

□ 論 文 □

# 도시부 ATIS 효율적 적용을 위한 탐색영역기법 및 양방향 링크탐색 알고리즘의 구현

An Integration of Searching Area Extraction Scheme and Bi-directional Link Searching Algorithm for the Urban ATIS Application

이 승 환                      최 기 주                      김 원 길

(아주대학교 교통공학과 교수) (아주대학교 교통공학과 조교수) (해태전자 정보통신사업본부 ITS 사업부)

## 目 次

- I. 서론
- II. 이론적 배경
  - 1. 네트워크의 표현
  - 2. 일반 경로탐색알고리즘
- III. 연구모형의 개발
  - 1. U-TURN을 고려한 최단경로 구현
  - 2. 링크탐색 알고리즘
- 3. 탐색영역제한 모형
- 4. Bidirectional Dijkstra 수정알고리즘
- IV. 최단경로알고리즘의 적용 및 수행도 분석
  - 1. 적용방법
  - 2. 평가결과
- V. 결론
- 참고문헌

## ABSTRACT

The shortest path algorithm for route guidance is implicitly required not only to support geometrical variations of transportation network such as U-TURN or P-TURN but to efficiency search reasonable routes in searching mechanism. The purpose of this paper is to integrate such two requirements; that is, to allow U-TURN and P-TURN possibilities and to cut down searching time in locating routes between two points (origin and destination) in networks. We also propose a new type of link searching algorithm which can solve the limitation of vine building algorithm at consecutively left-turn prohibited intersections. The test site is a block of Gangnam road network that has some left-turn prohibited and allowed U-TURN intersections. Four models have been identified to be comparatively analyzed in terms of searching efficiency. The Models are as follows:

- ( i ) Model 1 - Link Searching Dijkstra Algorithm without Searching Area Extraction (SAE);

- (ii) Model 2 - Link Searching Dijkstra Algorithm with SAE;
- (iii) Model 3 - Link Searching Bidirectional Dijkstra Algorithm without SAE; and
- (iv) Model 4 - Link Searching Bidirectional Dijkstra Algorithm with SAE.

The results of comparative evaluation show that Model 4 can effectively find optimum path faster than any other models as expected. Some discussions and future research agenda have been presented in the light of dynamic route guidance application of the urban ATIS.

## I. 서론

21세기 정보화사회를 대비하여 최근에 국내 외에서 활발한 연구가 진행되고 있는 첨단교통 체계(Intelligent Transportation System: ITS)는 교통의 전분야에 정보 전자 제어 통신 등의 신 기술을 이용하여 교통의 안전성, 쾌적성, 효율성을 극대화시키고자 하는 시도이다. 이 중 경로 안내시스템(route guidance system)은 운전자들의 출발지로부터 목적지까지 교통상황을 기초로 최적 경로를 제공해주기 위한 ITS의 부체계로 볼 수 있다. 이미 미국, 일본, 유럽 등의 국가에서는 이 시스템에 대한 중요성을 인식하고 연구 개발에 착수하여 현재 현장실험을 거쳐 이미 실용화수준에 이르러 있는 시스템이다 (예를들면, 일본의 VICS).

국내의 경우 1992년부터 자동차 회사를 중심으로 경로안내시스템의 개발을 시작하였으며, 최근까지의 국내 관련연구들은 항법요소에 GPS (Global Positioning System), 측위항법 (Dead Reckoning), 및 지도매칭(Map Matching) 등의 분야에서 연구가 진행중에 있으며, 현대자동차의 경우 이러한 서비스를 금년 하반기에 출시할 예정으로 되어있다.

한편, 이를 구현하는 요소로서 국내실정에 맞는 최단경로알고리즘의 네트워크상에서의 구현에 관한 연구가 그간 교통분야에서 진행되어온 바 있고(최기주, 1995) 도시부의 적용사례(노정현,

남궁성, 1995) 등이 제시된 바 있다. 본 연구는 최단경로안내시스템을 구축함에 있어서 첫째, 기존 개발된 Dijkstra 알고리즘을 기반으로 기존의 연구를 수용하면서 국내의 전형적인 도로망 기하구조 특성인 좌회전금지 및 U-TURN 및 P-TURN을 고려하고, 둘째, 출발지로부터 목적지까지의 최단경로를 탐색하는데 있어서 소요되는 탐색시간과 탐색노드를 줄일 수 있는 장치를 도출하며, 셋째, 기존의 덩굴망 알고리즘의 단점<sup>1)</sup>을 극복할 수 있는 링크탐색방법 및 양방향 링크탐색 알고리즘의 적용을 시도함으로써 도시부 가로망에 있어서의 보다 향상된 경로탐색모형을 구축해 보자는데 그 목적이 있다.

## II. 이론적배경

### 1. 네트워크의 표현

교차로에서의 노드체계에 대한 구분은 교차로를 하나의 노드로 나타내는 방법(이하 단일노드체계)과 사지교차로 방향별로 하나씩의 노드를 설정하여 4개의 노드로 나타내는 방법(이하 접근로별 노드체계), 노드수의 증가없이 회전교통류를 파악하기 위하여 링크블럭 중간에 노드를 주는방법(이하 Mid-Block 노드체계) 등이

1) 기존의 덩굴망 알고리즘은 연속적으로 회전이 금지된 구간에서는 최단경로를 도출하는데 단점을 가지고 있다.

주로 사용되고 있다(Roy Thomas, 1991). 그러나 네트워크 표현방식에서 하나의 사지교차로를 하나의 노드와 8개의 링크로 교차로를 표현할 경우 회전제한을 나타내는데 어려움이 있을뿐만 아니라 교차로로 들어오는 교통류는 회전방향별로 통행시간에 차이가 있게되지만, 모든 교통류가 목적지에 상관없이 똑같은 여행시간을 경험할 것이라고 가정하게 된다(D.E.Boyce, A.Tarko, S.Berka, Y.Zhang, 1994). 또, 각 회전교통류별로 파악하기 위하여 노드를 확장하여 사지교차로를 표현할 경우 하나의 노드와 8개의 링크에서 확장하여 4개의 노드와 18개의 링크로 표현할 경우 노드와 링크수가 늘어나게되어 각 이동류별로 구분하여 표현할 수 있다는 장점은 있지만 계산시간면에서의 효율성은 떨어지게 된다. 이 부분에 관한 보다 자세한 사항은 최기주(1995)를 참조하면 된다.

