A QUICK HEURISTIC ALGORITHM FOR TRAVELLING SALESMAN PROBLEM

[1]Sk.Mastan, [2]U.Balakrishna, [3]G.Sankar Sekhar Raju
[1]Research Scholar, JNTUA, Ananthapuramu, India-515002
[2]Professor, SITAMS, Chittoor, India-517127
[3]Professor, JNTUA, Pulivendula,, India-516390

Abstract: The Optimization of a large-scales travelling salesman Problem (TSP) mostly in telecommunication networks that may be a well-known NP-hard downside in combinatorial improvement, may be a long downside. During this paper, the planned heuristic algorithmic program is intended for quick parameter, accuracy and computation time. planned algorithmic program has been compared with brute force associated hymenopterous insect colony improvement that referred to as an algorithmic program that's accustomed confirm the shortest path and best price at minimum iterations attainable for a random knowledge attack the premise of Euclidean space formula. Planned algorithmic program takes solely 0.0075 seconds to supply shortest path answer that sixty nodes combination. The planned algorithmic program has 6 June 1944 less accuracy from brute force and provides 5.59% higher answer for forty-four nodes through sixty nodes.

***Index Terms* - Travelling salesman problem; Shortest path value; ring topology; brute force; and colony; heuristic algorithm, the shortest route.**

I. Introduction

Travelling salesman problem is a problem that has widely observed even there is an accomplishment on robotic using Genetic algorithm and Heuristic Algorithm in TSP. The point of TSP in telecommunication networks are to find the smallest value of connected to form a ring topology. This ring topology will be considered to be used as the early design of a network. The TSP is how to determine the combination of vertices by Heuristic algorithm combination of vertices with smallest cost of each visited nodes and fast computational time[1].

In this paper, TSP problem with random assigned lint cost and symmetric values will be solved using proposed Heuristic Algorithm. Where price the worth of node A to node B is usually a similar because the value of node B to node A and not allowing loop. Therefore, the loop value is always is zero.

To solve the problem, we designed an algorithm and then write it into code in Python programming language. The performance of proposed Heuristic Algorithm is measure comparing it with other algorithm with two keys component, the total cost and the computational time.

II. Associated work

The total price worth smallest minimum to determine algorithmic program is thought as Brute force. each single doable combination and compares its worth is thought as Brute force. There are many other algorithms that have been developed to solve TSP problem, among them Ant colony Optimization. Now a days most generally used heuristic formula for determination TSP issues is hymenopterous. The Algorithm shaped is planned as simple as possible to aim the fastest computation.

III. Planned Procedure

The procedure planned is completed with the aim to find the best combination with minimum value to solve symmetrical TSP combination with minimum value. A Symmetrical TSP with fast computation time so the result can be implemented as a design for ring topology in telecommunication networks. The Proposed algorithm to find out the performance we compared the result with brute force and ACO algorithm. Proposed algorithm coded with Python 2.7 language and running on Ubuntu 16.04 platform. In the experiment finding algorithm to get the minimum weight or price or distance to make ring topology is by studying the gap or price matrix[2]. First approach is connecting some vertices with two minimum distance elements. Table 1 is 10 X 10 samples shown below.

Table 1 : 10 X 10 samples planned procedure

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8	g_9	g_{10}
g_1	0.0	8.1	9.2	7.7	9.3	2.3	5.1	10.2	6.1	7.0
g_2	8.1	0.0	12.0	0.9	12.0	9.5	10.1	12.8	2.0	1.0
g_3	9.2	12.0	0.0	11.2	0.7	11.1	8.1	1.1	10.5	11.5
g_4	7.7	0.9	11.2	0.0	11.2	9.2	9.5	12.0	1.6	1.1
g_5	9.3	12.0	0.7	11.2	0.0	11.2	8.5	1.0	10.6	11.6
g_6	2.3	9.5	11.1	9.2	11.2	0.0	5.6	12.1	7.7	8.5
g_7	5.1	10.1	8.1	9.5	8.5	5.6	0.0	9.1	8.3	9.3
g_8	10.2	12.8	1.1	12.0	1.0	12.1	9.1	0.0	11.4	12.4
g_9	6.1	2.0	10.5	1.6	10.6	7.7	8.3	11.4	0.0	1.1
g_{10}	7.0	1.0	11.5	1.1	11.6	8.5	9.3	12.4	1.1	0.0

