

■ 論 文 ■

복잡한 대규모의 도로망에서 실시간 경로 탐색을 위한 단계별 세분화 방법

A Coarse Grid Method
for the Real-Time Route Search in a Large Network

김 성 인

(고려대학교 산업시스템정보공학과 교수)

김 현 기

(교통개발연구원 책임연구원)

목 차

- I. 연구의 배경 및 목적
- II. 단계별 세분화 방법

- III. 운영 시스템 및 적용 사례
- IV. 결론
- 참고문헌

Key Words : 경로 안내 시스템(Route Guidance System), 단계별 세분화 방법(Coarse Grid Method), 대규모 도로망(Large-scale Networks), 탐색 시간(Search Time), 메모리 요구량(Memory Requirement)

요 약

복잡한 대규모의 도로망에서 방대한 정보를 분석하여 실시간으로 최적 경로를 탐색해야 하는 경로 안내 시스템에서는 탐색 효율이 필수적이다. 이를 위하여 많은 연구들이 탐색 대상이 되는 노드와 링크의 수를 줄이려고 노력해 왔다. 이 논문에서는 일부 영역만의 탐색으로 함수의 최적값을 찾는 단계별 세분화 방법(Coarse Grid Method)의 원리를 도로망에 응용한다. 처음에는 간선 도로망, 다음에는 주요 도로망, 그 다음에는 세부 도로망 등으로 그 대상을 단계적으로 세분화함으로써 동시에 수많은 노드들간의 경로를 찾는 기존 방법에서의 탐색시간을 단축한다. 이 시스템을 우리나라 전국 규모의 충분히 세분화된 실제 도로망에 적용하여 시스템의 효율성, 실용성과 실시간 운영 가능성을 경로의 탐색 시간, 경로의 적합성 등에서 입증한다.

1. 연구의 배경 및 목적

주어진 도로망에서 경로를 탐색하는 많은 연구들을 찾아볼 수 있다. 하나의 출발지에서 하나의 도착지까지(One to One) 또는 모든 도착지까지의(One to All) 최단 경로를 구하는 해법으로는 Moore 알고리즘과 Dijkstra 알고리즘이 대표적이다(Nemhauser, 1988). 또한 모든 출발지에서 모든 도착지까지의(All to All) 최단 경로를 구하는 해법으로는 Floyd-Warshall 알고리즘이 대표적이다(Ahuja, et al., 1993). 최적 알고리즘들로 밝혀져 있는(Backhouse, et al., 1994) 이들 알고리즘들은 n 개의 노드를 갖는 도로망에서 각각 $O(n^2)$ 과 $O(n^3)$ 의 연산 횟수를 필요로 한다(Han, et al., 1997). Zahn & Noon(1998)은 미국 중·서부 및 남부 10개주의 실제 도로망에서 15개의 최단 경로 알고리즘을 적용하여 상대적인 효율성을 실험적으로 비교, 분석한 결과 Dijkstra 계통의 알고리즘들이 우수함을 입증하였다.

복잡한 대규모의 도로망에서 모든 노드들간의 최단 경로를 탐색하는 것은 오늘날의 컴퓨터를 사용하는 것이 당연하지만 매우 긴 계산 시간과 많은 양의 기억 용량을 필요로 한다. 예를 들면, Dantzig와 Floyd의 최단경로 알고리즘은 $n(n-1)(n-2)$ 번의 연산(덧셈과 비교)을 필요로 한다(Ahuja, et al., 1993). 행렬 방법(Matrix Method)의 가장 효율적인 층계형 방법(Cascade Method)은 $2n^2$ 의 변수들을 동시에 저장해야 한다. 50만 명이 거주하는 도시라면, 이 도시의 도로망은 500개 정도의 노드를 갖고 있을 것이고, 따라서 1억 2,500만번의 연산과 50만개의 기억 용량이 필요하다. 이보다 더 크고 복잡한 도로망이 많을 것이므로 실제의 도로망에서 기존의 여러 알고리즘을 적용하는데 있어서 한계와 어려움을 쉽게 상상할 수 있다(Mills, 1966).

이들의 연구 이래 수많은 알고리즘들이 개발되었는데(예를 들면, Lysgaard, 1995; Romeijn & Smith, 1999; Sanders, 2001 등), 이들 알고리즘들의 대부분은 위 알고리즘의 절차에서 노드 확장 과정을 개선하거나 이미 고려한 노드의 경로비용 갱신 과정을 개선함으로써 연산 횟수 및 저장 용량을 효과적으로 감소시키고자 한 것이다. 특히 Gallo & Pallotino(1988)는 그 동안 제안된 최적 경로 탐색을 위한 알고리즘의 대부분이 적용 데이터의 구조만을 바꾼 것임을 밝혔다.

이와 같이 전통적인 최단 경로 탐색 알고리즘들은 도로망의 규모가 크고, 복잡할수록 최적해의 탐색 시간이 길어지는 단점이 있다. 이는 주어진 도로망의 모든 링크를 한번에 탐색하는 비효율적인 탐색 구조에 기인한다. 특히 도시 도로망에 흔히 존재하는 회전 제약을 고려하기 위해 주로 사용되는 덩굴망 알고리즘의 연산량이 $O(n^3)$ 임을 감안할 때, 이와 같은 비효율적인 탐색 구조는 도로망의 규모가 커질수록 탐색 시간이 많이 소요되는 문제가 있음을 쉽게 짐작할 수 있다(Namkoong, et al., 1998).

Noto & Sato(2000)는 탐색 영역이 넓은 경우에 Dijkstra 알고리즘은 최단 경로를 찾기 위해 너무나 오랜 시간을 소요하므로 실시간 탐색 문제에는 적합하지 않다고 지적하였다. 이들은 자동차 항법 시스템(Car Navigation System)의 경로 탐색과 같이 실시간에 해를 얻기 위하여, 전통적인 Dijkstra 알고리즘을 확장한 새로운 알고리즘을 제안하였다. 2,000 x 2,000의 격자 망과 1~5의 링크 비용을 생성하여 수행된 모의 실험 결과, 이 방법은 기존의 방법보다 탐색 노드의 수를 50%, 탐색 시간을 20% 감소시킨다고 한다.

이러한 면에서 탐색 대상이 되는 노드나 링크의 수를 줄임으로써 탐색 시간을 줄이려는 연구가 이루어져 왔다. 주어진 도로망에서 모든 노드들간의 경로를 탐색하지 않고, 이 도로망을 분할하여, 분할된 도로망내의 노드들만을 고려하여도 주어진 도로망 전체의 최적 경로를 찾을 수 있는 해법이 Hu(1968)와 Hu & Torres(1969)의 분할 알고리즘(DA: Decomposition Algorithm)이다. 이들의 DA는 여러 개로 분할된 하나의 작은 도로망에서 구한 지역(Local) 최적해가 전체 도로망에서의 전역(Global) 최적해가 되게 한다. Blewett & Hu(1977)는 일반적인 그래프로 표시되는 모든 도로망에 DA가 적용됨을 보였다. 또한 도로망의 구조에 따라 출발지와 도착지의 양쪽 방향에서 동시에 탐색을 시도하는 양방향 탐색(BS: Bi-directional Search)도 대상 노드의 수를 줄이려는 의도이다.

