ISSN 2229-5518

A Hamiltonian Approach for a Traveling Salesman

Shyamala Venkatraman

Abstract— In this modern era, various applications of Hamiltonian's graph have come into existence and serve as very good measures over a large class of optimization problems. The Travelling Salesman Problem (TSP) is a problem in combinatorial optimization studied in operation research and theoretical computer science. Given a list of cities and their pair wise distances, the task is to find a shortest possible tour that visits each city exactly once. This paper analyzes various types of algorithms such as The nearest Neighbor Algorithm to find a (reasonably good) Hamiltonian cycle, Lower Bound Algorithm to find a lower bound for a Hamiltonian cycle , Tour Improvement Algorithm to look for possible improvement in the tour etc. In this research paper, we have taken a locality in Meghalaya state which is under development, having lesser main resources viz. roadways, telecome links etc.

Index Terms— Branch and Bound Algorithm, Brute-force Algorithm, Hamiltonian's Graph, Help-Karp Algorithm, Lower Bound Algorithm, Metric Approximation Algorithm, The Nearest Neighbor Algorithm, The Tour Improvement Algorithm.

1 Introduction

Hamiltonian circuit [2] in a connected graph is defined as A graph G [1] consists of two sets V and E, V is a finite non-empty set of vertices, E is a set of pair of vertices; these pairs are called edges. In an undirected graph the pair of vertices representing any edge is unordered. Thus, the pairs (v_1, v_2) and (v_2, v_1) represent the same edge. In a directed graph each edge is represented by a directed pair (v_1, v_2). v_1 is the tail and v_2 the head of the edge. Therefore (v_2, v_1) and (v_1, v_2) represent two different edges. A graph G is said to be connected [2] if there is at least one path between every pair of vertices in G. Otherwise, G is disconnected [2].

Hamiltonian circuit [2] in a connected graph is defined as a closed walk that traverses every vertex of G exactly once, except of course the starting vertex, at which the walk also terminates.

A circuit in a connected graph G is said to be Hamiltonian if it includes every vertex of G. Hence a Hamiltonian circuit in a graph of n vertices consists of exactly n edges.

Hamiltonian Path: If we remove any one edge from a Hamiltonian circuit, we are left with a path. This path is called a Hamiltonian path [2]. Clearly, a Hamiltonian path in a graph G traverses every vertex of G. The length of a Hamiltonian path (if it exists) in a connected graph of n vertices is n-1.

The Nearest Neighbor Algorithm: To find a (reasonably good) Hamiltonian cycle i.e. a closed trail containing every node of a graph.

Step1: Choose any starting node
Step2: Consider the arcs which join the node just chosen to nodes as yet not chosen. Pick the one with minimum weight and add it to the cycle.
Step3: Repeat step2 until all nodes have been chosen.
Step4: Then add the arc that joins the last chosen node to the first-chosen node.

The Lower Bound Algorithm: To find a lower bound for a travelling salesperson problem.

Step1: Pick any node and remove the two connecting arcs with least weight.
Step2: Find the minimum spanning tree for the other nodes using Prim's algorithm.
Step3: Add back in the two arcs removed previously.
Step4: The weight of the resulting graph (which may not be a cycle) is a lower bound i.e. any optimum solution must have at least this weight.

The Tour Improvement Algorithm: To look for possible improvement in a tour found by the nearest neighbor algoithm

Fig. 1 Tour Improvement

Step1: In Fig. 1, number the nodes in the order of the tour: start at node 1
Step2: Consider just the part of the tour 1-2-3-4.
Step3: Swap the middle nodes to change the order to 1-3-2-4.
Step4: Compare the two and keep the order with the lowest weight.
Step5: Move on to node2 and repeat until each node has been the start node once

2 Travelling Salesman's Problem

The Travelling Salesman's Problem [2] describes a salesman who must travel between N cities. He has to visit each city only once during his trip, and finishes where he was at first. Each city is connected to other close by cities, or nodes, by airplanes, or by road or railway. Each of those links between the cities has one or more weights (or the cost) attached.

ISSN 2229-5518

For example, cost of an airplane ticket or train ticket, or length of the edge, or time required to complete the transversal. The salesman wants to keep both the travel costs, as well as the distance he travels as low as possible.

For the present problem, the Jowai town in Jaintia Hills of Meghalaya state and some localities in Jowai are taken into account.

