■ 論 文 ■

중앙집중형 도로교통정보시스템에서 다중경로탐색 알고리즘

Multiple Path-Finding Algorithm in the Centralized Traffic Information System

김 태 진
(고려대학교 산업시스템정보공학과 박사과정)

한 민 홍
(고려대학교 산업시스템정보공학과 교수)

목 차

Key Words : 최단경로탐색, ITS, 다중경로탐색, 교통정보시스템, CTIS

요 약

중앙집중형 도로교통정보시스템은 실시간 교통정보를 수집하고, 사용자의 요청을 받아 경로탐색, 위치정보, 목적지탐색 등의 정보를 전달해주는 시스템이다. 이러한 시스템에서 서버는 매우 많은 클라이언트로부터 경로탐색 요청을 받게 되며, 이 요청을 서버에서 효율적으로 처리해야 하는 다중경로탐색 알고리즘이 필요하다.

본 연구에서는 다중경로탐색을 수행하기 위하여, 주기적으로 연산된 이웃노드의 최단경로탐색 결과를 이용하여 클라이언트의 경로탐색 수행시간을 감소시키는 휴리스틱(Heuristic) 알고리즘을 제시한다. 본 연구에서 제시하는 이웃노드 최단경로탐색 결과를 이용한 다중경로탐색 알고리즘은 많은 경우에 최단과 동일한 결과를 나타내며, 최단이 아닌 경우에도 최단경로 값과 오차가 크지 않으면서도 연산시간을 많이 줄일 수 있는 알고리즘이고, 도로교통과 같은 토폴로지(Topology) 형태에 효과적으로 적용되고, 계층을 이루는 형태의 모델에서도 효율적인 결과를 나타낸다.

이웃노드 최단경로탐색 결과를 이용한 다중경로탐색 알고리즘의 경로탐색시간은 다른 꼬리표설정 알고리즘과 꼬리표개선 알고리즘보다 50배 이상 빨랐으며, 경로탐색 결과가 최단이 아닌 경우 0.1%이하의 거리오차가 발생했다.

Ⅰ. 서론

지능형교통시스템(ITS:Intelligent Transport Systems)에서 첨단교통정보시스템(ATIS:Advanced Traffic Information System)은 실시간 교통정보를 수집, 분석, 전달하는 역할을 담당한다. 그 중 도로와 관련된 부분을 도로교통정보시스템이라 부른다. 첨단도로교통정보시스템은 병원, 주유소, 호텔 등의 여행자가 필요로 하는 위치정보와 목적지 도착예정시간, 사전경로안내 등의 사전여행정보와 전방교통상황, 우회도로, 기상 등의 운전중의 정보를 제공한다. 이러한 시스템의 목적을 달성하기 위해서 지리정보시스템, GPS(Global Positioning System), 무선통신시스템, 입력장치(HID:Human Interface Device), 도로정보수집시스템과 경로탐색 알고리즘이 필요하다.

도로교통정보시스템은 클라이언트와 서버가 기능을 분담하는 방법에 따라 분산형 도로교통정보시스템(DTIS:De-centralized Traffic Information System)과 중앙집중형 도로교통정보시스템(CTIS:Centralized Traffic Information System)으로 분류될 수 있다. 중앙집중형 도로교통정보시스템은 경로탐색이 서버에서 모두 연산되어져야 하기 때문에 클라이언트의 경로탐색 요청이 증가함에 따라서 경로탐색 성능이 저하된다. 이처럼 동시에 많은 경로탐색을 수행하는 알고리즘을 다중경로탐색(Multiple Path-Finding) 알고리즘이라 부른다. 본 연구에서는 휴리스틱(Heuristic) 접근방법으로 도로교통정보시스템에 적합한 다중경로탐색 알고리즘을 제시한다.

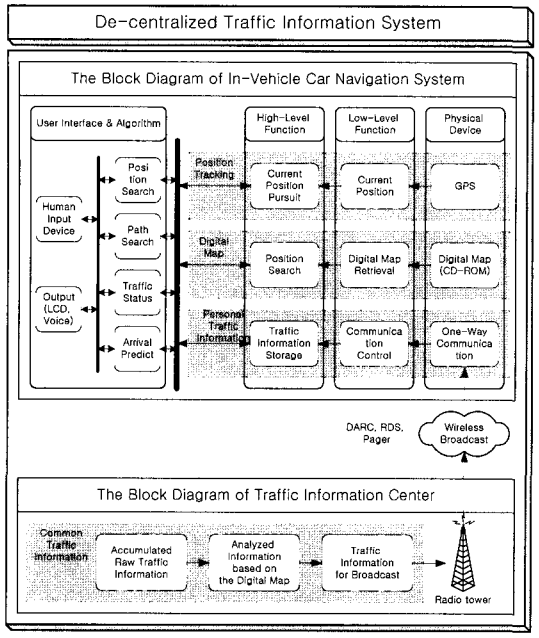
본 논문의 구성은 다음과 같다. Ⅱ장에서는 분산형 도로교통정보시스템과 중앙집중형 도로교통정보시스템을 비교하고, 중앙집중형 도로교통정보시스템으로 변해가는 이유를 설명한다. Ⅲ장에서는 중앙집중형 도로교통정보시스템에 적합한 다중경로탐색 알고리즘을 제시한다. Ⅳ장에서는 Ⅲ장에서 제안한 최단경로탐색 알고리즘 성능실험 결과를 보인 후 마지막으로 결론을 제시한다.

