The multi-point delivery problem: Shortest Path Algorithm for Real Roads Network using Dijkstra

Samaher Adnan*  enas Wahab Abood*  Wafaa Abdulmuhsin*

*Mathematical Science - Basrah university _Iraq

E_mail: enaswahab223@gmail.com

Abstract The Multi-point Delivery problem is a Vehicle Routing problem (VRP) in which the vehicle has single point as start and end point and must visit a set of points as customers to replay their requests. This paper produces an algorithm built for finding a shortest path for multi-points delivery problem can be used by drivers and autonomous vehicle. The system uses a real road map for a part of Altijiari center of Basrah as a delivery problem. The process is firstly built a proper graph to represent the roads then used Dijkstra Shortest Path algorithm to find the shortest path from start point to all points need to be served then taking the closest one and considered as new start point, continue researching to find short path to other points until all points visited, that for forward path, thus each point path is added to get the full path that pass in all points in delivery scope to get an optimize arrangement for visiting required points with least cost. The second step is to calculate the path for backward to start point. The experimental results show that the proposed algorithm performed efficiently in terms of cost for paths and time for calculation.

1. Introduction
Detection of a shortest path on real roads either for autonomous vehicles or for drivers became an essential and important issue for vehicle manufacturers as well as academic researchers; it minimizes the cost, risks of traffic jam, traffic accidents and improving road use. The shortest path problem (SPP) is a common problem intersected with applications in many fields such as Operation Research, Computer networks, game design, WSN networks and Artificial Intelligence. Many of scholars have deeply researched in this field and produced lots of new algorithms for shortest path problem SPP that intersected with the developing of graph theory and computer data structure [1][2].

It plays (i.e. SPP) a major role in most problems that essentially have any combinatorial optimization problems which can be expressed as a shortest path problem [3][4]. For multi-points delivery problem there are a set of points (customers) with known address and request to be visited and served in an optimized way that reduces time and cost for delivery [5][6].

Dijkstra algorithm is one of the most known methods that used to solve the shortest path problem and Path selection [7]. There are many algorithm used for finding optimal path like Dijkstra algorithm, Floyd algorithm, etc.[8].The Static Dijkstra is an iterative algorithm that find shortest path from a specific node(vertex) called a start node to all the other nodes(vertices) of the graph (Dijkstra, 1959) [3].

Mathematically its work can be represented as:
if d[h]+w(h,v) < d[v] then
    d[v]=d[h]+w(h,v)

Where d[h] is the estimated distance of node (vertex) h from the start node, whereas w(h,v) denotes the weight of edge (h,v). The Priority Queue used as data structure in the basic algorithm [3]. Dijkstra
The algorithm is known as the finest algorithm when all the weight number is positive. And its basic idea is expanding from the start and progressively to outward. During exploring the path, it can find out the shortest path from start to finish at most by n-1 (n is the number of nods in the graph) [8].

In our paper Dijkstra algorithm was used for finding a shortest path from start vertex that represent a start point for multi points delivery at a time, the paper find a shortest path for proceeding delivery issue from start to specific points all represents a real addresses from Altijari center of Basrah city and shortest path for backing to start point. the data of real points was represented as graph, In it, each two points connected by an edge represents a real road with weight equal to length of that street by supposing the fixity of traffic speed and jam conditions. The system works to find a short path from start point to all other points in delivery scope, taking the closest one as new start point. Repeat first step to all other points need to be visited each time, then finding the shortest path from last point back to start node.

2. Related Works

Many scholars invested Dijkstra algorithm in there study to find path for graphs which represent a roads for drivers or autonomous vehicles, A dynamic road network model based on Dijkstra’s algorithm was produced by Yi-zhou Chen (2014) the model contain a data of the urban route and results are analyzed to select the optimal path that apropos a different period, like in morning peak, common or evening peak [8].

Farther, some researchers tend to study and improve the work of Dijkstra algorithm to invest it in their study to get more optimal result like Mo Chen and et.al. (2007) study the influence of using different priority queues in the implementation of Dijkstra’s single-source shortest path (SSSP) algorithm. Their study compared between using a standard priority queue without the decrease-key operation which is better than using it with the decrease-key operation in most cases [10].

