Implementation of Charging Station Based Electric Vehicle Routing Problem Using Nearest Neighbour Search Algorithm

Abdur Rahman Daanish
Department of Electrical Engineering
Indian Institute of Technology (ISM)
Dhanbad, Jharkhand, India-826004.
e-mail: arahmandanish@gmail.com

B Krishna Naick
Department of Electrical Engineering
Indian Institute of Technology (ISM)
Dhanbad, Jharkhand, India-826004.
e-mail: krishnalalitha.b@gmail.com

Abstract—**Over the past few years, electric vehicles are gaining recognition as a feasible alternative for the present day automobiles. Today's environmental scenario has instigated the authorities and entities to reduce the usage of fuel. This has initiated a surge in the market share of electric vehicles. In this paper, a Nearest Neighbour Search-based optimal routing of electric vehicles has been presented with charging stations present in between and at the nodes.**

Keywords-vehicle routing problem; electric vehicle; nearest neighbour

I. Introduction

Over the past few decades, there has been a huge impact of combustion of fuel on the earth's atmosphere. The burning of hydrocarbons is a major factor in the depletion of ozone layer, green-house effect and many more environmental hazards. Due to these dangerous phenomena, there has always been a push to reduce the use of fuel in automobiles. This gave a spark to the idea of electrically powered vehicles or Electric Vehicles popularly known as EVs. A normal vehicle uses petrol or diesel as fuel but an Electric Vehicle (EV) uses a battery or any other electric source as fuel.

In 2013, transportation contributed more than 50% of the nitrogen oxide and carbon monoxide, and almost 25% of the hydrocarbons were emitted into air. This air pollution carries significant risks for human health and the environment. Through EV and latest industrial accomplishments, the air pollution caused due to the transportation vehicles may be reduced incomparably.

EV is the future means of transportation with minimal emissions and is also a boon to the earth's atmosphere. It uses battery as a fuel which is rechargeable. Thus, it needs charging nodes instead of petrol pump for recharging the battery. The use of a battery instead of an Internal Combustion(IC) engine makes the vehicle extremely simple, compact and robust. Many countries like China have implemented electrically powered personal vehicles. EVs are predicted to outnumber the fossil fuel powered automobiles in the next decade. Hence, it is the utmost priority to devise ways to optimize the path of an EV. Many algorithms have been proposed for the aforesaid. Dijkstra's algorithm is one of them which has been briefly discussed in this paper.

The most vital element of EVs are battery, motor and charging stations because electricity is the only driving power in the EVs as compared to the IC Engine Vehicles [1].

Using various vehicle parameters like Aerodynamic drag coefficient , Frontal area, Rollin Resistant coefficient, Vehicle mass, Gravitational acceleration, Air mass density, Rotational Inertia factor (Mass factor) and Regenerative Braking Factor, the rate of discharge of the battery may be calculated and used in this algorithm[2].

The Nearest Neighbour (NN) is usually chosen for the construction of a route. However, choosing the next node only considers the current node, which leads to the final route which may not be closed geographically [3].

II. Dijkstra's Algorithm

Dijkstra's algorithm is an algorithm for finding the shortest path between two different nodes in a graph, which may represent, for example, roads. The starting node is called the initial node. The distance of node *Y* is the distance from the initial node to *Y* [4].

- A value is set to every node which corresponds to its tentative distance. It is set to infinity for all other nodes except the initial node which is set to zero.
- The source node is the current node. The other nodes are set as untraversed. A set of all the untraversed nodes is called the *untraversed set*.
- For the current node, all of its unvisited neighbours are considered and their *tentative* distances are calculated. The newly calculated *tentative* distance is compared to the current assigned value and the smaller one is assigned.
- When all of the neighbours of the current node are considered, then the current node is marked as visited and is removed from the *untraversed set*.
- If the destination node is reached (when shortest route between two nodes is calculated) or if the smallest tentative distance among the nodes in the *untraversed set* is infinity (When a complete traversal is required from the source or the initial node), then the algorithm has finished.
- If the above condition is not true, the untraversed node is selected and is marked the "current node", and all the steps from step 3 will be repeated.

978-1-5090-6273-7/17/$31.00 ©2017 IEEE

III. Implementation of Nearest Neighbour Search Algorithm

The input to the program consisted of a data file which contained the coordinates of the cities to be traversed. The number of cities in this case was taken to be 32. Along with these, the file also contained the co-ordinates of charging stations between the cities. It was further assumed that every city has a charging station. In a network, the distance between two nodes may be defined as the minimum of the summation of weights on the shortest path joining the two nodes [5].