3 TRAVELLING SALESMAN'S PROBLEM IN JOWAI

-

Fig. 2 Road connectivity in and around Jowai town

A- Jowai Town
1- Thadlaskein
2- Laskein
3- Saipung
4- Khliehriat
5- Amlarem
6- Lum Kyndong
7- Panaliar
8- Chutwakhu
9- Mission Compound
10-Dulong
11-Khimusniang
12-Tpep Pale
13-Iaw Musiang
14- Mynthong
15- New Hill
16-Ladthadlaboh
17-Caroline Colony
18-Wah Nangbah
19-Mihmyntdu
20-Shangpung

Table(1) Localities

Fig. 2 shows the road connectivity between Jowai town (A) and 20 villages in and around Jowai town. There is no proper road connectivity between some places.

LOCATIONS HAVING NO DIRECT ROUTE

Locations	Possible Route
1 and 2	1-A-20-2
1 and 3	1-20-4-3
1 and 4	1-A-4
2 and A	2-20-A
2 and 3	2-20-4-3
2 and 5	2-20-A-5
2 and 8	2-20-A-8
2 and 10	2-20-A-10
2 and 17	2-20-A-17
2 and 18	2-20-18
2 and 19	2-20-19
3 to 10	3-4-10
3 to 11	3-4-A-11
3 to 19	3-4-19
3 to 20	3-4-20
5 to 20	5-A-20

Table(2)

WALK ONLY ROUTE

Location	Distance (in metres)	Time required (in minutes)
6 – 7	550	8
6-10	450	5
7-8	950	13
7-9	800	10
7-10	450	6
8-9	600	10
8-13	500	6
8-14	600	7
9-12	1400	10
9-14	900	11
12-13	350	4
13-14	700	9
16-A	1800	21
16-17	650	8

Table(3)

For a salesman to make a tour from one place to another, either he has to cover a particular distance through national high way and then the remaining distance by foot or to cover the whole distance by walk. There is no direct route between certain locations. Table (2) gives the detail of such locations and one possible route Table (3) gives the 'walk only' route together with the distance to be walked and time required

Any tour from Laskein or Saipung to any other place cannot be a closed tour

If a closed tour is planned between Jowai and the other locations, the one may be the tour A-1-17-14-8-13-15-16-12-9-7-10-6-5-4-19-20-18-A

Such a tour is given in figure 3

Fig. 3 A closed with 18 locations

4 SELECTION OF A TOUR

Tour-1 starting from Jowai, visiting each location (excluding 2 and 3) only once and finishing at Jowai is planned as A-1-17-16-15-13-14-8-9-12-11-10-7-6-5-4-19-20-18-A

In Fig. 4, distance is marked in black and time is marked in red.

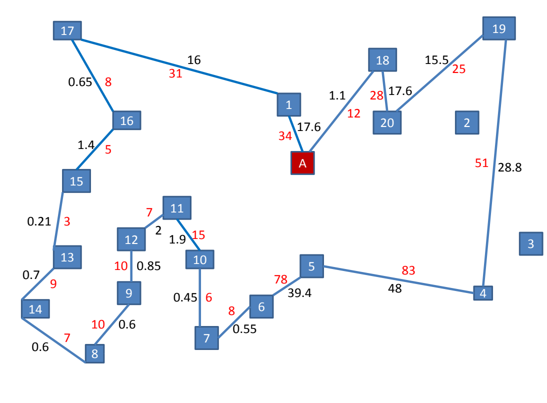

Fig. 4 Tour 1

The tour can be completed in 34+31+8+5+3+9+7+10+10+7+15+6+8+78+83+51+25+28+12=430 minutes or 7 hrs and 10 minutes.The distance to be travelled by vehicle is 17.6+16+1.4+2+1.9+39.5+48+28.8+15.5+17.6+1.1= 189.4 km. The routes 17-16, 15-13-14-8-9-12, 10-7-6 are to be covered by walk. The total distance to be walked is 4.6 km. Total distance to be travelled is 194 km.

