Efficient Route Determination Technique in LBS System

Sung-Soo Kim, Kwang-Soo Kim, Jae-Chul Kim, Jong-Hun Lee
Spatial Information Technology Center
Computer & Software Research Laboratory
Electronics and Telecommunications Research Institute
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, South Korea

sungsoo@etri.re.kr

Abstract—**Shortest Path Problems are among the most studied network flow optimization problems, with interesting applications in various fields. One such field is the route determination service, where various kinds of shortest path problems need to be solved in location-based service.**

Our research aim is to propose a route technique in real-time location-based service (LBS) environments according to user's route preferences such as shortest, fastest, easiest and so on. Turn costs modeling and computation are important procedures in route planning. There are major two kinds of cost parameters in route planning. One is *static cost parameter* **which can be pre-computed such as distance and number of traffic-lane. The other is** *dynamic cost parameter* **which can be computed in run-time such as number of turns and risk of congestion.**

In this paper, we propose a new *cost modeling method* **for turn costs which are traditionally attached to edges in a graph. Our proposed route determination technique also has an advantage that can provide service interoperability by implementing XML web service for the** *OpenLS route determination service specification*. **In addition to, describing the details of our shortest path algorithms, we present a location-based service system by using proposed routing algorithms.**

Keywords— **Location-Based Services, Route Planning, Web Services, Shortest Path Algorithms, Open Systems.**

I. Introduction

RE cently, the location-based services (LBS) is more and more popular as the killer applications in the ubiquitous environments. LBS is made available through systematically combining location-based content, GIS information, a terminal interface, and applications focused on the terminal user's location.

One of the issues in LBS system development is the *service interoperability*. The LBS systems should provide the interoperability among different LBS service platforms to widely provide a variety of services in the ubiquitous environments, such as location acquisition, location utility, directory, presentation service and so on. Therefore, we need the standard service interfaces for the implementation of the LBS open system.

This paper describes the portable and interoperable approach for providing a routing service for the large road networks by implementing a *web services* which provide platform independency. The ultimate goal of this research is developing route planning service for Open LBS system.

II. Related Works

The shortest paths problem in networks has been the subject of extensive research for many years resulting in the publication of a large number of scientific papers[Pallottino1998]. The algorithm which always select the candidate node with minimum distance label are known as *label-setting* or as *shortest-first search* algorithms. The first algorithm using this selection rule was proposed by Dijkstra. There are many versions of Dijkstra's algorithm depending on data structures for the candidate nodes, Q. The algorithms which select a candidate node by means of different strategies are known as *label-correcting* or as *list-search* algorithms. Table 1 shows the shortest path algorithms, their properties and time complexity.

Approach	Algorithms	Properties / Data structures	Time complexity
Label setting	Naive algorithm	unordered linked list	$O(n^2)$
	S-heap	binary heap	$O(m\log n)$
	S-bucket	bucket	$O(m + nc_{max})$
	F-heap	fibonacci heap	$O(n\log n + m)$
	Symmetric	bi-directional search	$O(b^{n/2})$
Label correcting	L-queue	queue	$O(mn)$
	L-deque	deque	$O(n2^n)$
	Topological ordering	topological graph visiting	$O(mn)$
Heuristic	A* search	heuristic for search procedure	$O(b_e^d)$
	Weighted A* search	parametrize multiplicative factor	$O(b_e^d)$
	Radius search	hierarchical search	$O(b_e^d)$

Table 1. The shortest path algorithms and their properties. (n : the number of nodes, m : the number of edges, c_{max} : the maximum edge cost, b_e : the effective branching factor.)

The analysis of transportation networks is one of the many application areas where the computation of shortest paths is one of the most fundamental problems. Since no "best" algorithm exists for every kind of transportation problem, i.e. no algorithm exists which shows the same practical behavior independently of the structure of the graph, of its size and of the cost measure used for evaluating the paths, research in this field has recently moved to the design and the implementation of "heuristic" shortest path algorithm, which are able to capture the peculiarities of the problems under consideration[Ahn2002]. Much of the focus has been on the choice and implementation of efficient data structures[Pallottino1998].

The major contribution of this work is to present a route determination service by providing service interoperability among other LBS services.

III. Route Planning Algorithm using Dual Graph

A. The Shortest Path Problem

We introduce some notation and give some definitions related to the shortest path problem.

