도로 네트워크 환경에서 운행 시간을 고려한 최단 경로 탐색 연산자 설계

이동규, 이양구, 정영진, 류근호
충북대학교 전자계산학과
e-mail : {dglee, leeyangkoo, yjjeong, khryu}@dblab.chungbuk.ac.kr

Design of Operator for Searching Trip Time Dependent Shortest Path in a Road Network

Dong Gyu Lee, Yang Koo Lee, Young Jin Jung, Keun Ho Ryu
Dept. of Computer Science, Chungbuk National University

요 약

최근 도로 네트워크 환경에서 날로 증가하는 교통 수요를 충족시키고 각종 교통 문제를 해결하기 위해서 지능형교통시스템(ITS, Intelligent Transportation System)을 적용하고 있다. 특히, 첨단교통정보 시스템(ATIS, Advanced Traveler Information System)은 개별 차량의 주행을 최적화시키는 시스템으로서 운전자에게 출발지에서 목적지까지 빠르고 쾌적한 주행경로를 제공하는 차량 경로계획 수립을 제공한다. 하지만 이러한 시스템은 도로 구간의 비용으로 정적인 값을 이용하므로 동적으로 변화하는 구간 비용을 가지고 도로 네트워크에서 최단 경로를 제공하기는 어렵다. 따라서, 이 논문에서는 교통 혼잡을 고려한 최단 경로 탐색 연산자를 제안한다. 제안된 연산자는 현재 시간 비용과 과거의 시간 비용 변화 량을 더하여 출발지에서 목적지까지 경로를 탐색하는데 이용한다. 이러한 방법은 시간에 따라 변화하는 도로의 상황을 반영하며 출발지에서 목적지까지의 최단 경로뿐만 아니라 예상 도착 시간을 추정할 수 있다. 또한 제안된 연산자는 효율적인 도로 이용, 물류비용 감소, 응급 상황 대체, 연료 절약 및 환경 오염 감소 등의 장점을 가지며 첨단교통정보시스템에서 응용 될 수 있다.

1. 서론

차량을 이용하는 사람들은 일상적인 생활 속에서 출발지부터 특정 목적지까지 어떠한 경로로 이동하는 것이 최적 또는 최상인지를 결정해야 하는 문제에 항상 직면한다. 그리고 또 하나의 문제는 최적 또는 최상의 결정이 교통 혼잡 비용 등을 모두 통합한 전체 비용을 감안해서 최상의 결정을 내려야 한다.

하지만 이러한 의사결정을 지원할 수 있는 수단이 부족하기 때문에 대부분의 사람들은 제한된 범위의 정보를 바탕으로 그때 그때 상황에 따라 임의로 운행 경로를 결정하기 마련이다. 최근 길안내 모바일 서비스, 자동차 네비게이션 시스템, 웹서비스 등이 등장하고 있지만 이들 모두 제한된 범위에서의 정보만 제공하기 때문에 이와 같은 문제를 해결하고, 교통체계의 효율성과 안정성을 제고하기 위해서는 기존의 교통체계에 전자, 정보, 통신 그리고 제어 등의 지능형 기술을 접목시킨 차세대 교통체계가 필요하다.

첨단 교통 정보 시스템은 개별 차량의 주행을 최적화 시키는 시스템으로서 운전자에게 출발지에서 목적지까지 빠르고 쾌적한 주행경로를 제공하는 차량 경로계획 수립을 제공한다. 그리고, 도로 네트워크에서 교통혼잡, 신호대기, 교통사고, 공사현장 등 교통의 흐름을 방해하는 많은 요인들이 있기 때문에 최단 경로를 찾고 목적지까지 소요 시간을 구하는데 단지 도로 네트워크의 거리 등의 정적인 정보만을 가지고 최단 경로를 탐색해 주는 것에는 많은 어려움이 따른다. 또한 목적지까지 소요 시간을 어떤 방법으로 추정 할 것인가에 대한 문제 또한 미지수로 남는다.