Tour-2 starting from Jowai, visiting each location (excluding 2 and 3) only once and finishing at Jowai is planned as A-17-14-13-15-16-12-9-8-11-10-7-6-5-4-19-20-18-1-A

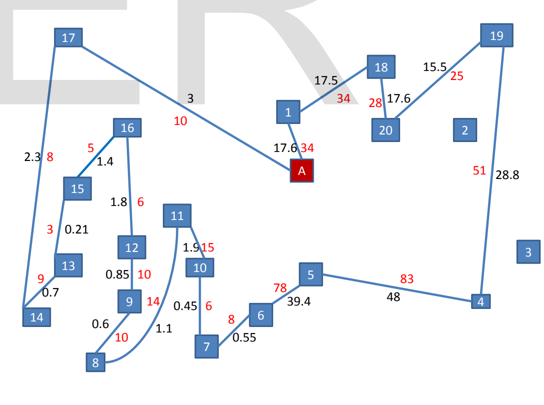

Fig. 5 Tour2

The tour can be completed in 10+8+9+3+5+ 6+10+10+14+15+6+8+78+83+54+25+28+34+34 = 437minutes (or 7 hours and 37 minutes). The distance to be covered by vehicle is 3+2.3+1.4+1.8+ 1.1+1.9+39.4+48+28.8+15.5+17.6+17.5+17.6 = 195.9 km.

The routes 14-13-15, 12-9-8 and 10-7-6 are to be covered by walk. The total distance to be walked is 3.35 km.

Total distance to be travelled is 199.25 km.

5 COMPARISON OF TOUR1 AND TOUR2

	Time required (minutes)	Distance covered by vehicle (km.)	Distance covered by walk (km.)	Total distance (km.)
Tour 1	430	189.4	4.6	194
Tour 2	437	195.9	3.35	199.25

Table(4)

On comparing Tour 1 and Tour 2, time required in tour 2 is greater than that of tour 1 by 7 minutes, the distance covered by vehicle is greater by 6.5 km and the total distance covered is greater by 5.25 km. But distance covered by walk is lesser by 1.25 km

On the basis of time required and total distance travelled, Tour 1 may be the better one but if distance to be walked is considered, priority will be given to tour 2.

Hence in places where the walking distance is greater than the distance to be covered by vehicle and there is no proper road connectivity, the tour planned by using the nearest neighbor algorithm may not be a preferable choice.

If the tour improvement algorithm is used for improving a tour, at times, the exact solution coincides either with a tour with less time, less distance by vehicle and more distance by walk or that over a tour with more time, more distance by vehicle but less distance by walk or further improvement may not be possible.

6 VERIFICATION OF NEAREST NEIGHBOR ALGORITHM

The Nearest Neighbor Algorithm:

Starting from A the choices of routes can be as in Fig. 6

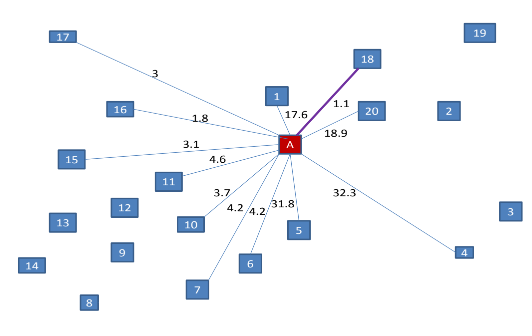

Fig. 6Possible routes from A

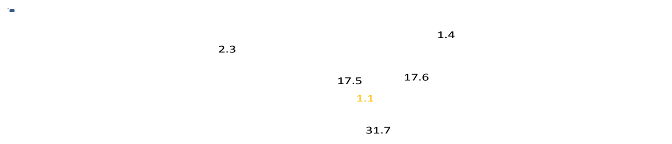

Fig. 7 Possible routes from 18

The first route selected is from A to18 since 1.1 km is the least. Route from 18 to 19 is chosen as 1.4 km is the least as in Fig. 7

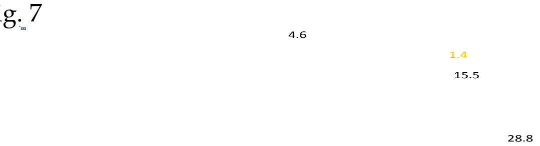

Fig. 8 Possible routes from 19

Then route from 9 to17 is chosen as 4.6 km is the least from Fig. 8

Next, route from 17 to 15 having a least distance of 1.5 km is chosen as in Fig. 9

Fig. 9 Possible routes from 17

Further selection of routes proceed as given in Table (5)