M. Xu and et.al.(2007) made a simple changes in original Dijkstra shortest path algorithm to get improved algorithm for finding shortest path for sparse network[7]. Wang Shu-Xi (2012) who introduced Dijkstra’s “label algorithm” and state that the algorithm must be improved then proposed an improved algorithm which is more effective than “label algorithm”, and can solve insufficiencies of “label algorithm”[4]. Hao Liu and Jae-Cheon Lee(2018) produced an algorithm for finding a shortest path for autonomous vehicle delivery problem, there algorithm was consists of two steps: first step was to sort the delivery point depending on a criterion index taking the distance from itself to the inlet and the outlet as a consideration . In second step, the local length was calculated for two direction paths from the last to the next points. The proper direction path can be selected according to their lengths [9]. Another paper was presented by Sunita and Deepak Garg (2018) for dynamizing the Dijkstra algorithm that helps to solve the dynamic single source shortest path problem efficiently. The retroactive priority queue data structure was used for achieving dynamization which identify a set of affected vertices (nodes) step by step that help to accommodate the changes in least number of computations [3].

3. Proposed Model

3.1. Multi-Points Model

The model presented in this study is a mission for delivering products from start point (called 1) to neighboring points in the area belongs to Altijari center of Basrah city in Iraq then back to start point which represents the main distributor. The shortest path problem(SPP) in this study is a static problem supposing that the edge weight static and represents the length of the road from point to point as showed in Figure(1).
There are 28 points in the area, the point (1) is the start (main disturber) and end point in the same time, and the other points represent a delivery points. An undirected graph is modeled to represent the problem, Figure (2).

3.2. Graph Notation
Let $G = (V, E, W)$ be a simple undirected graph, where $V$ is the set of points (vertices), thus $V_i, i = 1, 2,..,28$ and $E$ are the edges. Each vertex $V_i$ is represented by a number like 1,2,… etc. So the edge denoted here as $E(1, 2), E(15, 21),…$ [10]

An edge $E(u, v)$ is the edge between the point $u$ to point $v$ with weight $W(u, v)$, thus the $W$ is a distance value between connected vertices $u$ and $v$ (connected cities) and it’s a non-negative real numbers (i.e. $W$ gives the weights of the corresponding edges).
3.3. Dijkstra Algorithm

The Dijkstra algorithm works depends on partitioning all vertices (points) into two sets: temporary set and permanent set vertices.

At each iteration, a temporary (unvisited) vertex is selected with the minimum distance (set as zero) as the next point to be scanned [12] (Dijkstra 1959; Ahuja et al. 1993, p.109). Once a point is scanned, and gives the shortest path it added to permanent (visited) set that want be checked again and set it as the new start point then back to scan its neighbors. The Dijkstra algorithm stopped after visiting the destination point or when the temporary set is empty.

For calculating the path, example, if the current visited point A with a distance of 6, and the edge connecting it with a neighbor B has length 4, then the distance to B through A will be 6 + 4 = 10. If B was previously marked with a distance greater than 10 then change it to 10. Otherwise, keep the current value [11].

3.4. Multi-Points Shortest Path Problem

Our Delivery problem presented in Figure (1) is represented as an undirected graph shown in Figure (2), the aim of our research is to find the shortest path from start point (vertex) to a serial of points need to be served, then back to start point.

The algorithm for finding shortest path is containing two stages:
- First: finding shortest path forward from start point to points need to be visited.
- Second: calculating the shortest path for returning to start point.

Finding forward path is illustrated in below:
1. Read a G as graph of all points within the study, \( G = \{ V_1, V_2, \ldots, V_m \} \).
2. Read the set of points to be visited, \( S = \{ V_1, V_2, \ldots, V_n \} \).
3. Initialize a Short Path Vector (SPV) containing Start point \( V_{start} \).
   \( SPV = \{ V_{start} \} \).
4. Set \( V_{start} \) as a temporary start point \( V_{temp} = V_{start} \).
5. Find the shortest path using Dijkstra algorithm, section (2.3) from temporary start point \( V_{temp} \) to all points in \( S \): \( D_i(V_{temp}, V_i) \) (i=1,2,…,n).
6. Take the shortest calculated paths \( D_i \) (i.e. closest point) and add its path \( D_i \) to the list: \( SPV \rightarrow SPV' = \{ SPV', D_i \} \).
7. Delete the point \( V_i \) from list \( S \).
8. Set \( V_i \) as temporary start point \( V_{temp} \).
9. \( n = n - 1 \).
10. Repeat step (5-9) until \( n = 0 \).
11. The SPV is the forward path for all \( V \) in \( S \).

