

■ 論 文 ■

일반가로망에서 교통정보제공을 위한 $n-path$ 알고리즘의 개발

Development of a $n-path$ algorithm for providing travel information in general road network

임 용 택

(여수대학교 교통공학과 조교수)

목 차

- | | |
|--------------------------|------------------------|
| I. 서 론 | 2. $n-path$ 알고리즘의 개발 |
| II. 일반 가로망에서 최단경로탐색시 문제점 | IV. 모형의 평가 |
| 1. 기존 최단경로 탐색알고리즘의 문제 | 1. 연속 좌회전이 존재하는 단순 가로망 |
| 2. 안내경로 제공시의 중복문제 | 2. Sioux Falls 가로망 |
| III. $n-path$ 알고리즘의 개발 | V. 결 론 |
| 1. $k-path$ 알고리즘과의 차이 | 참고문헌 |

Key Words : $k-path$ 알고리즘, $n-path$ 알고리즘, 회전제약, 경로중복, 우회경로 안내정보

요 약

교통정보에 의한 교통량분산 효과를 실질적으로 얻기 위해서는 좌회전금지, U-turn, P-turn과 같은 교차로내 회전제약이 존재하는 일반 가로망에서 적정수의 경로를 도출하여 제공해야 한다. 이를 위하여 $k-path$ 알고리즘이 주로 이용되고 있으나 도출된 경로들간에 중복성이 문제가 되고 있다. 본 연구는 교차로내 회전제약들을 고려하면서 교통정보제공을 위한 n 개의 최단경로탐색($n-path$) 알고리즘을 개발하는 데 연구의 목적이 있다. 여기서 $n-path$ 알고리즘은 기존 $k-path$ 알고리즘과는 차이가 있는데, $k-path$ 알고리즘은 기종점간 통행비용을 기초로 첫 번째 최단경로의 2번째 최단경로, 3번째 최단경로,....식으로 k 개의 최단경로를 찾는 데 비해, $n-path$ 알고리즘은 각 경로간 일정수준 이상의 경로중첩(path overlap)이 발생하지 않도록 하면서 n 개의 경로를 탐색하는 방법이다. 이를 위하여 첫 번째 탐색된 경로를 중심으로 통행비용과 경로중복수준을 판단하여 이후 경로들을 탐색하게 된다. 또한, 본 연구에서 제시하는 $n-path$ 알고리즘은 기존 연구와는 달리 교차로상 회전제약을 반영하기 위하여 가로망을 확장할 필요가 없다는 장점이 있다. 개발된 알고리즘을 몇 개의 예제 네트워크에 적용하여 평가하였으며, 평가결과 원하는 결과를 도출하고 있음을 확인할 수 있었다

I. 서론

기중점을 연결하는 최단경로 탐색 알고리즘은 통행 배정모형을 구성하는 여러 부프로그램중 핵심을 이루는 부분이다. 정적 통행배정 문제뿐만 아니라, 확률적 통행배정 문제로부터 동적 통행배정 문제나 화물수송 문제 등 정확한 경로의 탐색은 교통 계획단계에서 매우 중요하다. 현재 교통분야에서 이용되는 전통적인 최단 경로 알고리즘은 주로 산업공학분야에서 개발된 수형망(Tree Building) 알고리즘에 그 기초를 두고 있다. 현재까지 최단경로 탐색에서 연구된 주요 문제는 회전 금지나 회전 벌점을 고려할 수 있는 최단경로 알고리즘을 개발하는 분야였다. 수형망 알고리즘의 경우 좌회전 금지, P-turn, U-turn 문제를 최단경로 탐색과정에서 효율적으로 반영하지 못하기 때문에 덩굴망(Vine Building) 알고리즘이 개발되었으나, 덩굴망 알고리즘 역시 일부 경우에 최단경로를 제대로 찾지 못하는 것으로 알려져 있다(김익기, 1998). 이런 문제를 해결하기 위하여 기존 연구들에서는 가로망을 확장(network expansion)하는 방법을 주로 사용하고 있다. 그러나 가로망이 대규모로 커지는 경우, 특히, 실시간으로 교통정보를 제공하는 경우에는 이런 최단경로 탐색법은 사실상 적용이 불가능해진다. 또한, 실질적으로 교통정보를 제공하기 위해서는 회전제약 등 일반 가로망의 교통운영특성을 반영하면서 적당한 수의 최단경로를 제공할 필요가 있다. 이와 관련된 기존 연구로는 k 개의 경로를 제공하는 k -path 알고리즘이 있으나 k -path 알고리즘은 산출된 경로들이 서로 유사하다는 중복문제가 발생하고 있다.

이런 문제들을 완화시키기 위하여 본 연구는 좌회전 금지, U-turn, P-turn과 같은 교차로내 회전제약이 존재하는 일반 가로망에서 교통안내 정보제공을 위한 n -path 알고리즘을 개발하는 데 연구의 목적이 있다. 여기서 n -path란 본 연구에서 새롭게 제시하는 개념으로 n 개의 경로를 의미하며, n -path 알고리즘은 이들 경로를 탐색하는 방법이다. n -path 알고리즘은 일반적으로 교통망분석시 사용하는 k -path와는 차이가 있는데, k -path 알고리즘은 기중점간 통행비용을 기초로 첫 번째 최단경로의 2번째 최단경로, 3번째 최단경로,.....식으로 k 개의 최단경로를 찾는 데 비해, n -path 알고리즘은 일정수준 이상으로 경로중첩이