SELECTED ROUTES

Route	Distance (km.)	Figure
From 15 to13	0.21	Fig. 10
From 13 to 12	0.35	Fig. 11
From 12 to 10	0.35	Fig. 12
From 10 to 6	0.45	Fig. 13
From 6 to7	0.55	Fig. 14
From 7 to 9	0.80	Fig. 15
From 9 to 8	0.60	Fig. 16
From 8 to 14	0.60	Fig. 17

Table (5)

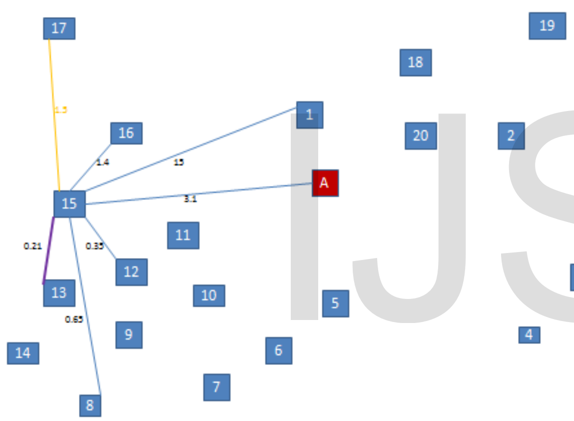

Fig. 10 Possible routes from 15

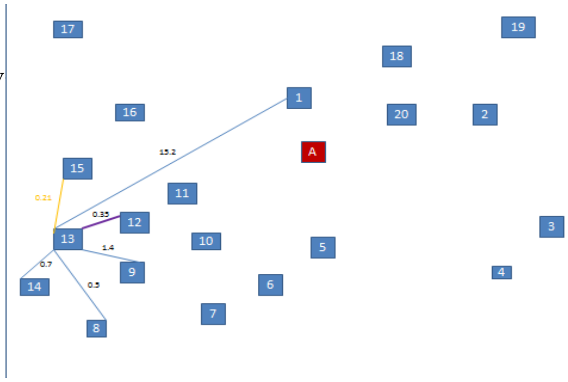

Fig. 11 Possible routes from 13

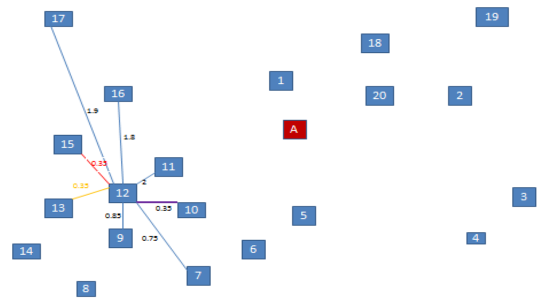

Fig. 12 Possible routes from 12

Route from 12 to15 cannot be considered since location 15 is already visited

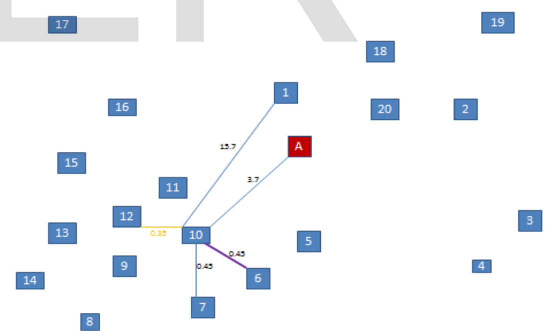

Fig. 13 Possible routes from 10

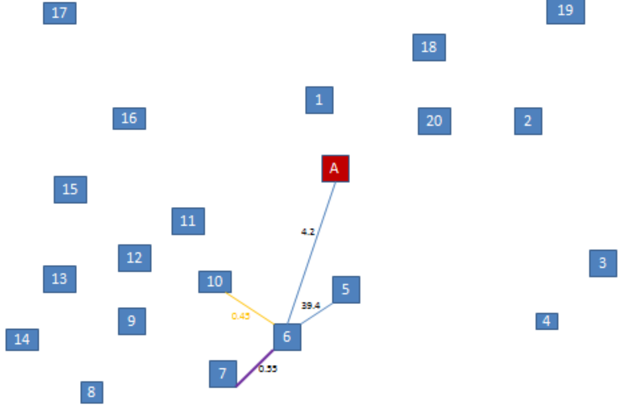

Fig. 14 Possible routes from 6

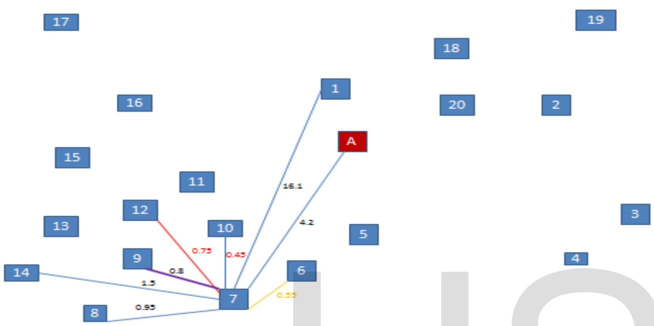

Fig. 15 Possible routes from 7

Choosing the route from 7 to12 will lead to an incomplete closed tour

