Journal of the Nigerian Association of Mathematical Physics
Volume35,(May, 2016), pp149 – 154
© J. of NAMP

Using Tabu Search Technique for Solving Travel Salesman Problem

P. E. Amiolemhen and J. K. Okpowo

Department of Production Engineering, Faculty of Engineering,University of Benin City, Nigeria

Abstract

The travelling salesman problem (TSP) is a classical problem in discrete or combinatorial optimization and belongs to the NP-complete classes, which means that it may require an infeasible processing time to be solved by an exhaustive search method, and therefore less expensive heuristics in respect to the processing time are commonly used in order to obtain satisfactory solutions in short running time.
This paper address the optimization of a Traveling Salesman Problem using the Tabu search heuristic, also implementing Tabu Search using Visual basic.Net programming language to optimize the Traveling Salesman problem.
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 proposed algorithm was done using the Visual Basic.Net programming language and the experimental results show that the proposed algorithm provides better compromise between CPU time and effectiveness among some recent algorithms for the TSP.

Keywords: Tabu Search; Travel salesman problem; Optimal tour distance

Nomenclature

d_{ij}	Distance value between cities
TS	Tabu Search
TSP-	Travelling salesman problem
aTSP	Asymmetric travelling salesman problem
mTSP	multi-travelling salesman problem
sTSP	Symmetric travelling salesman problem
$N^*(x)$	Neighbor solutions of *x*
n	Number of nodes (cities)
x_{ij}	Decision variable that indicates whether the path from city i to city jis
x	Current solution
x'	Improve current solution
x*	Best solution
x_{ij}	Cost of traveling route from city I to city j
z(x)	summation of distance of solution x
z(x')	Summation of distance of improved solution *x*
z(x*)	Summation of distance of best solution

1.0 Introduction

One of the key roles of a Production or Manufacturing Industry is to minimize it resources, time, cost or available material and improve productivity with efficiency while still delivering its goods or services in a more effective way at a cheaper and lesser time of delivery to suite the customers' demands and needs. The challenge faces by majority of Industries when dealing with optimization of service and goods delivery to its esteem costumer is the Optimization problem well known as the Traveling Salesman Problem.
Given n is the number of cities to be visited, the total number of possible routes covering all cities can be given as a set of feasible solutions of the TSP and is given as (n-1)!/2.

Corresponding author:P. E. Amiolemhen, E-mail:Patrick.amiolemhen@uniben.edu, Tel.: +2348038516786

Broadly, the TSP is classified as symmetric travelling salesman problem (sTSP), asymmetric travelling salesman problem (aTSP), and multi travelling salesman problem (mTSP). However, when there is a single salesman, then the mTSP reduces to the TSP [1].

The TSP has been applied to drilling of printed circuit boards and mask plotting in PCB production [2], to overhauling gas turbine engines [3], to analysis of the structure of crystals X-Ray crystallography [4 - 6] reported a special case of connecting components on a computer board, to the order-picking problem in warehouses. This problem is associated with material handling in a warehouse [7]; to vehicle routing [8].

The mTSP has been applied to printing press scheduling problem [9]; School bus routing problem [10] investigated the problem of scheduling buses as a variation of the mTSP with some side constraints; crew scheduling problem: An application for deposit carrying between different branch banks is reported in [11]. The application of the mTSP in mission planning is reported by [12 - 14].

There have been several types of approaches taken to solving the TSP. They include:

Memetic algorithms [15], Ant colony optimizations [16, 17], Simulated annealing [18,19], Genetic algorithms [20], Neural networks [21].

The proposed algorithm for solving the TSP in this work is Tabu Search Algorithm. Tabu Search Algorithm (TS) is a compact and robust technique, which provides excellent solutions to single and multiple objective optimization problems with a substantial reduction in computation time.

1.1 Traveling salesman Problem (TSP).

The TSP is a problem where a salesman is require to visit a list of n-city in a closed complete tour these are linked together at different distances and he is mandated to visit each cities only once. As the salesman tour across this cities it is ideal for him to search for the optimal route of traveling because it saves him less distance to cover, time and money.