Ⅱ. 분산형 도로교통정보시스템과 중앙집중형 도로교통정보시스템

초기의 도로교통정보시스템의 대표적인 장치에는 차량항법시스템(CNS:Car Navigation System)이 있다. 차량항법시스템은 GPS, 전자지도, LCD, 최단경로탐색 알고리즘과 단방향 무선통신장치로 구성되어 있다. 일단 사용자에게 최단경로를 알려주기 위하여 DARC(DAta Radio Channel)와 같은 단방향 무선통신장치를 이용하여 도로교통정보서버로부터 최근의 도로교통정보를 수신한다. 그 후에 수신된 도로교통정보와 차량항법시스템에 저장된 전자지도 정보를 이용하여, 최단경로를 계산한 후 LCD를 통하여 사용자에게 알려주게 된다. 차량항법시스템과 같이 전자지도와 최단경로탐색 알고리즘이 차량쪽에 존재하고, 무선통신장치를 통하여 교통정보를 수신하는 시스템을 분산형 도로교통정보시스템(DTIS:De-centralized Traffic Information System)이라 부른다. 반면에 무선통신기술의 발전은 단방향 통신을 쌍방향 통신으로 변화시켰을 뿐 아니라 데이터전송 속도도 비약적으로 향상시켰다. 이런 환경의 변화로 사용자는 전자지도와 최단경로탐색 알고리즘을 차량에 장착하지 않고, 서버에서 필요한 전자지도 일부와 최단경로탐색 결과만을 받아볼 수 있는 시스템으로 변화하게 되었다. 이와 같이 전자지도와 최단경로탐색 알고리즘이 서버쪽에 존재하는 시스템을 중앙집중형 도로교통정보시스템(CTIS:Centralized Traffic Information System)이라 부른다(Laurent MAINARD, 2000; Takeshi Natsuno, 2000; TaeJin Kim, 1997; 석현우, 1998; 홍원철, 1999).

분산형 도로교통정보시스템은 최단경로탐색을 수행하기 위하여 단말기의 가격이 고가여야 하고, 도로교통정보가 단방향 통신으로 방송이 이루어져야 하며, 최단경로탐색 알고리즘은 단말기에서 이루어 진다. 분산형 도로교통정보시스템은 클라이언트 부분에 사용자 인터페이스, 현재위치추적, 단방향무선통신장비, 전자지도, 현재교통상황, 최단경로탐색의 모든 기능이 배치되게 되고, 서버 부분에서는 방송형통신장비, 전자지도, 현재교통상황의 일부 기능이 배치되게 된다(<그림 1>).

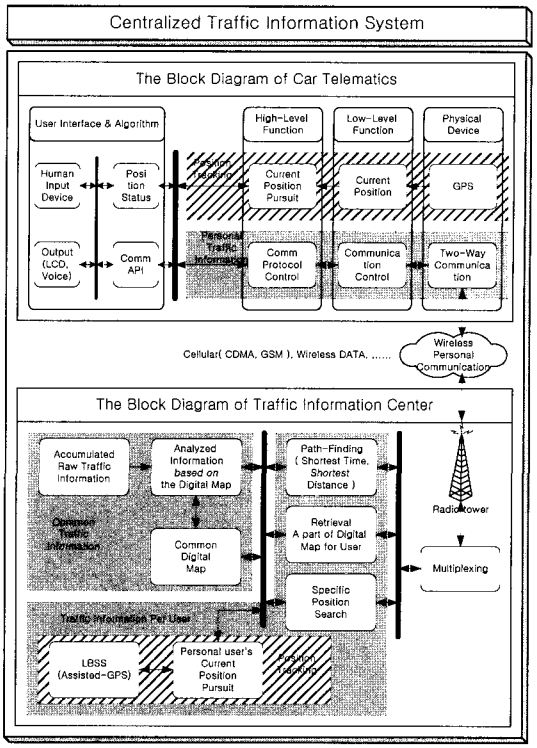
반면에 중앙집중형 도로교통정보시스템은 최단경로탐색을 수행하기 위하여 단말기의 가격이 상대적으로 저렴하다. 최단경로탐색과 전자지도를 도로교통정보서버가 보유하고, 단말기는 단순입출력 장비와 양방향 통신기술을 이용하게 된다. 또한 중앙집중형 도로교통정보시스템은 인터넷 망에서도 유사한 시스템형태로 서비스되는 것이 가능하다. 최근에 고속 유무선 통신기술의 발전에 힘입어 대량의 데이터 전송이 가



〈그림 1〉 분산형 도로교통정보시스템의 구성도

능해지면서, 새로운 형태의 도로교통정보시스템이 나타날 수 있게 되었다. 클라이언트 부분은 사용자 인터페이스, 양방향무선장비, 현재위치추적의 기능만을 보유하고, 서버부분은 양방향통신장비, 전자지도, 현재교통상황, 최단경로탐색, 현재위치추적의 대부분의 기능을 수행할 수 있게 되었다. 이러한 시스템은 사용자가 원하는 정보만을 해당 사용자에게 전송할 수 있으며, 단말기 제작비용을 절약할 수 있게 된다. 이러한 시스템을 중앙집중형 도로교통정보시스템이라 부른다. 한편 중앙집중형 도로교통정보시스템은 분산형 도로교통정보시스템에 비하여 시스템의 개선이 훨씬 용이하다. 분산형 도로교통정보시스템은 통신프로토콜, 전자지도, 교통정보상황이 클라이언트와 서버에서 동시에 정보의 정의 및 사용방법이 일치해야 하므로, 시스템 변경이나 개선 시에 모든 클라이언트를 동시에 개선시켜야 하는 부담을 안고 있다. 반면에 중앙집중형 도로교통정보시스템은 통신프로토콜의 개선만으로 시스템의 변경에 유연하게 대처할 수 있다. 또한 도로교통정보의 사용빈도에 따라 요금을 달리하는 시스템을 구축할 수 있어, 교통정보 사용자 부담원칙에 따라 요금을 부과할 수도 있다(〈그림 2〉).

중앙집중형 도로교통정보시스템에서 최단경로탐색을 수행할 때, 매우 많은 클라이언트들이 서버에 경로탐색

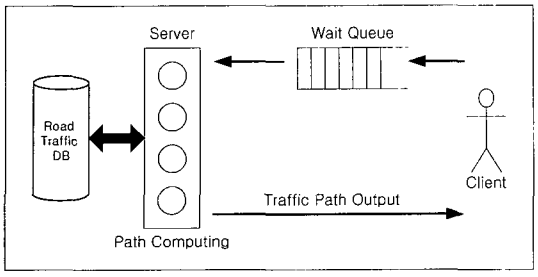


〈그림 2〉 중앙집중형 도로교통정보시스템의 구성도

요청을 하면 순서적으로 서버가 경로연산을 수행하게 된다. 다중경로탐색의 경우에 많은 클라이언트의 경로탐색 요청은 서버에서 처리를 기다리게 되며, 대기시간이 길어짐으로써 결과를 받는 시간까지 클라이언트는 오랜 시간을 기다려야 한다. 이러한 다중경로탐색의 성능을 높이기 위하여 본 연구에서는 도로교통네트워크의 특성이 성긴 네트워크 토폴로지(Topology)를 가진다는 점과 수초 내에 교통정보변경에 따른 최단경로가 변경될 가능성이 적다는 점, 그리고 이웃에 위치한 노드와 목적지가 유사한 경우 경로가 비슷할 가능성이 높다는 특성을 이용한다. 이러한 특성을 활용하여 다중경로탐색으로 이웃노드 최단경로탐색 결과를 이용한 다중경로탐색 알고리즘을 제시한다.

