□ 政策硏究 □

U-TURN을 포함한 가로망 표현 및 최단경로의 구현

Network Representation Schemes for U-TURN and Implementation in the Vine-Based Dijkstra Shortest Path Algorithm

최 기 주

(아주대학교 교통공학과 조교수)

目 次

ABSTRACT

This paper reviews both network representation schemes of transportation network and algorithms for finding the shortest path between two points in the network. Two types of network representation schemes for considering U-TURN have been proposed along with some modifications of the vine-based Dijkstra shortest path algorithm. The traditional Sioux-Fall network has been chosen and modified with the introduction of left-turn prohibitions and U-TURNs for the purpose of evaluating the performance of two modified algorithms and network representation schemes (NA1 and NA2). This type of modification in both network representation

scheme (including network data) and algorithm is not only supposed to be needed for route guidance but supposed to contribute to finding more realistic path and estimating more accurate traffic volumes throughout the entire network.

Ⅰ. 서 론

가로망(Network)은 노드(Node)와 링크(Link)로 이루어진 집합에서 이들 상호간의 위상관계(Topology)가 주어진 그래프(Graph)의 특수형태로서 통상 링크의 가중치(Weight)가 부가되어진 점이 그래프와 다르다고 볼 수 있다.

교통에서의 가로망은 이러한 가중치가 주로 길이, 용량, 통행비용 등의 교통관련 속성정보를 담고 있는 가로망이라고 볼 수 있는데, 그간 가로망의 연구분야에서 계속적으로 이론적 연구의 대상이 되어 온 부문은 크게 4분야로 구분할 수 있다 (Dolan 및 Aldous, 1993).

① 존재(Existence)의 문제 : A에서 B로의 경로가 존재하는가?
② 구축(Construction)의 문제 : 그 경로를 어떻게 찾을 것인가?
③ 나열(Enumeration)의 문제 : 몇 개의 경로를 열거할 수 있는가?
④ 최적(Optimization)의 문제 : 나열된 여러 개의 경로 중에서 최적경로는 어느 것인가?

이러한 4가지 분야 중 교통계획과정에서(특히 Traffic Assignment) 꼭 필요한 분야가 바로 ③과 ④라고 볼 수 있다. 이는 최단경로문제(Shortest Path Problem)의 도출을 위한 탐색으로 귀결되어지고, 이에 대한 실질적인 이론적 탐구는 1950년대 말에 이미 완성되었다고 할 수 있다 (Dijkstra, 1957, 및 Moore, 1957).

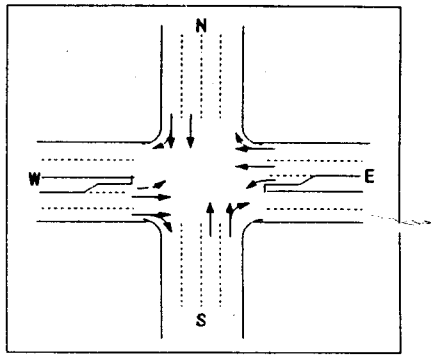
본 고에서는 그간 연구가 되어진 최단경로문제를 분류하고, 정리한 뒤 한국적 상황(광로 중심의 도로망에 U-TURN이 많이 허용되는 점)을 고려하기 위해 U-TURN을 포함한 가로망의 표현 및 최단경로문제의 도출을 위한 알고리즘의 구현을 제시해 보고자 한다. 본고에서 사용된 기존의 알고리즘은 Dijkstra 기반의 덩굴망(Vine) 알고리즘이며, 여러가지의 회전조건(U-TURN 및 좌회전 금지 등)이 가로망의 표현기법에 새로이 추가되어졌고, 이러한 네트워크 변형에 따른 알고리즘의 수정이 이루어졌고, 2가지의 접근방식에 따른 성능비교분석을 실시하였다.

Ⅱ. 문헌조사(Literature Review)

1. 가로망의 표현

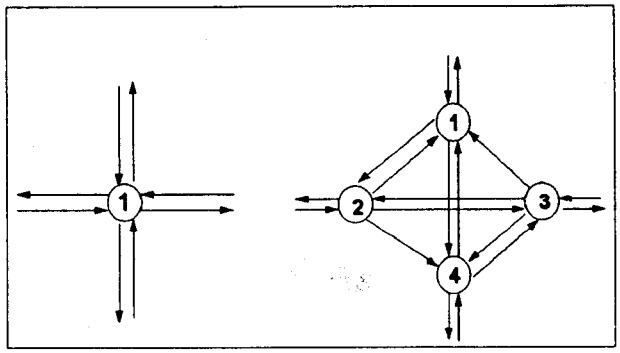
가로망은 객화(여객 및 화물)가 운반되어지는 도로와 이들 도로간의 교차점(Intersection)으로 구성되어지는 교통체계의 3대 필수요건 중의 하나이다. 이러한 실제의 물리적 가로망(그림1)을 표시함에 있어서(특히 통행배정 또는 가로망분석에서) 여러가지의 표현방식이 가능하다.

실제의 가로망 표현방식은 결과의 요구수준 또는 계산상의 부담 등에 따르는 함수로 볼 수 있다. <그림 1>의 실제 가로망은 <그림 2>의 [A]와 같이 교차로를 단일 노드로 처리하는 방안이 있을 수 있고 [B]와 같이 1개의 실제교차점을 4개의 노드로 분리하여 회전의 실제상황을 모델링 작업에 반영할 수도 있다. 즉, [B]의 노드 ④에서 ② (또는 ①에서 ③)으로의 좌회전이 실제로 <그림 1>에서 보는 바와 같이 ③에서 ④ (또는 ②에서 ①)로의 좌회전 보다 여러가지 조건에



<그림 1> 4지 교차로 (Sheffi, 1985)

서 어렵다고 가정하면 (실제 ③에서 ④ 또는 ②에서 ①의 좌회전시에는 좌회전포켓(Left-Turn Pocket)이 존재함으로 인해 좌회전이 용이하다고 봄) 결국 ④에서 ②로의 (또는 ①에서 ③으로의)방향전환은 ④→③→② (또는 ①→②→③)의 과정을 거쳐서 이루어짐으로 인하여 ④→③ 및 ③→②(또는 ①→② 및 ②→③)의 링크 저항을 고려하게 되므로 현실을 표현함에 있어 현실 모사능력을 제고할 수 있는 근거가 된다



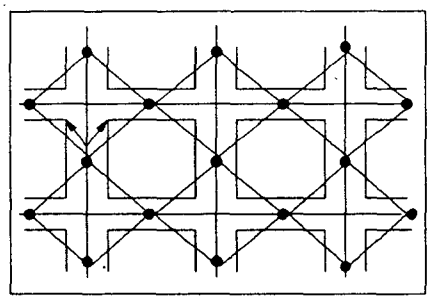
[A] [B]

<그림 2> 그림 1의 가로망 표현

1) [B]의 경우에 있어 S→N으로의 이동은 링크 ③→①을 이용할 수 있고, ③→④→①의 복합 링크를 이용할 수도 있는 것으로 됨. 그런데 ③→④→①은 현실적으로 불가능하므로 이 경우 회전페널티를 부여해야 함. ①→②→③, ②→⑤→④ 및 ④→①→②도 마찬가지임.

