

■ 論 文 ■

ATIS를 위한 수정형 덩굴망 최단경로 탐색 알고리즘의 개발

Development of a Modified Vine Building Shortest Path Algorithm for ATIS

김 익 기

(한양대학교 교통공학과 부교수)

목 차

- | | |
|----------------------------|-------------------|
| I. 서론 | IV. 단순 교통망에 의한 예제 |
| II. 기존의 수형망 및 덩굴망 알고리즘의 한계 | V. 결론 |
| III. 수정형 덩굴망 알고리즘 | 참고문헌 |

요 약

건설된 도로를 효율적으로 이용하고, 통행자의 편리성을 향상시키기 위해 첨단 여행자 정보체계(ATIS)를 활용할 수 있다. ATIS 체계하에서 노선정보를 통행자에게 제공하기 위해서는 교차로에서의 회전으로 인한 지체를 정확하게 반영할 수 있는 효율적인 최단경로 탐색 알고리즘이 필요하다. 하지만 기존의 최단경로탐색 알고리즘은 좌회전 금지, U-turn, P-turn 등 교차로에서의 회전으로 인한 지체를 정확히 반영 못한다는 단점을 갖고 있다. 그러므로 본 논문에서는 이러한 단점을 극복하기 위해 수정형 덩굴망 알고리즘을 개발하였다. 수정형 덩굴망 알고리즘은 노드표지(node labelling) 방법에 있어서는 기존의 덩굴망 알고리즘의 노드표지 방법과 개념적으로 동일하여 이용상의 편리성을 갖도록 하였으며, 최단경로 탐색기능에 있어서는 링크탐색알고리즘(혹은 링크표지기법)이 갖고 있는 장점을 다 반영할 수 있는 기법으로 개발하였다.

수정형 덩굴망 알고리즘은 노드표지에 있어 특정 노드로 유입하는 방향에 따라 노드표지를 별도로 기록하였다. 따라서 교차로에서의 좌회전, 우회전 및 직진은 물론 U-turn의 경우에도 추가적인 절차 없이 현실적인 최단경로를 탐색할 수 있도록 하였다. 또한 본 논문은 최단경로의 역추적 방법을 개선하여 좌회전 금지, U-turn, P-turn 및 기타 회전에 의한 지체 등을 각 교차로마다 정확히 반영함으로써 비합리적인 최단경로가 추적되는 것을 근본적으로 차단하도록 하였다. 따라서 본 연구에서 개발한 수정형 덩굴망 최단경로탐색 알고리즘은 교차로에서의 회전지체 및 회전금지를 현실적으로 잘 반영함으로써 정확한 노선정보를 요구하는 ATIS체계를 분석하는데 유용하게 활용될 수 있는 기법이다.

I. 서론

최근에 지능형 교통체계(Intelligent Transportation System:ITS)에 대한 연구가 세계적으로 활발하여지면서 ITS 분야의 하나인 첨단 여행자 정보체계(Advanced Traveler Information System:ATIS)에 대한 연구도 함께 활발하여졌다. ATIS체계하에서 효율적인 노선정보의 안내 및 유도를 수행하기 위하여 최단경로탐색 알고리즘은 필수적인 기법이다. 특히 노선정보체계를 도시에서 운영하기 위해서는 좌회전 금지, U-turn 및 P-turn 등 각 교차로에서의 회전으로 인한 지체를 정확하게 반영한 현실적인 최적노선 정보가 통행자에게 제공되어야 한다. 따라서 각 링크의 통행시간이 정확하다고 가정된 상태에서 현실적인 최적노선을 효율적으로 찾을 수 있는 최단경로탐색 알고리즘에 대한 연구는 장래 ATIS체계 운영을 위해 매우 필요하다.

또한 정통적인 교통수요분석 모형의 노선배정분석에서 가장 핵심적 내용 중의 하나가 최단경로탐색 알고리즘이다. 노선배정기법으로써 Wardrop의 제1법칙을 만족시키는 사용자 균형상태의 교통패턴을 찾기 위해 Frank-Wolf 알고리즘이나 기타 직관적 알고리즘(heuristic algorithm)이 활용된다. 이때 각 알고리즘의 기초가 되는 All-or-Nothing assignment는 절대적으로 최단경로탐색 알고리즘에 의존하고 있다. 그러므로 최단경로탐색 알고리즘이 현실적인 노선을 합리적으로 찾아낼 때 정적 교통망분석(static network analysis)의 정확성이 확보될 수 있다. 더욱이 이와 같은 노선배정기법은 ATIS 체계를 분석하기 위한 기초적 분석기법이기에 때문에 노선배정기법의 기초가 되는 최단경로탐색 알고리즘은 교통수요분석 뿐 아니라 ATIS 분석을 위해서도 중요한 핵심기술인 것이다.

하지만 기존에 개발된 최단경로탐색 알고리즘들은 계산과정이 복잡하거나 혹은 현실적 최단경로와는 다른 비현실적 노선을 찾게되는 단점을 갖고 있다. 다시 말해서 Dijkstra algorithm과 같은 표지 알고리즘(labeling algorithm)에 의한 정통적 수형망 알고리즘(tree building algorithm)은 좌회전 및 우회전, U-turn, P-turn 등의 회전에 의한 지체를 최단경로 탐색과정에서 반영하지 못하고 있다. 이러한 단점을 극복하기

위해 개발된 덩굴망 알고리즘(vine building algorithm)도 연속적 회전 및 U-turn, P-turn을 반영하지 못하고 있다. 그러므로 본 연구에서는 기존의 표지 알고리즘의 하나인 Dijkstra algorithm을 적용하면서도 덩굴망 알고리즘이 갖고 있는 단점을 극복할 수 있는 최단경로탐색 알고리즘을 개발하고자 한다.

본 연구에서는 교차로에서의 회전을 현실적으로 정확히 반영하기 위해 노정현과 남궁성(1995) 및 이승환, 최기주와 김원길(1996)이 제시한 링크탐색 알고리즘과 유사한 기본개념을 갖는 수정형 덩굴망 알고리즘을 개발하였다. 새로이 개발된 수정형 덩굴망 알고리즘은 기존의 연구와는 달리 기존의 덩굴망 알고리즘의 노드표지 방법과 동일한 노드의 표지방법을 적용하면서도 현실적인 회전에 의한 지체를 반영할 수 있도록 하였다. 수정형 덩굴망 알고리즘은 노드표지에 있어 특정 노드로 유입하는 방향에 따라 노드표지를 별도로 기록하여 교차로에서의 좌회전, 우회전 및 직진은 물론 U-turn의 경우에도 추가적인 절차 없이 현실적인 최단경로를 탐색할 수 있도록 하였다. 또한 분석에서 최종적으로 얻은 노드표지를 이용하여 최단경로를 역추적 하는 방법을 개선함으로써 좌회전 금지, U-turn, P-turn 및 기타 회전에 의한 지체 등을 각 교차로마다 정확히 반영하도록 하여 비합리적인 최단경로가 추적되는 것을 근본적으로 차단하고 있다. 따라서 본 연구에서 개발한 수정형 덩굴망 알고리즘은 교차로의 회전지체 및 회전금지를 현실적으로 반영하도록 한 최단경로탐색 알고리즘으로써 정확한 노선정보를 요구하는 ATIS체계의 분석에 유용하게 활용될 수 있는 기법이라고 고려된다.

