Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 07, July-2018

Solving Travelling Salesman Problem(TSP) Using Ant Colony Optimization(ACO)

Nwamae, Believe B., Kabari, Ledisi G.
1 Computer Science Department, Ignatius Ajuru University of Education, Port Harcourt, Nigeria
2 Computer Science Department, Ken Saro-Wiwa Polytechnic, Bori, Nigeria

***Abstract* - Vehicle traffic congestion leads to air pollution, driver frustration, and costs billions of dollars annually in fuel consumption. Finding a proper solution to vehicle congestion is a considerable challenge due to the dynamic and unpredictable nature of the network topology of vehicular environments, especially in urban areas. Ant colony optimization (ACO) has been widely used for different combinatorial optimization problems. Ant Colony Optimization (ACO) as a heuristic algorithm has been proven a successful technique and applied to a number of combinatorial optimization (CO) problems. The traveling salesman problem (TSP) is one of the most important combinatorial problems. We present a bio-inspired algorithm, food search behavior of ants, which is a promising way of solving the Travel Salesman Problem. in this paper, we investigate ACO algorithms with respect to their runtime behavior for the traveling salesperson (TSP) problem. Simulation results conducted using MATLAB suggests that the proposed system would perform consistently despite increase of vehicles with in the given area.**

Keywords: Traveling Sales Salesman Problem(TSP), Ant Colony Optimization(ACO), Congestion, Huristic Algorithm, MATLAB

INTRODUCTION

All over the world especially in Over the last decade, vehicle population has dramatically increased (Jabbarpour et al., 2014). This large number of vehicles leads to heavy traffic congestion, air pollution, high fuel consumption and consequent economic issues (Narzt et al., 2010). In 2010, the American people faced a lot of difficulties due to vehicle congestion which forced their government to spend 101 billion dollars on the purchase of extra fuel (Jabbarpour et al., 2014). Based on a report by Texas A&M Transportation Institute (Jabbarpour et al., 2014), it is estimated that fuel consumption will rise up to 2.5 billion gallons (from 1.9 billion gallons in 2010) with a cost of 131 billion dollars in 2015.

Building new, high-capacity streets and highways can mitigate some of the aforementioned problems. Nevertheless, this solution is very costly, time consuming and in most cases, impossible because of space limitations. On the other hand, optimal usage of the existing roads and streets capacity can lessen the congestion problem in large cities at a lower cost. Applying bio-inspired algorithm, promises to have potential in solving these problems. One such algorithm is the Ant Colony Optimization. In the case of ACO algorithms the theoretical analyses of their runtime behavior has been started only recently.

We increase the theoretical understanding of ACO algorithms by investigating their runtime behavior on the well-known traveling salesperson (TSP) problem. For ACO the TSP problem is the first problem where this kind of algorithms has been applied. Therefore, it seems to be natural to study the behavior of ACO algorithms for the TSP problem from a theoretical point of view in a rigorous manner. ACO algorithms are inspired by the behavior of ants to search for a shortest path between their nest and a common source of food. It has been observed that ants find such a path very quickly by using indirect communication via pheromones. This observed behavior is put into an algorithmic framework by considering artificial ants that construct solutions for a given problem by carrying out random walks on a so-called construction graph. The random walk (and the resulting solution) depends on pheromone values that are values on the edges of the construction graph. The probability of traversing a certain edge depends on its pheromone value. It is a relatively novel meta-heuristic technique and has been successfully used in many applications especially problems in combinatorial optimization. ACO algorithm models the behavior of real ant colonies in establishing the shortest path between food sources and nests. Ants can communicate with one another through chemicals called pheromones in their immediate environment. The ants release pheromone on the ground while walking from their nest to food and then go back to the nest. The ants move according to the amount of pheromones, the richer the pheromone trail on a path is, the more likely it would be followed by other ants. So a shorter path has a higher amount of pheromone in probability, ants will tend to choose a shorter path. Through this mechanism, ants will eventually find the shortest path. Artificial ants imitate the behavior of real ants, but can solve much more complicated problem than real ants can. Consider Fig. 1A Ants arrive at a decision point in which they have to decide whether to turn left or right (Dorigo and Gambardella, 1997).

