International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

A Shortest Path Algorithm for Autonomous Vehicle Delivery Problem

Hao Liu[1], Jae-Cheon Lee[2]
[1]Department of Mechanical and Automotive Engineering, Keimyung University, Daegu, Korea
[2]Department of Mechanical and Automotive Engineering, Keimyung University, Daegu, Korea

Abstract*—*This paper descripts an algorithm to find a shortest path for autonomous vehicle delivery problem. The process is divided into two steps. The first step is to sort the delivery point according to a criterion index considering the distance from itself to the inlet and the outlet. Because it is assumed that the autonomous vehicle always drives forwards the direction of passing each delivery point should be determined. Thus the second step is to calculate the local length for two direction paths including the last and the next points. After comparing their lengths, the proper direction path can be selected. A case is used to validate the presented algorithm. The result shows that the algorithm is feasible.

Keywords*— *Autonomous vehicle, delivery path planning, shortest path algorithm (SPA).

I. INTRODUCTION

Nowadays autonomous vehicle, including Advanced Driver Assistance System (ADAS), is a hot issue for not only automotive manufacturers and academic research institutions or universities but also algorithm development service companies. It can be foreseen that autonomous vehicle is a very useful and promising transportation to reduce traffic accidents, optimize traffic flow, relieve drive's fatigue, and provide service for old people and so on [1][2].

Implementation of an autonomous vehicle is composed of perception, decision, and control. Path planning is carried out in the process of decision after surroundings of the autonomous vehicle is detected. Functions of path planning are to avoid obstacles such as other vehicles, pedestrians, or static obstacles in local map and to find the shortest path in global map. In graph theory, the Shortest Path Problem (SPP) is the problem of finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized [3-5]. SPP is one of focuses of computer science, operations research, geographic information science. Lots of scholars have done deep research in this field and obtained many new algorithms of SPP with developing of graph theory and computer data structure [6-7].

II. DELIVERY PROBLEM OF AUTONOMOUS VEHICLE

There is one mission in an autonomous competition held in Korea, which is named global path planning for autonomous vehicle delivery. The mission aim is to deliver products to five points in a designated area in shortest time. There are 4 by 5 blocks array in the area, where one block size is 6 meters by 6 meters square and the distance of two neighbouring blocks in both longitudinal and lateral direction is 4 meters as illustrated in Fig. 1. Five delivery points (P_1, P_2, … P_5) which are one side of a certain block are designated beforehand. The autonomous vehicle is requested to enter into the specified port and pass the five delivery points and exit from another specified port. The entry and exit may be changed for different participating team.

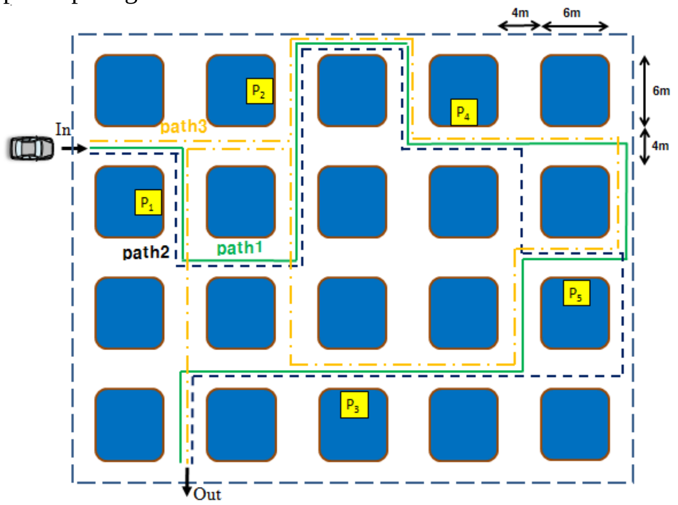

Fig. 1. Delivery problem for autonomous vehicle.