Now we start initial node decided by finding two smallest nodes of each node horizontally on atrix and then write sum result of that two smallest nodes beside the matrix. After that the process is continued by comparing the sum result vertically and takes the smallest result as start. Assuming P is picked as initial, continue to finds two smallest vertices on node A horizontally and connecting g1 with that two smallest nodes. The figure shows the initial vertices only pick initial node:

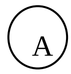


Figure 1 intial node

Figure 2(a) shows first connected vertices on both side:

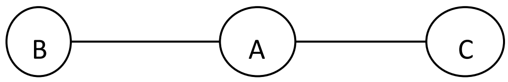


Figure 2(b) Initial node connected with two minimum vertices.

Next the entire step above there are three nodes connected already assuming g_1 wrote on the center, the two smallest nodes wrote on is left and right aspect. Next node should be determined connected to the left or right part. The decision made by counting the smallest node both side and compared the result and then the smallest node both side and compare result and then the smallest value is picked by discarding the bigger ones. After that, the counting is continued until the nodes remains three left. The last three nodes then counted with brute force by counting one by one and compare the result and the pick the minimum value. The pseudo code for proposed algorithm and the flowchart for TSP is shown below Figure 3:

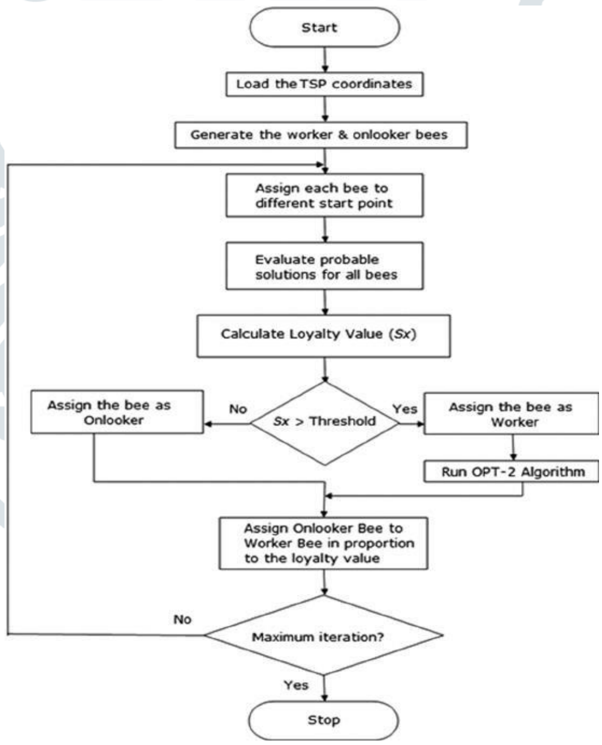


Figure 3:Algorithm-Proposed algorithm to solve TSP by Flowchart

IV. TSP HUERISTICS

Suppose to solve a TSP Heuristics that try all ring grouping to get a complete smallest path value. Proposed algorithm can solve for minimum nodes. From the data it can be concluded, the average minimum value error of proposed algorithm compared to brute force.

-	10	8	9	7
10	-	10	5	6
8	10	-	8	9
9	5	8	-	6
7	6	9	6	-

We begin nearest neighbor search five – one – three – four and one – three – four - two – five – one solution z =three four (feas) .The nearest neighbor search minimum spanning tree is as follows Figure 4:

