

■ 論 文 ■

링크표지확정 다수경로탐색 알고리즘과 대안경로선정을 위한 활용

A Kth Shortest Path Algorithm with the Link-Based Label Setting Approach and Its Application for An Alternative Routes Selection

이 미 영

(한국건설기술연구원 첨단도로시스템연구센터
객원수석연구원)

백 남 철

(한국건설기술연구원 첨단도로시스템연구센터
선임연구원)

강 원 의

(한국건설기술연구원 첨단도로시스템연구센터
수석연구원)

신 성 일

(서울시정개발연구원 도시교통연구부 부연구위원)

목 차

- I. 서론
- II. 이론적 배경
 - 1. 표식
 - 2. 표지확정 최단경로알고리즘
 - 3. 노드표지확정 다경로알고리즘
- III. 링크표지확정 다경로알고리즘
 - 1. 링크표지확정 다경로알고리즘
- IV. 사례연구
 - 1. 알고리즘 수행과정
 - 2. 회전금지 및 U-Turn
 - 3. 대안경로
- V. 결론
- 참고문헌

Key Words : Label Setting, Kth Shortest Path Algorithm, Reasonable Path Choice, Link Label, Alternative Route

요 약

합리적 통행경로는 "경로를 노드 또는 링크의 순서로 표현할 때, 경로를 구성하는 노드의 반복은 존재해도 링크의 반복은 존재하지 않는다"는 원리에 근거한다. 최적경로 탐색과정에서 합리적 통행현상을 포함하는 방법은 링크로 구성된 표지(Link-Based Label)를 적용하는 방안이 적용하기 용이하다. 링크표지를 활용하는 경우 링크의 중복 표현을 허용되지 않는 상황에서 노드의 중복표현이 가능하여 합리적인 통행을 원칙적으로 보장될 뿐만 아니라 목적지에서 출발지로 경로를 역 추적(Backward Trace)하는 과정에서 전 링크(Previous Link)의 단일정보만 이용하므로 Bellman의 최적원리(Optimality Condition)에 의한 최적해(Optimal Solution)가 보장된다. 본 연구는 다수의 경로를 선정함에 있어 합리적 통행행태를 고려하기 위한 링크표지고정방식(Link-Based Label Setting)을 제안한다. 기존에 표지고정(Label Setting)을 기반으로 제안되었던 노드기반 다수경로알고리즘을 기반으로 링크표지로 전환하는 방안을 개발한다. 또한 알고리즘의 대규모 교통망에 적용한 수행결과를 통해 대안경로정보제공을 위한 현실적용의 문제점을 도출하고, 도로의 연속주행을 보장하려는 운전자의 경로선택행태를 반영하는 방안에 대해 검토한다.

I. 서론

노드와 링크로 구성된 교통망에서 출발지와 도착지를 연결하는 합리적 통행경로는 "경로를 노드 또는 링크의 순서로 표현할 때, 경로를 구성하는 노드의 반복은 존재해도 링크의 반복은 존재하지 않는다"는 원리에 근거한다(Lee, 2004). 이 원리에 의해 도시부 교차로에서 회전지체나 금지 등에 의해 발생하는 복잡한 통행행태를 네트워크의 변형 없이 모사가 가능하다. 최근 첨단교통체계(ITS: Intelligent Transportation System)에 의하여 도시부 운전자/여행자에게 경로정보제공에 대한 활용성이 급격히 증가하므로, ITS에 핵심 알고리즘으로 탑재되어 있는 경로선정 알고리즘에 합리적인 통행패턴이 반영이 요구된다.

최적경로 탐색과정에서 합리적 통행현상을 포함하는 방법은 링크로 구성된 표지(Link-Based Label)를 적용하는 방안이 적용하기 용이하다. 링크표지를 활용하는 경우 링크의 중복표현을 허용되지 않는 상황에서 노드의 중복표현이 가능하여 합리적인 통행을 원칙적으로 보장될 뿐만 아니라 목적지에서 출발지로 경로를 역 추적(Backward Trace)하는 과정에서 단일 전 링크 정보만을 이용하므로 Bellman의 최적원리(Optimality Condition)를 만족함으로 최적해(Optimal Solution)가 보장된다. 링크표지에 근거한 최단경로 알고리즘은 Curby & Potts(1969)에 의해 처음 기본개념이 제안되었는데 네트워크를 확장하지 않고 회전비용의 고려가 가능하다는 장점 때문에 교통모형에 다양하게 활용되고 있다 (Potts & Oliver, 1972; 노정현 & 남궁성, 1995; 김현명 & 임용택, 1999; 이승환 외, 1996; 장인성, 2000; 임강원 & 임용택, 2003).

링크표지를 활용하여 다경로알고리즘 (KSPA: Kth Shortest Path Algorithm)에 적용하는 방안에 대한 연구는 Loopless기반으로 개발된 Yen 알고리즘(1971)을 근거로 기존의 링크기반 최적경로 알고리즘 (SPA: Shortest Path Algorithm)을 활용하여 다수의 경로를 순차적으로 탐색하는 알고리즘이 이미영외(2003)에 의해 제안되었다. 다수의 경로를 동시에 탐색하여 수행시간을 절약하기 위하여 Shier알고리즘(1979)를 기반으로 링크표지를 갱신하는 방안(Link-Based Label Correcting)을 적용하는 KSPA이 이미영 외(2004)에 의해 제안되었다.

본 연구는 다수의 합리적 통행행태를 고려하기 위해 링크표지고정방식(Link-Based Label Setting)을 제

안하는 데 목적이 있다. 기존에 표지고정(Label Setting)을 기반으로 제안되었던 KSPA (Shier,1979)을 기반으로 링크표지로 전환하는 방안을 제시한다. 부차적으로 제안된 알고리즘의 수행결과를 통해 대안경로정보제공을 위한 현실적용의 문제점을 도출하고, 서울시 같은 대도시를 포함하는 네트워크에서 경로정보제공을 위해 활용 가능한 기능향상 방안을 모색하는 방법론의 일환으로 도로의 연속주행을 보장하려는 운전자의 경로선택 행태를 경로선정과정에 반영하는 방안에 대해 검토한다.

II. 이론적 배경

최단경로알고리즘은 노드와 링크표지기법 모두 적용되고 있으나, 다경로알고리즘에서는 노드표지기법만 이용되고 있다. 본 절에서 표지고정기법에 근거한 최단경로와 다경로알고리즘에 대한 수행과정을 살펴본다.