또, 교차로에서의 회전특성을 반영하기 위해서 사용되는 Mid-Block 노드체계의 경우에도 단순히 직진, 좌회전, 우회전 등 각 이동류별로 구분하여 적용할 수 있다는 장점은 있지만, 국내와 같이 교차로에서의 U-TURN과 Mid-Block에서의 U-TURN 및 P-TURN이 상호 복잡하게

혼재되어 사용되는 환경에서는 정확하게 그 링크 및 교차로의 특성을 반영하기가 어렵게 된다.

## 2. 일반 경로탐색 알고리즘

일반적으로 최단경로알고리즘은 네트워크의 유형과 알고리즘의 접근방법, 최단경로문제의 유형에 <표 1>과 같이 구분될 수 있다. 먼저 네트워크의 유형에 따라서 수형망(Tree Building) 알고리즘과 덩굴망(Vine Building) 알고리즘으로 구분되며, 알고리즘의 접근방법에 따라서 Label-Setting 알고리즘과 Label-Correcting 알고리즘으로 구분된다. 또, 최단경로문제의 유형에 따라서 링크의 길이가 비음(non-negative)일 경우 하나의 노드로부터 모든 다른 노드들까지의 최단경로를 찾는 경우(Single-source shortest path problem 또는 shortest path problem), 모든 노드로부터 다른 모든 노드까지의 최단경로를 찾는 경우 (All-pairs shortest path problem), 다수의 경로를 찾는 경우, 외판원 문제(Travelling Salesman's Problem, TSP)로 나눌 수 있다(강명규, 1991).

<표 1> 최단경로 알고리즘의 구분

경로산출방식	경로제공방법	알고리즘 예
하나의 노드로부터 다른 모든노드들까지의 최단경로	하나의 최단경로 제시	Dijkstra, Moore
	여러개의 최단경로 제시	Shier (k개의 최단경로제시)
모든 노드로부터 다른 모든 노드들까지의 최단경로	하나의 최단경로 제시	Dijkstra(반복수행) Floyd-Warshall
	여러개의 최단경로 제시	Floyd-Warshall

수형망 알고리즘은 비교적 간단하고 계산이 빠르지만, 교차로에서의 회전제한에 대한 특성을 반영할 수 없다는 단점이 있다. 반면에 덩굴망 알고리즘은 탐색소요시간측면에서 수형망

알고리즘에 비해 2배 이상 소요되는 단점이 있으나 교차로에서의 회전금지나 회전벌점(TURN Penalty)이 적용되는 경우에는 장점이 있다(임강원, 1992).

### Ⅲ. 연구모형의 개발

본 연구는 덩굴망 알고리즘을 기반으로 다양한 U-TURN 혹은 P-TURN<sup>1</sup> 특성을 갖는 망구조에서 효과적으로 탐색될 수 있는 알고리즘을 구축하기 위하여 다음 <표 2>와 같이 4개의 연구모형을 정립하여 각 모형에 대한 수행도를 평가하였다.

위 <표 2>에서 탐색영역제한은 기존의 덩굴

망 알고리즘의 특성상 초기에 설정된 모든 영역을 탐색함으로 인하여 나타나는 탐색소요시간의 비효율성을 최소화 하기 위한 방법으로, 이는 기존 알고리즘들의 실시간 적용에 발생될 수 있는 근본적인 문제를 해결하기 위함이다. 따라서 본 연구에서는 탐색영역제한 방법을 적용하여 알고리즘이 탐색하는데 소요되는 시간을 최소화할 수 있는 방법을 이용하였다.

<표 2> 수행도 평가모형

	기본 알고리즘	U-TURN 고려	탐색영역제한(SAE)
Model 1	Dijkstra	○	×
Model 2	Dijkstra	○	○
Model 3	Bidirectional Dijkstra	○	×
Model 4	Bidirectional Dijkstra	○	○

○: 고려    ×: 고려하지 않음

#### 1. U-TURN을 고려한 최단경로 구현

##### 1) U-TURN 유형구분

우리나라의 간선도로에서 좌회전을 금지시킨 경우 일반적으로 사용되고 있는 U-TURN의 유형을 살펴보면 좌회전이 금지되어 다음 교차로에서 U-TURN 하는 경우와 인접 교차로사이의 횡단보도를 이용하는 경우, 인접지역을 P-TURN 하는 경우로 구분될 수 있다. 또, 링크중간에서의 U-TURN은 한방향으로만 허용하는 경우와 양방향 모두를 허용하는 경우로 나눌 수 있다. 이러한 다양한 경우들에 대한 U-TURN 및 P-TURN을 고려하여 최단경로탐색 알고리즘의 구축은 상기술된 각 경우들의 U-TURN과 P-TURN 등의 회전특성을 고려할 수 있는 네트워크 표현방법, 파일구조 등에 대하여 구현하고, 이들을 기반으로 본 연구에서 목적인 알고리즘을 구축하였다.

본 연구에서는 탐색모형에 링크 및 교차로의

기하구조적 특성을 고려하기 위하여 해당 링크의 U-TURN이 허용되지않는 경우에는 유형 0으로, 링크중간에 한방향 U-TURN일 경우와 양방향 U-TURN일 경우에는 유형 1로 통합하되 방향성을 부여하며, 신호교차로에서의 U-TURN은 유형 2로 두었다. 이밖에 P-TURN은 유형 3으로 설정하였으며, 각 유형이 혼합된 경우는 주로 신호교차로에서의 U-Turn과 블록중간에서의 U-TURN이 혼합된 경우로서 이를 유형 4로 분류하였다. 특히, 유형 3에 해당하는 P-TURN의 경우 U-TURN의 특수형태에 해당하는 것으로 보고 P-TURN을 U-TURN에 포함시켜 유형 3으로 분류하였다.