Since they have no clue about which is the best choice, they choose randomly. It can be expected that, on average, half of the ants decide to turn left and the other half to turn right. This happens both to ants moving from left to right (those whose name begins with an L) and to those moving from right to left (name begins with a R). Figs. 1B and 1C show what happens in the immediately following instants, supposing all ants walk at approximately the same speed. The number of dashed lines is roughly proportional to the amount of pheromone that the ants have deposited on the ground. Since the lower path is shorter than the upper one, more ants will visit it on average, and therefore pheromone accumulates faster. After a short transitory period the difference in the amount of pheromone on the two path is sufficiently large so as to influence the decision of new ants coming into the system (this is shown by Fig. 1D).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 07, July-2018

From now on, new ants will prefer in probability to choose the lower path, since at the decision point they perceive a greater amount of pheromone on the lower path (Dorigo and Gambardella, 1997). This in turn increases, with a positive feedback effect, the number of ants choosing the lower, and shorter, path. Very soon all ants will be using the shorter path. The travelling salesman problem (TSP) is the problem of finding a shortest closed tour which visits all the cities in a given set. In this article we will restrict attention to TSPs in which cities are on a plane and a path (edge) exists between each pair of cities (i.e., the TSP graph is completely connected).

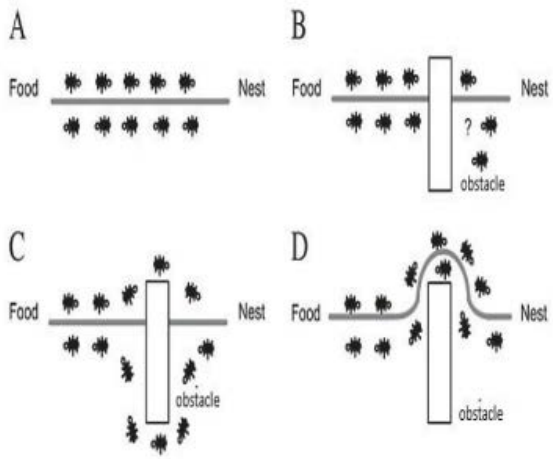

Figure 1 -How real ants find the Shortest Path (Dorigo and Gambardella, 1997)

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 formed along the shorter path

TRAVELING SALESMAN PROBLEM

The traveling salesman problem (TSP) is the problem of finding a shortest closed tour which visits all the cities in a given set (Hingrajiya, Gupta and Chandel, 2012). In this article we will restrict attention to TSPs in which cities are on a plane and a path (edge) exists between each pair of cities (i.e., the TSP graph is completely connected) (Dorigo and Gambardella, 1997). Traveling salesman problem (TSP) is one of the well-known and extensively studied problems indiscrete or combinational optimization and asks for the shortest round trip of minimal total cost visiting each given city (node) exactly once. TSP is an NP-hard problem and it is so easy to describe and so difficult to solve. The definition of a TSP is: given N cities, if a salesman starting from his home city is to visit each city exactly once and then return home, find the order of a tour such that the total distances (cost) traveled is minimum. Cost can be distance, time, money, energy, etc. A complete weighted graph G= (N, E) can be used to represent a TSP, where N is the set of n cities and E is the set of edges (paths) fully connecting all cities. Each edge (i,j)∈E is assigned a cost dij, which is the distance between cities i and j. dij can be defined in the Euclidean space and is given as in equation(1).

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

One direct solving method is to select the route which has minimum total cost for all possible permutations of N cities. The number of permutations can be very large for even 40cities. Every tour is represented in 2n different ways (for symmetrical TSP). Since there are n! possible ways to permute n numbers, the size of the search space is then |S|=n!/(2n)=(n-1)!/2.