일어나지 않도록 n 개의 경로를 제시한다. n -path 알고리즘의 개발이 필요한 이유는, 기존 k -path 알고리즘의 경우 탐색된 경로들간 경로의 중복이 필수적으로 발생하기 때문인데, 즉, 첫 번째 최단경로와 두 번째 최단경로는 거의 유사한 경로로 이루어져 있다. 이는 여러 대안도로로 교통량을 분산시키려는 목적으로 제공되는 교통안내 정보전략을 효과적으로 구현할 수 없다는 한계가 있다. 물론 일부 연구에서, 이들 경로중복을 완화시키려는 알고리즘들이 제시되고 있지만, 알고리즘 구조가 복잡할 뿐만 아니라, 근본적으로 경로문제를 해결하기에는 미흡한 측면이 많다. 이에 비해 본 연구에서 제시되는 n -path 알고리즘은 첫 번째 탐색된 최단경로를 중심으로 통행비용과 경로중복 수준(level of path overlap)을 판단하여, 이후 경로들을 탐색하게 된다. 본 연구에서 제시되는 n -path 알고리즘은 가로망을 확장할 필요가 없다는 장점이 있으며, 일반 가로망에서 실질적으로 교통정보를 제공할 수 있는 수준의 경로를 도출하게 된다.

II. 일반가로망에서 최단경로 탐색시의 문제점

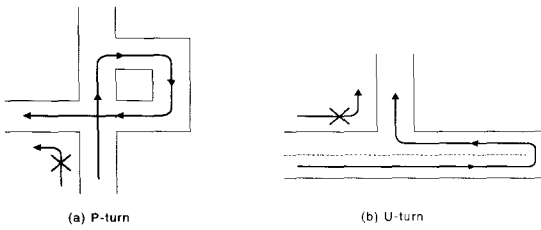
1. 기존 최단경로 탐색 알고리즘의 문제

기존 최단경로 탐색에서 연구된 주요 문제는 회전 금지나 회전 벌점을 고려할 수 있는 최단경로탐색 알고리즘을 개발하는 분야였다. Dijkstra 알고리즘(1959)으로 대표되는 수형망 알고리즘(Tree building algorithm)의 경우, 교차로내 좌회전금지, P-turn, U-turn 허용 등 회전제약문제를 최단경로 탐색과정에 효과적으로 고려할 수 없는데, 이는 최단경로를 탐색할 때 사용하는 기본 원리인 다음과 같은 Bellman의 최적조건을 위배하기 때문이다.

[Bellman의 최적조건] 양의 값인 링크 비용 $l(i, j)$ 을 갖는 교통망 $[N, A]$ 의 기점 h 에서 노드 j 까지의 최단경로 통행비용 $c^*(j)$ 는 다음 함수방정식의 유일해이다(여기서, N 은 노드집합, A 는 링크집합이며 $c(j)$ 는 기점 h 에서 노드 j 까지의 경로통행비용임).

$$c(h) = 0,$$

$$c^*(j) = \min_{j \neq h} [c^*(i) + l(i, j)]$$

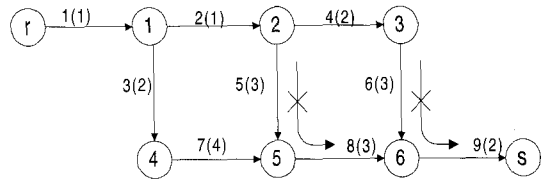


〈그림 1〉 P-턴과 U-턴의 예

위 Bellman의 최적조건은 최단경로로 결정된 경로상의 노드는 다시 중복되어 결정될 수 없는 성질을 갖고 있다. 그런데 실제 교통망에서는 이런 경우가 빈번히 발생하고 있는데, 대표적인 경우가 〈그림 1〉과 같이 P턴이나 U턴이 존재하는 경우이다. 이런 경우 동일한 노드를 2번 중복해서 결정되기 때문에 최적조건에 위배되어 최단경로에서 제외된다.

이런 회전제한문제를 풀기 위하여 덩굴망(Vine building algorithm) 알고리즘이 개발되었다. 덩굴망 알고리즘은 동일한 노드를 한번이상 포함할 수 있기 때문이다. 따라서, 좌회전 금지와 같이 회전별점이 있는 경우에도 최적 경로를 찾을 수 있게 된다. 그런데, 연속적으로 회전이 금지된 교통망의 경우 기존 최단경로 알고리즘으로도 정확한 최단경로를 찾지 못한다는 사실이 밝혀졌다(김익기, 1998; 이승환, 최기주, 김원길, 1996). 김익기(1998)는 〈그림 2〉와 같이 연속해서 좌회전 금지가 있는 교통망의 경우, 덩굴망 알고리즘으로 최단경로를 찾을 수 없음을 보여 주었다. 이런 경우 전통적인 덩굴망 알고리즘은 〈표 1〉과 같이 잘못된 최단경로를 찾게 된다. 그 이유는 최적조건을 만족하는 덩굴망 알고리즘이 가능 탐색영역(feasible searching area)을 제한하기 때문이다. 〈그림 2〉에서 덩굴망 알고리즘은 종점 s에 도착하는 경로가 노드 3과 노드5에서 출발하는 경로로 제한된다. 따라서 노드3과 노드 5 이전에 있는 다른 노드는 고려되지 않아 역추적 가능 탐색노드(backward tracking feasible searching nodes)가 한정되고 따라서 이전에 탐색된 노드(즉, 노드1,2,4)의 표지를 고칠 수(correct) 있는 기회를 갖지 못하기 때문에 잘못된 최단경로를 산출하게 되는 것이다.

위와 같은 문제점들을 해결하기 위하여 주로 사용하는 방법이 가로망을 확장(network expansion)하는 것이다. 즉, 교차로내 회전이 가능한 방향별로 링크를 추가하여 방향별 회전가능을 기하학적으로 도입하는 방법이다. 이 경우 기존의 최단경로 알고리즘을 이용하여



〈그림 2〉 연속회전제한이 있는 교통망 (김익기, 1998)

〈표 1〉 덩굴망 알고리즘에서 계산된 최적경로와 실제 최적경로

	calculated path
path from vine algorithm	r → ① → ② → ③ → ⑥ → s
optimal path	r → ① → ④ → ⑤ → ⑥ → s

위와 같은 문제들을 풀 수 있으나, 가로망을 확장해야 한다는 부담을 안게 된다. 특히, 이를 동적으로 확장할 경우, 이런 부담은 기하급수적으로 커지게 된다.