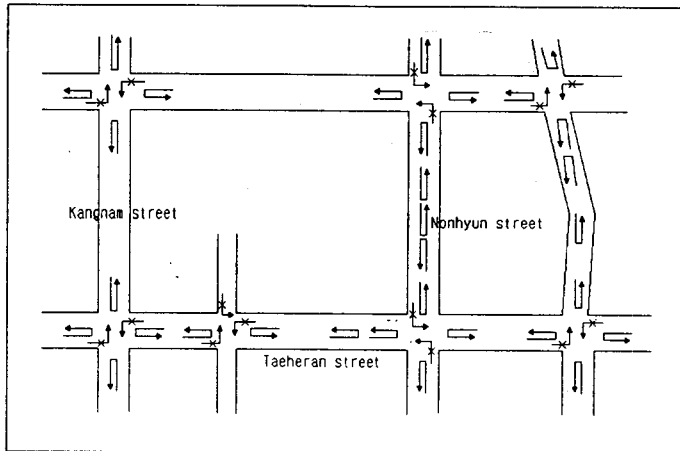
##### 2) U-TURN 노드체계

운전자의 출발지로부터 목적지까지의 U-TURN을 고려하여 최단경로를 탐색하기 위한 링크와 노드의 표현방법은 다양한 접근방법이 있을 수 있다. 회전지점을 하나의 노드로 설정

하는 단일노드설정 방법과 방향성에 따라 두 개이상의 노드로 설정하는 방향성 노드설정방법(복수개 노드설정)을 적용할 수도 있다. 그러나 장래에 우리나라에 경로안내체계가 구축될 경우 각 U-TURN 지점을 하나 또는 그 이상의 노드로 설정하게 되면 서울시의 경우 실제 교차로수의 50%이상의 노드가 더 필요하게 된다.<sup>2)</sup> 더우기 중요교차로와 이면도로를 동시에 고려할 경우에는 더 많은 노드설정으로 인하여 U-TURN을 기존노드와 같이 실제노드로 설정하

여 목적지까지의 최단경로를 찾는 것은 비효율적이다.

〈그림 1〉에 제시된바와 같이 우리나라의 경우 한 링크상에서 1지점이상의 U-TURN과 교차로에서의 좌회전 등의 U-TURN을 동시에 허용하는 곳이 많으며, 이런 지점들을 모두 노드로 지정하여 최단경로를 탐색하기 위해서는 과도한 노드설정으로 인하여 주요 교차로를 중심으로 한 노드체계를 표현하는데 어려움이 따르게 된다.



〈그림 1〉 네트워크의 예 (강남대로, 논현로, 언주로, 1995년 1월 현재)

따라서 본 연구에서 네트워크표현을 4지 교차로의 경우 하나의 노드와 8개의 링크로 구성하되 링크상의 회전지점에 대한 정보를 노드와 노드를 연결하는 링크에 대한 정보로서 사용하였다. 블록중간의 신호교차로 근처에서의 U-TURN을 하나의 노드로 볼 수도 있지만, 탐색과정에서 항상 모든 방향에 대해서 탐색대상이 되는 노드가 아니라 U-TURN을 고려할 필요가 있을 경우에만 노드로 인식되는 가상노드(Dummy Node)로 처리하였으며, 신호교차로 근처에서의 U-TURN은 신호교차로의 노드를 그대로 사용하였다. 본 연구에서 목적인 알고리

즘은 U-TURN과 같은 회전특성을 하나의 노드로 설정하지 않음으로 인해 직진차량의 경우 회전지점을 노드로 인식 및 탐색을 하지않고 원래의 연결노드를 탐색하게 된다.

### 3) 네트워크 자료구조

네트워크의 정보는 노드에 대한 속성정보와 링크에 대한 속성정보로 구분될 수 있다. 본 연구에서는 노드 및 링크에 대한 속성정보를

2) 수행도 분석대상 네트워크인 강남 일부지역의 경우 노드가 50%이상 증가하게 되며, 주요 교차로를 중심으로 한 링크비용의 산정이 어렵게 된다.

개별속성 파일로 구성하지 않고 인접리스트를 사용하여 하나의 파일로 구성하였다.

구성 노드에 대한 정보로는 각 교차로의 X, Y 상대 좌표값과 도로명, 링크길이, 고속도로나 간선 도로 등의 링크유형에 대한 정보와 회전 제한여부에 대한 정보, 회전금지노드, U-TURN 여부, 링크비용 등이 링크 속성정보에 해당한다. 이밖에 링크속성정보에 해당될 수 있는 정보항목으로는 차선수, 차선폭, 구배, 제한속도, 포장상태 등이 있다.

노드와 링크를 각각의 속성정보로 구분하는 방법외에 링크고정정보 프레임(Link Fixed Data Frame)과 링크가변정보 프레임(Link Variable Data Frame)으로 구분이 가능하다. 이런 구분은 링크유형이나 회전여부, 회전노드, U-TURN 유형 등의 링크고정정보 프레임은 정보의 갱신간격이 링크가변정보 프레임에 비해 상대적으로 더 긴 반면 링크비용이나 사고정보, 도시교통관제시스템으로 부터의 정보 등의 링크가변정보 프레임은 정보의 갱신간격이 5분 또는 15분간격(임의 갱신주기)으로 갱신되는점을 감안한 것이다.

또, 링크에 대한 정보외에 회전에 대한 정보를 하나의 파일로 구성하였다. 회전노드 ID와 상대 좌표값, 노드의 방향성(한방향 또는 양방향 회전여부) 등의 노드정보와 연결노드, 링크길이, 링크비용 등을 나타내는 링크정보를 회전 파일로 구성하였다.

#### 4) U-TURN을 고려한 최단경로알고리즘

운전자가 운행도중 U-TURN을 고려하는 경우는 일반적으로 두가지가 있다. 먼저 교차로에서 좌회전이 금지되어 교차로 통과후 링크중간이나 다음 신호교차로에서 U-TURN하는 경우와 목적지가 차량이 운행되는 방향과는 반대방향에 있어서 U-TURN을 고려해야 하는 경우로

나눌 수 있다.