MATHEMATICAL MODEL OF ACS

The ant colony system(ACS) is different from the ant system in three aspects: first is the state transition rule (pseudo-random proportional rules) which provide the balance between exploration of new edges and exploitation of a priori. Second is the local pheromone updating rule is applied while ants construct a solution. And third is the global updating rule is applied only to edges which belong to the best ant tour. In the ACS algorithm, we discuss the following three aspects of ACS in brief.

$$S = \begin{cases} \arg \max_{u \in j_k(r)} \{[\tau(r,u)]^{\alpha} . [\eta(r,u)]^{\beta}\}, & if\ q \le q_0 (exploitation) \\ S, & otherwise \quad (biased\ exploitation) \end{cases} \quad (2)$$

ve to by

Where τ is the pheromone, $\eta = 1/d$ is the inverse of the distance d (r , s). j (r) is the set of cities that remain to be visited by k ant positioned on city r. β is a parameter which determines the relative importance of pheromone density versus distance ($\beta > 0$). q is a random number uniformly distributed in [0…..1], q0 is a parameter ($0 \le q0 \le 1$), the parameter q0 determines the relative importance of exploitation versus exploration. S is a random variable selected according to the probability distribution given by: Every time an ant in city r has to choose a city s to move to, it samples a random number q $(0 \le q \le 1)$. If $q < q0$ then the best

$$P_k(r,s) = \begin{cases} \frac{[\tau(r,s)]^{\alpha} . [\eta(r,s)]^{\beta}}{\sum_{u \epsilon J_k(r)} [\tau(r,u)]^{\alpha} . [\eta(r,u)]^{\beta}}, & if\ S \epsilon J_k(r) \\ 0, & otherwise \end{cases} \quad \ldots(3)$$

wise an

Local Pheromone Update Rule

While construction a tour, an ant will modified the pheromone level on the visited edges using the local pheromone updating rule by (4)

$$\tau(r,s) \leftarrow (1-p).\tau(r,s) + p.\Delta\tau_0$$
$$where\ \Delta\tau_0 = \left(1/L_{mn}.n\right) \quad \ldots(4)$$

Global Pheromone Update Rule

After all ants have constructed a tour, only the globally best ant which produces the shortest tour from the beginning of the trail will be allowed to do pheromone update using the global pheromone updating rule(5).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 07, July-2018

$$\tau(r,s) \leftarrow (1-p).\tau(r,s) + p.\Delta\tau(r,s)$$

where

$$\Delta\tau(r,s) = \begin{cases} \frac{1}{L_{gb}} & ,if\ (r,s) \in Global\ best\ tour \ldots (5) \\ 0, & otherwise \end{cases}$$

Candidate List Strategy

The candidate list is a strategy that is applied to the TSP to improve the performance of the ACS algorithm.

Initialize Set the initial pheromone values for all edges, populate the cost matrix with the distances between the cities
Loop / at this level each loop is called an iteration */*
Each ant is positioned on a starting node
Loop / at this level each loop is called a step */*
Each ant applies a state transition rule to incrementally build a solution and a local pheromone updating rule
Untill all ants have built a complete solution A global pheromone updating rule is applied
Untill End_condition

It was proposed by Dorigo and Gambardella (1997) to accommodate searching procedure of ACS for larger data. In the ACS algorithm, when an ant choose the next city, the probability of its transfer from city i to city j needs to be computed, and then the city whose decision probability is first need to consider those preferred cities listed in the candidate list. Only when an ant cannot find suitable city to choose then the decision to choose a city will consider those which are outside of the candidate list. The algorithm for creating the candidate list is given as follows:

```
candidate_list=n/4 /*size of candidate list*/
    determine cities that not yet visited
        do
            for i=1 to n
                if city s is not yet visited determine distance
                between city r and city s
                if distance < distance of previous city s
                   move city s into node_list
            end for
                   candidate_list=node_list
            while (until candidate_list is full)
```

