Bulletin of Faculty of Science ,Zagazig University (BFSZU) ***2023***

Bulletin of Faculty of Science, Zagazig University (BFSZU)
e-ISSN: 1110-1555
Volume-2023, Issue-2, pp-1-5
https://bfszu.journals.ekb.eg/journal
Research Paper DOI: 10.21608/bfszu.2023.160904.1180

==

A Data Analytics Approach for Solving Vehicle Routing Problem Using Real Time Data

MOHAMED GAMAL [a*], AMAL F. A. EL-GAWAD[a], AMANY ABDELSAMEA[b]

[a] Faculty of Computers and Informatics, Zagazig University, Sharkia 44519, Egypt

[b] Electronics Research Institute, Cairo 12622, Egypt

Corresponding author: mgyounis@fci.zu.edu.eg

ABSTRACT **: This paper presents a new method for solving the traditional vehicle routing problem based on real-time data. The proposed approach starts with collecting and analyzing real time data from established sensors or data entered manually to make sure this data represent the real time status for routes and edges then selecting the most influential factors (route width, car flow, car speed and average number of cars that move in this rout in specific slot of time) that effect directly in classifying the available routes. Based on this selection, a comparison is made between different routes and the best one is selected accordingly. The approach is an iterative process that is repeated to obtain the global optimum. Experimental results show that the proposed approach has the potential to obtain global solutions that are optimal or more accurate because it affected by the real time data, so it reflects the real status of the rout in real time than other evolutionary methods.**

KEYWORDS: Real Time Data, Data Analytics, Vehicle Routing Problem.

Date of Submission: 2022-09-05=================================Date of acceptance: 2023-04-09

I. INTRODUCTION

The Vehicle Routing Problem (VRP) is one of the most studied routing problems which has a wide range of applications. Many problems in practice can be expressed in the form of VRP forms, such as, computer cabling, robot control, traffic routing and crystallography. Most of Travelling Salesman Problems (TSP) and Vehicle Routing Problem describe the situation in which a single vehicle or salesman completes all the routes in the shortest possible time. The size of these problems can be rather large. For example, according to recent reports on TSP applications, the circuit board boring problem was equivalent to a TSP of about 17,000 cities **[1]**, while the VLSI matrix construction problem was comparable to a TSP of 1.2 million cities **[2]**. With low complexity, medium accuracy and low computational time the capacitated vehicle routing problem has been successfully solved using metaheuristic algorithms such as simulated annealing **[3]**. There are algorithms available to give us exact solutions, but they are inadequate **[4]**.

The simplest way to solve these difficulties is to formulate them as a travelling salesman problem. The majority of applications are based on real-world issues and hence appear to be of great relevance **[10]**. However, in all of these studies, solutions offered based on static or approximated data. It is possible that these solutions are not based on rapid changes in the factors affecting the results, in other words, solutions are not based on real time data. In this paper, we present a way to solve this problem by applying the nearest neighbor algorithm on real collected traffic data.

II. MATERIALS AND METHODS

Data Analytics and Nearest Neighbor Algorithm

2.1. Data Analytics: Data analytics help us to understand the blind view of data so we can use it in our case to find the main characteristics that directly affect the weight of routes. Our data analytics approach consists of five main steps as discussed below **[11]**.

2.2. Nearest Neighbor Algorithm for VRP: The Nearest Neighbor Approach (NN) is a rough algorithm for locating a suboptimal VRP solution. This algorithm begins with a single city as a starting point and continues to visit all cities until they have all been visited exactly once. It seeks out the shortest route as shown in Fig.1 **[16]**

Step 1: Begin with any random vertex, which you'll refer to as current vertex.

Step 2: Find an edge called V that provides the shortest distance between the present vertex and an unvisited vertex.

Step 3: Change the current vertex's state to unvisited vertex V and mark it as visited.

Step 4: If all of the vertices are visited at least once, the condition is terminated.

Step 5: Now proceed to step 2

2.3 Modifies Nearest Neighbor Algorithm based on real time data Steps:

Step 1: Begin with any random vertex, which you'll refer to as current vertex.

Step 2: extract the real time data from sensors as data.

Step 2: recalculate the edges weight based on the new real time data.