Finding back:
The autonomous vehicle or drivers supposed always to go forward seeking for shortest path [9]. Away back to start point is a critical too and need to be optimized. Again, Dijkstra shortest path algorithm is used to find path from the last point in list SPV to \( V_{start} \).

4. Algorithm Implementation

In order to implement and validate the proposed algorithm, suppose having a delivery mission to \( S = \{ 4, 16, 2, 23, 20 \} \), \( V_{start} = 1 \). Taking \( V_{temp} = 1 \), \( SPV = \{ \} \).

The calculated paths \( D \) from \( V_{temp} \) to \( S \) with costs, Table 1:
Table 1. calculated shortest paths from start point to each point in S (n points)

| Di | Vi → Si | Shortest path D | Cost       |
|----|---------|-----------------|------------|
| 1  | 1 → 4   | {1,2,3,4}       | 51344.252  |
| 2  | 1 → 16  | {1,2,3,4,5,6,7,15,16} | 276821.192 |
| 3  | 1 → 2   | {1,2}           | 1049.694717|
| 4  | 1 → 23  | {1,2,3,4,5,6,7,15,16,23} | 356037.0734|
| 5  | 1 → 20  | {1,2,3,4,5,6,14,20} | 218343.00071|

The least cost is D3={1,2}, then the updated parameters will be:
SPV={1,2}, S={4, 16, 23, 20}, n=4, Vtemp=2. repeat the steps to get Table 2:

Table 2. calculated paths for n-1 points

| Di | Vi → Si | Shortest path D | Cost       |
|----|---------|-----------------|------------|
| 1  | 2 → 4   | {2,3,4}         | 50294.55827|
| 2  | 2 → 16  | {2,3,4,5,6,7,15,16} | 275771.497 |
| 3  | 2 → 23  | {2,3,4,5,6,7,15,16,23} | 354987.3786|
| 4  | 2 → 20  | {2,3,4,5,6,14,20} | 217293.3059|

The least cost is D1={2,3,4}, then the updated parameters will be:
SPV={1,2,3,4}, S={16, 23, 20}, n=3, Vtemp=4. repeat the steps to get Table 3:

Table 3. the calculated shortest paths for n-2 points in S

| Di | Vi → Si | Shortest path D | Cost       |
|----|---------|-----------------|------------|
| 1  | 4 → 16  | {4,5,6,7,15,16} | 225476.9597|
| 2  | 4 → 23  | {4,5,6,7,15,16,23} | 304692.8422598|
| 3  | 4 → 20  | {4,5,6,14,20}   | 166998.74775|

The least cost is D3={4,5,6,14,20}, then the updated parameters will be:
SPV={1,2,3,4,5,6,14,20}, S={16, 23}, n=2, Vtemp=20. repeat the steps of the algorithm until all S is scanned. We get a final result of Shortest path SPV={1,2,3,4,5,6,14,20,21,23,16}, and the points will be served in order {2,4,20,23,16}.

The back way to start point a shortest path from last visited point to the start point is directly calculated, for upper mission the back way is {16,15,7,6,5,4,3,2,1}. The result is shown in Figure (3)(4)(5).

Figure (3). The shortest path for delivery problem for points {4, 16, 2, 23, 20}
Figure (4). Back shortest path from last served point (16) in S to start point (1)

Figure (5). Drawing the forward shortest path for delivery problem on the real roads map

The total cost is 4.7197e+05 for forward way and 2.76821e+05 for back way while the path for serving the points as they come in request queue is:

SPV={ 1,2,3,4,5,6,7,15,16,15,7,6,5,4,3,2,3,4,5,6,7,15,16,23,21,20}
with cost = 1.08198831e+06.

