■ 論 文 ■

유전 알고리듬을 이용한 전역탐색 최단경로 알고리듬개발

Development of a Global Searching Shortest Path Algorithm
by Genetic Algorithm

김 현 명
((주)도화종합기술공사)

임 용 택
(서울특별시 교통관리실 교통연구팀장)

목 차

Ⅰ. 서론
Ⅱ. 탐색기법에 따른 최단경로 알고리즘
1. 통합교통망의 탐색영역문제
2. 최단경로 알고리즘의 분류
Ⅲ. 전역탐색 최단경로 알고리즘의 개발
1. 유전 알고리듬을 이용한 경로구축
2. 교통망 변환
3. 경로개선 알고리듬
4. 불필요한 링크삭제
5. 전역탐색 최단경로 알고리듬
Ⅳ. 모형의 평가
1. 탐색영역문제 (Ⅰ)
2. 탐색영역문제 (Ⅱ)
Ⅴ. 분할탐색법과 전역탐색법(GA) 비교평가
Ⅵ. 결론 및 향후연구
참고문헌

요 약

교통분야에서 이용되는 최단경로 알고리듬은 분할탐색 기법에 기초를 두고 있다. 분할탐색 기법이란 기점으로부터 일정 영역을 분할하여 경로를 탐색, 종점까지의 경로를 구축하는 방법으로써 수형망(Tree Building) 알고리듬이나 덩굴망(Vine Building) 알고리듬 등이 여기에 속한다. 그러나 이러한 분할탐색기법의 경우 교통망내에서 복수 수단간의 환승비용이 고려될 경우나 동적 최단경로를 탐색하는 경우에는 교통망을 확장하지 않으면 기종점간의 올바른 최단경로를 찾을 수 없다는 문제점을 가지고 있다. 이러한 문제를 본 연구에서는 탐색영역 문제(Searching Area Problem)라고 정의하였다.

본 연구에서는 탐색영역문제를 교통망 확장없이 해결할 수 있는 전역 탐색기법으로 유전 알고리듬을 이용하여 개발하였다.

Ⅰ. 서론

최단경로 탐색 알고리듬은 통행배정 모형의 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. 그러나 현재 교통분야에서 이용되는 전통적인 최단경로 알고리듬은 주로 산업공학에서 개발된 수형망(Tree Building) 알고리듬에 그 기초를 두고 있어 교통망에 존재하는 여러 문제들을 고려해 올바른 경로를 탐색하는데 한계를 가지고 있다.

현재까지 교통망의 최단경로 탐색에서 연구된 주요 문제는 회전 금지나 회전 벌점을 고려할 수 있는 최단경로 알고리듬 개발 분야였다. Dijkstra 알고리듬(1959)으로 대표되는 수형망 알고리듬의 경우 좌회전, 우회전, P-turn, U-turn 문제를 최단경로 탐색과정에서 효율적으로 반영하지 못하기 때문에 덩굴망(Vine Building) 알고리듬이 개발되었다.

그러나 최근 들어 다양한 교통수단들이 하나의 교통망에서 함께 고려되는 통합교통망(Integrated Transportation Network) 수요추정 문제가 대두되면서 수단간의 환승을 고려해 최단경로를 탐색하는 문제가 새로운 연구분야로서 주목받고 있다. 통합교통망에 있어 기존의 최단경로 알고리듬을 적용할 수 없는 이유는 수단간의 환승문제가 발생하기 때문이다. 수단간 환승비용을 고려하는 문제는 일부 노드에서만 발생하는 회전 벌점 문제와는 달리 분할 탐색법을 이용하기 힘들기 때문에 교통망을 확장하는 방법이 주로 개발되어왔다(최기주 외, 1998). 그러나 대규모 교통망에 여러 교통수단과 노선이 존재하는 통합 교통망의 특성상 각 수단별로 노드와 링크를 확장하는 기존기법은 계산비용이 크고 교통망 확장에 많은 시간이 소비되는 단점을 가지고 있다.

이에 비해 최근 연구가 시작된 전역탐색 기법(Global Searching Technique)은 교통망의 확장 없이 수단간 환승비용을 고려해 최단경로를 찾을 수 있는 기법이다. 본 연구에는 전역해 탐색기법의 하나인 유전알고리듬(Genetic Algorithm)을 이용해 수단간 환승비용(Transfer Cost)에 의해 발생하는 탐색영역 문제를 해결할 수 있는 전역 탐색 최단경로 알고리듬을 개발하고 적용예를 예시하였다. 또한, 몇가지 측면에서 현재 널리 사용되고 있는 네트워크 확장방법과 비교하였다.

Ⅱ. 탐색기법에 따른 최단경로 알고리듬

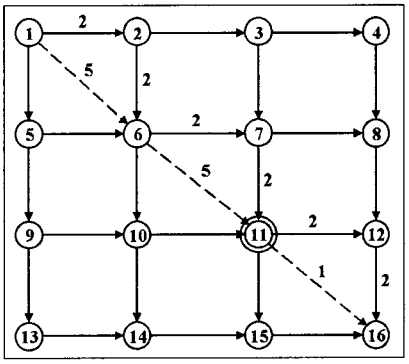
최단 경로 알고리듬은 탐색기법에 따라 분할 탐색 알고리듬(Partial Searching Algorithm)과 전역탐색 알고리듬(Global Searching Algorithm)으로 구분할 수 있다. 분할 탐색기법의 경우 계산이 편리한 장점이 있으나 수단간 환승비용이 존재하는 경우 정확한 경로와 비용을 찾기 힘들며 수단간 환승비용을 고려하려면 각 수단에 연결되는 모든 노드들을 확장해야 하는 단점이 있다.

이에 비해 전역탐색 알고리듬은 계산 비용이 큰 대신에 가로망 전체의 비용을 분할없이 탐색하기 때문에 수단간의 환승비용을 고려해 최단경로를 구축할 수 있는 장점을 가지고 있다. 이하 내용에서는 전통적인 분할탐색법을 통합 교통망에 적용할 때 발생하는 탐색영역 문제와 현재까지 개발된 전역탐색 알고리듬에 관해 살펴보도록 한다.

1. 통합 교통망의 탐색영역 문제(Searching Area Problem)

통합 교통망이란 성질이 상이한 링크들에 의해 구성된 교통망을 뜻한다. 즉, 통합 교통망에서는 링크에서 링크로 진행할 때 추가 비용을 지불하게 되며, 여러 수단이 함께 고려되는 교통망에서는 수단전환(환승)에 따른 비용이 추가된다.

탐색영역 문제란 통합 교통망에서 수단의 전환을 경로 구축시 반영하지 못함으로서, 최종적으로 선정된 최단경로가 실제 최단경로가 아니거나, 경로의 비용이 잘못 계산되는 문제를 의미한다. 탐색 영역문제의 기본개념은 〈그림 1〉과 같다.



주) 링크상의 값은 링크통행비용을 표시하며, 이외 링크의 통행비용은 매우 큰 값으로 가정

〈그림 1〉 최단경로 탐색과정에서의 탐색영역 문제

〈그림 1〉에서는 노드 1에서 노드 16까지 도달하는 최단경로를 찾는 단계를 나타낸 것이다. 여기서 점선은 지하철 링크이고 실선은 도로이다. 수단간 환승비용은 도로에서 지하철로의 환승비용이 4, 지하철에서 도로로의 환승은 불가능하다고 가정하자. 수형망(Tree) 알고리듬의 경우 전 노드까지만 저장하고 경로는 기억하지 않으므로 환승을 고려해 위의 문제를 풀 수 없다. 따라서 전전 노드를 기억할 수 있는 덩굴망(Vine) 알고리듬을 이용, 환승비용을 고려해 노드 1에서 16까지의 최단경로와 비용을 계산해 보도록 하자. 현재 도착비용을 계산해야할 노드가 16이라 할 때 덩굴망 알고리듬이 고려하는 링크와 노드의 조합 범위는 〈그림 1〉의 음영부분이다.

그림에서 보듯이 노드1에서 노드16까지 최단경로가 1→2→6→7→11→12→16번노드로 구성되므로, 승용차로 16에 도착하는 경우, 출발노드는 12, 경로비용은 12가 된다. 노드15의 경우 문제를 단순화하기 위해 최단경로에서 제외하고, 지하철을 이용하여 노드16에 도착하는 경우, 11번 노드가 연결링크의 출발 노드가 되며, 전 노드는 7인 경우와 6인 경우로 나누어지고 이들은 각각 다른 수단을 이용하여 경로를 구성하게 된다. 전 노드가 7인 경우 7까지의 비용은 6, 링크(7,11)의 비용은 2, 노드 11에서의 환승비용은 4, 링크 (11,16)의 비용은 1로 16에 도착하기 위한 비용은 13이다. 전 노드가 6인 경우에는 노드 6까지의 비용 4, 링크 (6, 11)의 비용은 5, 링크(11, 16)의 비용은 1 이므로 16까지의 도착비용은 10이 된다. 전 노드가 6인 경우에는 링크 (6, 11)과 (11, 16)이 동일한 수단을 이용하므로 환승비용은 없다. 따라서, 노드 1에서 16까지의 최단비용은 10이다. 이때 문제는 비용이 10인 경로는 가로망상에 존재하지 않는다는 것이다. 10이라는 비용은 노드 6으로부터 출발했을 때의 비용이기는 하지만 이는 노드 6에 도로를 이용해 도착해 노드 6까지의 비용이 4인 경우이고, 노드 6 이후 지하철을 이용하였다면 노드 6에서 환승비용을 더한 비용, 즉 6의 최소비용은 8로 인식했어야 한다. 그러나 기존의 덩굴망 알고리듬은 이런 수단 전환의 영향을 고려하는데 영역적으로 제한되기 때문에 잘못된 비용을 인식하게되며, 경로 역시 1-2-6-11-16으로 저장되어 노드 6 이전의 경로는 수정하지 못한다. 이것이 최단경로 탐색에 있어서의 탐색영역 문제(Searching Area Problem)이다.