II. 기존 수형망 및 덩굴망 알고리즘의 한계

기존에 개발된 최단경로탐색 알고리즘에 대해서는 Larson and Odoni(1981), Papageorgiou(1991), Thomas(1991), 강맹규(1991), Ahuja, Magnanti and Orlin(1993) 등의 서적에 정리가 잘 되어 있으며, 특히 교통분야에 적용되는 최단경로탐색에 대해서는 Thomas(1991)에서 이론 및 알고리즘에 대해서 더욱 구체적으로 설명되어 있다. 교통분야에 가장 보편적으로 많이 응용되고 있

는 Dijkstra의 알고리즘, Moore의 알고리즘은 기본적으로 수형망 알고리즘으로써 표지 알고리즘(labeling algorithm) 형식으로 최단경로를 탐색하고 있다. 이와 같은 최단경로탐색 알고리즘은 다음과 같은 정수계획법(integer programming)의 정식(formulation)을 직관적 기법(heuristic method)으로 최적해를 찾는 기법이라고 할 수 있다.

$$\begin{aligned} \text{Min. } & \sum_i \sum_j C_{ij} X_{ij} \\ \text{S. T. } & \sum_i X_{im} - \sum_j X_{nj} \begin{cases} = 1 \text{ 만일 } m=d \\ = 0 \text{ 만일 } m \neq d \text{ and } m \neq r, \forall m \\ = -1 \text{ 만일 } m=r \end{cases} \end{aligned}$$

여기서 C_{ij} : 링크 ij 의 통행비용

d : 도착노드

r : 출발노드

X_{ij} : 1 만일 링크 ij 가 최단경로에 포함되어 있으면

: 0 만일 아니면

위 식은 목적함수에서 볼 수 있듯이 출발지점에서 도착지점까지의 경로에 포함된 링크의 통행비용의 합이 최소화하는 최단경로를 찾으려 하는 문제이다. 이것은 링크에서 링크로 연결되는 노드에서의 비용이 존재하지 않는다는 가정이 내면적으로 내포되어 있는 것이다. 다른 방법의 해석으로써는 링크와 링크의 연결지점인 노드에서의 비용이 링크 비용에 포함되어 있으며, 방향에 상관없이 링크에서 링크로 연결되는 비용이 동일하다는 가정을 하고 있다고 볼 수 있다. 이와 같은 가정은 교통분야의 실질적 교통망 분석에 있어서 현실적이 아님을 쉽게 알 수 있다. ATIS의 노선정보체계를 위해서는 좌회전, 우회전, 직진, P-turn, U-turn 등의 교차로에서의 회전을 정확히 반영한 현실적 경로를 정확하게 제공되어야 하므로 위와 같은 가정은 적합하지가 않다. 이러한 단점을 극복하기 위해서 Sheffi(1985), Easa(1991), 최기주(1995) 등은 교통망의 코딩방법을 변형함으로써 해결할 수 있는 방법들을 거시적 및 미시적 교통망 표현기법(macro and micro network representation)과 함께 정리하였다. 즉 교차로에서의 각 회전에 대해 독립적인 링

크를 추가 코딩(coding)하여 별도의 비용을 부과함으로써 일반적인 수형망 최단경로탐색 알고리즘을 그대로 적용하면서도 교차로에서의 회전을 반영한 최단경로를 탐색할 수 있도록 하였다. 다른 방법으로는 교차로와 교차로 사이인 블록 중간(mid-blok)에 노드를 설정함으로써 링크가 회전비용과 블록 중간 점까지의 통행시간을 내포하도록 하여 교차로의 회전을 최단경로 탐색에서 반영하도록 한 방법이 있다. 이와 같은 mid-block coding 방법은 노정현과 남궁성(1995)의 링크표지기법 혹은 이승환, 최기주와 김원길(1996)의 링크탐색알고리즘과 계산과정은 많이 차이가 나지만 개념적으로는 매우 유사한 기법이다. 다만 링크탐색 알고리즘(링크표지기법)은 mid-block coding 방법과는 달리 교통망을 일반적인 코딩방법에 의해 입력하고 계산과정에서 링크에 가상적 번호를 별도로 준 후 최단경로를 탐색한다는 데에 차이가 있는 것이다.

미시적 교통망 코딩방법에 의해 교차로에서 회전을 반영하는 방법 외에도 FHWA(1973)에서는 수형망 알고리즘(tree building algorithm)에 대한 보완방법으로써 덩굴망 알고리즘(vine building algorithm)을 설명하고 있다. 이와 같은 덩굴망 알고리즘에 대해서는 Thomas(1991)가 구체적으로 설명되어 있다. 하지만 수형망 알고리즘이 교차로에서의 회전지체 및 급지를 반영하지 못한다는 단점을 보완하기 위해서 개발된 덩굴망 알고리즘도 또한 단점을 갖고 있다. 이와 같은 덩굴망 알고리즘의 단점을 최기주(1995), 노정현과 남궁성(1995)에서 구체적으로 설명하고 있다. 즉 U-turn, P-turn 및 좌회전금지 교차로가 2개 이상 연속적으로 위치한 경우에 기존의 덩굴망 알고리즘은 불합리한 경로를 추척할 가능성이 있다고 예제와 함께 설명하였다. 문제점의 근본은 기존의 수형망이나 덩굴망 알고리즘이 한 노드에 하나의 노드표지만 부여하고 있으며, 따라서 Bellman의 최적원리를 만족시키도록 되어 있다는 데 있다. 즉 U-turn과 P-turn과 같은 순환로의 경우 순환에 의한 추가적 비용으로 같은 노드에서 순환 전에 계산된 값보다 순환 후의 값이 항상 큰 값이 나오게 된다는 문제점이 있게 된다. 그러므로 노드표지는 순환전의 표지로서 수정되지 않은 채 그대로 기록되어 있게 되기 때문에 U-turn과 P-turn과 같이 현실적으로 순환로가 발생함에도 불구하고

하고 순환로의 노선비용, 전노드, 및 전전노드에 대한 기록이 남게 되지 않는다. 따라서 최단경로를 역추적할 때 현실적인 순환로의 추적은 불가능하게 된다. 즉 Bellman의 최적원리를 만족시키는 기존의 수형망 및 덩굴망 알고리즘은 하나의 노드가 최단경로 상에 두 번 이상 포함될 수 없다는 기본적인 특성 때문에 U-turn과 P-turn을 반영한 최단경로를 추적할 수 없게 되는 것이다. 또한 좌회전금지 교차로가 2개 이상이 연속해서 위치한 경우에도 수형망 알고리즘은 물론이고 기존 덩굴망 알고리즘도 합리적인 최단경로를 찾지 못한다는 단점을 갖고 있다.