1. 표식

본 연구에서 사용되는 표식은 다음과 같다.

r, s, i, j, k : 노드표지; 특히 r, s 는 각각 출발지와 도착지노드

a, b : 링크표지

π_k^r : 출발지 r 에서 노드 i 까지 k 번째 통행비용

π^r : 출발지 r 에서 노드 i 까지 k 개의 비용으로 구성된 벡터 $\pi^r = \{\pi_1^r, \pi_2^r, \dots, \pi_k^r\}$

π_k^a : 출발지 r 에서 링크 a 의 도착지점까지 k 번째 통행비용

π^a : 출발지 r 에서 링크 a 의 도착지점까지의 k 개의 비용으로 구성된 벡터

$\pi^a = \{\pi_1^a, \pi_2^a, \dots, \pi_k^a\}$

$c_{a(i,j)}$: 링크 $a(i, j)$ 의 통행비용

L : 링크집합 N : 노드집합,

R : 출발지노드집합 S : 도착지노드집합,

A_r : 출발노드가 r 인 링크집합

D_a : 출발링크가 a 인 방향별 회전집합,

Q : 탐색링크집합 M : 탐색노드집합,

B_s : 도착노드가 s 인 링크집합;

2. 표지확정기반 최단경로알고리즘

노드표지고정기법(Node-Based Label Setting)은 출발지에서 가장 비용이 적은 노드를 선정하여 이 노드를 출발노드로 하는 링크의 도착노드까지의 비용을 탐색하는 기법이다. 알고리즘은 노드의 개수만큼만 반복하여 노드까지의 비용을 계산하면 출발지에서 모든 노드까지의 최소비용이 계산된다. 식(1)은 출발지에서 각 노드까지의 최단경로비용이 확정되는 과정을 보여준다. 이 방법은 Dijkstra(1959)에 의해 처음 제안됐으며 현재 가장 활발하게 응용되고 있는 알고리즘이다. 경로 추적과정은 Bellman의 최적원리(1957)에 근거한 전노드 개념을 적용하여 최적해가 보장된다.

$$\pi_i^r = \min_{i \rightarrow j} \{ \pi_j^r + c_{ij}, \pi_i^r \} \quad (1)$$

식(1)은 출발지에서 모든 도착지까지의 최단경로비용을 계산하는 것으로 링크비용탐색과정에서 두개의 노드를 고려하는 방법이다. 반면 링크표지고정기법은 링크를 노드처럼 활용하는 기법이다. 즉, 노 출발지에서 모든 링크까지의 최단경로비용을 계산하기 위하여 2개의 인접된 링크를 노드처럼 활용하는 방안으로 마지막 링크의 최단비용을 계산하는 방법이다. 탐색링크는 출발지에서 가장 최단비용을 갖는 링크를 선정하며 따라서 알고리즘이 종료하기 위해서는 링크개수만큼의 반복횟수를 수행해야 한다. 식(2)에서 출발지 r에서 링크 b의 출발지점까지의 최단경로비용을 계산하기 위해서는 두 개의 인접된 링크 a와 b로 구성된 표지를 이용한다. 이때 노드표지와는 차이점은 두 인접링크에서 발생하는 회전비용(d_{ab})을 추가적인 네트워크의 확장 없이 고려할 수 있다는 것이다. 최적경로추적과정에서 기존의 최적경로알고리즘에서 채택하고 있는 전노드 개념(Dijkstra, 1959)을 링크로 유추된 개념을 적용하므로 Bellman의 최적원리(1957)에 의한 최적해가 보장된다. 이 방법은 Kirby 와 Potts(1969)에 의해 처음 제안됐다.

$$\pi_i^{rb} = \min_{a \rightarrow b} \{ \pi_i^{ra} + c_b + d_{ab}, \pi_i^{rb} \} \quad (2)$$

식(2)의 링크표지에 근거한 표지확정 최단경로알고리즘은 다음과 같다. 노드표지기법에 비해 다른점은 1)

링크개수 만큼 알고리즘을 반복해야 한다는 것과, 2) 링크표지에 근거해서 최단경로비용을 계산한 것이므로 알고리즘의 수행과정이 종료되면 노드표지에 근거한 비용으로 재환산 해주어야 한다는 것이다.

3. 표지고정기반 다경로알고리즘

다경로알고리즘은 경로비용의 순차적인 순서를 갖는 복수의 경로를 탐색하는 것이다. 기존에 제안된 알고리즘은 크게 2가지 방법으로 분류된다. 하나는 기존의 표지확정/교정의 방법을 확대 이용하는 것이고, 다른 하나는 경로의 부분삭제를 통한 다른 경로를 발견하는 방법이다. 두 방법의 가장 큰 차이로서, 전자는 K개의 경로를 동시에 탐색하는 방법 (Shier, 1979; Pollack, 1961; Bellman & Kalaba, 1968)을 따르며, 후자는 최단경로알고리즘을 활용하여 탐색된 경로(1부터 K-1까지)를 기반으로 K번째 경로를 탐색하는 방법 (Martins, 1984; Azevedo et al, 1993; Yen, 1971)을 따른다는 것이다. 부분삭제방법은 최단경로알고리즘을 그대로 활용하므로 해를 비교적 쉽게 산정할 수 있다는 장점이 있으나, 네트워크가 커지면 부분삭제의 링크개수가 많은 조합으로 나타나므로 컴퓨터 수행시간에 불리하다는 단점이 있다. 반면 표지확정/교정방법은 알고리즘의 해법이 복잡하나 적절한 k수에 대해 대규모 네트워크에 적용이 가능할 정도로 알고리즘 수행시간이 짧다는데 장점이 있다.

노드기반 다경로알고리즘과 최적경로알고리즘의 차이는 최적경로알고리즘은 노드개수만큼 반복해서 알고리즘 수행되나, 1) 다경로알고리즘의 경우에는 각 노드 k개의 비용에 변화가 없을 때 까지 수행되어야 하므로 알고리즘의 반복회수가 네트워크의 특성과 k의 값 등에 따라 변화할 수 있다는 것과, 2) 다음탐색노드가 결정되는 과정에서 1번부터 k번까지의 비용이 비교되는 과정에서 순번이 높은 비용이 최소가 되는 탐색노드를 결정한다는 것이다. Shier (1979)에 의해 제안된 노드기반 표지확정 다수경로 탐색알고리즘은 다음과 같다.