2. 안내경로 제공시의 중복문제

지능형 교통체계사업(ITS, Intelligent Transportation Systems)의 핵심분야중 하나인 대안도로에 대한 교통정보제공을 실질적으로 구현하기 위해서는 이에 맞는 최단경로의 탐색이 필요하다. 현재까지 주로 연구의 대상이 되고 있는 방법으로는 k개의 경로를 탐색하여 제공하는 방법이다. 그러나 이 경우 제공되는 k개의 경로간에 다수의 링크가 서로 중복되어 실질적인 대안경로로서의 기능이 불확실하다. 이런 문제는 확률적 통행배정문제에서 로짓모형을 적용할 경우 발생하는 경로간 IIA문제(Independence of Irrelevant Alternatives)와 동일하다. 즉, 첫 번째 최단경로와 2번째 그리고 3번째 최단경로가 서로 대부분 중첩되어 우회경로로서의 역할을 기대할 수 없게 되는 것이다. 따라서 현실적인 대안도로로서 기능을 발휘하기 위해서는 경로의 중첩정도를 고려해서 교통정보를 제공하는 전략이 필요하다. 이는 통행비용을 기준으로 k개의 경로를 제공하기보다는, 통행비용과 최단경로와의 중첩정도를 함께 고려하여 대안경로를 제공하는 것이 ITS사업의 교통정보전략에 부합한다고 볼 수 있다. 이를 통해서만이 우리가 기대하는 교통정보로 인한 교통량의 공간적인 분산을 실질적으로 유도할 수 있기 때문이다.

III. n-path 알고리즘의 개발

본 연구에서는 앞에서 살펴본 문제들을 해결하기 위

하여 $n-path$ 알고리즘을 제시한다. 먼저, 기존 $k-path$ 알고리즘과의 차이를 좀 더 살펴보면 다음과 같다.

1. $k-path$ 알고리즘과의 차이

본 연구에서 개발되는 $n-path$ 알고리즘과 기존 $k-path$ 알고리즘과의 차이를 설명하기 위하여 $k-path$ 알고리즘에 대하여 간단히 기술하면 다음과 같다. k 최단경로 알고리즘(k -shortest path algorithm)은 최단 경로 이외의 $k-1$ 개의 경로들을 발생시켜 저장하는 알고리즘이다. 이러한 연구는 최단경로 알고리즘 개발의 일환으로 진행되었는데 최근 들어서는 합리적인 통행배정, 혼잡시 우회도로안내, 복수 속성을 갖는 최적경로탐색 등 여러 분야에서 사용되고 있다. $k-path$ 알고리즘은 크게 두가지로 나누어 지는데, 기존의 표지확정/갱신 방법을 확대하여 k 개의 경로를 동시에 탐색하는 방법(Shier,1979)과 처음 도출된 최단경로에서 특정 노드를 이탈시켜 순차적으로 $k-1$ 개의 경로를 찾는 방법이다(Yen,1971). 이중 Yen의 방법은 기존 최단경로 알고리즘을 그대로 활용하면서 쉽게 해를 구할 수 있는 탐색법으로, 기본 개념은 일단 최단경로가 탐색되면 경로 상에서 이 최단경로를 제외하고 가장 비용이 낮은 경로를 탐색하면서 두 번째, 세 번째 등 비용이 낮은 경로들을 결정하는 방법이다. 즉, 최단경로 A^1 이 결정되면, 두 번째로 비용이 낮은 경로 A^2 는 A^1 과 대부분의 경로를 공유하게 되기 때문에 최단경로에 포함된 경로들 중 일부의 노드를 A^1 에 속하지 않은 노드들로 대체하여 여러 개의 경로를 발생시켜 충분한 수가 발생되면 이들의 비용을 비교해서 A^2, A^3, \dots 등 k 개의 최단경로를 구하는 방법이다.

그러나, 실제 k 최단경로 알고리즘을 통하여 구해진 경로들은 대부분 직접 적용이 어려운데, 이는 도출된 대부분의 경로들이 동일한 링크들을 공유하기 때문이다. 따라서, 산출된 경로들의 유사성을 파악하기 위해 여러 방법들이 사용왔으며, 가장 흔히 사용할 수 있는 방법은 두 경로의 전체 길이중 어느정도 비율이 공통되는가를 살펴보는 방법이다. 단순히 두 경로의 공통 링크의 길이를 더하는 방법을 사용할 수도 있고

<표 2> $k-path$ 알고리즘과 $n-path$ 알고리즘과의 차이

	$k-path$ 알고리즘	$n-path$ 알고리즘
경로선택 기준	통행비용	통행비용+경로중복 수준
입력자료	경로수(k)	경로 중복수준(Op)
산출되는 경로수	k 개 (사전에 설정)	- 사전에 경로수 미설정 - 결과로서 n 개의 경로가 도출

Cascetta et al.(1996)이 제시한 Commonality factor(CF_k)를 이용해 좀더 수치적인 분석을 시행할 수도 있다. 그러므로 이러한 Overlap criteria를 추가하면 일정구간 이상의 중복이 일어나지 않는 경로집합을 얻을 수 있다. 그러나 이 방법은 각 개별 경로들간의 중복 검토가 필요하므로 경로들이 많아지면 계산비용이 크게 증가하게 된다. 최근 Park et al.(2002)는 경로간 중복을 고려하면서 $k-path$ 를 산출하는 EVL(Efficient Vector Labeling)기법을 제시했는데, k 번째 최단경로의 통행시간제약과 중복 경로길이제약을 모형에 추가하였으며, 가능경로탐색을 효과적으로 수행하기 위하여 탐색공간을 축소시켜 최단경로를 탐색하는 방법을 적용하였다.

