Finding Realistic Shortest Path in Road Networks with Lane Changing and Turn Restriction

Oanh Tran Thi Kim, VanDung Nguyen, Seung Il Moon and Choong Seon Hong

Department of Computer Science and Engineering, Kyung Hee University, 446-701, Korea
{ttkoanh, ngvandung85, moons85, cshong}@khu.ac.kr

***Abstract*—Most existing work on path computation has been focused on the shortest-path problem, which is to find the optimal path between an origin and destination pair. To get an optimal route, they usually consider travel time moving forward on the road and distance from a source to a target as crucial factors to value and select a path. However, it is not sufficient for real life traffic. Firstly, when we drive on the road, the road driving time includes the duration of moving forward on the road section, going through an intersection and performing lane changes. Secondly, it is not possible, in practice, to go on any road section. Some road sections have restricted rules. Turn restrictions are commonly restricted in a real network to reduce disruption to traffic. Therefore, in our work, we direct to find not only the shortest path but also realistic or feasible path. Both lane changes and turn restrictions are considered in our work. Simulation results show that two these additional constraints are necessarily considered to achieve a realistic shortest path. In addition, our proposal is not only giving a practical path but also a satisfied path for each individual car.**

***Index Terms*—shortest path, lane changing, turn restriction**

I. INTRODUCTION

In future transportation, all of things on the roads are managed and analyzed at Traffic Management System (TMS). With combining VANET and TMS, each Roadside Unit (RSU) acts as a link between cars and TMS. Then, when a car starts a journey, it will send finding path request to RSU. Each RSU will communicate with TMS to get shortest path from car's source to car's destination. When TMS receives a finding path request from RSU, the requirement is processed.

In this paper, we put the finding shortest path problem into perspective of ITS underlying South Korean traffic rules. South Korea is one of the world's leaders in intelligent transportation systems. A central mission of the National ITS Service is to create a network of traffic systems that facilitate interactions and interconnection between South Korea's large cities [1].

Most existing works on path computation have been focused on the shortest-path problem with lane changing condition or turn restrictions. For the lane changing, only I. Yang et al [2] has conducted research to incorporate this problem. For the turn-restrictions, two approaches have been proposed, network modification and link-based labeling strategy. Both methods have disadvantages. For network modification, which requires a long time to load networks from memory, is very time consuming [3]. Link-based labeling strategy was proposed by Gutierrez and Medaglia [4]. They proposed a link-based Dijkstra's algorithm that can be directly used in the original network. However, this algorithm may suffer severe computational performance compared with original Dijkstra's algorithm. A hybrid link-node approach [5] can solve the shortest path problem with turn-restriction with high performance without requiring to modify the network structure. To the best of the authors' knowledge, none has conducted research to incorporate the lane changing and turn-restriction in shortest path problem. In addition, we will take additional conditions, such as weather forecast, or road construction/closure information to improves trip duration.

Our approach is based on the work by I. Yang et al [2]. Unlike this work, we take turn restriction constraint into the finding shortest path problem. Therefore, the shortest path problem now relates to lane changing and turn restriction. Next, the lane changing and turn-restriction problem is analyzed in details to demonstrate how the shortest path can be found with two additional constraints.

The rest of the paper is organized as follows. Section II elaborates the problem setting. In Section III, a problem statement and algorithm of finding realistic shortest path is proposed. Simulation results verify our proposal in Section IV and a summary concludes the paper in Section V.

II. PRELIMINARY

In this section, we introduce some definitions that will be helpful to understand the finding shortest path considering lane changing and turn restrictions and be used in formulating the problem.

A. Road networks

Definition A road network is a directed graph G(V, E), where V is a set of nodes representing road intersections, E is a set of links representing road segments.

A link-based adjacency list structure [4] is used to represent and store the road network in memory. For the set of node V, a vector is utilized. It means that each element of the vector is corresponding a node $i \in$ V. Next, for the set of link E, each link is considered as an object. Then, each

---

This research was supported by the MSIP, Korea, under the G-ITRC support program (IITP-2016-R6812-15-0001) supervised by the IITP

*Dr. CS Hong is the corresponding author.

node i points to list of links depicting the link $(id_i, id_j) \in$ E departing from this node.

