Journal of the Nigerian Association of Mathematical Physics
***Volume* 25 (November, 2013), pp 333 – 338**

Application of Ant Colony Optimization Algorithm to Solve Travelling Salesman Problem

Amiolemhen P.E.* and *Haruna J. I.

Production Engineering Department,
Faculty of Engineering, University Of Benin,
P.M.B 1154, Benin City, Nigeria.

Abstract

This paper addresses the optimization of a Traveling Salesman Problem (TSP) using the Ant Colony Optimization algorithm to find the optimal tour distance or minimal distances between cities. Ants of the artificial colony are able to generate successively shorter feasible tours by using information accumulated in the form of a pheromone trail deposited on the edges of the TSP graph.

To test the performance of the proposed method, a six-city symmetric TSP problem is solved using the data collected for the city-city distances. The implementation of the algorithm was done using the C++ programming language and the experimental results show a great effectiveness when the TSP problem is solved with the colony optimization algorithm.

***Keywords*:** Ant colony optimization; Travel salesman problem (TSP); optimal tour distance, pheromone, Tabu list

Nomenclature

K	Any nonempty proper subset of the cities 1… m.
L^k	Length of tour made by ant k
P_{ij}	Probability of an ant in choosing a city
Q	Product of the highest number of possible ways an ant can take and the longest distance travelled.
S^{best}	Length of best tour
d_{ij}	Distance value between cities
m	Number of ant
n	Number of nodes (cities)
t	number of iterations
$tabu_k$	Tabu list of ant k. This list consists of cities that have already being visited until time t and the ant is forbidden to choose such cities repeatedly
x_{ij}	the decision variable that indicates whether the path from city i to city j is included in the tour
$\Delta\tau_{ij}^{best}$	Value of pheromone deposited by best ant on their path
α	Exponent of pheromone value
β	Exponent of heuristic value
$\eta = 1/d_{ij}$	Heuristic value
ρ	Pheromone decay constant
$\tau_{ij}(t)$	Pheromone value
$\tau_{ij}^{old}(t)$	Amount of pheromone left after evaporation

Corresponding author: ***Amiolemhen P.E.,*** E-mail: pamaka2003@yahoo.co.uk, Tel.: +234038516786

1.0 Introduction

In the recent past the commodity industry and company face the challenges of meeting the demand of their customers or consumers for consistent and optimum delivery of services due to lack of proper optimization techniques. It is against this backdrop that the optimization problems are of high importance both for the industrial world as well as the scientific world.

The Traveling salesman problem belongs to the class of optimization problem called combinatorial optimization problem. The Traveling salesman problem (TSP) is one which has commanded much attention of mathematicians and computer science specifically because it is so easy to describe but difficult to solve [1].

1.1 The travelling salesman problem

This problem involves developing a minimum cost route for a salesman visiting m cities then returning home. The basic problem involves selection of a route visiting all cities which minimizes the total distant travelled or travel cost.

The mathematical structure of a traveling salesman problem is a graph where each city is denoted by a point (or node) and lines are drawn connecting every two nodes (called arcs or edges). Associated with every line is a distance (or cost). When the salesman can get from every city to every other city directly, then the graph is said to be complete. A round-trip of the cities corresponds to some subset of the lines, and is called a tour or a Hamiltonian cycle in graph theory. The length of a tour is the sum of the lengths of the lines in the round-trip. Depending upon whether or not the direction in which an edge of the graph is traversed matters, one distinguishes the asymmetric from the symmetric traveling salesman problem.

The symmetric travelling salesman problem then becomes [2]

$$\min \sum_{j=1}^{m} \sum_{i=1}^{m} d_{ij} x_{ij} \tag{1}$$

s. t. $\sum_{j=1}^{m} x_{ij} = 1$ for i = 1… m

$\sum_{i=1}^{m} x_{ij} = 1$ for j = 1… m

$\sum_{i \le K} \sum_{j \le K} x_{ij} \le |K| - 1$ for all $K \subset (1... m)$

$x_{ij} = 0$ or 1 for all i, j $i \neq j$

For the symmetric traveling salesman problem, the direction traversed is immaterial, so that $d_{ij} = d_{ji}$. Moreover, the decision variable (x_{ij}) equals one if the salesman goes from city i to city j, and zero otherwise.

