Finding shortest path considering lane changing and turn restriction

Oanh Tran Thi Kim, VanDung Nguyen, and Choong Seon Hong*
Department of Computer Science and Engineering, Kyung Hee University
{ttkoanh, ngvandung85, 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, usually they consider travel time moving forward on the road and distance from source to 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 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.

Key word: shortest path, lane changing, turn restriction.

1. Introduction

Information technology (IT) has transformed many industries, from education to health care and government, and is now in the early stages of transforming transportation systems. In the leading nations in the world, Intelligent Transportation Systems Information (ITS) bring significant improvement in transportation system performance, including reduced congestion and increased safety and traveler convenience. Intelligent transportation systems include a wide and growing suite of technologies and applications such as real-time traffic information systems, in-car navigation (telematics) systems, vehicle-to-infrastructure integration (V2I), vehicle-to-vehicle integration (V2V), adaptive traffic signal control, ramp metering, electronic toll collection, congestion pricing, fee-based express (HOT) lanes, vehicle usage-based mileage fees, and vehicle collision avoidance technologies [1].

In this paper, we put the finding shortest path problem into perspective of ITS. In South Korea, the Seoul Urban Expressway Traffic Management System can gathers and analyzes traffic information by using cutting-edge video detectors, Dedicated Short Range Communications (DSRC), and CCTVs, and then provides live traffic information through electric road signs, the Internet, and smart phones. So we can get factors affecting road speed, such as weather, time of day, and traffic density. These traffics data are often more useful than the simple Euclidean distance-based computation to choose these routes.

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, usually they consider travel time moving forward on the road and distance from source to 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 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.

Only I. Yang et al [2] has conducted research to incorporate the lane changing in the shortest path problem. For the turn-restrictions, two approaches have been proposed, network modification and link-based labelling strategy. Both methods have disadvantages. Network modification, which requires a long time to load networks from memory, is very time consuming [3]. Link-based labelling 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 improve trip duration.

The remainder of this paper is organized as follows. The algorithm for finding shortest path considering lane changing and turn restriction is analyzed in Section 2. Section 3 draws simulation. Section 4 summarizes the paper.

2. Algorithm

Before, we illustrate the algorithm in details, we will present about the lane changing and turn restriction problem first. Lane information of links in the work are managed and stored at database of traffic management system. Two kinds of lane information of link are the associated lane map, ALM and outing lane information, ELI. ALM is used to determine entering lane number. ELI is used to find outing lane number. Based on the outing lane and entering lane, path cost will be determined. Suppose that the link that a car is going on is link i, the next link is j, the type of turn to go to link j is tt_{ij}. Then the entering lane number l_j^{in} and outing lane number l_i^{out} are found by using Equation. 1 and 2, respectively.

$l_j^{in} = f_1(tt_{ij}, l_i^{out}, ALM_{ij})$ (1) and $l_i^{out} = f_2(tt_{ij}, ELI_{ij})$ (2)

Path cost is total travel time including travel time of moving forward and time of lane changing.

Given a road network G (V, E), a set of speed patterns S, a set of traffic patterns R, and a query q← (s, d, start time). Suppose at the start time, we have the start link s where the starting node is i_s is on, the destination link d where the target node id 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 $\overline{P}$, 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_0 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.
- l_e^c is current lane number.
- p_e is the preceding link to link e in the shortest path.
- $SDS(i)$ is a A set of successor nodes i
- $A(e_i)$ is link adjacency list of node e_i.

The detailed steps are given in figure 1.

This algorithm utilizes the label setting technique [19]. 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 the 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 a 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.

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 nodes $s, d \in$ G.
Output: shortest P^* from i_s to i_d

1: $\mathrm{P} \leftarrow \emptyset, \bar{\mathrm{P}} \leftarrow \mathrm{E}$
2: **for all** $e \in \mathrm{E}$ do
3: $\quad c_e^s \leftarrow +\infty$
4: $\quad p_e \leftarrow$ null
5: **end for**
6: $c_s = 0$
7: $p_s = 0$
8: $\ell_s^c = \ell_{SV}^c$
9: **for all** $e = (\mathrm{i}_s, \mathrm{j}) \in \mathrm{E}, \mathrm{j} \in \mathrm{SDS(i)}$ do
10: $\quad c_e^s \leftarrow c_e$
11: **end for**
12: **while** $|\mathrm{P}| < |\mathrm{E}|$
13: $\quad \bar{e} \leftarrow \arg\min_{e \in P} (c_e^s)$
14: $\quad \mathrm{P} \leftarrow \mathrm{P} \cup \{\bar{\mathrm{e}}\}$
15: $\quad \bar{\mathrm{P}} \leftarrow \bar{\mathrm{P}} \setminus \{\bar{\mathrm{e}}\}$
16: $\quad$ **if** $(c_{\bar{e}}^s) = \infty$ **then**
17: $\qquad$ There is no route between i_s and i_s
18: $\qquad$ STOP
19: $\quad$ **else**
20: $\qquad$ **if** $\bar{\mathrm{e}} = (\mathrm{i}, i_d), \mathrm{i} \in \mathrm{E}$ **then**
21: $\qquad\quad$ Build P^* backtracking from $p_{\bar{e}}$
22: $\qquad\quad$ STOP
23: $\qquad$ **end if**
24: $\quad$ **end if**
25: $\quad$ **for all** e such that $(\bar{\mathrm{e}}, \mathrm{e}) \in \mathscr{A}(\bar{\mathrm{e}}_o)$ and $e \in \bar{\mathrm{P}}$ **do**
26: $\quad$ **if** $c_e^s > c_{\bar{e}}^s + c_e$ **then**
27: $\qquad c_e^s \leftarrow c_{\bar{e}}^s + c_e$
28: $\qquad p_e \leftarrow \bar{e}$
29: $\qquad$ Determine ℓ_e^{in} by using Equation .1.
30: $\qquad$ Determine $\ell_{\bar{e}}^{out}$ by using Equation .2.
31: $\quad$ **end if**
32: $\quad$ **end for**
33: **end while**

