

듀얼 그래프를 이용한 라우팅 서비스 구현

김성수, 허태욱, 박종현, 이종훈
한국전자통신연구원 공간정보기술센터

Routing Service Implementation using a Dual Graph

Sung-Soo Kim, Tae-Wook Heo, Jong-Hyun Park, Jong-Hun Lee
Spatial Information Technology Center, ETRI
e-mail : 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. 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*.

I. 서론

최근 유비쿼터스 환경의 킬러 서비스로 위치기반서비스(LBS)가 주목을 받고 있다. 유비쿼터스 환경에서 LBS가 중요한 서비스로 널리 보급, 확산되기 위해서는 위치정보를 획득하고 서비스를 제공하는 다양한 장치들 간에 상호운용성이 제공되어야 한다. 즉 서로 다른 장치 및 플랫폼들 간에 위치정보 획득 및 서비스 제공에 대한 표준화된 인터페이스가 필요하다.

본 논문에서는 OpenLS의 "Route Determination Service"의 표준인터페이스[3]를 따르고, 개방형 LBS 플랫폼상의 상호운용 가능한 경로탐색 서비스의 시스템 구조 및 경로탐색 알고리즘을 제안한다.

II. 관련연구

다양한 교통체계문제에 대한 최적의 알고리즘이 아직까지 없기 때문에, 최근 연구 추세는 문제의 최적의 솔루션을 찾기보다는 탐색시간의 수행성능을 향상시킬 수 있는 휴리스틱(heuristic)한 최단 경로탐색에 대한 설계 및 구현이 연구되고 있다[7]. 표 1은 최단경로탐색을 위해 제안되어진 알고리즘들의 속성과 성능을 비교 분석하였다[8].

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	$O(m + n c_{max})$
	F-heap	fibonacci heap	$O(n \log n + m)$
	Symmetric	bi-directional search	$O(b^{1/2})$
Label correcting	L-queue	queue	$O(mn)$
	L-deque	deque	$O(n2^k)$
	Topological ordering	topological graph visiting	$O(mn)$
Heuristic	A* search	heuristic for search procedure	$O(lg)$
	Weighted A* search	parametrize multiplicative factor	$O(lg)$
	Radius search	hierarchical search	$O(lg)$

표 1. 최단경로 탐색 알고리즘들의 성능비교

III. 듀얼 그래프를 이용한 최단경로 탐색

3.1. 그래프 (Graphs)

최단 경로탐색 문제를 정형화하기 위해 그래프에 대한 개념을 소개한다.

그래프 G 는 노드집합 N 과 에지집합 E 로 구성된다. 에지는 노드들 사이의 이진관계, $e: N \rightarrow N$, 를 나타낸다. 출발노드 $s, s: E \rightarrow N$ 는 목적노드 $t, t: E \rightarrow N$ 로 표기한다.

일반적으로 에지는 비용 $w: E \rightarrow \mathbb{R}^+$ 를 가지게 되는데, 도로망에서는 초기값은 두 노드간의 거리이며, 사용자 경로탐색방식 선호도에 따라 변경된다.

$s(e_i)$ 에서 $t(e_k)$ 까지의 경로(path) p 는 $i \neq j$ 에 대해 $t(e_i) = s(e_{i+1}), i=1, \dots, k-1$ 이고 $e_i \neq e_j$ 인 에지들의 시퀀스 $p=(e_1, \dots, e_k)$ 가 된다. 경로는 $i < j$ 에 대해 $s(e_i) = t(e_j)$ 인 하나이상의 쌍이 존재하면 사이클(cycle)을 포함한다. 경로의 길이(length)가 k 일 때, 경로의 전체 비용은

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

가 된다[2].

3.2. 라인 유향그래프 (Line Digraphs)

그래프 G 에 대하여, 라인 디그래프 $D = L(G)$ 는 $N(D) = E(G)$ 인 정점집합을 가지며, 에지집합

$E(D) = \{ab: a, b \in N(D), a \text{의 시작과 } b \text{의 끝은 일치}\}$ 을 가진다 (그림 1).

정의 : 주어진 그래프 $G(N_G, E_G)$ 에 대해 다음 속성을 가지는 그래프 $D(N_D, E_D)$ 를 그래프 G 에 대한 선형 이원그래프(linear dual graph; 이하 듀얼 그래프) 혹은 라인 그래프 (line graph) 라고 한다[6].