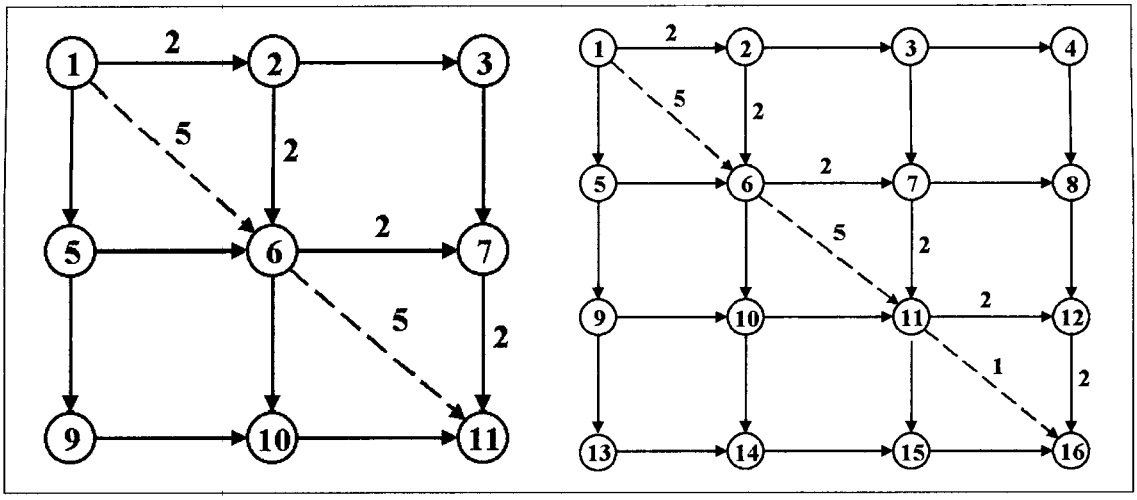
탐색영역 문제는 최단경로 탐색기법이 갖는 기본적인 가정이 침해되기 때문에 발생한다. 순차적 최단경로 탐색은 다음과 같이 선형문제로 구성되며 이 문제를 기점부터 종점까지 순차적으로 풀어 최단경로를 구하게 된다.

$$\begin{aligned} &\text{If } \text{Node_Mincost}(I) + \text{Link_cost}(I, J) \\ &\quad < \text{Node_Mincost}(J) \\ &\text{Then, } \text{Node_Mincost}(J) \\ &\quad = \text{Node_Mincost}(I) + \text{Link_cost}(I, J) \end{aligned} \quad (1)$$

여기서, 노드I, J ∈Node Set이며 Node_Mincost(I)는 기점(Origin)노드에서 노드I까지의 최소경로비용을, Link_cost(I, J)는 노드 I와 J간의 링크 통행비용을 나타낸다.

위의 문제에는 두 가지 가정이 있다. 링크 비용이 (-)이 아니며 확정된 Node_Mincost(I)는 탐색 범위의 확장과 관련 없이 고정적이라는 것이다. 이 두 가지 가정이 만족되면 경로 탐색문제는 선형 최적화 문제가 되며, 노드로 나누어지는 각 계산단계 별로 선형문제를 풀어 이를 결합하면 전역적인 최단경로는 얻을 수 있는 것이다. 그러나 수단간 환승이 고려되는 경우 교통망의 크기가 확장됨에 따라 최단경로상의 노드 도착비용은 상수가 아닌 변수가 되고 이 경우 위 선형문제를 풀어서는 최적 경로를 구축할 수 없다.

〈그림 2〉와 같이 환승노드 6과 11을 갖는 네트워크를 생각해보자(환승비용은 각각 4). 환승비용을 고려하지 않으면, 교통망이 확장되더라도 노드 11까지의 도착비용은 8로 변화가 없다. 그러나 (그림 1)에서와 같이 환승비용을 4라고 가정하면 확장 전 범위에서 노드 11의 도착비용은 8이지만, 확장 후에는 노드 11까지 10의 비용을 갖는 대각선 경로(1→6→11)를 이용해야만 16에 도착하는 비용이 11로 최소가 될 수 있다. 11번 노드에서 환승을 하는 경우, 통행비용은 8(노드1에서 11번까지 최단경로비용)+4(환승비용)+1(노드11에서 노드16까지 비용)=13이 된다. 따라서 단계별 최적경로를 따라 탐색을 진행하면 전역적 최적경로를 찾을 수 없게 된다.



〈그림 2〉 교통망 확장에 따른 최적 노드 도착비용 변화

이러한 탐색영역의 문제가 3차원으로 확장되면 동적 최단경로 탐색에서의 영역문제가 된다. 즉, 주어진 Time Horizon에서 순차적으로 최단 경로를 탐색하며 진행하면 도착 후 다른 경로에 비해 더 큰 비용을 지불하게 되는 것이다. 이 문제 역시 시간적으로 탐색범위가 확장되는데 따라 각 노드의 최적 도착비용이 교통망의 종점, 또는 범위에 따라 임의로 변화하기 때문이다. 이러한 최단경로의 성질은 현실상에서 최단경로를 탐색하는 경우 흔히 나타나는 문제이며, 결국 탐색영역 문제가 발생하는 교통망에서 순차적 탐색법으로 경로를 구축할 경우 전역적인 최단경로를 구축할 수 없게 되는 것이다.

탐색영역문제는 정확한 노드 도착비용을 계산하는 문제이기 때문에, 전통적인 탐색기법을 통해 위 탐색영역 문제를 해결하기 위해서는 각 노드에서 이용 가능한 수단만큼의 노드 확장이 필요하다. 만약 통합교통망에서 수단별로 노드를 확장하지 않고 수형망 알고리듬을 적용해 경로를 탐색한다면, 위와 동일한 오류를 범하게 된다.

2. 최단경로 알고리듬의 분류

최단 경로의 탐색 방법이란 최단경로의 비용을 계산하는데 기본이 되는 값, 즉 각 노드의 비용을 어떻게 결정하는가를 의미한다. 현재까지는 각 노드별로 최단경로를 결정하고 그 비용을 통해 종점까지 경로를 구축해가는 분할 탐색(Partial Searching) 또는 순차적 탐색법(Sequential Searching)을 이용해왔다. 본 연구에서는 분류의 편의상 이러한 탐색법을 분할 탐색법이라 정의하였다.

분할 탐색법의 특징은 한번 최단경로를 통해 노드 도착비용이 고정되면 그 비용은 최소비용으로 고정되고, 기점으로부터의 경로대신 이전 노드만을 순차적으로 저장해 경로를 구축한다는 것이다. 따라서 각 노드에서의 최단경로 탐색과정이 독립된 단계라고 한다면, 독립된 단계별 최적해를 저장하고 이를 산술적으로 조합해 전체 교통망의 최단경로를 구축한다.

즉, 분할 탐색법은 통과노드를 중심으로 단계별 최적경로를 찾아 이를 산술적으로 연결하면 최단경로가 된다고 가정하는 것이다. 따라서 탐색영역 문제와 같이 이 가정이 성립되지 않는 문제에서는 최단경로를 찾을 수 없다. 이러한 문제는 동적 최단경로 탐색과정에서도 동일하게 나타난다. 일정한 시간간격별로 종점까지의 최단경로를 계산해 이를 더하는 경우에도 한정된 정보에 의해 계산된 국지적 최단경로의 산술적 합이 전역적인 최단경로가 될 수 없기 때문이다.

이러한 탐색영역 문제를 해결할 수 있는 방법으로는 전역 탐색법이 있다. 전역 탐색법은 각 노드별 비용을 계산해 단계별로 최단경로를 계산하는 것이 아니라 전체 기·종점간의 경로비용만을 비교해 최단경로를 탐색한다. 따라서 전역 탐색법은 전역적 최적화 기법을 이용해 경로를 구축하며 순차적인 구축단계 없이 복수의 경로를 단계마다 발생시키고 이를 비교해 경로를 개선시키는 방법을 이용한다. 아래 내용에

서는 분할 탐색법과 전역 탐색법의 종류 및 그 특징에 대해 살펴보도록 하자.

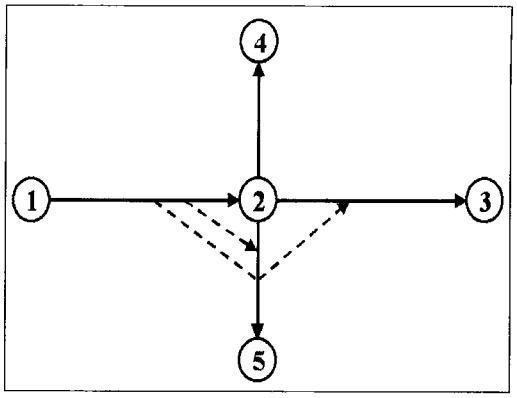
1) 분할 탐색법(Partial Searching Method)

분할 탐색법은 현재까지 개발된 대부분의 최단경로 탐색 알고리듬이 채택하고 있는 탐색법이다. 분할 탐색법이 이용되는 분야는 교통망의 유형에 따라 두 가지로 나눌수 있는데 첫 번째는 교통망 확장이 필요없는 경우 이용하는 수형망(Tree Building) 알고리듬과 덩굴망(Vine Building) 알고리듬 등 전통적 알고리듬이며, 두 번째는 교통망의 확장을 고려해야 하는 통합 교통망에서의 탐색 알고리듬이다.

수형망(Tree)이란 한 경로에서 순환(Cycle)없이 모든 마디가 다른 모든 마디와 서로 연결된 그래프를 뜻하며(임강원, 1992), 대표적인 예로는 Dijkstra 알고리듬(1959)이 있다. 수형망 알고리듬은 노드와 연결링크간의 관계를 이용해 경로를 구축하기 때문에 경로 구축시 고려되는 범위가 좁다는 점에서 단순 교통망에서는 효율적이나 경로구축과정에서 전 노드(Back Node)만을 기억하는 관계로 회전 제약과 같은 문제를 해결할 수 없는 단점이 있다.

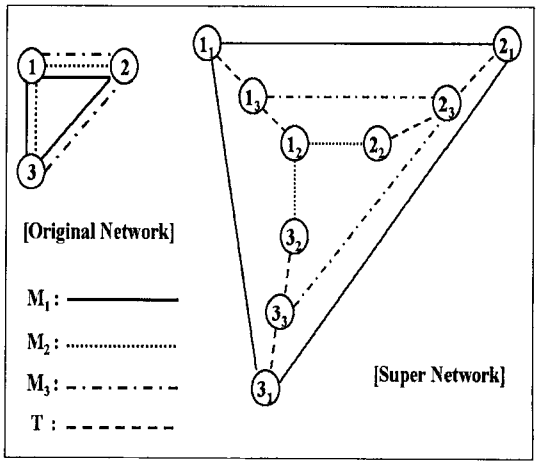
이에 비해, 덩굴망 알고리듬은 최소 도착비용을 결정해야 하는 노드의 전 노드와 전 링크, 전전노드(Back-Back Node)까지를 검색범위로 가지기 때문에 노드와 노드간의 관계뿐 아니라 제한된 범위내에서 링크와 링크간의 연결관계도 고려할 수 있다. 따라서 회전 제한 문제를 쉽게 해결할 수 있다. Moore 알고리듬, RRL 알고리듬, Shimbel 알고리듬이 여기에 속한다. 이밖에 연속된 회전제한 문제를 풀기 위해 개발된 Bi-Directional Dijkstra 알고리듬(이승환 외, 1996)의 경우 노드 비용을 저장하지 않고 링크 종점의 비용을 저장하기 때문에 연속적인 회전제약 문제를 해결할 수 있다. 하지만 이 경우 역시 제한된 범위의 비용만을 고려하기 때문에 분할 탐색 알고리듬의 일종이라 할 수 있다.

지금까지 살펴본 덩굴망 알고리듬은 교통망에서 링크와 링크간의 연결성이나, 경로의 링크 및 전전 노드의 정보등을 이용 좌회전 금지를 인식시켰다. 하지만 덩굴망 알고리듬을 이용하는 대신 링크의 단순한 확장만으로 좌회전 금지문제를 해결할 수 있는 방법이 있는데 이를 Reversed Graph라 한다.(Barra, 1993)



〈그림 3〉 Reversed Graph에 의한 좌회전 금지 표현

〈그림 3〉에서 링크 (1, 2)는 좌회전이 금지되어 있다. 이때 링크 (1, 2)와 우회전 방향의 링크 및 직진 방향의 링크를 연결한 점선 화살표를 Reversed Graph라하며, 경로 탐색시에는 Reversed Graph로 연결된 링크만이 다음 링크가 될 수 있다. Reversed Graph는 간단한 확장을 통해 수형망 구축 알고리듬에서도 좌회전 금지를 표현할 수 있다는 장점이 있으나 링크 확장을 통해 링크와 링크간의 연결성을 고려하므로 Reversed Graph로 연결된 전 링크 이외의 경로는 비용 결정시 고려할수 없고 따라서 탐색영역 문제를 해결할 수는 없다. 결국 분할 탐색법을 이용하는 모든 최단경로 알고리듬으로 탐색영역 문제를 해결하기 위해서는 노드의 확장이 필요하며 그 예는 P. Modesti and A. Sciomachen(1998)의 연구에서 아래 〈그림 4〉와 같이 검토된 바 있다.



〈그림 4〉 Node 확장을 통한 탐색 영역 문제의 해결

P. Modesti and A. Sciomachen(1998)은 위 그림과 같이 복수 통행수단이 존재하는 교통망에서 노드확장을 통해 분할 탐색법으로 탐색영역문제를 극복할 수 있는 방법을 제시하였다. 그림에서 M_1은 승용차, M_2는 대중교통수단, M_3는 도보를 의미한다. 확장된 가로망에서 추가된 링크 T는 환승링크로서 수단별로 확장된 노드간을 연결한다. 이 방법의 특징은 링크뿐 아니라 노드까지 수단 수만큼 확장함으로써 수단간의 환승까지 완전하게 묘사 가능하다는 점이다. 노드확장 기법의 또 다른 예는 김익기(1998)에서 제시한 U-turn, P-turn, 좌회전 금지를 고려하는 최단경로 탐색 알고리듬에서 볼 수 있다. 이 방법은 각 노드별로 유입 방향을 저장함으로써 노드를 유입방향만큼 확장한다. 따라서 각 노드에서 방향별로 노드를 확장하기 때문에 회전에 관련된 사항을 고려 할 수 있다.

최근 노드확장을 통한 분할 탐색법과는 달리 수단별 교통망을 계층적으로 구성하고 이를 분할 탐색하여 최단경로를 결정하는 알고리듬이 최기주 외(1998)에 의해 제시되었다. 이 방법은 환승이 이루어지는 지점마다 수단별 교통망의 노드간에 연결성을 부여하는 기법으로서 앞에서 본 노드 확장 기법과 같은 개념으로 탐색영역 문제를 극복한다고 할 수 있다.

그러나 노드 확장이나 계층적 교통망을 통한 분할 탐색법은 교통망이 소규모이고 고려되는 수단이 간단할 경우에는 문제가 없으나 대규모 교통망에서 버스와 같은 다수의 노선들을 고려해야하는 경우에는 적용이 상당히 어려운 문제가 있다. 이런 기법들은 동적 최단경로 탐색에도 이용할 수 있으나 이 경우 각 링크의 노드가 각 Rolling Horizon의 개수만큼 확장되어야 한다. 즉, 단순한 두 링크간에도 (Rolling Horizon의 개수)*(Rolling Horizon의 개수) 만큼의 노드 확장이 필요하다.

2) 전역 탐색법(Global Searching Method)

전역 탐색법이란 경로를 구축하는 과정이 단계별로 이루어지는 것이 아니라 경로 전체에 걸쳐 동시에 이루어지는 탐색법이다. 전역탐색법에서는 노드별 최소 도착비용이라는 개념을 이용하지 않고 종점까지의 도착 비용만 저장한다. 따라서 분할 탐색법과 같은 순차적 비용고정과정은 없으며 전역적 최적화 알고리듬을 이용하여 반복계산을 통해 종점 도착비용이 가장 낮은 경로를 탐색한다. 전역 탐색법은 분할 탐색법에 비해 계산량이 많고 계산시간이 긴 단점이 있는 대신 동적·정적인 모든 경우에 있어 가로망의 확장 없이 탐색영역 문제를 해결할 수 있다.

현재까지 개발된 전역 탐색법은 G. Zhou and M. Gen(1999)의 경우가 대표적이다. M. Gen은 전역탐색을 위해 확률적 전역 탐색기법인 유전 알고리듬(Genetic Algorithm)을 이용하였다. 경로구축시 유전알고리듬을 적용하려면 경로를 유전자(Chromosome)로 표현해야 하는데 M. Gen은 이를 위해 Prufer number라는 개념을 도입하였다. Prufer number란 순차적 순열로 교통망의 노드간 관계를 나타내는 방법으로 각 기종점 집합과 중간노드 집합을 순차적으로 배열함으로서 기종점간의 경로를 표시하는 방법이다. 이 방법을 통해 유전자는 미리 지정한 개수대로 임의의 경로를 저장한다. 저장된 경로들은 전통적 방법으로 비용을 계산하게 되고, 경로비용이 낮은 순서대로 정렬된 후 교배, 돌연변이 과정을 반복한다. 반복 계산은 일반적으로 미리 지정한 횟수만큼 계속하거나 전 유전자의 저장 경로가 같아질 때까지 계속된다. 이때, 저장되는 경로는 수송문제에 의해 구축된 경로이므로 모든 노드를 한 번씩은 방문해야 하고 유전자내의 염색체수는 항상 교통망의 노드수와 동일하다.

G. Zhou and M. Gen(1999)이 제시한 전역 탐색 알고리듬은 경로를 효과적으로 유전자로 표현함으로써 유전 알고리듬을 최단경로 문제에 적용할 수 있도록 하였으나 수송 문제(Transportation Problem)을 다루고 있기 때문에 노드를 순서대로 저장하는 것만으로 경로가 이루어진다. 그러나 최단경로 알고리듬의 경우 노드와 노드는 링크에 의해서만 연결이 가능하므로 노드만을 저장할 경우 연결링크가 없는 연속 노드가 저장되어 경로가 이루어지지 않을 수 도 있고 수송문제와는 달리 모든 노드를 지나지 않을 수 도 있기 때문에 각 유전자마다 포함되는 노드수가 다를 수 있다. 이 경우 G. Zhou and M. Gen(1999)이 제시한 교배방법은 이용할 수 없다. 따라서 G. Zhou and M. Gen(1999)의 연구는 유전 알고리듬을 경로 구축과정에 적용한 의의는 있으나 최단경로 구축과정에는 적합하지 않다.

유전 알고리듬을 통한 전역 탐색법이 교통망에서 탐색영역 문제를 극복할 수 있는 이유는 반복 계산을 통해 교통망에서 나타날 수 있는 대부분의 경로를 검색하기 때문이다. 초기에 임의로 발생된 경로는 그

비용에 있어 분할 탐색법보다 클 수 있으나 반복적으로 경로를 교배하는 과정에서 여러 경로의 구간들이 합성되므로 개선을 거듭해 전역 최단경로를 도출하게 된다.

Ⅲ. 전역탐색 최단경로 알고리듬의 개발

1. 유전 알고리듬을 이용한 경로 구축

1) 유전 알고리듬

유전 알고리듬(Genetic Algorithm)은 자연세계의 진화현상에 기초한 계산모형으로서 전역적(Global)인 최적화 기법이다. 유전 알고리듬은 자연 선택의 원리와 자연계의 생물 유전학에 기본을 둔 병렬적이고 전역적인 탐색 알고리듬으로서, 모든 생물은 주어진 다양한 환경속에 적응함으로써 살아남는다는 Darwin의 적자생존(Survival of the Fittest)의 이론을 기본 개념으로 한다. 유전알고리듬은 풀고자하는 문제에 대해 가능한(Feasible) 해들을 정해진 형태의 자료구조로 표현하여 이들을 점차적으로 갱신함으로서 좀 더 좋은 해들을 만들어 낸다.

유전 알고리듬은 재생산(reproduction), 교배(crossover), 돌연변이(mutation)의 세 가지 과정을 반복하여 최적해를 도출하게 된다. 여기서 재생산이란 지정된 전략에 따라 새로운 세대를 발생시키는 것이고 교배란 개체간의 염색체 교환을 통해 새로운 세대를 형성하는 것이다. 돌연변이란 모체의 성질과는 전혀 다른 새로운 개체를 발생시키는 과정이다. 돌연변이 확률이 높을수록 광범위한 영역을 탐색할 수 있어 국지해(Local Solution)에 수렴할 가능성을 낮출 수 있지만 수렴속도는 늦어지게 된다. 본 모형에 적용한 유전 알고리듬은 재생산 규칙으로는 엘리트 보존방법과 랭킹방법을 결합하여 사용하였다. 이 방법은 목적함수 값이 우수한 상위 개체의 일정비율을 다음세대에 그대로 복제하는 방법이다.

2) 염색체를 이용한 링크의 표현

유전 알고리듬을 최적화 문제에 이용하는 경우 가장 중요한 단계는 최적 문제의 변수를 유전자형(Genotype)으로 어떻게 표현하는가에 있다. 본 연구의 최적화 문제는 최소의 비용으로 기종점 간을 잇는 것이므로 각 개체(또는 염색체)들은 유전자(Gene)를 이용해 기종점간 경로를 나타낼 수 있어야 한다. 유전 알고리듬이 경로 탐색 문제에 이용된 예는 주로 수송문제(Transportation Problem)이며 이 경우 연결되어야 할 각 기종점 수만큼의 행렬을 구성, 각 열과 행에 임의의 링크들에 의한 연결비용을 발생시키고 총 수송비를 계산하는 과정을 반복 수행하여 최적해를 얻는다. 그러나 수송문제의 경우 단일한 기종점 쌍 간의 최단경로를 찾는 문제가 아니라 복수의 기종점간을 연결하는 경로를 탐색하는데 그 목적이 있다. 본 연구에서는 각 염색체의 유전자를 가로망의 링크 수 만큼 확보하고, 링크 이용값을 임의로 발생시키는 방법을 선택하였다. 예를 들어 가로망의 링크가 12개일 경우 각 유전자는 〈표 1〉과 같이 구성할 수 있다.

여기서, 각 링크는 유전자를 이용하여 다음과 같이 염색체로 표현된다.

$X_n[m][l]$: n 세대 m 번째 유전자의 l 번째 링크 이용 값

n : 세대(generation)

m : gene number ($1 \le m \le$ *population*수)

l : link number

유전 알고리듬을 이용한 최단경로 탐색법이 지금까지 개발된 알고리듬들에 비해 효율적인 이유는 노선수가 증가되거나 동적인 상황에 적용되더라도 주어진 교통망을 확장할 필요가 없다는데 있다. 기존에 이용되는 알고리듬을 복수

〈표 1〉 Gene and Chromosome design의 예

Genes	LINK Number(l)											
	1	2	3	4	5	6	7	8	9	10	11	12
Chromosome1 $X_1[1][l]$	0.12	0.28	0.54	0.15	0.65	0.82	0.12	0.02	0.89	0.29	0.54	0.60
Chromosome2 $X_1[2][l]$	0.33	0.25	0.79	0.98	0.25	0.11	0.54	0.68	0.42	0.88	0.12	0.25
.....												

수단이나 복수 노선이 존재하는 교통망에 적용할 경우 노드나 링크가 수단과 노선을 묘사할 수 있도록 확장하는 과정이 필요하다. 또, 동적 통행배정 모형에서 최단경로를 탐색하기 위해서는 주어진 교통망을 시간에 따라 Time-Space Network의 형태로 수정해야 한다. 그러나 유전 알고리듬을 이용할 경우 교통망의 확장대신 아래와 같이 염색체의 저장공간을 확장함으로써 문제를 간단히 해결할 수 있다.

$$X_n[m][l][Mode][Line][Time\ slice] \quad (2)$$

수단(*Mode*),노선(*Line*),시간대(*Time slice*)를 갖는 n 세대 l 번째 링크의 m 번째 염색체 값

본 연구의 경우 유전 알고리듬을 공간적인 최적해 문제 해결에 이용가능한지를 살펴보는데 연구의 주 목적이 있기 때문에 위의 예와 같은 복잡한 형태의 염색체는 이용하지 않았다. 향후 대중교통 통행배정 문제나 동적 노선안내 모형분야에서는 위와 같은 형태의 염색체를 적용할 수 있을 것이다.

2. 교통망 변환

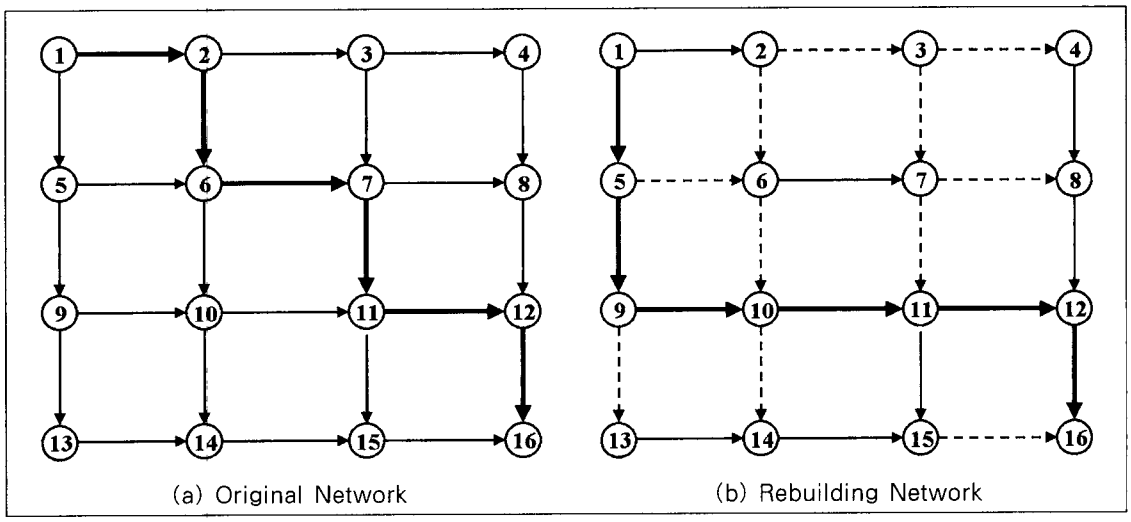
유전 알고리듬을 적용하기 위해서는 유전자 값을 0~1 사이에서 임의로 발생시켜 초기화시켜야 한다. 〈표 1〉과 같이 염색체내의 각 유전자가 0~1사이의 값을 부여받으면 이 값을 이용해 경로를 구축하게된다.

우선 다양한 경로 발생을 위해 유전자·염색체 공간에 가상 유전자들을 발생한 후 미리 지정된 값(0.5) 이상인 유전자들에 대해서는 1의 값을 그 이하의 유전자에는 0의 값을 가상 유전자에 저장한다. 이 과정을 전 염색체가 거치면 각 염색체들을 1이나 0의 값을 갖는 집합이 된다.

위의 변환과정을 거친 염색체들에 대해 각 유전자별로 값이 1이면 해당 번호 링크의 통행비용을, 값이 0이면 매우 큰 값을 갖는 가상비용을 부여한다. 이러한 과정을 거칠 경우 Original 교통망은 실제 비용을 가지는 링크(Real Link)들과 가상 비용을 가지는 링크(Dummy)로 나누어진다. 이렇게 각 염색체 별로 재구성된 교통망에서 기존 분할 탐색법(Dijkstra Algorithm)을 이용해 기 종점간 최단 비용을 계산한다.

위와 같은 방법으로 교통망의 링크를 Real Link와 Dummy Link로 구분한 뒤 분할 탐색법을 실시하면 각 염색체 별로 Real Link가 상이하게 분포하므로 최단 경로가 실제 최단경로와 다르게 나타난다. 〈그림 5〉가 하나의 예를 보이고 있다.

〈그림 5〉의 왼쪽은 원형 교통망이고 오른쪽은 Real Link와 Dummy Link로 재구성된 교통망(Rebuilding network)이며 굵은 화살표가 최단경로를 나타낸다. 재구성된 교통망에서 점선으로 표시된 링크들이 매우 큰 비용을 갖는 Dummy Link 들이다. 이 경우 원형 교통망의 최단경로를 구성하는 링크 중에 dummy Link가 포함되어 있으므로 원래의 최단경로와는 다른 경로를 찾게 되고 이중 비용을 최소화시키는 경로가 재구성된 교통망의 새로운 최단경로가 된다.



〈그림 5〉 Original Network and Rebuilding Network

재구성된 교통망에서 최단경로를 탐색하는 방법은 기존 분할 탐색법(Dijkstra algorithm)을 이용한다. 이렇게 탐색된 경로의 비용을 염색체의 목적함수 값(Fitness Value)으로 저장하고 이 값의 크기에 따라 염색체간의 우열을 판단해 서열(Ranking)을 결정한다.

3. 경로개선 알고리듬

각 염색체의 서열이 결정되면 우수한 염색체들은 보존하고 열등한 염색체들은 제거한 뒤 제거한 수만큼 우수한 염색체들간에 교배(Crossover)를 실시해 새로운 개체들을 발생시켜야 한다. 본 연구에서는 상위 50%만을 보존하도록 모형을 구성하여 매 세대마다 절반의 염색체가 교배에 의해 새로 탄생된다. 이러한 염색체간의 교배 식은 식(3)과 같다.

$$X_{n+1}[k][l] = \alpha * X_n[1][l] + (1-\alpha) X_n[2][l] \quad (3)$$

$X_n[1][l]$: n번째 세대에서 첫 번째로 우수한 l 번째(link 번호) 염색체의 값

k : 교체대상 집합의 염색체 중 가장 높은 서열(정수).

α : 0~1 사이의 임의 발생 수(Random Number)

〈표 2〉는 위 교배규칙에 따라 제1세대 상위 1, 2번째 염색체간의 교배과정을 거쳐 새로운 제2세대 염색체가 발생되는 과정을 보여주고 있다. 이때 계산의 편의를 위해 α 값은 0.5라고 가정하였다. 〈표 2〉에서 Path란 가상의 염색체 공간으로서 Real Link의 Incident Relationship를 표현하는 변수이다.

모든 교배과정을 거쳐 새로운 세대가 구성되면 각 세대들은 다시 분할 탐색법에 의해 경로를 탐색하고, 목적함수 값을 갱신한다. 이때 돌연변이(Mutation)라는 과정을 거쳐야 한다. 돌연변이란 각 세대집합에 대해 임의의 염색체를 임의의 값으로 치환하는 과정을 의미한다. 유전 알고리듬에 있어 돌연변이 과정은 모형에서 계산된 해(Solution)가 국지적 최적해에서 정지하는 것을 방지하고 더 많은 영역을 탐색하여 전역적 최적해를 찾을 수 있도록 하는 과정이다. 본 연구에서는 돌연변이율을 5%로 사용하였다. 따라서 각 세대에 결쳐 5%의 염색체들은 돌연변이에 의해 유전자가 완전히 교체된다.

그러나 교배 및 돌연변이를 이용한 경로 개선과정이 비록 수렴조건을 만족하더라도 항상 국지적 최적해(Local Optimum)에 빠질 우려가 있다. 따라서, 유전 알고리듬에서는 최소 반복횟수라는 조건을 주어 최소 시행 횟수 내(최소 세대수 내)에서는 수렴에 도달하더라도 시행을 반복해 최적해를 계속 탐색한다. 본 연구에서 쓰이는 교통망은 그 크기가 작으므로 최소 시행횟수는 50회로 설정하였다.

〈표2〉 Crossover for Generating New Genes

세대	Chromosome	LINK Number(l)											
		1	2	3	4	5	6	7	8	9	10	11	12
1세대	Chromosome1 $X_1[1][l]$	0.12	0.28	0.54	0.15	0.65	0.82	0.12	0.02	0.89	0.29	0.54	0.60
	Chromosome2 $X_1[2][l]$	0.33	0.25	0.79	0.98	0.25	0.11	0.54	0.68	0.42	0.88	0.12	0.25
	...	...											...
2세대	Chromosome1 $X_2[1][l]$	0.12	0.28	0.54	0.15	0.65	0.82	0.12	0.02	0.89	0.29	0.54	0.60
	Chromosome2 $X_2[2][l]$	0.33	0.25	0.79	0.98	0.25	0.11	0.54	0.68	0.42	0.88	0.12	0.25
	...	...											...
	Chromosome k $X_2[k][l]$	0.23	0.27	0.67	0.57	0.45	0.47	0.33	0.35	0.67	0.59	0.33	0.43
	...	...											...
	Path	0	0	1	1	0	0	0	0	1	1	0	0

4. 불필요한 링크삭제

위와 같은 과정을 통해 최적 경로를 구축할 경우 최종적으로 구성되는 최단경로의 링크집합 외에 불필요한 링크, 즉 잉여 링크(Excess Link)가 발생할 수 있다. 이러한 잉여 링크의 발생은 유전 알고리듬의 수렴시간에 영향을 미치거나 잘못된 경로비용을 도출할 수 있다. 잉여 링크의 발생을 방지하는 방법으로는 여러 가지 방법을 이용할 수 있다. 가장 손쉬운 방법은 전체 링크 집합 중 분할 탐색과정에서 경로구축에 이용되지 않은 링크에 해당하는 유전자의 값에 일정한 가중치를 곱해 그 크기를 낮추는 것이다. 가중치가 0.9 라고 가정하면 식(4)가 된다

$$X_n^{unused}[k][l] = 0.9 \cdot X_n^{unused}[k][l] \qquad (4)$$

또 다른 방법은 염색체의 정렬과정에서 단순히 경로비용에 따라 서열을 결정하는 것이 아니라 같은 경로비용을 가진 염색체들을 사용한 링크의 개수에 따라 재정렬하는 방법이 있다. 본 연구에서는 이 두 방법을 동시에 고려하여 잉여 링크를 효과적으로 제거하였다.

5. 전역탐색 최단경로 알고리듬

앞에서 설명한 알고리듬을 정리하면 아래와 같다.

[STEP 1] 초기화

① 각 염색체내의 유전자 $X_n[Chrom\ number][l]$ 에 대해 0~1 사이의 임의 값 발생
여기서 n : 세대수, $Chrom\ number$ 는 Population 수, l 은 링크번호를 나타냄.

[STEP 2] 염색체별 교통망 변환

① $X_n[Chrom\ number][l]$ 값을 이용해 Real Link와 Dummy Link 집합 구성
② $X_n[Chrom\ number][l]$ 의 값이 미리 지정한 값(0.5) 이하인 Dummy Link 집합의 비용을 매우 큰 값으로 바꿈

[STEP 3] 염색체별 최단경로 구축

① 각 염색체 별로 아래의 과정을 순서대로 실시

- 초기화
  모든 노드의 Node_Mincost에 무한대 값(9,999,999,999) 저장하고 기점의 Node_Mincost에 0 값 저장
- 모든 노드의 Node_Mincost 계산
  I의 Back_Node가 A 일 때,
  If, Node_Mincost(I)+Mode_Change_cost〔Link(A,I)〕〔Link(I,J)〕+Link_cost(I, J) 〈 Node_Mincost(J)
  Then, Mode_Mincost(J)=Node_Mincost(I) +Mode_Change_cost〔Link(A,I)〕〔Link(I,J)〕+Link_cost(I, J)
  Back_Node(J)=I
  즉, 각 노드별로 진행하면서 각 노드를 기점으로 갖는 링크 종점 노드의 Node_Mincost를 환승비용(Mode_Change_cost)을 고려하여 갱신하고 Back_Node를 저장
- 최단 경로 결정
  종점까지 위 과정을 반복한 뒤 종점까지의 최소경로비용을 계산하고 그 비용을 염색체별로 저장
  이때 경로에 이용되지 않는 링크의 유전자에 식(4)와 같이 가중치 적용

② 각 염색체를 비용 크기가 적은 순서대로 정렬

[STEP 4] 교배(crossover)

① 적합도가 높은(여기서는 경로비용이 낮은) 염색체들간의 교배를 실시, 새로운 염색체를 미리 지정한 수(여기서는 전체 염색체의 50%)만큼 식 (3)과 같이 발생
② 적합도가 낮은 염색체(전체 염색체의 50%)를 새로운 염색체로 교체

[STEP 5] 돌연변이(mutation)

① 각 염색체별로 0에서 1까지 난수값(Random Value)을 발생시켜 그 값이 미리 지정한 돌연변이율(0.05)보다 작거나 같으면 해당 염색체를 임의로 재 발생시킴

[STEP 6] 적합도 계산(Fitness Value Calculation)

① STEP 3.의 과정 반복
② 최소 적합도(Minfit)와 최대 적합도(Maxfit) 값 저장

Minfit=경로비용이 최소인 상위 첫 번째 염색체의 경로비용 ($X_n[1][l]$ 의 경로비용)

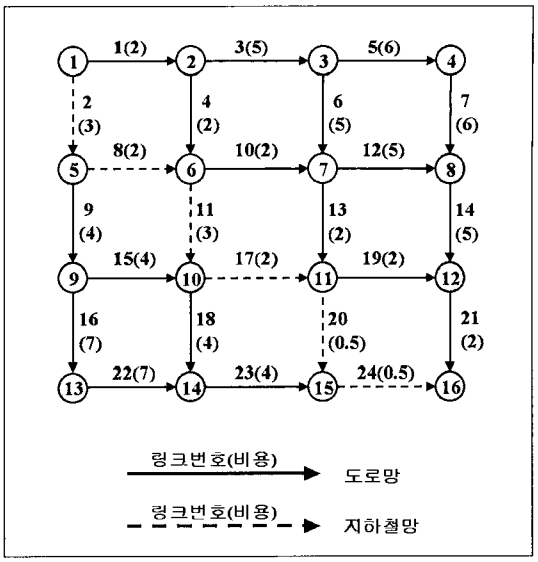
Maxfit=경로비용이 최대인 최하위 염색체의 경로비용($X_n[Chrom\ Max\ number][l]$ 의 경로비용)

여기서, *Chrom Max number* =마지막 유전자 (한 세대의 염색체가 총 40개라면 40)

[STEP 7] 수렴성 검증(Convergence Criteria)

① 각 염색체들을 비교해 모든 염색체의 Real Link들이 같고, Minfit = Maxfit 이면 Stop, 그렇지 않으면 STEP 4.로 돌아감

② 단, 최소 반복횟수(본 연구의 경우 50회)이내에서 수렴조건이 만족되더라도, 정지하지 않고 최소반복횟수까지 반복시행



〈그림 6〉 Example Network 1

Ⅳ. 모형의 평가

본 연구에서 개발한 유전 알고리듬을 이용한 전역탐색 알고리듬을 평가하기 위하여 탐색 영역 문제가 발생하는 2가지 경우에 대해 기존 알고리듬과 비교하였다.

1. 탐색 영역문제 (I)

본 연구에서는 수단간 환승비용이 존재하는 교통망에서 노드 확장없이 수형망 알고리듬과 덩굴망 알고리듬 및 본 연구에서 개발한 전역탐색 최단경로 알고리듬을 이용해 최단경로를 탐색하게 한 뒤 구축 경로 및 경로 비용을 비교하였다. 수형망 알고리듬의 경우 Dijkstra 알고리듬을, 덩굴망 알고리듬은 Moore가 제시한 알고리듬을 이용하였으며, 전역탐색 알고리듬의 경우 경로 비용 계산은 Dijkstra 알고리듬을 이용하였고 경로구축과정은 본 연구에서 개발한 유전 알고리듬을 이용하였다.

단, 환승비용을 고려하기 위해 비교 대상인 수형망과 덩굴망 알고리듬 모두 이전 링크 및 이전 링크의 수단을 저장하도록 설계하였다. 분석에 이용한 교통망은 〈그림 6〉에서 예시한 교통망을 사용하였다. 〈그림 6〉에서 실선은 도로망을 점선은 지하철망을 나타낸다. 이때 승용차에서 지하철로의 환승은 가능하나지하철에서 승용차로는 환승할 수 없다고 가정하였다. 승용차에서 지하철로 환승할 경우에는 4라는 비용을 지불하도록 하였다. 전역탐색 알고리즘에 의한 경로구축 과정은 〈표 3〉과 같다. 〈표 3〉를 보면 유전 알고리듬에 의해 반복과정을 거치면서 경로 비용이 감소하는 과정을 알 수 있다. Real Link와 Dummy Link의 구분 기준값은 0.5로 정하였으며, 이에 따른 Real Link는 표 내에 음영으로 표현하였다.

세대수 1의 경우 재구성된 교통망에서 실제 통행비용으로 이용 가능한 Real Link의 수는 11개이고 최소 경로의 비용은 30,012였다. 경로는 시행과정을 거듭하면서 비용이 개선되는데 경로간 교배가 1번 시행된 세대수 2의 경우 경로비용이 14로서 최소 비용은 아니지만 Real Link만으로 경로가 구성되었으며, 세대수 4의 경우 경로비용이 12, 시행을 거듭한 세대수 16의 경우 경로비용이 최소비용인 11에 도달하였다.

세대수 16에서 최소비용 11을 갖는 경로를 구축하였으나 경로 구축에 필요 없는 여분의 링크들이 Real Link 집합에 다수 포함되어있다. 이러한 여분의 링크들은 시행을 거듭하면서 차례로 도태되며, 세대수 50회가 지나면 여분의 링크는 모두 도태되고 경로 구축에 필요한 링크들만이 Real Link 집합에 남는다.

〈표 3〉 Example Network 1의 교통망 최단경로 탐색과정

	세대수 1	세대수 2	세대수 4	세대수 16	세대수 40	세대수 50
X[1][1]	0.001	0.775	0.574	0.369	0.830	0.146
X[1][2]	0.535	0.298	0.306	0.856	0.868	1.258
X[1][3]	0.203	0.785	0.547	0.087	0.475	0.172
X[1][4]	0.768	1.007	0.625	0.839	0.044	0.349
X[1][5]	0.556	0.829	0.363	0.325	0.148	0.235
X[1][6]	0.503	0.689	0.729	0.765	0.687	0.243
X[1][7]	0.333	0.285	0.274	0.741	0.905	0.326
X[1][8]	0.851	0.897	0.654	0.862	0.801	1.308
X[1][9]	0.782	0.121	0.464	0.186	0.402	0.180
X[1][10]	0.709	0.062	0.572	0.586	0.004	0.384
X[1][11]	0.165	0.824	0.427	0.867	0.649	1.185
X[1][12]	0.902	0.498	0.386	0.581	0.961	0.241
X[1][13]	0.675	0.579	0.089	0.669	0.998	0.162
X[1][14]	0.539	0.908	0.450	0.449	0.326	0.209
X[1][15]	0.289	0.069	0.224	0.359	0.458	0.385
X[1][16]	0.014	0.832	0.321	0.661	0.923	0.357
X[1][17]	0.087	0.686	0.541	0.554	0.803	1.313
X[1][18]	0.347	0.306	0.473	0.067	0.251	0.413
X[1][19]	0.140	0.099	0.542	0.267	0.411	0.404
X[1][20]	0.158	0.530	0.485	0.667	0.619	1.037
X[1][21]	1.038	0.216	0.709	0.798	0.819	0.466
X[1][22]	0.423	0.275	0.474	0.425	0.585	0.435
X[1][23]	0.113	0.873	0.398	0.723	0.167	0.378
X[1][24]	0.004	0.579	0.394	1.041	0.551	1.215
경로비용	30.012	14	12	11	11	11

〈표 4〉는 유전알고리듬을 이용한 전역 탐색 최단경로 알고리듬의 경로구축 결과를 수형망 알고리듬 및 덩굴망 알고리듬의 결과와 비교한 것이다. 수형망 알고리듬의 경우 전 노드의 비용을 저장하면서 전노드의 전 링크를 저장한다. 따라서 점선으로 표시된 지하철 수단의 경우 노드 5에서의 비용은 3, 노드 6은 4, 노드 10은 11, 노드 11은 8, 노드 15는 12.5, 노드 16은 13의 비용을 나타내게 되어 승용차만을 이용해 도착하는 비용 12에 비해 그 비용이 큰 것으로 계산된다. 이러한 결과는 전 노드와 전 링크만을 저장하기 때문인데, 예를 들어 노드 10의 경우 노드 4의 전 링크가 승용차 수단을 이용하는 링크 4이므로 노드 4의 비용 4에 환승비용 4, 링크비용 3을 더해 11이 노드의 최소 도착비용이 된다.

덩굴망 알고리듬의 경우 비용을 계산할 노드의 전 노드(Back Node)와 전전 노드(Back-Back Node)만을 고려할 수 있기 때문에 비용을 계산할 노드의 전 노드에서 발생하는 환승 비용만을 고려하게된다. 따라서 수단 2의 경우 노드 5에서의 비용은 3, 노드 6은 4, 노드 10은 8, 노드 11은 8, 노드 15는 10.5, 노드 16은 9의 비용을 나타내게 되어 승용차만을 이용해 도착하는 비용 12에 비해 그 비용이 작은 것으로 계산된다.

여기서 노드 16에 대한 비용의 경우, 노드 11의 비용(8)+링크 (11, 15)의 비용(0.5)+환승 비용(0)+링크 (15, 16)의 비용(0.5)=9가 되는데 실제 비용은 노드 11에서의 환승비용(4)가 포함되어야 하지만 탐색영역 문제 때문에 고려되지 않는다.

반면 전역탐색 알고리듬의 경로는 지하철을 이용한 경로를 최소비용경로로 구축했으며 경로 비용은 11로서 올바른 결과를 얻었다. 따라서 전역 탐색기법을 이용하는 최단경로 알고리듬은 노드의 확장 없이도 환승비용을 고려해 올바른 경로를 구축할 수 있음을 확인할 수 있다.

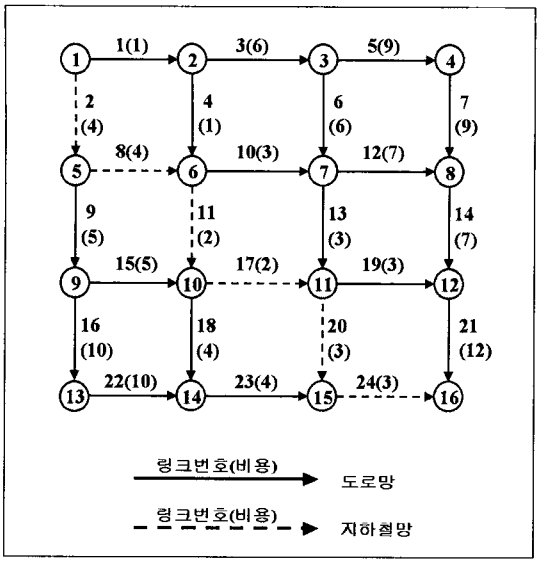
〈표 4〉 Example Network 1의 최단경로 알고리듬 별 경로 비교

최단경로 알고리듬	경로구축에 사용된 링크	경로 비용
수형망 알고리듬	1-4-10-13-19-22	12
덩굴망 알고리듬	1-4-10-13-20-24	9
전역탐색 알고리듬	2-8-11-17-20-24	11

2. 탐색 영역문제 (II)

두 번째 예제는 경로 탐색도중 2개 수단간에 환승이 발생하는 경우를 분석코자 하며, 분석 교통망은 〈그림 7〉과 같다. 여기서 환승비용은 앞 탐색영역 문제 I 과 같이 지하철에서 승용차로는 환승할 수 없다고 가정하고, 승용차에서 지하철로 환승할 경우에는 4라는 비용을 지불하도록 하였다. 본 예제의 최단경로는 링크 1-4-11-17-20-24를 연결시키는 경로로 노드 6에서 승용차로부터 지하철로 환승이 발생하며 최단경로 비용은 16이다.

수형망 알고리듬과 덩굴망 알고리듬의 탐색과정은 앞 예제1의 교통망 비용계산에서 설명한 바 있으므로 각 노드의 비용은 보이지 않았다. 각 알고리듬에 의해 탐색된 경로와 비용은 아래 〈표 5〉와 같다.



〈그림 7〉 Example Network 2

〈표 5〉에서 수형망 알고리듬의 경우 노드 11의 최소 도착 비용이 8이므로 노드 15의 비용은 8+4(환승비용)+3(링크 20의 비용)=15가 되고 노드 16의 비용은 18이 된다. 이에 비해 덩굴망 알고리듬의 경우 노드 15의 최소 도착 비용은 13, 노드 16의 비용은 노드 11의 비용 6+링크 20의 비용 3+환승비용 0+링크 24의 비용 3=12가 된다. 그러나 이 값은 노드 6의 도착비용이 2이므로(승용차) 노드 6에서 링크 11을 이용하려면 환승비용 4를 부담하여야 하지만 노드 6의 환승비용은 고려하지 못하게 되어 잘못된 경로비용을 산출하게 된다. 그러나 이 문제를 전역탐색알고리듬에 의해 계산할 경우 최단경로 비용은 16이며, 아래 〈표 5〉와 같이 올바른 최단경로를 선정하게 된다.

전역 탐색을 통해 경로가 구축되는 과정을 살펴보면 세대수 1의 경우 경로 비용이 10,025로서 Real Link 만으로는 경로를 구축할 수 없었다. 세대수 4의 경우 18의 비용을 가진 경로가 탐색되었으며, 세대수 11까지는 18의 경로를 구축하는데 필요 없는 링크를 줄여나가는 작업이 계속되었다. 세대수 12의 경우 돌연변이에 의해 16의 비용을 갖는 경로가 나타났다. 세대수 반복을 계속한 결과 세대수 50에서는 불필요한 링크들이 포함되지 않은 경로가 구축되었으며 경로비용은 16이었다.

〈표 5〉 Example Network 2의 최단경로 알고리듬 별 경로 비교

최단경로 알고리듬	경로구축에 사용된 링크	경로 비용
수형망 알고리듬	1-4-10-13-20-24	18
덩굴망 알고리듬	1-4-11-17-20-24	12
전역탐색 알고리듬	1-4-11-17-20-24	16

〈표 6〉 Example Network 2의 교통망 최단경로 탐색과정

	세대수 1	세대수 4	세대수 6	세대수 8	세대수 12	세대수 50
X[1][1]	0.480	0.863	0.949	1.044	0.686	4.961
X[1][2]	0.166	0.472	0.424	0.382	0.235	0.388
X[1][3]	0.736	0.548	0.493	0.444	0.444	0.235
X[1][4]	0.379	1.051	1.157	1.273	0.862	4.848
X[1][5]	0.370	0.065	0.058	0.052	0.199	0.293
X[1][6]	0.939	0.788	0.709	0.064	0.443	0.146
X[1][7]	0.602	0.588	0.529	0.476	0.399	0.306
X[1][8]	0.087	0.289	0.261	0.234	0.771	0.316
X[1][9]	0.485	0.094	0.084	0.076	0.542	0.067
X[1][10]	0.312	0.555	0.611	0.672	0.619	0.087
X[1][11]	0.595	0.204	0.184	0.165	0.754	4.437
X[1][12]	0.493	0.261	0.235	0.211	0.361	0.216
X[1][13]	0.566	1.011	1.113	1.224	0.646	0.375
X[1][14]	0.292	0.496	0.446	0.401	0.604	0.075
X[1][15]	0.122	0.596	0.536	0.483	0.801	0.321
X[1][16]	0.182	0.103	0.093	0.083	0.165	0.157
X[1][17]	0.344	0.443	0.399	0.359	0.762	4.085
X[1][18]	0.650	0.341	0.307	0.276	0.737	0.383
X[1][19]	0.418	0.447	0.402	0.362	0.479	0.240
X[1][20]	0.980	0.872	0.959	1.056	1.111	6.257
X[1][21]	0.608	0.458	0.413	0.372	0.409	0.108
X[1][22]	0.279	0.344	0.309	0.278	0.396	0.108
X[1][23]	0.137	0.619	0.557	0.502	0.364	0.137
X[1][24]	0.651	0.585	0.644	0.708	1.076	5.855
경로비용	10025	18	18	18	16	16

V. 분할탐색법과 전역탐색법(GA) 비교평가

본 절에서는 분할탐색법을 기초로 교통망을 확장하여 탐색영역 문제를 해결하는 경우와 본 연구에서 개발한 GA를 이용한 전역탐색법을 몇가지 측면에서 비교하였다. 두 탐색법의 비교는 탐색 알고리듬 적용전의 사전작업정도, 알고리듬의 계산 수행속도, 교통망 수정시 작업여부, 동적 탐색 가능여부 등으로 구분하

였다.

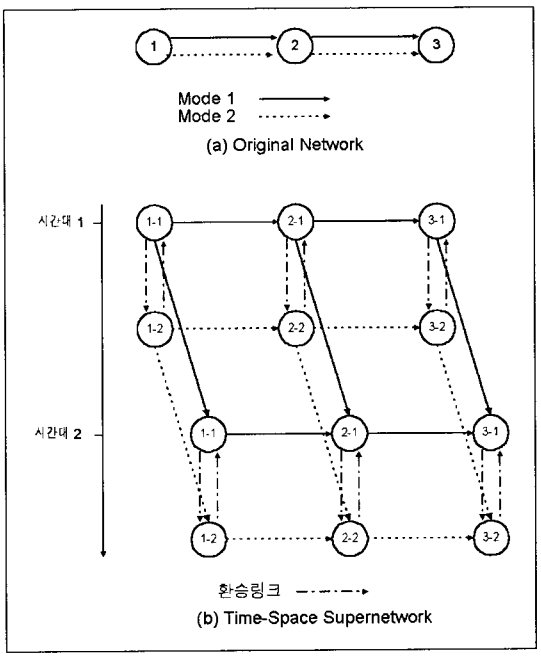
먼저, 교통망을 확장해 분할탐색법을 이용하는 경우 탐색 알고리듬 적용을 위한 사전작업시간이 매우 크다. 특히 지하철망이 포함되는 경우 다양한 환승 동선을 고려해 환승시간(환승벌점)을 계산하고 이를 링크와 노드로 표현해야 하기 때문에 환승 동선이 복수인 경우 사전 작업량이 매우 많게 된다. 즉, 아래 〈그림 8〉과 같이 3개의 노드와 4개의 링크(2개수단)로 구성된 단순한 네트워크인 경우에도 각 수단별, 시간대별 그리고 환승을 표현하기 위하여 노드를 $N\times M\times T$ 배로, 링크는 $(A\times M+3N)\times T$ 배로 확장해야 한다. 여기서 N 은 노드수, M 은 교통수단수, A 는 링크수 그리고 T 는 시간대수를 나타낸다. 그러나 전역탐색법을 이용할 경우 원형 교통망을 확장없이 그대로 이용하기 때문에 사전 작업이 필요없다는 잇점이 있다.

계산시간측면에서 가로망 확장을 통한 분할탐색법과 본 연구의 전역탐색법을 직접 비교하기가 쉽지 않는 데, 이는 탐색영역문제가 발생하는 경우와 같이 대규모 가로망의 경우, 네트워크 확장이 쉽지 않기 때문이다(아래 〈그림 8〉과 같이 단순한 경우라도 가로망을 확장하기가 쉽지 않기 때문에 대규모 가로망을 확장한다는 것은 상당히 어렵다). 그러나 일반적으로 전역탐색법이 교통망을 확장해 분할탐색법을 적용하는 경우보다 계산시간이 더 클 것으로 예상되는 데, 전역 탐색법의 경우 유전자 수만큼의 경로비용 계산과정이 필요하기 때문이다. 예를 들어 염색체 수가 10개이고 최소 반복횟수가 10회, Real Link 발생확률이 50%라면 단순 가로망에 적용할 경우, (염색체수×최소 반복횟수×Rebuilding Network의 크기(0.5)) 만큼의 계산시간이 추가되기 때문이다. 그러나 탐색영역 문제가 발생하는 대규모 교통망의 경우 통행수단의 수, 환승형태 등에 따라 가로망확장 탐색방법 역시 계산시간이 급증하기 때문에 GA를 이용한 탐색방법과의 계산시간 차이는 줄어들며, 특히 동적인 경우에는 두 탐색법이 큰 차이를 보이지 않을 것으로 예상된다.

교통망 수정시 추가작업량이 얼마나 더 요구되는가의 측면에서도 두 탐색법사이에 차이를 보이고 있다. 예를 들어 지하철 망이 추가되는 경우, 교통망 확장의 경우 신설된 지하철 망이 통과하는 노드들은 환승 링크와 환승노드를 모두 수정해야 한다. 이 경우 교통망이 크게 증가하게 되며 이론적으로 모든 수단간에 환승이 가능하다면, 환승노드는 환승가능한 수단 수만큼 증가하고, 링크의 수는 이용가능 수단 개수의 제곱에 비례하여 증가하게 된다. 그러나 전역탐색법을 이용할 경우 해당 링크만 추가하면 작업이 가능하기 때문에 교통망 수정 등에 쉽게 대응할 수 있는 장점이 있다.

위에서 비교한 내용들에 비해 두 탐색법간의 차이가 명확히 드러나는 것은 동적인 모형에 적용할 경우이다. 시간대가 2개 이상인 동적 통행배정 모형등에 교통망 확장을 이용한 분할탐색법을 적용할 경우 교통망이 크게 증가하기 때문이다. 시간대가 2개이고 수단이 2개인 교통망의 예가 〈그림 8〉에 나타나 있다.

〈그림 8〉의 (a)와 같은 단순한 교통망의 경우에도 시간대가 2개인 동적인 경우에는 (b)와 같이 복잡한 교통망으로 재구성되어야 한다. (a)의 경우 노드가 3개, 링크가 4개이지만 탐색영역문제를 해결하기 위해 교통망을 확장할 경우 그림(b)와 같이 노드가 12개, 링크가 26개로 커지며, 이러한 교통망 확장은 곧 계산비용의 증가를 의미하게 된다. 특히 이 경우 환승에 걸리는 최대시간은 한 단위의 분석시간 길이보다



〈그림 8〉 Original Network and Time-Space Supernetwork

〈표 6〉 분할탐색법과 전역탐색법의 비교

평가항목	교통망확장을 통한 분할탐색법	전역탐색법(GA)
사전작업정도 (교통망 확장정도)	탐색영역문제를 해결하기 위하여 노드와 링크 확장 확장된 노드수 = $N \times M \times T$ 확장된 링크수 = $(A \times M + 3N) \times T$ N : 노드수, M : 교통수단수 A : 링크수, T : 시간대수	확장 불필요
계산시간	교통망 내 수단이 단순한 경우에는 빠르나 수단(mode)의 수가 많아지면 계산시간 증가	염색체 수에 비례하여 계산량 증가. (염색체수×최소반복횟수×Rebuilding Network의 크기)배 만큼증가
교통망 수정시 추가작업	교통망 재구성필요	해당 링크만 추가
동적 탐색 가능여부	시간대 확장에 따라 교통망 재구성	(식2)와 같이 염색체 차원(Dimension)만 확장

매우 작다고 가정하였기 때문에 분석 시간대간의 환승링크가 존재하지 않지만, 환승에 걸리는 시간이 큰 경우에는 환승 링크도 확장되어야 하기 때문에 교통망의 크기는 기하급수적으로 증가한다. 이러한 경우 교통망 확장작업에 상당한 시간이 소요되며 따라서 효과적으로 최단경로를 찾기가 어렵게 된다. 그러나 유전 알고리즘을 이용할 경우, 식(2)와 같이 각 염색체의 차원(dimension)만 확장하면 되기 때문에 쉽게 적용할 수 있다. 그러므로 동적인 통합교통망의 경우 교통망확장 보다는 전역탐색법이 훨씬 효율적일 수 있다.

이상의 내용을 정리하면, 교통망 확장을 통한 분할탐색법은 교통망이 단순한 경우 계산속도가 빠르나 교통망내의 교통수단의 수가 늘어나고, 동적인 상황이 고려되면 교통망 확장에 따른 준비시간이 커지는 단점이 있다. 특히 시간대가 2개 이상이고, 수단이 2개 이상만 되어도 실제적용에 상당한 어려움이 있다. 이에 비해, 전역탐색법의 경우 교통수단과 분석 시간대가 증가하더라도 교통망 수정이 간단해 쉽게 최적경로를 탐색할 수 있는 장점이 있다. 아래 〈표 6〉은 각 탐색법에 대하여 앞에서 검토한 내용을 정리한 것이다.

Ⅵ. 결론 및 향후연구

본 연구는 전통적인 Node-Based 최단경로 알고리듬을 이용할 경우, 교통망 확장 없이는 해결할 수 없었던 탐색영역 문제(Searching Area Problem)를 전역 최적해 탐색 기법인 유전알고리듬을 이용하여 해결할 수 있음을 보였다. 탐색 영역문제는 주로 여러 수단이 존재하는 통합 교통망이나, 동적 최단경로 탐색시 나타나는 문제로서 기존에는 노드를 각 수단별로 확장함으로써 해결하여 왔다. 그러나 노드확장 기법은 교통망이 대규모이거나, 수단이 2개 이상, 또는 수단 내 노선이 복수인 경우 확장에 따른 비용이 크고 동적 최단경로 탐색의 경우 시간대별로 노드를 확장해야 하는 등 적용에 한계가 있었다.

이런 측면에서 본 연구에서 개발한 전역탐색 최단경로 알고리듬은 교통망의 확장 없이도 탐색영역 문제를 쉽게 해결할 수 있음을 보였다. 또, 각 염색체가 링크정보 뿐 아니라 수단정보나 노선정보까지 저장하는 경우에는 링크의 확장 없이도 수단 및 노선간의 환승비용까지 고려해 최단경로를 찾을 수 있는 장점이 있다. 유전 알고리듬이나 분지한계법을 이용한 최단경로 탐색의 경우 사전에 교통망을 확장하는등 준비작업 시간이 필요없는 대신에 계산비용이 전통적 탐색기법보다 큰 단점이 있다. 그러나 계산기의 성능향상등을 고려할 경우 그 차이는 실제로 더욱 줄어들 것으로 기대되며, 동적인 경우와 같이 복잡한 교통망에서도 정확한 경로를 찾을 수 있다는 점에서 활용가능성이 높을 것으로 예상된다.

교통 문제에 있어 최단경로 탐색과정이란 근본적으로 개개인의 경로 탐색과정을 묘사하는 것이다. 이러한 관점에서 볼 때 유전 알고리듬을 이용해 경로를 구축하는 것은 현실상의 경로구축과정을 합리적으로 묘사한다고 할 수 있는데, 이는 반복적인 학습과정에 의해 경로를 찾기 때문이다. 즉, 유전 알고리듬은 학

습을 통한 경로 구축과정을 수학적 알고리듬에 의해 재현한 기법으로서 기존의 분할 탐색법이 갖는 탐색 영역의 한계를 해결한 것이다. 다만, 위에서 제시한 경로비용 산정기법은 완전한 전역 탐색기법에 이용되기에는 불필요한 정보들을 많이 가지고 있다. 엄격한 의미에서 전역 탐색법에 이용되는 비용 산정기법은 링크간의 연결성만을 고려해 기점과 종점간의 정확한 최소비용만을 계산하면 되기 때문이다. 향후 연구로는 Loop 문제나 회전문제를 고려한 전역 탐색법과 계산시간을 단축시킬 수 있는 효율적인 알고리즘 개발 그리고 좀 더 규모가 큰 네트워크를 대상으로 각 알고리즘을 정량적으로 비교하는 부분들이 남아 있다.

참고문헌

1. 김익기(1998) "ATIS를 위한 수정형 덩굴망 최단경로 탐색 알고리듬의 개발", 대한교통학회지, 제16권 2호.
2. 이승환, 최기주, 김원길(1996) "도시부 ATIS 효율적 적용을 위한 탐색영역기법 및 양방향 링크탐색 알고리듬의 구현", 대한교통학회지, 제14권 3호, pp.45~59.
3. 임강원(1992) "도시교통계획-이론과 모형", 서울대학교 출판부, pp.146~151.
4. 임영도, 이상부(1998) "퍼지·신경망·유전진화", 도서출판 영과일.
5. 장인성, 김현철(1999) "대중교통망 최단경로 탐색을 위한 유전알고리즘", '99춘계학술발표대회 논문집, 한국보전공학회, pp.135~139.
6. 최기주, 장원재(1998) "복합 교통망에서의 최적경로산정 모형개발", 대한교통학회지, 제16권 4호.
7. Azevedo, J. A. and Martins E. Q. V.(1991) "An Algorithm for the Multiobjective Shortest Path Problem on Acyclic Networks", Investigacao Operacional, Vol. 11, pp.52~69.
8. Barra, T. et al.(1993) "Multidimensional Path Search and Assignment", 21st PTRC Summmer Annual Conference.
9. Climaco, J. C. N. and Martins E. Q. V.(1980) "On the Determination of the Nondominated Path in a Multiobjective Network Problem", V Symposium uber Operation Research.
10. Dijkstra, E. W.(1959) "A note on two problems in connection with graphs", Numer.Math.1, 269-271.
11. Ennio Cascetta, 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.
12. Gen M. and Cheng R.(1997) "Genetic Algorithms and Engineering Design", John Wiley & Sons.
13. Modesti P. and Sciomachen A.(1998) "A Utility Measure for Finding Multiobjective Shortest Paths in Urban Multimidal Transportation Networks", European Journal of Operation Reasearch.
14. Wong, S. C. and Tong, C. O.(1998) "Estimation of Time-Dependent Origin-Destination Matrices for Transit Networks", Transportation Research B 32, 35-48.
15. Sheffi. Y.(1985) "Urban Transportation Networks", Prentice-hall, Inc.
16. Thomas, R.(1991) "Traffic Assignment Techniques", Avebury Technical.
17. Tong, C. O. and Richardson, A. J.,(1984) "A Computer Model for Finding the Time-Dependent Minimum Path in Transit System with Fixed Schedules", Journal of Advanced Transportation 18, 145-161.
18. Zhou G. and Gen M.(1999) "Genetic Algorithm Approach on Multi-Criteria Minimum Spanning Tree Problem", European Journal of Operation Reasearch.