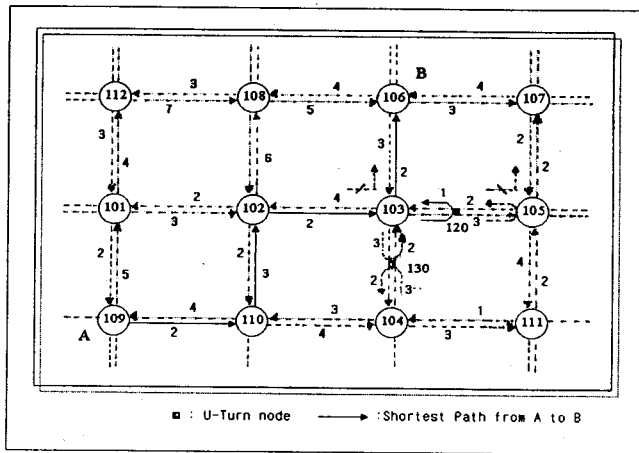
덩굴망 알고리즘은 좌회전이 금지된 경우 해당노드에서의 좌회전에 대해 회전벌점(TURN Penalty)을 부과하여 좌회전노드로의 접근을 제한하고 있다. 본 연구알고리즘에서는 회전이 금지된 경우 현재의 탐색노드와 이전노드(Backnode)의 교차로 운영정보에 회전여부에 대한 정보를 회전이 허용되는 경우에는 0의 값을, 회전이 금지된 경우에는 1의 값을 부여하였다. 만약 회전이 금지된 경우 현재의 탐색노드의 회전노드와 이전노드를 제외한 모든 인접노드들에 대한 U-TURN 여부와 유형을 파악하여 현재의 탐색노드까지의 통행비용에 최소비용을 갖는 회전노드의 비용과 회전금지노드의 통행비용을 더하여 회전통행비용으로 주게된다.

예를들어 <그림 2>에서 현재의 노드가 103이고 前노드가 102, 前前노드가 110이라고 하자. 링크 (102, 103)로부터 106 방향으로의 좌회전이 금지되었으므로 링크 (102, 103)의 좌회전 허용여부의 값은 1이 된다. 따라서 106까지의 전체 통행비용은 103번 노드와 105번 노드사이의 120번 U-TURN 노드와 103번 노드와 104번 노드사이의 130번 U-TURN 노드의 회전비용중 최소의 비용을 갖는 노드를 탐색한후 이 비용을 좌회전비용으로 더해지게 된다. 이때 좌회전금지위를 위한 회전노드에 대한 정보는 링크 (102, 103)에 부여하는 것이 아니라 링크 (103, 105)가 유형 1의 회전정보를, 링크 (103, 104)가 유형 2의 회전정보를 갖게된다.

U-TURN을 고려하여 최단경로를 찾는 본 연구알고리즘은 U-TURN 지점을 기존노드와 같은 하나의 고정노드로 인식하지않고 좌회전이 금지된 링크에서만 인접 연결노드의 U-TURN 허용여부와 유형을 파악하여 최단비용의 U-TURN 노드를 탐색한 후 좌회전이 금지된 노드의 링크비용으로 부여한다는 점이다. 만약 좌

회전이 금지되어있는 노드의 인접노드의 U-TURN 유형이 모두 0 즉, U-TURN 노드가 없을 경우에는 좌회전비용으로서 회전별점을 부여하게 된다. 해당링크에서의 U-TURN이 두 지점 이상에서 허용된다면, I-node를 기준으로 가까운 곳에 위치한 지점부터 회전에 대한 정보를

제공해주면 된다. <그림 2>의 네트워크를 파일로 나타내면 <표 3>에 제시되어 있다. <표 3>의 윗부분은 <그림 2>에 대한 네트워크 파일구조, 아랫부분은 그에 따른 U-TURN 특성에 대한 속성파일이다.



<그림 2> 네트워크의 예

<표 3> 네트워크 파일의 구조

I node	J node	링크 유형	회전 여부	회전 별점	회전 노드	U-TURN 유형	U-TURN 노드	링크 길이	유고 정보	UTCS 정보	링크 비용
102	101	3	0			0		100	-	-	2
102	103	3	1	99	106	0		120	-	-	2
102	108	3	0			0		200	-	-	6
102	110	3	0			0		150	-	-	2
103	102	3	0			0		120	-	-	4
103	104	3	0			1	130	150	-	-	3
103	105	3	1	99	107	4	120	250	-	-	3
103	106	3	0			0		200	-	-	2
...								...			
109	101	3	0			0		150	-	-	5
109	110	3	0			0		100	-	-	2
110	102	3	0			0		150	-	-	3
110	104	3	0			0		120	-	-	4
110	109	3	0			0		100	-	-	4
회전노드	X좌표	Y좌표	방향성	J-node	거리	유고 정보	UTCS 정보	링크 비용			
120	620	350	1	103	100	-	-	1			
130	550	400	2	103 104	75 75	-	-	2 2			

## 2. 링크탐색 알고리즘

### 1) 링크탐색 알고리즘의 구현

덩굴망 알고리즘은 前노드의 개념을 확장하여 前노드의 前노드인 前前노드의 정보를 이용하여 최단경로를 도출하고 있다. 그러나 덩굴망 알고리즘의 경우 회전이 연속적으로 금지된 경우 회전이 금지된 수만큼 前前노드의 개념을 확장하여야만 문제를 해결할 수 있기 때문에 확장된 노드수만큼의 메모리가 더 소요되게 된다.

본 알고리즘에서는 노드에 표지를 설정하는 방법이 아니라 노드와 노드를 연결하는 링크에 前링크의 표지를 설정하는 방법을 사용하였으므로 前前노드에 대한 정보가 필요없으며, 회전이 연속적으로 금지된 구간에서도 前링크에 대한 정보만 있으면 쉽게 최단경로의 도출이 가능하게 된다.

일반적으로 링크의 수는 노드수에 비해 3배 이상이 되므로 링크를 탐색하기 위해서는 그만큼 더 많은 계산시간이 소요된다. 그러나 본 알고리즘에서는 링크를 탐색하되 노드를 기반으로 하여 링크를 탐색함으로써 계산시간면에서는 노드에 라벨을 설정하는 덩굴망 알고리즘과 비교하여 볼 때 차이가 없을뿐만 아니라 연속적으로 회전이 금지된 경우에 대해서도 효과적으로 해결할 수 있다. 이는 해당 노드가 단지 하나의 前노드에 대한 정보를 가지고 있는 것이 아니라 연결된 링크에 라벨을 설정하므로 복수개의 前노드를 갖도록 함으로서 가능하다. 이 뿐만 아니라 사지 교차로의 경우 하나의 노드와 8개의 링크로 구성하면서 이동류별로 교차로의 교통상황을 반영하여 좌회전, 우회전, 직진 이동류에 대해 보다 정확한 통행시간을 제공할 수 있다.

