to visit the nodes in future.

Recently, biogeography-based optimization algorithms are interested by researchers to find approximate solutions for optimization problems. BBG is a new heuristic algorithm proposed by D. Simon [13] in 2008. The BBG algorithm draws inspiration from the mechanism of species mutation and migration and is employed to solve the optimization problems. The solutions of optimization problems or "islands" in BBG are analogous to the habitat in biogeography. The High Suitability Index (HSI) is used for evaluating the goodness of the islands. Which mean that the habitat having a high HSI va1ue is considered a good is1and and the best is1ands are is1ands having the most species? The features of a habitat are called Suitability Index Variables (SIVs).

The mutation and migration operators are very important and strongly affect the algorithm's performance in BBG. With these operators, problem solutions can communicate with one another. Thus, the problem solutions in BBG algorithm have higher flexibility in terms of individual communication in comparison with some other classica1 meta-heuristic methods and also show a better performance when compared with other evolutionary algorithms [14].

Huynh Thi Thanh Binh and Pham Dinh Thanh Presented Biogeography-based on optimization algorithm (BBO) is a new evolution algorithm inspired by the science of biogeography and designed based on the migration strategy of animals [15]. According to investigations and analysis on this algorithm, BBO has great success in numerous optimization problems and it has been used in different types of applications. In BBO, the migration operator is an important operator which is able to efficiently share the good information among solutions.

In the work proposed by Fozia Haneef Khan in Solving TSP [16]. The main purpose of this study is to propose a new representation method of chromosomes using binary matrix and new fittest criteria to be used as method for finding the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

optimal solution for TSP. The concept of the proposed method is taken from genetic algorithm of artificial inelegance as a basic ingredient which has been used as search algorithm to find the near-optimal solutions. We introduces additional improvements, providing an algorithm for symmetric as well as Asymmetric TSP, here we are implementing the new fittest criteria as well as new representation of asymmetric matrix and improving our solution by applying the crossover and mutation again and again in order to get the optimal solution.

Ant Colony Optimization (ACO) algorithm is a meta-heuristic inspired by the foraging behaviour of real ants. ACO uses "specially defined" artificial ants, which move through the solution space within a certain nearest-neighbour scope by following pheromone trails and heuristic information. The pheromone trails accumulate the collective knowledge about the quality of the solutions that the ants have found by performing the algorithm. The use of the heuristic information, which is not observed in the foraging behaviour of real ants, enables the ants to have visual capability. Figure2 below shows the characteristics of ants with their pheromone behaviour.

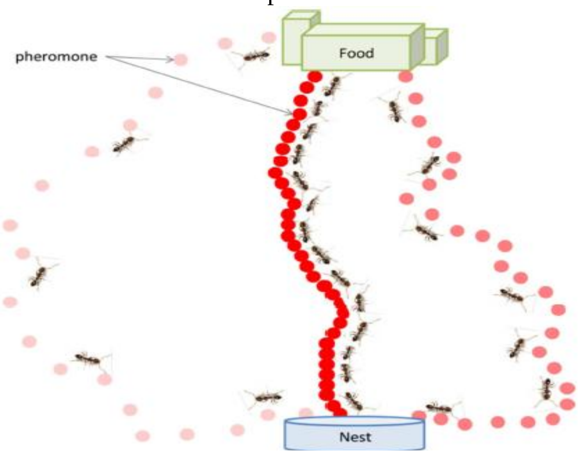


Figure 2: Pheromone Behaviour of Ants.

In figure 2 the solid red color lines shows the strong pheromone with more number of ants following the path. The lighter ones are followed by less number of ants. Stronger the pheromone value, easier to find the next move of ants. This paper focus on the pheromone updating characteristics to reduce the time and distance of salesman.

ACO algorithms have successfully been applied to many optimization problems. These problems include combinatorial optimizations such as versions of scheduling problems, finding edge-disjoint paths problem, types of quadratic assignment problem, QoS-based service selection problems, types of vehicle routing problem, types of travelling salesman problem and knapsack problem. The ACO has also applications in mobile networks, community mining in social networks, solving clustering problems, finger print matching and many other problems. ACO algorithms also have successfully been applied to continuous optimization problems. A previous work proposes a type of ACO that is applied to continue function optimization by archiving solutions and utilizing a type of local search algorithm in its baseline algorithm. Ant algorithms are easy to implement and cover wide range of applications, but their performance dramatically decreases in dealing with large-scale problems.

III. Proposed Algorithm

In this paper we propose a new algorithm to find shortest path in Travelling Salesman problem by enhancing Ant Colony Optimization Algorithm by the study on Pheromone factor. The pheromone factor has been enhanced by using Dynamic Candidate Set property.

A) **Limitations of Existing System:** Following are some of the limitations found in existing modified ACO algorithm.

1) Ants are guided by Random Factor, but its value is not defined properly.
2) Roulette algorithm is used to find the better route. This algorithm works on the basis of "Luck by Chance" method.
3) The argmax algorithm mentioned in section 3 of [17] traps in "Local Optima". I.e. it always works for shorter value between i and j.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

To overcome the above limitations of [17], we propose a new algorithm which is based on Candidate set generation that helps in selecting next move by the salesman. First the mathematical model of existing Ant Colony is described. Then the proposed method will be explained in detail in coming sections.

B) Proposed Algorithm:

Symbols	Meaning
m	The number of ants in the algorithm
η	The number of cities to be visited
$d_{i,j}$	Distance between city i to j.
$\eta_{i,j}$	Expectation from city i to j.
p	Pheromone evaporation rate
$\tau^{k}_{,i,j}(t+1)$	Pheromone concentration from city i to j at time (t+1)
$\Delta\tau^{k}_{,i,j}(t+1)$	Pheromone concentration increment from city i to j

Table 1: Mathematical model of the proposed method

(i) Ants Colony Algorithm Model:

For the sake of convenience, we list all of the mathematical symbols used in this article in Table 1.

According to different pheromone update rule, there are three different models of $\Delta\tau^{k}_{ij}(t+1)$ formulated by Dorigo, called ant-quantity system, ant-density system and ant-cycle system, separately. Number of experiments show that ant-cycle system performs better than the others, for it considers the global feature [15]. In ant-cycle system,

$$\Delta\boldsymbol{\tau}^{k}_{ij}(t+1)= \begin{cases} Q/L_K, & k^{th} \text{ ant passes route} \\ 0, & \text{otherwise.} \end{cases} \quad ....(1)$$

in which, Q is the amount of pheromone ants released, Lk is the length of path the kth ant visited.

We choose ant-cycle system in this thesis. At time t, k^{th} ant follows the transition probability $p^{k}_{ij}(t)$ to choose next city j to visit,

$$P^{k}_{ij}(t) = \{ \frac{\tau^{\alpha}_{ij}(t)*\eta^{\beta}_{ij}(t)}{\sum_{j \in allowed\ k} \tau^{\alpha}_{ij}(t)*\eta^{\beta}_{ij}} \quad .... (.2)$$

allowedk = {1, 2, 3, ..., n} is the list of the cities the k^{th} hasn't been visited yet. Weight α is pheromone factor, which denoted as the influence of pheromone concentration to the path choosing. When it equals to 0, the ant currently selects completely according to greedy rule for path planning. Weight β is the heuristic factor, denoted as the influence of distance of two cities to the path choosing. When it equals to 0, the path choosing depends entirely on the pheromone concentration.

In the existing ACO algorithm the value of next city j is determined by following formula:

$$J= \begin{cases} \text{argmax } \{ \tau^{\alpha}_{ij}(t)*\eta^{\beta}_{ij}(t) & , \ q< \text{random set} \\ S, & \text{otherwise, where S denotes the equation (2).} \end{cases} \quad ....(3)$$

A random set q is selected in between 0 and 1 which denotes the value of pheromone. The random factor q is used as a search guide to search for the next city.

In the existing method the random factor q is used as next city finder. By using random factor, the ant not only can be guided to search within the field of the optimal path but also can use the accumulated knowledge to find the better route following roulette algorithm.

(ii) Proposed Algorithm: Existing Improved ACO algorithm has some limitations:

a) The best route i.e. next city (j) is found by using pheromone parameter α. The value of α has been selected in between 0 and 1.If the value of α becomes zero, the algorithm will only work for local minimum. It will not work for global minimum i.e. the algorithm will not work for longest distance or large solution space. To overcome the problem, a proper selection method of α is required.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

b) It uses a roulette algorithm to find the better route. The roulette algorithm works upon selecting a random value and it is based on exceptions. Results generated by this algorithm cannot be guaranteed and is based on luck by chance policy.

To overcome the above limitations we propose a new algorithm that will work global optimal solutions and will always gives the best results. The proposed method works with following enhancements in existing algorithms;

1) In existing algorithm of [39], local search method does not use pheromone information but it is logical that local search will be using existing information when pheromone values are ready to use.

Instead of logical, we can directly use pheromone values in local search. In the proposed method we are using pheromone information to calculate Dynamic candidate sets of best possible nodes. After every iteration, Candidate sets are updated using pheromone information. Figure below shows the proposed method.

In the proposed method candidate sets of every node is updated by Global pheromone update rule given by:

$$\boldsymbol{\tau}^{\alpha}_{i,j}(t) = (1-\rho).\ \tau_{i,j} + \sum^{M}_{i=1} \Delta\boldsymbol{\tau}^{k}_{,i,j} \quad \text{.........................}(4)$$

Where $\boldsymbol{\tau}_{i,j}(t)$ = is the amount of pheromone on the edge (i, j) at time t; ρ is a parameter governing pheromone decay such that $0 < \rho < 1$.

At first Candidate Set of every node is initialized using nearest node in the set. Then after every iteration, for a node its corresponding nodes with maximum value of pheromone are inserted into the first place of the candidate set of that particular node. Before next iteration took place, a node in the last place of candidate set is removed to avoid repetition.

The algorithm to find candidate sets is mentioned below:

Algorithm Candidate ( )

Step I: For each node n in Iteration find the best.

i= argmax {τ [n][0], τ [n][1], . . . τ [n][size - 1] }

Best= i, where ∀ J ≠ n and [n][J] ≤ [n][i]

Step II: Remove the last node of candidates-set of node n.

Step III: Insert best into the first place of candidates-set of node n.

Step IV: Remove the last node of candidates-set of node n.

Step V: Insert next successor node of n in best tour into the first of candidates-set of node n.

End for loop.

End Algorithm.

By using above method every time we are getting the highest value of pheromone, and so we can come up from local trap. A higher pheromone values represents a dense route that helps ants to search the next path fastly and lower ones represents weak value that reduces the searching process.

If random set value in [39] is too small, the effect of introducing random factor is not obvious. By using very small value of random factor; the ants cannot be guided properly for optimal path. By introducing the concept of candidate set these problems can be solved. Also if random set value is closes to 1.0, the algorithm will easily trapped into local optimum by ignoring most of the path.

In order to guide the ants visit effectively and not trapped in local optimum we use boundary mutation operator within the candidate set to select the value of next move.

The Pseudo code of proposed algorithm is given below:

Algorithm GenACO ( )

{

Step 1: Set A [M][N]=Candidate( ).

Step 2: Find the best pheromone by using following rule:

a) Find the best pheromone from randomly selected by sorting in non-decreasing order.

b) For k=N to 0 do

c) Apply Boundary Mutation Operator with the best.

d) Say parent1= A [0](Low value) and parent2= A [1] (the best found)

B= (A [0] +A [1])/2 ;(Every time select the average ones)

Return B (every time best Pheromone is returned)

e) Collect the remaining unused pheromone to get a new Candidate set.

f) Repeat step until we end up with the new city.

}

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

(iii) Proposed algorithm for finding shortest path:

Algorithm TSP Proposed (City Info, p, i, j, k, B)

{

I represent initial point represents final point of city and k represents total visited city.

Input: City info, Ants number, p. (Assign some values).

Step I: Initialization, the initial population of candidate solutions is usually generated randomly across the search space in between S.

Step II: Select a random number

Random number=B.

Step III: while i> 0 and j>0 do

Step IV: while j< Iteration time do

IF q<B, the first element from candidate set

Choose city where the transition probability is max.

$\tau^{\alpha}_{i,j}(t) = (1-\rho).\ \tau_{i,j} + \sum^{M}_{i=1} \Delta\tau^{k}_{,i,j}$

Else

$\tau^{\alpha}_{i,j}(t) = (1\text{-}p).\ \tau_{i,j} + P * \tau_0$

0, otherwise

End IF.

}

One more improvement has been done in this thesis for selecting the value of τ_0, (Initial value of Pheromone). Most of the researches have fixed this initial value, and it is fixed in every iterations. To avoid the local trap we are suggesting taking the average value. i.e. $\tau_{0=}\ \{\tau_{min} + \tau_{max}\}$.

The lower and upper bounds of the pheromone values (i.e., τ_{min} and τ_{max}) are obtained on the basis of Stutzle and Hoos' analytical method.

IV. Simulation Results

We have conducted several experiments to test the performance of proposed algorithm in terms of Execution time and this section also shows performance comparison between the proposed algorithm and existing scheduling algorithms FCFS. One data center and one scheduler are used for the simulation. Number of virtual machines used for simulation is 10, 20 and 30. Numbers of virtual machines are increased per evaluation to check the performance of proposed algorithm in each environment. During each evaluation, proposed algorithm has tested ten times for each workflow Montage50. Below subsections shows the results of each evaluation.

Table below shows the result set with different values of Alpha for Ant System and Proposed system.

Pheromone Value (Alpha)	Ant System (Time in Seconds)	Proposed Method (Time in Seconds)
0.6	11.9761	0.0983
0.7	11.347	0.0981
0.8	11.290	0.0977
0.9	11.201	0.0782
1.0	11.035	0.0773

Table 1: Time comparisons for Pheromone values.

Number of Ants	Ant System (Time in Sec.)	Proposed Method (Time in Sec.)
100	2.1299	0.0093
200	3.9997	0.0089
500	10.009	0.0088
1000	19.0848	0.0083

Table 2: Time comparisons with Number of Ants.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

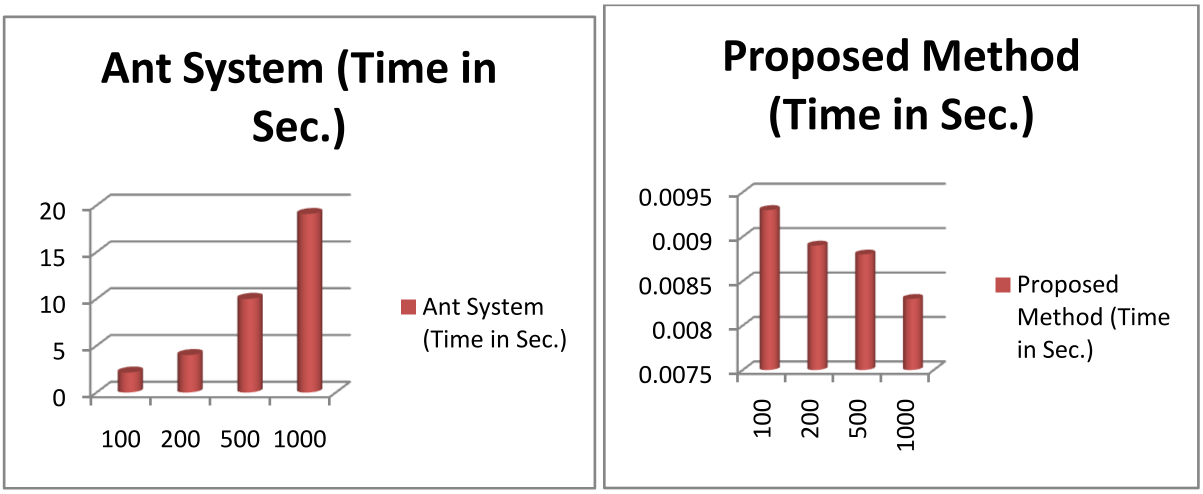


Figure 3: Graph showing different behaviour for two Algorithms.

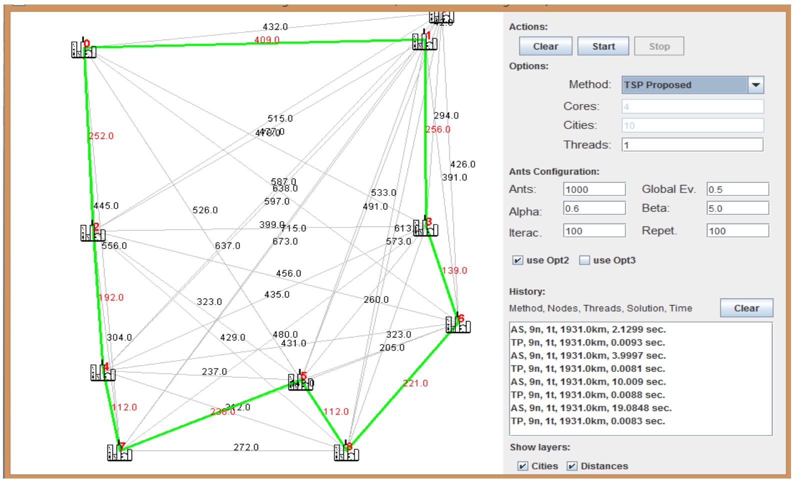


Figure 4: Results with time for different number of Ants.

Results of above evaluations show that proposed algorithm completes shortest path with lower searching time and higher performance as compared to Ant System. Performance of proposed algorithm is better than AS for 100, 200,500, and 1000 ants. Results show that proposed algorithm behaves better in terms of searching time. As we increased a number of ants which are used for searching, still the total time of searching the shortest path for the proposed algorithm is less than the AS.