이와 같은 기존 덩굴망 알고리즘의 단점을 정확히 이해하기 위해 Thomas (1991)가 설명한 기존의 덩굴망 알고리즘을 다음과 같이 정리하였다. 우선 본 논문에서 사용된 기호의 의미는 다음과 같다.

- C_{ij} : 출발노드 r에서 노드 j까지의 최단경로의 노선비용
- d : 도착노드
- i : 현재 최단경로탐색 대상이 되는 기준노드 (첫 번째 노드)
- j : 노드 i와 하나의 링크로 연결된 노드로써 현재 탐색대상이 되는 두 번째 노드
- J_i : 노드 i와 하나의 링크로 직접 연결된 노드로써 노드 i에서 노드탐색에 사용될 노드의 집합
- k : 노드 j와 하나의 링크로 연결된 노드로써 현재 탐색대상이 되는 세 번째 노드
- K_j : 노드 j를 거쳐 노드 i와 두개의 링크에 의해 연결된 노드로써 노드 i에서 노드탐색에 사용될 노드의 집합
- N : 검색 대상의 기준노드 i가 될 노드의 집합, $N = \{n_1, n_2, n_3, \dots\}$
- p_{rk} : 출발노드 r에서 노드 k까지의 최단경로로써 노드 k의 전노드
- pp_{rk} : 출발노드 r에서 노드 k까지의 최단경로로써 노드 k의 전전노드
- P_{ijk} : 시작노드 i에서 회전노드 j를 거쳐 끝 노드 k 이르는 차량회전의 회전 벌점(turn penalty)
- r : 출발지점의 노드번호(origin node number)
- R : 최종결과로써 최단경로의 노선노드 집합

t_{ij} : 링크 ij의 통행비용

◀단계 1▶ 모든 노드에 비용, 전노드 및 전전노드의 노지표지를 초기화한다.

- (1) $C_{rr}=0, p_{rr}=0, pp_{rr}=0 \quad \forall r$ (r : 출발노드)
- (2) $C_{ij}=t_{ij}, p_{ij}=r, pp_{ij}=0$
 $\forall j$ (j : r과 하나의 링크로 직접 연결된 노드)
- (3) $C_m=\infty, p_{m}=0, pp_m=0$
 $\forall n$ (n : $n \neq r$ 그리고 $n \neq j$)
- (4) r을 집합 N의 목록에 기록하여 둔다. 그리고 i=r로 놓는다.

◀단계 2▶ 노드 i를 집합 N의 목록에서 제거한다. 선택된 노드 i와 하나의 링크로 직접 연결된 노드 j를 집합 J_i 에서 선택하고, 이 과정을 모든 j에 대해서 다음의 과정을 수행한다. 만일 선택할 노드 j가 집합 J_i 에 없으면 (단계 3)으로 간다.

- (1) $C_{ij} = C_{ii} + t_{ij} + P_{hij}$ (여기서 h는 i의 전노드)를 계산한다.
- (2) 노드 j와 하나의 링크로 연결된 집합 K_j 의 모든 노드 k에 대해 다음과 같이 노선비용을 계산한다. 만일 집합 K_j 에 선택할 노드 k가 없으면 (1)로 간다.
 $T_{ik} = C_{ij} + t_{jk} + P_{ijk}$
만일 $T_{ik} < C_{ik}$ 이면, $C_{ik} = T_{ik}, p_{ik} = j, pp_{ik} = i$ 로 수정하고 노드 k를 집합 N의 목록에 기록하여 둔다. 만일 아니면, 아무 것도 수정하지 않는다.

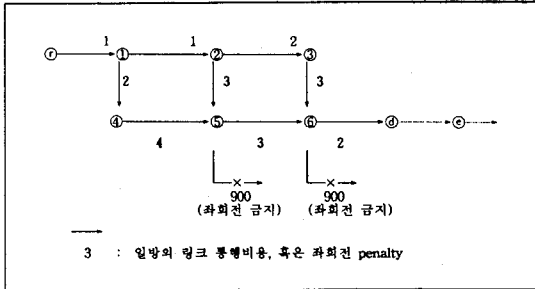
◀단계 3▶ 집합 N에서 순서에 따라 하나의 노드 n을 선택하고 $i=n$ 으로 놓고, (단계 2)로 간다. 만일 선택할 노드 n이 없으면 (단계 4)로 간다.

◀단계 4▶ 최종 결과인 노지표지를 이용하여 최단경로를 역추적한다.

- (1) 도착노드 d에 대해 C_{id} 을 읽고, d를 집합 R에 기록하여 둔다. 그리고 $k=d$ 로 놓는다.
- (2) p_{rk} 와 pp_{rk} 를 읽는다.
- (3) 만일 $pp_{rk}=0$ 이면, p_{rk} 를 집합 R에 기록하고 (4)로 간다.
만일 $pp_{rk}=r$ 이면, p_{rk} 와 pp_{rk} 를 집합 R에 기록하고 (4)로 간다.

만일 $pp_k \neq 0$ 이며 $pp_k \neq r$ 이면, p_k 와 pp_k 를 집합 R에 기록하고, $k=i$ 로 놓은 다음 (2)로 간다.

(4) 과정을 멈추고 집합 R를 출력한다.



<그림 1> 좌회전금지 교차로가 2개 이상 연속하여 위치한 예

<그림 1> 과 <표 1>은 위에서 설명한 기존의 덩굴망 알고리즘이 좌회전금지 교차로가 2개 이상이 연속해서 위치한 경우에 합리적인 최단경로를 제대로 찾지 못하고 있음을 보여주고 있는 단순한 예이다. 기존의 덩굴망 알고리즘이 연속된 좌회전금지 교차로에서 비합리적인 결과가 나오는 이유는 우선 노선비용 산출방법과 노드표지 기록방법에 기인한 것이다. <그림 1>의 예제에서 ①-②-⑤의 경로를 통해 노드 5에 도달하는 것이 ①-④-⑤의 경로를 통해 노드 5에 도달하는 것보다 빠르기 때문에 노드 5의 표지는 $C_5=5, p_5=2, pp_5=1$ 로써 최종적으로 기록되게 된다. 노드 6은 ②-③-⑥의 경로, ②-⑤-⑥의 경로 및 ④-⑤-⑥의 경로로 도달할 수 있다. 이 세가지 경로 중에 ②-③-⑥의 경로가 가장 작은 비용으로 도달하므로 최종적인 노드 6의 노드표지

는 $C_6=7, p_6=3, pp_6=2$ 가 된다.