이에 반해, 본 연구에서 제시하는 $n-path$ 알고리즘은 $k-path$ 과는 달리 기중점간 통행비용과 경로중복수준(level of path overlap, Op)을 경로 선택기준으로 탐색하기 때문에 사전에 경로수를 확정하지 않고 경로 중복수준에 따라 n 개의 경로가 산출된다는 특징이 있다. 본 연구에서 개발되는 $n-path$ 알고리즘과 기존 $k-path$ 알고리즘과의 차이를 정리하면 <표 2>와 같다.

2. $n-path$ 알고리즘의 개발

1) $n-path$ 알고리즘의 기본개념

본 연구에서 개발되는 $n-path$ 알고리즘은 앞에서 기술한 일반 가로망의 교통운영특성(즉, 회전제약)들을 고려하면서, n 개의 경로를 산출해야 하기 때문에, 기존에 사용되는 노드에 기초한(node-based) 최단경로 탐색은 비효율적이다. 따라서, 본 연구에서는 링크에 기초(link-based)하여 최단경로를 탐색하게 된다. 링크 기반 최단경로 알고리즘은 Potts & Oliver(1972)에 의해 처음 개념적으로 제시되었는데, 그들은 노드를

기반으로 한 경로탐색 대신에 링크의 순차성만을 고려해 경로를 탐색할 경우 회전벌점(Turn penalty) 문제가 쉽게 해결될 수 있음을 보여 주었으며 김현명의 (1999)에서는 이를 교통수단간 환승을 고려하는 통합교통망에 적용하였다. 본 연구에서도 이를 기초로 *n-path* 알고리즘을 개발하게 되는데, 링크기반 최단 경로 알고리즘의 최적 조건은 식(1)과 같으며 도식화된 개념은 <그림 3>에 나타나 있다.

$$\begin{aligned}
 &LEC(o, i) + TP[link(o, i), link(i, j)] \\
 &+ LC(i, j) \leq LEC(i, j), \\
 &\forall o, i, j \in Nodeset
 \end{aligned}
 \tag{1}$$

여기서,

LEC(o, i) : 링크 (o, i)의 Link-end cost, 즉, 기점(origin)에서 링크 (o, i)의 종점노드 i까지 도착하는데 필요한 최소도착 비용

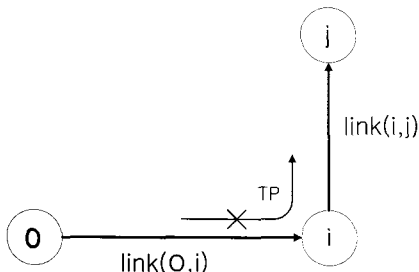
TP[link(o, i), link(i, j)] : 링크 (o, i)에서 링크 (i, j)로 진행하는 경로에 대한 회전벌점(Turn penalty)

LC(i, j) : 링크 (i, j)의 통행비용

최적경로 탐색이 링크기반으로 이루어짐에 따라 기존 알고리즘에서 전 노드(Predecessor Node)를 기억하는 대신, 링크기반에서는 전 링크(Predecessor Link, PL)를 기억하게 된다. 즉, <그림 3>에서 *link(i, j)*의 전 링크는 다음과 같이 저장된다.

$$PL(i, j) = link(o, i)
 \tag{2}$$

식(1)과 식(2)를 가지고 링크기반 최단경로 알고리



<그림 3> 회전제약을 고려한 링크기반 최단경로 알고리즘의 기본개념

즘을 단계별로 정리하면 다음과 같다. 여기서 *R*은 표지된(labelled) 링크집합이며, \bar{R} 는 표지되지 않은(unlabelled) 링크집합이다. 또, *O*는 기점(Origin)과 연결된 모든 노드의 집합을 나타낸다.

[Step 1] Label the *link(r, i)*, connecting origin node *r* with node *i*, $i \in O$
 enter *link(r, i)* into set *R* : $R = \{link(r, i)\}$
 set $LEC(r, i) = LC(r, i)$ and $LEC(r, j) = \infty$
 $\forall j \neq i$

$$PL(r, i) = \emptyset$$

[Step 2] Find an unlabelled link

If $LEC(i, j) + TP[link(i, j), link(j, k)] + LC(j, k) \leq LEC(j, k)$

then, $LEC(j, k) =$

$LEC(i, j) + TP[link(i, j), link(j, k)] + LC(j, k)$

$PL(j, k) = link(i, j)$

[Step 3] Label the *link(i, j)*

Add the *link(i, j)* to the set *R*, and delete it from the set \bar{R}

[Step 4] If $\bar{R} = \emptyset$ stop, otherwise go to Step 2.

링크기반 알고리즘의 장점은 쉽게 프로그래밍(coding)할 수 있으며 회전제약이 있는 경우에도 이를 쉽게 고려할 수 있다는 것이다. 또한, 이 알고리즘은 전통적인 수형망(Tree-based) 알고리즘과 매우 유사해서 현재 널리 사용되고 있는 기존 수형망 알고리즘을 약간 수정해서 사용할 수 있으며, 기존 수형망 알고리즘과의 차이는 노드를 기반으로 최단경로를 찾지 않고 링크를 기반으로 찾는다라는 점 뿐이다.

2) *n-path* 알고리즘

여기서는 앞에서 기술한 링크기반 최단경로 탐색 알고리즘을 이용하여 회전제약을 고려하면서 여러 개의 경로를 산출하는 링크기반 *n-path* 최단경로 알고리즘을 설명한다. 새롭게 사용되는 변수들은 다음과 같다.