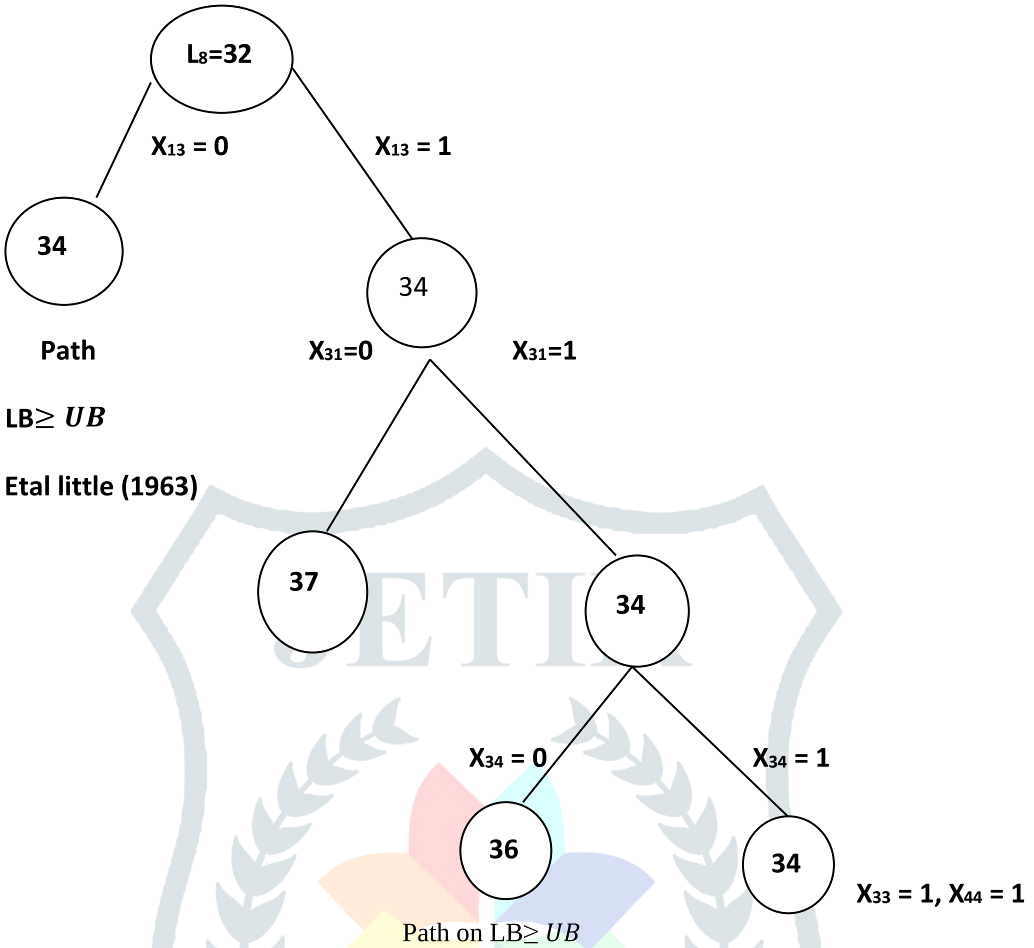


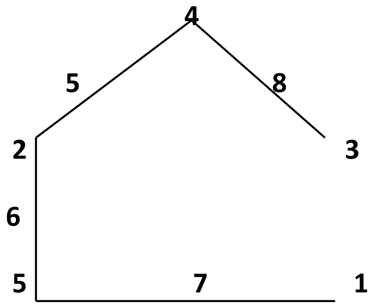
Figure 4: The nearest neighbor search minimum spanning tree.

Next the neighbor search one – three – four - two – five – one value z = three-four is complaints, exponential and non-polynomial. The TSP Heuristics are good solutions not exact polynomial time quick.
We calculate nearest neighbor searches as follows:
One – five – two – four – three – one is z = three -four, two – four – five – one -three – two is z = three -six three– one – five – two – four – three is z = three- four, four – two – five – one – three – four is z = three four, five – two – four – three – one is z = three four.

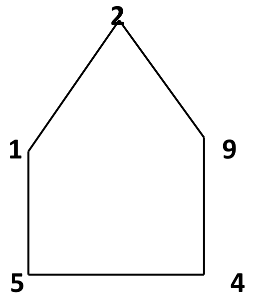
Feasible solution UB is $L_H \geq L_0 \rightarrow \frac{L_H}{L_0} \leq 1 + \frac{log_2}{2} \leq 2.5 \leq 4 \; at \; n = 16 \; and \; n = 128$

Perfect matching ≤ 1.5.