이 연구는 University IT Research Center 프로젝트의 지원으로 연구되었음

따라서 이 논문에서 도로 네트워크에서 정적인 정보뿐만 아니라 동적인 정보도 함께 고려하여 최단 경로와 목적지까지 소요 시간을 예측할 수 있는 연산자를 제안한다. 제안된 연산자는 각 구간의 현재 운행 시간 비용과 과거 운행 시간 비용의 변화 량을 더하여 실제 구간을 통과하는 운행 시간 비용을 계산한다.

이로 인해, 효율적인 도로 이용, 물류비용 감소, 빠른 응급 상황 대체, 운행시간 감소 및 연료 절약으로 인한 환경오염 감소 등의 장점을 가지고 있다.

이 논문의 내용은 다음과 같이 구성한다. 2 장에서는 최단 경로 연산자에 관한 관련 연구를 간략히 설명하고, 3 장에서는 도로 네트워크에서 구간의 시간 비용을 생성하는 방법에 대해 설명한다. 그리고 4 장에서는 교통 혼잡을 고려한 최단 경로 탐색 연산자를 제안한다. 마지막으로 5 장에서는 결론을 맺고 향후 연구 방향을 제시한다.

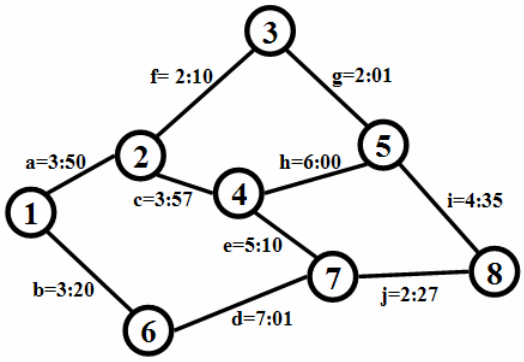
2. 관련 연구

[1]에서 Güting 은 도로 네트워크에서 위치, 최단 경로 계산, 그리고 시간을 고려한 최단 경로를 연산할 수 있도록 다음과 같은 연산자들을 제안하였다. Shortest_path 는 출발지로부터 목적지까지 최단 경로를 반환 한다. Trip 은 시간을 고려한 최단 경로를 생성하는 연산자이다. Trip 은 출발 시간에 출발지에서 목적지까지 이동하는 점 객체를 서술적 묘사로 반환한다. 이러한 계산은 시간마다 적정한 속도로 도로의 각 구간을 운행하는 차량으로 미리 가정하였다. 따라서 Shortest path 연산자는 네트워크 거리 비용으로 최단 경로를 연산하는 연산자이고 trip 은 시간을 고려하지만 시간이라는 비용이 명확하지 않고 단순히 서술적인 묘사로 연산자 값을 대신하는 정도이다. 이것은 도로 네트워크에서 동적으로 변화하는 네트워크 비용을 제대로 반영할만한 확실한 근거가 없는 것이고 실제 노드들이 많은 네트워크나 연결이 복잡한 네트워크에서 단순히 서술적인 묘사로 결과를 반환하는 것은 GUI 로 결과를 보여주는 것보다 효율성이 낮다. 이와 같은 연산자와 관련 있는 알고리즘은 [2]에서 설명하였다.

[2]에서는 시간을 고려한 최단 경로를 예측하는 시스템을 구현했다. [2]에서 말하는 시간은 최고 속도와 교통량을 고려한 시간이다. 도로에서 수집한 속도와 교통량을 가지고 최단 경로 알고리즘이 아닌 시스템을 통해 최단 경로를 제공한다. 하지만 이 시스템은 시간과 거리간의 Trade-off 가 발생하고 시간 이득을 얻는 그룹과 손실을 입는 그룹이 존재하여 형평성에 어긋나는 문제가 있다. 즉, 시스템이 제안한 경로를 따라 차량이 운행될 때, 제안된 경로에 먼저 진입한 차량들은 운행 시간에 이득을 얻지만 나중에 진입한 차량들은 다소 처음 제안된 시점보다 제안된 경로가 혼잡한 경로로 변화되므로 운행 시간의 이득이 거의 없다.