- P^{rs} : 기종점 rs 간 n -path 최단경로 집합,
 $\{ P_n^{rs}, l_n^{rs} \}$
- P_n^{rs} : 기종점 rs 간 n 번째 최단경로
- l_n^{rs} : 기종점 rs 간 n 번째 최단경로의 길이
- $ol_{k/n}^{rs}$: 기종점 rs 간 n 번째 최단경로의 길이중,
 k 번째 경로와 중복된 길이(overlapping length),
 $(\forall P_k^{rs} \in P^{rs}, k = n-1, n-2, \dots, 1)$
- Op : 허용 경로중복도
- $Op_{k/n}^{rs}$: 기종점 rs 간의 n 번째 최단경로와 k 번째
 경로와의 중복정도
 $(Op_{k/n}^{rs} = \frac{ol_{k/n}^{rs}}{l_n^{rs}}, \forall P_k^{rs} \in P^{rs},$
 $k = n-1, n-2, \dots, 1)$
- Oz : 링크 패널티(penalty, $Oz = [\frac{1}{Op}]^a$)
- a : 파라메타

[단계0] 초기화

허용경로 중복수준(Op) 설정, $n = 1$

[단계1] 링크기반 최단경로 탐색알고리즘을 이용하여
 기점 r 에서 종점 s 까지의 최단경로를 탐색하여
 P^{rs} 에 저장, $P^{rs} = \{ P_n^{rs}, l_n^{rs} \}$

[단계2] n 번째 최단경로에 속하는 모든 링크에
 패널티(penalty) 부여
 최단경로에 속하는 링크의 갱신된 통행비
 용 = 링크 통행비용 * Oz

[단계3] 최단경로 탐색

- ① $n = n + 1$
- ② 링크기반 최단경로 탐색법을 이용하여
 새로운 최단경로 탐색, $\{ P_n^{rs}, l_n^{rs} \}$

[단계4] 수렴조건

다음 2가지 조건중 하나라도 만족하면 중지,
 그렇지 않으면 $\{ P_n^{rs}, l_n^{rs} \}$ 을 P^{rs} 에 저장한 후,
 [단계2]로 진행

- ① 만약 $Op_{k/n}^{rs} \geq Op$ 이면 중지,
 $(\forall P_k^{rs} \in P^{rs}, k = n-1, n-2, \dots, 1)$
- ② 새로 도출된 최단경로가 기존에 저장된
 최단경로와 동일하면 중지

위 알고리즘의 [단계2]에서 패널티를 부여하는 하는
 방법으로 경로중복 수준(Op)을 사용하게 된다. 즉, 경
 로 중복수준을 높게 설정할 경우, 예를 들어 모든 링크
 의 중복을 허용할 경우($Op=1.0$), $Oz=1$ 이 되어 단
 하나의 경로($n=1$)만 산출된다. 반면, 중복수준을 낮
 게 하면, $Oz>1.0$ 이 되어 다수의 경로($n>1$)가 산출
 하게 된다.

IV. 모형의 평가

개발된 모형을 평가하기 위한 2가지 가상가로망을
 이용하는데, 먼저 앞에서 살펴본 <그림 2>와 같이 연속
 좌회전이 존재하는 경우에 본 연구에서 개발된 알고리
 즈가 정확히 해를 찾는지를 평가한다. 이후 좀 더 현실
 적인 가로망으로 Sioux Falls 가로망을 대상으로 모형
 을 평가하며, 기존 k -path 알고리즘과 비교해 본다

1. 연속 좌회전이 존재하는 단순 가로망

<그림 2>와 같이 연속적으로 좌회전이 금지된 경우
 에, 링크기반 알고리즘을 이용하여 최단경로를 탐색하
 는 과정이 <표 3>에 자세히 나타나 있다. 기존 알고리
 즈를 이용했을 경우, 링크1, 2, 4, 6, 9를 계산하여
 잘못된 경로를 산출하는 데 비해, 링크기반 알고리즘은
 링크1, 3, 7, 8, 9로 정확한 경로와 12라는 최단경로
 비용을 정확히 산출하고 있다.

2. Sioux Falls 가로망

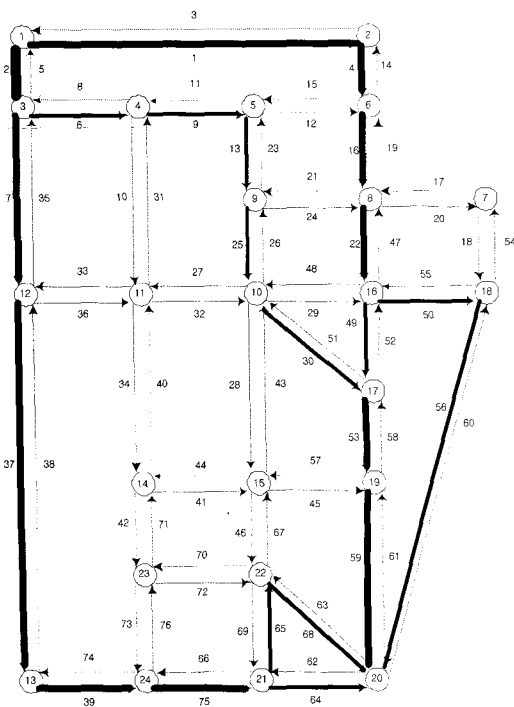
1) k -path 알고리즘과 n -path 알고리즘의 비교

기존 k -path 최단경로 알고리즘과 본 연구에서 제
 시한 n -path 알고리즘을 24개의 노드와 76개의 링크
 로 구성된 Sioux Falls 가로망에 적용하였으며,
 Sioux Falls 가로망의 입력자료는 [부록]에 수록되어
 있다. 여기서 기점은 노드1이며 종점은 노드20이고 분
 석의 편의를 위하여 모든 노드에 회전제약이 없다고 가
 정한다. n -path 알고리즘의 경우, $Op=0.6$ (즉, 경
 로간 60%이하의 중복허용), $a=1.8$ 을 적용한다