Step 1 : 초기화

$$\begin{aligned} \pi^r &= \{ \infty, \infty, \dots, \infty \}, \forall i \in N, r \in R; \\ \pi^j &= \{ c_{ij}, \infty, \dots, \infty \}, \forall (r, j) \in A, r \in R; \\ M &= M \cup \{ j \}; \end{aligned}$$

Step 2 : 다음탐색노드결정

if $M = \phi$: 종료.
 Else: $i = \underset{i \in j}{\text{arg min}} [\pi_p^{ri} < \pi_p^{rj}]$,
 $i, j \in M, p = 1, 2, \dots, k$;
 $M = M - \{i\}, i \in M$;

Step 3 : 다음노드($i \rightarrow j$)로 확장

If: $\pi_p^{ri} + c_{ij} < \pi_q^{rj}$; $\pi_q^{rj} = \pi_p^{ri} + c_{ij}, \forall j \in A_i$,
 $p, q = 1, 2, \dots, k$;
 $M = M \cup \{j\}$;
 Go to Step 2;
 Else : Go To Step 2;

Step 1 : 초기화

$\pi^{ra} = \{\infty, \infty, \dots, \infty\}, \forall a \in L, r \in R$;
 $\pi^{ra} = \{c_a, \infty, \dots, \infty\}, \forall a(r, j) \in A_r, r \in R$;
 $Q = Q \cup \{a\}, \forall a(r, j) \in A_r$;

Step 2 : 다음탐색링크결정

if: $Q = \phi$: $\pi_q^{ri} = \min\{\pi_i^{ra}\}$,
 $a \in B_i, \{\pi_i^{ra}\} = \pi^r - \{\pi_i^{ra}, \pi_i^{rb}, \dots, \pi_i^{rk}\}$,
 $p = q - 1$; 종료.
 Else: $a = \underset{a \in b}{\text{arg min}} [\pi_p^{ra} < \pi_p^{rb}]$,
 $a, b \in Q, p = 1, 2, \dots, k$;
 $Q = Q - \{a\}, a \in Q$;

Step 3 : 다음링크($a \rightarrow b$)로 확장

If: $\pi_p^{ra} + d_{ab} + c_b < \pi_q^{rb}$; $\pi_q^{rb} = \pi_p^{ra} + d_{ab} + c_b$,
 $\forall b \in A_a, p, q = 1, 2, \dots, k$;
 $Q = Q \cup \{b\}$;
 Go to Step 2;
 Else : Go To Step 2;

III. 링크표지확정 다경로알고리즘

링크표지 기반 최적경로가 갖는 장점인 합리적 통행 행태를 고려할 수 있고, 다경로알고리즘은 다수의 대안 경로의 제공이 가능하다는데 있다. 따라서 본 연구에서 제안하는 알고리즘은 노드기반의 Shier(1979) 알고리즘을 링크표지로 전환 함으로서 위의 두 가지 장점을 포함하여 합리적 통행행태가 반영된 다수의 경로를 제공이 가능하다.

1. 링크표지확정 다경로알고리즘

본 연구에서 제안하는 링크표지확정 다경로알고리즘의 수행과정은 아래에 나타나 있다. 링크기반 알고리즘이 위에서 위의 2절에서 설명한 기존 알고리즘과 다른 점은 1) 각 링크의 k개의 비용에 변화가 없을 때 까지 수행되어야 함으로 알고리즘의 반복회수가 네트워크의 특성과 k의 값 등에 따라 변화할 수 있다는 것과 (Step 3), 2) 다음탐색링크가 결정되는 과정에서 1번 부터 k번까지의 비용이 비교되는 과정에서 순번이 높은 비용이 최소가 되는 탐색링크를 결정한다는 것과 (Step 2), 3) 링크표지에 근거해서 최단경로비용을 계산한 것이므로 알고리즘의 수행과정이 종료되면 노드표지에 근거한 비용으로 재환산 해주어야 한다는 것이다 (Step 2).

IV. 사례연구

본 절에서는 3가지의 각각 다른 사례연구를 통해 알고리즘의 수행과정 및 특성을 파악한다. Visual Studio 6.0에서 제공하는 C++언어와 MFC(Microsoft Foundation Class)에서 제공되는 자료구조를 활용하여 작성되었다.

1. 알고리즘수행과정

간단하게 표현된 네트워크를 통해 알고리즘의 수행 과정을 추적한다. <그림 1>은 적용된 네트워크를 나타내는데 10개의 링크와 5개의 노드로 구성되어 있다. 링크상에는 거리(비용)가 표현되어 있고 노드의 사각형은 링크를 중심으로 전방향 회전점수(turn penalty)를 나타낸다. U-방향회전(U-Turn)은 방향화살표가 존재하는 방향의 2 → 3 → 2와 3 → 2 → 3 두 지점만 가능하다. 출발지와 도착지는 노드1과 5이며 출발지1에서 도착지까지 5개(K=5)의 경로를 탐색한다.

〈표 1〉 기 종점간 탐색경로 알고리즘 수행(K=5)