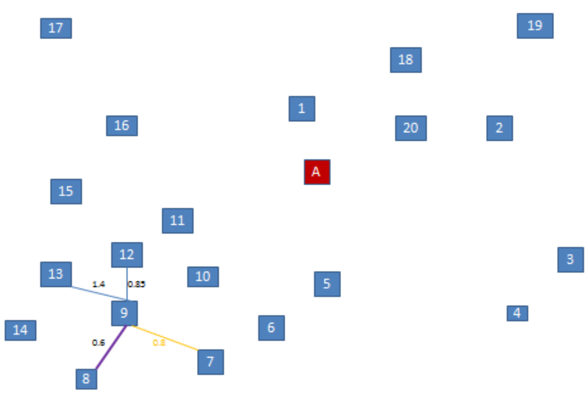

Fig. 16 Possible routes from 9

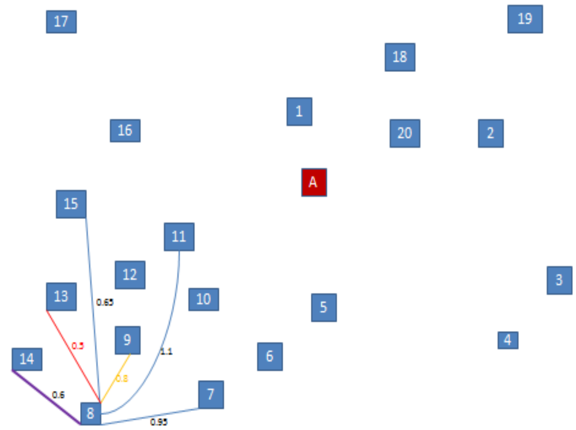

Fig. 17 Possible routes from 8

Route from 8 to 13 though shortest, cannot be chosen since location 13 is visited already

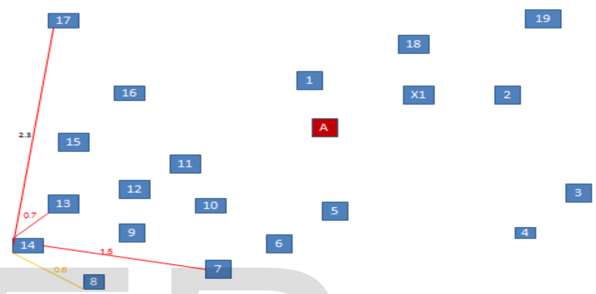

Fig. 18 Possible selections from 14

After visiting 14, there is no way to go to a place which has not been visited before.

Hence, getting a Hamiltonian Cycle covering all the places is not possible since the locations 1,4,5,11,16 and 20 are not visited

Now, suppose, route from 10 to 7 is chosen instead of route from 10 to 6, then the tour becomes complete after 14-1but the two places 4 and 20 are not visited since there is no direct route from 1 to 4 as well as from 1 to 20. Hence, in this case also, getting a Hamiltonian Cycle covering all the places is not possible.