The mission is essentially the shortest path problem. Thus, the problem can be described how to determine the shortest path from entry to exit, in which all five delivery points should be passed one time. If we use 4 by 5 grids to enclose the entire delivery block area to ensure each grid include one block, each grid is a square with side of 10 meters. The autonomous vehicle drives along edges of grids from inlet to outlet so the total distance of the driving path is the sum of all edge length. One shortest path, *path1*, in green solid line in Fig. 1is presented, where the grids are not drawn. The distance of the *path1* is 16 times of the side length L (L=10 m), i.e. 60 m. Obviously, the shortest path is not unique. If some segments of the *path1* is adjusted another shortest path with length of 16L, *path2*, can be obtained as shown in black dashed line in Fig. 1. On the other hand, *path3* in orange dot-dashed line can fulfil the delivery mission but it covers 18L distance, which is not the shortest path for the autonomous vehicle delivery. The reason is that the path does not firstly pass the point P1which is closest to the in port and finally makes a detour across P1 after passing other points. The approach to finding the shortest path for autonomous vehicle delivery is discussed in next section.

Hao Liu and Jae-Cheon Lee, "A shortest path algorithm for autonomous vehicle delivery problem," *International Research Journal of Advanced Engineering and Science*, Volume 3, Issue 1, pp. 86-89, 2018.

III. SPA FOR DELIVERY PROBLEM

A. *Problem Analysis*

If the three delivery paths are compared together it can be found that, in fact, *path1* is the same as *path2* because

- they have same path length and
- the number of left turn and that of right turn are same although their part segment of path passing the point P5 are different.

However, *path3* is not the shortest path because P1 is the point passed at last, yielding the detour path. Thus, first of all the most important thing is to sort the points from. The following rule can be obtained.

Rule 1: *Five delivery points should be successively sorted in the order from the entry to the exit by distance.*

The delivery problem for the autonomous vehicle is a little bit different from the normal shortest path such as travel salesman problem in that the vehicle forward driving should be considered in the former. The driving direction has been determined forwards when the autonomous vehicle passing one point, the driving path can't be retraced to the next point while the path can be determined without limitation from one point to next point in latter. Thus we can get Rule 2 as below.

Rule 2: *The delivery path cannot be designed retrace considering that the autonomous vehicle should drive forwards.*

B. *Algorithm for Sorting Delivery Points*

The delivery problem presented in Fig. 1 can be abstracted by the grid shown in Fig. 2. The grid means the path that the autonomous vehicle can drive. Each grid is square with side length of 2*L*. Our goal is to find a shortest path along edges of grid from inlet (point P_{in}) to outlet (point P_{out}), which should pass delivery five points. In order to specify the position, we give 2D subscript for each node on the grid. Thus we have

- Inlet: $P_{in}(3, 1)$
- Outlet: $P_{out}(9, 3)$
- Delivery points: $P_1(4, 3)$, $P_2(2, 5)$, $P_3(7, 6)$, $P_4(3, 8)$, $P_5(5, 10)$

For two node $P_i(x, y)$ and $P_j(s, t)$, their D_4 distance (also called city-block distance) between them is defined as below [8]

$$D_4(P_i, P_j) = |x-s| + |y-t| \tag{1}$$

According the two rules the shortest path algorithm for the autonomous vehicle delivery includes two parts: one is to sort the five points and other is to find path between two successive points. As a result, the shortest path algorithm to sort delivery points is presented as below.

1. Let inlet point as temperate point, $P_t=P_{in}$; The unsorted set comprises P_i (i=1, 2,…, 5);
2. Calculate D_4 distances between outlet and each delivery point, $D_4(P_{out}, P_i)$ (i=1, 2,…, 5);
3. Calculate D_4 distances between the temperate pint and each delivery point, i.e. $D_4(P_t, P_i)$ and $D_4(P_{out}, P_i)$;
4. Calculate the following criterion index;
$$k = (D_4(P_t, P_i))^2 / D_4(P_{out}, P_i) \tag{2}$$
5. The delivery point (P_j) whose k value is minimal is the point next to the inlet. Let this point be the new temperate, namely $P_t=P_j$;. And Take out of P_j from the unsorted set.
6. Repeat from step 3 for all delivery points in the unsorted set until the set is empty.
7. The order of P_j is one of delivery paths

The entire calculation process is expressed in Table I. After calculating the first column shows the order of the delivery point, $P_{in} \rightarrow P_1 \rightarrow P_2 \rightarrow P_4 \rightarrow P_5 \rightarrow P_3 \rightarrow P_{out}$.