Ⅲ. 이웃노드 최단경로탐색 결과를 이용한 다중경로탐색

알고리즘다중경로탐색 알고리즘은 〈그림 3〉과 같이 하나의 서버시스템이 많은 클라이언트로부터 경로경로탐색 요청을 받아 처리하여 결과를 알려주는 시



〈그림 3〉 다중경로탐색 시스템 개념도

스템에 사용되는 알고리즘이다. 새로 도착한 클라이언트의 요청은 일단 서버의 대기행렬에 저장되어 이전에 요청되었던 경로탐색 요청이 처리되기를 기다린다. 일단 서버에서 경로탐색 요청처리가 완료되면 결과를 클라이언트에 알려주고 요청을 종료하게 된다. 서버가 여러 개의 프로세서로 업무를 분담하는 시스템 구성에서도 기본적으로 여러 개의 서버 프로세서가 클라이언트의 요청을 처리한다는 점 외에는 기본적으로 경로탐색 처리절차는 동일하다. 시스템의 성능은 서버에서 클라이언트의 요청을 빠른 시간에 처리해 주는 시간으로 평가되어진다.

일반적으로 최단경로탐색 알고리즘은 시점노드부터 종점노드까지 가장 적은 비용으로 도달할 수 있는 네트워크 그래프 상의 경로를 찾는 알고리즘이다. 알고리즘은 링크값이 변하지 않는 네트워크 그래프 상에서 시점노드와 종점노드가 오직 하나인 알고리즘, 한 시점노드에서 모든 노드까지를 탐색하는 알고리즘 그리고 모든 노드사이의 경로를 탐색하는 알고리즘으로 나뉘어 진다.(B. V. Cherkassky, 1996; E. W. Dijkstra, 1959; L. R. Ford, 1958; Narsingh Deo, 1984) 다중경로탐색 알고리즘은 시간에 따라 링크의 비용이 변하지 않는 네트워크 그래프 조건에서 여러 개의 시점노드와 여러 개의 종점노드를 연결하는 최단경로를 탐색하는 알고리즘이다(Ning Jing, 1998).

도로교통정보시스템에서 도로교통정보를 네트워크 모델링 하는 경우와 시간에 따라 변하는 도로환경정보는 다음과 같은 특성을 가지고 있다. 도로정보를 토폴로지 형태의 노드와 링크로 모델링 하는 경우 노드와 링크의 구성변화가 거의 발생하지 않거나 드물게 발생한다. 즉, 새로운 도로의 생성 및 파괴가 드물게 발생하여, 다중경로탐색 시스템 운영 중에 도로교통정보 데이터베이스의 노드와 링크의 토폴로지가 변할 가능성이 매우 희박하다. 한편 도로교통에서 링크의 시간비용정보는 시간에 따라 변동을 한다. 시간변동은 수초 내에 교통체증정도의 변화에 따라 그 시간 내에 최단경로가 변경될 확률이 매우 적다. 이러한 특성은 한 시점노드에서 다른 시점노드까지의 최적경로가 짧은 시간이 경과하여도 링크의 도로체증정도 변화에 따라 최단경로가 바뀌어 질 가능성이 작으며, 최단경로가 변경되어도 그 정도가 크지 않게 된다. 도로교통정보시스템 네트워크는 매우 많은 노드와 링크로 구성되어 있다. 노드는 교차로라는 정보를 대표하고, 링크는 교차로 사이의 연결도로를 대표함으로, 노드는 3개에서 5개 정도 사이의 링크와 연결되어 있다. 또한 네트워크의 연결정도가 복잡하지 않은 특성을 지니고 있다(F. Benjamin 1998; 김익기, 1998; 홍원철, 1999).

다중경로탐색 알고리즘은 시점노드와 종점노드가 한 쌍인 알고리즘을 클라이언트가 경로탐색을 요청할 때마다 수행하여 처리할 수 있다. 이러한 방법은 기본적으로 도로교통 네트워크의 특성 중 실제로 이웃한 노드와 종점노드가 동일한 경우 유사한 경로를 가지게 되는 특성을 고려하지 않으므로 성능을 향상시킬 수 없다. 또한 모든 노드 사이의 최단경로를 탐색하여 이를 데이터베이스에 저장한 후 시점노드와 종점노드가 일치하는 노드 사이의 정보를 찾아 경로탐색 결과를 알려줄 수도 있다. 이러한 방법은 주기적으로 네트워크의 모든 노드 사이의 최단경로를 찾아 데이터베이스에 저장하고 있어야 하므로, 변경주기마다 많은 시간을 소비하며 시점노드와 종점노드 사이의 경로를 모두 연산하고 저장하여야 한다. 역시 좋은 성능을 기대하기 어렵다(Mayiz B. 1994; Narsingh Deo, 1984; Yun-Wu 2000).

도로교통정보시스템에서 각 클라이언트의 요청은 다양한 시점노드와 종점노드의 쌍으로 이루어져 있다. 이러한 클라이언트의 요청 중 시점노드의 위치가 가까운 거리에 위치하는 요청이 많을 수 있다. 시점노드가 유사한 요청은 종점노드까지 비슷한 경로를 가질 가능성이 많다. 이러한 특성을 활용하여 본 연구에서는 이웃노드 최단경로탐색 결과를 이용하여 경로를 탐색하는 방법을 제안한다. 이 방법은 도로교통정보시스템 환경에서 휴리스틱으로 최단경로를 알려주지는 않지만 최단경로인 경우가 많고, 최단경로가 아니어도 최단경로와 큰 차이가 나지 않는 결과를 나타낸다.