이 논문에서의 방법도 모든 노드를 동시에 탐색 대상으로 하지 않고, 단계적으로 도로망을 세분화하면서 점점 좁은 지역의 노드들을 탐색 대상으로 하여 대상 노드의 수를 줄이려는 의도이다. 이를 단계별 세분화 방법(CGM: Coarse Grid Method)이라 부르기로 한다. 본래 이 방법은 일부 영역만의 탐색으로 함수의 최적값을 찾는 원리이다(Nemhauser, 1966).

제II장에서는 도로망에서 경로 탐색을 위한 CGM을 개발하고, 탐색 효율을 분석한다. 제III장에서는 우리나라 전국 규모의 실제 도로망에 적용하여 실용성을 입증한다.

II. 단계별 세분화 방법 (CGM: Coarse Grid Method)

이 절에서는 노드들의 수가 상당히 많아서 효율적으로 다루기 어려운 대규모의 복잡한 도로망에서, 그 대상을 단계적으로 축소하여 분할하는 경로탐색에서의 CGM을 개발한다. 각 단계에서는 분할된 여러 개의 도로망 중에서 하나의 도로망을 고려한다. 여러 단계를 거치지만, 매 단계에서는 적은 수의 노드만을 고려하여 전체적으로 연산과 기억 용량을 현저하게 감소시킨다.

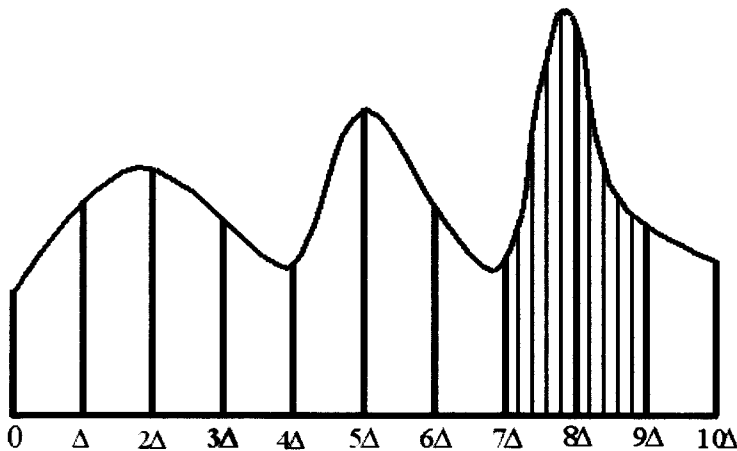
CGM은 본래 수리계획법에서 함수의 최적 값을 찾는 데 연산(덧셈 및 비교)의 횟수를 줄이려는 방법이다 (Nemhauser, 1966). <그림 1>에 CGM의 절차가 예시되어 있다. 영역(Domain)이 정수(整數)인 함수

$$Q(d), \quad d = 0, 1, 2, \dots, 200,000$$

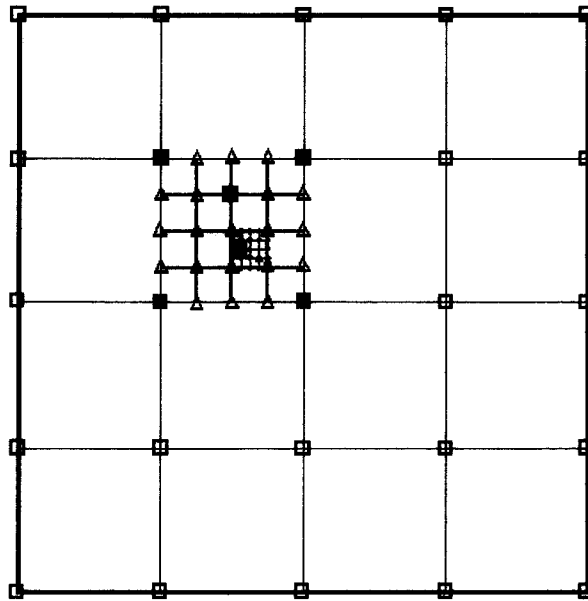
의 최대 값을 찾는다고 가정하자. 만약 격자 크기 Δ 를 1로 한다면, $d = 0, 1, 2, \dots, 200,000$ 에서 $Q(d)$ 를 평가하여야 되고, 따라서 함수는 총 200,001개의 점들에서 평가된다. 그러나 처음에 격자 크기를 $\Delta_1 = 10,000$ 으로 하여 $d = 0, 10,000, 20,000, \dots, 200,000$ 의 21개의 점에서 함수를 평가한다. 만약 d

$= d_1^* = 80,000$ 에서 최대값을 찾았다고 가정하면 $Q(d)$ 의 최대값이 $d_1^* - \Delta_1 = 70,000 \leq d \leq d_1^* + \Delta_1 = 90,000$ 에 있다고 본다. 이제 격자 크기를 $\Delta_2 = 1,000$ 으로 줄여서 $d = 70,000, 71,000, 72,000, \dots, 90,000$ 의 21개의 점에서 $Q(d)$ 를 평가한다. 다시 $d = d_2^* = 78,000$ 에서 $Q(d)$ 의 최대 값이 찾아졌다고 하면, 최대 값은 $d_2^* - \Delta_2 = 77,000 \leq d \leq d_2^* + \Delta_2 = 79,000$ 에 있다고 본다. 이제는 격자 크기를 100으로 줄인다. 이와 같은 절차를 격자의 크기가 1이 될 때까지 반복한다.

이 방법은 평가(연산) 횟수를 줄이지만, 함수의 형태에 따라 전역 최적해가 아닌 지역 최적해에 빠질 수 있는 위험이 있다. 이와 같이 단계적으로 격자의 크기를 줄여가면서 함수의 최적 값을 찾는 CGM을 본 연구의 최적 경로 탐색에 응용한다. 첫 번째 단계에서는 출발지 O 가 주어진 좁은 지역의 충분히 세분화되어 분할된 도로망내의 노드만을 고려한다. 두 번째 단계에서는 이보다 넓은 지역의 보다 거친 도로망내의 노드들을 고려한다. 세 번째 단계에서는 다시 보다 넓은 지역과 보다 거친 도로망내의 노드들을 고려한다. 이러한 단계를 전체 규모의 지역에 도달할 때까지 계속한다. 분할된 각 도로망내의 모든 노드들은 연결되어(Connected) 있다. 어떤 도로망내의 일부 노드들은 다음 단계 또는 전 단계 도로망내의 노드와 중복된다. 이들 노드들을 각각 다음 단계 도로망으로의 입구 노드, 전 단계 도로망으로의 출구 노드라고 부른다. 어떤 단계의 도로망에서 다음 단계 또는 전 단계 도로망으로의 연결은 이들 입출 노드들을 통하게 된다. 실제적인 예를 들어, 세 단계의 세분화 방법이 적용된다



<그림 1> 함수에서의 단계별 세분화 방법



〈그림 2〉 도로망에서의 단계별 세분화 방법

면, 첫 번째 단계에서는 구 또는 군 규모의 세부 도로망을 고려하고, 두 번째 단계에서는 시 또는 도 규모의 주요 도로망을 고려하며, 세 번째 단계에서는 전국 규모의 간선 도로망을 고려하는 것이다. 도시의 격자 도로망에서 3단계를 거치고 각 단계에서 $N=16$ 개의 도로망으로 분할하는 경우가 〈그림 2〉에 나타나 있다.