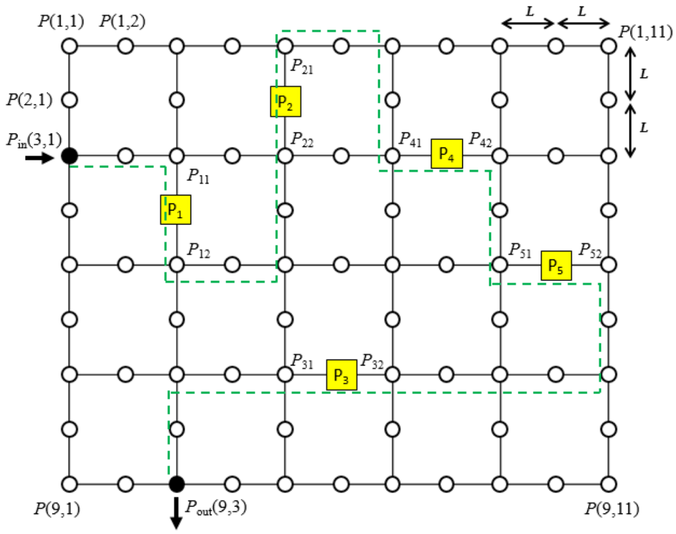

Fig. 2. Grid representation for delivery problem.

TABLE I. Calculation of delivery points sorting.

P_t		$P_1(4,3)$	$P_2(2,5)$	$P_3(7,6)$	$P_4(3,8)$	$P_5(5,10)$
	$P_{out}(9,3)$					
	$D_4(P_{out},P_i)$	5	9	5	11	11
P_{in} (3,1)	$D_4(P_t, P_i)$	3	5	9	7	11
	k	1.8	2.78	16.2	4.45	11
P_1	$D_4(P_t, P_i)$	------	4	6	6	8
	k	------	1.78	7.2	3.27	5.82
P_2	$D_4(P_t, P_i)$	------	------	6	4	8
	k	------	------	7.2	1.45	5.82
P_4	$D_4(P_t, P_i)$	------	------	6	------	4
	k	------	------	7.2	------	1.45
P_5	$D_4(P_t, P_i)$	------	------	6	------	------
	k	------	------	7.2	------	------
P_3	$D_4(P_t, P_i)$	------	------	------	------	------
	k	------	------	------	------	------

C. *Algorithm for Path Generation*

After sorting the delivery points according to the above algorithm, the next work is, considering that the autonomous vehicle always drives forward, how to find the proper path (not shortest path) between two successive delivery points so as to obtain the whole shortest path from inlet to outlet. Table II shows this process.

Hao Liu and Jae-Cheon Lee, "A shortest path algorithm for autonomous vehicle delivery problem," *International Research Journal of Advanced Engineering and Science*, Volume 3, Issue 1, pp. 86-89, 2018.

International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

The path starts from inlet P_{in} and there exist two paths which pass P_1 in inverse directions to arrive at P_2: $P_{in} \rightarrow P_{11} \rightarrow P_1 \rightarrow P_{12} \rightarrow P_2$ and $P_{in} \rightarrow P_{12} \rightarrow P_1 \rightarrow P_{11} \rightarrow P_2$. We can calculate their total length by summing each length between two points. The calculation results show that both are 9. The former path with the frame in the Table II is selected if other restriction condition is not applied. And then the start point is P_{12} and there also exit two paths passing point P_2. By calculation we find that the path $P_{12} \rightarrow P_{22} \rightarrow P_2 \rightarrow P_{21} \rightarrow P_4$ has shorter length (11). Similarly, we can finish Table II until arriving outlet P_{out}. Finally, the shortest delivery path is determined: $P_{in} \rightarrow P_{11} \rightarrow P_1 \rightarrow P_{12} \rightarrow P_{22} \rightarrow P_2 \rightarrow P_{21} \rightarrow P_{41} \rightarrow P_4 \rightarrow P_{42} \rightarrow P_{51} \rightarrow P_5 \rightarrow P_{52} \rightarrow P_{32} \rightarrow P_3 \rightarrow P_{31} \rightarrow P_{out}$, which is the path2 shown in Figure 1.