1. 이웃노드 최단경로탐색 결과를 이용한 경로탐색 알고리즘

〈그림 4〉에서 임의의 이웃노드 P_0에서 종점노드P_t까지의 최단경로가 P_0, P_1, P_2, …, P_t이고, P_0, P_1, …, P_{r-1}까지 r개의 노드가 이웃노드영역 안에 존재하고, 시점노드 S는 이웃노드영역에 포함되어 있다. 이때 시점노드 S에서 종점노드 P_t까지의 경로는 식(1)에 의하여 결정한다.

$$d[S, P_t] = \min\{d^*[S, j] + d^*[j, P_t], \quad J \in P_0, P_1, \cdots, P_{r-1}\} \tag{1}$$

S : start node
P_t : end node
P_j : one node of optimal path which exists in the neghbor node area
d[i, j] : distance of from node i to node j
j*[i, j] : minimum distance of from node i to node j

이 경우 만약 P_2에서 가장 작은 값이 결정되어지면, 시점노드 S에서 종점노드 P_t까지의 경로는 S, P_2, …, P_t의 경로로 결정한다.

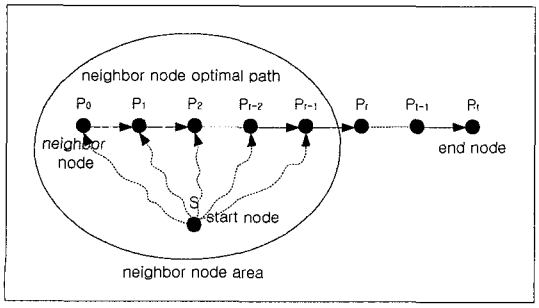
이웃노드 P_0에서 종점노드 P_t까지 최단경로는 이웃노드의 최단경로탐색결과로 이미 시점노드 S의 경로계획 전에 이미 알고 있는 값이다. 즉, $d^*[j, P_t]$값은 이미 알고 있는 거리값이 된다. 한편 $d*[S, j]$값은 시점노드에서 노드 j까지의 거리로 S노드에서 P_t까지 경로계획 시에 연산 되어야 한다. 이 연산은 시점노드 S에서 이웃노드영역 안에 존재하는 P_0, P_1, …, P_{r-1} r개 노드까지의 경로를 포함한다.

시점노드 S의 이웃노드가 여러 개인 경우를 살펴보면 〈그림 5〉와 같이 N_0, N_1, …, N_k의 k+1개의 이웃노드가 존재하는 경우 각각의 이웃노드에 대해서 식(1)을 적용하여 그 값들 중 가장 작은 값을 시점노드 S에서부터 종점노드 T까지의 경로로 결정한다. S의 이웃노드에서 종점노드 T까지의 경로는 시점노드 S를 거쳐갈 수도 있으며, 다른 이웃노드를 거쳐갈 수도 있다. 특히 시점노드 S를 거쳐가는 이웃노드가 존재하는 경우 시점노드 S에서 종점노드 T까지의 경로는 최단임을 쉽게 알 수 있다. 이를 식으로 나타내면 식(2)와 같다.

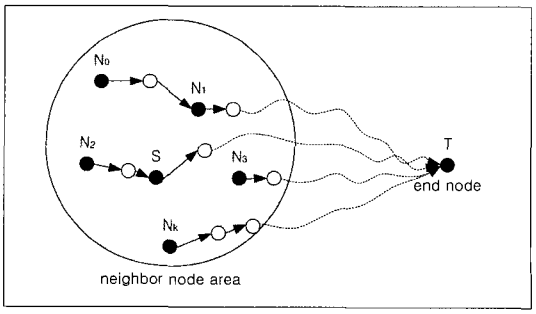
$$d[S, T] = \min\{d^*[S, n_{ij}] + d^*[n_{ij}, T], \quad i \in 0, 1, \cdots, k \;\; j \in 0, 1, \cdots, r_{k-1}\} \tag{2}$$

S : start node
T : end node
n_{ij} : one node of optimal path which exists in the neighbor node area i is neighbor node number, j is node number of optimal path node i which exists in the neighbor node area
d[i, j] : distance of from node i to node j
d*[i, j] : minimum distance of from node i to node j

이웃노드 최단경로탐색 결과를 이용한 경로탐색 알고리즘은 최적 알고리즘인 아닌 휴리스틱이다. 즉, 시점노드에서 종점노드까지의 경로가 최단임을 증명할 수 없다. 그러나, 시점노드 S에서 종점노드 T까지 최



〈그림 4〉 이웃노드 최단경로탐색 결과를 이용한 경로탐색



〈그림 5〉 여러 개의 이웃노드 최단경로탐색 결과를 이용한 경로탐색

단경로가 이웃노드영역 안에서 이웃노드 들로부터 종점노드까지의 최단경로의 어느 한 노드를 지나면 그 이후의 종점노드까지의 경로는 동일하게 결정한다. 이러한 특성은 종점노드 T를 향하여 진행하는 많은 경로를 알고 있으면, 시점노드 S에서 종점노드 T를 향하여 진행하는 새로운 경로도 종점노드에 가까워질수록 공통의 경로를 가지게 될 가능성이 높아지게 된다.

이처럼 이웃노드 최단경로탐색 결과를 이용한 경로탐색 알고리즘이 최적일 가능성을 높이는 방법은 이웃노드영역을 넓히는 것이다. 이웃노드영역이 넓어질수록 이웃노드의 수가 많아지고, 많아진 이웃노드는 종점노드를 향한 보다 많은 경로정보를 포함하게 된다. 또한 시점노드에서 종점노드까지의 최단경로의 한 노드와 만날 가능성도 높아진다. 그러나, 이웃노드영역이 넓어질수록 이웃노드의 수가 많아지면, 많은 이웃노드 정보를 찾아와야 하고, 시점노드에서 이웃노드의 최적경로 상에 존재하는 노드까지의 최단경로를 많이 탐색해야 하는 문제를 안고 있다. 결국 정확도와 연산시간은 두 특성을 감안하여, 이웃노드영역을 결정해야 한다. 다른 한가지 방법은 이웃노드영역을 늘리지 않고 동일한 영역에서 이웃노드의 수를 늘리는 방법이다. 이 방법 역시 이웃노드의 수가 늘어나 이웃노드의 최단경로를 미리 연산해야 하는 양이 많아져 상대적으로 성능이 떨어지게 된다. 이 또한 시스템에 따라 적절히 조정해야 할 필요가 있는 요소이다.