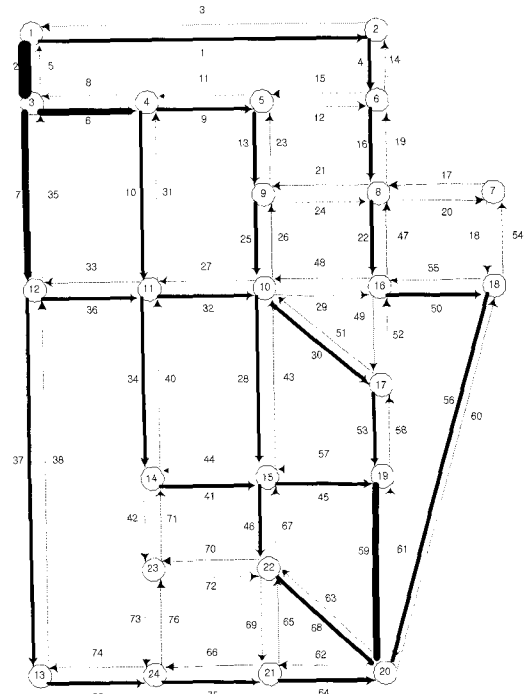
분석결과가 <그림 4>, <표 4>에 나타나 있는데, 각
 알고리즘별로 5개의 경로가 도출되었다. 그림에서 보듯
 이 k -path 알고리즘의 경우 상당수의 링크가 서로 중

(표 3) 링크기반 최단경로 탐색과정과 결과

number of link	i	j	$LC(i, j)$	$LEC(i, j)$	$PL(i, j)$	labelled link set (R)	link set for path
1	r	1	1	1	0	1	1
2	1	2	1	2 (1+1)	1	1,2	1,2
3	1	4	2	3 (1+2)	1	1,2,3	1,3
4	2	3	2	4 (1+1+2)	2	1,2,3,4	1,2,4
5	2	5	3	5 (1+1+3)	2	1,2,3,4,5	1,2,5
6	3	6	3	7 (1+1+2+3)	4	1,2,3,4,5,6	1,2,4,6
7	4	5	4	7 (1+2+4)	3	1,2,3,4,5,6,7	1,3,7
8	5	6	3	908 (1+1+3+900+3)	5	1,2,3,4,5,6,7,8	1,2,5,8
	5	6	3	10 (1+2+4+3)	7	1,2,3,4,5,6,7,8	1,3,7,8
9	6	s	2	909 (1+1+2+3+900+2)	6	1,2,3,4,5,6,7,8,9	1,2,4,6,9
	6	s	2	12 (1+2+4+3+2)	8	1,2,3,4,5,6,7,8,9	1,3,7,8,9
link set for shortest path				1-3-7-8-9	shortest path cost (r → s)		12



(a) k -path 알고리즘에서 산출된 경로



(b) n -path 알고리즘에서 산출된 경로

(그림 4) k -path와 n -path 알고리즘의 비교

복되어 나타나고 있는데 반해, n -path 알고리즘은 5개의 경로가 전체적으로 분산되어 나타나고 있음을 알 수 있다. 이는 k -path 알고리즘의 최단경로 탐색 특성상 당연한 결과이며, 제III장에서 기술했듯이 이들 경로를 직접 교통정보에 활용하기에는 무리가 있음을 알 수 있다.

한편, 경로별 통행비용을 살펴보면, 첫 번째 최단경로와 2번째 최단경로는 모두 동일한 비용을 갖고 있

나 이후 경로의 경우, n -path 알고리즘이 k -path 알고리즘 보다 크게 나타나고 있는데, 이는 n -path 알고리즘이 비용이 적은 순서대로 경로를 찾기 보다는 중복정도가 적은 경로를 찾기 때문에 나타나는 결과라 볼 수 있다. 따라서, 경로간 중복정도는 높지만 경로 통행비용이 적게 드는 경로들을 찾느냐(k -path), 아니면 경로비용은 좀 높지만 경로간 중복정도가 낮은 경로들을 찾느냐(n -path) 하는 문제가 대두된다. 이중

〈표 4〉 각 알고리즘별 경로비용과 사용된 링크

path	k-path		n-path	
	path cost	used links	path cost	used links
1	1260	2-7-37-39-75-64	1260	2-7-37-39-75-64
2	1320	1-4-16-22-50-56	1320	1-4-16-22-50-56
3	1320	1-4-16-22-49-53-59	1440	2-6-9-13-25-30-53-59
4	1440	2-6-9-13-25-30-53-59	1500	2-7-36-34-41-46-68
5	1440	2-7-37-39-75-65-68	1680	2-6-10-32-28-45-59

어떤 기법을 선택할 것인가는 사용목적에 따라 달라질 수 있으며, 교통량 분산을 목적으로 하는 우회경로 정보제공의 경우 *n-path* 알고리즘이 좀 더 적합하다고 판단되며, 여기에 본 연구에서 제시한 *n-path* 알고리즘의 의미가 있다.

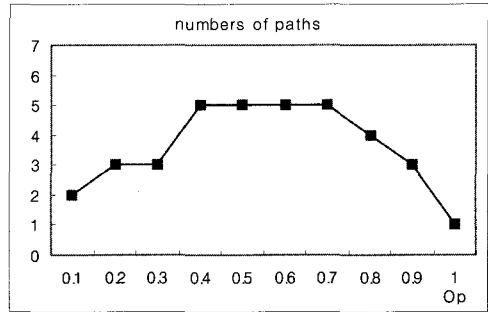
2) 회전제약이 있는 경우

n-path 알고리즘을 회전제약을 고려하여 분석한 결과는 다음과 같다. 여기서 기중점은 동일하게 노드1과 노드20이며, 3개의 연속된 노드(노드11, 노드14 그리고 노드23)에 대하여 좌회전을 금지시켰다.