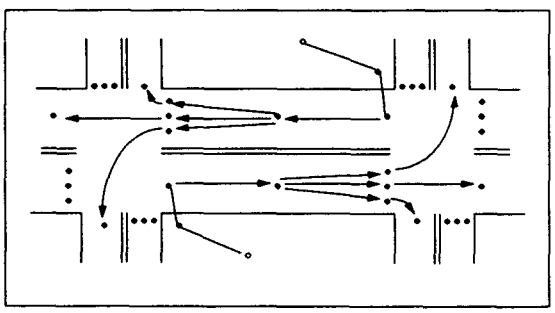
한편, Sheffi(1985)에 의하면 [A]와 같은 가로망 표현은 간단한 반면 2가지 점에서 단점이 있다고 지적하였다. 첫번째는, 교차로에서의 회전을 나타내기가 용이하지 않고 둘째, [B]와 같이 방향에 따른 회전에 대하여 동일한 비용을 갖는다는 것이다. (일반적으로 우회전은 좌회전보다 쉬움) 반면 증가되는 노드와 링크 수 때문에 계산상의 부담은 늘어난다는 것은 자명한 것으로 보여진다.

또다른 방식의 가로망 표현은 몇몇 교통계획 모형에서 쓰이고 있는데 이는 블록의 중간(Mid-block)에 노드를 설정함으로써 <그림 2>에서 보듯이 교차로를 노드로 표현하는 방식과는 다소 차이가 있다. 이를 그림으로 도식화하면 <그림 3>과 같이 나타낼 수 있다. 이는 실제적으로 가로망에 노드나 링크의 수에 있어서의 이득은 없지만 회전교통량의 총량을 방향별로 간파하기 쉬운 장점이 있다.



<그림 3> Mid-Block 노드 체계

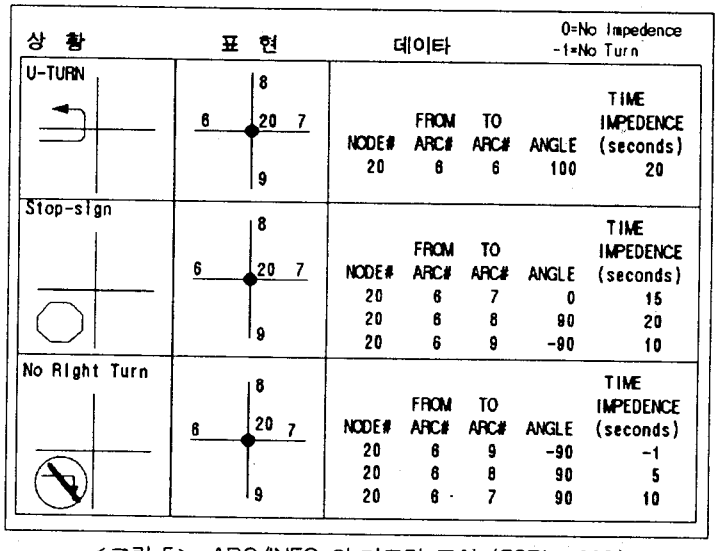
속칭 SDI (Stochastic Dynamic Incremental) 모델로 알려진 교통계획모형에서는 구체적인 링크인식을 위해서 노드체계를 개발하였는데 <그림 4>에 나타난 바와 같이 교차로에서 각 접근로의 우측에 노드를 3개씩 부여하여 방향별 이동을 파악할 수 있게 하였고 이는 <그림 3>의 기본적 미드블럭 노드체계위주의 가로망표현에 보다 정밀한 방향성을 부여한 것이라고 볼 수 있다. 그러나 노드 및 링크의 수가 어떤 면에서는 불필요하게 증가하는 단점이 있다.



<그림 4> SDI모형의 가로망 표현

최근에는 지리정보체계(GIS: Geographic Information System)에서도 가로망분석이 들어오면

서 점, 선, 면으로 이루어지는 GIS구성요소에서 상호간의 관계를 나타내 주는 위상관계가 가로망 모델에서도 기본적으로 반영되어지고 있다. 예를 들면 <그림 5>에서 보듯이 미국의 Environmental Systems Research Institute (ESRI) 사의 ARC/INFO의 가로망 데이타모델에서는 U-TURN, 스톱(Stop-Sign) 및 전면회전금지(No-TURN)의 경우에 있어 노드를 중심으로 하는 가로망 표현 (기존의 대부분은 링크 중심)을 회전의 각도와 함께 표현해 주는 방식을 취하고 있기도 한다.



<그림 5> ARC/INFO 의 가로망 표현 (ESRI, 1992)

2. 최단경로 알고리즘

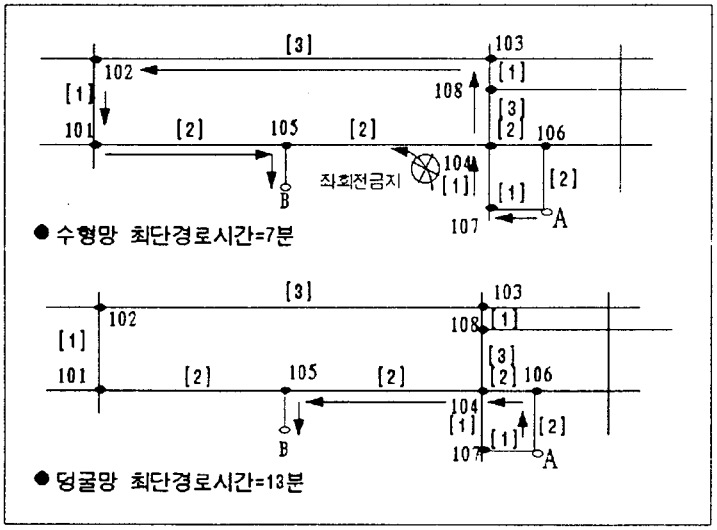
서론에서 언급한 바와 같이 가로망분석에서 기본이 되는 것으로 이미 고전이 되어버린 이 부분에 대한 알고리즘적 기틀은 이미 1950년대 말에 완성이 되었다. 이는 '존재'의 문제 이외에 '구축'을 하고 '나열'을 한 뒤 '최적'을 도출하는 일련의 과정으로 볼 수 있다. 여러가지 분류가 가능하겠지만 이를 크게 구분하면 크게 네가지로 구분할 수 있는데 첫째가, 수형망과 덩굴망(Tree와 Vine) 방식에 따른 구분이고, 두번째가 출발 및 도착 노드의 개수에 따른 구분이며, 세번째는 알고리즘의 접근방식에 따른 구분이며, 네번째는 링크의 가중치가 양수냐 음수냐에 따른 구분으로 나눌 수가 있다. 한편, 교통에서는 링크의 가중치는 통행비용으로서 이것이 음수가 될 경우는 없으므로 앞의 세가지 경우에 대해서만 고려하는 것이 일반적이다.