(The edge weight = avg speed on edge / edge length)

Step 3: Find an edge called E that provides the shortest distance between the present vertex and an unvisited vertex.

Step 4: Change the current vertex's state to unvisited vertex V and mark it as visited.

Step 5: If all the vertices are visited at least once, the condition is terminated.

Step 6: Now proceed to step 2

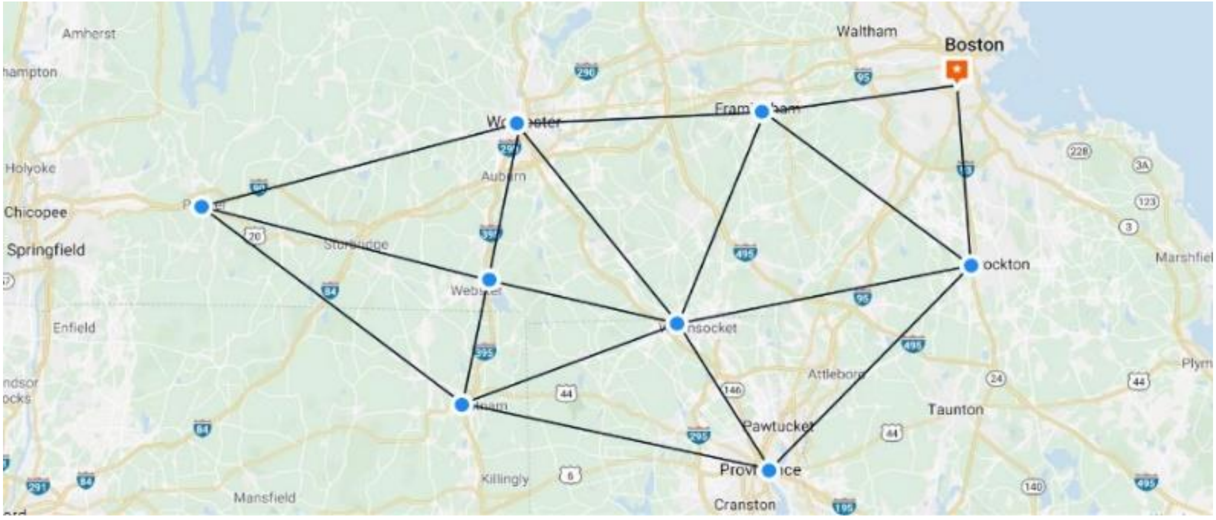

Fig.1 Possible routes between visited locations.

Algorithm 1: Modifies Nearest Neighbor Algorithm based on real time data

```
// the input of this algorithm is Vertex, Edges and Edges weights
// the weights of edges are updated in each step
Data: Vertex, Edges and Edges weights
// the output is the best way between source and destination based on real time data
Result: Return the shortest path between source and destination
Function Nearest Neighbor (Vertex, Edges , Edges weights):
create vertex set Q
// Initialization
// loading the real time data for each round
Edges_data = Load_sensor_data(YYYY,MM,DD)
// calculate the edge weight based on new data
Edge_weight = avg_speed /edge_length
For Each vertex v in Graph:   // initial distance from source to vertex v is set to ∞
    dist(v) ← INFINITY        // Previous node in optimal path from source
    prev(v) ← UNDEFINED
    add v to Q
end
dist(source) ← 0      // main loop
while Q is not empty:
        U ← vertex in Q with min dist(u)
        remove U from Q
        S ← empty sequence
        U← target // Do something only if the vertex is reachable
        if prev(u) is defined or u = source: // Construct the shortest path with a stack
        S
            while u is defined:                // Push the vertex onto the stack
                u at the beginning of S     // Traverse from target to source
                u ← prev(u)
            end
        end
end
```

III. RESULTS

In this section, we will present some results from practical experiments for which data are entered in different formats (estimated data, real fixed data, and real time data which updated after each round), using this we will compare those results and we must know that the main goal of using this data is real experiment, even if it does not have the best Practical results.