먼저, 〈그림 5〉은 경로중복정도(Op)에 따라 생성되는 경로수를 나타내고 있다. 예상했던대로 $Op=1.0$ 인 경우, 즉 최단경로에 속하는 모든 링크를 중복해서 다른 경로를 찾을 수 있도록 허용한 경우에는 오직 1개의 경로(최단경로와 동일한 경로)만이 산출되었다. 그러나, $0.0 < Op < 1.0$ 경우에는 다양한 수의 경로들이 산출되어 본 연구에서 개발한 알고리즘이 링크의 중복정도를 고려하며 경로를 생성함을 확인할 수 있었다.

이중 가장 많은 수의 경로가 발생하는 $Op=0.5$ 와 $Op=0.6$ 에 대하여 생성된 경로 및 경로비용과 각 경로별 중복정도를 나타낸 행렬이 〈표 5〉와 〈표 6〉에 있다. 경로측면에서 두 경우 모두 5번째 경로를 제외하고는 동일한 경로를 산출하고 있다. 또한, 연속3개의 좌회전금지기가 존재하는 노드에서는 좌회전이 발생하지 않고 있으며, 〈표 5〉(a)의 경우, 노드23에서 좌회전이 허용되지 않기 때문에 노드24에서 U-턴을 하고 있음을 알 수 있다.

〈표 6〉은 생성된 경로들간의 중복도를 보여주고 있는데, 모두 허용된 중복수준(Op) 이하를 나타내고 있으며, 대각요소(diagonal element)는 자기 자신과의 경로중복을 의미하기 때문에 모두 1.0값을 보이고 있다. 예를 들어 $Op=0.5$ 에서 경로1의 경우, 경로2와는 전혀



〈그림 5〉 중복정도(Op)와 생성된 경로수의 관계

〈표 5〉 생성된 경로와 경로비용

(a) $Op=0.5$ 인 경우

산출된 경로	사용된 링크	경로 비용	비고
1st path (최단경로)	2-7-37-39-75-64	1260	
2nd path	1-4-16-22-50-56	1320	
3rd path	2-6-9-13-25-30-53-59	1440	
4th path	2-7-36-32-28-46-68	1560	
5th path	2-6-10-34-42-73-76-72-68	1860	노드24에서 U-턴

(b) $Op=0.6$ 인 경우

산출된 경로	사용된 링크	경로비용
1st path (최단경로)	2-7-37-39-75-64	1260
2nd path	1-4-16-22-50-56	1320
3rd path	2-6-9-13-25-30-53-59	1440
4th path	2-7-36-32-28-46-68	1560
5th path	2-6-10-34-42-73-75-64	1620

중복이 발생하지 않으며, 경로3과는 16.7%의 중복을 보이고 있고 경로4와는 최대 33.3%의 중복을 나타내고 있다. 이런 결과들을 볼 때 본 연구에서 개발된 알고리즘이 목적하는 결과를 도출하고 있음을 알 수 있다.

마지막으로 〈그림 6〉은 파라메타 α 값과 Op 값에 따른 발생경로수를 보여주고 있다. 경로중복도가 낮은 $Op=0.2$ 경우, α 값에 관계없이 3개의 경로만을 도출하는데 비해, $Op=0.5$, $Op=0.8$ 로 커지면, 다양한 수의 경로를 발생시키고 있는데, $\alpha=7.0$ 이상이 되면 모두 7개의 경로만을 도출하고 있다. 여기서 파라메타 α 값은 경로간의 중복을 방지하는 지수항으로 작용하기 때문에 α 값이 커지면 기 발생된 경로와는 다른 경로를 찾게 되기 때문에 여러개의 경로를 도출하는 것으로 판단된다.

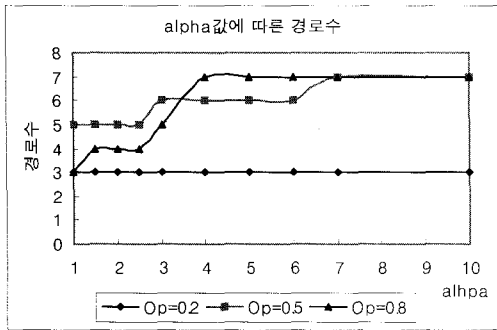
〈표 6〉 생성된 경로들간 중복정도

(a) $Op=0.5$ 인 경우

경로	1	2	3	4	5
1	1.000	0.000	0.167	0.333	0.167
2	0.000	1.000	0.000	0.000	0.000
3	0.125	0.000	1.000	0.125	0.250
4	0.286	0.000	0.143	1.000	0.286
5	0.111	0.000	0.222	0.222	1.000

(b) $Op=0.6$ 인 경우

경로	1	2	3	4	5
1	1.000	0.000	0.167	0.333	0.500
2	0.000	1.000	0.000	0.000	0.000
3	0.125	0.000	1.000	0.125	0.250
4	0.286	0.000	0.143	1.000	0.143
5	0.375	0.000	0.250	0.125	1.000



〈그림 6〉 α 값과 Op 값에 따른 발생 경로수

V. 결론

본 연구에서는 교차로상의 회전금지과 경로간의 중복정도를 고려하면서 합리적인 수준의 대안경로를 탐색하는 알고리즘을 제시하였다. 개발된 모형을 2개의 가상 가로망을 대상으로 평가한 결과, 최단경로(1st path)와의 중복정도를 고려하며 최단경로와는 다른 제 2, 제 3 등의 경로들이 생성되고 있어 기존 k -path 알고리즘과는 다른 결과를 보이고 있음을 알 수 있었다. 이는 다수의 링크 중복이 발생하는 k -path 알고리즘의 한계를 어느 정도 해결할 수 있을 것으로 보인다. 그러나, 본문에도 기술했듯이 경로비용측면에서는 k -path 알고리즘보다는 큰 값을 갖고 있는데, 이는 n -path 알고리즘과 k -path 알고리즘과의 탐색방법의 차이에서 발생한다고 볼 수 있다.