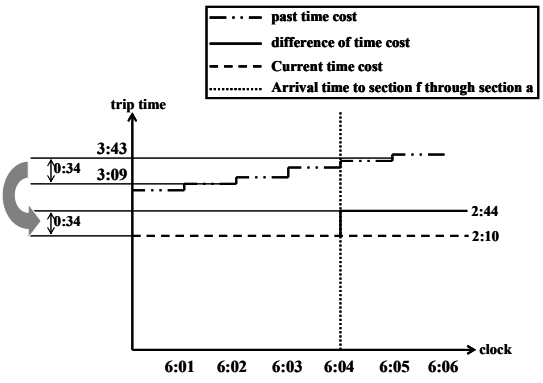
3. 도로 구간의 시간 비용 생성 방법

도로 네트워크에서 구간의 비용은 통행 시간이다. 전제 조건으로 차량은 GPS 로부터 시간, 좌표와 속도 값을 받아 중앙에 있는 서버로 전송하고 서버는 받은 데이터를 시간으로 계산하여 저장한다. 이렇게 모인 정보는 차량의 최단 경로 서비스에 사용된다. 기존의 연구처럼 구간의 시간 비용을 가지고 단순히 최단 경로 탐색 알고리즘에 적용하는 것은 시간의 흐름에 따라 구간의 통행 시간이 변화는 것을 적용하지 못하는 문제점이 있다. <표 1>과 (그림 1)을 가지고 도로 네트워크에서 구간의 시간 비용 생성에 관하여 설명한다.



(그림 1) 8 개의 노드들을 가진 그래프

예를 들어, 노드 1 에서 노드 3 까지 이동한다고 가정하자. (그림 1)은 노드 1 에서 출발해서 노드 2 로 가는데 시간 비용이 3 분 50 초가 소요되고 노드 2 에서 노드 3 으로 이동하는데 2 분 10 초가 소요된다는 의미이다. 그러므로 노드 1 에서 출발에서 노드 2 를 거쳐 노드 3 에 도착하는데 6 분이 소요된다. 하지만 노드 1 에서 출발해서 노드 2 에 도착하는 3 분 50 초 동안 섹션 f 의 시간 비용이 변화될 수도 있기 때문에 단순히 현재 각 구간의 시간 비용만으로는 도로의 속성이 동적으로 변화하는 도로 네트워크에서 최단 시간 경로를 탐색하기 어렵다. 그러므로 이 문제를 해결하기 위해서 현재 위치한 노드의 현재 통행 시간 비용과 이전 구간의 통행 시간 동안 현재 구간의 변화 량을 더하여 구간의 예상 통행 시간을 추정하는 방법을 제안한다.



(그림 2) 구간 f 의 시간 비용 생성 방법

<표 1>은 1 분 간격으로 모든 구간의 현재 시간 비용(*t*)만을 저장한다. 현재 시간 비용(*t*)은 구간의 거리(*d*)와 그 구간 객체들의 평균 속도(*v*)로 구할 수 있다. 차량이 구간 f 에 진입하기 전의 시간 비용(*bc*)은 (그림 2)에서 보는 것처럼 3:09 이다. 구간 f 에 진입한 이후의 구간의 시간 비용(*ac*)은 3:43 이다. 식 (2)에 의해 과거 f 구간의 통행 시간 변화 량은 0:34 이다. (그림 2)에서와 같이 이 값을 적용하여 구간 f 의 현재 시간 비용(*t*)인 2:10 에 과거 f 구간의 통행 시간 변화 량(*dif*)인 0:34 를 더한다. 이와 같은 방법으로 구간 f 의 예상 통행 시간 비용(*T*)은 2:46 가 된다. 이와 같은 방법으로 그래프에서 출발지와 목적지의 예상 통행 시간 비용을 추정할 수 있다.