The TSP can be defined on a complete undirected graph, H = (VE) if it is symmetric or on a directed graph, H = (VA) if it is asymmetric. The set V = {1,. . ., n} is the vertex set,

E = {(i,j): i ,j ∈ V, i <j} is an edge set and A = {(i,j): i ,j ∈ V, i ≠j} is an arc set.

A cost matrix C = (c_{ij}) is defined on E or on A. The cost matrix satisfies the triangle inequality whenever $c_{ij} \leq c_{ik} + c_{kj}$, for all i, j, k.

In particular, this is the case of planar problems for which the vertices are points

Pi = (X_i, Y_i) in the plane, and $c_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}$ is the Euclidean distance.

The triangle inequality is also satisfied if c_{ij} is the length of a shortest path from i to j on H.

1.2 Formulation of the Symmetric TSP (sTSP)

Many TSP formulations are available in literature, but here, we adopt the formulation in [22] which is as follows:

$$\min \sum_{j=1}^{m} \sum_{i=1}^{m} d_{ij} x_{ij} \qquad (1)$$

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

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

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

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

For the symmetric travelling 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 importance of the TSP is that it is representative of a larger class of problems known as combinatorial optimization problems [23]. 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 travelling 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 [24].

2.0 The Tabu Search Methodology

Tabu search (TS) was first published in the late 1980’s[25].

Tabu search uses a local or neighborhood search procedure to iteratively move from a solution to another solution until some stopping criterion has been satisfied. To explore regions of the search space that would be left unexplored by the local search procedure and by doing this escape local optimality, tabu search modifies the neighborhood structure of each solution as the search progresses.

2.1 Basic Tabu Search Algorithm

The most basic form of the Tabu search algorithm consists of the following:
1- A methodology for generating an initial solution.
2- A mechanism for generating a neighboring solution of the current solution.
3- A function that measures each neighboring solution.
4- A tabu list in order to prevent cycling and leads the search to unexplored regions of the solution space.
5- An aspiration criterion (if a tabu move satisfies the aspiration criterion, it is considered admissible).

2.2 General Framework of Tabu Search Algorithm

A general framework of tabu search is given below:
Begin
Step1: Initialization:
- Input the number of cities n, and their distances that are between them.
- Input the stop criterion it; {stopping after a number of iterations}.
-Set tabu tenure t; {is number of iteration for move will be kept tabu where t= sqrt(n)}.
- Initialize tabu list is empty; {tabu list = [position of city1 position of city2 ….. position of city n], therefore tabu list becomes = none

Step2: Use Greedy procedure to find best solution x* with objective function z* at the same time is consider as current solution x as follows:
Let x , current solution.
Let x* = x, the best solution.
Let z* = z(x), where z* = {summation of distance for x solution}.

Step3: Iteration:
While stop criterion is not satisfied Do
Begin
- Identify Neighborhood set by creating many solutions with used Move operation. The move operation in the TSP is making swap between any two cities randomly of the current solution x:
each x Î X has an associated Neighborhood N(x)Ì X.

Step3.1: Select best admissible move from N(x) by choosing the solution that has minimum z(x) from this is transforms x into x' Î N(X) and objective function value transforms to z(x') and add its attributes to the tabu list
Check x' is Tabu:
IF No Goto step 3.2
Else step 3.3

Step3.2: Perform exchange:
x = x' , z(x) = z(x').
If z(x) < z* then
z* = z(x) , x* = x.
End if
Go to step 4

Step 3.3 Check x' is Aspiration
IF Yes Goto step 3.2
Else
Check Neighbor set.

Step 4: Record tabu for the current move in tabu list, i.e. update the tabu list. By change the contents of tabu list as following:
- If solution not tabu add solution to tabu list by change the position of city from zero to t For example if the problem consist of 5 cities and solution that resulted from (swap city 4 &city 3) is not tabu the tabu list must be [0 0 0 t 0].in the next iteration the tabu tenure decreases by one.