본 연구에서 구분한 U-TURN 유형들중 유형

1 (Mid-Block에서의 U-TURN)의 경우 별도의 링크로 설정되어 있으므로 덩굴망 알고리즘에서 前노드를 추적하는데 별 문제점이 없지만, 유형 2 (신호교차로 부근에서의 U-TURN)의 경우에는 탐색과정에서 회전이 발생하는 지점을 가상의 노드를 설정함으로써 쉽게 해결이 가능하다. 가상노드를 설정하는 것은 탐색과정중에 회전이 금지된 교차로에서 회전후의 노드가 라벨이 설정될 경우에만 해당하므로 계산시간에는 별다른 차이를 보이지 않게 된다. 본 연구에서 검토된 알고리즘을 살펴보면 다음과 같다.

링크탐색 최단경로알고리즘(Link Searching Shortest Path Algorithm)

#### Step 1 : 초기화

- 모든 노드  $i, j$ 에 대해  $C_{ij}$ 를 임시표지로 둔다. 출발노드  $h$ 의  $C_{hh}$ 는 0로 두고 나머지 노드는 모두 0로 둔다.
- 각 링크의 前링크를 모두 0로 둔다.

#### Step 2 : 최소비용의 노드탐색

- 각 노드에 대해 연결노드와 라벨이 설정되어 있는 링크  $(i, j)$ 를 찾는다.
- 노드  $(i)$ 와 노드  $(j)$ 의 회전여부를 파악하여, 만약 회전이 금지된 경우에는 단계 3으로 간다.
- 노드  $(j)$ 와 연결되어 있는 노드  $(k)$ 의 비용을 탐색하여  $C_{jk}$ 를 임시표지로 둔다. ( $C_{jk} = C_{ij} + Cost_{jk}$ )
- 단계 4로 간다.

#### Step 3 : 회전노드 탐색

- $j$ 와 연결된 링크들중  $i$ 와  $k$ 를 제외한 링크의 U-Turn 유형과 비용을 탐색하여 최소의 비용



을 갖는 링크를  $C_{jk}$ 의 임시표지로 두고, 회전 노드는  $l$ 로 둔다.

- 단계 4로 간다.

**Step 4 : 최소비용의 링크선정**

- 임시표지로 설정된 링크들중 최소비용의  $C_{jk}$ 를 탐색하여  $C_{jk}$ 를 영구표지로 두고, 이때  $(j, k)$ 의 前링크는  $(i, j)$ 로 둔다.
- 만약  $(i, j)$ 의 회전이 금지되었다면,  $(j, k)$ 의 前링크는  $(l, j)$ ,  $(l, j)$ 의 前링크는  $(j, l)$ ,  $(j, l)$ 의 前 링크는  $(i, j)$ 로 둔다. 이때,  $(j, l)$ 의 U-Turn 유형이 2에 해당한다면,  $l$ 을 가상의 노드로 설정한다.
- $k$ 가 목적노드이면 단계 5로 가고 그렇지않은 경우에는 단계 2로 간다.

**Step 5 : 종료 및 최단경로도출**

- $(j, k)$ 의 前링크  $(i, j)$ 를 출력한다.

-  $j$ 를  $k$ 로  $i$ 를  $j$ 로 둔다.

- $i$ 가 출발노드일때까지 단계 5를 반복한다.

링크탐색 알고리즘의 또다른 장점으로서는 기존 단일노드체계의 경우 교차로로 접근하는 모든 교통류에 대해 교차로에서 좌회전, 우회전, 직진에 관계없이 일률적인 통행시간을 적용하였지만, 링크탐색 알고리즘의 경우에는 교차로를 하나의 노드로 표현하되 링크를 탐색해나가는 도록 함으로써 노드나 링크수의 증가없이 교차로를 하나의 노드와 8개의 링크로 구성하면서 이동류별로 교차로의 교통상황을 반영하여 좌회전, 우회전, 직진에 보다 정확한 통행시간을 제공할 수 있다.

<그림 2>에 제시한 예제 네트워크에서 노드 A에서부터 B까지의 최단경로를 본 연구에서 제시한 링크탐색 최단경로알고리즘으로 탐색하여 보면 그 과정은 <표 4>와 같다.

<표 4> 최단경로탐색결과

Iteration No.	A Node	B Node	Back Link	U-TURN	Total Cost
1	109	110	-	-	2
2	110	102	109 → 110	-	5
3	109	101	-	-	5
4	110	104	109 → 110	-	6
5	102	103	110 → 102	-	7
6	101	102	109 → 101	-	8
7	101	112	109 → 101	-	9
8	104	103	110 → 104	-	9
9	104	111	110 → 104	-	9
10	103	106	102 → 103	120	10
최단경로(A→B) : 109 → 110 → 102 → 103 → 120 → 103 → 106					

**3. 탐색영역제한 모형**

임의의 두 지점간의 최단경로는 출발노드를 원의 중심으로하여 탐색대상을 넓혀가는 형태가 되며, 목적노드가 탐색노드가 될경우 탐색을

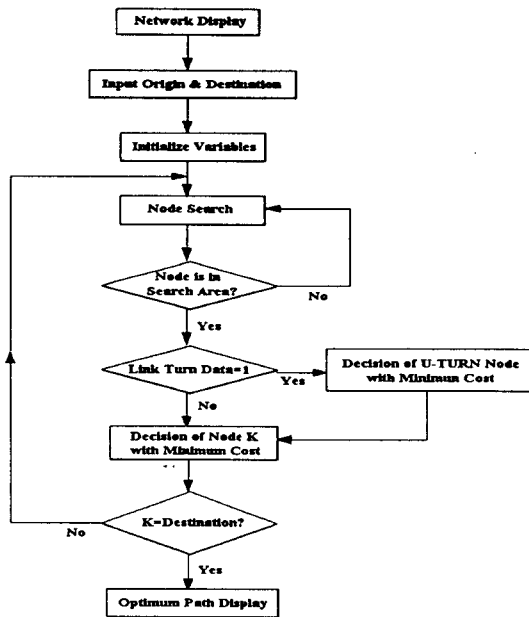
종료하게 된다. 그러나 출발노드와 목적노드의 거리가 상당히 길 경우 최단경로탐색과정에서 최단경로와는 아무런 상관이 없는 노드까지도 탐색대상에 포함되는 경우가 발생한다. 이는 계산시간상에서 상당한 시간을 필요로 하게 되며,