- G 의 모든 에지 e_i 에 대하여 D 내에 함수 $d: v_i = d(e_i)$ 에 의해 부여된 노드 v_i 가 있다. d 는 $d^{-1}(v_i) = e_i$ 를 만족하는 전단사 함수(bijective function)이다. 따라서, $N_D = d(E_G)$ 이다.
- 그래프 G 에서 $t(e_i) = s(e_j)$ 인 각 에지쌍 (e_i, e_j) 에 대하여 D 내에 대응되는 노드들($v_i = d(e_i), v_j = d(e_j)$) 사이의 에지 ε 가 존재한다. 여기서, $s(\varepsilon) = v_i, t(\varepsilon) = v_j$ 이다. $E_D = \cup_i \varepsilon_i$.
- 비용함수 $w_D: E_D \rightarrow \mathbb{R}^+$

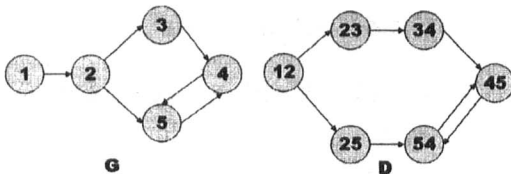


그림 1. 그래프 G 와 라인 디그래프 $D = L(G)$

3.3. 회전문제 (Turn Restrictions Problem)

도시의 도로망에는 다수의 좌회전 금지와 함께 유턴이 존재한다. 이는 교차로에서 좌회전하고자 하는 차량의 대기행렬로 인한 주요통로의 소통 저해를 최소화하기 위해 시행되고 있다. 따라서, 경로안내체계를 위한 최적경로 탐색 알고리즘은 유-턴, 피-턴, 회전금지 등이 반영된 정확한 최적경로를 제공할 수 있어야 한다[1]. 주어진 c_{ij} 를 에지 $(i, j) \in E$ 의 가중치로 갖는 그래프 $G = (N, E)$, G 상의 연속된 에지 $a = (i, j)$ 와 $b = (j, k)$ 에 대해 a 에서 b 로 가는 패널티(penalty) $\sigma_{ab} (\geq 0)$ 이 존재한다고 할 때, 다음과 같은 통행 조건을 가진다.

- $\sigma_{ab} \geq 0$ 인 양수 : 통행가능
- $\sigma_{ab} = \infty$: 통행금지

도로망에서 회전문제를 해결하기 위한 접근방법으로, 도로망상의 교차노드에 허상노드(dummy node)와 에지를 추가하는 노드 확장(node expansion) 방법과 앞서 언급한 선형이원그래프를 이용한 방법이 있다[5].

노드 확장법(그림 2)은 원래 도로망 그래프 G 에서 노드 확장된 그래프 G_e 를 생성하기 위해 많은 허상노드와 에지가 추가되므로 저장공간을 많이 요구하는 단점이 있다.

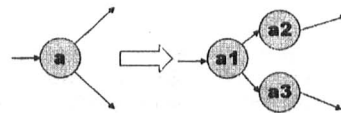


그림 2. 노드 확장법 (node expansion)

듀얼 그래프를 이용할 경우 그래프의 크기를 조사해 보면, 듀얼 그래프 D 의 노드 수 $|N_D|$ 는 주어진 그래프 G 의 에지 수 $|E_G|$ 와 같으며, D 에서 에지 수는 G 에서 경로길이가 2 인 경로의 수와 같다.

G 의 노드의 차수(degree) δ 에 대해 Cauchy-Schwarz 부등식을 기반으로 상한수(upper estimation)를 구하면

$$|E_D| \leq |E_G| \frac{\delta_{\max}}{2} \text{ 가 됨을 알 수 있다[2].}$$

노드 확장법의 경우, 노드 확장 그래프 G_e 의 노드수는 주어진 그래프 G 의 각 노드 차수만큼 더 추가되므로 $|N_{G_e}| = \delta_{\max} |N_G|$ 가 되며 에지수는 G 에 있던 에지수에서 경로길이가 2 인 경로의 수 $(|E_G| \frac{\delta_{\max}}{2})$ 를 합

친 것으로, $|E_{G_e}| = (1 + \frac{\delta_{\max}}{2}) |E_G|$ 가 된다.

아래 표 2 는 이러한 요구되는 저장공간측면에서 두 기

법을 비교한 것이다.

기법	$ N $	$ E $
노드확장법 (G_c)	$\delta_{\max} N_G $	$(1 + \frac{\delta_{\max}}{2}) E_G $
듀얼그래프 (D)	$ E_G $	$\frac{\delta_{\max}}{2} E_G $

표 2. 노드확장법과 듀얼그래프의 공간적 비교

(δ_{\max} : 노드의 최대차수)

본 논문에서는 듀얼그래프 기법을 이용하여 경로탐색 서비스를 수행하였다. 여기서, 주어진 그래프 G 에서 듀얼 그래프를 생성하는 데 시간을 줄이기 위하여 데이터 베이스의 노드 테이블과 링크(에지) 테이블의 상호 연관 관계를 이용하여 듀얼 그래프를 이진힙(binary heap)과 해쉬 테이블 구조를 이용하여 직접 생성할 수 있도록 구현하였고, 수정된 Dijkstra 알고리즘(modified Dijkstra's algorithm)을 이용하여 경로 탐색을 수행하였다[4].