This Algorithm provides flexibility as less constraints are taken into account in the proposed method. The protocol involved is dynamic and apt for the routing problem. Nearest Neighbour Search accounts for greater maneuverability. On the other hand, algorithms like Particle Swarm Optimization [6] and Ant Colony Optimization [7] involve more constraints and are less flexible.

A. *Distance Matrix Calculation*

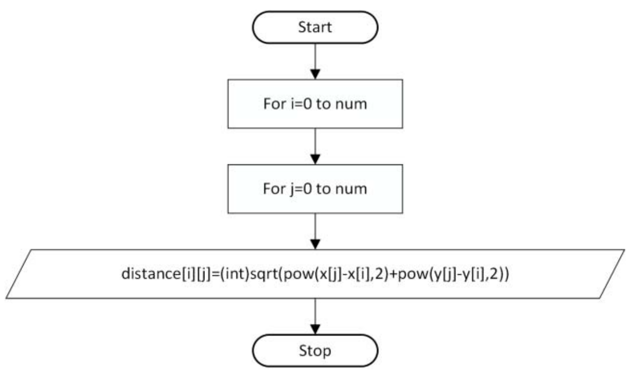

Figure 1. Flow Chart for Distance Matrix Calculation.

Where num = number of cities

The Distance matrix is a square matrix in which the number of rows and columns is equal to the number of cities. Now, to compute the distance matrix, the following algorithm was implemented:-

- The coordinates of the i^{th} city was supposed to be (x_i, y_i).
- Distance between two cities i and j was calculated = $((x_j - x_i)^2 + (y_j - y_i)^2)^{1/2}$
- The elemental distance *(i, j)* of the distance matrix may be represented as the distance between i^{th} and j^{th} city.
- The diagonal elements of the distance matrix are zero because a diagonal element (suppose a_{ii}) of the distance matrix means distance between city i and city i. Since both the cities are same, the distance between them will be zero.

B. *Searching the Nearest Node*

While traversing the cities, the city nearest to the source city was found by comparing the distance of all the cities and the smallest distance is found using the distance matrix. Then, the next nearest city is found by the same technique as used above but this time the source city and the first nearest city are not considered because cities should be traversed only once! This is repeated until we reach the last city untraversed.

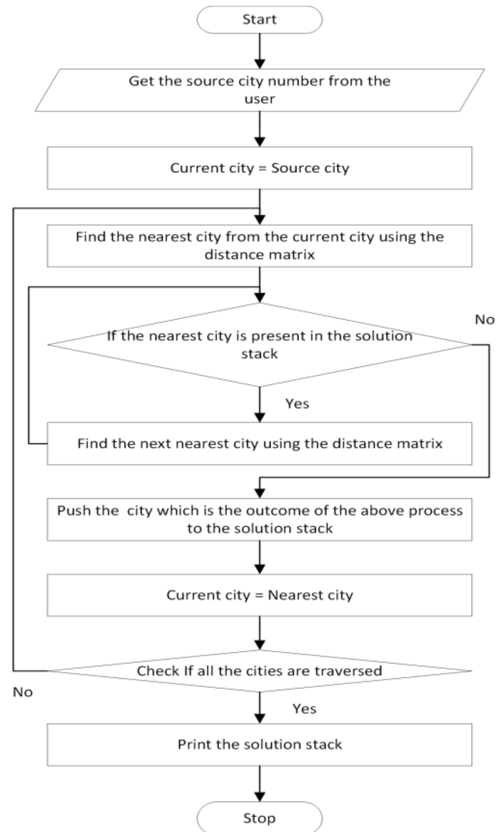

Figure 2. Flow Chart for Generating the solution.

C. *Charging Stations*

It may be noted earlier that EVs were taken into consideration for solving the problem and it requires battery as fuel to operate. Since, the battery discharges on being used, charging stations were supposed to be installed between the cities and at the cities. In the problem addressed, it was assumed that every city has a charging station. A charging station matrix *station (i, j)* was used which represents the number of charging stations present between the cities *i* and *j*. The diagonal elements of this matrix were equal to one a diagonal element (suppose a_{ii}) of the charging station matrix means the number of charging stations between city i and city i. Since both the cities are same and it was assumed that there is a charging station at every city, the number of charging station between them will be one. In other words, the diagonal elements depict a single city.

Corresponding to the charging station matrix, a three dimensional station distance matrix was used. The algorithm for calculation of the element *station distance (i, j, k)* was as follows:-

- The element *station_distance (i, j, k)* represents the distance of i^{th} charging station between cities j and k from the city j.

- The element *station_distance (i, j, k)* was equal to zero if the i[th] charging station don't exist between the cities j and k.