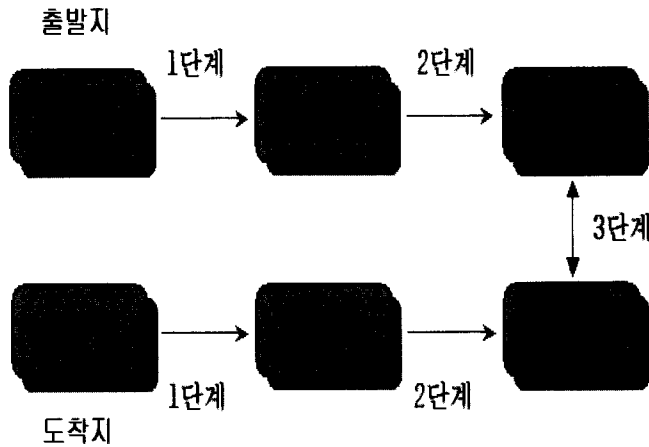
이러한 다단계 도로망에 대한 연구는 Shapiro, et al.(1992)에 의하여 "Convenient Route"이라는 아이디어에서 시작되었다. 이들은 전체 도로망에서 노드와 링크에 단계를 매겨, 단계별로 전체 도로망을 고려한다. 단계에 따라 대상 도로망의 규모를 축소하거나 분할하지는 않는다. 이들의 경우를 2단계(지방도로와 고속도로)로서, 지방도로상의 출발지 O 에서 역시 지방도로상의 도착지 D 까지의 경로를 찾는 예로 설명하면, 그 경로로서 우선 지방도로→고속도로→지방도로를 생각할 것이다. 지방도로와 고속도로를 여러번 오가는 경로는 그 거리가 짧다고 하여도 운전의 편의상 바람직하지 않을 것이다. 이와 같은 의미에서 "Convenient Route"이다. 운전자는 O 에서 가까운 고속도로 진입구를 찾아 고속도로로 진입하여 그 고속도로를 주행하고, D 근처의 가까운 출구로 나와서 D 에 도착한다. 이때 고려하는 도로망은 처음 단계에서 주로 O 와 D 근처의 좁은 지역의 지방도로망과 다음 단계에서 넓은 지역의 고속도로망이 된다. 이와 같이 주로 2단계의 도로망에

서 알고리즘의 개발 및 효율 분석, 도로망의 분할, 실제 도로망에의 적용 등이 연구되었다(Chou, et al., 1998; Liu, 1997; Park, et al., 2001). 이 문제에 대한 보다 정형적인 모델화, 알고리즘 개발 및 그 효율성 분석 등은 이들의 연구를 참조하기 바란다.

CGM에서 분할된 도로망 내의 모든 노드들은 완전히 연결되어야 한다. 이를 만족시키지 못할 때에는 전문가가 인위적으로 조정한다. 이때 분할은 상위 도로망의 형태에 따라서 자연적으로 정해질 수도 있다(Liu, 1997). 예를 들면 고속도로로 둘러싸이는 지역의 하나의 도로망으로 분할된다. CGM에서는 같은 단계에서 분할되는 도로망들은 비슷한 수의 노드와 링크를 갖게 한다. 이렇게 함으로써 $O-D$ 에 따른 경로 탐색 시간을 일정하게 할 수 있다. 이러한 분할 과정 및 작업은 많은 시간을 소요하겠지만 일단 완성되면 상당한 기간동안 변동없이 사용될 수 있다. 또한 전문가의 경험과 지식을 반영하여 자동적으로 수행하는 프로그램이 개발될 수 있다.

3단계의 소규모, 중규모 및 대규모 도로망을 고려하는 예에서 출발 노드 O (Origin)에서 도착 노드 D (Destination)까지의 경로를 탐색하는 절차 〈그림 3〉을 설명한다.

[절차 1] 소규모 도로망 내에서 출발 노드 O 의 주위에 있는 적당한 개수(예를 들어 K_1)의 중규모 노드



(그림 3) 단계별 세분화 방법의 경로 탐색 절차

를 정하고, O 에서 이들까지의 경로를 찾는다. 여기에서 중규모 노드는 O 와 소규모 도로망으로 연결되는 중규모 도로망 내의 노드 중에서 선정된다. O 까지의 최종 경로는 선정된 중규모 노드들 중의 하나를 거쳐 연결된다. 이 중규모 노드들은 도로망의 연결구조에 따라 4~8개가 결정된다.

[절차 2] 중규모 도로망 내에서 앞에서 정한 각각의 중규모 노드 주위에 있는 적당한 개수(예를 들어 K_2)의 대규모 노드를 정하고, 이들까지의 경로를 찾는다. 여기에서 대규모 노드는 절차 1에서와 같은 방법으로 정해진다. 이제 총 $K_1 \cdot K_2$ 개의 경로가 찾아진다.

[절차 3] 도착 노드 D 에서도 절차 1, 절차 2를 밟아 총 $K_3 \cdot K_4$ 개의 경로가 찾는다.

[절차 4] 양쪽으로부터의 모든 경로의 조합인 $K_1 K_2 \cdot K_3 K_4$ 개의 경로 중에서 최적 경로를 찾는다. 결국 알고리즘은 O 와 D 의 양쪽에서 상위 또는 하위 도로망으로 연결되는 입구 노드와 출구 노드를 찾는 것이다.

예를 들어 고속도로로의 입구 노드를 찾는데 Shapiro, et al.(1992)은 가장 가까운 노드를, Chou, et al.(1998)은 모든 노드를, Park, et al.(2001)은 거리의 단축 효과에 따라 노드의 수를 제한하고 있다. CGM에서는 전문가들이 모여 그들의 지식과 경험에 따라 노드마다 주위의 입구 노드들을 선정한다. 이들의 방법은 프로그램화될 수 있다. 이러한 알고리즘들은 모두 그 특성상 고려 방법 및 고려 대상의 제한으로 인하여 최단(최적) 경로를 찾는 것이 목적이 아님은 물론이다. 그러나 지역마다 다른 도로망의 형태 및 연결구조로 인하여 적절한 노드를 결정하는 로직(Logic)이나 알고리즘

(Algorithm)이 존재하지 않는다. 소위 구조가 명확치 않은 체제(ISP: Ill-Structured Paradigm)이다. ISP란 명확한 객관적인 해법이 존재하지 못하고 다양하고 주관적이고 모호하고 불확실하며 때로는 상충적인 해답이 존재할 수밖에 없는 분야를 일컫는다. 이러한 체제 아래에서는 인공지능을 이용하는 전문가 시스템의 방법론이 적절할 수 있다(Giarratano & Riley, 1998).

따라서 전문가가 지역마다 다른 도로망의 형태 및 연결구조를 보고, 경험과 판단에 의하여 노드를 결정하게 된다. 여기에서 전문가가 결정하는 과정의 한 가지 예를 알고리즘으로 구현하면 다음과 같다.

[세부절차 1] 주어진 노드 O 를 중심으로 하여 북쪽 방향을 0도로 하고 매 45도마다 방사선을 긋는다. 즉, 지역을 8개로 나눈다.

[세부절차 2] 이웃하는 방사선 사이의 지역마다 가장 가까운 상위 규모(도로망)의 노드를 선정한다. 이에 따라 최대 8개의 상위 노드가 결정되지만, 어떤 지역에서는 상위 규모의 노드가 존재하지 않을 수도 있다.

[세부절차 3] 이 노드들 중에서 노드 O 와의 거리가 가장 가까운 노드를 찾고 그 거리를 읽는다.

[세부절차 4] 노드 O 와의 거리가 위 절차에서 구한 거리의 3배가 넘는 노드를 제외한다.

[세부절차 5] 이상의 절차를 노드를 바꾸어 가면서 수행하고 결과를 데이터베이스에 저장한다. 지역 구분, 노드까지의 거리 계산 등의 절차는 수치 지도로부터 쉽게 수행된다.

만약 세 단계를 거치고, 각 단계에서 주어진 도로망이 N 개로 세분된다면, 1개의 전체(대규모) 도로망, N

개의 중규모 도로망, N^2 개의 소규모 도로망으로 모두 $(1+N+N^2)$ 개의 도로망을 고려하게 된다. 이 각각의 모든 도로망에 M 개의 노드가 있다고 하자. 우리는 모든 노드들간의 경로를 찾는다. $(1+N+N^2)$ 개의 도로망을 고려하고 하나의 도로망에서 $M(M-1)$ 개의 경로를 탐색하므로, 총 $(1+N+N^2) \cdot M(M-1)$ 개의 경로를 탐색하게 된다. 단계별로 세분화하지 않고 한번에 구하려면 총 MN^2 개의 노드를 고려하므로 총 $MN^2 \cdot (MN^2-1)$ 개의 경로를 탐색해야 한다.

수치적인 예로서 3단계, $N=10$, $M=200$ 이라면 탐색하는 경로 수는 CGM을 적용하는 경우에는 총 4,417,800개, 단계별로 세분화하지 않고 한번에 구하는 경우에는 399,980,000개의 경로를 탐색해야 한다. CGM의 탐색 효율은 1/90.5에 달한다.

각 단계에서 적용하는 알고리즘으로 최적 알고리즘이 아닌 경험적(Heuristic) 알고리즘을 사용하여 탐색 시간을 단축시킬 필요는 없다. CGM 자체에 의하여 계산량이 현저하게 감소하기 때문이다. 이들 알고리즘들을 온라인(On-line)으로 구성하는 데에는 여러 가지 방법이 있다. Chou, et al.(1998)은 하위(분할된) 도로망에서의 모든 경로는 오프라인(Off-line)에서 미리 구하고 상위 도로망에서의 경로만을 온라인에서 구하는 방법을 제안하였다. CGM에서는 이들 모두를 오프라인에서 구하여 저장하는 방법을 제안한다. 온라인에서는 알고리즘을 가동하지 않고 비교만 할 뿐이다.

III. 운영 시스템 및 적용 사례

실시간 RGS에서 경로 탐색에 소요되는 시간을 절감하기 위하여 많은 방법이 제안되고 있다. 특히 오늘날의 차량 멀티미디어 서비스인 텔레매틱스는 교통경로 안내뿐만 아니라, 교통정보, 생활정보, 엔터테인먼트 등 다양하고 방대한 정보 서비스를 제공해야 하기 때문에, 그 일부인 교통경로 탐색은 매우 빠른 시간내에 이루어질 필요가 있다. Sen & Thakuriah(1995)는 중앙관제센터의 부담을 덜어주기 위하여 각각의 차량에 통행시간에 대한 추정치를 담은 CD-ROM을 장착하고 개별적으로 처리하는 방법을 제안하였다. 또한 Wahle, et al.(2001)은 실시간에 방대한 자료를 가공하는 것은 많은 시간이 소요되므로 오프라인으로 가장 중요한 변수들(실제 거리, 과거 통행 시간의 평균 및 분산, 운전자의 선호도, 회전 제약 등)을 미리 선정하고, 이 변

수들을 바탕으로 모의 실험하여, 최적 경로를 운전자에게 제공하는 방법을 제안하였다. 김태진과 한민홍(2001)은 매우 많은 클라이언트로부터의 경로탐색 요청을 서버에서 효율적으로 처리하기 위한 다중경로탐색 알고리즘을 개발하였다.

본 논문의 시스템에서는 정상적인 교통 상황의 경우 모든 $O-D$ 에 대한 각종 도로망에서의 경로는 오프라인에서 결정되어 DB에 구축된다. 구체적인 $O-D$ 가 주어지면 CGM에 의하여 최소한의 비교로써 경로를 조연한다. 이로 인한 경로 탐색시간의 단축은 RGS에서 다른 정보들을 실시간으로 탐색, 분석, 제공하는 데에 할당할 수 있다. 경로는 수리적 알고리즘, 인공지능 알고리즘 및 전문가의 지식과 경험에 의하여 결정된다. 각 경로는 전문가에 의하여 적합도에 대한 등급이 매겨져 있다. 교통사고 등의 돌발 상황의 경우는 별도로 고려한다. 이들은 전문가 시스템으로 구축되며, 이 시스템은 학습기능을 가져, 시스템이 활용되면서, 시스템이 제시하는 경로는 적절함을 더하게 된다. 이 시스템의 설계 및 세부 사항에 대해서는 Kim & Kim(2002)의 논문에 설명되어 있다. 실제 도로망에 적용하여 알고리즘의 효율성을 분석하는 연구들은 모두 2단계를 적용하고 있다. Park, et al.(2001)은 4,075개의 노드와 9,212개의 링크를 갖는 우리나라 지방 도시의 도로망을 1단계에서 734개의 노드와 1,570개의 링크를 갖는 도로망을 고려하였다. Liu(1997)는 12,697개의 노드와 30,687개의 링크를 갖는 Singapore 도로망을 1단계에서 1,096개의 노드와 2,574개의 링크를 고려하고 2단계에서 이를 503개의 도로망으로 분할하였다. Chou, et al.(1998)는 3,189개의 노드와 5,658개의 링크를 갖는 Southeast Michigan의 도로망을 1,543개의 도로망으로 분할하였다. 한편 Shapiro, et al.(1992)은 66,449(257x257) 개의 노드를 갖는 가상의 격자 도로망에서 4단계를 적용하는 모의실험을 하였다. 그 결과들은 모두 다단계 적용의 효율성을 입증하고 있다.

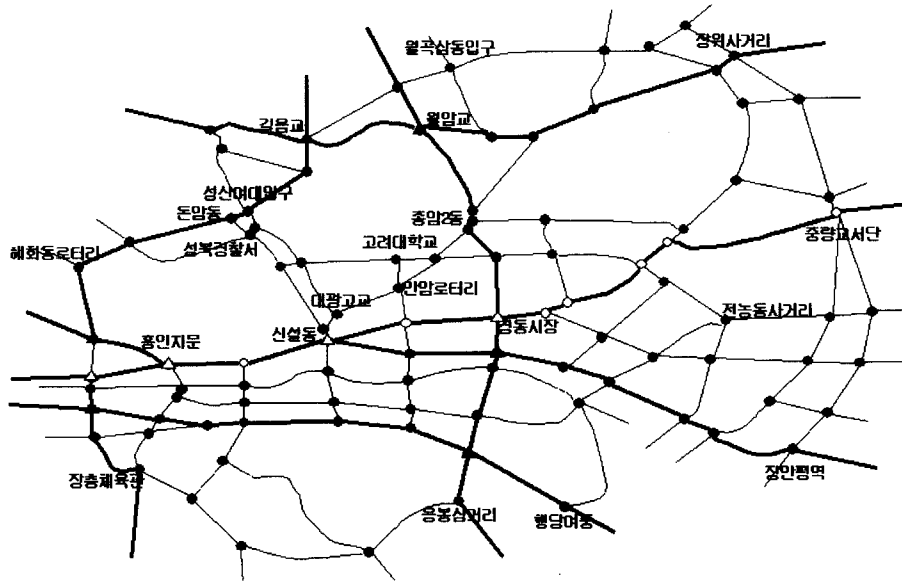
이제 우리나라 전국 규모의 도로망에서, 고려대학 교로부터 부산대학교까지의 최단 경로를 탐색하는 사례로 이 시스템이 복잡한 대규모의 도로망에서 효율적인 경로 조연 시스템이 될 수 있음을 보인다. 이 과정에서 시스템의 타당성을 검증하고, 결과의 우수성을 보인다.

세 단계의 CGM으로서, 첫 단계는 성북-동대문구

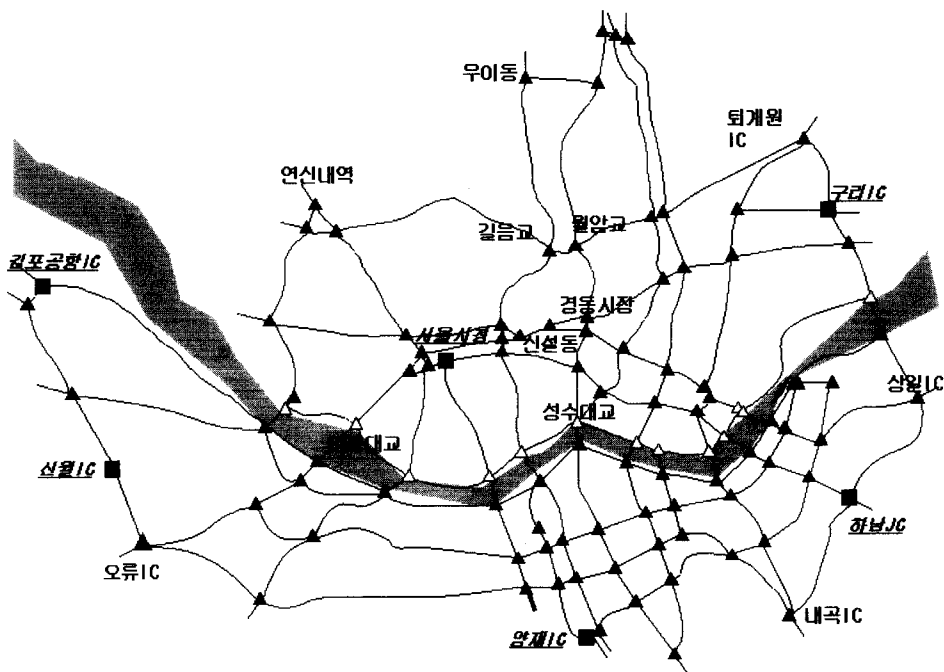
규모의 세부 도로망이 되고, 두 번째 단계는 서울특별시 규모의 주요 도로망이 되며, 세 번째 단계는 전국 규모의 간선 도로망(고속도로 및 국도)이 된다. 각 단계에서 고려하는 도로망은 각각 <그림 4>, <그림 5> 및 <그림 6>과 같이 구성되어 있으며, 각 도로망의 노드 수와 링크 수는 <표 1>과 같다.

<표 1> 적용 대상 도로망의 분류

도로망	전국	서울특별시	성북-동대문구
링크 수	254	354	320
노드 수	100	102	96
링크 수/ 노드 수	2.54	3.47	3.33



<그림 4> 성북-동대문구 도로망



<그림 5> 서울특별시 도로망

이와 같은 상황에서 CGM은 <그림 7>과 같이 수행된다. CGM의 절차에 의한 결과는 다음과 같다.

[절차 1] 우선적으로 성북-동대문구의 세부 도로망에서 출발지인 고려대학교에서 주위에 있는 서울특별시 주요 도로망의 노드들을 찾는다. 이 경우에

신설동역, 경동시장, 월암교, 길음교의 4개 노드가 찾아지고, 고려대학교로부터 이들까지의 최단 경로를 찾는다.

[절차 2] 다음으로 서울특별시 주요 도로망에서 앞서의 4개 노드들에서 각각 주위에 있는 전국 간선 도로망의 노드들을 찾는다. 이 경우에 모두들

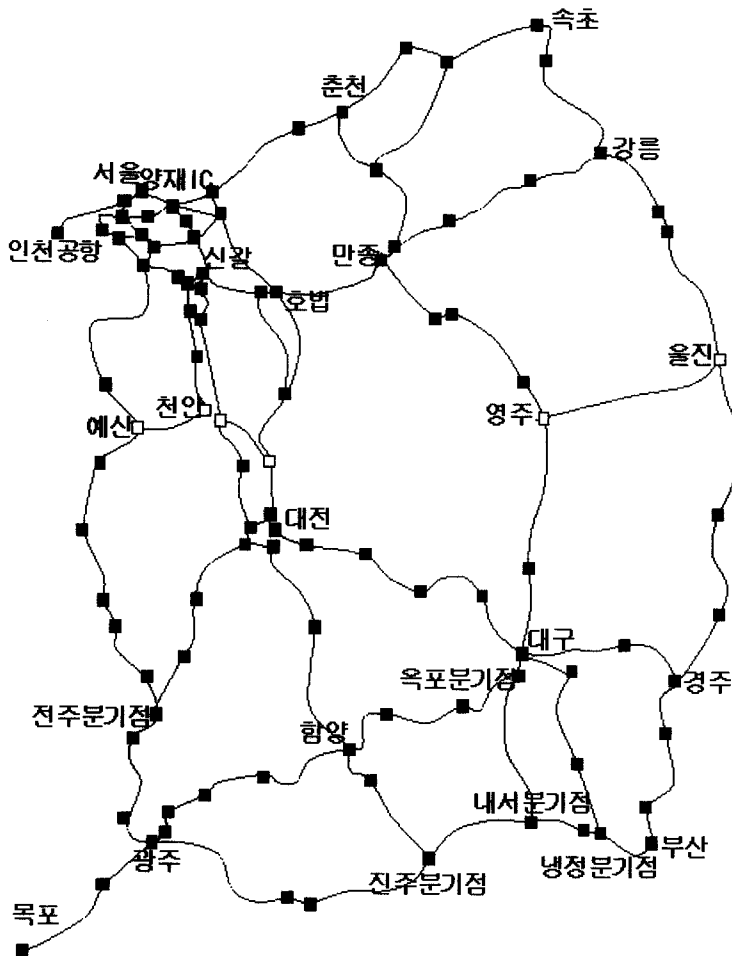
서울시청, 구리IC, 상일IC, 양재IC, 신월IC, 김포공항IC의 6개 노드들이 찾아지고, 따라서 이들과 위 노드들의

조합으로 총 24개의 최단 경로를 찾는다.

[절차 3] 이상의 절차를 부산 금정구내의 부산대학교가 위치한 지역의 세부 도로망, 부산직할시의 주요 도로망, 전국의 간선 도로망에 단계적으로 적용한다. 부산대학교로부터도 여러 개의 전국 규모의 노드들이 찾아지나 여기에서는 부산IC 하나만을 고려하기로 한다.

[절차 4] 마지막으로 위 24개의 경로와 부산IC를 가장 짧게 연결시키는 경로를 찾는다.

이상의 결과를 종합하면 <표 2>와 같다. 이 표를 보면 CGM에 의한 최단 경로는 소규모 도로망의 고려대학교에서 출발하여 소규모 도로망과 중규모 도로망에 동시에 존재하는 경동시장에 도착한다. 다음은 중규모 도로망의 경동시장에서 중규모 도로망과 대규모 도로망에 동시에 존재하는 양재IC에 도착한다. 마지막으로



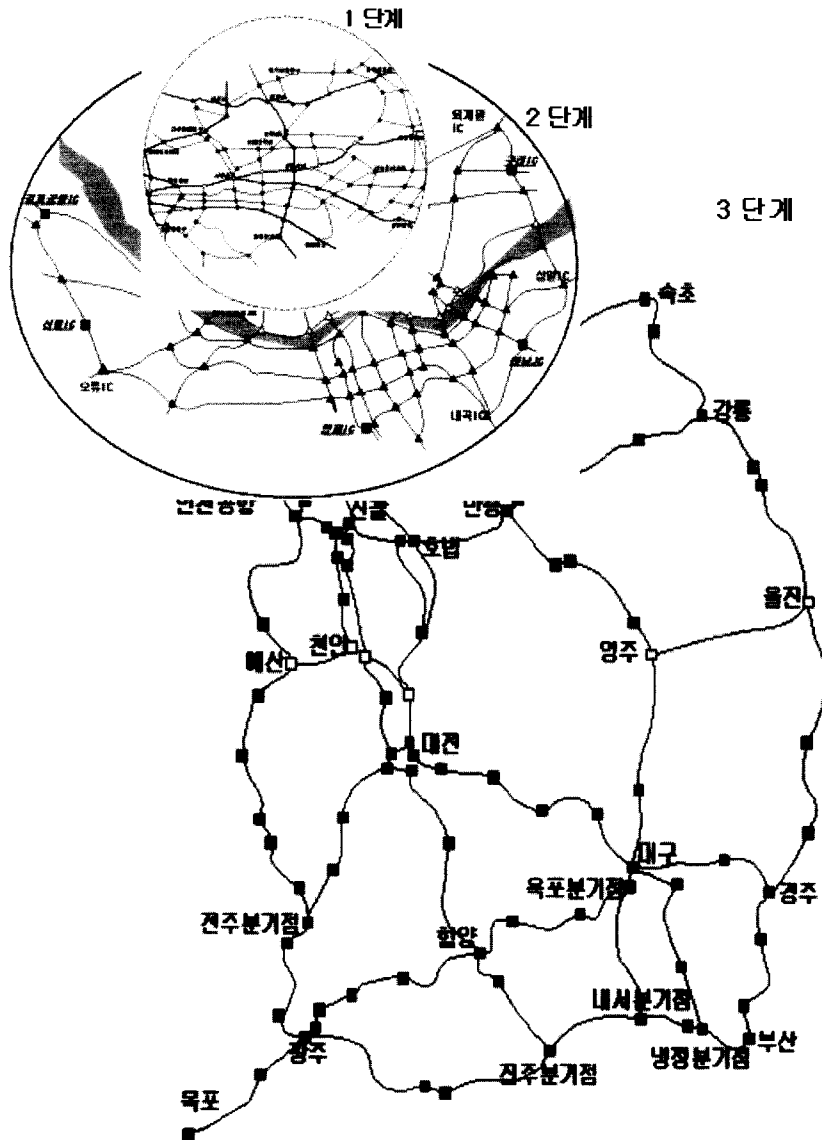
<그림 6> 전국 도로망

대규모 도로망의 양재IC에서 부산대학교쪽의 대규모 도로망에 존재하는 부산IC에 도착한다. 이 경우의 CGM에 의한 최단 거리는 427.95 Km가 된다.

이상의 고려대학교 → 부산대학교간의 성북-동대문구 도로망의 경로는

고려대학교 → 홍파초등학교앞 → 경동시장, 서울특별시 도로망의 경로는 경동시장 → 동마장사거리 → 왕십리로터리 → 성수대교북단 → 성수대교남단 → 르네상스호텔 → 현대체육관앞 → 개포IC → 염곡사거리 → 양재C,

전국 도로망의 경로는 양재C → 판교분기점 → 신갈분기점 → 수원IC → 오산IC → 천안분기점 → 남이분기점 → 회덕분기점 → 대전IC → 옥천IC → 영동IC → 김천IC → 구미IC → 왜관IC → 대구분기점 → 옥포분기점 → 내서분기점 → 냉점분기점 → 서김해IC → 부산IC 이다. 이상의 사례는 기점이 고려대학교이나 성북-동대문구의 어떤 노드도 기점이 될 수 있다. [절차 1]에서의 중규모 노드들인 신설동역, 경동시장, 월암교, 길음교와 [절차 2]에서의 대규모 노드들인 서울시청, 구리



<그림 7> 실제 도로망에서의 CGM 절차

〈표 2〉 구체적 경로 결정 과정

경로	소규모 도로망 고려대학교		중규모 도로망		대규모 도로망 부산IC	순위
	도착지	거리	도착지	거리	거리	
1	신설동	1.61	김포공항IC	31.61	474.41	
2			신월IC	27.61	466.41	
3			하남IC	21.61	430.21	4
4			구리IC	16.61	435.21	
5			서울시청	8.61	432.41	
6			양재IC	21.61	429.21	3
7	경동시장	1.35	김포공항IC	33.35	476.15	
8			신월IC	29.35	468.15	
9			하남IC	20.35	428.95	2
10			구리IC	14.35	432.95	
11			서울시청	10.35	434.15	
12			양재IC	20.35	427.95	1
13	월암교	2.09	김포공항IC	35.09	477.89	
14			신월IC	21.09	469.89	
15			하남IC	24.09	432.69	
16			구리IC	18.09	436.69	
17			서울시청	12.09	435.89	
18			양재IC	22.09	431.69	
19	길음교	3.11	김포공항IC	35.11	477.91	
20			신월IC	31.11	469.91	
21			하남IC	26.11	434.71	
22			구리IC	20.11	438.71	
23			서울시청	12.11	435.91	
24			양재IC	25.11	432.71	

IC, 상일IC, 양재IC, 신월IC, 김포공항IC는 전문가가 실제 도로망의 형태 및 연결구조를 보고 결정한 것이다. 이와 같은 방법은 쉽게 다른 지역으로 확장될 수 있다. 그러나 전문가가 전국의 노드 모두에서 이를 결정하기에는 많은 노력과 시간을 요한다. 따라서 앞에서 언급한 세부절차를 적용할 수 있다. 그러나 전문가의 결정 과정을 보다 더 효율적인 방법으로 개발하고 이를 검증하는 일은 앞으로의 과제이다. 참고로 구리IC와 하남IC로부터는 중부고속도로로 진입하며, 나머지 노드로부터는 경부고속도로로 진입한다. 위의 경로를 살펴보면 전국 규모의 도로망에서 대구분기점으로부터는 경부고속도로를 빠져 나와 구마고속도로를 거치고 있다. 이 경로가 CGM에 의한 최단 거리이기 때문이다. 경부고속도로를 계속 주행할 경우에는 428.38 Km가 된다. 그러나 대부분의 운전자들은 이와 같이 거리에 큰 차이가 있지 않으면 계속 같은 고속도로를 주행하는 것을 선호할 것이다.

따라서 앞으로 운전자의 선호도, 운전의 편의도, 안전

성, 혼잡도, 운행시간대 등을 반영하는 연구가 필요하다.

이제 CGM의 수치적 효율을 살펴보기 위하여 서울 특별시의 규모가 전국 규모의 1/10, 다시 성북/동대문구의 규모가 서울특별시 규모의 1/10이라고 하자. 이때 도로망의 개수는 전국 규모 1개, 시/도 규모 10개, 구/군 규모 100개가 된다. 상세 도로망내의 노드 수가 100개라면 전국 규모의 도로망에는 총 10,000개의 노드가 있게 된다.

따라서 일반 해법의 경우 1억(10,000²)개의 O-D 쌍에 대한 경로를 데이터베이스로 구축하여야 한다. CGM을 적용할 경우 하나의 도로망에는 1만(100²)개의 O-D에 경로가 구축된다. 따라서 총 111만(111x100²)개에 대한 경로가 필요하다. 이는 CGM을 적용하지 않을 경우의 대략 1/100에 해당된다. 그리고 세분화된 전국 도로망에서 Floyd-Warshall 알고리즘을 적용하여 모든 노드간의 최단 경로를 구하는데 필요한 연산의 수는 1조(10,000³), 변수의 저장 용량은 2억(2x10,000²)에 달한다. 그러나 CGM을 적용하여

각각의 망에서 모든 노드간의 최단 경로를 구하는데 필요한 연산의 수는 같은 알고리즘을 사용할 경우 1억 (111×10^8)이다. 결과적으로 CGM을 이용하여 경로를 구하는 것은 0.0001의 연산 회수와 0.0111의 저장 용량을 필요로 할 뿐이다. 연산의 수는 같은 알고리즘을 사용할 경우 1억(111×10^8)이다.

IV. 결론

이 연구에서는 경로 안내 시스템(RGS)의 핵심 기술인 경로 탐색 방법으로 단계별 세분화 방법(CGM)을 적용하였다. 기존의 탐색 방법과 비교하여 탐색 시간과 저장 용량을 현저하게 감소시켜 복잡한 대규모의 도로망에서 실시간 경로 조연을 가능하게 한다. 이 논문의 방법은 도로망의 규모가 커지고 구조가 복잡하여져도 신속성 및 효율성의 면에서 크게 영향을 받지 않는다.

구체적인 사례로서 이 시스템은 우리나라 전국 간선 도로망에서 100개 노드와 254개 링크, 서울특별시의 주요 도로망에서 102개 노드와 354개 링크, 그리고 성북-동대문구의 상세 도로망에서 96개 노드와 320개 링크를 대상으로 운영되었다. CGM에 의한 연산수의 절감은 99.99%, 저장 용량의 절감은 98.9%에 달한다. 그리고 이 시스템은 실시간(1초 이내)으로 경로를 탐색한다. 더욱 이 과정의 대부분은 오프라인으로 수행되어 저장될 수 있기 때문에 저장 공간이 확보되는 한 탐색 시간은 전혀 장애 요인이 되지 않는다.

CGM에는 DA 및 BS가 적용될 수 있다. 예를 들어 서울특별시에는 한강 이남과 이북을 분할하여 BS를 적용하고, 전국에는 충청-강원 이남과 이북을 분할하여 DA를 적용할 수 있다. 이 방법들의 적용으로 탐색시간은 더욱 단축될 것이다.

이 논문은 시간에 따라 교통 상황이 바뀌는 동적 환경을 고려하지 않고 있으나, 이경우에도 미리 전체 기중점에 대한 적절한 경로들을 찾아 저장해 놓으면 쉽게 실시간으로 최단경로를 제시할 수 있고, 상황에 따른 경로 변경도 가능하게 된다. 사실, 실시간 RGS를 위하여는 통행 거리, 과거와 현재의 통행 시간 평균 및 분산, 미래의 이에 대한 예측치, 운전자의 선호도, 신호 대기시간, 회전 제약 등의 방대한 자료를 실시간으로 수집, 분석하여야 하며, 또한 교통 혼잡, 교통 사고, 도로 보수 등의 도로 상황을 종합적으로 고려하여 이를 반영하여야 한다 (Kaysi, et al., 1993; Hounsell & Ishtiaq, 1997;

Lee, 1994; Sen, et al., 1998; Ziliaskopoulos, 1997; Ziliaskopoulou & Mahmassani, 1993; Yu & Young, 1998; Fu, 2001; Wahle, et al, 2001; Adler & Blue, 2002; Defflorio, 2003; Choi, et al., 2002; 김영찬·김태용, 2001). 한편 장인성·문형수(2001)는 기하구조적 특성과 가변적 특성이 부여된 도시도로망의 현실성을 반영하기 위하여 유전자 알고리즘을 이용하는 최단경로 탐색 방법을 제안하였다. 아울러 인터넷(Internet) 연계, 게시판 시스템(Bulletin Board System)과 오디오 응답 시스템(Audio Response System) 등을 통하여 운전자가 만족하는 정보 제공이 필요하다 (Thakuriah & Sen, 1996). 이 논문의 결과로 경로 탐색에서 얻어지는 단축 시간은 이와 같은 사항들을 실시간으로 고려하는데 사용될 수 있어 교통 정보의 품질 향상에 기여할 것이다.

이 논문에서는 최소한의 실험으로 CGM이 우리나라 전국의 도로망에 적용될 수 있음을 보였다. 많은 인력과 시간이 필요한 보다 더 완벽한 실험 및 실제 구축은 앞으로의 과제이다. 끝으로 이 연구는 고려대학교 교원 특별연구비 지원에 의하여 이루어졌으며, 이 연구의 내용은 특허 제 0437969호로 등록되어 있어, 관련 분야에서 적극적인 활용을 위한 협의를 기대한다.

참고문헌

1. 김영찬·김태용(2001), "검지자료합성을 통한 도시 간선도로 실시간 통행시간 추정모형," 대한교통학회지, 제19권 제6호, 대한교통학회, pp.171~182.
2. 김태진·한민홍(2001), "중앙집중형 도로교통정보 시스템에서 다중경로탐색 알고리즘," 대한교통학회지, 제19권 제6호, 대한교통학회, pp.183~194.
3. 장인성·문형수(2001), "유전자 알고리즘을 이용한 도시도로망에서의 첨단 여행자 정보시스템 운영계획," 대한교통학회지, 제19권 제4호, 대한교통학회, pp.85~96.
4. Adler, J. L., and Blue, V. J. (2002), "A Cooperative Multi-agent Transportation Management and Route Guidance System," Transportation Research. Part C, Emerging Technologies, Vol.10, pp.433~454.
5. Ahuja, R. K., Magnanti, T. L. and Orlin, J. B. (1993), Network Flows, Prentice Hall.