따라서, binary heap 을 생성하는 데 $O(\log N)$ 이 요구되며, 만약 임의의 한 노드에서 모든 노드까지 도달 가능하다면 전체 요구되는 시간은 $O((N+E)\log N)$ 이 된다.

3.4. 사용자 선호도에 따른 경로 탐색

본 연구에서 구현 시 사용된 데이터는 한국도로정보협회(KRIS)의 KRIS 포맷을 도로 노드(Node) 테이블과 링크(Link) 테이블로 구분하여 데이터베이스(DB)에 저장하여 사용하였다.

노드 테이블에는 해당노드의 식별자(id), 인접한 링크수(linkNum), 최대 8 개의 인접노드에 대한 정보(adjNode) 및 기하정보(geometry) 등이 저장되어 있으며, 링크 테이블은 id, 시점 및 종점노드의 id(sn, tn), 도로거리(dist), 도로종류(roadClass), 차선수(laneCnt) 및 기하정보 등이 저장되어 있다.

여기서, 인접노드에 대한 정보는 노드 식별자(id), 인접노드와의 교차점 통행정보(passInfo), 그리고 현재 노드와의 인접각도(angle)가 있다.

이러한 도로망 데이터베이스의 스키마(schema)는 다음과 같이 나타낼 수 있다.

$DB = \{Node, Link\}$

$Node(id, linkNum, adjNode, geometry),$

$adjNode = (id, passInfo, angle),$

$0 \leq |adjNode| \leq 8, geometry = \{P \mid P_i \in \mathbb{R}^2\}.$

$Link(id, sn, tn, dist, roadClass, laneCnt, geometry),$

$geometry = \{(P_1, \dots, P_k) \mid P_i \in \mathbb{R}^2, k \geq 2\}.$

사용자가 선호하는 경로탐색결과를 제공하기 위해 OpenLS에서 정의하고 ADT (Abstract Data Type)의 경로 요청에 포함된 타입인 RoutePlanType 에 따른 서비스 설계가 필요하다[3].

사용자의 경로탐색 시 고려되는 경로탐색 기준은 RoutePreference 에 정의되어 있으며, 크게 basic profile 과 full profile 로 나누어진다.

● Basic profile

- Fastest : 경로에 대한 주행시간(travel time)을 최소화하는 경로를 선택
- Shortest : 경로에 대한 거리(travel time)을 최소화하는 경로를 선택

● Full profile

- Easiest : 경로에 대한 회전(Turn) 수 및 주행난이도를 최소화하는 경로를 선택
- Pedestrian : 도보로 이용할 때 좋은 경로를 선택
- PublicTransportation : 대중 교통을 이용할 때 좋은 경로를 선택

사용자는 아래 5 가지 탐색 선호기준 중 하나를 선택하게 된다. 동시에 두 가지 기준을 만족하는 최단경로를 찾는 일반적인 문제를 bicriterion shortest path problem 이라고 하며 이 문제는 NP-hard 문제다.

여기서, Fastest 기준에 대해서는 고속도로, 도시고속도로, 일반국도 등으로 구분되는 도로종류(roadClass)와 차선수(m)를 비용함수 $\omega(e_i) : \omega(e_i) = c_i \cdot \lambda, 0 < \lambda \leq 1$, 에 반영하여 그래프를 탐색하였다.

Shortest 기준은 도로거리(dist) 만을 고려하였고 ($\omega(e_i) = c_i$), Easiest 기준은 그래프의 가중치를 인접각도(angle)로 두고 알고리즘을 수행하였다($\omega(e_i) = \theta_i$).

경로탐색기준 중에서 도보나 대중교통을 기준으로 하는 경우는 그래프의 구조 및 대중교통망 데이터 구조가 다르기 때문에 본 논문에서 논의하지 않는다.

경로탐색기준	비용함수
Fastest	$\omega(e_i) = c_i \cdot \lambda$ ($\lambda = 1/(\tau \cdot m), 0 < \lambda \leq 1$)
Shortest	$\omega(e_i) = c_i$
Easiest	$\omega(e_i) = \theta_i$

표 3. 경로탐색기준에 따른 비용함수 (τ : 규정속도)