문제는 노드 d에 대한 노드표지를 구하는데서 발생한다. 노드 d는 ③-⑥-④의 경로, ②-⑤-⑥-④의 경로 및 ④-⑤-⑥-④의 경로로 도달할 수 있다. 하지만 ③-⑥-④와 ②-⑤-⑥-④의 경로는 좌회전 금지가 포함되어 있으므로 실질적으로는 ④-⑤-⑥-④의 경로를 통해 노드 d에 도달하는 것이 최단경로가 된다. 그러나 기존 덩굴망 알고리즘에 있어 노드 d까지의 경로탐색 및 노선비용을 계산하는 방법으로는 노드 3과 노드 5에서부터 탐색하는 방법밖에 없다. 노드 3에서 경로를 탐색할 경우 좌회전 금지에 의해 무한대의 회전벌점이 부과되므로 선택되지 않을 것이다. 노드 5에서부터 탐색할 경우는 기존 덩굴망 알고리즘의 (단계 2)의 (1)부터 계산하게 되므로 노드 5의 노드표지에서 전노드가 노드 2로써 기록되어 있으므로 $C_{i5} = C_{i5} + t_{56} + P_{256}$ 으로 $5+3+900=908$ 이 계산되고, (단계 2)의 (2)에서 노드 d의 노드표지는 <표 1>에서와 같이 최종적으로 $C_{rd} = C_{r6} + t_{6d} + P_{56d}$ 에 의해 $C_{rd}=908+2+0=910$ 이 되고, $p_{rd}=6, pp_{rd}=5$ 가 되기 때문에 결국 불합리한 노드표지를 기록하게 됨을 알 수 있다. 이와 같은 문제는 노드 5와 노드 6에서 좌회전이 있기 전까지의 최단경로로써의 노드표지가 하나만 기록되어 있기 때문에 발생하는 것이다. 즉 좌회전을 고려한 노드 5의 실질적인 전노드 ④를 추적하여 노드 d까지의 비용을 계산할 수 있는 방법이 기존의 덩굴망 알고리즘에서는 없기 때문에 문제가 발생하는 것이다.

기존 덩굴망 알고리즘의 문제점은 비록 좌회전금

<표 1> 연속된 좌회전금지 교차로에의 기존 덩굴망 알고리즘의 적용

노선추적과 비용계산				노선표지 수정				집합 N의 수정		
i	j	k	C_k	n	p	pp	C_k	추가노드	N의 노드목록	
r	1	-	1	1	r	0	1	1	r, 1	
r	1	2	$1+1+0=2$	2	1	r	2	2	1	
		4	$1+2+0=3$	4	1	r	4	4		
		3	$2+2=4$	3	2	1	4	3		
1	2	5	$2+3+0=5$	5	2	1	5	5	2, 4	
		4	$3+4+0=7$	5	x	x	x	x		
		3	$4+3+0=7$	6	3	2	7	6		
2	5	6	$5+3+900=908$	6	x	x	x	x	4, 3, 5	
		6	$(3+4+0)+3+0=10$	6	x	x	x	x		
4	5	6	$(4+3+0)+2+900=909$	d	6	3	909	d	3, 5, 6	
3	6	d	$(5+3+900)+2+0=910$	d	x	x	x	x	5, 6	
5	6	d	$7+2+900=909$	d	x	x	x	x	6, d	
6	d	-		d	x	x	x	x	d	
최종 최단경로				r-1-2-3-6-d				최단경로 비용		909

n : 노드번호 p : 전노드번호 pp : 전전노드 번호 C_n : 출발지점에서 노드 n까지의 노선비용

x : 노드표지가 수정되지 않는 경우

지 교차로가 한군데가 독립적으로 있을 경우에도 발생한다. 만일 노드 d 다음에 노드 e가 계속해서 더 있다고 할 경우 노드 6에서부터 탐색을 시작하게 된다. 이때 노드 e까지의 비용을 산출하기에 앞서 노드 d까지의 비용을 계산하여야 하는데 노드 6의 전노드로써 노드 3이 기록되어 있으므로 실질적인 최단경로가 될 노드 6의 전노드로써 노드 5가 추적되지 못하여 비록 ②-⑤-⑥이 좌회전 금지 교차로가 아니더라도 최단경로를 합리적으로 추적하지 못하게 된다. 이와 같은 현상은 특정한 노드 n에 도달하는 최단경로와 노드 n을 거쳐 그 다음 노드에 도달하는 최단경로가 서로 다를 경우에 발생하는 문제이다.

최단경로탐색이 최종 완결되어 경로를 역추적하는(단계 4)에서도 “어떤 노드가 기준이 되어 전전노드까지 경로를 추적하는가”에 따라 다른 최단경로를 추적하게 된다는 문제점이 또한 있을 수 있다. 다시 말해서 도착노드에서 역추적하여 노드 e까지 도달하였다면, 노드 e의 전노드는 ④, 전전노드는 ⑥이 되고, 다시 노드 6에서 역추적하면 전노드는 ③, 전전노드는 ②가 되므로, 추적된 경로는 ②-③-⑥-④-⑥이 된다. 하지만 만일 도착노드에서 역추적하여 노드 d로 도달되었다면, d의 전노드는 ⑥, 전전노드는 ⑤가 되고, 다시 노드 5에서 역추적하면 전노드가 ②, 전전노드 ①이 되어 ①-②-⑤-⑥-④-⑥의 경로가 추적되어 노드 e에서 추적된 경로와는 다른 경로라는 것을 알 수 있다. 특히 좌회전금지 교차로가 2개 연속 위치되어 있을 경우에는 항상 둘 중에 한 교차로가 역추적의 기준 노드가 되므로 잘못된 경로가 항상 추적된다는 문제점이 존재함을 알 수 있다.

수행망 알고리즘 및 기존의 덩굴망 알고리즘은 이상에서 설명한 문제점으로 정교하고 정확한 최단경로 정보가 필요한 ATIS체계에 기존 알고리즘의 적용하는데는 한계성이 있다. U-turn, P-turn 및 좌회전금지 교차로가 연속해서 위치한 경우에서의 문제점을 개선하기 위해 노정현과 남궁성(1995) 그리고 이승환, 최기주와 김원길(1996)은 링크탐색알고리즘(링크표지기법)을 제안하였다. 링크탐색알고리즘은 min-block coding 방법과 개념상으로 유사하므로 교차로에서의 회전을 반영할 수 있다는 장점을 갖고 있다. 하지만 링크탐색알고리즘(링크표지기법)은 더미노드가 필요하고, U-turn시

에 추가적인 고려가 필요하며, 계산과정에서는 링크를 노드로 인식시키고 최종결과에서는 다시 링크를 노드로 바꾸는 작업이 필요하다. 그러므로 본 연구에서는 기존의 덩굴망 알고리즘의 기본적 골격을 유지하고, 또한 링크가 아닌 노드에 표지를 붙이면서도 U-turn, P-turn 및 좌회전금지 교차로가 연속해서 위치한 경우에서도 문제점이 발생하지 않는 수정형 덩굴망 알고리즘을 개발하여 앞에서 제시된 기존 알고리즘의 문제점을 해결하였다. 또한 수정형 덩굴망 알고리즘은 한번의 iteration에서 one-to-many의 최단경로를 탐색하고 링크가 아닌 각 노드에 노드표지가 붙어 있기 때문에 출발지점과 도착지점들간의 최단경로뿐 아니라 그 중간의 교통망 어떠한 노드까지도 최단경로를 쉽게 추적할 수 있다는 장점을 갖고 있다.