이웃노드 영역 설정은 두 가지 방법이 존재할 수 있다. 첫째는 시점노드에서 연산된 후보노드의 수가 한계노드 수에 존재하였을 때 종료하는 방법이다. 이 방법은 링크의 값이 변동하여도 이웃노드 영역이 일정하게 유지되는 장점이 있지만 네트워크 토폴로지가 밀집해 있는 지역과 성긴 지역에서 이웃노드 영역이 차이가 나게 된다. 둘째는 시점노드에서 일정거리 안에 존재하는 노드를 이웃노드 영역으로 정할 수 있다. 이 방법은 네트워크 토폴로지에 상관없이 이웃노드 영역이 결정되지만 링크의 값이 변함에 따라 이웃노드 영역이 변하는 특성을 지닌다.

알고리즘이 최단의 결과를 나타내지 못하는 경우는 이웃노드영역 거리보다 먼 곳에 종점노드가 존재하면서 시점노드에서 종점노드까지의 실제 최단경로가 이웃노드와 만나는 노드가 이웃노드영역 밖에 존재할 경우이다. 이런 경우가 발생하는 이유는 이웃노드영역 내의 이웃노드의 수가 적거나 이웃노드영역이 작아 시점노드에서 이웃노드의 최단경로 거치지 않는 경우에 발생한다.

본 알고리즘은 도로환경의 시간변화에 따라 링크의 정보가 변하는 정도를 미세하다는 가정에 의해서 시작한다. 이 의미는 일정주기마다 이웃노드의 최단경로탐색 정보가 변경되는 정도에 비하여 링크정보 변화정도가 작다는 의미를 포함한다. 만약에 돌발적인 사고나 예정된 도로통제에 의한 도로환경 변화 시 이웃노드의 최단경로탐색 정보를 즉시 변경해 주어야 한다. 일정주기를 감안하여 식(2)를 정리하면 식(3)과 같다.

$$d_t[S, T] = \min\{d_t^*[S, n_{ij}] + d_{t-1}^*[n_{ij}, T], \quad i \in 0,1,\cdots,k \quad j \in 0,1,\cdots,r_{k-1}\} \quad (3)$$

S : start node
T : end node
n_{ij} : one node of optimal path which exists in the neighbor node area i is neighbor node number, j is node number of optimal path node i which exists in the neighbor node area
t : neighbor node optimal path period
d[i, j] : distance of from node i to node j
d*[i, j] : minimum distance of from node i to node j

2. 중앙집중형 도로교통정보시스템에서 다중경로탐색 알고리즘

중앙집중형 도로교통정보시스템에서 다중경로탐색 알고리즘으로 본 연구에서는 이웃노드최단경로탐색 결과를 이용한 경로탐색 알고리즘을 사용한다. 본 알고리즘은 휴리스틱으로 최단은 아니지만 최단일 가능성이 높고, 최단이 아니어도 오류가 작음을 실험결과에서 볼 수 있다. 본 알고리즘을 시스템에 적용한 알고리즘의 절차는 다음과 같다.

[STEP 1] 일정한 시간주기 P와 이웃노드가 될 수

있는 주요 노드 N들을 결정한다. 또한 이웃노드영역을 시점노드 S에서부터 일정한 영역 B를 정한다.

〔STEP 2〕 도로교통상황의 변화에 따른 링크의 정보를 현재 링크의 값에 적용하여 수시로 적용한다.

〔STEP 3〕 일정 시간주기 P마다 이웃노드로 부터 모든 다른 노드까지의 최단경로를 연산하고 이를 저장한다.

〔STEP 4〕 클라이언트에서 시점노드 S부터 종점노드 T까지 경로탐색 요청을 받으면 시점노드로부터 일정영역 B안에 존재하는 모든 이웃노드를 검색하고, 그 이웃노드까지의 최단경로를 탐색한다.

즉, $d^*[S, n_{ij}]$을 연산하다.

〔STEP 5〕 해당 이웃노드영역에 속하는 모든 이웃노드를 찾아 각 이웃노드의 다른 노드까지의 최단경로정보를 저장된 공간에서 찾아온다.

즉, $d^*[d_{ij}, T]$정보를 찾는다

〔STEP 6〕 STEP 4와 STEP 5의 정보를 이용하여 다음에 의해 노드 n_{ij}를 결정한다.

$$d[S, T] = \min\{d^*[S, n_{ij}] + d^*[n_{ij}, T],$$
$$i \in 0, 1, \cdots, k \quad j \in 0, 1, \cdots, r_{k-1}\}$$

경로를 클라이언트에 알려준다. STEP 4로 가서 새로운 요청을 받는다.

Ⅳ. 실험방법 및 실험결과

본 실험은 Pentium Ⅲ 750Mhz CPU프로세서, 메모리 256 메가 바이트의 하드웨어와 Windows 2000의 소프트웨어 환경에서 실험하였다. 기본 비교대상 알고리즘은 시점노드와 종점노드사이의 경로를 과거의 정보를 사용하지 않으면서, 알고리즘의 성능이 보편적으로 가장 우수한 Double Bucket을 이용한 꼬리표설정 알고리즘(DIKBD), Heap을 이용한 꼬리표설정 알고리즘(DIKH), 그리고 두 개의 큐를 사용하는 꼬리표개선 알고리즘(TWO-Q)이다(B. V. Cherkassky 1996; Johnson D.B. 1977; Pallottino 1984). 비교대상 알고리즘과 다중 최단경로탐색 알고리즘은 C++언어로 코딩하여, Visual C++ 6.0으로 컴파일 하였으며, 컴파일러의 최적화 옵션은 사용하지 않았다. 실험에서 비교된 알고리즘의 요약은 〈표 1〉에 나타내었다.

〈표 1〉 실험에서 사용된 알고리즘의 요약설명

이름	요약 설명
TWO-Q	Pallottino의 Two Queue 알고리즘 (Pallottino 1984)
DIKH	k-array Heap 알고리즘 (Johnson D.B. 1977)
DIKBD	Double Bucket 알고리즘 (B. V. Cherkassky 1996)
HNN	이웃노드 최단경로탐색 결과를 이용한 휴리스틱(Heuristic) 알고리즘

이웃노드 최단경로탐색 결과를 이용한 경로탐색 알고리즘 실험을 위하여 이웃노드의 최단경로탐색 방법은 DIKH를 사용하였으며, 시점노드에서 가까운 이웃노드탐색은 DIKH의 변형을 사용하였다. DIKH에서 노드의 꼬리표(Label)가 설정되는 시점에 이웃노드인지 여부를 판단하여 이웃노드를 검색하였고, 종료시점을 현재노드의 최단거리가 이웃노드탐색 영역을 넘는 시점에서 종료하도록 하였다.

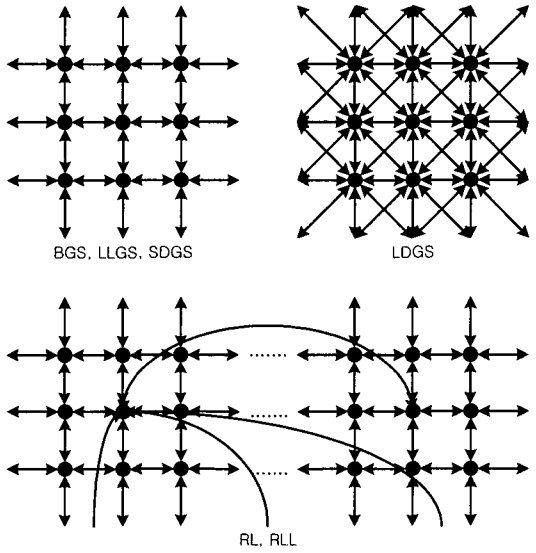
네트워크 그래프를 다양한 형태로 발생시켜 최단경로탐색 알고리즘의 성능을 비교하였다. 기본적으로 링크의 거리값은 상수분포[5000, 15000]를 사용하였고, 이웃노드 영역설정은 16384개의 일정노드 수로 정하였으며,. 표준정규분포를 따르는 실험도 일부 병행하였다. 그래프는 기본적으로 정방형으로 노드를 배열한 후 이웃한 노드에 링크를 연결시키도록 하였다. 또한 링크의 연결을 변형하여 이웃한 노드가 아닌 임의의 노드에 링크 연결을 만들어 실험하였다. 실험은 발생되어진 네트워크 그래프의 중앙노드를 시점노드로 정하고, 끝에 존재하는 노드를 종점노드로 정하여 주로 실험하였다. 실험에 사용된 네트워크 그래프는 〈표 2〉, 〈그림 6〉과 같이 성질을 변형해가면서 실험을 하였다. 각 실험은 그래프의 임의생성 값을 변형해 가면서 64회 반복 실험하여 측정한 시간결과를 평균하여 나타내었다(〈그림 7〉).

이웃노드 최단경로탐색 결과는 1024개의 노드마다 한 개씩을 선정하여 미리 최단경로탐색 결과를 메모리에 저장해 두었다. 본 알고리즘을 수행하여 얻어진 결과가 시점노드와 종점노드의 최단경로가 아닌 경우, 회수와 최단경로와의 차이를 평균하여 표에 나타내었다.

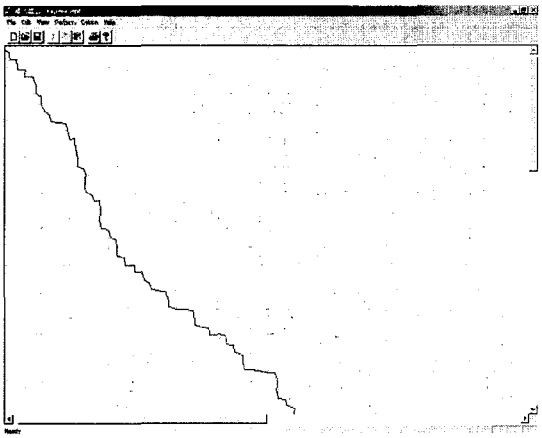
마지막으로 도로교통정보시스템에서 링크의 시간거

〈표 2〉 실험에 사용된 네트워크 그래프 발생방법

이름	요약 설명
Basic Grid Square(BGS)	정방형 그래프 구조의 기본형태 노드와 링크의 비는 1:4 링크의 길이 방생방법은 상수분포(5000, 15000)
Link Double Grid Square(LDGS)	BGS에서 링크의 연결을 두 배로 늘린다. 대각선으로 가까운 노드에 연결한다. 노드와 링크의 비는 1:8
Long Link Grid Square(LLGS)	BGS에서 링크길이의 차가 급격히 나타날 수 있도록 상수분포로 발생된 거리를 제곱한다.
Random Link(RL)	BGS에서 임의의 링크를 무작위로 추가한다. BGS의 링크길이를 임의의 링크길이보다 상대적으로 작게 한다. 노드와 링크의 비는 1:8
Random and Long Link(RLL)	RL에서 링크길이 발생방법을 상수분포에서 생성된 값에 제곱한다.
Standard-Dev Grid Square(SDGS)	BGS에서 링크길이 발생방법을 평균이 10000인 표준정규분포에 따라 발생시킨다.



〈그림 6〉 실험에 사용된 네트워크 그래프 발생방법의 그림 표현



〈그림 7〉 실험용 프로그램에서 BGS의 경로탐색 결과

리정보는 시간에 따라 조금씩 변해간다. 이 변화정도가 알고리즘에 미치는 영향을 판단하기 위하여, 과거 시점의 이웃노드 최단경로탐색에서 발생된 네트워크 그래프의 링크값을 일정비율로 변형해 가면서 실험하여 결과를 나타내었다.

실험결과는 〈표 3〉에서부터 〈표 8〉까지 나타내었다. 실험은 TWO-Q, DIKH, DIKBD는 최적값을 구하는 시간을 나타낸다. HNN의 초(seconds)는 연산의 결과를 얻는 데 소요되는 시간을 의미하며, 최적이 아닌 실험(No Optimal #) 수는 64회 실험에서 최적값이 나오지 않은 실험회수를 나타내고 있다. 마지막 칼럼의 평균 오차 비율(Avg err rate)는 최적이 아닌 실험결과를 나타낸 실험의 최적값으로 나눈 값을 평균한 것이다. 이 값은 최적값이 아닌 실험의 오차가 얼마나 차이가 나는 지를 나타내는 척도로 사용한다.

TWO-Q알고리즘은 BGS, LLGS, SDGS 그래프 환경에서 DIKH와 DIKBD보다 우수한 성능을 나타내었으나, LDGS, RL, RLL 그래프 환경에서는 성능이 DIKH와 DIKBD보다 급격히 저하됨을 보여줬다. 그 이유는 TWO-Q와 같은 꼬리표개선 알고리즘 계열은 링크의 구조가 복잡하여 후보노드집합(Candidate Node Set)에 한번 연산에 참여했던 노드가 재진입을 빈번히 발생시키게 되면, 성능이 나빠지게 된다. 반면에 DIKH와 DIKBD는 꼬리표설정 알고리즘 계열로 알고리즘이 비교적 링크의 연결 구조에 상관없이 안정적인 연산시간을 보여줬다. DIKBD가 DIKH보다 시간이 대체적으로 긴 연산결과를 나타낸 이유는 DIKHD는 링크의 길이에 따라 알고리즘의 성능이 달라지는 특징을 가지기 때문이다.

〈표 3〉 Basic Grid Square(BGS) 에서 알고리즘별 처리시간(64회 평균) (단위:sec)

Nodes/Links 1048576/4190208	TWO-Q	DIKH	DIKBD	HNN		
				seconds	No optimal #	Avg err rate
-30% change	0.801	1.196	2.347	0.041	0	0
-20% change	0.791	1.197	2.374	0.031	0	0
-10% change	0.802	1.196	2.353	0.024	0	0
0% change	0.791	1.195	2.345	0.020	0	0
10% change	0.803	1.199	2.356	0.017	37	0.00060
20% change	0.802	1.235	2.354	0.015	41	0.00070
30% change	0.802	1.205	2.357	0.013	46	0.00083

〈표 4〉 Link Double Grid Square(LDGS)에서 알고리즘별 처리시간(64회 평균) (단위:sec)

Nodes/Links 1048576/8380416	TWO-Q	DIKH	DIKBD	HNN		
				seconds	No optimal #	Avg err rate
-30% change	8.612	1.730	2.542	0.069	42	0.00082
-20% change	8.131	1.732	2.537	0.053	44	0.00089
-10% change	7.771	1.727	2.539	0.041	47	0.00088
0% change	8.052	1.729	2.543	0.032	49	0.00091
10% change	8.302	1.723	2.547	0.027	56	0.00259
20% change	8.472	1.730	2.547	0.023	57	0.00311
30% change	8.543	1.725	2.549	0.019	57	0.00330

〈표 5〉 Long Link Grid Square(LLGS)에서 알고리즘별 처리시간(64회 평균) (단위:sec)

Nodes/Links 1048576/4190208	TWO-Q	DIKH	DIKBD	HNN		
				seconds	No optimal #	Avg err rate
-30% change	0.841	1.235	2.137	0.047	0	0
-20% change	0.841	1.238	2.138	0.036	0	0
-10% change	0.841	1.240	2.136	0.027	0	0
% change	0.841	1.233	2.135	0.023	0	0
10% change	0.842	1.233	2.133	0.018	33	0.00131
20% change	0.841	1.233	2.139	0.014	39	0.00149
30% change	0.842	1.234	2.135	0.014	41	0.00174

〈표 6〉 Random Link(RL)에서 알고리즘별 처리시간(64회 평균) (단위:sec)

Nodes/Links 1048576/8380416	TWO-Q	DIKH	DIKBD	HNN		
				seconds	No optimal #	Avg err rate
30% change	60.104	2.574	3.573	0.138	64	0.01716
-20% change	61.342	2.575	3.584	0.107	64	0.01798
-10% change	58.567	2.576	3.579	0.083	59	0.01438
0% change	59.566	2.574	3.571	0.067	59	0.01322
10% change	59.934	2.572	3.578	0.052	59	0.01363
20% change	60.254	2.575	3.577	0.042	60	0.01388
30% change	59.395	2.576	3.579	0.034	60	0.01502

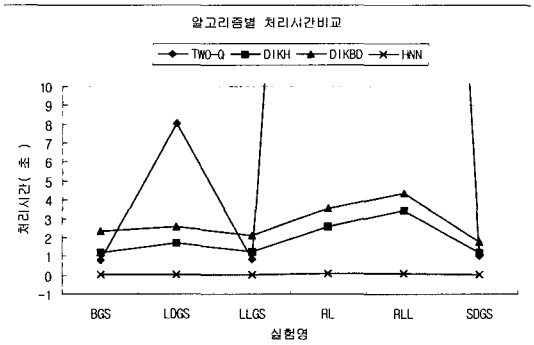
〈표 7〉 Random and Long Link(RLL)에서 알고리즘별 처리시간(64회 평균) (단위:sec)

Nodes/Links 1048576/8380416	TWO-Q	DIKH	DIKBD	HNN		
				seconds	No optimal #	Avg err rate
-30% change	67.456	3.425	4.365	0.142	64	0.01865
-20% change	69.675	3.426	4.367	0.127	64	0.01875
-10% change	68.165	3.425	4.374	0.087	60	0.01578
0% change	68.566	3.424	4.378	0.078	60	0.01422
10% change	68.798	3.424	4.375	0.064	60	0.01534
20% change	67.354	3.426	4.377	0.052	61	0.01586
30% change	68.476	3.427	4.379	0.043	61	0.01505

〈표 8〉 Standard-Dev Grid Square(SDGS)에서 알고리즘별 처리시간(64회 평균) (단위:sec)

Nodes/Links 1048576/4190208	TWO-Q	DIKH	DIKBD	HNN		
				seconds	No optimal #	Avg err rate
-30% change	1.054	1.206	1.788	0.044	0	0
-20% change	1.035	1.207	1.765	0.037	0	0
-10% change	1.043	1.207	1.789	0.026	0	0
0% change	1.034	1.204	1.789	0.033	0	0
10% change	1.038	1.203	1.790	0.019	27	0.00074
20% change	1.056	1.124	1.786	0.017	36	0.00078
30% change	1.044	1.227	1.775	0.015	38	0.00093

본 연구에서 제시하는 HNN알고리즘은 〈그림 8〉에서 같이 다른 알고리즘보다 연산시간이 많게는 100배에서 작게는 30배까지 성능이 우수함을 알 수 있다. BGS, LLGS, SDGS 그래프 구조에서 1024개 중 하나마다 이웃노드의 연산결과를 알고 있을 경우, 64회 실험에서 모두 최적값을 얻었으며, LDGS, RL, RLL의 링크의 구조가 복잡한 구조에서 실험은 대체적으로 최적값을 얻지는 못하였다. 그러나, 최적값과의 차이가 0.1%로 그 차이가 경미하였음을 보여주었다.