Hence Nearest Neighbor Algorithm fails to give a solution

On applying the Tour Improvement Algorithm for Tour1 the improved tour becomes:

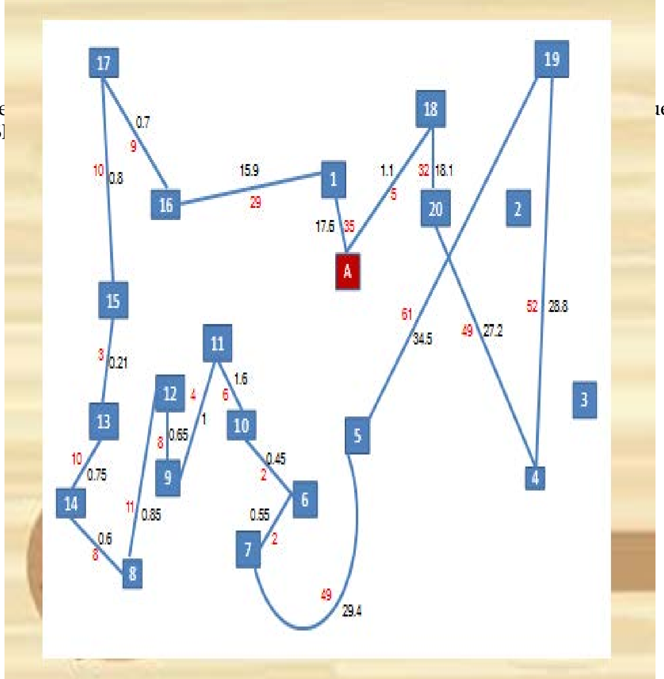

Fig. 19 Tour Improvement Algorithm for tour1

- Considering the tour-1 : A-1-17-16-15-13-14-8-9-12-11-10-7-6-5-4-19-20-18-A

  Pick A as node1 and consider the tour from A-1-17-16
- This has distance 34.3 kms
- Now swap the middle two nodes
- And you get A-17-1-16
- This also has weight 34.3 kms
- So keep the same tour. Now look at 1-17-16-15.
- 1-17-16-15 has distance 18.1 kms and 1-16-17-15 has distance 17.4 kms
- So swap 17 and 16 and the tour becomes A-1-16-17-15-13-14-8-9-12-11-10-7-6-5-4-19-20-18-A
- Now look at 16-17-15-13
- 16-17-15-13 has distance 1.71 kms. and 16-15-17-13 has distance 5.3 kms
- So leave the tour as A-1-16-17-15-13-14-8-9-12-11-10-7-6-5-4-19-20-18-A
- Now look at 17-15-13-14
- 17-15-13-14 has distance 1.76 kms and 17-13-15-14 has distance 2.7 kms
- So leave the above tour.
- Now look at 15-13-14-8
- 15-13-14-8 has distance 1.56 kms and 15-14-13-8 has distance 2.05 kms.
- So leave the above tour.
- Now look at 13-14-8-9.
- 13-14-8-9 has distance 1.64 kms. and 13-8-14-9 has distance 1.95 kms.
- So leave the same tour
- Now look at 14-8-9-12
- 14-8-9-12 has distance 1.54 kms and 14-9-8-12 has distance 1.99 kms.
- So leave the same tour
- Now look at 8-9-12-11
- 8-9-12-11 has distance 2.94 kms and 8-12-9-11 has distance 2.5 kms.
- This gives a tour which has a length of 186.15 kms in which 4.35 kms are to be covered by walk and 181.80 kms using vehicle.
- The total time required to cover the tour is 389 minutes.
- So swap 9 and 12 and the tour becomes A-1-16-17-15-13-14-8-12-9-11-10-7-6-5-4-19-20-18-A
- Now look at 12-9-11-10
- 12-9-11-10 has distance 3.25 kms and 12-11-9-10 has distance 3.55 kms
- So keep the above tour
- Now look at 9-11-10-7
- 9-11-10-7 has distance 3.05 kms and 9-10-11-7 has distance 4.25 kms
- So keep the same tour
- Now look at 11-10-7-6
- 11-10-7-6 has distance 2.6 kms and 11-7-10-6 has distance 3 kms.
- So keep the same tour
- Now look at 10-7-6-5
- 10-7-6-5 has distance 30.6 kms and 10-6-7-5 has 30.4 kms.
- So swap 7 and 6 and the tour becomes A-1-16-17-15-13-14-8-12-9-11-10-6-7-5-4-19-20-18-A
- Now look at 6-7-5-4
- 6-7-5-4 has distance 80.25 kms and 6-5-7-4 has distance 93.6 kms.
- So keep the above tour.
- Now look at 7-5-4-19
- 7-5-4-19 has distance 108.5 kms and 7-4-5-19 has distance 119.4 kms
- So keep the same tour.
- Now look at 5-4-19-20
- 5-4-19-20 has distance 95.1 kms and 5-19-4-20 has 90.5 kms.
- So swap 4 and 19 and the tour becomes A-1-16-17-15-13-14-8-12-9-11-10-6-7-5-19-4-20-18-A