D. Route Navigation

The battery of the EV was assumed to discharge at a constant rate which was taken as an input from the user. The discharge coefficient of the battery was supposed to be k. Also, the source city was taken from the user as input. The whole route of the EV was decided as follows:-

- The EV will depart from the source with the battery fully charged.
- It will be checked whether the EV had enough charge to reach the next city according to the optimized route calculated before. If it is so, then the EV was allowed to move otherwise it will be calculated whether it would reach the nearest charging station or not.
- If it is having enough charge to reach the nearest charging station, then it will be allowed to move. If not, then the route would considered to be impossible.
- After reaching the charging station, the battery of the EV will be fully charged. Then it will be checked whether it is having enough charge to reach the next city. If so, then it will be allowed to move otherwise it will be checked whether it could reach the next nearest charging station or not.
- Same steps will be followed till the EV reaches the last city according the optimized route.

The flow chart of the algorithm used is depicted in the figure:-

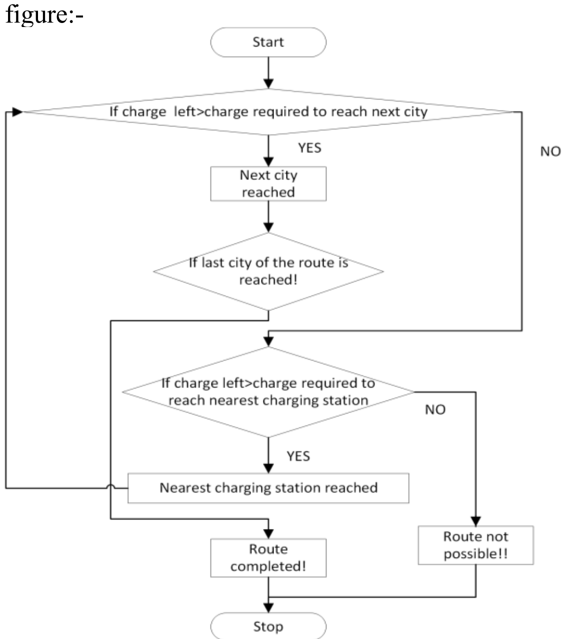

Figure 3. Flow Chart to determine the route of EV

When an impossible route will be encountered, then another optimized route would be calculated using the next nearest neighbor algorithm and the whole process will be repeated. With considerable number of iterations, the desired route will be obtained with accuracy and the accuracy is directly proportional to the number of iterations. The parameters used for the previous result were:-

- Battery discharge coefficient.
- Source city.
- Distance matrix.
- Station_distance matrix.
- Shortest route calculated before.

IV. Output Of The Program

Following was the output of the program when executed through the Turbo C++ compiler.

1->31->27->17->2->13->8->14->22->32->20->18->3->4->24->29->5->26->12->9->19->23->10->16->11->30->6->21->28->25->15->7->!!!
Total Optimal Value: 421

TABLE I. Route Navigation of EV.

City Reached	Percentage of Charge left in the battery	Status
1	100	SOURCE
31	84	
27	77	
17	69	
2	59	
13	51	
8	32	
14	18	
22	1	Battery charged at city 22
32	91	
20	86	
18	84	
3	43	
4	40	
24	33	
29	14	
5	2	
Charging Station	1	Battery charged at charging station
26	97	
12	93	
9	77	
19	68	
23	51	
10	47	
16	21	
11	4	
Charging station	0	Battery charged at charging station
30	84	
6	63	
21	42	
28	17	
25	9	
15	6	Battery charged at city 22
7	71	Destination Reached

The following table represents the optimal route length when different cities are taken as source.

TABLE II. OPTIMAL ROUTE LENGTH TAKING DIFFERENT CITIES AS SOURCE.

Source City	Optimal Route length
1	421
2	450
3	449
4	450
5	473
6	481
7	453
8	460
9	487
10	485
11	476
12	450
13	477
14	465
15	455
16	477
17	453
18	448
19	459
20	450
21	490
22	455
23	486
24	489
25	457
26	477
27	455
28	458
29	454
30	481
31	454
32	452

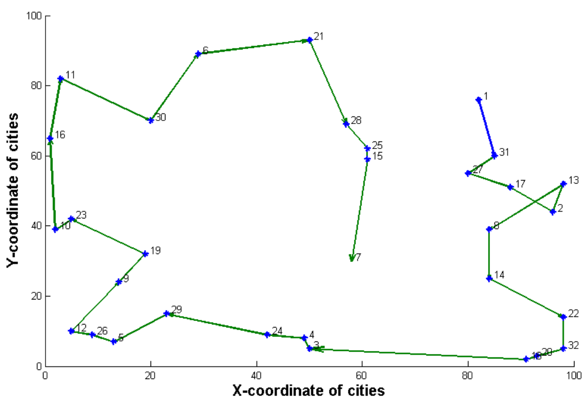

Figure 4. Route Map taking the 1st city as the source.