Three minimum spanning trees for nearest neighbor



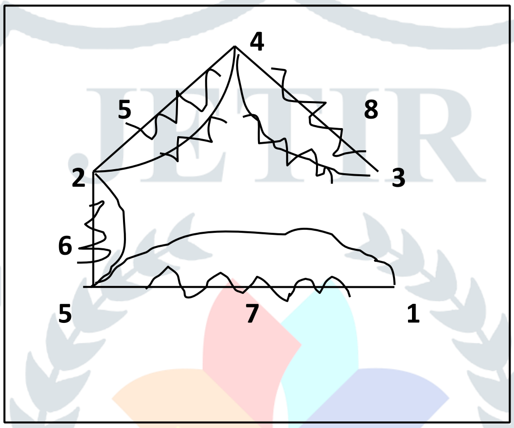
$C_{ij} \geq 0$;



Spanning tree 1 _ 2 _ 3 _ 4 _ 5 - 1

After that {2, 4} and {1, 3, 5} the spanning tree values are LMST =26; LST $\leq L_0 \rightarrow L_{MST} \leq L_{ST} \leq L_0 = L_{MST} = 26$.

Now another Minimum spanning tree is

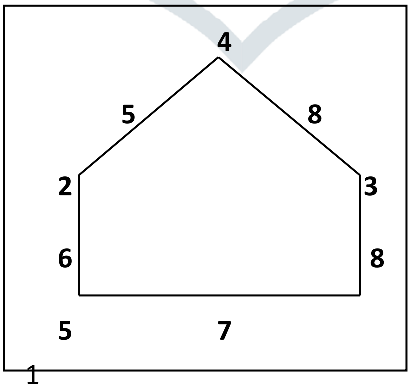


According to minimum spanning tree connective is one – five – two – four – three – four – two – five – one i.e. Le = 52

And one – five – two – four – three - one is L = three-four; one – two – four – three – five – one is L = thirty-nine.

One – four – three – two – five – one is L = four-zero; one – three – four – two – five – one is L = thirty-four [3].

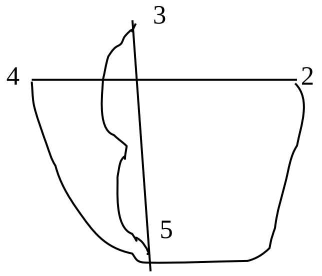
The values of $L_{MST} \leq 40$; le = two L_{MST} and $L_H \leq le \rightarrow L_H \leq L_{MST} \leq 2L0 \rightarrow \frac{LH}{LO} \leq 2$.



The higher than spanning tree connective is one – five – two – four – three – one is z = thirty-four
The minimum weighted matching two – three two – four two – five four – five three - five three - four autoimmune disease = LMST + Ladd

2	3	4	5
-	10	5	6

	10	**-**	**8**	**9**
2	**5**	**8**	**6**	**-**
3	**6**	**9**	**6**	**-**
4				
5				



The minimum spanning tree higher than connective is one – a pair of – four – one - three – five – one.

IV. CONCLUSION

The projected algorithmic rule achieves higher shortest path answer compared with pismire colony improvement on differing types of nodes. Projected algorithmic rule will resolve shortest diagram of spanning trees. During this paper we tend to develop algorithms of spanning trees.

REFERENCES

[1] Supriadi, B, Budi, B., "Comparison of Proposed 'Algorithm to Ant Colony Optimization Algorithm to from a Ring Topology with Minimum Path value". IEEE ICRAIE December – 2016.

[2] Baranwal, M, Roehl, B., Salaka, S.M., Multiple Travelling Salesmen and Related Problems A Maximum-Entropy Principle based Approach," USA, May – 2017.

[3] Dorigo, Margo, Gambardella, L.M., "Ant Colony system A cooperative learning approach to the representative drawback," IEEE Transctions on biological process Computation, Vol.1, No:1, April 1997.