목적노드까지의 통행에 불필요한 노드들까지 탐색대상에 포함되므로 하나의 출발노드로부터 하나의 목적노드까지 최단경로를 탐색할 경우에는 비효율적이다. 최근 일본 등지에서도 차량항법시스템에서의 최단경로 탐색시간을 줄이기 위해 탐색영역을 제한하는 여러 가지 시도가 있었으며, 이는 탐색시간을 감소시키는 것은 차량항법시스템의 효율성에 상당한 영향을 미치리라 사려된다..

본 연구에서는 출발노드와 목적노드를 중심으로 거리나 링크비용 등을 고려하여 기준에 적합한 탐색대상영역을 형성한후 탐색대상내부의 노드와 설정된 기준에 적합한 노드에 대해서만 탐색을 하는 것은 탐색결과로 나온 최단경로면에서는 아무런 문제점이 없었으며, 계산시간이나 탐색반복횟수에서는 상당한 이점을 가져오게 된다.

탐색영역제한 방법은 여러 가지 접근방법이 있을 수 있다. 일반적으로 링크간의 상대좌표에

의해서도 제한이 가능하지만, 상대좌표에 의한 영역제한은 고속도로나 도시고속도로, 순환도로, 고가차도(overpass) 또는 지하차도(underpass)로 연결된 링크를 탐색도중 상대좌표에 의해서 탐색대상에서 제외하게 되는 경우가 발생할 수도 있는 위험요소가 뒤따르게 된다. 따라서 각 노드의 상대좌표값과 링크비용이나 도로유형, 유도발생여부 등을 함께 고려하여 모형에 적용시켜야 한다. 링크비용을 고려하여 탐색대상노드를 결정하기 위해서는 교통량, 통행시간, 포화도, 대기행렬길이 등의 도시교통관제시스템(Urban Traffic Control System)에서의 교통정보를 이용해서 링크의 실시간 교통상황을 반영할 수 있는 변수들을 이용하여 링크의 혼잡도를 정확히 반영할 수 있는 링크지표를 만든후 탐색영역을 결정해나가도록 모형을 구축하는 것이 보다 바람직하리라 사려되며, 탐색시간을 줄이기 위해서는 이에 대한 보다 구체적인 연구가 필요하리라 생각된다.



<그림 3> 탐색영역제한 모형의 알고리즘 흐름도

#### 4. Bidirectional Dijkstra 수정알고리즘

Forward Dijkstra 알고리즘과 Reverse Dijkstra 알고리즘을 동시에 적용하여 출발노드와 목적노드를 원의 중심으로하여 동시에 노드를 탐색해나가는 Bidirectional Dijkstra 알고리즘은 계산 시간면에서 비교해볼 때 Forward Dijkstra 알고리즘을 적용할 때보다 빠르다는 장점이 있다. Bidirectional Dijkstra와 Forward Dijkstra의 차이점을 비교해보면, 후자의 경우에는 목적노드가 영구표지로 표시되면 탐색이 종료되게 되지만 전자의 경우에는 탐색과정중 출발노드로부터 영구표지된 노드와 목적노드로부터 영구표지된 노드중 서로 overlap되는 노드가 있으면 일단 탐색이 종료되게 된다. 그러나 출발노드와

overlap되는 노드, 목적노드를 연결한 경로가 항상 최단경로가 되지는 않으므로 탐색과정중에 각 노드로부터의 최소비용을 갖는 경로를 찾아서 비교한후 최소의 비용을 갖는 경로가 최단경로가 된다.

이 과정에서 Overlap되는 링크를 기준으로 한 최단경로와 출발지와 목적지까지 경로로 연결된 모든 route들을 비교하여 최소의 비용을 갖는 경로를 산정하기 때문에 양방향 Dijkstra 알고리즘에서 산출된 최단경로와 일반 Forward Dijkstra 알고리즘에서 산출된 최단경로와는 항상 같은 최단경로를 찾아내게 된다.

〈그림 2〉의 네트워크의 A에서 B까지의 최단 경로탐색에 Bidirectional Dijkstra 알고리즘을 적용해본 결과는 〈표 5〉와 같다.

〈표 5〉 Bidirectional Dijkstra 알고리즘을 적용한 최단경로 탐색결과

Itera No	A	B	BackLink	U-TURN	Cost	A	B	BackLink	U-TURN	Cost
1	109	110	-	-	2	103	106	-	-	2
2	110	102	109-110	-	5	107	106	-	-	4
3	109	101	-	-	5	108	106	-	-	5
4	110	104	109-110	-	6	102	103	103-106	120	5
5	102	103	110-102	-	7	104	103	103-106	-	5
최소비용 경로산정	① Overlap된 경로(102-103) 109(A) → 110 → 102 → 103 → 120 → 103 → 106(B) (비용: 10) ← <b>최소비용</b>									
	② 연결경로 109→110→104→103→106 (비용: 11)									
최단경로	109(A) → 110 → 102 → 103 → 120 → 103 → 106(B) (비용: 10)									

### IV. 최단경로알고리즘의 적용 및 수행도 분석

#### 1. 적용방법

수행도 평가대상 네트워크는 링크간의 U-TURN 유형과 기하구조가 다소 복잡하며 좌회전을 금지한 교차로가 많은 강남 일부지역의

실제 네트워크를 대상으로 평가하였다. 특히, 수행도 평가 네트워크에 포함되는 테헤란로의 경우에는 강남역에서부터 삼성역까지 연속적으로 7개의 교차로에서 좌회전금지가 시행중에 있으며, 강남대로의 경우에도 6개의 교차로가, 언주로에서는 5개의 교차로에서 연속적으로 좌회전금지를 시행하고 있다.(1995년 1월 현재)