Let $G = (N, E)$ be a simple directed graph, where N is the set of the *nodes*, of cardinality *n*, and E is the set of the *edges*, of cardinality *m*. We distinguish the start node s and the end node t. Let w: $E \rightarrow R$ be a function which assigns a cost w_{ij} to each $(i, j) \in E$. A path p from node $s(e_1)$ to $t(e_k)$ is the sequence of edges $p = (e_1, ..., e_k)$ such that $t(e_i) = s(e_{i+1}), i = 1, ..., k-1$ and $e_i \neq e_j$ for $i \neq j$. The path include cycles if there exist any pairs $s(e_i) = t(e_j)$ for $i < j$. The length of a path is k; the total cost of a path is

$$W(p) = \sum_{i=1}^{k} w(e_j)$$

Given a root $r \in N$, the *Shortest Path Tree problem* (SPT) consists in finding a directed tree T such that the path from r to i in T is one of the shortest paths from r to i in G, for each $i \in N$ which is connected to r, i.e. a directed path from r to i exists.

B. The Line Digraphs

For a directed graph G, the *line digraph* $D = \mathcal{L}(G)$ has vertex set $N(D) = E(G)$ and edge set $E(D) = \{ab : a, b \in N(D)$, the head of a coincides with the tail of $b\}$[Jensen2002]. Given a graph $G(N, E)$, which will be called the *primal graph* for better distinction, its line graph is defined as follows[Anez1996][Winter2002B]:

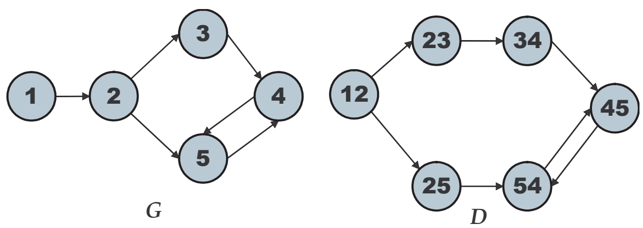

Fig. 1. A digraph G and its line digraph $D = \mathcal{L}(G)$.

Definition 1: The graph $D(N_D, E_D)$ with s_D, t_D, w_D is called the *linear dual graph* of $G(N, E)$ with s_G, t_G, w_G if:

- For each edge $e_i \in G$ there is a node $v_i \in D$ with $v_i = d(e_i)$ and d being bijective so that $d^{-1}(v_i) = e_i$. Thus, $N_D = d(E_G)$.
- For each pair of consecutive edges $(e_i, e_j) \in G$ there is an edge $\varepsilon \in D$ between the corresponding nodes $v_i = d(e_i)$, $v_j = d(e_j)$, such that $s(\varepsilon) = v_i$ and $t(\varepsilon) = v_j \cdot E_D = \bigcup_i \varepsilon_i$.
- There is a cost function $w^{\varepsilon} : E_D \rightarrow \mathbf{R}^+$, and a cost function $w^v : N_D \rightarrow \mathbf{R}^+$.

C. Turn Prohibitions Problem

In urban private car models, turning left is often forbidden in order to minimize the congestion and, when allowed, traffic lights and counter flow cause an extra travel cost which has to be taken into account when finding the minimum cost itinerary between two points. Generally speaking, given a graph $G = (N, E)$, where c_{ij} denotes the cost of the edge $(i, j) \in E$, assume that a *penalty* $\sigma_{ab} (\geq 0)$ is associated with each pair of *subsequent edges* $a = (i, j)$ and $b = (j, k)$. Given graph G, the turn prohibition has the following conditions.

- $0 \leq \sigma_{ab} \leq n$: passable, n denotes a positive number.
- $\sigma_{ab} = \infty$: turn prohibition.

Several approaches have been followed to address the turn prohibition problem among the others. The common way to handle turn restrictions in current network analysis tools is *node expansion*[Winter2002A]. This approach builds up the expanded network, G_e, which is obtained by highlighting each movement in the intersections by means of dummy nodes and edges, where the costs of the dummy edges are the penalties. The major disadvantage of this approach is that the resulting network G_e is significantly larger than the original graph G.

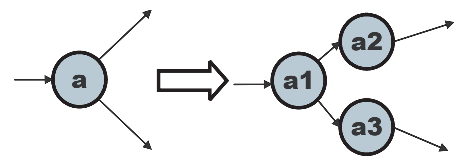

Fig. 2. Node Expansion.