① 수형망 및 덩굴망 알고리즘

수형망 알고리즘은 교차로에서 회전 방향별 링크 개념, 즉 교차로에서 각 방향으로의 회전을 독립된 비용을 갖는 하나의 링크로 인식하는 개념을 적용하여 무한(Acycle) 네트워크상에서의 기종점간 최단경로를 도출하는 방식을 말한다. 이러한 수형망 알고리즘은 계산이 간단하고 빠르다는 장점이 있지만, 교차점의 노드에 회전금지 등이 설정되어 있는 경우 실패하게 된다(FHWA, 1973).

노드 107에서 시작하여 104를 경유한 후 105로 이동하는데 있다. 좌회전이 금지되어 있다고 가정하고 수형망 알고리즘에 최단경로를 도출한다면,

출발 노드 A에서의 최단경로가 노드 106대신에 이미 107을 경유한 것으로 확정되어 있다. 즉, 107이 최단경로라는 영구 표지가 기재되어 있기 때문에 수형망 알고리즘에서는 (107, 104, 108, 103, 102, 101, 105)의 최단경로를 선택하게 된다. 이에 비해 덩굴망 알고리즘은 모든 노드에 있어서 각 분기(Leg)에 대해 통행시간을 검토하는 것이다. 즉, 노드 104에서 보면 상류측의 노드 107에서 오는 통행에 대해서는 노드 108방향과 노드 107에서 노드 106방향, 또 노드 106에서 오는 경우에는 각각 노드 105, 104, 107 에서의 방향을 모두 검토하는 것이다. 따라서 덩굴망 과정에서는 105, 107, 108 의 경로를 탐색하게 된다.



<그림 6> 수형망경로와 덩굴망경로의 비교 (FHWA, 1973)

따라서 링크의 교차점이 노드로 인식되어 모든 회전방향을 인식하고 가능성을 탐색·비교하는 경우에는 덩굴망 알고리즘이 보다 정확한 해답이 된다. 그러나 연산속도는 일반적으로 덩굴망이 수형망 알고리즘의 2배에 달한다. 이는 덩굴망 알고리즘은 궁극적으로 있을 수 있는 모든 경우의 경로에 대한 통행을 검토하게 됨으로 연산과정이 더욱 복잡해지기 때문이다.

② 출발, 도착 노드의 갯수에 따른 구분

최단경로문제는 기본적으로 하나의 출발 노드에서 네트워크상의 다른 모든 노드로의 최단경로를 산정하는 문제, 즉 한개의 기점/여러개의 종점(Single Origin/ Multiple Destination)의 경우와, 모든 노드에서 모든 노드로의 최단경로를 산정하는 문제 (Multiple Origin/ Multiple Destination)로 구분할 수 있다. 이때, 개념적으로는 하나의 노드에서 다른 하나의 노드로의 최단경로를 찾는 문제는 별도의 문제로 구분할 수 있으나, 이러한 문제에 대한 알고리즘은 기본적으로 하나의 노드에서 다른 모든 노드로의 최단경로를 도출하는 문제와 동일하다. 예를 들면 알고리즘에서는 어떤 노드 i에서 또다른 어떤 노드 j까지의 최단경로를 구하기 위해서는 실제로 그 노드 뿐만 아니라 다른 모든 노드까지의 최단경로도 함께 포함 하고 있는 것이 된다.

③ 알고리즘의 접근방식에 따른 구분

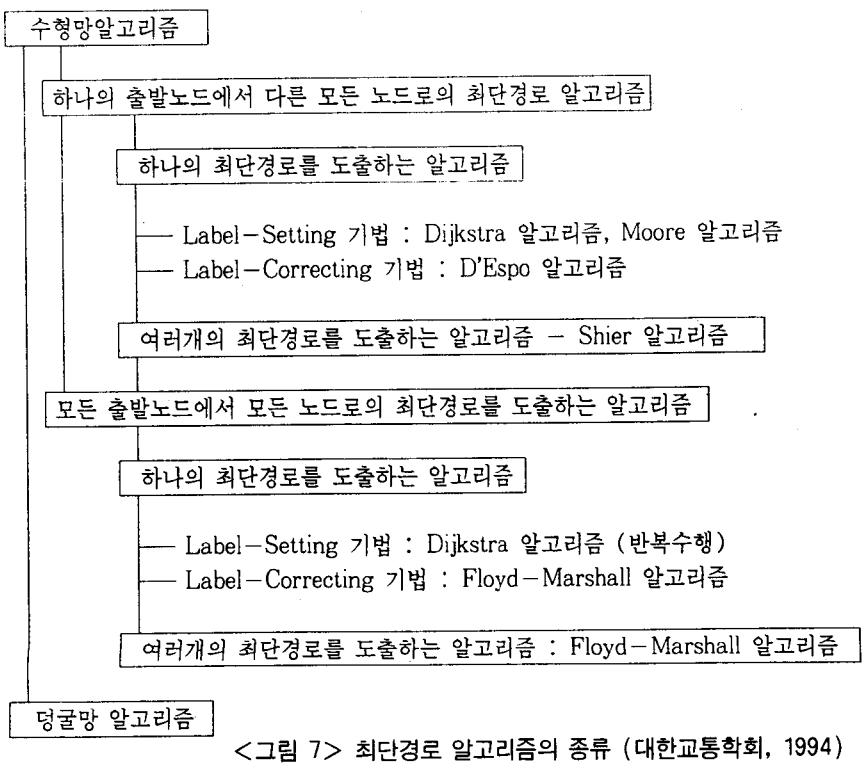
최단경로 알고리즘의 접근방식은 각 노드마다 표지(Lable)를 설정하여, 경로상에서 링크로 연

결된 이전 노드(Predecessor Node)의 번호와 그 때의 비용을 추적하여 최단경로를 찾아내는 레이블링(Lablelling) 알고리즘에 기초하고 있다.

레이블링 알고리즘에서 사용되는 표지에는 영구 표지 (Permanent Lable)와 임시 표지(Temporary Lable)의 두 종류가 사용된다. 최단거리경로를 찾는 경우를 예를 들면 , 출발 노드 1에 대하여 영구 표지는 노드 1로부터 특정 노드 까지 최종적으로 결정된 최단거리를 나타낸다. 즉 영구표지가 표시된 노드는 최단경로를 구성하는 노드가 된다. 임시표지는 노드 1로부터 특정 노드 까지의 어떤 한 경로의 길이를 나타내는데, 이 경로가 최단경로가 아닐 수도 있기 때문에, 즉 추후에 더 작은 값으로 수정될 수 있기 때문에 임시 표지는 실제 최단거리의 상한치(Upper Bound)를 나타낸다. 즉 노드 I의 임시표지를 노드 I까지의 최단거리로 추정치라고 간주할 수 있으며, 이 추정치를 더 이상 줄일 수 없으면 이 추정치가 영구 표지로 확정된다. 영구 표지가 표시된 노드를 영구 노드(Permanent Node)라 한다.

레이블링 알고리즘은 영구 표시(Permanent Lable)를 결정하는 방식에 따라 레이블지정(Lable-Setting) 기법과 레이블수정(Lable-Correcting) 기법으로 구분하게 된다. 레이블지정 기법은 하나의 노드에 대한 고려가 끝나는 반복과정(Iteration) 마다 영구적으로 표지하면서 최단경로를 찾는 방법이며, 레이블수정 기법은 반복과정이 되풀이되는 동안 표지 값을 기록하고 있다가 전체 노드에 대한 고려가 끝난 시점에서 영구 표지로 결정하게 된다.