B. Managing lane information of links in the network

To solve the lane changing and turn restriction problem, we will manage lane information of links in the network. This work is used to decide the lane number where the vehicle enters in next link depending on the the lane number where a vehicle exit from the current link and the type of turning. For this action, the driver at intersection needs to enter right lane by following traffic rules. In this study, we comply with the traffic rules of South of Korea. Traffic rules are mentioned in [2] as associated lane mapping data. This data

To manage the type of turning at each link *id*, three bits are used to represent turn type. Type of turn is presented by first and second bit. there are 00, 01, 10, 11 that indicate straight, left-turn, right-turn, and U-turn, respectively. Last bit is used for turn restriction (0) or not (1). for more detail, it can be viewed in TABLE. Ia.

TABLE I: Lane information of links in the network

(a) Type of turn representation

Value	Type of turning
000	through
011	left turn
010	restricted left turn
101	right turn
100	restricted right turn
111	U-turn
110	restricted U-turn

(b) Associated lanes by turning types and restricted turn

Type of turning of previous link	Associated current lane
000	out lane
011	out lane
010	∞
101	Rightmost lane
100	∞
111	Rightmost lane
110	∞

For a specific link *id*, there are two parameters related to lane information of link is listed as below:

- $\mathrm{ALM}_{id_i id_j}$: the associated lane map between link id_i and id_j. The result of association lane map is the enter lane number. It is based on turn type and the outing lane number of the previous link as shown in TABLE. Ib. In the case of turn restriction, we cannot enter in this link, so we will assign a large value to the entering lane. Beside, for both right-turn and U-turn, we suppose vehicles have to use the rightmost lane in the current link without attention to what lane they used in the previous link.
- $\mathrm{ELI}_{id_i id_j}$: exit lane information. It is array that is used to find out lane number when vehicles is going to move out link *id*. The result is based on the type of turning at the next intersection.

C. path cost

Before, we explain the path cost in details, some notations that are used in this study are as in Table II

TABLE II: Important notations

Notation	Description
c_{id}	path cost
d_{id}	length of link *id*
d^{r}_{id}	The remaining distance to the next intersection of link *id*
t_{id}	moving forward time of link *id*
t^{LC}_{id}	lane changing time of link *id*
q_{id}	traffic density of link *id*
r_{id}	traffic velocity of link *id*
$\mathrm{tt}_{id_i id_j}$	turn type from link id_i to id_j
ℓ^{c}_{id}	current lane number
ℓ^{i}_{id}	enter lane number
ℓ^{o}_{id}	out lane number

Path cost is considered as travel time. The travel time is considered as sum of the time for moving forward and lane changing. Then, path cost is calculated as follow:

$$c_{id} = t_{id} + t^{LC}_{id}, \forall id \in E. \tag{1}$$

Travel time of moving forward, t_{id}, is the time consumed only for running the length of the link. It is calculated by using Equation 2.

$$t_{id} = d_{id} / r_{id}, \forall id \in E. \tag{2}$$

The lane changing time depends on the number of lanes that a car need to change, traffic density, prevailing speed. We adopt the result in [2] to calculate the lane changing time. As we mentioned before the associated lane map **ALM** and exit lane information **ELI** are used to determine the enter lane number and out lane number, respectively. Suppose that the link that a car is going on is link id_i, the next link is id_j, the type of turn to go to link id_j is $tt_{id_i id_j}$. Then, the enter lane number and out lane number are found by using Equation. 3 and 4, respectively.