반복	현재 탐색링크	다음 탐색링크	K번째 비용	1부터 5까지의 경로(노드순서)
0	초기화	(1→2) (1→3)	(1.0, ∞, ∞, ∞, ∞) (3.0, ∞, ∞, ∞, ∞)	
1	(1→2)	(1→3) (2→3) (2→4) (2→5)	(3.0, ∞, ∞, ∞, ∞) (9.0, ∞, ∞, ∞, ∞) (12.0, ∞, ∞, ∞, ∞) (7.0, ∞, ∞, ∞, ∞)	[1]
2	(1→3)	(2→3) (2→4) (2→5) (3→2) (3→4)	(9.0, ∞, ∞, ∞, ∞) (12.0, ∞, ∞, ∞, ∞) (7.0, ∞, ∞, ∞, ∞) (12.0, ∞, ∞, ∞, ∞) (14.0, ∞, ∞, ∞, ∞)	[1]
3	(2→5)	(2→3) (2→4) (3→2) (3→4)	(9.0, ∞, ∞, ∞, ∞) (12.0, ∞, ∞, ∞, ∞) (12.0, ∞, ∞, ∞, ∞) (14.0, ∞, ∞, ∞, ∞)	[1]
4	(2→3)	(2→4) (3→2) (3→4)	(12.0, ∞, ∞, ∞, ∞) (12.0, 14.0, ∞, ∞, ∞) (14.0, 20.0, ∞, ∞, ∞)	[1]
5	(3→2)	(2→4) (3→4) (2→3) (2→5)	(12.0, 23.0, 25.0, ∞, ∞) (14.0, 20.0, ∞, ∞, ∞) (9.0, 18.0, ∞, ∞, ∞) (7.0, 18.0, 20.0, ∞, ∞)	[1], [2], [4]
6	(2→5)	(2→4) (3→4) (2→3)	(12.0, 23.0, 25.0, ∞, ∞) (14.0, 20.0, ∞, ∞, ∞) (9.0, 18.0, ∞, ∞, ∞)	[1], [2], [4]
7	(2→3)	(2→4) (3→4)	(12.0, 23.0, 25.0, ∞, ∞) (14.0, 20.0, 29.0, ∞, ∞)	[1], [2], [4]
8	(2→4)	(3→4) (4→3) (4→5)	(14.0, 20.0, 29.0, ∞, ∞) (20.0, 31.0, 33.0, ∞, ∞) (18.0, 29.0, 31.0, ∞, ∞)	[1], [2], [3], [4]
9	(3→4)	(4→3) (4→5) (4→2)	(20.0, 31.0, 33.0, ∞, ∞) (18.0, 20.0, 26.0, 29.0, 31.0) (20.0, 26.0, 35.0, ∞, ∞)	[1], [2], [3], [4], [5]
10	(4→5)	(4→3) (4→2)	(20.0, 31.0, 33.0, ∞, ∞) (20.0, 26.0, 35.0, ∞, ∞)	[1], [2], [3], [4], [5]
11	(4→2)	(4→3) (2→3) (2→5)	(20.0, 31.0, 33.0, ∞, ∞) (9.0, 18.0, 28.0, ∞, ∞) (7.0, 18.0, 20.0, 26.0, 32.0)	[1], [2], [3], [4], [5]
12	(2→5)	(4→3) (2→3)	(20.0, 31.0, 33.0, ∞, ∞) (9.0, 18.0, 28.0, ∞, ∞)	[1], [2], [3], [4], [5]
13	(2→3)	(4→3) (3→2)	(20.0, 31.0, 33.0, ∞, ∞) (12.0, 14.0, 33.0, ∞, ∞)	[1], [2], [3], [4], [5]
14	(3→2)	(4→3) (2→4)	(20.0, 31.0, 33.0, ∞, ∞) (12.0, 23.0, 25.0, 44.0, ∞)	[1], [2], [3], [4], [5]
15	(2→4)	(4→3)	(20.0, 31.0, 33.0, ∞, ∞)	[1], [2], [3], [4], [5]
16	(4→3)	(3→2)	(12.0, 14.0, 29.0, 33.0, ∞)	[1], [2], [3], [4], [5]
17	(3→2)	(2→3) (2→4)	(9.0, 18.0, 28.0, 35.0, ∞) (12.0, 23.0, 25.0, 44.0, ∞)	[1], [2], [3], [4], [5]
18	(2→3)	(2→4) (3→4)	(12.0, 23.0, 25.0, 44.0, ∞) (14.0, 20.0, 29.0, 46.0, ∞)	[1], [2], [3], [4], [5]
19	(2→4)	(3→4)	(14.0, 20.0, 29.0, 46.0, ∞)	[1], [2], [3], [4], [5]
20	(3→4)			[1], [2], [3], [4], [5]

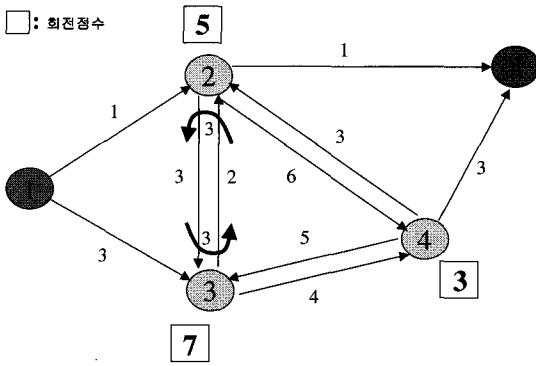
[K=1] (비용: 7.0) : 1→2→5,

[K=2] (비용: 18.0) : 1→3→2→5

[K=3] (비용: 18.0) : 1→2→4→5

[K=4] (비용: 20.0) : 1→2→3→2→5

[K=5] (비용: 20.0) : 1→3→4→5



〈그림 1〉 적용된 네트워크

〈표 1〉은 링크표지로 적용된 다경로알고리즘을 적용하여 출발지1에서 모든 노드까지 5개의 탐색하는 과정을 보여주고 있다. 현재링크에서 다음탐색링크로 진행되면서 각 링크까지의 비용이 갱신되는 과정을 나타내고 있다. 이 과정에서 노드5에서 노드1까지의 5개의 경로와 비용을 탐색하는 과정을 보여주고 있다. 총 20회의 반복수행을 통하여 모든 링크까지의 K개의 비용이 더 이상 갱신되지 않았으며, 현재탐색링크를 선정하는 과정의 예를 반복횟수 10에서 보면, 두 링크(4→3, 4→2)의 첫번째(k=1) 비용이 같을 경우 두 번째(k=2) 비용을 비교하여 최소비용을 갖는 링크가 현재 탐색링크가 된다.

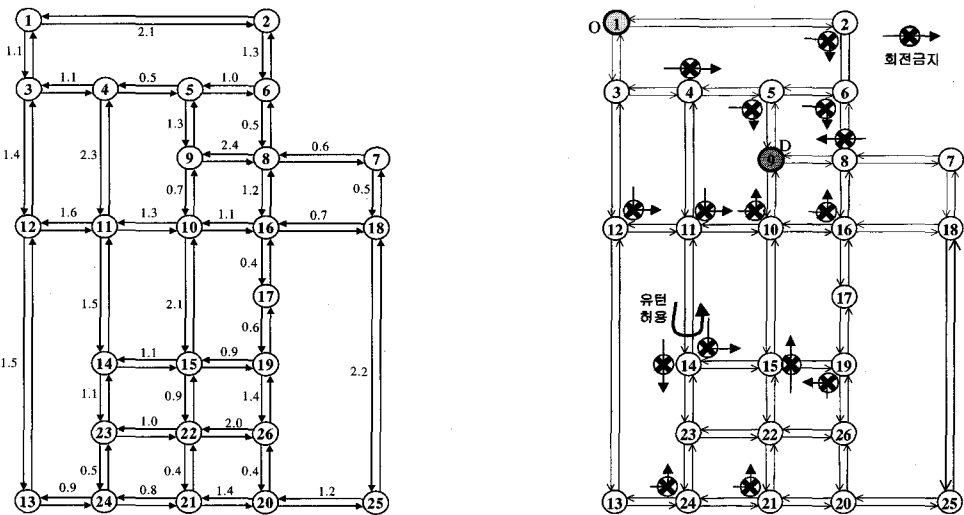
2. 회전금지 및 U-Turn

도시가로망의 교차로에서 회전금지와 U-Turn이 통

행의 효율성을 증진시킬 목적으로 규제방법으로 활용된다. 본 사례연구에서는 이러한 통행규제를 포함하는 네트워크에서의 알고리즘 활용성을 점검한다. 〈그림 2〉은 수정된 Sioux Fall Network (26노드, 76링크)를 대상으로 회전금지 15방향에서 존재하며, U-Turn은 한 방향(11→14→11)을 제외하고 전방향에서 금지된다.