한편, 지금까지 알려진 최단경로 알고리즘에는 Dijkstra 알고리즘, Moore알고리즘 D'Esopo-Pape알고리즘, Floyd-Marshall알고리즘, 그리고 Shier알고리즘 등이 있다(자세한 것은 강맹규, 1991 참조). 앞서 살펴본 알고리즘의 구분에 의해 위의 알고리즘들을 분류, 정리하면 <그림 7>과 같다.



<그림 7> 최단경로 알고리즘의 종류 (대한교통학회, 1994)

④ Dijkstra 알고리즘을 이용한 덩굴망 알고리즘[2] 의 개요

알고리즘은 출발노드의 초기화 및 반복과정으로 이루어지되 페널티를 고려하는 부분이 덩굴망의 특징이라고 볼 수 있다. 사용되어질 기호를 정리해 보면 다음과 같다.

X : 탐색할 노드의 집합
d(x) : 출발지부터 x노드까지의 최단비용
s 출발노드
t : 다른모든노드
w(x,y) : 링크(x,y)의 비용
P(x) : x의 전노드
PP(x) : x의 전전노드

i) 출발노드에서의 초기화

$x \in N \setminus \{s\}$: $d(x) := w(s,x)$
$P(x) := s$
$PP(x) := \{\}$
$X := \{s\}$

ii) 반복수행 I : 기존의 Dijkstra 알고리즘의 수행

$d(y) := \min\{ d(x) \mid x \notin X \}$
$X := X \cup \{\bar{y}\}$
if $t \in X$ then STOP
else for all $y \notin X$
if $d(y) > d(\bar{y}) + w(\bar{y},y)$ then $d(y) := d(\bar{y}) + w(\bar{y},y)$
$P(y) := \bar{y}$
$PP(y) := P(\bar{y})$

iii) 반복수행 II : 회전페널티의 고려

$d(y) := \min\{ d(x) \mid x \notin X \}$
$X := X \cup \{ y \}$
if $t \in X$ then STOP
else for all $y^* \notin X$
if $d(y^*) > d(\bar{y}) + w(\bar{y},y) + Penalty(\bar{y},y,y^*) + w(y,y^*)$
then $d(y^*) = d(\bar{y}) + w(\bar{y},y) + Penalty(\bar{y},y,y^*) + w(y,y^*)$
$P(y^*) := y$
$PP(y^*) := \bar{y}$

덩굴망 알고리즘 역시 기본적인 최단경로 알고리즘은 필수적이므로 포함되어야 한다. 이중 i)과 ii)는 일반적인 최단경로 알고리즘(여기서는 Dijkstra 알고리즘)을, iii)은 회전 Penalty를 고려하는 루틴을 나타낸 것으로 3부분이 합쳐지면 덩굴망 알고리즘이 골격을 갖추게 된다.

즉, X를 출발지를 포함하는 탐색노드의 집합이고, y X인 노드에 대하여 d(y)를 y노드까지의 최단경로비용이라 하고 $d(\bar{y}) := \min\{d(x) \mid x \notin X\}$로 결정한 후 집합X에 $\bar{y}$를 추가하면 출발지 s부터 $\bar{y}$ 까지의 최단비용은 $d(\bar{y})$이다. 이것은 s에서 $\bar{y}$까지의 모든 경로는 적어도 $(x',y'),x' \in X. y' \notin X$인 하나의 링크를 사용한다는 논리에 따르기 때문이다.

한편 출발지에서 모든 노드까지의 최단비용을 구하기 위해서는 반복적인 루틴을 따르게 된다. 먼저 $d(y)=w(s,y)$로 두며, 만약 w(s,y)가 존재하지 않으면 y까지의 비용은 무한대이다. 계속 X에 더해지는 노드(t)가 모든 노드가 되면 최단경로탐색은 끝나게 된다.

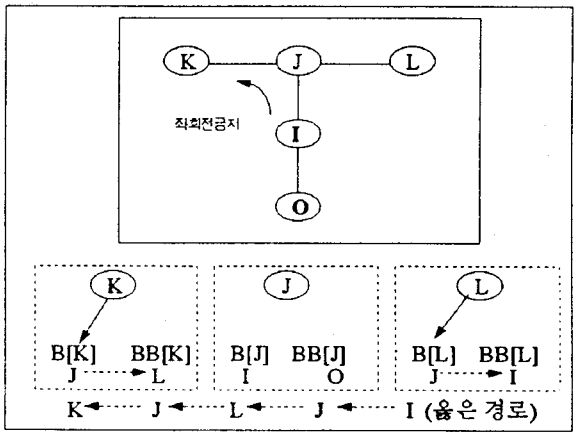
$y \notin X$인 d(y)레이블의 갱신에 있어서 $d(\bar{y}) = \min(d(y) \mid y \in X)$라고 하고 어떤 $y \notin X, y \neq \bar{y}$에 대하여 $d(y) > d(\bar{y}) + w(\bar{y},y)$이면 d(y)는 $d(\bar{y}) + w(\bar{y},y)$로 대체되며 전노드 P(y)는 $\bar{y}$로, 전전노드 PP(y)는 $\bar{y}$의 전노드($P(\bar{y})$)가 된다. $(\bar{y}, y)$와 (y,y^*)인 두링크 사이에 회전페널티$(\bar{y},y,y^*)$를 고려하기 위해서 $d(y^*) > d(\bar{y}) + w(\bar{y},y) + Penalty(\bar{y},y,y^*) + w(y,y^*)$이면 $d(y^*) = d(\bar{y}) + w(\bar{y},y) + Penalty(\bar{y},y,y^*) + w(y,y^*)$이 되고 전노드 $P(y^*)$는 y로, 전전노드 PP(y)는 $\bar{y}$가 된다.

2) 이 부분은 M. Papageorgiou(1991)의 pp461-468을 중심으로 덩굴망 알고리즘 부분을 확장시킨 것이다.

⑤ 덩굴망 알고리즘의 한계

회전페널티를 고려하기 위하여 덩굴망 알고리즘은 일반적으로 수형망 알고리즘의 정보에 더해서 前노드(Back Node)의 前노드인 前前노드(Back Node의 Back Node로서 Back back Node 라고 함)에 대한 정보를 필요로 한다. 회전 페널티에 접하는 노드에서 前노드(Backnode)만으로 최단경로를 찾을 수 없는 경우가 존재하며 이 경우 前前노드에 포함된 정보를 이용하여 목적지 노드까지 도달하게 된다.

<그림 8>에서는 각 노드들의 前노드와 前前노드를 나타낸 것이다. 여기서 K노드의 前노드는 J노드이며 前前노드는 L노드이나, 각 노드들의 前노드 정보만 이용해서 K노드까지의 최단경로를 찾을 경우에는 회전페널티가 포함된 J노드의 前노드가 I노드이므로 회전페널티가 무시된다. 따라서 I→J→K[3]인 틀린 경로를 최단경로로 선택하게 된다. 다행히 회전페널티(P(I,J,K)) 중간노드인 J노드가 K노드까지 도달하는데 틀린 정보를 가지고 있다고 하더라도 K노드의 前前노드가 L이므로 前前노드의 정보를 이용 할 경우 I→J→L→K[4]의 맞는 경로를 선택할 수 있다.