$$\ell^{in}_{j} = f_1(tt_{ij}, \ell^{out}_{i}, \mathbf{ALM}_{ij}) \tag{3}$$

$$\ell^{out}_{i} = f_2(tt_{ij}, \mathbf{ELI}_{ij}) \tag{4}$$

Next, the time lane changing is determined as follow,

$$t^{LC}_{id_i} = \begin{cases} 0, & \ell^{o}_{id_i} = \ell^{i}_{id_i} \\ \infty, & \ell^{o}_{id_i} = \infty \\ \infty, & |\ell^{o}_{id_i} - \ell^{i}_{id_i}| * r_{id_i} * \gamma <= d^{r}_{id_i}, 0 < \gamma <= 1 \\ a, & otherwise. \end{cases} \tag{5}$$

Here $a = \ell^{o}_{id_i} - \ell^{i}_{id_i}| * (\alpha * q_{id_i} + \beta)$. When a subject vehicle is just going straight so $\mathrm{t}^{LC}_{id_i}$ qual to 0 (In case 1). When a path is restricted turn or the remaining distance is infeasible to perform lane changing, $\mathrm{t}^{LC}_{id_i}$ is assigned a larger number (In case 2) and (In case 3), respectively. And the last case, $\mathrm{t}^{LC}_{id_i}$ is a product of the number of lanes to move and the effective one-lane changing time, with α = 0.0456 and β = 1.1505 that are referred to [2].

Algorithm 1: Pseudo-code for the proposed algorithm for the *finding shortest path considering lane changing and turn restriction*

```
   Input  : G= (V,E) including ALM, ELI; traffic pattern R,
            speed pattern S, source and destination node
            $s$, $d \in$ G.
   Output: shortest $P^*$ from $i_s$ to $i_d$
1  P← ∅, P̄ ← E
2  for all $e \in$ E do
3      $c_e^s \leftarrow +\infty$
4      $p_e \leftarrow$ null
5  end for
6  $c_s = 0$
7  $p_s = 0$
8  $\ell_s^c = \ell_{SV}^c$
9  for all $e$ = (i$_s$, j) ∈ E, j ∈ SDS(i) do
10     $c_e^s \leftarrow c_e$
11 end for
12 while |P| < |E|
13     $\bar{e} \leftarrow \arg\min_{e \in \bar{P}} (c_e^s)$
14     P ← P ∪ {ē}
15     P̄ ← P̄ \ {ē}
16     if $(c_{\bar{e}}^s) = \infty$ then
17         There is no route between $i_s$ and $i_s$
18         STOP
19     else
20         if ē = (i, $i_d$), i ∈ E then
21             Build $P^*$ backtracking from $p_{\bar{e}}$
22             STOP
23         end if
24     end if
25     for all $e$ such that (ē, e) ∈ $\mathscr{A}$(ē$_o$) and $e \in$ P̄ do
26      if $c_e^s > c_{\bar{e}}^s + c_e$ then
27         $c_e^s \leftarrow c_{\bar{e}}^s + c_e$
28         $p_e \leftarrow \bar{e}$
29         Determine $\ell_e^{in}$ by using Equation .3.
30         Determine $\ell_{\bar{e}}^{out}$ by using Equation .4.
31      end if
32     end for
33 end while
```

III. THE SHORTEST PATH PROBLEM CONSIDERING LANE CHANGING AND TURN RESTRICTION

Suppose at the start time, we have the start link s where the starting node i_s is on, the destination link d where the target node i_d is on, and the current lane number of the subject vehicle v, ℓ_{SV}^c. We will set two sets: permanently and temporarily labeled link, P and P̄, respectively. In addition, for a given link $e \in$ E, these following data will be considered.

- c_e is the cost label of the link e.
- e_i, e_o are enter and out node of link e in moving direction, respectively.
- c_e^s is the smallest cost from the starting point s_i to the link e.
- p_e is the preceding link to link e in the shortest path..
- $\mathscr{A}(e_i)$ is link adjacency list of node e_i.

The detailed steps are given in algorithm 1. This algorithm utilizes the label setting technique [4]. At the beginning of the algorithm, temporary cost are assigned to those links departing from the starting node (line 9 - 10). Line 12-33 is main loop. For each interaction in main loop, the algorithm chooses the temporarily labeled link with minimal cost from the starting node. This link is labeled permanently and added in the preceding link to this link. When an link incident to the final node is reached or there are no more link with temporary labels, the algorithm stops. Then, using the information that is stored in preceding link, the shortest path is found.