6. Backhouse, R. C., van den Eijnde, J. P. H. W. and van Gasteren, A. J. M. (1994), "Calculating Path Algorithms," *Science of Computer Programming*, Vol.22, pp.3~19.
7. Blewett, W. J. and Hu, T. C. (1997), "Tree Decomposition Algorithm for Large Networks," *Networks*, Vol.7, pp.289~296.
8. Choi, G. S., Seo, K. S. and Park, J. J. (2002), "Using Evolution Program to Develop Effective Search Method for Alternative Routes," *Journal of Korea Society of Transportation*, Vol.20, No.2, pp.71~79.
9. Chou, Y.-L., Romeijn, H. E. and Smith, R. L. (1998), "Approximating Shortest Paths in Large-scale Networks with an Application to Intelligent Transportation Systems," *INFORMS Journal on Computing*, Vol.10, No.2, pp.163~179.
10. Deflorio, F. P. (2003), "Evaluation of a Reactive Dynamic Route Guidance Strategy," *Transportation Research. Part C. Emerging Technologies*, Vol.11, No.5, pp.375~388.
11. Fu, L. (2001), "An Adaptive Routing Algorithm for In-vehicle Route Guidance Systems with Real-time Information," *Transportation Research Part B, Methodological*, Vol.35, pp.749~765.
12. Gallo, G. and Pallottino, S. (1988), "Shortest Path Algorithms," *Annals of Operations Research*, Vol.13, pp.3~79.
13. Giarratano, J. and Riley, G. (1998), *Expert Systems, Principles and Programming*, PWS Publishing Company.
14. Han, Y., Pan, V. Y. and Reif, J. H. (1997), "Efficient Parallel Algorithm for Computing All Pair Shortest Paths in Directed Graphs," *Algorithmica*, Vol.17, pp.399~415.
15. Hounsell, N. B. and Ishtiaq, S. (1997), "Journey Time Forecasting for Dynamic Route Guidance Systems in Incident Conditions," *International Journal of Forecasting*, Vol.13, pp.33~42.
16. Hu, T. C. (1968), "A Decomposition Algorithm for Shortest Paths in a Network," *Operations Research*, Vol.16, pp.91~102.
17. Hu, T. C. and Torres, W. T. (1969), "Shortcut in the Decomposition Algorithm for Shortest Paths in a Network," *IBM Journal of Research and Development*, Vol.13, pp.387~390.
18. Kaysi, I., Ben-Akiva, M. and Koutsopoulos, H. (1993), "Integrated Approach to Vehicle Routing and Congestion Prediction for Real-time Driver Guidance," *Transportation Research Record*, Issue 1408, pp.66~74.
19. Kim, S. I. & Kim, H. (2002), "Design and Implementation of an Intelligent System for Real-time Route Guidance," *IE Interface*, Vol.15, No.4, pp.374~381.
20. Lee, C. (1994), "A Multiple-Path Routing Strategy for Vehicle Route Guidance Systems," *Transportation Research: Part C*, Vol.2, No.3, pp.185~195.
21. Liu, B. (1997), "Route Finding by Using Knowledge about the Road Network," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol.27, No.4, pp.436~448.
22. Lysgaard, J. (1995), "A Two-phase Shortest Path Algorithm for Networks with Node Coordinates," *European Journal of Operational Research*, Vol.87, pp.368~374.
23. Mills, G. (1966), "A Decomposition Algorithm for the Shortest-route Problem," *Operations Research*, Vol.14, pp.279~291.
24. Namkoong, S., Rho, J. H. and Choi, J. U. (1998), "Development of the Tree-based Link Labeling Algorithm for Optimal Path-finding in Urban Transportation Networks," *Mathematical and Computer Modelling*, Vol.27, pp.51~65.
25. Nemhauser, G. L. (1966), *Introduction to Dynamic Programming*, John Wiley & Sons, Inc.
26. Nemhauser, G. L. (1988), *Integer and Combinatorial Optimization*, John Wiley & Sons, Inc.
27. Noto, M. and Sato, H. (2000), "A Method for the Shortest Search by Extended Dijkstra Algorithm," *IEEE International Conference*

- on Systems, Man, and Cybernetics, Vol.3, pp.2316~2320.
28. Park, C.-K., Sung, K., Doh, S. and Park, S. (2001), "Finding a Path in the Hierarchical Road Networks," 2001 IEEE Intelligent Transportation Systems Proceedings, pp.936~942.
 29. Romeijn, H. E. and Smith, R. L. (1999), "Parallel Algorithms for Solving Aggregated Shortest-path Problems," Computers and Operations Research, Vol.26, pp.941~953.
 30. Sanders, B. A. (2001), "The Shortest Path in Parallel," Information Processing Letters, Vol.77, pp.213~217.
 31. Sen, A., Soot, S., Thakuriah, P. and Condie, H. (1998), "Estimation of Static Travel Times in a Dynamic Route Guidance System - II," Mathematical and Computer Modelling, Vol.27, pp.67~85.
 32. Sen, A. and Thakuriah, P. (1995), "Estimation of Static Travel Times in a Dynamic Route Guidance System," Mathematical and Computer Modelling, Vol.22, pp.83~101.
 33. Shapiro, J., Waxman, J. and Nir, D. (1992), "Level Graphs and Approximate Shortest Path Algorithms," Networks, Vol.22, pp.691~717.
 34. Thakuriah, P. and Sen, A. (1996), "Quality of Information Given by Advanced Traveler Information Systems," Transportation Research: Part C, Vol.4, No.5, pp.249~266.
 35. Wahle, J., Annen, O., Schuster, C., Neubert, L. and Schreckenberg, M. (2001), "A Dynamic Route Guidance System Based on Real Traffic Data," European Journal of Operational Research, Vol.131, pp.302~308.
 36. Yu, G. and Yang, J. (1998), "On the Robust Path Problem," Computers and Operations Research, Vol.25, No.6, pp.457~468.
 37. Zhan, F. B. and Noon, C. E. (1998), "Shortest Path Algorithms: An Evaluation Using Real Road Networks," Transportation Science, Vol.32, No.1, pp.65~73.
 38. Ziliaskopoulos, A., Kotzinos, D. and Mahmassani, H. S. (1997), "Design and Implementation of Parallel Time-dependent Least Time Path Algorithms for Intelligent Transportation Systems Applications," Transportation Research: Part C, Vol.5, No.2, pp.95~107.
 39. Ziliaskopoulos, A. and Mahmassani, H. S. (1993), "Time-dependent, Shortest-path Algorithm for Real-time Intelligent Vehicle Highway System Applications," Transportation Research Record, Issue 1408, pp.94~100.

✉ 주 작 성 자 : 김성인
 ✉ 논문투고일 : 2003. 10. 7
 논문심사일 : 2003. 12. 4 (1차)
 2004. 4. 13 (2차)
 2004. 7. 7 (3차)
 2004. 8. 9 (4차)
 2004. 8. 24 (5차)
 심사판정일 : 2004. 8. 24
 ✉ 반론접수기한 : 2005. 2. 28