<그림 8> 각 노드의 전노드와 전전노드 (I)

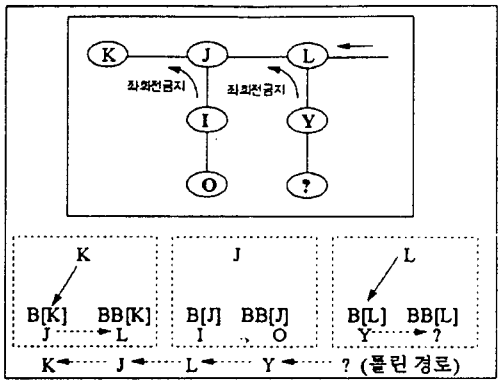
그러나 <그림 9>에서 처럼 2개의 회전페널티가 인접노드에 각각 붙어 있는 경우에는 기존의 덩굴망 알고리즘에 한계가 발생한다. J노드 및 L노드가 각각 K노드의 前 및 前前노드인데 L노드에서 회전페널티가 존재하므로 <그림 8>의 J노드처럼 K노드까지 도달하는데 틀린 정보(L노드의 前노드 Y와 前前노드 ?가 K노드까지 도달하는데 틀린 정보)를 가지고 있는 경우가 발생할 수 있다. 이때는 前노드개념을 확장하여[5] 문제를 해결할 수 있으나 프로그램의 구성에 많

3)K노드의 전노드는 J : K <- J
J노드의 전노드는 I : K <- J <- I

4)K노드의 전노드는 J, 전전노드는 L : K <- J <- L
L노드의 전노드는 J, 전전노드는 I : K <- J <- L <- J <- I

5)前前前前 · · · · 노드

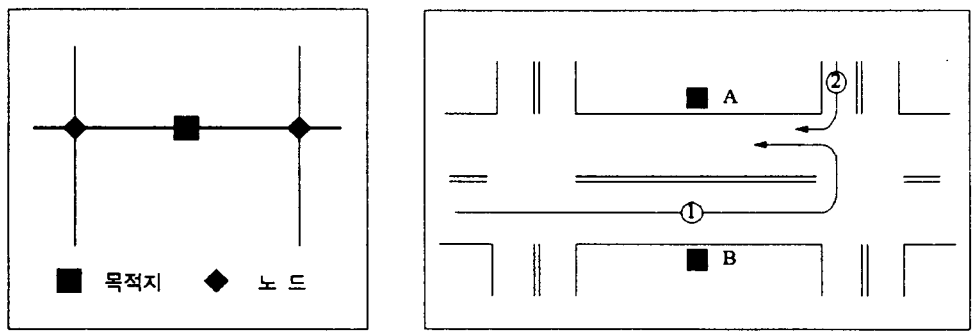
은 어려움이 존재(아직 알고리즘상으로 시도된 적이 없으나 이론적으로 그러함)하며 최단경로를 찾는데 속도가 현저히 감소하고 컴퓨터의 메모리를 많이 요구하게 된다. 예를들어 3000개의 노드를 갖는 네트워크에서 최대로 연속연결된 페널티 구간이 10개인 경우 필요한 메모리의 요구량은 페널티구간에 따라 선형적으로 증가하며 10개의 페널티를 동시에 고려해야 하는 부담이 있다[6].



<그림 9> 각 노드의 전노드와 전전노드 (Ⅱ)

Ⅲ. U-TURN을 위한 가로망의 표현 및 알고리즘 보완

현재 한국의 도로실정은 일본이나 미국에 비해 광로가 많이 존재하는 편이고 어떤 도로는 편도 10차선에 육박하는 것도 존재한다. 한편 이러한 광로를 효율적으로 이용하는 이차의 방안이 또한 U-TURN이 아닌가 한다. 현실적으로 U-TURN은 그림 10에서 나타난바와 같이 노폭이 조금 좁은 편도 3차선 또는 2차선에서는 우회전

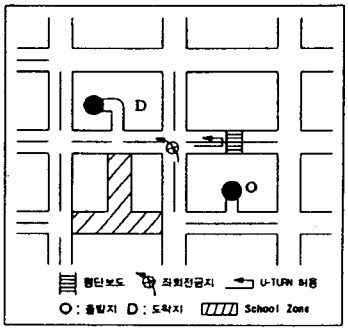


<그림 10> U-TURN 허용 단일교차로

6) 이론적으로 10개의 좌회전 금지 Penalty가 동시에 붙어 있는 경우 소요 Memory
Memory = Maxnode * Byte 수 * 10 = 3,000 * 2byte * 10 = 60,000 byte
단. Integer를 2byte로 보고 Maxnode는 3000임. 회전금지가 하나도 없는 경우는 3,000 * 2(byte)로서 6,000 byte면 충분 함.

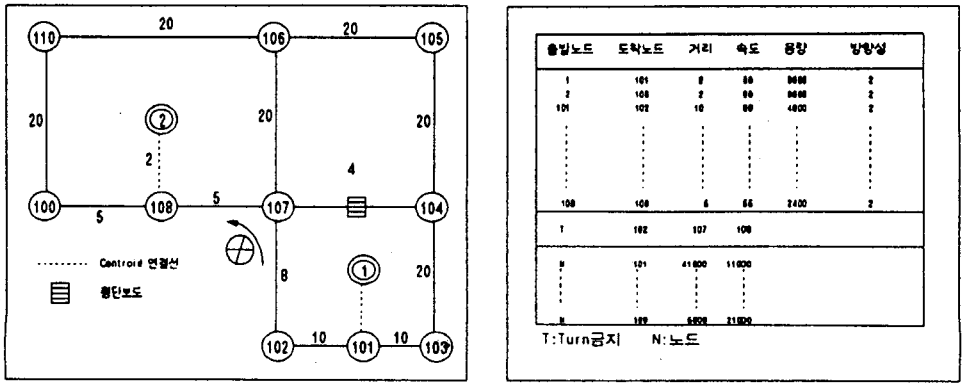
하는 (Movement ②)차량과 U-TURN을 하는 차량 (Movement ①)와 충돌(Conflict)이 생길 수 있지만 좌회전이 설치된 도로 실정을 감안하면 때로는 매우 목적지에 접근함에 있어 유용한 점도 있다.

<그림 10>에서와 같은 가로망을 [A]와 같이 표현 했을 때, 미드블럭상의 목적지가 실제 도로의 우측 또는 좌측이냐에 관계없이 경로안내를 끝내게 된다. 한편 첨단도로교통체계 (Intelligent Transportation Systems: ITS)의 첨단여행자정보체계(Advanced Traveler's Information System: ATIS)와 경로안내기능의 일상화는 <그림 10>의 [B]에서 보듯이 최종목적지가 A일 경우 ①번의 이동은 B가 아닌 A까지 도달해야 할 것을 요구하고 있으며, 이를 위해서는 U-TURN을 실시해야 하여야만 진정한 접근(Door-To-Door) 서비스를 받을 수 있는 것이 된다.