또한, 본 연구에서는 본격적으로 다루지 못했지만 현실에 사용할 수 있는 적정 경로 중복수준(optimal Op)

을 어느 정도로 할 것인가하는 문제가 대두된다. 저자의 판단으로 적정 Op 수준은 Ben-Akiva et al.(1991)이 제시한 특정경로에 대한 집중(concentration)현상이 발생하지 않을 정도의 경로중복이 적당할 것으로 생각되나, 이에 대한 구체적인 연구가 없는 실정이다. 따라서, 본 연구의 결과를 이용할 경우, 교통정보제공을 위한 대안도루수를 3~5개 정도로 설정한다면, 〈그림 5〉로부터 $Op=0.2\sim 0.4$ 정도, 즉, 경로간 중복수준이 20~40% 정도가 적당하다고 판단된다. 그러나 이는 단편적인 결과로 이에 대해서는 좀 더 심도 있는 연구가 필요할 것으로 보인다.

본 연구에서 제시된 알고리즘이 합리적인 수준의 대안경로들을 도출함에 따라 교통정보 제공분야에 활용될 수 있을 것으로 보이며, 확률적 통행배정모형에서 발생하는 비관련 대안경로간 독립성(IIA)문제를 완화시킬 수 있을 것으로 보인다. 향후 연구과제로는 제공된 안 내도로에 대한 정보를 얼마나 운전자들이 수용할 것인가에 대한 연구와 개발된 모형을 정적인 상태에서 동적 최단경로로 확장시켜 평가하는 분야가 있다.

참고문헌

1. 김익기(1998), ATIS를 위한 수정형 덩굴망 최단 경로 탐색 알고리즘의 개발, 대한교통학회지, 제16권 제2호, 대한교통학회, pp.157~167.
2. 김현명·임용택·이승재(1999), 통합교통망 수단선택-통행배정모형 개발에 관한 연구, 대한교통학회지, 제17권 제5호, 대한교통학회, pp.87~98.
3. 이승환·최기주·김원길(1996), 도시부 ATIS 효율적 적용을 위한 탐색영역기법 및 양방향 링크탐색 알고리즘의 구현, 대한교통학회지, 제14권 제3호, 대한교통학회, pp.45~59.
4. 최기주·장원재(1998), 복합 교통망에서의 최적경로산정 모형개발, 대한교통학회지, 제16권 제4호, 대한교통학회, pp.167~189.
5. Ahuja, R.K., Magnanti, T.L., Orlin, J.B. (1993) Network Flows: Theory, Algorithms and Applications. Prentice-Hall.
6. Ben-Akiva, M., A. De Palma, Kaysi, I.(1991) Dynamic Network models and Driver Information System," Transportation Research(A), Vol. 25A, pp.251~266.

7. Dijkstra, E. W.(1959) "A note on two problems in connection with graphs", Numer.Math.1, pp.269~271.
8. Cascetta Ennio, Agostino Nuzzolo, Francesco Russo, Antonino Vietta(1996) A Modified Logit Route Choice Model Overcoming Path Overlapping Problems, Specification and Some Calibration Result for Interurban Networks, ISTTT, Pergamon.
9. Park,D., S.L.Sharma, L.R.Rilett, M.Chang, (2002) Identifying multiple and reasonable alternative routes: Efficient Vector Labeling Approach, Transportation Research Record 1783, pp.111~118.
10. Potts,R.B., R.M. Oliver(1972) Flows in transportation networks, Academic press.
11. Shier, R.D.(1979) On algorithms from finding the k shortest paths in a network, Networks, Vol.9, pp.195~214.
12. Thomas, R.(1991) Traffic Assignment Techniques, Avebury Technical.
13. Yen J.Y.(1971) Finding the K shortest loopless paths in a network, Management Science, Vol 17, No. 11, pp.712~716.

✉ 주 작 성 자 : 임용택

✉ 논문투고일 : 2004. 4. 3

논문심사일 : 2004. 5. 17 (1차)

2004. 6. 14 (2차)

심사판정일 : 2004. 6. 14

✉ 반론접수기한 : 2004. 12. 31

부록

Sioux Falls 가로망 데이터

링크번호	출발노드	도착노드	링크비용	링크번호	출발노드	도착노드	링크비용
1	1	2	360	39	13	24	120
2	1	3	120	40	14	11	240
3	2	1	360	41	14	15	240
4	2	6	120	42	14	23	180
5	3	1	120	43	15	10	240
6	3	4	300	44	15	14	240
7	3	12	300	45	15	19	180
8	4	3	300	46	15	22	180
9	4	5	180	47	16	8	120
10	4	11	300	48	16	10	180
11	5	4	180	49	16	17	120
12	5	6	180	50	16	18	180
13	5	9	120	51	17	10	180
14	6	2	120	52	17	16	120
15	6	5	180	53	17	19	180
16	6	8	180	54	18	7	300
17	7	8	180	55	18	16	180
18	7	18	300	56	18	20	360
19	8	6	180	57	19	15	180
20	8	7	180	58	19	17	180
21	8	9	180	59	19	20	240
22	8	16	120	60	20	18	360
23	9	5	120	61	20	19	240
24	9	8	180	62	20	21	180
25	9	10	120	63	20	22	240
26	10	9	120	64	21	20	180
27	10	11	300	65	21	22	120
28	10	15	240	66	21	24	180
29	10	16	180	67	22	15	180
30	10	17	180	68	22	20	240
31	11	4	300	69	22	21	120
32	11	10	300	70	22	23	240
33	11	12	180	71	23	14	180
34	11	14	240	72	23	22	240
35	12	3	300	73	23	24	120
36	12	11	180	74	24	13	120
37	12	13	360	75	24	21	180
38	13	12	360	76	24	23	120