TABLE II. Calculation of path length (length is time of *L*)

Last point	Middle point1	Middle point	Middle point2	Next point	Total length
length	length	length	length		
P_{in}	P_{11}	P_1	P_{12}	P_2	9
2	1	1	5		
P_{in}	P_{12}	P_1	P_{11}	P_2	9
4	1	1	3		
P_{12}	P_{21}	P_2	P_{22}	P_4	15
10	1	1	3		
P_{12}	P_{22}	P_2	P_{21}	P_4	11
4	1	1	5		
P_{21}	P_{41}	P_4	P_{42}	P_5	9
4	1	1	3		
P_{21}	P_{42}	P_4	P_{41}	P_5	13
6	1	1	5		
P_{42}	P_{51}	P_5	P_{52}	P_3	11
2	1	1	7		
P_{42}	P_{52}	P_5	P_{51}	P_3	11
4	1	1	5		
P_{52}	P_{31}	P_3	P_{32}	P_{out}	20
12	1	1	6		
P_{52}	P_{32}	P_3	P_{31}	P_{out}	12
6	1	1	4		

IV. ALGORITHM VALIDATION

In order to validate the above algorithm, another delivery mission is designed as illustrated in Figure 3. The inlet is same but the outlet is changed to $P_{out}(9,9)$. By calculating the criterion index (2), the delivery points are sorted as $P_{in} \rightarrow P_2 \rightarrow P_1 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5 \rightarrow P_{out}$. Next Table IV is constructed to determine passing direction of each delivery point. It can be seen in the table that the path is $P_{in} \rightarrow P_{21} \rightarrow P_2 \rightarrow P_{22} \rightarrow P_{11} \rightarrow P_1 \rightarrow P_{12} \rightarrow P_{31} \rightarrow P_3 \rightarrow P_{32} \rightarrow P_{42} \rightarrow P_4 \rightarrow P_{41} \rightarrow P_{51} \rightarrow P_5 \rightarrow P_{52} \rightarrow P_{out}$. The length of this path is 34*L*, which is the shortest path satisfying to requirements.

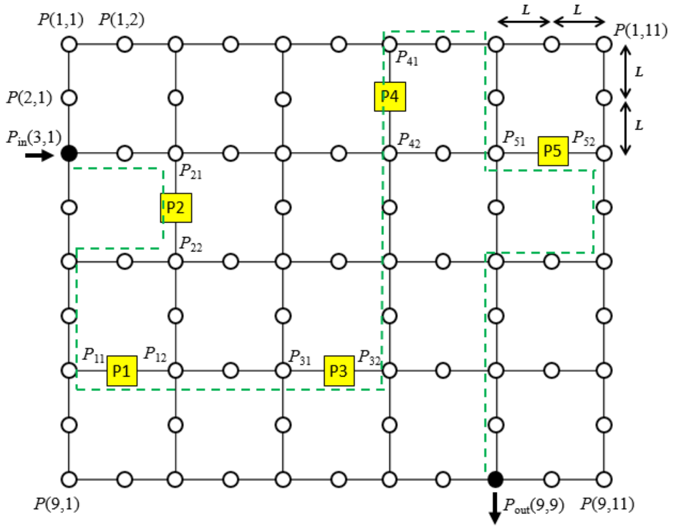

Fig. 3. Grid representation of validation case.

TABLE III. Calculation of delivery points sorting (validation case).