The symmetric TSP can simply be stated as follows:

If a traveling salesman wishes to visit exactly once each of a list of m cities (where the cost of traveling from city i to city j is c_{ij} and then return to the home city, what is the least costly route and the shortest possible distance the traveling salesman can take?.

The importance of the TSP is that it is representative of a larger class of problems known as combinatorial optimization problems [3]. The TSP problem belongs in the class of combinatorial optimization problems known as NP-hard. Specifically, if one can find an efficient algorithm (i.e., an algorithm that will be guaranteed to find the optimal solution in a polynomial number of steps for the traveling salesman problem, then efficient algorithms could be found for all other problems in the NP-complete class. If there are n cities, the number of possible tours is given by (n -1)! /2 [4]. The most fascinating part of the TSP problem is that it often arises in real life problems. This is one of the main reasons for the huge interest in the TSP, and hence the field of real life applications for the TSP deserves a closer look. Some of them include: Vehicle routing; Drilling of printed circuit board; Crew scheduling problem; The order-picking problem in warehouses; Mask plotting in printed circuit board(PCB) production; School bus routing problem; Mission planning problem during war; etc.

1.2 The Conventional Ant Colony Optimization Algorithm

The Ant Colony Optimization (ACO) algorithm is a meta-heuristic that has a combination of distributed computation, autocatalysis (positive feedback), and constructive greediness to find an optimal solution for combinatorial optimization problems. This algorithm tries to mimic the ant's behavior in the real world. Since its introduction, the ACO algorithm has received much attention and has been incorporated in many optimization problems, namely the network routing, traveling salesman, quadratic assignment, and resource allocation problems [4].

The ACO algorithm has been inspired by the experiments run by Goss et al. [6] using a colony of real ants. They observed that real ants were able to select the shortest path between their nest and food resource, in the existence of alternate paths between the two. The search is made possible by an indirect communication known as stigmergy amongst the ants. While traveling their way, ants deposit a chemical substance, called pheromone, on the ground. When they arrive at a decision point, they make a probabilistic choice, biased by the intensity of pheromone they smell. This behavior has an autocatalytic effect because of the very fact that an ant choosing a path will increase the probability that the corresponding path will be chosen again by other ants in the future. When they return back, the probability of choosing the same path is higher (due to the increase of pheromone). New pheromone will be released on the chosen path, which makes it more attractive for future ants. Shortly, all ants will select the shortest path.

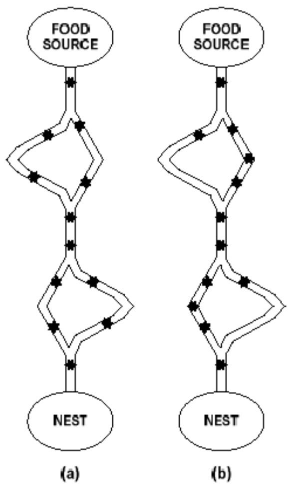


Fig.1: Double bridge experiment. (a) Ants start exploring the double bridge. (b) Eventually most of the ants choose the shortest path [7].

Fig. 1 shows the behavior of ants in a double bridge experiment. In this case, because of the same pheromone laying mechanism, the shortest branch is most often selected. The first ants to arrive at the food source are those that took the two shortest branches. When these ants start their return trip, more pheromone is present on the short branch than the one on the Long Branch. This will stimulate successive ants to choose the short branch. Although a single ant is in principle capable of building a solution (i.e., of finding a path between nest and food resource), it is only the colony of ants that presents the “shortest path finding” behavior. In a sense, this behavior is an emergent property of the ant colony. This behavior was formulated as Ant System (AS) by Dorigo et al. [5].

Based on the AS algorithm, the Ant Colony Optimization (ACO) algorithm was proposed in accordance with Dorigo and Di Caro formulation [8].

2. Methodology

The framework of the Ant colony optimization (ACO) approach is presented. Therefore the algorithm is designed based on the approach and it is applied in traveling salesman problem to
find the path for connected cities at minimum distance.

The solution to the travel salesman problem involves the following step by step procedure.