III. 수정형 덩굴망 알고리즘

앞에서 서술한바와 같이 기존의 덩굴망 최단경로 탐색 알고리즘은 두 개 이상의 연속된 좌회전 금지 교차로와 U-turn 및 P-turn 허용 교차로 등에서 비현실적 최단경로를 제공하게 되는 경우가 발생할 수 있다. 이와 같은 단점을 개선하기 위해서 본 연구에서는 통행비용 계산과정과 노선 역추적 과정을 수정하여 수정형 덩굴망 알고리즘(modified vine building algorithm)을 개발하였다. 앞에서 정의되지 않았던 추가적인 기호의 의미는 다음과 같다.

- C_k^g : 현재까지 계산된 값 중에서 출발지점 r에서 전노드코드 g인 전노드를 거쳐 노드 j까지 이르는 최소 통행비용
- g : 임의의 노드 k에 도달하는 방향을 나타내기 위한 코드로서 노드 k에 하나의 링크로 연결된 노드의 전노드 코드 g, $g \in G_k = \{1, 2, \dots\}$
- p_n^g : 출발지점 r에서 노드 j에 이르는 경로로서 전노드 코드 g에 해당하는 전노드의 번호(predecessor 혹은 back node)
- pp_n^g : 출발지점 r에서 노드 j에 이르는데 전노드코드 g에 해당하는 전노드의 전노드, 즉 노드 j의 전노드 코드 g에 해당되는 전전노드의 번호(back

back node)

P_{ij} : 노드 i에서의 j-i U-turn 벌점(penalty), 만일 i 노드가 U-turn 허용 노드인 경우

T_{rk} : 노선탐색 중 출발지점 r에서 전노드 p_{rk}^h 를 거쳐 노드k 까지의 현재 계산된 노선비용

본 연구에서 제시한 수정형 덩굴망 알고리즘은 다음과 같다.

◀단계 0▶ 모든 노드에 노드표지(node label)의 초기화

(1) $C_{ri}^1=0, p_{ri}^1=0, pp_{ri}^1=0$

(2) $C_{ij}^1=t_{ij}, p_{ij}^1=r, pp_{ij}^1=0$

$\forall j$ (j : r과 하나의 링크로 직접 연결된 노드)

(3) $C_{rk}^k=\infty, p_{rk}^k=0, pp_{rk}^k=0$

$\forall k, g$ (k : $k \neq r$ 그리고 $k \neq j$)

(4) r을 집합 N의 목록에 기록하여 둔다.

◀단계 1▶ 집합 N에서 순서에 따라 노드 i를 선택하고 노드 i를 집합 N의 목록에서 제거한다.

$J_i = \{j_1, j_2, \dots\}$ 의 집합을 구한다.

하지만 이때 $j=p_{ri}^h, (C_{ri}^h = \min\{C_{ri}^1, C_{ri}^2, \dots\})$ 인 경우는 j를 집합에 포함시키지 않는다. 단 j-i의 U-turn이 허용될 경우에는 j를 집합에 포함시킨다. 만일 집합 N에서 선택할 노드가 없으면 (단계 5)으로 간다.

◀단계 2▶ 집합 J_i 에서 순서에 따라 노드 j를 선택하고 집합 J_i 의 목록에서 j를 제거한다.

$K_j = \{k_1, k_2, \dots\}$ $k \neq i$ 인 집합을 구한다. 단 i-j의 U-turn이 허용되어 $k=i$ 인 경우에는 i를 집합 K_j 에 포함시킨다. 하지만 U-turn이 허용되지 않거나 $k=p_{ri}^h, (C_{ri}^h = \min\{C_{ri}^1, C_{ri}^2, \dots\})$ 인 경우에는 k를 K_j 에 포함시키지 않는다. 만일 선택할 j가 없으면 (단계 1)로 간다.

◀단계 3▶ 집합 K_j 에서 순서에 따라 k를 선택한다. 집합 K_j 목록에서 k를 제거한다.

k노드에 대한 전노드로써 j노드가 되는 전노드코드 g를 찾는다. 만일 j노드가 전노드로 기록이 되어 있지 않으면 전노드번호가 0으로 나오는 첫 번째 g를 노드 k에 대한 노드 j의 전노드 코드로 할당한다.

만일 선택할 노드 k가 없으면 (단계 2)로 간다.

◀단계 4▶ 선택된 노드 k까지의 노선비용을 계산하고 노드표지를 수정한다.

(1) j노드의 전노드들 중에 i 노드번호를 전노드로 하는 전노드코드 h를 찾는다. ($i = p_{ij}^h$)

(2) $T_{rk}^g = C_{ij}^h + t_{jk} + P_{jk}$ 를 계산한다. (j가 i-j U-turn 허용노드인 경우 : $P_{jk}=P_{ij}$)

(3) 만일 $T_{rk}^g < C_{rk}^k$ 이면, $C_{rk}^k = T_{rk}^g, p_{rk}^k = j, pp_{rk}^k = i$ 로 수정한다. 그리고 만일 $T_{rk}^g < \min\{C_{rk}^1, C_{rk}^2, \dots\}$ 이면, 노드 k와 j를 각각 집합 N의 목록에 추가하도록 한다. 하지만 k 또는 j가 집합 N에 이미 있으면 해당되는 노드는 포함시키지 않는다.

(4) 만일 $T_{rk}^g \geq C_{rk}^k$ 이면, 노드표지를 수정하지 않고, k와 j를 N에 추가하지도 않는다.

(5) (단계 3)로 간다.

◀단계 5▶ 노드표지를 이용하여 최단경로를 역으로 추적한다.

(1) $C_{rd}^g = \min\{C_{rd}^1, C_{rd}^2, \dots\}$ 인 전노드코드 g를 찾아, 전노드 $k=p_{rd}^g$, 전전노드 $j=pp_{rd}^g$ 를 찾는다.

(2) j, k, d를 순서대로 노선노드집합 R의 목록에 $R=\{j, k, d\}$ 와 같이 추가한다.

(3) 노드 k의 전노드가 j 즉 $j=p_{rk}^g$ 인 전노드코드 g를 찾아, g에 속한 k노드의 전전노드 $i=pp_{rk}^g$ 를 추적한다.

(4) i 노드를 집합 R 목록의 첫 번째 위치로 $R=\{i, j, k, \dots, d\}$ 와 같이 추가한다.

(5) 만일 $i=r$ 이면 최단경로 추적을 완료한다.

만일 아니면, $k=j, j=i$ 로 놓고 (단계 5)의 (3)으로 가서 반복한다.