PROPOSED APPROACH

To improve the performance of ACS algorithm, the proposed approach is based on the divide and conquer partition scheme. A 2-dimensional partitioning scheme was proposed by Karp which was similar to the construction of a k-dimensional tree data structure (Karp, 1977). It involved geometrically partitioning of the cities and then applied local optimization for improvement. Alternative partitioning schemes were introduced by Reinelt (1994), but the experimental result concluded that these approaches are not preferable to Karp's partitioning scheme. The ACS algorithm is better for problem of smaller size having less number of cities because of the random nature of the algorithm in which a large number of random decisions made on weighted choice are required to come together to construct an efficient solution as the size of the problem is increases. In the proposed approach the set of all the cities for a problem (S) is further sub divided into two disjoint sets S1 and S2 (In the proposed algorithm we use the nearby cities partition scheme to create the sub problems) and then proceed to find the solution of these two sub problems separately by applying the ACS algorithm using candidate list that generate a candidate tour. The candidate list applying only when the number of cities in the sub problems is very large or the number of cities is greater than 4. The overall solution or the original path is obtained by the conjunction of the solution given by these sub problems. Now focusing on reducing the length of the segments formed by the sub sets in the original problem in such a way that the nearby city of the two separate sub problems are joined at end point of the path and, after joining the end point of all the sub problems the original path can be obtained. The greedy edge exchange is used for optimizing the path by joining two sub problems. For joining the two sub problems we have chosen the closed city between the two sub problems and apply the path between these two cities and avoid the closed path of the sub problems for joining, repeat the same process for all sub problems to obtain the complete path. As the search space for these two sub problems get reduced because of the division of the cities from the original problem, the efficiency and accuracy of the ACS algorithm is much greater for these sub problems. The accuracy of the overall solution is upper bounded by the accuracy of the division of the cities for each sub sets, which is depend on the accuracy of the candidate tour. The proposed algorithm is combined with candidate list and the recursive approach that is applied to the sub problems to find the solution. Each ant builds a tour by repeatedly applying a stochastic greedy rule. While constructing the tour, an ant also modifies the amount of pheromone on the visited edges by applying the local updating rule. Once ants have completed a tour they use their memory to evaluate the built solution and to retrace the same tour backward and increase the intensity of the pheromone trails (,*j*). If j has not been visited previously, it can be selected with a probability that is proportional to the pheromone associated with edge (i,j). In the proposed system all the ants deposit pheromone. Once all ants have terminated their tour, the amount of pheromone on edges is modified again. Ants are guided in building their tour, by both heuristic information and by pheromone information. An edge with a large amount of pheromone is a very desirable choice.

EXPERIMENTAL RESULT AND DISCUSSION

The algorithm was implemented in MATLAB 2016.

Step 1: Partitioning the problem into N number of sub problems using the nearby city strategy, with the help of cost matrix.

Step 2: Initialization of parameters for each sub problems

Step 3: for sub problems 1 to N

Loop /* this level loop is called iteration*/

Each ants is positioned on a starting node according to distribution strategy.

For k=1 to m /* this level called step*/

Repeat:

Compute Candidate List /* for the large group size or group size>4.*/

Select node j to be visited next according to eq. (3)

Until ant k has completed a tour

End for

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 07, July-2018

Apply the local search to improve tour.

A local updating rule is applied as eq (5)

Until end_condition

End for

Step 4: for each sub problems

Find the closed city between two sub problems using

cost matrix.

Apply edge exchange for the local optimization of

the tour.

Apply Global updating rule as eq. (4).

For each test case of TSP 300 trails were conducted. The performance of the algorithm is based on the correct tuning of the parameters. We set the parameters for the ACS as follows: α = 1, β is the dynamically value for the proposed algorithm and β =1 for the TSP, ρ = 0.05 and the number of ants is [40, 80, 200 and 500].

Experiment1 (Number of Ants = 40)