<그림 11> 테스트용 가로망

본 고에서는 테스트를 위해 <그림 11>과 같은 현실의 가로망을 대상으로 가로망표현 및 알고리즘을 개발하고자 한다. <그림 11>에서는 아동보호구역(School Zone)이 특수한 시간대에 존재한다고 보고 또한 횡단보도 역시 존재하며 모든 링크는 양방향이되 횡단보도 앞에서의 U-TURN 및 좌회전 금지가 존재하는 구간이다. 이것을 컴퓨터에서 구현하기 위한 노드, 링크 기반의 가로망 위상관계 및 기타의 가로망 정보를 담고 있는 소위 가로망자료(Network Data)의 일반적인 형태는 TRANPLAN 교통패키지의 경우 <그림 12>에 나타나게 된다.

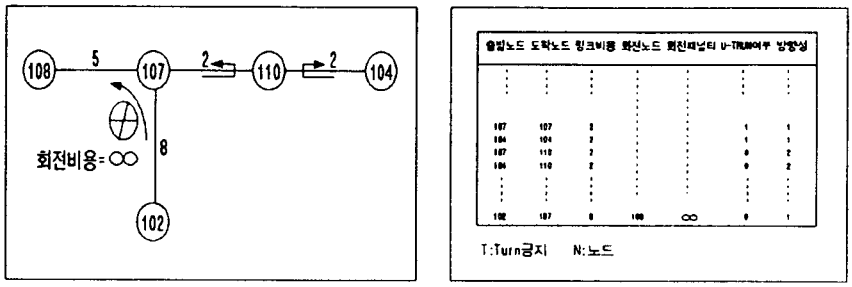


<그림 12> 가로망의 표현 및 가로망 데이타

그러나 현실적으로 104와 107사이에서의 U-TURN이 허용되는 경우 이를 나타낼 수 있는 장치가 없는 것이 현실이다. 이를 보완해 줄 수 있는 2가지 가로망 표현방법 및 알고리즘을 제시해 보고자 한다.

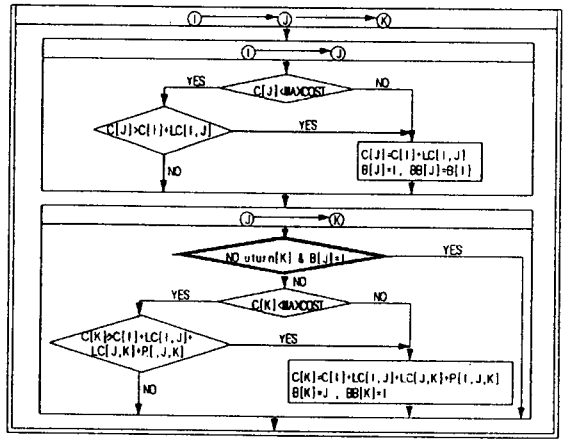
1. 가로망표현 및 알고리즘 1(NA1: Network Representation 및 Algorithm 1)

NA1은 노드 104와 107 사이에 횡단보도의 존재를 알릴 수 있는 추가 노드를 설치함과 동시에 가로망 데이타에서 회전의 가능성을 그림 13과 같이 알려주는 방안이다. 그림 12에서의 가로망 데이타 중에서 104와 107 사이만을 확대해 보면 다음과 같다.



<그림 13> U-TURN 부여시 가로망의 변화 (1)

이는 노드 3개로써 링크 4개를 표현함과 동시에 2개의 U-TURN 링크를 포함하고 있으며 알고리즘은 기존의 Dijkstra 기반의 덩굴망 알고리즘으로 처리가 가능하되 가로망데이타 부분에서 U-TURN의 여부를 알리는 신호가 알고리즘에서 필요하다. (1이면 U-TURN가능, 0이면 불가) 한편 이 부분의 알고리즘 수정 및 실현은 플로챠트 1 (그림 14)과 같고 구현은 C++언어로 수행되어졌다.



C[I] : I 노드 까지의 총비용 LC[I,J] : I→J 링크의 비용

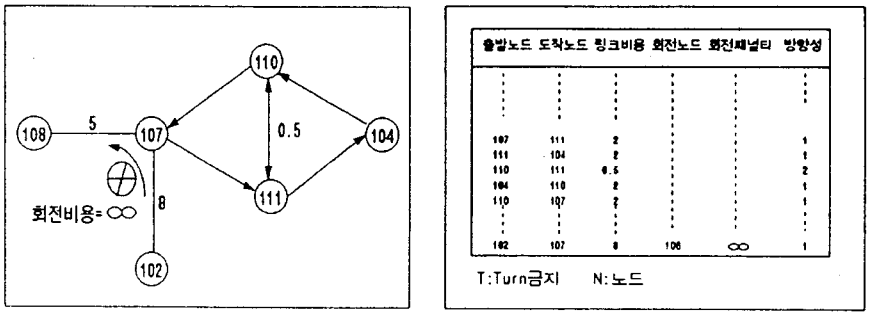
B[I] : I 노드의 back 노드 BB[I] : I 노드의 back back 노드

P[I,J,K] : I→J→K 회전의 페널티

<그림 14> NA1의 경우 플로챠트

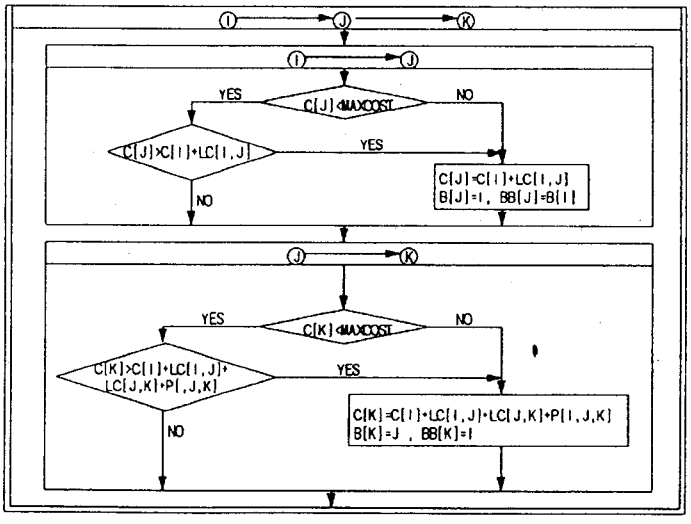
2. 가로망표현 및 알고리즘 2(NA2: Network Representation 및 Algorithm 2)

이는 노드 104와 107구간을 일방향 링크를 확대하면서 횡단보도 구간을 2개의 노드로서 대치하면서 횡단보도에 해당하는 링크를 양방향 링크로 처리해 주는 방식이다[7]. 이를 그림으로 나타내면 <그림 15>와 같다.