Thomas(1991)의 기존 덩굴망 알고리즘과 본 연구에서 제시한 수정형 덩굴망 알고리즘은 노드표지(node labeling)하는 방법, (단계 1)과 (단계 2)의 경로추적의 대상 노드집합 J_i, K_j 을 설정하는 방법, (단계 4)에서 U-turn을 처리하는 방법 및 경로탐색 기준 노드 i의 집합 N을 설정하는 방법, (단계 5)에서 최종 탐색결과를 이용하여 노선을 역추적하는 방법에서 차이가 있다.

기존 덩굴망 알고리즘은 각 노드에 노선비용, 전노

드, 전전노드의 노드표지가 하나만 있으나, 수정형 덩굴망 알고리즘은 각 노드에 하나의 링크로 직접 연결된 노드에 접근방향별로 별도의 전노드 코드를 주어 그 전노드 코드의 수 만큼 노드표지가 있게 된다. 이것은 한 노드에 도달하는 방향별로 구분하여 최소비용을 산출하기 위한 것이다. 즉 링크탐색 알고리즘이 갖고 있는 교차로에서의 회전에 대한 처리의 장점을 반영하면서도 노드에 표지를 기록하는 방법을 제안한 것이다. 제안된 알고리즘에서 전노드 코드는 4지 교차로에서는 최대 4개가 있을 수 있으나 최소경로로 활용될 수 있는 방향은 대체로 3개 이하가 될 것이다. (단계 1)과 (단계 2)은 노드 i 에서 앞으로 탐색할 j 와 k 의 집합을 설정하는 단계이다. 이때 U-turn 이 허용되지 않는 구간에서는 최소의 노선비용을 갖는 전노드와 전전노드 그리고 바로 전에 탐색이 끝난 i 는 J (혹은 K)의 집합에 포함시키지 않아 전산속도를 빠르도록 하였다.

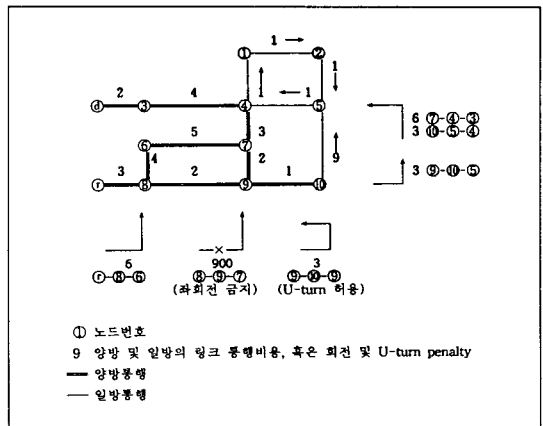
(단계 4)에서 수정형 덩굴망 알고리즘은 U-turn, P-turn 및 좌회전금지 교차로가 2개 이상으로 연속하여 위치한 경우일지라도 다른 일반적인 알고리즘과 동일한 방법으로 노선을 추적하면서도 현실적인 경로를 찾을 수 있다. 따라서 어떠한 교통망에서도 모든 교차로에 동일한 통행비용 산출방법을 적용할 수 있으며, 그러면서도 모든 교차로의 특성을 정확하게 반영하며 현실적 최단경로를 추적할 수 있게 된다.

(단계 5)은 모든 표지(labeling)가 최종적으로 결정된 후 최종적 최단경로의 노선비용과 도착지로부터 출발지까지 역으로 최단경로를 추적하는 단계이다. 이 단계는 U-turn 및 P-turn은 물론 좌회전금지 교차로가 2개 이상으로 연속하여 위치한 경우에 대한 경로탐색에 있어도 현실적인 노선을 추적할 수 있도록 수정된 부분이다. 최종 결과인 노드표지를 이용하여 최단경로를 역으로 추적할 때 항상 k 노드와 k 노드의 전노드 j 의 두개 노드를 항상 동일하게 하면서 하나의 노드 i 를 추적하여 가기 때문에 교차로의 회전벌점(turn penalty) 및 회전금지를 항상 반영하면서 역 추적하게 된다. 따라서 좌회전금지 교차로가 2개 이상으로 연속하여 위치한 경우에도 기존 덩굴망 알고리즘과는 달리 불합리한 경로는 최단경로에 포함되지 않게 된다.

기존의 링크탐색 알고리즘의 경우는 출발노드와 도착노드 사이의 임의의 노드까지 출발노드에서의 최단경로 추적을 위해 추가적인 계산이 필요하다. 그러므로 기존 알고리즘은 ATIS에 적용하여 교통망의 각 지점까지의 즉각적인 최단경로와 노선비용을 산출하는데 불편함이 있다는 단점이 있다. 하지만 수정형 덩굴망 알고리즘은 one-to-many 즉 하나의 출발노드에서 교통망의 모든 다른 노드까지의 최단경로를 하나의 계산과정(단계 1에서 단계 5)을 통해 한꺼번에 탐색이 가능하기 때문에 출발지점과 도착지점 사이의 어떠한 교차로까지의 최단경로와 최소비용을 쉽게 계산할 수 있다는 장점이 있어 ATIS 분야에의 응용이 좀 더 용이할 것으로 보인다.

IV. 단순 교통망에 의한 예제

수정형 덩굴망 알고리즘이 기존의 덩굴망 알고리즘에서는 탐색하지 못했던 경로추적을 어떻게 할 수 있는가를 <그림 2>와 같이 단순화된 예제 교통망을 가지고 구체적으로 검토하고자 한다. 예제 교통망은 U-turn, P-turn 및 좌회전 지체가 있는 교차로(좌회전 금지교차로의 예로도 설명될 수 있음)가 2개 이상 연속되어 위치한 경우가 모두 포함된 단순한 교통망이다. <표 2>는 기존의 덩굴망 알고리즘에 의한 최단 경로탐색 알고리즘의 탐색과정을 계산 순서에 따라 단계별로 정리한 것이다. <표 3>은 수정형 덩굴망 알



<그림 2> 최단경로탐색을 위한 단순교통망 예제

고리즘을 이용하여 최단경로 탐색과정을 계산 순서에 따라 한 단계 한 단계씩 그 과정을 상세히 정리한 것이다. 기존 알고리즘에 의해 추적된 최단경로는 ①-

⑧-⑨-⑩-⑤-④-③-①인 반면에 수정된 알고리즘의 결과는 ①-⑧-⑨-⑩-⑨-⑦-④-①-②-⑤-④-③-①로써 그 결과에 차이가 있음을 알 수 있다.