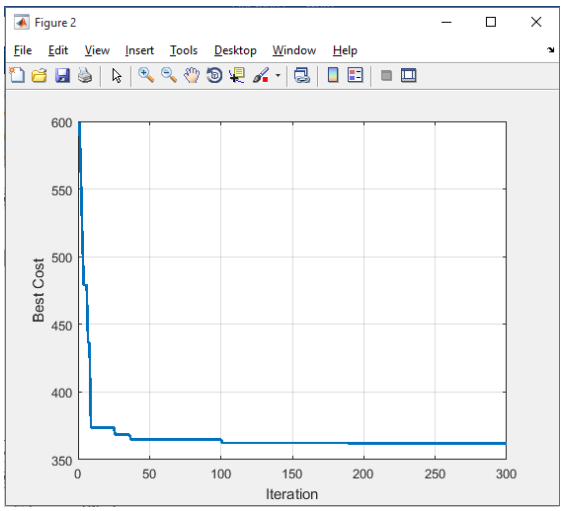

Figure2 - *Best Cost Iteration for Experiment1*

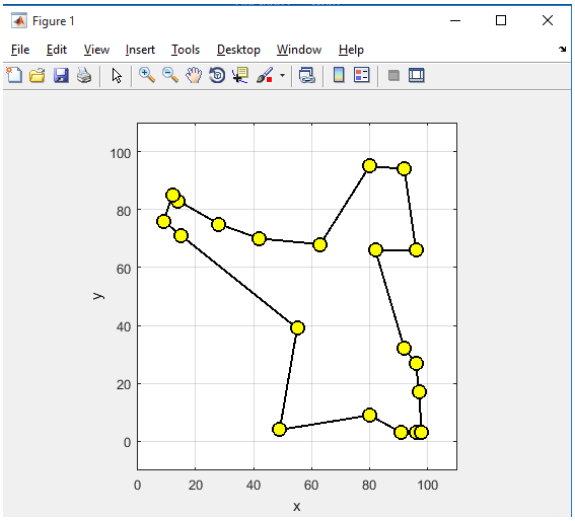

Figure3 - *Best Path for Experiment1*

Experiment2 (Number of Ants = 80)

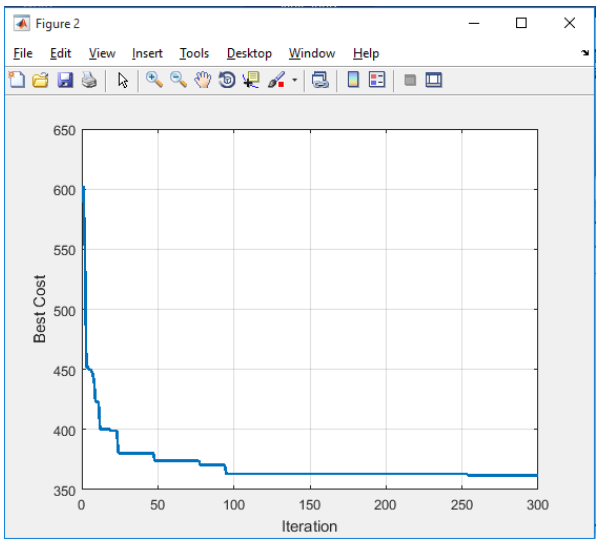

Figure4- *Best Cost Iteration for Experiment 2*

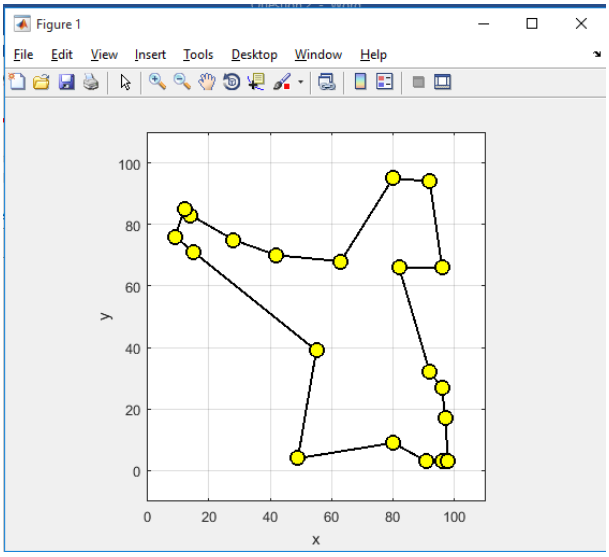

Figure5- Best Path for Experimen 2

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 07, July-2018

Experiment3 (Number of Ants = 200)