〈표 2〉는 출발지1에서 도착지 9까지의 통행규제를 고려하여 20개의 경로를 탐색한 결과이다. 결과에서 나타나듯이 K개의 경로를 탐색하는 과정에서 순차적으로 비용이 증가되며, 탐색된 경로에서 유턴이나 피턴등 합리적인 도시가로망 통행행태가 반영된다.

〈표 2〉는 출발지에서 도착지노드까지의 노드비용을 표현하고 있다. 본 연구에서 제안된 알고리즘은 링크표지로 제안된 것이므로 각 링크까지 발견된 비용을 노드비용으로 재 추정하는 과정이 필요하다. 〈표 3〉~〈표 5〉는 목적지 노드9까지 K(20)개의 순차적인 비용을 탐색하기 위해서는 노드9를 도착노드로 하는 3개의 링크의 K(20)개의 비용이 고려되고 있음을 보여준다. 3개의 링크비용을 통해 탐색된 59개의 비용중에서 상위 K개의 링크비용이 노드비용으로 환산되는 과정을 나타낸 것이다. 목적지 노드9까지 20개의 순차적인 노드비용과 경로를 추적하기 위하여, 노드9가 도착노드로 하는 각각의 링크 (5→9), (8→9), (10→9)의 비용이 우선 탐색되는 것을 나타낸다. 알고리즘의 이러한 추가적인 작업은 링크표지기반의 알고리즘이 노드표지기반에 비하여 메모리와 수행속도면에서 불리해지는 원인이 된다.



〈그림 2〉 교통망과 통행규제 (U-Turn/회전금지)

〈표 2〉 통행규제가 고려된 경로탐색 (K=20)

순서	비용	노드순서에 의한 경로표시
1	11.70	9←10←16←17←19←26←20←21←24←13←12←3←1
2	13.00	9←10←16←18←25←20←21←24←13←12←3←1
3	13.50	9←8←16←17←19←26←20←21←24←13←12←3←1
4	13.90	9←5←6←8←16←17←19←26←20←21←24←13←12←3←1
5	14.10	9←10←16←17←19←15←22←26←20←21←24←13←12←3←1
6	14.40	9←5←6←8←7←18←25←20←21←24←13←12←3←1
7	14.50	9←5←6←8←7←18←16←10←11←14←11←4←3←1*
8	14.50	9←5←6←8←7←18←16←17←19←26←20←21←24←13←12←3←1
9	14.60	9←10←15←19←17←16←10←11←14←11←4←3←1*
10	14.60	9←10←16←17←19←15←10←11←14←11←4←3←1*
11	14.60	9←10←16←8←7←18←25←20←21←24←13←12←3←1
12	14.70	9←10←16←8←7←18←16←10←11←14←11←4←3←1*
13	14.70	9←10←16←8←7←18←16←17←19←26←20←21←24←13←12←3←1*
14	14.70	9←10←16←18←7←8←16←17←19←26←20←21←24←13←12←3←1*
15	14.80	9←8←16←18←25←20←21←24←13←12←3←1
16	15.20	9←5←6←8←16←18←25←20←21←24←13←12←3←1
17	15.30	9←10←16←17←19←26←20←21←24←13←12←11←4←3←1
18	15.50	9←10←15←14←23←22←26←20←21←24←13←12←3←1
19	15.90	9←8←16←17←19←15←22←26←20←21←24←13←12←3←1
20	15.90	9←10←15←19←17←16←18←25←20←21←24←13←12←3←1

* 당굴망포함

〈표 3〉 링크(5→9)에서 포함된 경로

순서	비용	노드순서에 의한 경로표시
1	13.90	9←5←6←8←16←17←19←26←20←21←24←13←12←3←1
2	14.40	9←5←6←8←7←18←25←20←21←24←13←12←3←1
3	14.50	9←5←6←8←7←18←16←10←11←14←11←4←3←1
4	14.50	9←5←6←8←7←18←16←17←19←26←20←21←24←13←12←3←1
5	15.20	9←5←6←8←16←18←25←20←21←24←13←12←3←1

〈표 4〉 링크(8→9)에서 포함된 경로

순서	비용	노드순서에 의한 경로표시
1	13.50	9←8←16←17←19←26←20←21←24←13←12←3←1
2	14.80	9←8←16←18←25←20←21←24←13←12←3←1
3	15.90	9←8←16←17←19←15←22←26←20←21←24←13←12←3←1

〈표 5〉 링크(10→9)에서 포함된 경로

순서	비용	노드순서에 의한 경로표시
1	11.70	9←10←16←17←19←26←20←21←24←13←12←3←1
2	13.00	9←10←16←18←25←20←21←24←13←12←3←1
3	14.10	9←10←16←17←19←15←22←26←20←21←24←13←12←3←1
4	14.60	9←10←15←19←17←16←10←11←14←11←4←3←1
5	14.60	9←10←16←17←19←15←10←11←14←11←4←3←1
6	14.60	9←10←16←8←7←18←25←20←21←24←13←12←3←1
7	14.70	9←10←16←8←7←18←16←10←11←14←11←4←3←1
8	14.70	9←10←16←8←7←18←16←17←19←26←20←21←24←13←12←3←1
9	14.70	9←10←16←18←7←8←16←17←19←26←20←21←24←13←12←3←1
10	15.30	9←10←16←17←19←26←20←21←24←13←12←11←4←3←1
11	15.50	9←10←15←14←23←22←26←20←21←24←13←12←3←1
12	15.90	9←10←15←19←17←16←18←25←20←21←24←13←12←3←1

3. 대안경로

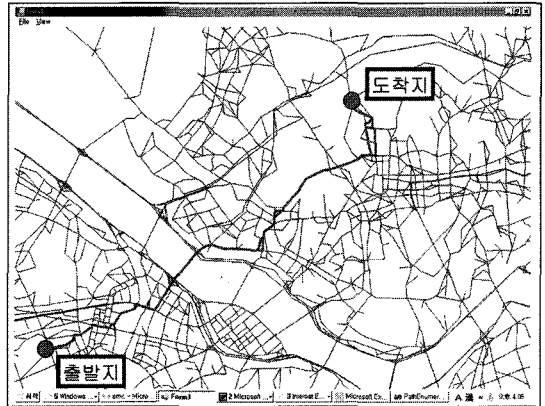
대안경로 탐색을 위해서 적용된 네트워크는 <그림 3>과 같이 우리나라의 내륙지역 전체를 대상으로 하며, 37944개의 링크와, 15680개의 노드로 구성되어 있다. 출발지와 도착지는 서울시의 임의의 두 지점으로, 출발지는 서울시의 양천구 신월동의 주거지역의 한 지점과 도착지는 종로구 부암동 도심의 한 지점이다.