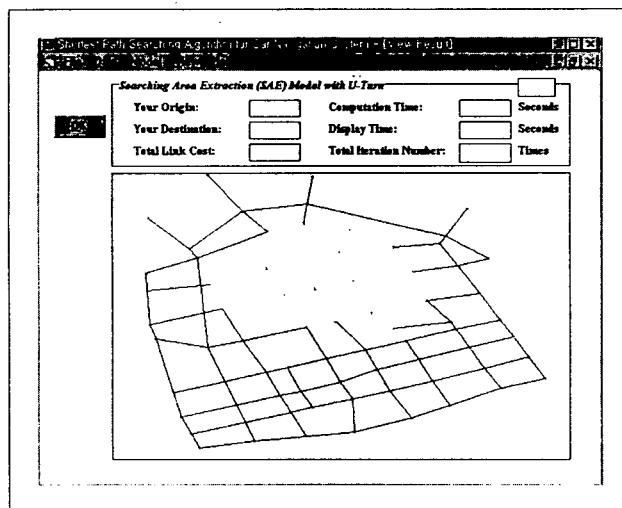
본 모형은 윈도우즈 환경하의 Micorssoft

Visual Basic을 기반으로 구현되었다. 본 연구의 목적은 차량내의 하나의 출발노드로부터 하나의 목적노드(One-To-One)까지의 최단경로를 탐색하는 것이므로 알고리즘상에서는 탐색과정에서 선택된 노드가 목적노드일 경우에는 탐색을 종료한후 최단경로를 디스플레이하게 된다. 그러므로 수행도 평가방법도 어떤 하나의 특정노드와 특정노드간의 최단경로만을 평가대상으로 해서는 정확한 수행도를 평가할 수가 없으며, 대상 네트워크가 70개의 노드와 38개의 U-TURN을 위한 (가상)노드, 링크수 235개의 비교적 작은 네트워크를 대상으로 평가하였으므로 만약 구축된 네트워크상에서 출발노드와 목적노드가 각각 한쪽 끝의 노드로부터 다른쪽 끝부분의 노드로 입력될 경우에는 모형별 비교평가에서 탐색영역제한모형의 경우에는 영역제한을 할 네트워크가 없게되므로 정확한 수행도를 평가하기 어렵다. 따라서 탐색영역제한모형을 비교적 잘 반영할 수 있는 출발노드와 목적노드에 대해서 반복해서 각각 100회씩 링크통행비용을 랜덤화시켜서 최단경로탐색을 시키는 방법으로 486 PC(50MHz)상에서 모델별로 비교

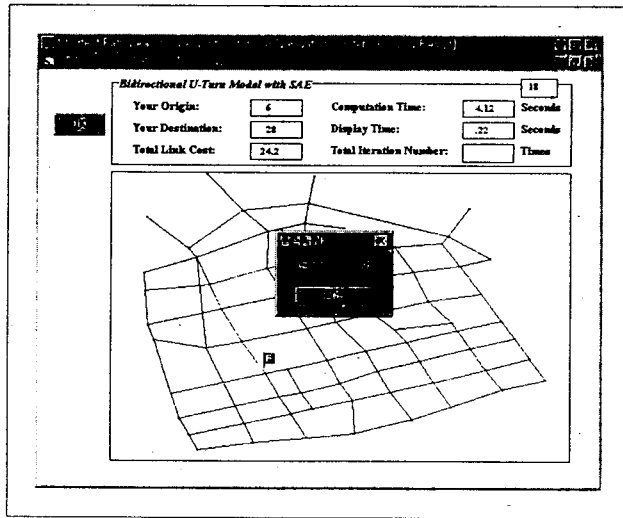
평가하였다. 랜덤화된 링크비용을 사용하는데에는 두 가지 이유가 있다. 첫 번째 가장 큰 이유는 랜덤화된 통행비용의 사용으로 실시간 최단경로의 탐색이 가능하며, 두 번째는 각 링크의 통행비용이 목적노드까지 최단경로의 탐색 계산시간에 영향을 미칠 수도 있기 때문에 링크비용을 랜덤화시킨후 사용하였다. 각 수행시간에는 탐색과정에서 탐색된 링크를 스크린상에 디스플레이하고 목적노드까지의 1회 탐색이 종료되면 전체지도를 다시 스크린에 디스플레이하는 시간까지 포함되었으며, 탐색과정에서의 U-TURN, P-TURN 노드는 <표 4>의 탐색노드수에서 제외시켰다.

## 2. 평가결과

본 평가 프로그램은 하나의 윈도우내에 여러개의 하위 윈도우를 갖는 다중-문서 인터페이스 MDI(Multiple-Document Interface, MDI)로 구성되어 있으며, <그림 4>와 <그림 5>는 그 하위 윈도우중 최단경로 탐색 양식과 탐색된 최단경로를 디스플레이하는 양식이다.



<그림 4> 최단경로탐색과정



〈그림 5〉 최단경로 탐색결과와 디스플레이

본 연구에서 적용한 4가지 모형 각각에 대한 수행도 평가 결과는 다음 〈표 6〉와 같다.