ISSN 2229-5518

- Now look at 19-4-20-18
- 19-4-20-18 has distance 74.1 kms and 19-20-4-18 has distance 74.2 kms
- So keep the above tour
- Now look at 4-20-18-A
- 4-20-18-A has distance 46.4 kms and 4-18-20-A has distance 68.6 kms
- So keep the same tour .
- We have now had each node at the front and so we stop.
- The final tour is A-1-16-17-15-13-14-8-12-9-11-10-6-7-5-19-4-20-18-A
- In this improved tour, the total distance to be travelled is 180.16 kms in which 174.6 kms are to be travelled by vehicle and 5.56 kms need to be covered by walk.
- The tour can be completed in 385 minutes
- Like Tour1, In case of Tour2 also, the Application of Tour Improvement Algorithm gives an improved tour as the one follows:

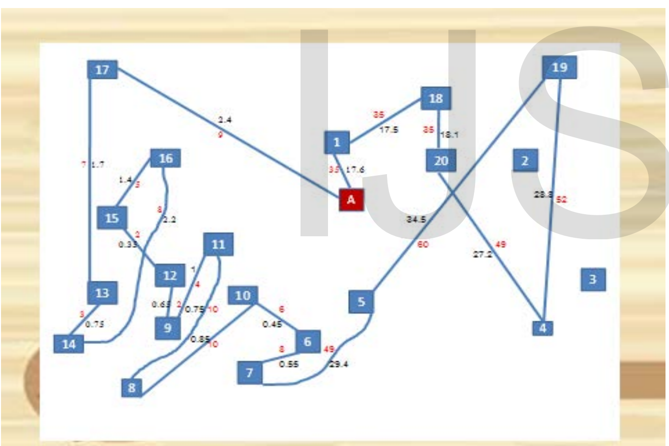

Fig. 20 Tour Improvement Algorithm for tour2

This gives a tour which has a length of 186.15 kms in which 4.35 kms are to be covered by walk and 181.80 kms using vehicle.

The total time required to cover the tour is 389 minutes.

On Comparing the improved tours for tour 1 and tour 2,time required in tour 2 is greater than that of tour 1 by 4 minutes, the distance covered by vehicle is greater by 7.2 kms and the total distance covered is greater by 5.99 kms. But distance covered by walk is lesser by 1.21 kms

	Time required (minutes)	Distance covered by vehicle (kms.)	Distance covered by walk (kms.)	Total distance (kms.)
Tour 1	385	174.6	5.56	180.16
Tour 2	389	181.80	4.35	186.15

Table(6)

Though the total distance to be covered and time required to cover the distance are considerably reduced in the improved tours the distance to be walked has been increased by 0.96 kms in case of Tour1 and 1km in case of Tour2

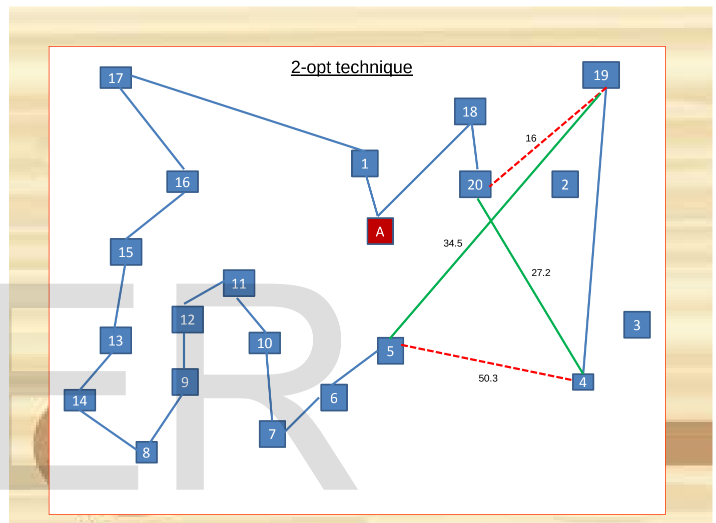

Fig. 21 Application of 2-opt technique

. When we apply the 2-opt technique with 5-4 and 19-20 replaced by 5-19 and 4-20, there is a little improvement that the distance to be travelled is reduced by (16+50.3)-(27.2+34.5)=66.3-61.7=4.6 kms and the time required is reduced by 113-109=4 minutes