The other approach works on the so-called *"dual network"* by using *linear digraph*, which has been mentioned in previous section. In determining the performance of the approaches, it is useful to summarize their storage requirements. The node expansion method requires $\delta_{max}|N_G|$ nodes for the primal graph G since any node n expands into $\delta(n)$ nodes (where, δ_{max} denotes maximum degree of node). This method also requires the number of edges in G, $|E_G|$, plus the number of path of length 2, $\frac{\delta_{max}}{2}|E_G|$[Winter2001]. The *boundary node* is a node which has a single-source and single-target in G. The linear dual graph (D) method requires the number of nodes which $|E_G|$ adds to the number of boundary nodes in G, $|N'_G|$. The storage requirement for the edges in D is the number of edges which $\frac{\delta_{max}}{2}|E_G|$ plus $|N'_G|$ (Table 2).

Method	$\|N\|$	$\|E\|$
Node expansion (G_e)	$\delta_{max}\|N_G\|$	$(1 + \frac{\delta_{max}}{2})\|E_G\|$
Dual graph (D)	$\|E_G\| + \|N'_G\|$	$\frac{\delta_{max}}{2}\|E_G\| + 2\|N'_G\|$

Table 2. Estimation of upper limits for node expansion and linear dual graph. (N'_G denotes boundary nodes of a primal graph G and δ_{max} denotes maximum degree of node)

D. Route Planning for Route Preference

The *route service* is a network-accessible service that determines travel routes and navigation information between two or more points. The OpenLS route determination service specification is one of the OpenLS service specifications for route service[OpenLS2002].

The RoutePlan abstract data type (ADT) specifies the criteria upon which a new route is determined (Fig. 3). The RoutePlan has the RoutePreference ADT which is routing preference to be taken into consideration when determining the route. The following preferences are defined in the specification.

- "Fastest" - Minimize the travel time.
- "Shortest" - Minimize the travel distance.
- "Easiest" - Minimize the driving difficulty.
- "Pedestrian" - Best route by foot.
- "PublicTransportation" - Best route by public transportation.

In "Fastest" preference case we use the cost function $\omega(e_i)$: $\omega(e_i) = c_i \cdot \lambda$, which includes road information, such as, a

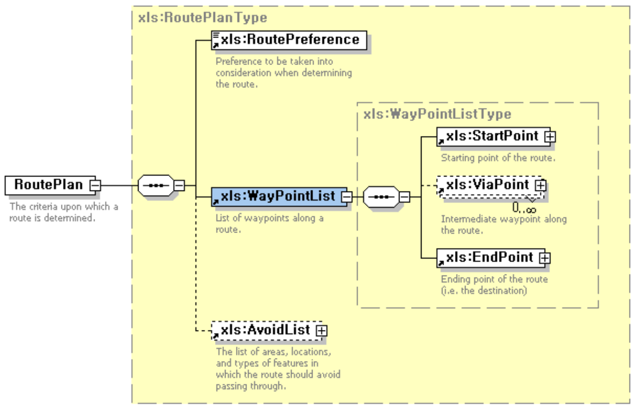

Fig. 3. RoutePlan abstract data type of the OpenLS.

regular speed according to the road classes and the number of traffic-lane. We also perform graph search by applying distances to $\omega(e_i)$ in "Shortest" preference case. In "Easiest" preference case we construct the graph by using adjacent angles, θ_i, as $\omega(e_i)$ (Table 3).

Route Preference	Cost Function
Fastest	$\omega(e_i) = c_i \cdot \lambda$ $(\lambda = \frac{1}{\tau \cdot m}, 0 < \lambda \leq 1)$
Shortest	$\omega(e_i) = c_i$
Easiest	$\omega(e_i) = \theta_i$

Table 3. The cost functions for the route preferences(τ : regular speed).

IV. Implementation

We have implemented a routing service in Java as a web service, tested our implementation on the large road network in Seoul, Korea, about 160,000 nodes and 120,000 links, including turn and one-way restrictions. In our implementation code, we use a *modified Dijkstra's algorithm* which use paring heaps and binary heaps and the time complexity is $O(m\log n)$. The program generates the line graph in $O(\delta_{max}^2 n)$ from the road network database. In our research case, $\delta_{max} = 8$, therefore, we can construct the line graph in linear time.