Distance		Points					
P_t	$P_{out}(9,9)$	$P_1(7,2)$	$P_2(4,3)$	$P_3(7,6)$	$P_4(2,7)$	$P_5(3,10)$	
	$D_4(P_{out},P_i)$	9	11	5	9	7	
P_{in} (3,1)	$D_4(P_t, P_i)$	5	3	9	7	9	
	k	2.78	0.82	16.2	5.44	11.6	
P_2	$D_4(P_t, P_i)$	4	------	6	6	8	
	k	1.78	------	7.2	4	9.14	
P_1	$D_4(P_t, P_i)$	------	------	4	10	12	
	k	------	------	3.2	11.1	20.6	
P_3	$D_4(P_t, P_i)$	------	------	------	6	8	
	k	------	------	------	4	9.14	
P_4	$D_4(P_t, P_i)$	------	------	------	------	8	
	k	------	------	------	------	9.14	
P_5	$D_4(P_t, P_i)$	------	------	------	------	------	
	k	------	------	------	------	------	

TABLE IV. Calculation of path length (validation case) (length is time of *L*)

Last point	Middle point1	Middle point	Middle point2	Next point	Total length
length	length	length	length		
P_{in}	P_{21}	P_2	P_{22}	P_1	7
2	1	1	3		
P_{in}	P_{22}	P_2	P_{121}	P_1	13
4	1	1	7		
P_{22}	P_{11}	P_1	P_{12}	P_3	9
4	1	1	3		
P_{22}	P_{12}	P_1	P_{11}	P_3	13
2	1	1	9		
P_{12}	P_{31}	P_3	P_{32}	P_4	9
2	1	1	5		

Hao Liu and Jae-Cheon Lee, "A shortest path algorithm for autonomous vehicle delivery problem," *International Research Journal of Advanced Engineering and Science*, Volume 3, Issue 1, pp. 86-89, 2018.

continued

Last point	Middle point1	Middle point	Middle point2	Next point	Total length
length	length	length	length		
P_{12}	P_{32}	P_{3}	P_{31}	P_{4}	17
8	1	1	7		
P_{32}	P_{41}	P_{4}	P_{42}	P_{5}	15
10	1	1	3		
$\boxed{P_{32}}$	$\boxed{P_{42}}$	$\boxed{P_{4}}$	$\boxed{P_{41}}$	P_{5}	$\boxed{12}$
5	1	1	5		
$\boxed{P_{41}}$	$\boxed{P_{51}}$	$\boxed{P_{5}}$	$\boxed{P_{52}}$	P_{out}	$\boxed{14}$
4	1	1	8		
P_{41}	P_{52}	P_{5}	P_{51}	P_{out}	14
6	1	1	6		

V. CONCLUSION

The paper presents an algorithm to find the shortest path for autonomous vehicle delivery problem. The presented algorithm is feasible via the validation case. The next work is to optimize the shortest path considering some restriction condition such as less number of left/right turn for the autonomous vehicle.

ACKNOWLEDGMENT

Authors would like to appreciate finical support by the National Research Foundation of Korea through Lifetime First Research Program (Project No. NRF-2017R1C1B5018003).

REFERENCES

[1] Senator Hernandez of California Senate. “Vehicles: Autonomous vehicles: safety and performance requirements,” SB-1298.
[2] A. Mukhtar, L. Xia, and T. B. Tang, “Vehicle detection techniques for collision avoidance systems: A review,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, issue 5, pp. 2318-2338, 2015.
[3] N. Deo and C. Y. Pang, “Shortest path algorithms: Taxonomy and annotation,” *Networks*, vol. 14, issue 2, pp. 275-323, 1984.
[4] B. V. Cherkassky, A. V. Goldberg, and T. Radzik, “Shortest paths algorithms: Theory and experimental evaluation,” *Mathematical Programming*, vol. 73, issue 2, pp. 129-174, 1996.
[5] F. B. Zhan and C E. Noon, “Shortest path algorithms: An evaluation using real road networks,” *Transportation Science*, vol. 32, issue 1, pp. 65-73, 1998.
[6] A. Orda and R. Rom, “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, 1990.
[7] M. R. Henzinger, P. Klein, and S. Rao, “Faster shortest-path algorithms for planar graphs,” *Journal of Computer and System Sciences*, vol. 55, issue 1, pp. 3-23, 1997.
[8] R. C. Gonzalez and R. E. Wood, *Digital Image Processing*, 2nd ed., Prentice-Hall, pp. 68-69, 2002.