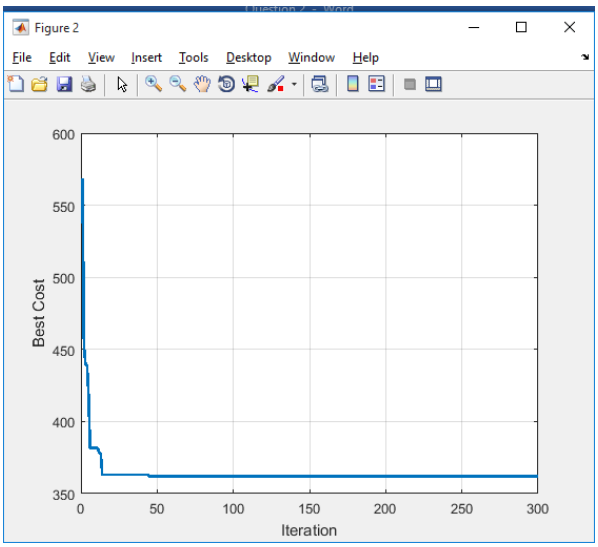

Figure6- Best Cost Iteration for Experiment3

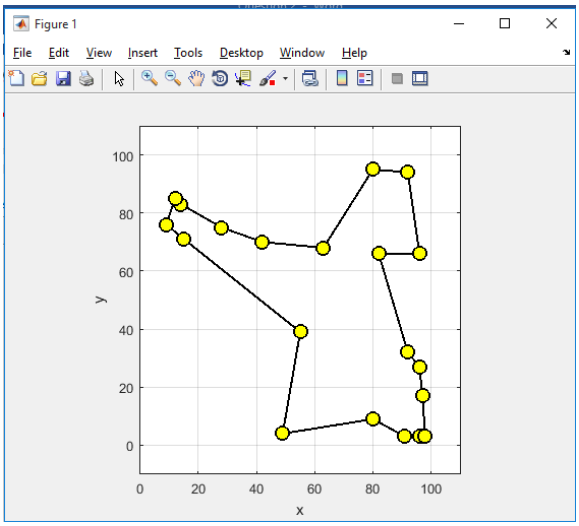

Figure7- Best Path for Experimen3

Experiment4 (Number of Ants = 500)