End while
Step 5: Result: solution x* has length of tour must be minimum of all
determined solutions, with objective function value z*.
End

2.3 Initial Solution (Configuration)

The initial configuration specifies where the search begins in the search space. In this research to generate the initial solution we used Greedy algorithm, below Greedy algorithm:

1- Sort all edges.
2- Select the shortest edge and add it to our tour if it doesn't create a cycle with less than E edges, or increases the degree of any node to more than 2.
3- Do we have E edges in our tour? If No,repeat step2.

2.4 Neighborhood Structure

The neighborhoods they define contain all the feasible route configurations that can be obtained from the current solution. In neighborhood search such solution $x \in N$ has an associated set of neighbors, $N(x) \subset X$, called the neighborhood of x, each solution $x' \in N(x)$ can be reached directed directly from x by an operation called Move. And x is said to move to x' when such an operation is performed.

The neighborhood generally consists of solution obtained by making small changes in the previous solution. In the present research a Neighborhood configuration by making a single swap of current values of two randomly cities. There are many possible ways of selecting the best neighbor, including: first, choosing the first neighboring solution that improves objective function. Second, considering a subset of the neighbors and selecting the best of the set. Third, evaluating the whole set of neighboring solutions and selecting the best terms of the objective functions. The neighborhood is not a static set, but rather a set that can change according to the history of the history of the search.

In our implementation in the present research, second method is used for selecting the best neighbor.

2.5 Swapping

The swap move mechanism, which is used from this point onward, replaces a selected distance between two cities by another distance. Or the swapping operation neighborhood considers each possible exchange of two (cities) in the problem.

2.6 Aspiration Criterion

In order to override the tabu list when there is a good tabu move, aspiration criterion is used, the Tabu move is accepted if it produces better solution than the best obtained so far.

2.7 Termination Criterion

In most commonly used stopping criteria in tabu search are:

1- Terminate the search after a number of iterations.
2- After some number of iterations without an improvement in the objective function value.
3- When the objective reaches a pre-specified threshold value. In this research the first criteria is used for stopping criteria.

3.0 Numerical Experiment with Tabu Search (TS)

Using the travelling distance chart below involving six cities (Benin, Abuja, Calabar, Enugu, Ibadan, and Jos).We apply the Tabu Search heuristic to find the optimal tour distance.

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	758
Abuja2	493	0	729	393	645	297
Calabar3	490	729	0	276	518	870
Enugu 4	254	393	276	0	558	608
Ibadan 5	300	645	518	508	0	928
Jos6	758	279	870	608	928	0

4.0 Results and Discussion

This study is focus at creating a model for which the travel salesman can use to ensure an optimal tour distance in his Traveling salesman Problem. Considering the Traveling salesman problem as stated above, the optimal tour was implemented using a Visual Basic.Net programming language and the optimal tour was achieved within the shortest possible time.

The optimal route that was developed from the implementation is:

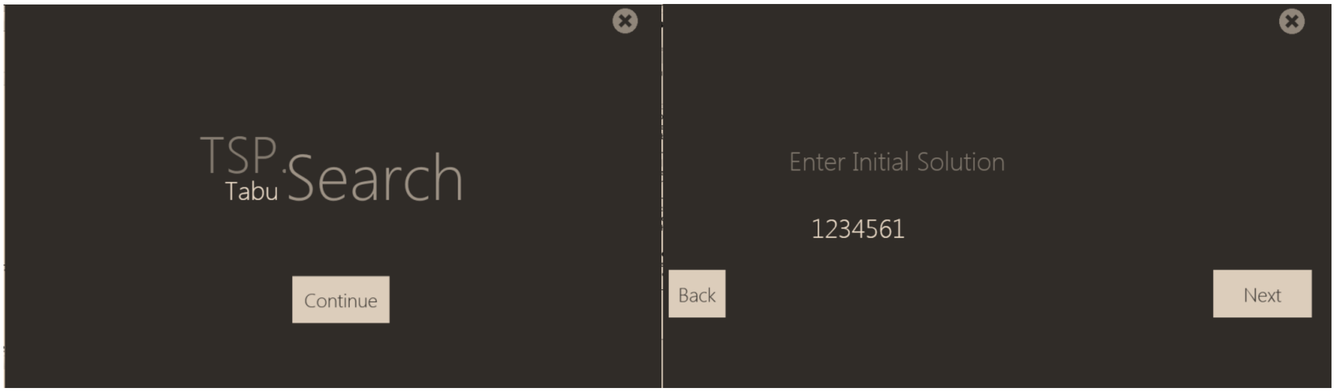
1 ⟶ 5 ⟶ 3 ⟶ 4 ⟶ 6 ⟶ 2 ⟶ 1

That is:

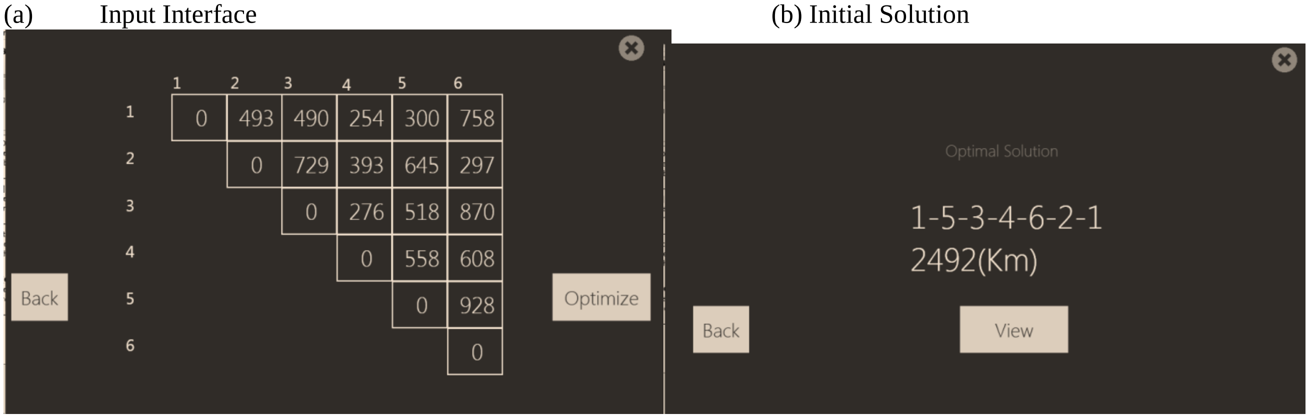
Benin ⟶ Ibadan ⟶ Calabar ⟶ Enugu ⟶ Jos ⟶ Abuja ⟶ Benin

(Benin - Ibadan) + (Ibadan - Calabar) + (Calabar - Enugu) + (Enugu - Jos) + (Jos - Abuja) + (Abuja- Benin).

Therefore the total distance: 300km + 518km + 218km + 608km + 297km + 493km = 2492 km.



(a) Input Interface (b) Initial Solution



(c.) **Travelling Distance Chart for Six Cities** (d) Final Solution

Fig.1: Solution to a 6 city problem

5.0 Conclusion

In this paper we briefly state the Traveling salesman Problem and explain the Tabu search heuristic with also an explicit method tosolve an optimization problem. In general the complexity of the Traveling salesman Problem increases with the number of routes (cities).This paper has shown how easy and effective a Traveling Salesman Problem can be optimize in a short time when implemented on a Visual Basic programming language.

6.0 References

[1] Bektas, T. (2006). The multiple travelling salesman problems: an overview of formulations and solution procedures. Omega, Vol. 34, pp. 206-219.

[2] Grötschel, M., Jünger, M. and Reinelt, G. (1991). Optimal Control of Plotting and Drilling Machines: A Case Study. Mathematical Methods of Operations Research, Vol. 35, No. 1, pp.61-84.

[3] Plante, R.D., Lowe, T.J. and Chandrasekaran, R. (1987). The Product Matrix Travelling Salesman Problem: An Application and Solution Heuristics. Operations Research, Vol. 35, pp. 772-783.

[4] Bland, R.E. and Shallcross, D.E. (1989). Large travelling salesman problem arising from experiments in X-ray crystallography: a preliminary report on computation. Operations Research Letters, Vol. 8(3), pp. 125-128.