Step 1: (initialization)

Select a suitable number of ants (m) which is to be placed at each node (cities) of the construction graph. Take equal amount of pheromone value $\tau_{ij}(t)$ initially along all edges of the graph at first iteration (t=1).

Step 2: Compute the probability of selecting a city to be visited by the traveler. We define the transition probability from town i to town j for the k-th ant as:

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha} * [\eta]^{\beta}}{\Sigma[\tau_{ij}(t)]^{\alpha} * [\eta]^{\beta}} \qquad j \in tabu_k \tag{2}$$

Step 3: Determine the best path followed (S^{best}) among the path chosen by different ant. The path with minimal tour length tends towards the best path taken.

Step 4: Test for the convergence of the solution .The process is assumed to have converged if the ant take the same path. If convergence is not achieved, assume that all ants return back home and start again in search of food, then set iteration number as (t+1) and update the pheromone on the different edges as

$$\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t). \qquad \rho\ \varepsilon\ (0,1) \tag{3}$$

The pheromone deposited by the best ant on its path is given as:

$$\Delta\tau_{ij}^{best} = Q/S^{best} \tag{4}$$

The ant update its path as: $\tau_{ij}(t+1) = \tau_{ij}(t) + \sum\Delta\tau_{ij}^{best}$ (5)
The evaporation rate or pheromone decay factor ρ is assumed to be in the range of 0 to 1 .With new value of pheromone go to step 2, 3 and 4 until the process converge.
Termination point occurs at convergence i.e. until all ants choose the same best path or until the process has completed a pre-specified maximum number of iteration [9].

Pseudo Codes of Ant algorithm for TSP

Initialize:
Set t: =1 {t is the number of iterations}
For every edge (i,j) set an initial value $\tau_{ij}(t)$ for trail intensity
Place the m ants on the n nodes
For k: =1 to m do
Place the starting town of the k-th ant in tabuk
Repeat until tabu list is full
For k: =1 to m do
    Choose the town j to move to; with probability p_{ij} (t) {the k-th ant is now on town i in $tabu_k$ at time t}
        Move the k-th ant to the town j
    Insert town j in $tabu_k$
For k: =1 to m do
Compute the length L^k of the tour described by $tabu_k$
Update the shortest tour found
For the edge (i,j) (edges with the highest probability)
For k: =1 to m do
    $\Delta\tau_{ij}^{best} = Q/S^{best}$
Pheromone Update:
Set t: = t+1
For every edge (i,j) compute $\tau_{ij}(t+1)$ according to equation $\tau_{ij}(t+1) = (1-\rho)\,\tau_{ij}(t)$
For edges (i,j) (edges with highest probability) compute $\tau_{ij}(t+1)$ according to equation $\tau_{ij}(t+1) = \tau_{ij}(t) + \sum\Delta\tau_{ij}^{best}$
If ($t < t_{MAX}$) (and convergence not yet reached)
Then
    Empty all tabu lists
    Goto line 6
Else
    Print shortest tour
Terminate.

3. Experiment

The data collection for TSP is the cities and the distance between the cities. And, using the travelling distance chart involving six cities in Nigeria (i.e. Benin, Abuja, Calabar, Enugu, Ibadan, and Jos), we have Table 1.

Table 1: Travelling Distance Chart for Six Cities (in Kilometers)

	Benin 1	Abuja 2	Calabar 3	Enugu 4	Ibadan 5	Jos 6
Benin 1	0	493	490	254	300	7.58
Abuja 2	493	0	729	393	645	297
Calabar 3	490	729	0	276	518	870
Enugu 4	254	393	276	0	558	608
Ibadan 5	300	645	518	508	0	928
Jos 6	758	279	870	608	928	0

4. Results and discussion

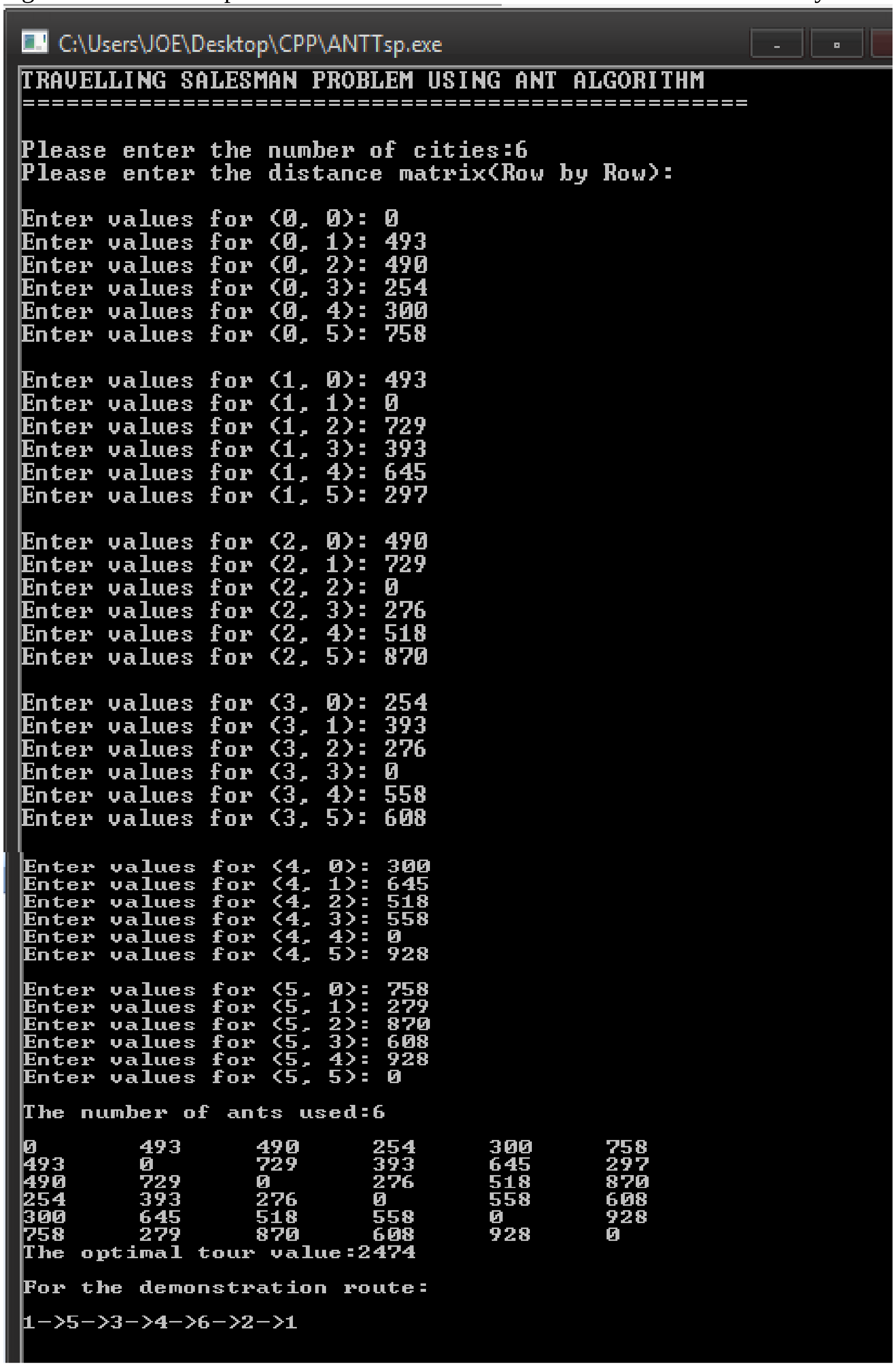
We apply ant colony optimization algorithm to find the optimal tour distance. Hence using the data of the travelling distance chart of Table 1, the Ant Colony Optimization algorithm was implemented in Microsoft C++ programming language and the optimal tour distance was achieved within few seconds.

The optimal route that was developed from the implementation is given as:

Benin ⟶ Ibadan ⟶ Calabar ⟶ Enugu ⟶ Jos ⟶ Abuja ⟶ Benin

Thus the Optimal tour distance = 300 + 518 + 276 + 608 + 279 + 493 = 2474 km

Fig. 2 shows the computer interface in Microsoft C++ domain for the Six-City TSP.



```
C:\Users\JOE\Desktop\CPP\ANTTsp.exe
TRAVELLING SALESMAN PROBLEM USING ANT ALGORITHM
=================================================

Please enter the number of cities:6
Please enter the distance matrix(Row by Row):

Enter values for (0, 0): 0
Enter values for (0, 1): 493
Enter values for (0, 2): 490
Enter values for (0, 3): 254
Enter values for (0, 4): 300
Enter values for (0, 5): 758

Enter values for (1, 0): 493
Enter values for (1, 1): 0
Enter values for (1, 2): 729
Enter values for (1, 3): 393
Enter values for (1, 4): 645
Enter values for (1, 5): 297

Enter values for (2, 0): 490
Enter values for (2, 1): 729
Enter values for (2, 2): 0
Enter values for (2, 3): 276
Enter values for (2, 4): 518
Enter values for (2, 5): 870

Enter values for (3, 0): 254
Enter values for (3, 1): 393
Enter values for (3, 2): 276
Enter values for (3, 3): 0
Enter values for (3, 4): 558
Enter values for (3, 5): 608

Enter values for (4, 0): 300
Enter values for (4, 1): 645
Enter values for (4, 2): 518
Enter values for (4, 3): 558
Enter values for (4, 4): 0
Enter values for (4, 5): 928

Enter values for (5, 0): 758
Enter values for (5, 1): 279
Enter values for (5, 2): 870
Enter values for (5, 3): 608
Enter values for (5, 4): 928
Enter values for (5, 5): 0

The number of ants used:6

0       493     490     254     300     758
493     0       729     393     645     297
490     729     0       276     518     870
254     393     276     0       558     608
300     645     518     558     0       928
758     279     870     608     928     0
The optimal tour value:2474

For the demonstration route:

1->5->3->4->6->2->1
```

Fig. 2: Solution to the Six-City Problem

5. Conclusion

The result of the graphical user interface clearly explains the connecting cities, shortest path between cities. A number of advantages for Ant colony optimization method can be seen: searching techniques are intuitive and successfully provide near-optimal solutions to complex real-life problems in a reasonable amount of computational time and with minimum distance.
The major limitation of this work is that for any given number of ants, there is a proper pheromone dispensing level and evaporation rate that will lead to convergence.
The main contribution of this work is that the Ant Colony Optimization Algorithm has been applied to solve a local TSP (i.e. the transportation problem). This can be extended to solve other similar TSP, such as Vehicle routing; Drilling of printed circuit board; etc.
Furthermore, future work will focus on further improvements on the approach as well as comparison with other meta-heuristics such as Simulated Annealing, Particle Swarm Optimization, Tabu Search, etc.

References

[1] Blum, C. (2006). Ant colony Optimization: Introduction and recent trends. Physics of Life Reviews,Vol.2, p.353-373.
[2] Papadimitriou, C.H.; Steiglitz, K. (1998). Combinatorial optimization: algorithms and complexity. Mineola, NY: Dover.
[3] Hoffman, A. J. and Wolfe, P. (1985), "History in The Traveling Salesman Problem", Lawler, Lenstra, Rinooy Kan and Shmoys, eds., Wiley, 1-16.
[4] Oystein, M.H. (2002).A multi level scheme for the travel salesman problem. University of Bersen.
[5] Dorigo, M., Maniezzo M. and Colorni, A. (1991). The Ant Systems: An Autocatalytic Optimizing Process. Revised 91-016, Dept. of Electronica, Milan Polytechnic.
[6] Goss, S., Aron, S.. Deneubourg J. L., and Pasteels, J. M. (1989). Self-organized Shortcuts in the Argentine Ant. Naturwissenschaften, 76:579–581.
[7] Dorigo, M., Di Caro, G. and Gambardella, L. M. (1998). Ant Algorithms for Discrete Optimization. Technical Report Tech. Rep. IRIDIA/98-10, IRIDIA, Universite Libre de Bruxelles, Brussels, Belgium.
[8] Dorigo, M. and Di Caro, G. (1999). New Ideas in Optimisation. McGraw Hill, London, UK,
[9] Dorigo, M., Maniezzo, V and Colorni A. (1996).The Ant system: Optimization by a colony of cooperative agent. IEEE Transactions on system, Man and cybernetic.p.1-33