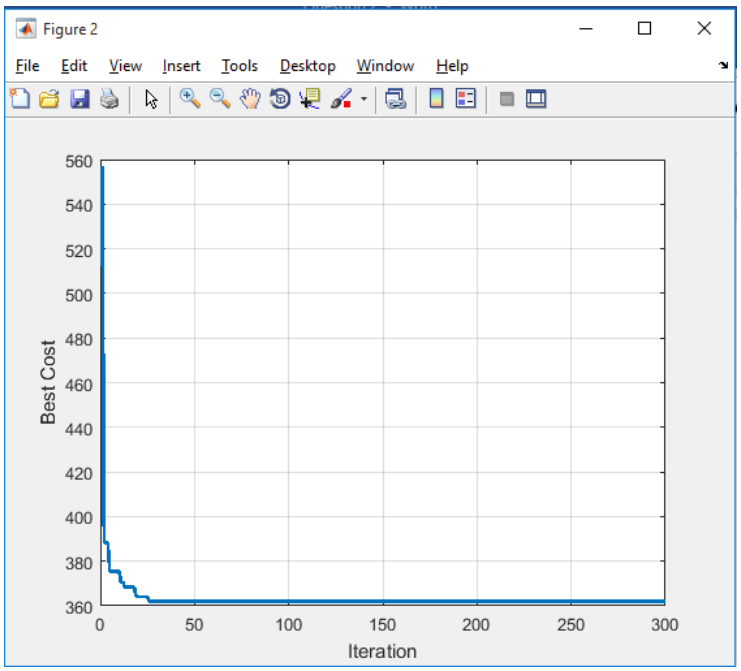

Figure8- Best Cost Iteration for Experiment4

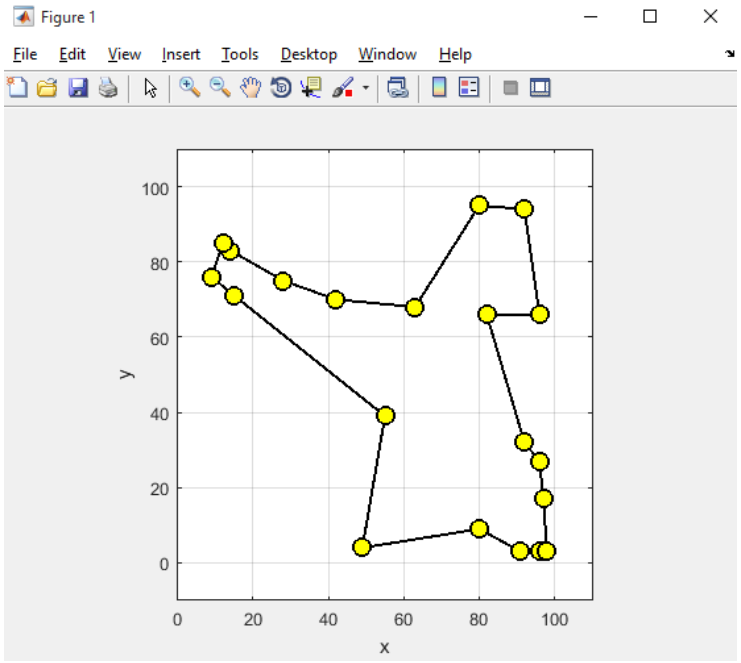

Figure9- Best Path for Experimen4

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 7 Issue 07, July-2018

DISCUSSION

For experiment1, ant size is 40 and number of iteration is 300, it does not converge. As Iteration increases, the best cost reduces significantly immediately before the 50th iteration. It continues reducing and stops reducing at 100 iterations. The best cost is 362.038.

For experiment2, ant size is 80 and number of iteration is 300, It Does not converge. As the Iteration approaches 50 there is a significant reduction in the best cost. We notice slight reduction of the best cost before 100. Very little reduction is observed before the 250th Iteration. The best cost is 362.038.

For experiment3, ant size is 200 and number of iteration 300. It does not converge. As Iteration increases, the best cost reduces significantly before 50 iterations. The best cost remains the same at 362.038.

For experiment4, ant size is 500 and number of iteration 300. Convergence is achieved. As iteration increase, we notice that best cost reduce significantly and achieve convergence before the 50th iteration. The best cost is 362.038.

From the simulation results and discussion, it summarizes the results of the proposed algorithm on a few test cases. It suggests that ACO is algorithm is very robust and is effect with significant increase in the number of ants in the network.

CONCLUSION

This paper presents an efficient approach for solving traveling salesman problem based on the divide and conquers strategy (in which cities are divided into the number of sub city) in which the ant colony algorithm using the candidate list is applied. From our experimental result the proposed algorithm is effective to find the better solution. In the future, it would be interesting to study the application of the proposed algorithm to other combinational problems and check the efficiency by comparing this method with other proposed method.

REFERENCES

[1] Bajpai, A. and Yadav, R. (2015). Ant Colony Optimization (ACO) For the Traveling Salesman Problem (TSP) Using Partitioning. *International Journal of Scientific & Technology Research*, 4(09).

[2] Dorigo, M. and Gambardella, L. (1997). Ant colonies for the travelling salesman problem. *Biosystems*, 43(2), pp.73-81.

[3] Dorigo, M. and Gambardella, L. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1), pp.53-66.

[4] Hingrajiya, K., Gupta, R. and Chandel, G. (2012). An Ant Colony Optimization Algorithm for Solving Travelling Salesman Problem. *International Journal of Scientific and Research Publications*, [online] 2(8). Available at: https://pdfs.semanticscholar.org/3140/5915ad098f10e8e2b225e8c8cbf6a4c5a832.pdf [Accessed 3 Jun. 2018].

[5] Hölldobler, B. and Wilson, E. (1990). *The ants*. Berlin: Springer-Verlag.

[6] Jabbarpour, M., Jalooli, A., Shaghaghi, E., Noor, R., Rothkrantz, L., Khokhar, R. and Anuar, N. (2014). Ant-based vehicle congestion avoidance system using vehicular networks. *Engineering Applications of Artificial Intelligence*, 36, pp.303-319.

[7] Karp, R. (1977). Probabilistic Analysis of Partitioning Algorithms for the Traveling-Salesman Problem in the Plane. *Mathematics of Operations Research*, 2(3), pp.209-224.

[8] Narzt, W., Wilflingseder, U., Pomberger, G., Kolb, D. and Hortner, H. (2010) 'Self-organising congestion evasion strategies using ant-based pheromones', *Intelligent Transport Systems*, IET, Vol. 4, No. 1, pp.93–102.

[9] Reinelt, G. (1994). *The Traveling Salesman*. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg.