

■ 論 文 ■

유전자 알고리즘을 이용한 도시도로망에서의 첨단 여행자 정보시스템(ATIS) 운영계획

Minimum Travel Time Paths for ATIS in Urban Road Networks Using Genetic Algorithms

장 인 성

(숭실대학교 산업·정보시스템공학과 교수)

문 형 수

(KIS정보통신연구소 연구원)

목 차

- I. 서론
- II. 기존 알고리즘의 고찰
- III. 유전자 알고리즘의 개발
 - 1. 모집단의 초기화
 - 2. 평가함수
 - 3. 선택
 - 4. 교배
 - 5. 돌연변이
- IV. 적용예제
- V. 결론
- 참고문헌

Key Words : 첨단 여행자 정보시스템, 회전제한, 시간차, 동적 최단경로, 유전자 알고리즘

요 약

본 연구에서는 좌회전 금지 및 U-turn과 같이 회전제한이 있는 기하구조적 특성과 통행을 위한 시간차와 같이 특정 시간대에만 통행이 허용되는 가변적 특성이 부여된 도시도로망에서 기종점을 잇는 합리적인 최단경로를 탐색하는 문제를 다룬다. 본 연구의 동기는 지능형 교통시스템(ITS)의 응용분야인 첨단 여행자 정보시스템(ATIS)의 운영을 위해 필수적으로 요구되는 최단여행시간 경로를 탐색하고자 하는 문제에 의해서 유발되었다. 기하구조적 특성과 가변적 특성이 부여된 교통망은 도시도로망의 현실성을 보다 더 사실적으로 표현하지만 전통적인 알고리즘과 지금까지 개발된 알고리즘들은 가변적 특성이 반영된 최단경로를 탐색하는데 실패한다. 본 논문은 ATIS를 도시도로망에서 운영하는데 필요한 합리적인 최단경로 정보를 실시간으로 탐색할 수 있는 최단경로 알고리즘을 제안한다. 최단경로 알고리즘은 유전자 알고리즘 접근법을 이용하여 개발되었으며 모의실험에 의해 제안된 알고리즘의 우수성을 입증하였다.

1. 서론

첨단 여행자 정보시스템(Advanced Traveler Information System:ATIS)은 컴퓨터, 전자, 정보통신 등의 신기술을 이용하여 여행자에게 실시간(real time)의 제반 교통정보를 제공함으로써 교통수요를 분산하고 교통시설의 이용효율을 극대화하기 위한 체계로 차세대 교통체계인 지능형 교통시스템(Intelligent Transportation System:ITS)의 부 체계(subsystem)를 이루고 있다. ATIS를 운영하기 위해서는 현실성이 반영된 합리적인 경로를 실시간으로 탐색할 수 있는 알고리즘의 개발이 필수적이다. 특히 ATIS를 대도시의 복잡한 도로망에서 운영하기 위해서는 경로계산과정에 도로망의 기하구조적 특성뿐만 아니라 가변적 특성을 반영할 수 있어야만 한다. 교차로에서의 좌회전 금지나 이로 인한 P-turn 및 U-turn 허용 등의 회전제한은 도시도로망에서 자주 발생하는 대표적인 기하구조적 특성이다. 도시도로망의 가변적 특성은 도로내의 시설 공사나 공공 행사에 의한 통행시간 제한, 터널이나 교량의 통행시간 제한, 도로의 원활한 교통소통을 위한 좌회전 및 U-turn의 허용시간 제한 등을 들 수 있다.

지금까지는 도시도로망의 독특한 기하구조적 특성인 회전제한을 효율적으로 반영할 수 있는 최단경로탐색 알고리즘의 개발이 연구의 주류를 이루었다(김익기, 1998; 노정현, 남궁성, 1995; 이승환 외, 1996; 최기주, 1995; Caldwell, 1961; Sheffi, 1985; Thomas, 1991). 하지만 이들 알고리즘들은 도시도로망에 가변적 특성이 부여되면 합리적인 최단경로를 탐색하는데 실패한다. 통행시간 제한, 회전허용시간 제한 등의 가변적 특성은 어떤 링크의 통행이나 어떤 링크에서의 방향 전환이 특정 시간대에서만 가능하다는 것을 의미한다. 도로망의 가변적 특성을 고려하기 위한 연구는 근래에 몇몇 연구자들에 의해 수행되어진 바 있으나 어떤 노드의 통행이 특정시간 또는 특정시간대에서만 가능한 최단경로문제를 대상으로 하고 있다(Chen and Tang, 1997, 1998; Desrochers and Soumis, 1988). 어떤 노드의 통행시간이 제한되면 그 노드를 출발노드로 하는 모든 링크의 통행시간도 제한된다. 그러나 실제로 이러한 경우는 드물며 어떤 노드에 연결된 특정 링크만이 통행시간이 제한되는 경우가 일반적이다. 또한, 이들 알고리즘들은 또 다른 가변적 특성인

회전허용시간 제한이나 회전제한 등의 기하구조적 특성이 부여되면 합리적인 최단경로를 탐색하는데 실패한다. 그러므로 ATIS를 도시도로망에서 운영하기 위해서는 기하구조적 특성과 가변적 특성이 반영된 합리적인 최단경로를 신속하게 탐색할 수 있는 알고리즘의 개발이 요구된다.

본 연구에서는 좌회전 금지와 U-turn이 허용되는 기하구조적 특성과 좌회전 허용시간이 제한되는 가변적 특성이 부여된 도시도로망을 대상으로 발견적 탐색기법(heuristic search method)중의 하나인 유전자 알고리즘(Genetic Algorithm:GA)을 이용하여 합리적인 최단경로를 신속하게 효율적으로 도출할 수 있는 해법을 제시하고자 한다. 본 연구에서 개발된 알고리즘의 특징은 첫째, 링크탐색 알고리즘(노정현, 남궁성, 1995; 이승환 외, 1996)과 수정형 덩굴망 알고리즘(김익기, 1998)과 같이 도로망의 확장 없이 경로계산과정에서 연속좌회전 금지를 포함한 회전제한을 반영할 수 있을 뿐만 아니라 회전허용시간 제한도 고려하기 때문에 도시도로망의 특성을 보다 더 반영한다는 점과 둘째, 특정 노드들에 대해서만 단계별로 경로계산이 수행되어지므로 모든 노드나 모든 링크에 대하여 단계별로 경로계산을 수행하는 기존의 방법보다 계산속도가 상당히 빠르다는 점, 셋째, 모집단을 이용하는 유전자 알고리즘의 특성으로 k 최단경로 알고리즘(Lawler, 1972)과 같이 최적경로외에도 복수개의 부가적인 경로가 함께 제공되기 때문에 경관과 편의성 등을 고려해서 이용자가 자신에게 가장 적합한 대체 경로를 선택할 수 있다는 점이다.

II. 기존 알고리즘의 고찰

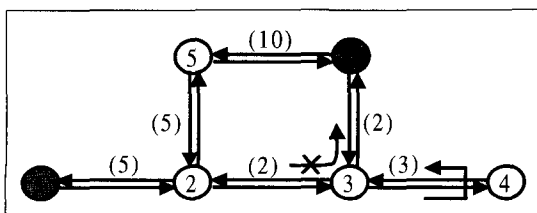
지금까지 개발된 대부분의 최단경로탐색 알고리즘들은 노드기반의 순차적 탐색법을 채택하고 있다. 노드기반의 순차적 탐색법에 기초로하는 이들 알고리즘들은 경로계산을 위해 전노드만을 검색하는 수형망 알고리즘(tree building algorithm)과 전전노드까지를 검색범위로 하는 덩굴망 알고리즘(vine building algorithm)의 두 가지 형태로 크게 분류할 수 있으며 Dijkstra 알고리즘(1959)은 수형망 알고리즘의 대표적인 예이다(Thomas(1991)). 노드기반의 순차적 탐색법은 다음과 같은 가정의 전제하에 최단경로를 탐색한다.

[가정 1] 동일한 노드가 최단경로 상에 두 번 이상 포함될 수 없다.

[가정 2] 기점노드로부터 각 노드까지의 최단경로를 단계별로 계산하고 최종적으로 이들을 산술적으로 연결하면 기점과 종점을 잇는 최단경로가 구축된다.

[가정 1]은 P-turn 및 U-turn이 허용되는 도로망에서는 성립되지 않는다. 예를 들어 <그림 1>과 같은 도로망을 고려해보자. 노드 1과 노드 6은 각각 기점과 종점을 나타내고 있으며 링크 위에 있는 괄호안의 숫자는 링크를 통과하는데 소요되는 시간(분)을 나타낸다. 노드 3에서 좌회전이 금지되며 노드 3을 경유해서 노드 6으로의 여행이 가능하도록 노드 4에서만 오직 U-turn이 허용된다. <그림 1>의 도로망에서 기점과 종점을 잇는 통행 가능한 경로는 ①-②-⑤-⑥, ①-②-③-④-③-⑥, ①-②-③-④-③-②-⑤-⑥이며 총 소요시간은 각각 20분, 15분, 30분이다. 따라서 노드 1에서 노드 6까지의 최단경로는 ①-②-③-④-③-⑥이며 노드 4에서의 U-turn허용으로 인해 노드 3이 최단경로 상에 중복된다. P-turn이 허용되는 도로망에서 P-turn이 최단경로 상에 포함되는 경우에도 동일한 노드가 최단경로 상에 중복된다.

마찬가지로 [가정 2]는 <그림 1>에서와 같이 좌회전 금지가 발생하는 도로망에서 성립되지 않는다. [가정 2]는 기점노드 1로부터 노드 2, 3, 4, 5까지의 최단경로를 단계별로 계산하고 최종적으로 노드 3까지의 최단경로인 부분경로 ①-②-③에 노드 6을 산술적으로 연결한 경로 ①-②-③-⑥을 최단경로로 탐색한다. 그러므로 기하구조적 특성인 회전제한이 발생하는 도시 도로망에서는 [가정 1]과 [가정 2]가 위배되므로 전통적인 노드기반의 순차적 탐색법을 이용해서 합리적인 최단경로 ①-②-③-④-③-⑥을 탐색할 수 없다.



<그림 1> 회전제한이 있는 도로망

이러한 문제점을 해결하기 위해 Caldwell(1961), Sheffi(1985) 등은 도로망을 확장 또는 변형하여 도로망에 회전제한을 표현함으로써 노드기반의 순차적 탐색법을 적용할 수 있도록 하였다. 그러나 이와 같은 방법은 경로탐색을 위한 노드 및 링크 수가 증가하여 경로산정에 많은 시간이 소비된다는 단점을 갖고있다. 따라서, 보다 신속하고 효율적으로 합리적인 최단경로를 탐색할 수 있는 방법에 관한 연구가 진행되었으며 도로망의 확장 없이 경로 계산과정에서 회전제한을 반영할 수 있는 링크탐색 알고리즘(노정현, 남궁성, 1995; 이승환 외, 1996)과 수정형 덩굴망 알고리즘(김익기, 1998) 등이 개발되었다.

노드기반의 순차적 탐색법은 통행시간이 제한되는 가변적 특성이 부여된 도로망에서도 합리적인 최단경로를 탐색하는데 실패한다. <그림 1>에서 링크 (2, 5)와 링크 (5, 6)의 통행시간을 각각 1분으로 수정하고 노드 3에서 좌회전을 허용하면 노드 1에서 노드 6까지의 최단경로는 ①-②-⑤-⑥이며 총 소요시간은 7분이다. 이때, 노드 2와 노드 5를 연결하는 링크 (2, 5)가 특정 시간대인 [10분, 20분]에서만 통행이 허용되는 시간창(time window)을 갖는다고 가정하자. 기점 1을 출발하는 시점을 10분이라 하면 노드 1을 출발해서 노드 2에 도착한 시간은 15분이며 링크 (2, 5)의 통행이 허용되는 시간창에 있기 때문에 링크 (2, 5)는 통행 가능하다. 따라서, 최단경로 ①-②-⑤-⑥은 현실적으로 이용 가능하다. 반면에 기점 1을 출발하는 시점을 0분이라 하면 노드 1을 출발해서 노드 2에 도착한 시간은 5분이며 링크 (2, 5)의 통행이 허용되는 시간창에 있지 않기 때문에 링크 (2, 5)는 통행 불가능하다. 따라서, 경로 ①-②-⑤-⑥은 통행 불가능한 경로이다. 즉, 시간창이 존재하면 기점의 출발시간에 따라서 최단경로가 변하지만 노드기반의 순차적 탐색법은 가변적 특성이 부여되지 않은 도로망을 위해 개발된 정적 최단경로 알고리즘(static shortest path algorithm)이기 때문에 기점을 출발한 시간과 관계없이 언제나 동일한 경로 ①-②-⑤-⑥을 최단경로로 탐색한다.

Desrochers and Soumis(1988)는 특정 노드가 시간창을 갖고있기 때문에 노드의 통행이 제한되는 도로망에서 최단경로를 탐색할 수 있는 알고리즘을 개발하였다. Chen and Tang(1997, 1998)은 노드에 부여된 시간창을 (10분, 35분, 50분)과 같이 일

반화하여 특정 시간에서만 노드의 통행이 허용되는 최단경로문제에 적용할 수 있는 알고리즘을 개발하였다. 어떤 노드의 통행이 특정 시간이나 특정 시간대에만 허용되면 그 노드를 출발노드로 하는 모든 링크의 통행도 제한된다. 그러나 이러한 경우 심각한 교통체증을 유발하며 기점과 종점을 연결하는 통행 가능한 경로가 존재하지 않을 수 있다. 도시도로망에서 노드의 통행이 제한되는 경우는 드물며 시설 공사나 공공행사 등에 의해 특정 시간대에 특정링크의 통행이 제한되는 경우가 일반적이다. 최근에 장인성(2000b, 2001)은 링크에 시간창이 부여되어 링크의 통행이 제한되는 최단경로문제에 적용할 수 있는 알고리즘을 개발하였다. 그러나 이 알고리즘은 다수의 교통수단이 혼용되는 복합교통망에서 환승문제 및 수단별 이용시간제한을 다루기 위해 개발되었기 때문에 도시도로망에서 자가운전자를 위한 경로탐색에는 적합하지 않다.

한편, 회전허용시간 제한은 대도시의 복잡한 도로망에서 원활한 교통소통을 위해 자주 발생하는 가변적 특성이다. <그림 1>의 노드 3이 좌회전 허용을 위한 시간창 [20분, 30분]을 갖는다고 가정하자. 기점 1을 출발하는 시점을 15분이라 하면 노드 1을 출발해서 노드 3에 도착한 시간은 22분이며 링크 (3, 6)로의 회전이 허용되는 시간창에 있기 때문에 최단경로는 ①-②-③-⑥으로 변하며 총 소요시간은 9분이다. 회전제한을 반영하기 위해 개발된 링크탐색 알고리즘과 수정형 덩굴망 알고리즘도 정적 최단경로 알고리즘이기 때문에 회전이 가변적인 도로망에는 적용할 수 없다. 회전허용시간이 제한되는 도로망에서 합리적인 최단경로를 탐색하기 위해서는 통행시간 제한뿐만 아니라 기하구조적 특성인 회전제한을 동시에 고려해야하므로 매우 복잡하다. 본 연구에서는 대도시의 복잡한 도로망에서 좌회전 금지와 U-turn이 허용되는 기하구조적 특성뿐만 아니라 좌회전 허용시간이 제한되는 가변적 특성이 반영된 동적 최단경로 알고리즘(dynamic shortest path algorithm)을 개발하기 위해 유전자 알고리즘을 적용하였다.

III. 유전자 알고리즘의 개발

유전자 알고리즘에서는 대상이 되는 문제의 후보해들을 문자나 기호의 배열인 염색체(chromosome)로 표현

하며 염색체를 구성하는 문자나 기호들을 유전자(gene)라 부른다. 이들 염색체들의 적당한 크기(population size)로 구성되는 모집단(population)을 생성하고 평가하여 점차적으로 갱신함으로써 전역적인 최적해를 탐색하게 된다. 모집단의 생성과정을 초기화(initialization)라 하며 모집단의 갱신은 선택(selection)과 유전 연산자(genetic operator)에 해당되는 교배(crossover), 돌연변이(mutation)의 과정을 통해서 이루어지고 이를 1세대(generation)라 한다. 해의 탐색과정은 설정한 최대세대 수(maximum generation)에 도달할 때까지 반복되며 모집단을 이용한 병렬처리를 수행하기 때문에 탐색소요시간이 매우 짧다. 유전자 알고리즘에 대한 전체적인 절차(Michalewicz, 1994)는 다음과 같으며 t 는 세대수, max_gen 은 최대세대 수를 나타낸다.

Procedure : GA

begin

$t=0$;

initialize the population of solutions $P(t)$ by initialization procedure ;

evaluate $P(t)$ by evaluation procedure ;

while ($t \leq max_gen$) **do**

begin

select $P(t+1)$ from $P(t)$ by selection procedure ;

recombine $P(t+1)$ using genetic operators ;

evaluate $P(t+1)$ by evaluation procedure ;

end

$t=t+1$;

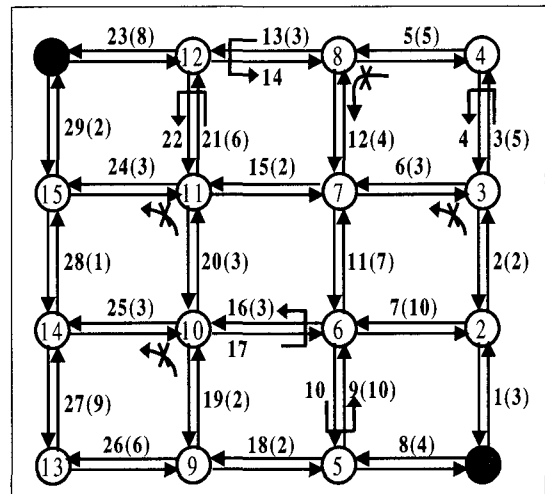
end

1. 모집단의 초기화

유전자 알고리즘의 핵심은 대상이 되는 문제의 후보해(후보경로)를 염색체로 표현하기 위한 방법을 설계하는 것이다. 최근에 최단경로문제에 유전자 알고리즘을 이용한 연구들이 몇몇 연구자들에 의해 수행되었다(김현명, 임용택, 1999; 이승재 외, 2000; 장인성, 2000a, 2001; 장인성, 박승현, 2001). 김현명과 임용택(1999)은 복합교통망에서의 환승문제를 다루기 위해 유전자 알고리즘을 적용하였으며, 이승재 외(2000)

는 2목적 최단경로문제에 유전자 알고리즘을 적용하였다. 이들 연구들에서는 유전자를 교통망의 링크 수만큼 확보하고 각 유전자에 링크의 이용값을 할당함으로써 염색체를 표현하였다. 이들 방법에 의해 생성된 염색체는 후보경로들에 대응되는 것이 아니라 링크의 이용값이 서로 다른 교통망들을 나타낸다. 따라서 염색체의 평가가 Dijkstra 알고리즘(1959)과 같은 수리적인 최단경로 알고리즘을 이용해서 교통망들에 대해 독립적으로 이루어지기 때문에 교통망의 규모가 대규모인 경우에는 많은 계산수행시간이 요구되는 단점을 갖고 있다. 이와 같은 문제점을 극복하기 위해 장인성(2000a)은 염색체의 평가과정에 A* 알고리즘(Winston, 1992)을 이용하여 탐색영역을 제한함으로써 탐색효율을 향상시켰다. 그러나, 이들 연구들은 순수한 유전자 알고리즘을 이용한 접근법이라기보다는 수리적인 최단경로 알고리즘과 유전자 알고리즘을 혼합한 접근법에 가깝다.

장인성과 박승헌(2001)은 링크의 방향이 단방향인 복합교통망에서 환승문제를 다루기 위해 유전자 알고리즘을 적용하였으며, 장인성(2001)은 링크의 방향이 양방향인 복합교통망에서 환승문제 및 서비스 이용시간 제한을 다루기 위해 유전자 알고리즘을 적용하였다. 이들 연구들에서는 기점과 종점을 잇는 임의의 경로를 구성할 수 있는 노드들을 나열함으로써 후보경로들을 염색체로 표현하였다. 본 연구에서는 장인성(2001)이 제시한 염색체 표현방법을 도시도로망에 적용할 수 있도록 수정하였으며 염색체의 표현 절차를 유연하게 하기 위해 경로를 구성할 수 있는 노드들을 나열하는 과정에서 회전제한 및 좌회전 허용시간 제한을 무시하였다. <그림 2>에 예시된 격자형 도로망(노드×노드=4×4)은 48개의 링크와 16개의 노드로 구성되어있으며 노드 1과 노드 16은 기점과 종점을 나타낸다. 링크 위에 있는 숫자는 대응되는 링크번호이며, 괄호 안의 숫자는 링크를 통과하는데 소요되는 시간(분)을 나타낸다. 임의의 두 노드 사이에 놓여진 양방향 링크에 대해 동일한 링크번호가 부여되지만 양방향 링크의 어느 한쪽에서라도 U-turn이 허용되면 서로 다른 링크번호가 부여된다. 예를 들면, 노드 1과 노드 2 사이에 놓여있는 링크 (1, 2)와 링크 (2, 1)의 링크번호는 동일하게 1이 부여되지만 노드 3과 노드 4 사이에 놓여진 링크 (3, 4)에서 U-turn이 허용되므로 링크 (3, 4)의 번호는 3, 링크 (4, 3)의



<그림 2> 도시도로망의 예

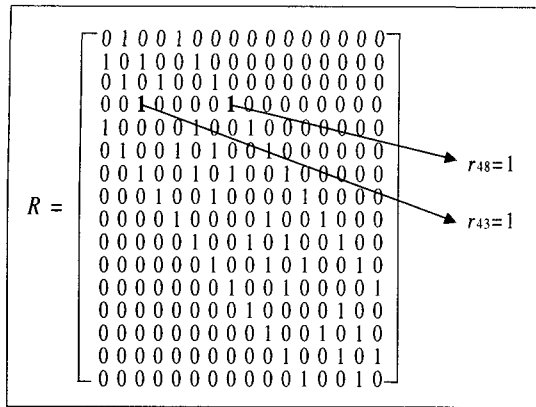
번호는 4로 서로 다르게 부여된다. 이와 같이 각 링크에 번호를 부여함으로써 회전제한 및 좌회전 허용시간 제한에 의해 현실적으로 통행 불가능한 경로에 대응되는 염색체들을 구별할 수 있으며 후술되는 2절의 평가 과정에서 이러한 경로들이 최단경로에 포함되는 것이 방지된다. 링크번호 3의 종결노드 4와 링크번호 10의 종결노드 5, 링크번호 13의 종결노드 12, 링크번호 17의 종결노드 6, 링크번호 21의 종결노드 12에서 U-turn이 허용된다. 링크번호 5에서 링크번호 12의 방향과 링크번호 19에서 링크번호 25의 방향, 링크번호 20에서 링크번호 24의 방향으로 각각 좌회전이 금지되며 링크번호 2에서 링크번호 6의 방향으로는 특정 시간대인 [7분, 10분]에서만 좌회전이 허용된다.

<그림 2>의 도로망에서 후보경로 $p_1 = \textcircled{1}-\textcircled{2}-\textcircled{3}-\textcircled{4}-\textcircled{3}-\textcircled{7}-\textcircled{11}-\textcircled{15}-\textcircled{16}$ 과 $p_2 = \textcircled{1}-\textcircled{5}-\textcircled{6}-\textcircled{10}-\textcircled{11}-\textcircled{12}-\textcircled{16}$ 에 대응되는 염색체를 각각 A와 B라 하면 염색체 A, B는 다음과 같이 표현된다.

$$A = (g[1] \ g[2] \ g[3] \ g[4] \ g[5] \ g[6] \ g[7] \ g[8] \ g[9]) \\ = (1 \ 2 \ 3 \ 4 \ 3 \ 7 \ 11 \ 15 \ 16)$$

$$B = (g[1] \ g[2] \ g[3] \ g[4] \ g[5] \ g[6] \ g[7]) \\ = (1 \ 5 \ 6 \ 10 \ 11 \ 12 \ 16)$$

여기서 염색체 A와 B의 $g(i)$ 는 각 염색체의 i 번째 유전자를 나타내며 경로 p_1 과 p_2 의 i 번째 노드가 각각 이에 대응된다. 특히 첫 번째 유전자인 $g(1)$ 에는 기점노드가 대응되며 마지막 번째 유전자인 염색체 A의



〈그림 3〉 선행관계 행렬

$g(9)$ 와 염색체 B 의 $g(7)$ 에는 종점노드가 대응된다. 모집단 생성을 위한 컴퓨터 코딩작업을 효율적으로 수행하기 위해 노드들의 선행관계 행렬 $R = \{r_{ij}\}_{n \times n}$ 을 이용하였다. 선행관계 행렬 R 은 n 차원의 정방행렬이며 n 은 종점노드, r_{ij} 는 선행관계 행렬의 성분을 나타낸다. 여기서 노드 j 가 노드 i 와 하나의 링크로 연결된 노드 i 의 직접적인 후속노드라면 $r_{ij}=1$, 노드 j 가 노드 i 의 직접적인 후속노드가 아니라면 $r_{ij}=0$ 이다.

〈그림 2〉에 예시된 복합교통망에 대하여 선행관계 행렬 R 을 작성하면 〈그림 3〉과 같다. 예를 들면, 노드 4의 후속 노드는 노드 3, 노드 8이므로 $r_{43}=r_{48}=1$ 이다. R 을 이용해서 염색체를 나타내기 위해 먼저 기점노드인 노드 1을 첫 번째 유전자 $g(1)$ 에 대응시킨다. $i=1$ 로 치환하여 $r_{ij}=1$ 인 노드 1의 후속노드 j 를 찾는다. j 가 2, 5인 경우에 대해 $r_{12}=r_{15}=1$ 이므로 이중 하나를 임의로 선택해서 $g(2)$ 에 대응시킨다. 선택된 j 가 2라 하면 $i=j=2$ 로 치환하여 $r_{2j}=1$ 인 노드 2의 후속노드 j 를 찾는다. $r_{21}=r_{23}=r_{26}=1$ 이므로 임의로 $j=3$ 을 $g(3)$ 에 대응시키고 $i=j=3$ 으로 치환한다. $r_{32}=r_{34}=r_{37}=1$ 에서 임의로 $j=4$ 를 $g(4)$ 에 대응시키고 $i=j=4$ 로 치환한다. 이 과정을 i 가 종점노드일 때까지 반복함으로써 다음과 같은 염색체 A 를 얻을 수 있다.

$$A = (g[1] \ g[2] \ g[3] \ g[4] \ g[5] \ g[6] \ g[7] \ g[8] \ g[9]) \\ = (1 \ 2 \ 3 \ 4 \ 3 \ 7 \ 11 \ 15 \ 16)$$

염색체 A 의 유전자 $g(4)$ 까지를 생성하는 과정에서 만일 $g(3)=1$, $g(4)=2$ 로 할당하면 링크 (1, 2)가 중복되고 유전자 $g(6)$ 까지를 생성하는 과정에서 $g(6)=4$ 로 할당하면 링크 (3, 4)가 불필요하게 중복된다. 동

일한 링크가 중복되면 무한 루프가 발생할 수 있고 이로 인해 염색체 A 의 길이를 무한대로 만들 수 있다. 본 연구에서는 $g(3)$ 이후의 유전자를 생성하는 과정에서 동일한 링크가 중복되는 것을 방지하도록 하였으며 구체적으로는 아래와 같은 초기화 절차에 의해 모집단을 생성하였다. c 는 염색체 번호, pop_size 는 모집단의 크기, ψ_i 는 특정 노드 i 에 대해 $r_{ij}=1$ 을 만족시키는 노드 j 들의 집합을 나타낸다.

Procedure : Initialization

[Step 1]

$c=1$ 로 설정한다.

[Step 2]

1. 만일 $c \leq pop_size$ 이면 i =기점노드, $m=1$, $g(m)$ =기점노드로 설정하고 [Step 3]으로 이동한다.
2. 만일 $c > pop_size$ 이면 초기화 절차를 종료한다.

[Step 3]

1. 만일 $i \neq$ 종점노드이면 $\psi_i = \emptyset$, $j=1$ 로 설정하고 [Step 4]로 이동한다.
2. 만일 $i =$ 종점노드이면 $c=c+1$ 로 하고 [Step 2]로 재이동한다.

[Step 4]

1. 만일 $j \leq \max\{\text{노드번호}\}$ 이면 선행관계 행렬 R 의 성분 r_{ij} 를 조사하고 [Step 5]로 이동한다.
2. 만일 $j > \max\{\text{노드번호}\}$ 이면 $m=m+1$ 로 하고 [Step 6]으로 이동한다.

[Step 5]

1. 만일 $r_{ij}=1$ 이면 $\psi_i = \psi_i \cup \{j\}$, $j=j+1$ 로 하고 [Step 4]로 재이동한다.
2. 만일 $r_{ij}=0$ 이면 $\psi_i = \psi_i$, $j=j+1$ 로 하고 [Step 4]로 재이동한다.

[Step 6]

집합 ψ_i 에서 하나의 노드 k 를 임의로 선택해서 $g[m]=k$ 로 설정한다.

[Step 7]

1. 만일 $m \leq 3$ 이면 $i=k$ 로 하고 [Step 3]으로 재이

동한다.

- 만일 $m > 3$ 이면 [Step 8]로 이동한다.

[Step 8]

- 만일 $g[a]=g[m-1]$ 이고 $g[a+1]=g[m]$ 인 $a(1 \leq a < m-1)$ 가 존재하면 $h=m-1$, $\Psi_{g[h]} = \Psi_{g[h]} - \{g[h+1]\}$ 로 하고 [Step 9]로 이동한다.
- 만일 $g[a]=g[m-1]$ 이고 $g[a+1]=g[m]$ 인 $a(1 \leq a < m-1)$ 가 존재하지 않으면 $i=k$ 로 하고 [Step 3]으로 재이동한다.

[Step 9]

- 만일 $\Psi_{g[h]} \neq \emptyset$ 이면 $\Psi_{g[h]}$ 에서 하나의 노드 q 를 임의로 선택하고 $g[h+1]=q$ 로 수정한다. $i=q$, $m=h+1$ 로 하고 [Step 3]으로 재이동한다.
- 만일 $\Psi_{g[h]} = \emptyset$ 이면 [Step 10]으로 이동한다.

[Step 10]

$h=h-1$, $\Psi_{g[h]} = \Psi_{g[h]} - \{g[h+1], g[h-1]\}$ 로 하고 [Step 9]로 재이동한다.

[Step 4], [Step 5]는 특정 노드 i 에 대해 $r_{ij}=1$ 을 만족시키는 노드 j 들의 집합 Ψ_i 를 생성하는 단계이다. [Step 7]은 현재 생성되는 유전자 $g[3]$ 이후의 유전자인지의 여부를 검색한다. [Step 8]의 1은 염색체의 생성과정에서 $g[3]$ 이후에 동일한 링크가 발생된 경우 이를 수정하기 위한 단계이다. 예를 들어, 단계적으로 $g[1]=1$, $g[2]=2$, $g[3]=1$, $g[4]=2$ 로 할당된 경우 $m=4$, $k=2$ 이고 $g[a]=g[3]$, $g[a+1]=g[4]$ 인 $a=1$ 이 존재한다. 따라서, $h=3$, $\Psi_{g[3]} = \Psi_{g[3]} - \{g[4]\} = \{2, 5\} - \{2\} = \{5\}$ 이다. $\Psi_{g[3]} \neq \emptyset$ 이므로 [Step 9]의 1에서 $g[4]=5$ 로 수정되어 동일한 링크가 중복되는 것이 방지된다. 또한, 위 초기화 절차는 노드들의 선행관계가 항상 만족되는 염색체를 생성한다.

2. 평가함수

본 연구에서는 목적함수인 기종점을 있는 경로의 총 소요시간을 평가함수로 사용하여 모집단을 구성하는 각 염색체(경로)에 대하여 해로써의 우수성을 평가하였다. n 개의 유전자로 구성되는 염색체 $A=(g[1] g[2] \dots g[n])$ 에 대해 평가함수 $eval(A)$ 을 계산하기

위해 다음과 같은 변수들을 정의하였다.

- $\cdot arrival(g[i])$: 염색체(경로) A 의 i 번째 유전자(노드)에 도착하는 시간
- $\cdot departure(g[i])$: 경로 A 의 i 번째 노드를 출발하는 시간
- $\cdot type(g[i], g[i+1])$: 경로 A 의 i 번째 노드와 $i+1$ 번째 노드를 잇는 링크의 유형(번호)
- $\cdot service(g[i], g[i+1])$: 경로 A 의 i 번째 노드와 $i+1$ 번째 노드사이의 차량운행소요시간
- $\cdot [l_i(g[i]), l_u(g[i])]$: 좌회전 허용을 위한 시간제한을 갖는 경로 A 의 i 번째 노드의 시간차
- $\cdot T = \{(4, 3), (9, 10), (14, 13), (16, 17), (22, 21)\}$
- $\cdot Q = \{(5, 12), (19, 25), (20, 24)\}$
- $\cdot S = \{(2, 6)\}$

여기서, 집합 T, Q, S 의 각 원소 (a, b) 는 링크번호 $a, b(a \neq b)$ 의 순서쌍을 나타낸다. 집합 T 의 각 원소는 링크번호 a 에서 링크번호 b 로 U-turn이 금지되는 경우를 나타내며, 집합 Q 와 S 의 각 원소는 링크번호 a 에서 링크번호 b 로의 좌회전 금지와 좌회전 허용을 위한 시간차가 존재하는 경우를 각각 나타낸다. 경로의 총 소요시간은 마지막 노드의 도착시간과 기점의 출발시점 사이의 차이이며 따라서 평가함수 $eval(A)$ 는 다음과 같이 정의된다.

$$eval(A) = arrival(g[n]) - departure(g[1])$$

마지막 노드의 도착시간 $arrival(g[n])$ 의 계산 과정은 다음과 같다.

Procedure : Evaluation

[Step 1]

$i=1$, $departure(g[1]) =$ 기점의 출발시점으로 초기화한다.

[Step 2]

- 만일 $i \neq n-1$ 이면 노드 $i+1$ 의 도착시간을 다음과 같이 계산하고 [Step 3]으로 이동한다.

$$arrival(g[i+1]) = departure(g[i]) + service(g[i], g[i+1])$$

2. 만일 $i=n-1$ 이면 마지막 노드 n 의 도착시간을 다음과 같이 계산하고 종료한다.

$$arrival(g[n]) = departure(g[i]) + service(g[i], g[i+1])$$

[Step 3]

1. 만일 $type(g[i], g[i+1]) = type(g[i+1], g[i+2])$ 이면 $departure(g[i+1]) = \infty$, $i=i+1$ 로 하고 [Step 2]로 재이동한다.
2. 만일 $type(g[i], g[i+1]) \neq type(g[i+1], g[i+2])$ 이면 [Step 4]로 이동한다.

[Step 4]

1. 만일 $(type(g[i], g[i+1]), type(g[i+1], g[i+2])) \in TUQ$ 이면 $departure(g[i+1]) = \infty$.
2. 만일 $(type(g[i], g[i+1]), type(g[i+1], g[i+2])) \in S$ 이고
 - ① $l_i(g[i+1]) \leq arrival(g[i+1]) \leq l_{i+1}(g[i+1])$ 이면 $departure(g[i+1]) = arrival(g[i+1])$.
 - ② $arrival(g[i+1]) < l_i(g[i+1])$ 또는 $arrival(g[i+1]) > l_{i+1}(g[i+1])$ 이면 $departure(g[i+1]) = \infty$.
3. 만일 $(type(g[i], g[i+1]), type(g[i+1], g[i+2])) \notin TUQS$ 이면 $departure(g[i+1]) = arrival(g[i+1])$.
4. $i=i+1$ 로 하고 [Step 2]로 재이동한다.

제안된 초기화 절차는 좌회전 금지, U-turn 금지로 인해 실제로 통행이 불가능한 경로를 생성할 수 있다. 예를 들어 초기화 절차에 의해 생성된 다음과 같은 경로들을 고려해보자.

$$A = (1 \ 2 \ 1 \ 5 \ 9 \ 13 \ 14 \ 15 \ 16)$$

$$B = (1 \ 5 \ 6 \ 5 \ 9 \ 13 \ 14 \ 15 \ 16)$$

$$C = (1 \ 5 \ 9 \ 10 \ 14 \ 15 \ 16)$$

$$D = (1 \ 2 \ 3 \ 7 \ 11 \ 15 \ 16)$$

여행자가 경로 A 와 B 를 이용하려면 노드 2와 노드 6에서 U-turn이 이루어져야 하고 경로 C 를 이용하려면 노드 10에서 좌회전이 이루어져야 한다. 그러나 노드 2와 노드 6에서의 U-turn 금지와 노드 10에서의 좌회전 금지로 인해 경로 A , B , C 는 통행 불가능하다. 경로 A 의 노드 2와 같이 특정 노드 전후의 링크번호가 동일하거나 경로 B 의 노드 6과 같이 특정

노드 전후의 링크번호 쌍이 집합 T 에 속하면 U-turn 금지에 위반되며 경로 C 의 노드 10과 같이 특정 노드 전후의 링크번호 쌍이 집합 Q 에 속하면 좌회전 금지에 위반되므로 통행 불가능하다. [Step 3]의 1과 [Step 4]의 1은 경로 A , B , C 와 같이 U-turn 금지나 좌회전 금지가 위반되는 노드가 발생하는 경우 해당 노드의 출발시간을 매우 큰 값(∞)으로 설정함으로써 결과적으로 경로 A , B , C 가 낮게 평가되어 이들 경로가 최단경로가 되는 것을 방지한다. 경로 D 의 노드 3과 같이 특정 노드 전후의 링크번호 쌍이 집합 S 에 속하면 좌회전 허용을 위한 시간차가 존재한다. [Step 4]의 2는 링크번호 2를 이용해서 노드 3에 도착한 시간이 노드 3에서의 좌회전 허용을 위한 시간차에 포함되지 않으면 노드 3의 출발시간을 매우 큰 값(∞)으로 설정함으로써 경로 D 가 최단경로가 되는 것을 방지하고 있다.

3. 선택

선택이란 현재의 모집단을 갱신시키기 위한 준비단계로써 지정된 전략에 따라 우수한 염색체를 다음세대로 복제(reproduction)시키거나 교배를 위한 염색체의 쌍을 만들기 위해 현재의 모집단을 재편성하는 과정이다. 대표적인 선택전략으로써 엘리트 보존전략(elitist strategy)과 roulette wheel 전략 등이 있다. 본 연구에서는 두 전략을 결합한 혼합전략을 사용하였으며 그 절차는 다음과 같다.

Procedure : Selection

[Step 1] 엘리트 보존전략에 의해 현 모집단 중에서 평가함수의 결과치(목적값)가 우수한 상위 $x\%$ 의 염색체는 교배나 돌연변이의 대상이 되지 않도록 다음세대로 복제한다.

[Step 2] 현 모집단으로부터 roulette wheel 전략에 의해 모집단의 부족한 수 $N = (pop_size - (pop_size \times x\%))$ 만큼 목적값에 비례한 확률로 선택해서 모집단을 재편성한다.

이 과정에서 각 세대의 가장 우수한 염색체(경로)가 다음세대에 유전되며 궁극적으로는 최적경로가 다음세대에서 제거되어지는 것을 방지할 수 있다.

4. 교배

교배는 재편성된 모집단으로부터 일정한 교배확률 p_c 에 따라 교배를 위한 염색체의 쌍을 만들어 각 염색체의 유전자 교환을 통해 모체보다 우수한 염색체를 생성시키기 위한 단계이다. 일반적인 교배연산자로는 일점교배, 다점교배, 일양교배가 있으며 그 외에도 PMX(Partially Mapped Crossover), OX(Order Crossover), CX(Cycle Crossover)등의 교배연산자가 다양한 문제에 적용되어져왔다. 본 연구에서는 상기의 일점교배 연산자를 제안된 염색체에 적용 가능하도록 수정하였다. 일점교배는 두 개의 염색체에 대해 교배위치를 임의로 한점 설정하고 후반의 유전자들을 교환한다. 본 연구에서는 기점노드와 종점노드를 제외한 두 염색체의 공통유전자(노드)를 교배위치로 선정하였다. 만일 두 염색체가 공통으로 소유하고 있는 유전자가 두 개 이상인 경우에는 그 중에서 임의로 하나의 유전자를 교배위치로 선정하였다. <그림 2>의 도로망에서 교배를 위해 선택된 한 쌍의 염색체 A , B 가

$$A=(1 \ 2 \ 3 \ 4 \ 3 \ 7 \ 11 \ 15 \ 16)$$

$$B=(1 \ 5 \ 6 \ 10 \ 11 \ 12 \ 16)$$

인 경우, 두 염색체의 공통유전자는 11이므로 유전자 11후반의 유전자들을 서로 교환하면 다음과 같은 새로운 염색체 A' , B' 가 생성된다.

$$A'=(1 \ 2 \ 3 \ 4 \ 3 \ 7 \ 11 \ 12 \ 16)$$

$$B'=(1 \ 5 \ 6 \ 10 \ 11 \ 15 \ 16)$$

염색체 A 와 B 의 공통유전자 11후반에 위치한 유전자들은 각각 노드 11에서 종점 노드 16까지의 선행관계를 만족시키는 노드들을 나타낸다. 따라서 유전자 11후반에 위치한 유전자들을 교환한다는 것은 노드 11에서 노드 16까지의 선행관계를 만족시키는 새로운 노드들을 노드 1에서 노드 11까지의 선행관계를 만족시키는 원래의 노드들에 연결시키는 것을 의미한다. 따라서 초기화 절차와 마찬가지로 교배 후 생성된 각 염색체에서 노드들의 선행관계가 만족된다. 두 염색체 A , B 의 공통유전자가 존재하지 않는 경우에는 교배가 이루어질 수 없기 때문에 $A'=A$, $B'=B$ 로

설정하였다.

5. 돌연변이

돌연변이란 각 염색체에 대해 일정한 돌연변이 확률 p_m 으로 모체의 성질과는 전혀 다른 새로운 염색체를 발생시키는 과정이다. 본 연구에서는 염색체를 구성하는 유전자들 중에서 임의로 하나의 유전자를 선정하고 선정된 유전자 후반의 유전자들을 <그림 3>의 선행관계 행렬 R 을 이용해서 재구성하였다. <그림 2>의 도로망에서 돌연변이를 위해 선택된 염색체가

$$B=(1 \ 5 \ 6 \ 10 \ 11 \ 12 \ 16)$$

이고 임의로 선정된 유전자가 6이라면, $i=6$ 으로 치환하여 선행관계 행렬 R 을 이용해서 유전자 6후반의 유전자들을 노드 6에서 노드 16까지의 선행관계를 만족시키는 새로운 노드들로 재구성한다. r_{67} 에서 $j=7$, r_{78} 에서 $j=8$, r_{12} 에서 $j=12$, r_{16} 에서 $j=16$ 이 선택되었다면 돌연변이가 수행된 후 생성되는 새로운 염색체 B' 는 다음과 같다.

$$B'=(1 \ 5 \ 6 \ 7 \ 8 \ 12 \ 16)$$

IV. 적용예제

<그림 2>에 예시된 도로망을 이용해서 제안된 알고리즘의 성능을 평가하였다. 제안된 알고리즘은 Borland C++ 5.02를 사용하여 구현되었으며 Pentium 500MHz CPU, 128MB RAM을 장착한 IBM-PC에서 수행되었다. 교배 확률(p_c)과 돌연변이 확률(p_m)은 실험적으로 각각 0.3, 0.2로 설정하였으며 모집단의 크기(pop_size)는 20, 상위 10%의 염색체만 엘리트 보존 전략에 의해 다음세대로 복제되도록 하였다. 최대세대수(max_gen)를 50으로 하여 알고리즘을 수행하였으며 그 결과를 전통적인 노드기반의 순차적 탐색법과 회절제한을 반영하기 위한 수정형 덩굴망 알고리즘과 비교하면 <표 1>, <표 2>와 같다. <그림 2>의 노드 3에서 좌회전을 의한 시간차 [7분, 10분]이 존재하기 때문에 기점을 출발한 시간이나 노드 3에 도착한 시간에 따라서 노드 7로의 좌회전 허용 여부가 결정된다. 따라서, 좌회전 허용이 이루어지는 시간대에 노드

3에서 노드 7로 향하는 경로를 이용하는 부분경로가 최단경로에 포함되는 도로망에서는 기점을 출발하는 시간대에 따라서 최단경로가 가변적으로 변한다.

〈표 1〉과 〈표 2〉는 기점의 출발시점이 0분인 경우와 4분인 경우에 대해 각 알고리즘에 의해서 탐색된 최단경로들을 나타낸다. 정적 최단경로 알고리즘인 순차적 탐색법과 수정형 덩굴망 알고리즘은 기점의 출발 시간과 관계없이 항상 동일한 경로 ①-②-③-⑦-⑪-⑮-⑯, ①-⑤-⑨-⑩-⑥-⑩-⑭-⑮-⑯를 각각 최단경로로 탐색한다. 특히, 경로 ①-②-③-⑦-⑪-⑮-⑯은 기점의 출발시간이 4분인 경우에는 통행 가능하지만 기점의 출발시간이 0분인 경우에 통행 불가능하다. 그러나, 제안된 동적 최단경로 알고리즘은 기점의 출발 시간대에 따라 가변적인 경로를 탐색한다. 기점의 출발시간이 0분인 경우에는 노드 3에서 노드 7로의 좌회전이 제한되기 때문에 회전제한이 표시된 도로망의 모든 링크에서 회전이 금지된다. 따라서, 현재시간 0분의 정적인 상태에서 도로망의 회전제한을 반영하는 수정형 덩굴망 알고리즘을 적용하면 합리적인 최단경로를 도출할 수 있으며 그 결과는 제안된 알고리즘에 의해 산출된 경로와 동일하다.

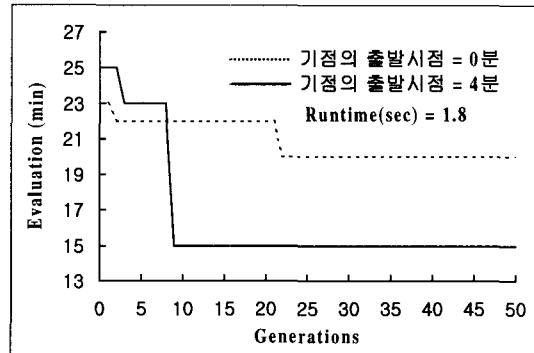
반면에, 기점의 출발시간이 4분인 경우에는 노드 3에서 노드 7로의 좌회전이 허용된다. 현재시간 4분의 정적인 상태에서 도로망의 회전제한을 고려하지 않는 순차적 탐색법을 적용하면 경로 ①-②-③-⑦-⑪-⑮-⑯가 도출되며 통행 가능하다. 현재시간 4분의 정적인 상태에서 도로망의 회전제한을 반영하는 수정형 덩굴망 알고리즘은 통행 가능한 경로 ①-②-③-⑦-⑪-⑮-⑯를 탐색과정에서 제외시키며 경로 ①-⑤-⑨-⑩-⑥-⑩-⑭-⑮-⑯를 최단경로로 탐색하지만 순차적 탐

〈표 1〉 알고리즘의 비교 (기점의 출발시점=0분)

알고리즘	최단경로	총 소요시간
순차적 탐색법	①-②-③-⑦-⑪-⑮-⑯	15분
수정형 덩굴망	①-⑤-⑨-⑩-⑥-⑩-⑭-⑮-⑯	20분
GA	①-⑤-⑨-⑩-⑥-⑩-⑭-⑮-⑯	20분

〈표 2〉 알고리즘의 비교 (기점의 출발시점=4분)

알고리즘	최단경로	총 소요시간
순차적 탐색법	①-②-③-⑦-⑪-⑮-⑯	15분
수정형 덩굴망	①-⑤-⑨-⑩-⑥-⑩-⑭-⑮-⑯	20분
GA	①-②-③-⑦-⑪-⑮-⑯	15분



〈그림 4〉 세대별 진화과정

색법에 의해 탐색된 경로보다 소요시간이 더 걸린다. 순차적 탐색법에 의해 탐색된 경로가 덩굴망 알고리즘의 경로 탐색과정에서 회전제한에 의해 탐색이 제외된 경로이지만 보다 우수하며 통행 가능한 경로라면 순차적 탐색법에 의해 도출된 경로는 최단경로라고 유추할 수 있다. 따라서, 순차적 탐색법에 의해 계산된 경로는 기점의 출발시간이 4분인 경우의 최단경로이며 그 결과는 제안된 알고리즘에 의해 산출된 경로와 동일하다. 결과적으로 제안된 동적 최단경로 알고리즘에 의해 탐색된 시간대별 경로들은 합리적인 최단경로로써 가변적이며 좌회전 허용을 위한 시간제한이 반영되어있다.