IV. SIMULATION

For simulations, a real sub-network is utilized. The choosing sub-network is the surrounding Yeoksam Station. Fig. 1 illustrates the sub-network including seven nodes and eight links with length and limited velocity of each link that are measured from reality. This network is used to implement our algorithm.

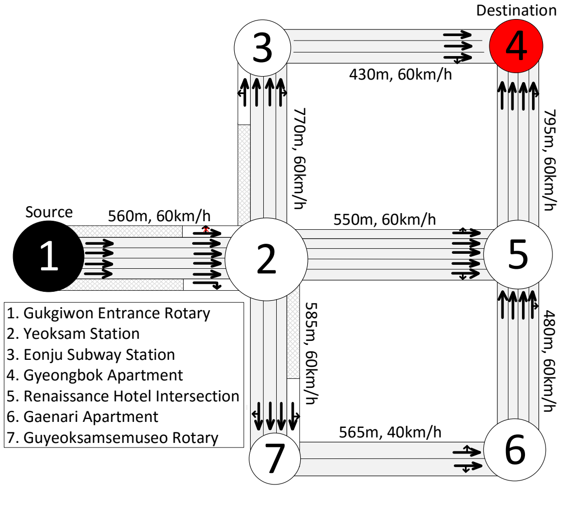

Fig. 1: Test network

First, we compare our proposal with original Dijkstra and link-based Dijkstra with turn restriction [4] for the same problem instance. The results are showed in In Fig. 2. For our proposal, lane number 5 is set as starting lane. If we use original Dijkstra, the shortest path is 1⟶2⟶3⟶4 with the smallest path cost, but the path does not exist when turn left at node 2 is restricted. Next, considering the result of link-based Dijkstra with turn restriction and our proposal, the shortest path is similar but path cost is different. Since going from link l_{12} to link l_{25} is straight through, lane changing is unnecessary, path cost of both algorithms equal. However, travel time changes at link l_{25}. It is the reason why the path cost now is greater than of link-based Dijkstra with turn restriction. Indeed, we need to change lane a lot of time when we are going on the road. Thus, we not only care about shortest path but also realistic path. By including lane changing and turn restriction, our shortest path is more realistic compared with original Dijkstra and link based Dijkstra with turn restriction.

TABLE III: Velocity and density with different congestion level

p	r_{id}								q_{id}							
	l12	l23	l25	l27	l34	l54	l76	l65	l12	l23	l25	l27	l34	l54	l76	l65
1	60	60	60	60	60	60	40	60	0	0	0	0	0	0	0	0
2	54	54	54	54	54	54	36	54	48	48	48	48	48	48	48	48
3	48	48	48	48	48	48	32	48	96	96	96	96	96	96	96	96
4	42	42	42	42	42	42	28	42	144	144	144	144	144	144	144	144
5	36	36	36	36	36	36	24	36	192	192	192	192	192	192	192	192
6	30	30	30	30	30	30	20	30	240	240	240	240	240	240	240	240
7	24	24	24	24	24	24	16	24	288	288	288	288	288	288	288	288
8	18	18	18	18	18	18	12	18	336	336	336	336	336	336	336	336
9	12	12	12	12	12	12	8	12	384	384	384	384	384	384	384	384
10	6	6	6	6	6	6	4	6	432	432	432	432	432	432	432	432