V. Conclusion and Future Work

Through this paper we investigate the limitation of pheromone update rule and proposed a new algorithm based on Candidate set property that will help in finding the best route for travelling salesman problem. The proposed algorithm performs better for Global optimum solution i.e. it works well when the size of path will be large. It may also be helpful for real time application to find the shortest path. In AS algorithm most of the research has focused on reducing the shortest path with less concentration on pheromone factor. This research focus mainly on improving the range of pheromone value and always predicting the best route and reducing the total time for finding shortest path.

In future more studies are required in improving the pheromone values that will help to reduce the distance between two cities. However, each algorithm has its own limitations and cannot be universally accepted to perform well on all

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

optimization problems. In future more studies are required in improving the pheromone values that will help to reduce the distance between two cities.

REFERENCES

1. E.W. Dijkstra, A note on two problems in connexion with graphs, Number. Math. 1 (1) (1999) 269–271.
2. K. Savla, E. Frazzoli, and F. Bullo, “Travelling salesperson problems for the dubins vehicle,” IEEE Trans. Autom. Control. vol.53, no.6, pp.1378-1391, Jul.2008.
3. G. Gutin and A. Punnen, “The travelling salesman problem and its variations,” Combinatorial Optimization, vol.4, no.2, pp. 193-25, May 2007.
4. S. S. Ray, S. Bandyopadhyay, and S. K. Pal, “Gene ordering in partitive clustering using microarray expressions,” J. Biosci., vol. 32, no. 5, pp. 1019–1025, Aug. 2007.
5. R. M. Newton and W. H. Thomas, “Bus routing in a multischool system,” Computers & Operations Research, vol. 1, no. 2, pp. 213- 222, Aug. 1974.
6. [17] C. Cook, D. A. Schoenfeld, and R. L. Wainwright, “Finding rural postman tours,” in Proceedings of ACM symposium on Applied Computing, Mar. 1998, pp. 318–326.
7. A. Colorni, M. Dorigo, V. Maniezzo et al., “Distributed optimization by ant colonies,” in Proceedings of the first European conference on artificial life, vol. 142.Paris, France, 1991, pp. 134–142.
8. X. Wang, T.-M. Choi, H. Liu, and X. Yue, “Novel ant colony optimization methods for simplifying solution construction in vehicle routing problems.”
9. I. Chaari, A. Koubaa, H. Bennaceur, S. Trigui, and K. Al-Shalfan, “smartpath: A hybrid aco-ga algorithm for robot path planning,” in 2012 IEEE Congress on Evolutionary Computation. IEEE, 2012, pp. 1–8.
10. B. Zhang, X. Sun, L. Gao, and L. Yang, “End member extraction of hyper spectral remote sensing images based on the ant colony optimization (aco) algorithm,” IEEE transactions on geo science and remote sensing, vol. 49, no. 7, pp. 2635–2646, 2011.
11. H. N. Psaraftis, “Vehicle routing: Methods and studies,” Dynamic Vehicle Routing Problems. North Holland, Amsterdam, the Netherlands, pp. 223–248, 1988.
12. Yan Cao and Jun Zhang,” Dynamic Swarm Intelligence to solve TSP using Reuse strategy”, IEEE- 2016.
13. W. Guo, M. Chen, L. Wang, Y. Mao, and Q. Wu, "A survey of biogeography-based optimization," Neural Computing and Applications, pp. 1- 18, 20J6.
14. J. Lin, “A hybrid discrete biogeography-based optimization for the permutation flow shop scheduling problem," International Journal 01 Production Research, pp. J- 1O, 2015.
15. Huynh Thi Thanh Binh and Pham Dinh Thanh,” New Migration Operator in Biogeography-based Optimization for Solving Travelling Salesman Problem” ,IEEE 2016.
16. Fozia Hanif Khan, Nasiruddin Khan, SyedInayaTullah, And Shaikh Tajuddin Nizami, “Solving TSP problem by using Genetic Algorithm”, International Journal of Basic &Applied Sciences IJBAS Vol: 9 No: 10
17. Wei Luo, Dong Lin,” An Improved Ant Colony Optimization and Its Application on TSP Problem” IEEE 2016.

BIOGRAPHY

Sonia Rajput is a Research Assistant in the Computer Science Department, Shri Ram Institute of Science & Technology, Jabalpur, M.P.