〈그림 8〉 링크값의 변동이 없는 경우의 각 실험별 알고리즘 처리시간 비교

V. 결론

본 논문에서는 중앙집중형 도로교통정보시스템과 분산형 도로교통정보시스템의 구성이 차이점에 대해서 설명하였고, 그 차이의 기본적인 원인은 무선통신의 기술의 발전에 의해 진행되었음을 설명하였다. 또한 도로교통정보시스템이 분산형에서 중앙집중형으로 변해가면서 많은 클라이언트의 경로계획 요구를 처리해 주는 다중경로탐색 알고리즘의 필요성에 대해서 언급하였다. 이러한 다중경로탐색 알고리즘으로 도로교통정보시스템에서 효율적인 결과를 나타내는 이웃노드 최단경로탐색결과를 이용한 경로탐색 알고리즘을 제안하여 그 실험적인 결과를 나타내었다.

이웃노드 최단경로탐색결과를 이용한 경로탐색 알고리즘은 휴리스틱으로 최단임을 증명할 수 는 없다. 그러나, 다른 알고리즘보다 50배 이상의 빠른 시간적인 효과를 거뒀으며, 최단이 아닌 경우의 오차는 0.1%로 나타났다. 다만 실험결과의 비교를 최단경로탐색 결과와 비교하였으며, 유사한 휴리스틱 알고리즘과 실험결과를 비교하지는 못하였다. 본 알고리즘

은 도로교통정보시스템의 토폴로지에 잘 적용되며, 도로환경이 계층적인 구조로 디자인되면 더욱 효과가 잘 나타날 것임을 예상한다.

참고문헌

1. B. V. Cherkassky, A. V. Goldberg and T. Radzik(1996), "Shortest paths algorithms: Theory and experimental evaluation", Mathematical Programming 73, pp.129~174, 1996.
2. E. W. Dijkstra, "A note on two problems in connection with graphs", Numerical Mathematics 1, pp.269~271, 1959.
3. F. Benjamin Zhan, Charles E. Noon, Shortest Path Algorithms: An Evaluation using Real Road Networks, Transportation Science, Vol. 32, No. 1, 1998, pp.65~73.
4. Goran swedberg, Ericsson's mobile location solution, Ericsson Review, No. 4, 1999, pp.214~221.
5. GPS NAVSTAR, Global Positioning System Standard Positioning Service Signal Specification, 2nd Edition, 1995.
6. GPS Positioning System, Vol. 1,2,3,4, The Institution of Navigation, 1980.
7. Johnson D.B., "Efficient special-purpose priority queues", Proceedings of the 15th Annual Allerton Conference on Communication, Control and Computing, pp.1~7, 1977.
8. Laurent MAINARD, Oliver PERRAULT, Synapsis: a Palm-Based System to Provide ITS through Cellular Networks, ITS2000 conference, 2000, 2404.pdf.
9. Location-Based Services System(LBSS) version 1.0.0, 3GPP2 S.R0019, 2000.
10. L. R. Ford Jr., Network flow theory, Rand Co., p.293, 1958.
11. Mayiz B. Habbal, Haris N. Koutsopoulos, Steven R. Lerman, A Decomposition Algorithm for the All-Pairs Shortest Path Problem on Massively Parallel Computer Architectures, Transportation Science, Vol. 28, No. 4, 1994, pp.292~308.
12. Narsingh Deo and Chi-yin Pang. "Shortest-Path Algorithms: Taxonomy and Annotation", Networks 14, pp.275~323, 1984.
13. Ning Jing, Yun-Wu Huang, and Elke A. Rundensteiner, Hierarchical Encoded Path Views for Path Query Processing: An Optimal Model and Its Performance Evaluation, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, Vol. 10, No. 3, 1998, pp.409~432.
14. NMEA Standard for Interfacing Marine Electronic Devices, National Marine Electronic Association, 1992.
15. Pallottino. S., "Shortest-Path Methods: Complexity, Interrelations and New Propositions", Networks, 14, pp.257~267, 1984.
16. Takeshi Natsuno, David Macdonald, Junji Asatssuma, Traffic and Traveller Information on the Mobile Phone Terminal, ITS2000 conference, 2000, 3154.pdf.
17. TaeJin Kim, TaekJun Jang, HyungSung Yoon, MinHong Han, Development of Personal Traffic Information on Demand Service (PTIODS), ITS1997 conference, 1997.
18. Yun-Wu Huang, Ning Jing, Elke A. Rundensteiner, Optimizing path query performance: graph clustering strategies, Transportation Research Part C, 8, 2000, pp.381~408.
19. 김익기, ATIS를 위한 수정형 덩굴망 최단경로 탐색 알고리즘의 개발, 대한교통학회지, 제16권 제2호, 1998, pp.157~167.
20. 박성진 · 이중원 · 지규인 · 이영재 · 이장규 · 최홍석 · 김진대, 무선통신 시스템에서의 위치측정 방식의 성능분석, TELECOMMUNICATION REVIEW, 제9권 제1호, 1999, pp.2~16.
21. 석현우 · 한민홍, 중앙집중형 교통정보 시스템에 관한 연구, 대한교통학회지, 제16권 제4호, 1998, pp.127~137.
22. 홍원철 · 명선영, ITS를 위한 차량항법시스템 개발, TELECOMMUNICATION REVIEW, 제

9권 제4호, 1999, pp.551~565.
23. 성태경·명선영·홍원철, 회전제한이 있는 도로망을 위한 고속 최적경로 알고리즘, 대한교통학회지, 제17권 제3호, 1999, pp.73~85.
24. FM부가방송망을 이용한 ITS 기술세미나, 한국전파진흥협회, 1998.

♧ 주 작 성 자 : 김태진
♧ 논문투고일 : 2001. 9. 3
논문심사일 : 2001. 10. 22 (1차)
2001. 11. 19 (2차)
2001. 11. 30 (3차)
심사판정일 : 2001. 11. 30