The subsequent graphs show the route map of the electric vehicle taking City 5, City 10, City 15 and City 32 as source. The numbers in the graph correspond to the cities which are plotted according to the coordinates which were given as input. The arrows show the direction of movement of the electric vehicle. For example, if the arrow is from city 1 to city 12, then it means the electric vehicle travelled from city 1 to city 12.

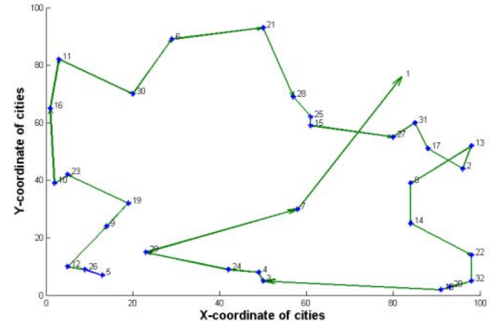

Fig.5 : Route Map taking the 5th city as the source.

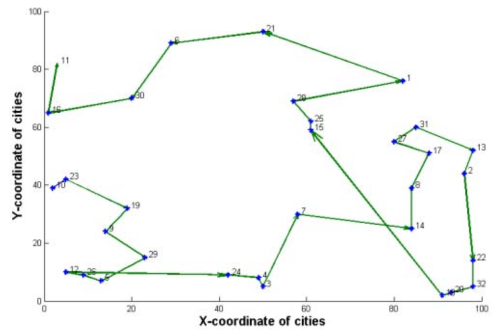

Figure 6. Route Map taking the 10th city as the source.

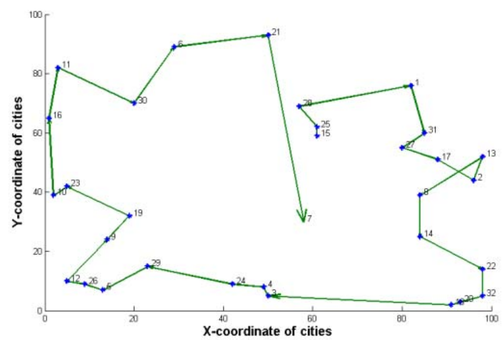

Figure 7. Route Map taking the 15th city as the source.

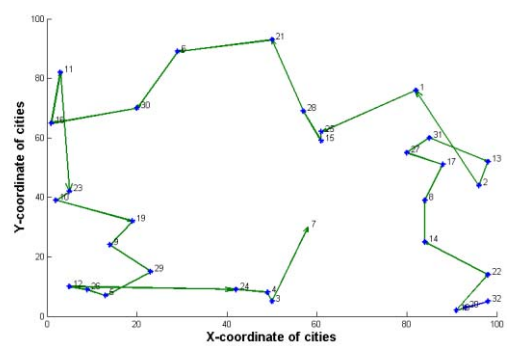

Figure 8. Route Map taking the 32nd city as the source.

V. CONCLUSION

In conclusion, a Nearest Neighbour Search based solution to the energy efficient routing problem for electric vehicles was presented including the charging stations. The algorithm was then applied to a VRP data and simulation results proved the accuracy of the algorithm in generating the optimized solution to the electric vehicle routing problem with charging nodes taken into account. The Nearest Neighbour Search based algorithm proved to be productive, dynamic and economical and thus can be further explored and modified to be applied on real-time problems.

REFERENCES

[1] Wu Zhihong, Wang Qi, Zhu Yuan, Du Meidan "*A new route searching method for EVs Considering electric motor efficiency and charging stations"*.

[2] Rami Abousleiman, Osamah Rawashdeh *"Tabu Search Based Solution to the Electric Vehicle Energy Efficient Routing Problem"*.

[3] Ruhan He, Weibin Xu, Ying Wang, Wei Zhan "*A Route-Nearest Neighbor Algorithm for Large-Scale Vehicle Routing Problem"*.

[4] Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L., Stein, Clifford (2001). *"Section 24.3: Dijkstra's algorithm. Introduction to Algorithms"*.

[5] Frank Harary, Robert Z. Norman and Dorwin Cartwright (1965) "*Structural Models: An Introduction to the Theory of Directed Graphs"*.

[6] R. Abousleiman and O. Rawashdeh. *"Energy Efficient Routing for Electric Vehicles using Particle Swarm Optimization"*.

[7] R. Abousleiman and O. Rawashdeh *"An Application of Ant Colony Optimization to Energy Efficient Routing for Electric Vehicles"*.