1) 대안경로를 위한 활용

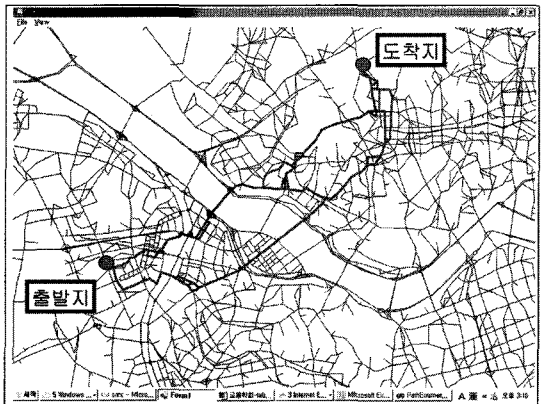
운전자에게 제공되는 경로정보의 대안은 운전자가 비교적 용이하게 경로의 특성을 이해하고 선택하도록 구성되어야 한다. 이를 위해 일반적으로 경로대안이 갖추어야 할 두 가지 조건은: 1) 운전자는 기 종점간 거리가 멀수록 운전자는 위계가 높은 도로를 이용하려는 특성과, 2) 명확한 특성을 갖는 한정된 수개의 대안을 제공하여 경로의 선택이 쉽게 이루어지도록 되어야 한다. 알고리즘을 활용하여 이러한 요건을 만족하기는 일반적으로 용이하지 않다. 네트워크의 특성에 따라 차이가 있을 수 있으나, 기존의 알고리즘을 활용하여 탐색되는 경로는 일반적으로 경로간 유사성이 크게 나타난다(Park,1998). <그림 4>은 기종점간 10개 경로의 궤적을 네트워크상에 표현한 것이다. 전체적인 10개 경로의 특성은 거의 유사한 것으로 파악되고 있다.

<그림 5>은 500개의 경로궤적을 네트워크상에 나타낸 것이다. <그림 5>에서 나타난 10개의 궤적보다는 다양한 대안을 제시하는 것으로 나타나고 있다. 그러나, 교량을 중심으로 문제점을 파악하여 보면, 좌측으로 성산대교와 우측의 서강대교만이 이용되었을 뿐 두

교량의 중간지점인 서강대교는 궤적에 나타나지 않아, 결과가 대안으로서 충분하지 않은 정보를 나타내고 있다고 판단된다. 만약 중간지점에 교량이 없다고 한다면 <그림 4>와 유사한 형태인 경로간의 유사성이 매우 높을 것으로 판단된다.

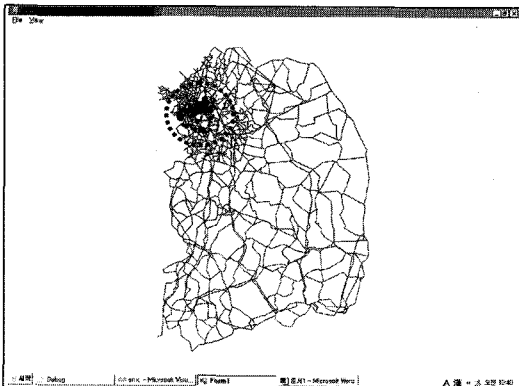


<그림 4> 기종점간 알고리즘으로 탐색된 10개 경로

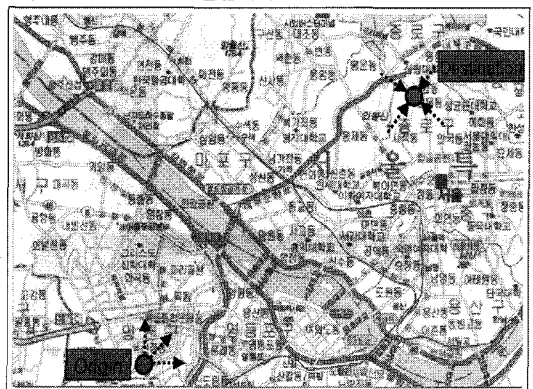


<그림 5> 기종점간 알고리즘으로 탐색된 500개 경로

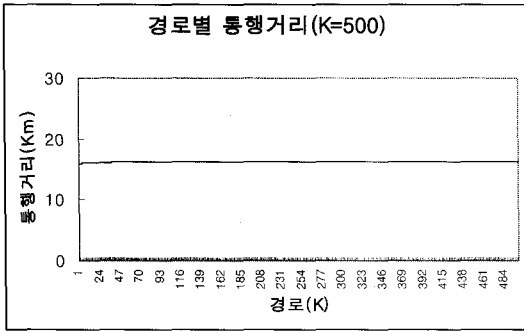
적용 네트워크



출발지와 도착지



<그림 3> 적용된 네트워크와 기종점



〈그림 6〉 500개 경로의 통행거리변화

이는 500개의 대안을 생성하는 과정에도 예상된 서강대교가 포함되지 않을 정도로 알고리즘의 민감도가 작다는데 문제가 있다. 〈그림 6〉는 500개의 경로에 대한 통행거리의 민감도를 그래프로 나타낸 것으로, 경로의 순차적인 증가에도 불구하고 각 경로의 통행거리의 변화가 없을 정도로 매우 작은 민감도로 변화하는 것으로 나타났다.

2) 도로의 연속주행 인지비용 고려방안 검토

운전자에게 제공되는 경로정보의 대안은 운전자가 비교적 용이하게 경로의 특성을 이해하고 선택하도록 구성되어야 한다. 이를 위해 일반적으로 경로대안이 갖추어야 할 우선 조건은 경로를 구성하는 구간의 운행동질성 또는 편의성을 가능한 한 확보하는 소수의 경로대안을 탐색하는 것이다. 지금까지의 관련연구로서는 중복벌점(Overlapping Penalty; Park, 1998)과 경로변경인지비용(Lee, 2003)을 활용하는 방안이 가능하나, 본 연구에는 Lee(2003)가 제시한 경로변경인지비용을 통해 제시된 알고리즘의 활용가능성을 검토한다.