<그림 15> U-TURN부여시 가로망 변화 (2)

가로망 데이타에서 보듯이 NA1 에서 보는 바와 같은 U-TURN의 여부는 누락되어 있다. 이는 가로망의 표현을 일방향으로 되도록 처리함으로서 가로망 데이타 한개의 데이타항목(Item)이 생략될 수 있으나 노드 및 링크의 수가 증가하는 단점이 있을 수 있고 이는 가로망의 크기가 커지면 계산 효과에 영향을 줄 수 있을 것이다 한편 NA2의 플로챠트를 그림으로 나타내면 <그림 16>과 같고 C++ 언어로 구현되었다.



<그림 16> NA2의 경우 플로챠트

7) 물론 횡단보도에서 한쪽만이 U-TURN이 허용되는 경우는 일방향이 됨.

Ⅳ. NA1 및 NA2의 비교 분석

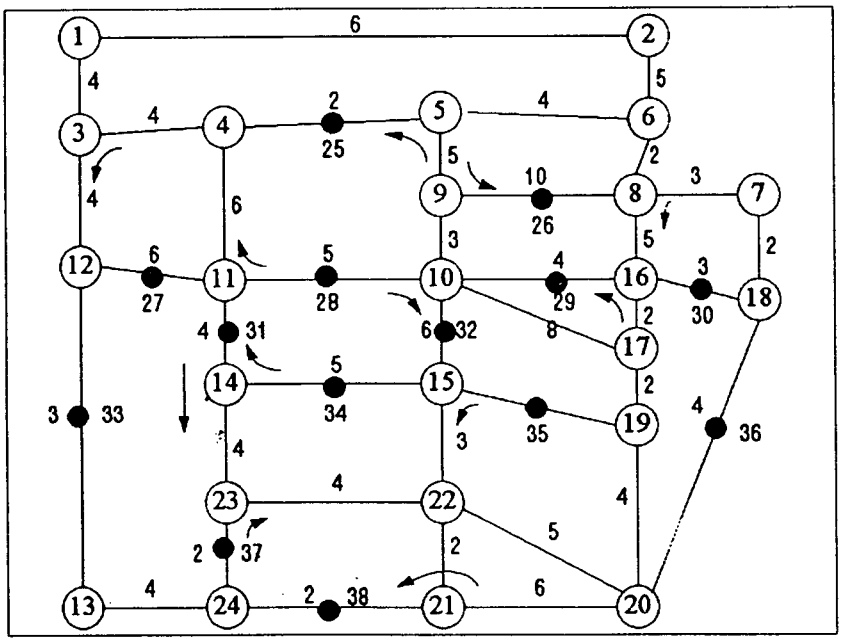
NA1과 NA2에서 보듯이 NA2는 이해하기가 쉽고, 표현이 보다 직접적이라고 볼 수 있다. 그러나 가로망이 커짐에 따라서 노드 및 일방향 링크의 수가 증가함으로서 가로망 데이타의 총량(총 레코드의 수)이 다소 증가한다고 볼 수 있다. 본 고에서는 두 가지 접근방법의 성능을 평가하기 위해서 교통분야에서 이미 잘 알려진 Sioux Fall Network을 가지고 NA1과 NA2를 비교해 보기로 한다.

원래 Sioux Fall Network은 LeBlanc(1973) 등이 가로망 설계 (Network Design) 문제의 테스트를 위해 사용한 바 있는 고전적 가로망으로서 노드 수는 24개이며, 링크 수는 76개(양방향고려)이나, 본 고에서는 회전페널티를 13개, U-TURN 지점을 14개 도입하였다. 이러한 결과는 위에서 본바와 같이 NA1의 경우 노드가 14개, 링크 수가 28개 그리고 NA2의 경우 노드가 2배 이상인 28개, 링크수가 56개 증가하였다.

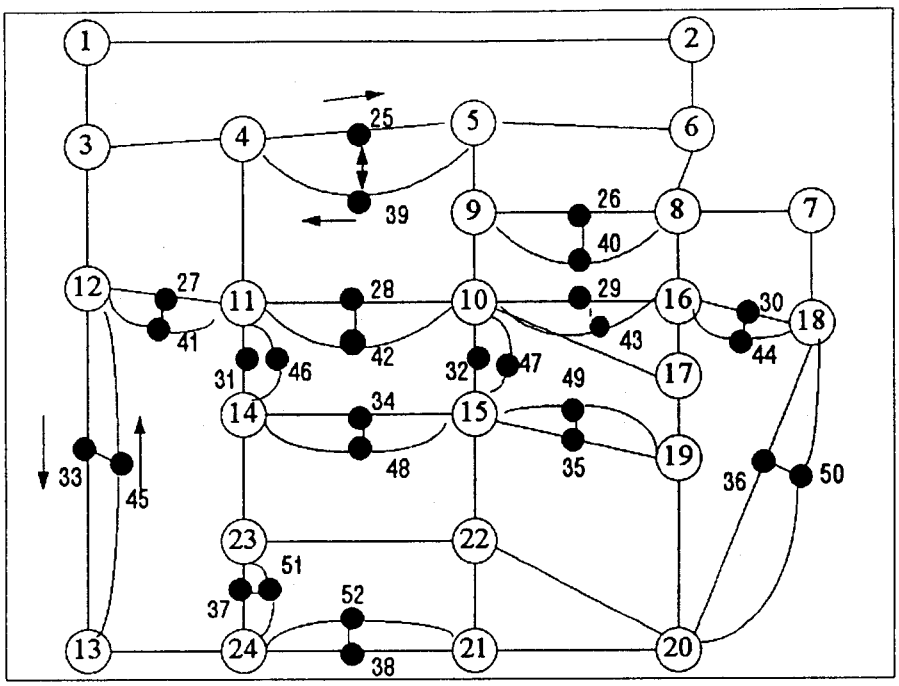
<표 1> NA1과 NA2 의 특성 비교

구 분	NA 1	NA 2
노드(개)	38	52
링크(개)	104	132
회전페널티(지점)	13	13
U-TURN(지점)	14	14

NA1과 NA2에 의해 수정되어진 가로망 표현은 그림 17에 나타난 바와 같고 가로망간의 세부적 사항을 정리하면 <표 1>과 같다.



NA 1



NA 2

○ 노드 ● U-TURN → 회전 또는 진행금지

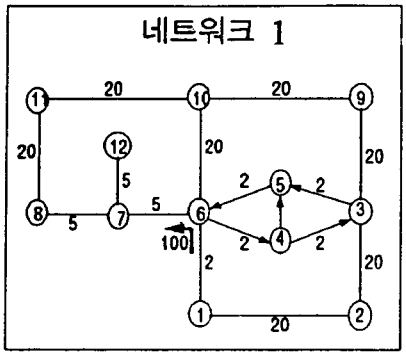
<그림 17> Sioux Fall Network의 NA1 및 NA2에 의한 표현

한편, 가로망의 표현과 알고리즘 성능을 비교하기 위해 486 DX2 50Mhz(286이나, 386이면 더욱 분명한 비교가 되겠지만)컴퓨터에서 수행시간을 Check 하였다. 가로망이 작은 관계로 수행횟수는 100회를 기준으로 하였다[8]. 사용되어진 컴파일러는 Borland C++ Version 3.1이고, 그 결과는 <표 2>에 정리되어 있다.