<표 2> 기존 덩굴망 알고리즘의 경로탐색 과정

노선추적과 비용계산				노선표지 수정				집합 N의 수정	
i	j	k	C _k	n	p	pp	C _k	추가노드	N의 노드목록
r	8	-	3	8	r	-	3	r, 8	r, 8
r	8	6	3+4+6=13	6	8	r	13	6	8
		9	3+2+0=5	9	8	r	5	9	
8	9	6	13+5+0=18	7	6	8	18	7	6, 9
		7	5+2+900=907	7	x	x	x	x	
		10	5+1+0=6	10	9	8	6	10	
6	7	4	18+3+0=21	4	7	6	21	4	9, 7, 10
		9	18+2+0=20	9	x	x	x	x	
9	7	4	907+3+0=910	4	x	x	x	x	7, 10, 4
		6	907+5+0=912	6	x	x	x	x	
		10	6+9+3=18	5	10	9	18	5	
7	4	1	21+1+0=22	1	4	7	22	1	10, 4, 5
		3	21+4+6=31	3	4	7	31	3	
10	5	4	18+3+1=22	4	x	x	x	x	4, 5, 1, 3
		1	22+1+0=23	2	1	4	23	2	
4	3	d	31+2+0=33	d	3	4	33	d	5, 1, 3
		1	22+1+0=23	1	x	x	x	x	
		3	22+4+0=26	3	4	5	26	3	
1	2	5	23+1+0=24	5	x	x	x	3, 2, d	
3	d	-	26+2+0=28	d	3	0	28	d	2, d
2	5	4	24+1+0=25	4	x	x	x	x	d
최종 최단경로			r-8-9-10-5-4-3-d				최단경로 비용		28

n : 노드번호 p : 전노드번호 pp : 전전노드 번호 C_n : 출발지점에서 노드 n까지의 노선비용
 x : 노드표지가 수정되지 않는 경우

<표 3> 수정형 덩굴망 알고리즘의 경로탐색 과정

노선추적과 비용계산				노선표지 수정				집합 N의 수정	
i	j	k	C _{ik} ^r	노드	p _{ik} ^r	pp _{ik} ^r	C _{ik} ^r (g)	추가노드	N의 노드목록
r	8	-	3	8	r	0	3 (1)	8	r, 8
r	8	6	3+4+6=13	6	8	r	13 (1)	6	8
		9	3+2+0=5	9	8	r	5 (1)	9	
8	9	6	13+5+0=18	7	6	8	18 (1)	7	6, 9
		7	5+2+900=907	7	9	8	907 (2)	x	
		10	5+1+0=6	10	9	8	6 (1)	10	
6	7	4	18+3+0=21	4	7	6	21 (1)	4	9, 7, 10
		9	18+2+0=20	9	7	6	20 (2)	x	
9	7	4	907+3+0=910	4	x	x	x	x	7, 10, 4
		6	907+5+0=912	6	7	9	912 (2)	x	
		10	6+9+3=18	5	10	9	18 (1)	5	
7	4	1	21+1+0=22	1	4	7	22 (1)	1	10, 4, 5
		3	21+4+6=31	3	4	7	31 (1)	3	
		5	18+3+1=22	4	5	10	22 (2)	x	
10	9	7	10+2+0=12	7	9	10	12 (2)	9, 7	4, 5, 1, 3
		8	10+2+0=12	8	9	10	12 (2)	x	
4	3	d	31+2+0=33	d	3	4	33 (1)	3, d	5, 1, 3, 9, 7
		1	22+1+0=23	1	x	x	x	x	
5	4	3	22+4+0=26	3	4	5	26 (1)	4, 3	1, 3, 9, 7, 2, d
		1	23+1+0=24	5	2	1	24 (2)	x	
9	7	4	12+3+0=15	4	7	9	15 (1)	7, 4	7, 2, d, 4
		6	12+5+0=16	6	7	9	16 (2)	x	

i	j	k	C_{ik}^g	노드	p_{ik}^g	pp_{ik}^g	$C_{ik}^g(g)$	추가노드	N의 노드목록
7	1	1	15+1+0=16	1	4	7	16 (1)	4, 1	2, d, 4
		3	15+4+6=25	3	4	7	25 (1)	4, 3	
2	5	4	24+1+0=25	4	5	2	25 (2)	×	d, 4, 1, 3
1	1	2	16+1+0=17	2	1	4	17 (1)	1, 2	1, 3
	3	d	26+2+0=28	d	3	4	28 (1)	3, d	
1	2	5	17+1+0=18	5	2	1	18 (2)	2, 5	3, 2, d
2	5	4	18+1+0=19	4	5	2	19 (2)	5, 4	d, 5
5	4	1	19+1+0=20	1	×	×	×	×	4
		3	19+4+0=23	3	4	5	23 (1)	4, 3	
4	3	d	23+2+0=25	d	3	4	25 (1)	3, d	
최종 최단경로			r-8-9-10-9-7-4-1-2-5-4-3-d				최단경로 비용		25

× : 노드표지를 수정하지 않는 경우

〈표 3〉의 최종결과에서 수정형 덩굴망 알고리즘에 의해 탐색된 최종경로에는 ⑨-⑩-⑨의 U-turn이 포함되어 현실적으로 찾아내었으며, 또한 ④-①-②-⑤-④의 P-turn도 현실에서와 마찬가지로 찾아냈음을 보여 주고 있다. 이것은 노드 9와 노드 4에서 기존의 덩굴망 알고리즘과 달리 방향별로 노드표지가 있어 방향별로 접근하는 것을 구분하여 표지하기 때문에 최단 경로 상에 동일한 노드가 두 번 이용될 수가 있었던 것이다. 이것은 각 노드에 하나의 노드표지만 있을 경우 Bellman의 최적원리를 만족시키므로 기존의 덩굴망 알고리즘과 같은 정통적인 labeling approach에 의한 최단경로탐색 알고리즘에서는 있을 수 없는 현상이다. 하지만 본 연구의 수정형 덩굴망 알고리즘은 방향별로 노선비용을 산출함으로써 같은 노드가 최단 경로상에 두 번 포함될 수 있으며, 또한 현실적인 최단경로와 일치하는 노선을 합리적으로 찾을 수 있음 을 예제를 통해 보여 준 것이다.

수정형 덩굴망 알고리즘의 또 다른 장점은 좌회전 금지 혹은 좌회전 체증이 있는 교차로가 2개 이상 연속되어 위치해 있을 때에도 합리적인 경로를 찾아낼 수 있다는 것이다. ⑦-④-③과 ⑩-⑤-④는 좌회전에 대한 회전별점이 있으며 최단경로 측에 2개 이상의 좌회전 회전별점이 연속적으로 위치한 경우의 예를 보여주는 것이다. 이와 같은 예는 좌회전 금지 교차로가 2개 이상 연속해서 위치한 경우와 같은 예라고 볼 수 있다. 이 예제의 경우는 좌회전이 허용됨에도 불구하고 노선경비가 더 작은 P-turn을 수정형 덩굴망 알고리즘이 최단경로로써 찾은 예인 것이다.