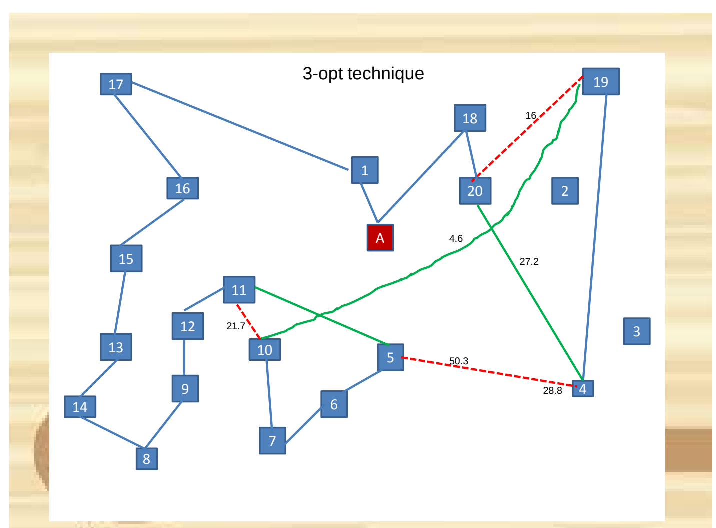

Fig. 22 Application of 3-opt Technique

After applying 3-opt technique, it is observed that the distance to be travelled is reduced by

(16+50.3+1.6)-(29.2+4.6+27.2)=67.9-61=6.9kms and the time required is reduced by 123-115=8 minutes.

Here, 11-10, 5-4, and 19-20 are being replaced by 11-5, 10-19 and4-20

- Kruskal's Algorithm

  Step 1: Find the cheapest edge in the graph (if there is more than one, pick one at random). Mark it with any given colour, say red.
  Step 2: Find the cheapest unmarked (uncoloured) edge in the graph that doesn't close a coloured or red circuit. Mark this edge red.
  Step 3: Repeat Step 2 until you reach out to every vertex of the graph (or you have N ; 1 coloured edges, where N is the number of Vertices.) The red edges form the desired minimum spanning tree.

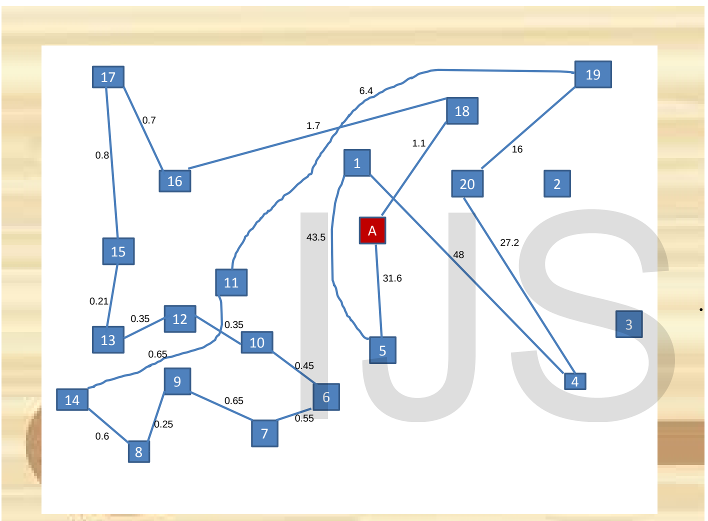

Fig. 23 Application of Kruskal's Algorithm

On applying the Kruskal's Algorithm, a tour is formed in which the total distance to be travelled is 180.36km.

After analyzing Nearest Neighbour Algorithm, Tour Improvement Algorithm, 2-opt technique, 3-opt technique, Insertion Algorithm and Kruskal's Algorithm, it is observed that the Tour Improvement Algorithm gives the best tour

References

[1] Ellis Horowitz and Sartaj Sahni, "Fundamentals Data Structures in Pascal" Galgotia Booksource, New Delhi, pp. 272-276, 1996

[2] Narsingh Deo, Graph Theory with applications to Engineering and Computer Science,Prentice-Hall of India Private Limited, New Delhi 1997, pp.19-35.