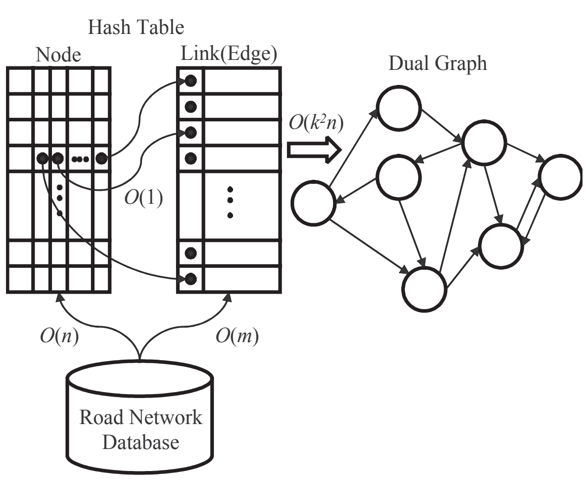

Fig. 4. Dual Graph Transformation($k = \delta_{max}$).

The traffic network database has a node table and link table. We use **DB** to denote the road network database which is maintained in the Oracle Spatial 9i DBMS. The objects for geometry column access are implemented by using Oracle Spatial Java Library[Oracle2002]. The relation schema of node and link to describe the association can be represented as the followings. The ID would serve to uniquely identify nodes and links. The

DB = { NODE, LINK }
Node(ID, LINKNUM, ADJNODE, GEOMETRY),
ADJNODE = (ID, PASSINFO, ANGLE),
$0 \leq |\text{ADJNODE}| \leq 8$, GEOMETRY = $\{P | P_i \in \mathbf{R^2}\}$.
Link(ID, SN, TN, DIST, ROADCLASS, LANECNT, GEOMETRY),
GEOMETRY = $\{(P_i, ..., P_k) \,|\, P_i \in \mathbf{R^2}, k \geq 2\}$.

LINKNUM denotes the number of adjacent links ADJNODE. The LINK table has start node (SN), destination node (DN), distance (DIST), class of road (ROADCLASS), the number of traffic-lane (LANECNT) and geometry (GEOMETRY). The ADJNODE includes pass information at the intersection (PASSINFO), adjacent angle (ANGLE).

V. Conclusion

We presented an approach to model turning costs in road networks for the OpenLS route determination service, based on a linear dual graph. The work is a part of effort to develop better location-based services. In addition to, we provide the interoperability for Open LBS platform by implementing the routing web service compliant to OpenLS specification. The dual graph represents all pair of consecutive edges and allow individual weighting. The data structure of line graph is simple as a primal graph. We can simply construct a line graph by replacing edges by nodes and adding all paths of length 2 by an edge and shortest path algorithms can be applied unchanged.

Future works include :

- Topology simplification for large traffic networks based on the locality of navigation.
- Hierarchical data structure for simplified networks to improve the processing speed.

References

[Ahn2002] Chang Wook Ahn, R. S. Ramakrishna, A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations, *IEEE Transaction on Evolutionary Computation*, pp. 566-579, Vol. 6, No. 6., 2002.

[Anez1996] J. Anez, T. de la Barra, B. Perez, Dual Graph Representation of Transport Networks, In *Transportation Research*, Vol. 30, No. 3, pp. 209-216. 1996.

[Jensen2002] Jorgen Bang-Jensen, Gregory Gutin, DIGRAPHS: Theory, Algorithms and Applications, *Springer*, pp. 182-187, 2002.

[OpenLS2002] OpenLS Initiative, OpenLS Route Determination Service Specification, *OpenGIS Interoperability Program Report* (OGC 02-090), Open GIS Consortium Inc., 11 November 2002.

[Oracle2002] Oracle Technology Network, Oracle Spatial Java Library User's Guide Release 8.1.7 or later, *Oracle Spatial Documentation*, pp. 12-20, July. 2002.

[Pallottino1998] Stefano Pallottino, Maria Grazia Scutella, Shortest Path Algorithms in Transportation Models: Classical and Innovative Aspects, Technical Report, Univ. of Pisa, 1998.

[Winter2001] Stephan Winter, Weighting the Path Continuation in Route Planning, In *Proceedings of 9th ACM International Symposium on Advances in Geographic Information Systems*, pp. 173-176, 2001.

[Winter2002A] Stephan Winter, Route Specifications with a Linear Dual Graph, In *Proceedings of Advances in Spatial Data Handling*, pp.329-338, 2002.

[Winter2002B] Stephan Winter, Modeling Costs of Turns in Route Planning, *GeoInformatica*, Vol. 6, No. 4, pp. 345-360, 2002.