Figure 1: Algorithm for finding shortest path considering lane changing and turn restriction

3. Simulation

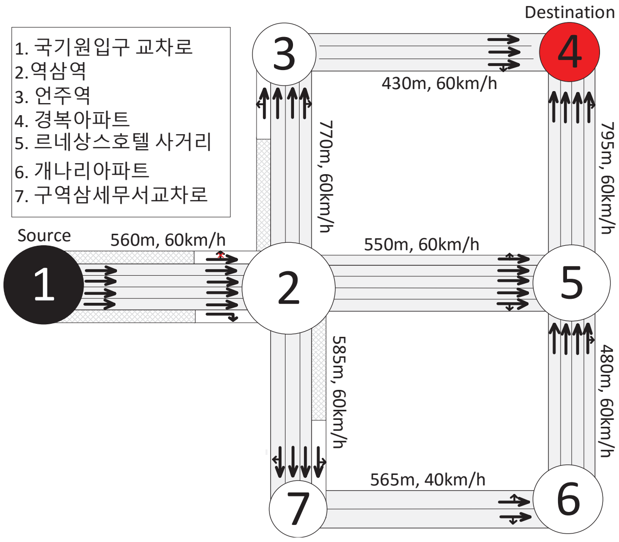

Figure 2: Test network

For simulations, a real sub-network is utilized. The choosing sub-network is the surrounding 역삼역. Figure 2 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.

A test network is given in Figure. 2.

We compare our proposal with original Dijkstra and link-based Dijkstra with turn restriction [4] for the same problem instance. In Fig. 3, the horizontal axis shows path cost of link. Path cost is observed as travel time on each link for original Dijkstra and link-based Dijkstra with turn restriction. For our proposal, lane number 5 is set as starting lane. Value of vertical axis indicates links which participated in shortest path. It is obvious that shortest found by our proposal have higher path cost than others. However, it is extremely important, if we use original Dijkstra, the result is unfeasible path. Shortest path of original Dijkstra is 1→2→3→4. It has the smallest path cost, but the path does not exist when turn left at node 2 is restricted. Therefore, taking turn restriction into finding shortest path problem is necessary to ensure that recommended path is feasible. 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 l12 to link l25 is straight through, lane changing is unnecessary, path cost of both algorithms equal. However, travel time changes at link l25. To enter in link l25, it has to change from lane 5 at entering time to lane 1 at exit time on link l25. 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.

4. Conclusion

In this paper, our expectation is selecting a shortest path. However, the importance here is finding the path by taking into consideration about intersections. Our main target is non-stop traffic signal intersection, and thus, we need to consider the effect of intersection on path decision. Therefore, 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.

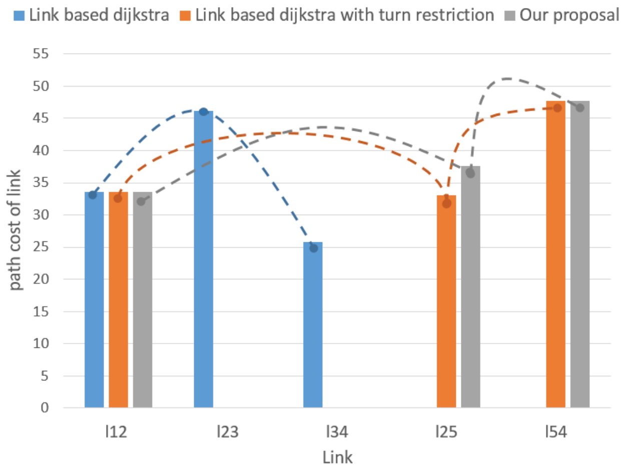

Figure 3: Shortest path in different algorithms

5. Acknowledgement

This research was supported by Basic Science Research Program through National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2014R1A2A2A01005900). Dr. CS Hong is the corresponding author.

6. Reference

[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. Guti´errez 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.