Another mission is taken for validate the algorithm, a series of points need to be served S={18, 6, 25, 8, 13, 10}, the result of proposed algorithm is:

SPV={ 1,2,3,10,12,13,5,6,7,8,7,6,14,20,25,20,19,18} with cost = 7.87177e+05
and arrangement as {10,13,6,8,25,18} for visiting and a way back=[18,19,12,10,3,2,1], Figure(6).
Figure (6). Shows the forward and backward path for delivery mission $S=\{18, 6, 25, 8, 13, 10\}$

The parameters for calculating the path to serve as they come was:

$SPV=\{1, 2, 3, 10, 12, 19, 18, 19, 20, 14, 6, 14, 20, 14, 6, 7, 8, 7, 6, 5, 13, 12, 10\}$

Total cost $=1.16830065e+06$, the way back is $\{10, 3, 2, 1\}$, Figure(7).

Figure (7). Shows the path for delivery problem to serve the points as they come

Notice: between any two point in $S$ using Dijkstra for finding path
5. Conclusion and Future Work
The algorithm presented in this paper is to find the shortest path for multi-points delivery problem for drivers and autonomous vehicle. The proposed algorithm proves its efficiency to determine the shortest path for multi-points via comparison with the cost need to serve them without this algorithm, the time for arranging points and calculating the shortest path is to short less than 0.5 second. This work assumes the real roads unconstrained and undirected. So the future work can optimize the shortest path regarding to some restrictions and conditions, such as the road capacity in different day time, road direction, traffic jam (time for road passing)... etc..

References
[1] A. Orda and R. Rom 1990 Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length Journal of the Association for Computing Machinery vol. 37 issue 3 pp. 607-625.
[2] M. R. Henzinger P. Klein and S. Rao 1997 Faster shortest-path algorithms for planar graphs Journal of Computer and System Sciences vol. 55 issue 1 pp. 3-23.
[3] Sunita Deepak Garg 2018 Dynamizing Dijkstra: A solution to dynamic shortest path problem through retroactive priority queue Journal of King Saud University – Computer and Information Sciences xxx (2018) xxx–xxx Article in press.
[4] Wang Shu-Xi 2012 The Improved Dijkstra's Shortest Path Algorithm and Its Application 2012 International Workshop on Information and Electronics Engineering (IWIEE) Procedia Engineering 29 (2012) 1186 – 1190.
[5] L. Grandinetti F. Guerriero F. Oezzella O. Pisacane 2014 The Multi-Objective Multi-Vehicle Pickup And Delivery Problem With Time Windows EWGT2013- 16th Meeting of EURO working Group on Transportation Procedia – Social and Behavioral Sciences 111 (2014) 203-212.
[6] M. I. Hosny C. L. Mumford 2012 Constructing initial solutions for the multiple vehicle pickup and delivery problem with time windows Journal of King Saud University – Computer and Information Sciences 24 59-69.
[7] M. Xu Y. Liu Q. Huang Y. Zhang G. Luan 2007 An improved Dijkstra shortest path algorithm for sparse network Applied Mathematics and Computation 185 pp.247-254.
[8] Yi-zhou Chen Shi-fei Shen Tao Chen Rui Yang 2014 Path Optimization Study for Vehicles Evacuation Based on Dijkstra algorithm Procedia Engineering 71 159 – 165.
[9] Hao Liu and Jae-Cheon Lee 2018 A shortest path algorithm for autonomous vehicle delivery problem International Research Journal of Advanced Engineering and Science Volume 3 Issue 1 pp:86-89.
[10] Chen M. Chowdhury R. A. Ramachandran V. Roche D. L. Tong L. 2007 Priority Queues and Dijkstra’s Algorithm UTCS Technical Report TR-07-54 — 12 October. Austin Texas.
[11] S. Gass M. Fu 2013 Dijkstra's Algorithm Encyclopedia of Operations Research and Management Science. Springer. 1. doi:10.1007/978-1-4419-1153-7.
[12] F. Benjamin Zhan 1997 Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures Journal of Geographic Information and Decision Analysis vol.1 no.1 pp. 70-82.