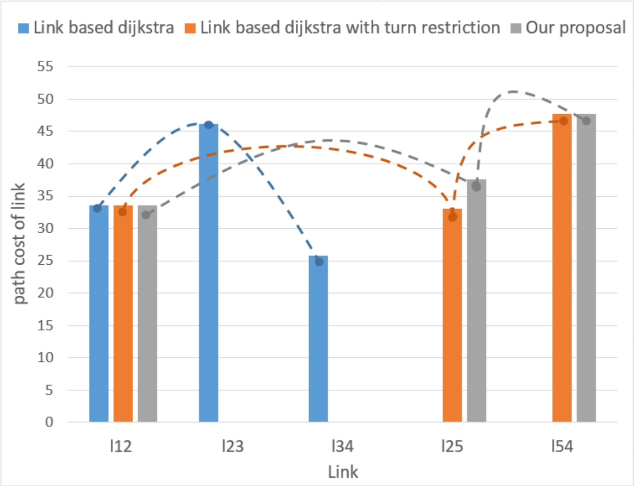

Fig. 2: Shortest path in different algorithms

Next, we evaluate the effect of lane changing with different congestion level. Free velocity, limited speed, is divided into ten parts corresponding to tenth congestion level. The density is calculated by using Greenshield's Model. With value of maximum flow of each link and free speed of each link as shown in Fig. 1, velocity and density according to different congestion level is shown in TABLE III. These data are used to observe the changing of path cost underlying our proposal.

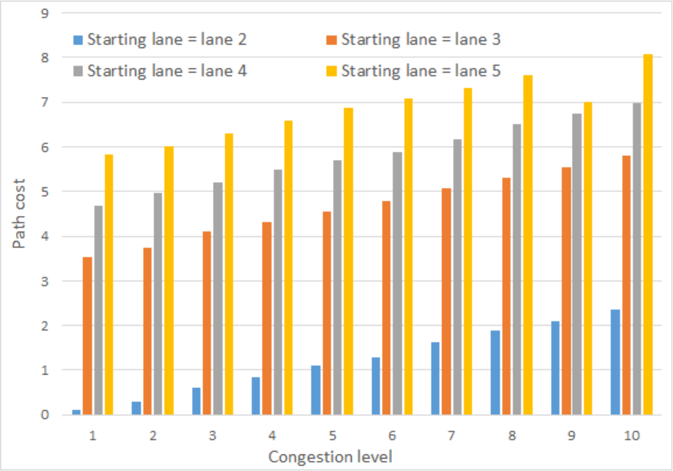

Fig. 3: Route travel time differences by starting lane

When lane changing time is included, a shortest path has different path cost under various starting lane as shown in Fig. 3. Although vehicles are on the same road, the cost of the shortest path is totally dissimilar. As the path has fewer number of lane changing, it has the smallest path cost. Recommended shortest path now is appropriate for each specific vehicle. It is different from Dijkstra without lane changing. A shortest path of that approach always has the same cost for all vehicle on the same original road and forwarding same target place. Our proposal is not only give a practical path but also satisfied path for each individual car.

V. CONCLUSION

In this study, our expectation is selecting a shortest path. However, the importance here is finding the path by taking into consideration about intersections. We need to consider the effect of intersection on path decision. Thus, turn restriction and lane changing are observed. At intersection, if a car need to turn (left, right, or U-turn), it have to change lane. Therefore, the total time must include that action to reflect in the path cost to decide which path is the shortest path. Additionally, in practice, it is not possible to make any turn at any intersection as some roads are restricted by traffic rules. Hence, the discover path must also be a feasible path. We performed simulations to prove that our approach can find the shortest path considering lane changing and turn restriction.

REFERENCES

[1] S. Ezell, "Explaining international it application leadership: Intelligent transportation systems. the information technology and innovation foundation (itif)," *Washington DC*, p. 1, 2010.

[2] I. Yang, H. J. Kim, W. H. Jeon, and H. Kim, "Development of realistic shortest path algorithm considering lane changes," *Journal of Advanced Transportation*, 2015.

[3] E. P. Chan and H. Lim, "Optimization and evaluation of shortest path queries," *The VLDB journal*, vol. 16, no. 3, pp. 343–369, 2007.

[4] E. Gutiérrez and A. L. Medaglia, "Labeling algorithm for the shortest path problem with turn prohibitions with application to large-scale road networks," *Annals of Operations Research*, vol. 157, no. 1, pp. 169–182, 2008.

[5] Q. Li, B. Y. Chen, Y. Wang, and W. H. Lam, "A hybrid link-node approach for finding shortest paths in road networks with turn restrictions," *Transactions in GIS*, 2015.