식(3)은 본 연구에서 적용하고 있는 경로변경인지비용이 포함된 총 비용을 나타낸다. 총 비용은 총 통행거리와 경로주행 인지비용으로 구성된다. 식(4)은 도로위계가 낮은 경로로 주행전환을 시도하는 경우 도로의 위계차이와 β_{ab} 의 곱만큼 증가하게 되는 과정을 보여주고 있으며, 도로위계가 높은 경로로 주행하는 경우에는 경로인지비용이 없는 것으로 산정된다. 예를 들면, 고속도로의 도로등급은 1, 주 간선도로 2, 보조간선도로 3, 집 분산도로 4라고 한다면, 고속도로로 주행하는 운전자는 목적지 근처에 도달할 때 까지 가능한 한 고속도로를 연속으로 주행하는 속성을 유지하도록 하며, 목

적지 근처에서도 도로위계가 급격하게 변화되는 주행에 대해서는 운전자가 비용으로서 인지하여 도로위계의 변화가 적은 방향에서 접근하는 속성을 갖도록 구성되었다.

$$TC_k^{rs} = LC_k^{rs} + PC_k^{rs} \tag{3}$$

여기서 TC_k^{rs} : 출발지r과 도착지s를 연결하는 k번째 경로의 총비용

LC_k^{rs} : 출발지r과 도착지s를 연결하는 k번째 경로의 총통행거리 (Km)

PC_k^{rs} : 출발지r과 도착지s를 연결하는 k번째 경로의 경로변경인지비용

$$PC_k^{rs} = \sum_a \sum_b \{ \beta_{ab} \cdot \delta_{ab} \} \cdot \phi_{abk}^{rs} \tag{4}$$

여기서 β_{ab} : 링크(a)에서 링크(b)로 전환할 때 인지비용(상수)

δ_{ab} : 링크(b)와 링크(a)의 도로등급위계 (Rank)의 차이

$$\delta_{ab} > 0 \text{ if } (Rank(b) - Rank(a) > 0);$$

$$\delta_{ab} = 0 \text{ else if } (Rank(b) - Rank(a) \leq 0);$$

ϕ_{abk}^{rs} : 링크(a)에서 링크(b)로 회전방향이 출발지r과 도착지s를 연결하는 k번째 경로에 포함되는지 여부를 결정(1: 포함, 0:포함되지 않음)

사례연구는 선정된(A,B,C) 3개의 시나리오를 수행을 통해 검토된다. 현황과 인지비용(β_{ab})의 민감도로 구분한다. 〈표 6〉은 각 시나리오의 경로비용의 특성을 나타낸 것이다. 시나리오A의 통행비용은 출발지에서

〈표 6〉 적용된 시나리오의 특성

시나리오	경로비용의 특성	비고
A	통행비용	〈그림 4〉
B	통행비용+인지비용
C	통행비용+인지비용

* 도로위계는 4등급으로 구분: 1, 2, 3, 4
 1: 고속도로, 도시고속화도로;
 2: 간선도로;
 3: 보조간선도로;
 4: 집분산도로

목적지까지의 통행거리를 비용으로 환산한 것으로 3.1 절의 <그림 4>의 결과와 같다.

먼저 알고리즘에 의한 10개의 경로에 대한 수행결과를 통행거리와 경로간의 유사성에 대한 분석은 <그림 7~8>와 같다. <그림 7>은 인지비용을 고려하는 경우에 보다 통행거리가 길어지는 것으로 나타났다. 이는 연속주행이 보장되는 경로정보를 제공하기 위해서는 위해서는 경로의 통행거리 이외에 추가적인 인지비용이 고려되어야 함을 나타내고 있다. 식(5)에 근거한 경로간 중복도를 분석한 <그림 8>을 통해 인지비용을 고려한 시나리오인 B와 C를 통해 명확하게 인지비용의 고려만으로는 대안 간의 명확한 구분을 나타내기가 용이하지 않음을 보여준다.

$$V_{ijk} = \frac{OL_{ijk}}{L_k} * 100 \quad (5)$$

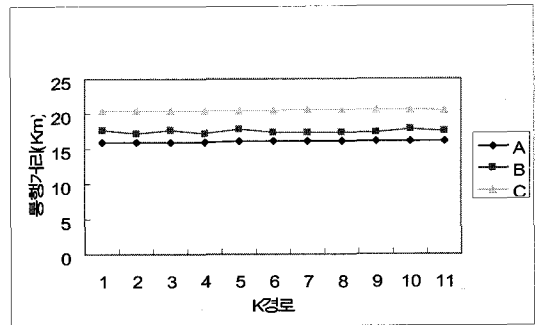
여기서 V_{ijk} : 최적경로(k)에 대한 경로(i)의 중복도(%)

OL_{ijk} : 최적경로(k)와 중복되는 경로(i)의 통행거리(Km)

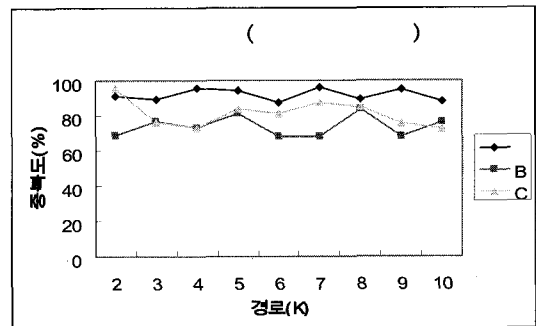
L_k : 최적경로(k)의 통행거리(Km)

<그림 9,10>은 <그림 7,8>의 결과를 경로계획을 통하여 확인하고자 하는 것이다. <그림 9>은 연속주행에 대한 인지비용의 영향이 통행거리비용에 비하여 그다지 크지 않을 경우($\beta_{ab} = 0.5$) 도로위계가 잘 정립되어 급격한 경로전환이 이루어지지 않는 종로부근에서 경로의 전환에 따른 인지비용이 발생하며, 위계가 커질 가능성이 있는 양천구 지역은 인지비용이 최소화 가능한 단일 경로를 통해 주행하는 것으로 파악되었다.

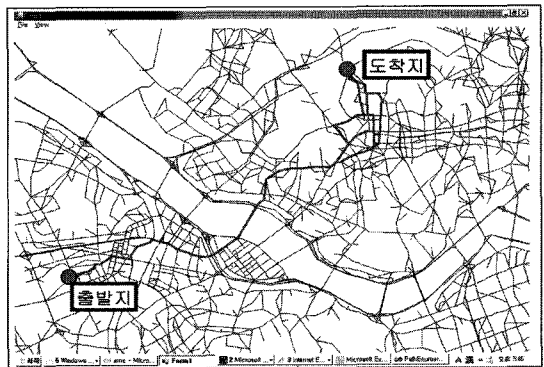
<그림 10>은 연속주행의 인지비용의 영향이 통행거리 비용에 비하여 큰 영향을 나타내는 것이다($\beta_{ab} = 10.0$). <그림 9>와 비교하여 경로전환가능성이 작아지도록 종로접근방향에서 경로간의 전환이 적게 발생되도록 경로 접근이 가능한 가로망 지역이 축소되었다. 또한 주행경로의 동질성이 유지되도록 주요 접근축이 시나리오B의 서강대교에서 마포대교로 이동되었으며, 출발지인 양천구 지역에서 경로의 우회가 이루어져, 통행거리가 증가해도 도로위계가 정립된 연속주행을 선택하는 현상을 보여주고 있다.