기점의 각 출발시점에 대하여 제안된 알고리즘의 수행결과를 세대별로 나타내면 〈그림 4〉와 같다. 〈그림 4〉의 결과는 세대별 가장 우수한 염색체의 평가치를 나타내며 해의 수렴속도가 우수할 뿐만 아니라 알고리즘의 수행시간이 1.8초로 매우 빠른 시간에 최단경로가 산출되는 것을 알 수 있다. 기점의 출발시점이 0분인 경우에 대하여 알고리즘 수행의 마지막세대의 모집단을 조사한 결과 평가치의 평균이 35분이었으며 최단경로외에도 ①-②-⑥-⑩-⑭-⑮-⑯(총 소요시간 22분), ①-⑤-⑨-⑩-⑥-⑦-⑧-⑫-⑯(총 소요시간 33분) 등의 경로들이 추가적으로 탐색되었다. 이들 경로들은 주변경관과 도로의 편의성 등을 고려해서 이용자가 자신에게 가장 적합한 대체 경로를 선택할 수 있는 정보로써 제공되어질 수 있다.

V. 결론

ATIS를 대도시의 복잡한 도로망에서 운영하기 위해서는 경로계산과정에 도로망의 기하구조적 특성뿐

만 아니라 가변적 특성을 반영할 수 있는 알고리즘의 개발이 요구된다. 지금까지 개발된 대부분의 알고리즘들은 도시도로망의 독특한 기하구조적 특성인 회전 제한만을 고려하기 때문에 현실성이 결여되어있다. 본 연구에서는 좌회전 금지와 U-turn이 허용되는 기하구조적 특성과 좌회전 허용시간이 제한되는 가변적 특성이 부여된 도시도로망에서 합리적인 최단경로를 신속하게 효율적으로 도출할 수 있는 해법을 개발하기 위해 유전자 알고리즘을 적용하였다. 제안된 알고리즘을 기존의 알고리즘과 비교한 결과 본 알고리즘의 효율성 및 우수성이 입증되었다. 본 알고리즘은 도시도로망에 대한 현실성을 잘 반영하고있을 뿐만 아니라 최단경로와 함께 제공되는 부가적인 경로들은 주변경관과 도로의 편의성 등이 고려된 다양한 경로정보로써 제공될 수 있기 때문에 ATIS를 대도시의 도로망에서 운영하기 위해 실제로 유용하게 활용될 수 있는 기법이다.

참고문헌

1. 김익기(1998), "ATIS를 위한 수정형 덩굴망 최단경로 탐색 알고리즘의 개발", 대한교통학회지, 제16권 제2호, pp.157~167.
2. 김현명·임용택(1999), "유전자알고리즘을 이용한 전역탐색 최단경로 알고리즘개발", 대한교통학회지, 제17권 제2호, pp.163~178.
3. 노정현·남궁성(1995), "도시가로망에 적합한 최단경로탐색 기법의 개발", 대한국토 도시계획학회지, 제30권 제5호, pp.153~168.
4. 이승재·장인성·박민희(2000), "유전 알고리즘을 이용한 이중목적 최단경로 모형개발에 관한 연구", 대한교통학회지, 제18권 제3호, pp.77~86.
5. 이승환·최기주·김원길(1996), "도시부 ATIS 효율적 적용을 위한 탐색영역기법 및 양방향 링크탐색 알고리즘의 구현", 대한교통학회지, 제14권 제3호, pp.45~59.
6. 장인성(2000a), "다목적 환경에서의 ATIS운영을 위한 A*탐색 알고리즘과 유전자 알고리즘의 혼합 모형", 대한산업공학학회지, 제26권 제4호, pp.421~430.
7. 장인성(2000b), "서비스시간 제약이 존재하는 도시부 복합교통망을 위한 링크기반의 최단경로탐색

알고리즘", 대한교통학회지, 제18권 제6호, pp.111~121.

8. 장인성·박승현(2001), "대중교통망에서의 최단 경로 탐색을 위한 유전자 알고리즘" 한국경영과학회 경영과학지, 제18권 1호, pp.107~118.
9. 장인성(2001), "서비스시간 제한이 있는 복합교통망에서의 경로안내시스템을 위한 유전자 알고리즘", 대한산업공학학회지, 제27권 제2호, pp.140~149.
10. 최기주(1995), "U-TURN을 포함한 가로망의 표현 및 최단경로의 구현". 대한교통학회지, 제13권 제3호. pp.35~31.
11. Caldwell, T.(1961), "On Finding Minimum Routes in a Network with Turn Penalties", Communications of the ACM, Vol. 4, pp.107~108.
12. Chen. Y. L. and Tang. K.(1997), "Shortest Paths in Time-Schedule Networks", International Journal of Operations and Quantitative Management, Vol. 3, pp.157~173.
13. Chen. Y. L. and Tang. K.(1998), "Minimum Time Paths in a Network with Mixed Time Constraints", Computers and Operations Research, Vol. 25, pp.793~805.
14. Desrochers, M. and Soumis, F.(1988), "A Reoptimization Algorithm for the Shortest Path Problem with Time Windows", European Journal of Operational Research, Vol. 35, pp.242~254.
15. Dijkstra, E. W.(1959), "A Note on Two Problems in Connection with Graphs", Numerische Mathematik, Vol. 1, pp.269~271.
16. Lawler, E. L.(1972), A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem, *Management Science* 18, 401~405.
17. Michalewicz, Z.(1994), "Genetic Algorithm +Data Structure=Evolution Programs", Springer-Verlag.
18. Sheffi, Y.(1985), "Urban Transportation Networks", Prentice-Hall.

19. Thomas, R.(1991), "Traffic Assignment Techniques", Avebury Technical.
20. Winston, P. H.(1992), "Artificial Intelligence", Addison-Wesley.
- ✻ 주 작성 자 : 장인성
- ✻ 논문투고일 : 2001. 4. 17
- 논문심사일 : 2001. 5. 31 (1차)
2001. 6. 14 (2차)
2001. 7. 13 (3차)
- 심사판정일 : 2001. 7. 13