〈표 6〉 수행도 평가결과

모 형	평 가 척 도	100회 수행시간	1회 평균수행시간	탐색노드수
모 형 1 (Forward Dijkstra without SAE)	I	442.0 초	4.42 초	45
	II	522.8 초	5.22 초	50
모 형 2 (Forward Dijkstra with SAE)	I	276.6 초	2.76 초	23
	II	425.3 초	4.25 초	31
모 형 3 (Bidirectional Dijkstra without SAE)	I	310.0 초	3.10 초	20
	II	362.0 초	3.62 초	25
모 형 4 (Bidirectional Dijkstra with SAE)	I	227.7 초	2.27 초	13
	II	330.6 초	3.30 초	18

I : 선릉역 사거리 →안세병원 교차로, II : 강남역 사거리 →청담동 사거리

〈표 6〉은 선릉역 사거리에서부터 안세병원 교차로까지, 강남역 사거리에서 청담동 사거리까지의 최단경로를 탐색한 결과를 보여주고 있다. 〈표 6〉에서 나타나듯이 선릉역 사거리부터 안세병원 교차로까지의 탐색영역제한모형을 적용하여 출발노드와 목적노드 양방향에서 동시에 최단경로를 찾는 모형 4가 1회 평균탐색시간이 2.27초로 가장 짧았으며, 이에 비해 출발노드

에서만 경로탐색을 하는 모형 1은 1회 평균탐색시간이 4.42초로 가장 길었다. 특히, 모형 1에서의 탐색결과 탐색노드수가 45개인데 비해 모형 4에서는 탐색노드수가 13개가 되며, 이때 모형 1에서 나온 최단경로와의 변동은 없었다. 또, 강남역 사거리에서부터 청담동 사거리까지의 최단경로 탐색결과와 모형 4가 1회 평균수행시간이 3.30초로 가장 좋다고 볼 수 있지만, 모형 3

이 모형 2에 비해 수행시간이 짧은 것으로 나타났다. 이 결과는 출발노드로부터 목적노드까지의 거리가 짧고 70개의 네트워크로 한정되어 있기 때문에 모형 3이 우수한 것으로 나타났지만, 서울시 전역의 네트워크로 확장하여 적용할 때에는 탐색영역을 제한하는 모형에서의 탐색시간이 보다 짧게 될 것이 예상된다. 또한 탐색과정에서 정확히 U-TURN과 P-TURN을 반영하여 최단경로를 탐색하는 것으로 나타났으며, 도로 유형, 목적지까지의 거리 등을 고려하여 탐색영역을 설정하는 것에도 별다른 문제점이 없는 것으로 나타났다.

## V. 결론

본 연구에서는 국내 도로의 기하구조적 특성중 중요한 요소인 U-TURN과 P-TURN을 고려하여 4개의 경로 탐색모형을 제안하였으며, 수행도 평가결과는 1회 평균 탐색시간과 알고리즘에서 고려하는 탐색노드의 수를 주요 평가지표로 활용하였다. 본 연구결과를 토대로 제안된 모형에 대한 연구의 결론은 다음과 같이 요약할 수 있다.

첫째, 기존의 Dijkstra 알고리즘을 기반으로 링크를 탐색하도록 함으로서 회전이 연속적으로 금지된 교차로에서도 추가적인 정보(前前노드, 前前前노드,.....)로 인한 메모리의 소요없이 효율적으로 최단경로의 도출이 가능한 새로운 기법을 제시하였다.

둘째, 제안된 4개의 모형들의 수행도는 모형 4, 모형 2, 모형 3, 모형 1의 순서로, 이들중 모형 4가 탐색노드의 수 및 평균 탐색시간 측면에서 매우 우수한 것으로 나타났다.

이는 알고리즘에서 고려하여야 할 탐색영역을 제한함으로써 탐색노드의 수를 최소화함으로써 알고리즘의 수행도가 향상된 것이다.

특히 제안 모형 2와 모형 4가 동일한 탐색제한 모형인데도 불구하고 모형 4가 더 우수한 이유는 모형 2가 단방향 탐색인 반면에 모형 4는 양방향 탐색이기 때문이다. 따라서 탐색영역제한 모형은 알고리즘의 특성상 서울시와 같이 고려될 탐색수가 많은 경우 매우 효과적으로 운영될 수 있으리라 판단된다.

본 연구의 결과를 토대로 본 연구의 제약 및 향후 연구과제를 제시하면 다음과 같다.

첫째, 본 연구에서 제안 모형들중 탐색제한 방법은 출발지-도착지간 거리의 관계식에 기초하여 설정하였다. 탐색영역의 제한방법은 탐색 알고리즘의 최적경로 결정에 있어서 매우 민감한 사항으로 향후 이와 관련한 여러 사례연구가 필수적으로 요구되며, 탐색영역 제한을 위한 관련 연구가 필요하다.

둘째, 본 연구에서는 수행도 평가를 위해 강남 일부지역의 70개 교차로를 대상으로 모형별 비교평가를 하였지만, 보다 정확한 비교평가 및 효율성 측정을 위해서는 서울시 전역의 네트워크로 확장하여 평가하는 것이 보다 의미있을 것이라 생각된다.

셋째, 본 연구에서 제안된 모형들은 기존의 개발 및 구현된 타 모형들과 같이 평균 여행시간을 링크비용으로 대체하여 활용하고 있다. 이 링크비용의 산출은 경로탐색모형이 도시가로망에서 효과적으로 수행되어야 한다는 점에서 도시교통관제시스템(UTCS) 또는 다른 정보원로부터 통행시간을 산정하는 연구가 수행되어야 한다. 예를들면, 교통관제시스템으로 부터 수집가능한 포화도와 교통량, 대기길이 등의 교통상태에 관한 기초자료들을 기반으로 하는 링크통행시간에 대한 추정기법등이 도시부에서의 활용을 위한 다음단계의 연구로서 필요하다고 판단된다.

## 참 고 문 헌

- 【1】 강맹규, 네트워크와 알고리즘, 박영사, 1991
- 【2】 노정현, 남궁성, 도시가로망에 적합한 최단 경로탐색 기법의 개발, 대한국토 도시계획 학회지, 제30권, 5호, pp 153-168, 1995, 10.
- 【3】 임강원, 도시교통계획-이론과 모형, 서울대학교 출판부, 1992, pp.146~151
- 【4】 최기주, U-TURN을 포함한 가로망의 표현 및 최단경로의 구현, 대한교통학회지, 제13권, 3호, pp. 35-51, 1995, 9.
- 【5】 D.E.Boyce, A.Tarko, S.Berka, Y.Zhang, Estimation of Link Travel Times with a Large-scale Network Flow Model for a Dynamic Route Guidance System, 1994 IVHS America Annual Meeting, 1994
- 【6】 Hiroshi Shimoura, Kenji Tenmoku, Development of Elemental Algorithms for Future Dynamic Route Guidance System, 1994 VNIS Conference Proceedings, 1994, pp.321~326
- 【7】 Roy Thomas, Traffic Assignment Techniques, Avebury Technical, 1991