<표 2> NA1과 NA2간의 성능 비교

성능비교대안	수 행 시 간	
	NA1	NA2
동일노드(NA1 38개, NA2 노드 1에서 38) 비교	9분 4초	11분 9초
최대노드(NA1 38개, NA2 52개) 비교	9분 4초	20분 54초
원래 Sioux Fall 가로망(24개 노드, 72개 링크)	2분 25초	

8) 현저한 차이를 보기 위해서 100회를 수행하였으나, 실제로는 U-TURN들이 모두 고려된 서울시 가로망 등이 더욱 의미 있을 것이나, 가로망의 준비가 여의치 않아 동일한 수행을 여러번 실시하였음.



입력 1

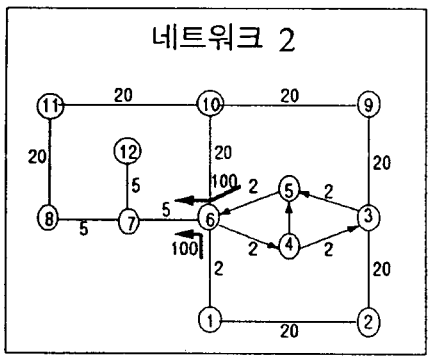
inode	jnode	cost	to penalty-node	penalty
1	2	20	0	0
1	6	2	7	100
2	3	20	0	0
3	5	2	0	0
3	9	20	0	0
4	3	2	0	0
4	5	1	0	0
5	6	2	0	0
6	4	2	0	0
6	7	5	0	0
6	10	20	0	0
7	12	5	0	0
8	7	5	0	0
9	10	20	0	0
10	11	20	0	0
11	8	20	0	0

출력 1

NODE	BACKNODE	COST	BBACKNODE	CCCOST
1	0	0	0	0
2	1	20	0	20
3	4	6	6	6
4	6	4	1	4
5	4	5	6	5
6	1	2	0	2
7	6	7	11	12
8	11	62	10	62
9	3	26	4	26
10	6	22	1	22
11	10	42	6	42
12	7	12	8	17

12 ← 7 ← 6 ← 5 ← 4 ← 6 ← 1

<그림 18> NA1에 의한 입력 및 출력결과



입력 2

inode	jnode	cost	to penalty-node	penalty
1	2	20	0	0
1	6	2	7	100
2	3	20	0	0
3	5	2	0	0
3	9	20	0	0
4	3	2	0	0
4	5	1	0	0
5	6	2	7	100
6	4	2	0	0
6	7	5	0	0
6	10	20	0	0
7	12	5	0	0
8	7	5	0	0
9	10	20	0	0
10	11	20	0	0
11	8	20	0	0

출력 2

NODE	BACKNODE	COST	BBACKNODE	CCCOST
1	0	0	0	0
2	1	20	0	20
3	4	6	6	6
4	6	4	1	4
5	4	5	6	5
6	1	2	0	2
7	8	7	5	67
8	11	62	10	62
9	3	26	4	26
10	6	22	1	22
11	10	42	6	42
12	7	12	6	72

12 ← 7 ← 8 ← 11 ← 10 ← 6 ← 1

<그림 19> NA2에 의한 입력 및 출력결과

최대 노드 수를 구성하는 측면과 같은 노드 수로 비교할 때도 NA1이 우수한 것으로 파악되었다. 이는 NA1에 의한 가로망의 표현이 전체적으로 탐색 노드 수를 감소시킴으로써 수행시간을 감소시키는 것으로 보이기 때문이다. 한편, 원래의 네트워크 자료를 가지고 동일한 횟수의 시행을 한결과 2분 25초로서(덩굴망 알고리즘 사용) NA1과 NA2 보다는 현저히 적은 시간이 소요된 것을 보면 U-TURN의 고려는 계산에 있어서 더 많은 시간이 소요됨을 알 수 있다.

한편 NA2에 의한 테스트 가로망의 결과를 잠시 소개하면, 그림 18, 그림 19와 같다. 그림 19는 노드 ⑥을 중심으로 2개의 회전페널티가 존재할 경우이며, 각각의 최적경로가 제시되어 있다(이 경우는 어쩔 수 없이 우회하는 경우가 발생함).

V. 결 론

본고는 U-TURN 을 고려하기 위한 가로망의 표현방안을 제시하였고, 이를 기존의 Dijkstra 기반의 덩굴망 최적경로 알고리즘에 투영하여 실현함을 소개하고 있다. 현실적으로 U-TURN이 활성화되어 있는 우리의 경우 소형차의 U-TURN은 빈번히 발생하고 있는 점을 감안할 때 본 고에서 제시된 U-TURN을 고려한 최단경로 알고리즘의 보완은 통행배정을 위시한 교통계획 과정에서 보다 정확한 수요의 추정은 물론 물론 교통축 또는 국부지역의 교통분석을 위한 모의실험모형(Micro Simulation Model) 패키지 등에서도 바람직한 영향분석을 위해서는 반드시 고려되어야 할 것으로 보인다. 즉 최적경로의 탐색은 그 이후의 모든 과정에 영향을 미치게 되므로 이의 현실적인 고려가 정확한 교통수요를 도출하기 위한 전제조건이 되기 때문이며, 이는 특히 작은 회전반경을 필요로 하는 승용차의 통행특성을 반영시킬 수 있는 기반이 된다고 볼 수 있다.

한편, 전술한 바와 같이 첨단교통체계의 ATIS가 실현 될 경우 접근성이 강조되는 경로안내(Route Guidance)를 제공 받는 경우 목적지에 다다르면서 이러한 U-TURN을 확실히 고려해야지만 올바른 최종적인 경로안내를 받을 수 있다는 점에서도 이부분의 중요성은 부각 되어 진다. 이외에도 로터리(서울의 경우 신촌, 영등포, 공덕동 등)의 처리문제도 쉽게 처리가 가능하므로, 경로안내에서의 이러한 가로망의 표현은 중요하다고 판단되어지며 향후에도 알고리즘의 효율성 측면에서의 개선이 요망되어 진다고 볼 수 있다.

참 고 문 헌

1. 강맹규(1991), 네트워크와 알고리즘, 박영사.
2. 대한교통학회(홍익대학교 과학기술연구소)(1994), 도로안내체계개발, 교통개발연구원.
3. A. Dolan & J. Aldous(1993), Network and Algorithms, Wiley.
4. E. Moore(1957), The Shortest Path Through A Maze, Proc. Of The Int'l Symposium On The Theory of Switching, pp285-292.
5. ESRI(1992), ARC/INFO User's Guide 6.1, Network Analysis.
6. E. W. Dijkstra(1957), Note On Two Problems In Connection With Graphs, Numerical Mathematics, 1, pp 269-271.
7. L. LeBlanc(1973), Mathematical Programming Algorithm For Large Scale Network Equilibrium And Network Design Problems, Ph.D Thesis, Northwestern Univ. Evinston, Illinois.
8. R. Thomas(1991), Traffic Assignment Technique, Avebury Technical(Academic

Publishing Group).
9. Y. Sheffi(1985), Urban Transportation Network, Prentice-Hall.
10. FHWA(1973), Traffic Assignment.
11. M. Papageorgiou (1991), Concise Encyclopedia of Traffic & Transportation Systems, Pergamon Press.