This experiment is executed with estimated data, real fixed data, and real time data which updated after each round, the particles in the initial population are selected at random. The calculating results are shown as table 1:

The results of table 1 show that: our algorithm is quite efficient, and it was able to obtain fairly ideal results

Type of data	Number of executions	best result	worst result	average result
estimated data	**25**	**35**	**53**	**44**
real fixed data	**25**	**30**	**46**	**38**
real time data	**25**	**33**	**60**	**46.5**

Table.1 The comparisons of the results

Although the average results of this method are not the best, but it is the closest to reality because it is based on real data and up to date.

REFERENCES

[1] **Litke, J.D.**: An Improved Solution to the Traveling Salesman Problem with Thousands of Nodes. Communications of the ACM 27(12), 1227–1236 (1984)

[2] **Korte, B.**: Applications of Combinatorial Optimization. Talk at the 13th International Mathematical Programming Symposium, Tokyo (1988)

[3] **T. Vidal, T.G. Crainic, M. Gendreau, C. Prins**, Heuristics for multi-attribute vehicle routing problems: a survey and synthesis, Eur. J. Oper. Res. 231 (2013) 1–21.

[4] **G. Laporte**, The vehicle routing problem: an overview of exact and approximate algorithms, Eur. J. Oper. Res. 59 (1991) 345–358.

[5] **G. Clarke, J.W. Wright**, Scheduling of vehicles from a central depot to a number of delivery points, Oper. Res. 12 (4) (1964) 568–581.

[6] **B. Gillett, L. Miller,** A heuristic algorithm for the vehicle-dispatch problem, Oper. Res. 22 (2) (1974) 340–349.

[7] **M.L. Fisher, R. Jaikumar,** A generalized assignment heuristic for vehicle routing, Networks 11 (1981) 109–124.

[8] **R. Holmes, R. Parker,** A vehicle scheduling procedure based upon savings and a solution perturbation scheme, Oper. Res. 27 (1976) 83–92.

[9] **Laporte, G. and S. Martello,** THE SELECTIVE TRAVELLING SALESMAN PROBLEM. 2019.

[10] **LENSTRA Mathematisch Centrum,** J. K., &G Rinnooy Kan, A. H. (2019). Some Simple Applications of the Travelling Salesman Problem. 26, 717-73.

[11] **Li, B., Yang, J., & Hu, D.** (2020). Dam monitoring data analysis methods: A literature review.

[12]**Feng, Y., Duives, D., Daamen, W., & Hoogendoorn, S.** (2021). Data collection methods for studying pedestrian behaviour: A systematic review. 187. doi: 10.1016/j.buildenv.2020.107329

[13]**Xu, X., Lei, Y., & Li, Z. (2020).** An Incorrect Data Detection Method for Big Data Cleaning of Machinery Condition Monitoring. 67, 2326-2336.

[14] **Farouk, A., & Zhen, D. (2019)**. Big data analysis techniques for intelligent systems. 37, 3067-3071.

[15] **Qin, X., Luo, Y., Tang, N., & Li, G. (2020).** Making data visualization more efficient and effective: a survey. 29, 93-117

[16] **Cheikhrouhou, O., & Khoufi, I. (2021).** A comprehensive survey on the Multiple Traveling Salesman

Problem: Applications, approaches and taxonomy. 40

[17] **Ariyaluran Habeeb, R. A., Nasaruddin, F., Gani, A., Targio Hashem, I. A., Ahmed, E., & Imran, M. (2019).** Real-time big data processing for anomaly detection: A Survey. 45, 289-307.

[18] **Krishnamurthi, R., Kumar, A., Gopinathan, D., Nayyar, A., & Qureshi, B. (2020).** An overview of iot sensor data processing, fusion, and analysis techniques. 20, 1-23.

[19] **Madyatmadja, E. D., Nindito, H., Bhaskoro, R. A., Verasius, A., Sano, D., & Sianipar, C. P. M. (2021).** Algorithm To Find Tourism Place Shortest Route: A Systematic Literature Review. 28.

[20] **Boyko, N. (2021).** An Overview of Existing Methods for Solving the Travelling Salesman Problem in Order to Find the Optimal Solution for Economic Problems. 1(7), 85-102.

[21] **Aneiros, G., Cao, R., Fraiman, R., Genest, C., & Vieu, P. (2019).** Recent advances in functional data analysis and high-dimensional statistics. Journal of Multivariate Analysis, 170, 3-9.