[5] Dreissig W. and Uebaeh W. (1990). Personal communication. Linear programming with pattern constraints. PhD Thesis, Department of Economics, Harvard University, Cambridge, MA.
[6] Lenstra, J.K. and Rinnooy, A.H.G. (1974). Some Simple Applications of the Travelling Salesman Problem.BW 38/74, Stichting Mathematisch Centrum, Amsterdam.
[7] Ratliff, H.D. and Rosenthal, A.S. (1983). Order-Picking in a Rectangular Warehouse: A Solvable Case for the Travelling Salesman Problem. Operations Research, Vol. 31, pp. 507-521.
[8] Carter, A.E. and Ragsdale, C.T. (2002). Scheduling pre-printed newspaper advertising inserts using genetic algorithms. Omega, Vol. 30, pp. 415–21.
[9] Angel, R.D., Caudle, W.L., Noonan, R. and Whinston, A. (1972). Computer assisted school bus scheduling. Management Science, Vol. 18, pp.279–88.
[10] Svestka, J.A. and Huckfeldt, V.E. (1973). Computational experience with an m-salesman traveling salesman algorithm. Management Science, Vol. 19, No. 7, pp. 790–9.
[11] Yu, Z., Jinhai, L., Guochang, G., Rubo, Z. And Haiyan, Y. (2002). An implementation of evolutionary computation for path planning of cooperative mobile robots.
[12] Brummit, B. and Stentz, A. (1996). Dynamic mission planning for multiple mobile robots. Proceedings of the IEEE international conference on robotics and automation, April 1996.
[13] Brummit, B. and Stentz, A. (1998). GRAMMPS: A generalized mission planner for multiple mobile robots. Proceedings of the IEEE international conference on robotics and automation.
[14] Ryan, J.L., Bailey, T.G., Moore, J.T. and Carlton, W.B. (1998). Reactive Tabu search in unmanned aerial reconnaissance simulations. Proceedings of the 1998 winter simulation conference, Vol. 1, pp. 873–879.
[15] Dorigo, M. and Gambardella, L. (1997). Ant Colonies for the Travelling Salesman Problem. BioSystems, 43, 73-81.
[16] Amiolemhen, P.E. and Haruna J. I. (2013). Application of Ant Colony Optimization Algorithmto Solve Travelling Salesman Problem. Journal of the Nigerian Association of Mathematical Physics, Vol. 25, No.1. pp. 333 -338.
[17] Dorigo, M., and Gambardella, M. L. (1996). Ant Colonies for the Traveling Salesman problem. Technical Report TR/IRIDIA/1996, IRIDIA, Universite Libre de Bruxelles, Belgium.
[18] Amiolemhen, P.E. and Abikoye, B.O. (2013). Application of Simulated Annealing method to Solve Travelling Salesman Problem. Journal of the Nigerian Association of Mathematical Physics, Vol. 25, No.2. pp. 95 – 106.
[19] Kirkpatrick, S., Gelatt,C. D. and Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science, New Series, Vol. 220, No. 4598. pp. 671-680.
[20] Goldberg, D. (1989). Genetic Algorithms in Search, Optimization & Machine Learning. Addison Wesley.
[21] Matsuda, S., (1996). Set Theoretic Comparison of Mappings of Combinatorial Optimization Problems to Hopfield Neural Networks. Systems and Computers in Japan 27, 45-59.
[22] Papadimitriou, C.H. and Steiglitz, K. (1998). Combinatorial optimization: algorithms and complexity. Mineola, NY: Dover.
[23] Hoffman, A. J. and Wolfe, P. (1985), History in The Travelling Salesman Problem, Lawler, Lenstra, Rinooy Kan and Shmoys, eds., Wiley, 1-16.
[24] Oystein, M.H. (2002).A multi-level scheme for the travel salesman problem. PhD Thesis. University of Bersen, Prague, Czech Republic.
[25] Fred Glover (1986). Tabu search- Part 1,. ORSA Journal on Computing 1 (2), 190-206.