〈표 3〉를 통한 예에서 볼 수 있듯이 ⑨-⑦-④의

경로로 출발지점 r에서 노드 4에 도착할 경우 노선비용이 15인 반면에 ②-⑤-④의 경로로 노드 4에 도착할 경우 노선비용이 19가 됨을 알 수 있다. 그러므로 좌회전이 이루어지기 전까지 노드 4에 이르는 최단경로는 ⑨-⑦-④가 된다. 또한 수정형 알고리즘은 노드 5까지 도달하는 노선비용으로써 ⑨-⑩-⑤의 경로와 ①-②-⑤의 경로가 18로 동일하다. 하지만 기존 알고리즘에서는 ⑨-⑩-⑤의 경로를 통해 노드 5에 이르는 비용은 18이고, ①-②-⑤의 경로를 통해 노드 5에 이르는 비용은 24가 됨을 알 수 있다. 이것은 U-turn을 반영하지 못하고 다른 경로를 탐색한 결과로 노선비용이 증가하여 24가 되었기 때문이다. 여하튼 좌회전 별점이 다 반영된 후 노드 3 까지의 노선비용은 ⑨-⑦-④-③의 경로를 이용할 경우 비용이 25가 되고, ⑨-⑩-⑤-④-③의 경로를 이용할 경우에 26이 되며, 그리고 ①-②-⑤-④-③의 경로를 이용할 경우가 23이 되므로 ①-②-⑤-④-③의 P-turn을 이용한 경로가 실질적인 최단경로가 되어야 한다. 이러한 P-turn경로를 수정형 덩굴망 알고리즘은 합리적으로 탐색한 반면에 기존의 덩굴망 알고리즘은 이와 같은 최단경로를 찾아내지를 못했다. 이것은 앞에서 서술한 바와 같이 기존의 덩굴망 알고리즘은 U-turn과 P-turn을 근본적으로 반영하지 못하며, 연속된 좌회전 교차로에서의 지체를 제대로 반영 못하는 노드표지 기법 및 최단경로의 역추적 기법 등에 원인이 있는 것이다.

결론적으로 기존의 수정형 혹은 덩굴망 알고리즘에서 현실적으로 불합리한 최단경로를 추적하는 경우가 있는 반면에 본 연구에서 제시한 수정형 덩굴망 알고리즘은 U-turn, P-turn 및 좌회전금지 혹은 회전

지체 교차로가 2개 이상 연속되게 위치할 경우에도 현실적인 최단경로를 추적할 수 있음을 단순화된 교통망 예제를 통해 확실히 확인 할 수 있었다.

V. 결론

건설된 도로를 효율적으로 이용하고, 통행자의 편리성을 향상시키기 위해 첨단 여행자 정보체계(ATIS)를 활용할 수 있다. ATIS 체계 하에서 노선정보를 통행자에게 제공하기 위해서는 교차로에서의 회전으로 인한 지체를 정확하게 반영할 수 있는 효율적인 최단경로 알고리즘이 필요하다. 하지만 기존의 최단경로탐색 알고리즘은 좌회전 금지, U-turn, P-turn 등 교차로에서의 회전으로 인한 지체를 정확히 반영 못한다는 단점을 갖고 있다. 수형망 알고리즘의 단점을 보완하여 교차로 회전을 반영토록 개발된 덩굴망 알고리즘도 U-turn, P-turn 그리고 좌회전금지 교차로가 2개 이상 연속해서 위치한 경우 등에서 불합리한 최단경로를 탐색하는 단점을 갖고 있다. 또한 이와 같은 기존의 덩굴망 알고리즘의 단점을 극복하기 위해서 개발된 링크탐색알고리즘(혹은 링크표지기법)은 링크를 노드로 인식시키고 다시 노드화된 것을 다시 링크로 전환시키는 추가적인 작업을 필요로 한다. 또한 수정형 덩굴망 알고리즘은 링크탐색알고리즘(혹은 링크표지기법)과는 달리 U-turn을 반영하기 위해서 추가적인 더미노드 설정 등의 절차가 필요 없으며, 한번의 iteration에서 one-to-many의 최단경로를 탐색하며 각 노드에 표지를 붙임으로써 출발지점과 도착지점들간의 최단경로뿐만 아니라 그 중간의 교통망 어떠한 노드까지도 최단경로를 쉽게 추적할 수 있다. 그럼으로써 수정형 덩굴망 알고리즘은 기존의 덩굴망 알고리즘이 갖고 있는 단점을 극복하고 현실적인 최단경로를 탐색할 수 있다.

수정형 덩굴망 알고리즘은 노드표지에 있어 특정 노드로 유입하는 방향에 따라 노드표지를 별도로 기록하여 교차로에서의 좌회전, 우회전 및 직진은 물론 U-turn의 경우에도 추가적인 절차 없이 현실적으로 합리적인 최단경로를 탐색할 수 있도록 하였다. 또한 분석 결과 최종적으로 얻은 노드표지를 이용하여 최단경로를 역추적 하는 방법을 개선함으로써 좌회전 금지, U-

turn, P-turn 및 기타 회전에 의한 지체 등을 각 교차로마다 정확히 반영함으로써 비합리적인 최단경로가 추적되는 것을 근본적으로 차단하고 있다. 따라서 본 연구에서 개발한 수정형 덩굴망 최단경로탐색 알고리즘은 교차로에서의 회전지체 및 회전금지를 현실적으로 잘 반영함으로써 정확한 노선정보를 요구하는 ATIS체계를 분석하는데 유용하게 활용될 수 있는 기법이다.

참고문헌

1. 강맹규, 네트워킹과 알고리즘, 박영사, 1991.
2. 노정현, 남궁성, "도시가로망에 적합한 최단경로 탐색 기법의 개발", 대한국토도시계획학회지 국토계획 제30권 제5호, 1995.
3. 이승환, 최기주, 김원길, "도시부 ATIS 효율적 적용을 위한 탐색영역기법 및 양방향 링크탐색 알고리즘의 구현", 대한교통학회지 제14권 제3호, 1996.
4. 최기주, "U-Turn을 포함한 가로망 표현 및 최단경로의 구현", 대한교통학회 제13권 제3호, 1995.
5. Ahuja, R. K., Magnanti, T. L., and Orlin, J. B., Network Flows : Theory, Algorithms, and Applications, Prentice-Hall, Inc., 1993.
6. Easa, S. M., "Traffic Assignment in Practice : Overview and Guidelines for Users", Journal of Transportation Engineering, Vol. 117, No. 6, pp.602~623, 1991.
7. FHWA, Traffic Assignment : Methods, Applications, Products, U.S. Department of Transportation, August, 1973.
8. Larson, R. C. and Odoni, A. R., Urban Operations Research, Prentice-Hall, Inc., 1981.
9. Papageorgiou, M., Concise Encyclopedia of Traffic & Transportation System, Pergamon Press, 1991.
10. Sheffi, Y., Urban Transportation Networks : Equilibrium Analysis with Mathematical Programming Methods, Prentice-Hall, Inc., 1985.
11. Thomas, R., Traffic Assignment Techniques, Avebury Technical, 1991.