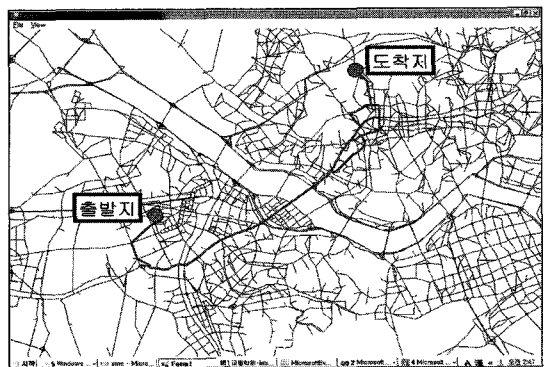
<그림 7> 시나리오의 경로별(K=10) 통행거리



<그림 8> 시나리오간 최단경로(K=1)와 비교된 경로중복도



<그림 9> 시나리오 B의 경로계획 (K=10)



<그림 10> 시나리오 C의 경로계획 (K=10)

V. 결론

합리적인 통행이란 “경로를 노드 또는 링크의 일련의 순서로 표현할 때, 경로를 구성하는 노드의 반복은 허용해도 링크의 반복은 허용하지 않는다”는 것이다. 최적경로 탐색과정에서 링크로 구성된 표지를 적용하는 경우 이러한 합리적 통행현상을 알고리즘에 포함 가능 하다. 본 연구는 기존에 표지고정(Label Setting)을 기반으로 제안되었던 다수경로 탐색알고리즘을 기반으로 링크표지로 전환하는 방안을 제시했다. 알고리즘의 회전지체 및 통행금지등 도시가로망 특성이 반영된 모의 네트워크를 통해 알고리즘의 기본특성을 파악하였다. 합리적인 대안경로제공의 역할을 수행하기 위한 가능성을 타진하기 위해 그 동안 제시되었던 도로위계에 따른 인지비용을 대안경로선정과정에 미치는 영향을 시나리오로 구성하여 파악하였다. 결론적으로 제안된 알고리즘은 최적경로알고리즘위주로 개발되었던 링크표지 기반 알고리즘이 갖는 장점을 수용하는 것으로 파악되었고, 연속주행에 대한 인지비용의 추가고려를 통해 대안경로선정을 위한 실용적인 개발방향으로 한단계 진전되었다.

추후연구과제로서 운전자에게 합리적인 대안경로정보를 제공하는 방안을 알고리즘상에 포함하기 위해서는 대안간에 유사성에 차이가 존재하는 소수의 경로정보제공 방식에 대한 연구가 필요하다. 기존에 제시된 중복벌점(Overlapping Penalty.)에 대한 활용과 운전자 인지비용을 통합하는 문제와 각각의 방안에서 경로정보 제공에 적합한 파라메타의 산정에 관한 내용 등이 존재한다.

참고문헌

1. 김현명·임용택(1999), 유전 알고리즘을 이용한 전역탐색 최단경로 알고리즘개발, 대한교통학회지, 제17권 제2호, 대한교통학회, pp.163~178.
2. 노정현·남궁성(1995), 도시가로망에 적합한 최단경로탐색기법의 개발, 국토계획, 제30권 제5호, 대한국토도시계획학회, pp.153~168.
3. 이미영·유기윤·김정현·신성일(2003), 덩굴망 통행패턴을 고려한 One-To-One 다경로알고리즘, 대한교통학회지, 제21권 제6호, 대한교통학회, pp.89~99.
4. 이미영·백남철·최대순·신성일(2004), 링크표지개신 다수경로탐색 알고리즘, 대한교통학회지, 제22권 제2호, 대한교통학회, pp.131~143.
5. 이승환·최기주·김원길(1996), 도시부 ATIS 효율적 적용을 위한 탐색영역기법 및 양방향 링크탐색 알고리즘의 구현, 대한교통학회지, 제14권 제3호, 대한교통학회, pp.45~59.
6. 임강원·임용택(2003) 교통망분석론, 서울대학교출판부.
7. 장인성(2000), 서비스시간 제약이 존재하는 도시부 복합교통망을 위한 링크기반의 최단경로탐색 알고리즘, 대한교통학회지, 제18권 제6호, 대한교통학회, pp.111~121.
8. Azevedo J. A., Costa M. E. O. S., Madeira J.J.E.R.S., and Martins E.Q.V. (1993), An Algorithm from the Ranking of Shortest Paths, European Journal of Operational Research, Vol. 69, pp.97~106.
9. Bellman R. (1957), Dynamic Programming, Princeton University Press, Princeton, New Jersey.
10. Bellman R. and Kalaba R. (1968), On Kth Best Policies. J. SIAM 8, pp.582~588.
11. Dijkstra E. W. (1959), A Note of Two Problems in Connected with Graphs. Numerical Mathematics. I, pp.269~271.
12. Kirby R. F. and Potts R. B. (1969), The Minimum Route Problem for Networks with Turn Penalties and Prohibitions. Transportation Research 3, pp.397~408.
13. Lee M. (2004), Transportation Network Models and Algorithms Considering Directional Delay and Prohibition for Intersection Movement, Ph.D. Thesis, University of Wisconsin-Madison.
14. Martins E.Q.V. (1984), An Algorithm for Ranking Paths that May Contain Cycles, European Journal of Operational Research, Vol. 18, pp.123~130.
15. Moore E. F. (1957), The Shortest Path through A Maze. Proc. Int. Conf. on the Theory of Switching. Harvard Univ., Cambridge, MA.
16. Pollack M. (1961), The Kth Best Route

- Through A Network, Operations Research, Vol. 9, pp.578~580.
17. Potts R.B. and Oliver R.M.(1972), Flows in Transportation Networks. Academic Press.
18. Shier R. D. (1979), On Algorithms from Finding the k Shortest Paths in a Network, Networks, Vol. 9, pp.195~214.
19. Yen J.Y. (1971), Finding the K shortest Loopless Paths in a Network, Management Science, Vol.17, pp.711~715.

✻ 주 작 성 자 : 이미영

✻ 논문투고일 : 2003. 11. 29

논문심사일 : 2004. 5. 15 (1차)

2004. 7. 14 (2차)

2004. 7. 29 (3차)

심사판정일 : 2004. 7. 29

✻ 반론접수기한 : 2004. 12. 31