<표 1> 각 구간의 통행 시간 저장 자료

time / section	06:00	06:01	06:02	06:03	06:04	06:05	06:06	06:07	06:08	06:09	06:10
a	03:01	03:04	03:00	03:20	03:15	03:02	03:10	03:30	03:46	04:05	04:09
b	03:08	03:15	03:19	03:19	03:18	03:26	03:29	03:30	03:27	03:23	03:29
c	04:45	04:57	04:59	04:47	04:53	04:37	04:48	04:23	04:21	04:14	04:10
d	07:50	07:45	07:43	07:41	07:39	07:37	07:35	07:33	07:31	07:29	07:27
e	05:19	05:21	05:22	05:25	05:27	05:30	05:28	05:25	05:23	05:21	05:20
f	03:01	03:05	03:09	03:14	03:29	03:41	03:44	03:46	03:49	03:52	03:53
g	04:29	04:27	04:26	04:26	04:26	04:30	04:31	04:32	04:24	04:20	04:16
h	05:00	05:03	05:06	05:09	05:11	05:16	05:19	05:30	05:36	05:43	05:50
i	03:50	03:54	03:52	03:51	03:50	03:48	03:47	03:46	03:45	03:44	03:41
j	02:27	02:25	02:27	02:11	02:09	02:10	01:59	01:58	01:54	01:51	01:45

식 (1)은 도로 네트워크의 구간의 거리(*d*)를 객체들의 평균 속도(*v*)로 나누어 현재 통행 시간 비용(*t*)을 구한다. 식 (1)에 의해서 <표 1>과 같은 테이블을 만든다. 또한 이 테이블을 바탕으로 식 (2)에서 과거 통행 시간 변화 량을 구할 수 있다. 그러므로 식 (1)의 결과(*t*)는 구간의 현재 시간 비용이고 식 (2)의 결과(*dif*)는 과거 정보에서 시간에 따른 변화 량을 반환한다. 식 (1)과 식 (2)의 결과를 합하여 식 (3)과 같이 시간에 따라 변화하는 각 구간의 예상 통행 시간 비용을 추정할 수 있다.

$$t = d/v \qquad (1)$$

$$dif = ac - bc \qquad (2)$$

$$T = t + dif \qquad (3)$$

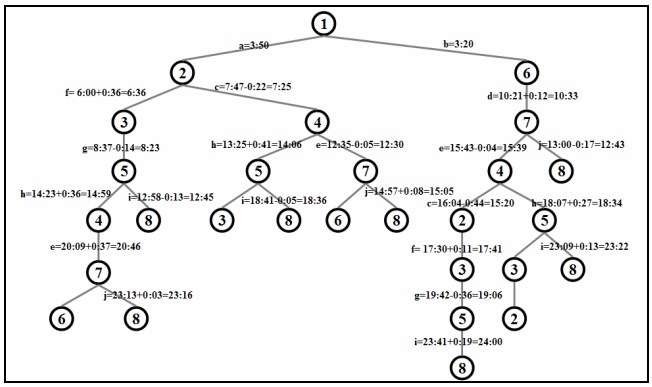
Where

t =current trip time cost
d= distance of the section
v= average velocity of the vehicle
dif=difference between after costs and before costs among the past information
bc=consumed time cost from node u to node v
ac=current time cost of node v from past informa-tion
T = total time cost of the section

따라서, 노드 1 에서 노드 2 까지는 3 분 50 초가 소요되었고 노드 2 에서 노드 3 까지는 2 분 10 초가 소요되는데 교통혼잡, 날씨 등의 문제로 노드 1 에서 노드 2 를 통과하는 동안 34 초가 증가하는 변화를 예상할 수 있으므로 구간 f 의 예상 통행 시간은 2 분 44 초로 추정한다. 이 두 값을 합하면 노드 1 에서 노드 3 까지의 예상 통행 시간(*T*)은 6 분 34 초를 추정할 수 있는데 이 값은 교통 혼잡, 날씨 등의 요인에 의해 노드 1 에서 노드 2 를 통과하는 동안에 34 초만큼 통행 시간이 지연되는 것을 의미한다.

4. 통행 시간을 고려한 최단 경로 탐색 연산자 설계

4.1 최단 통행 시간을 고려한 경로 탐색



(그림 3) 최단 경로 탐색 트리

(그림 3)과 (그림 1)의 그래프에서 Dijkstra 알고리즘과 도로 구간의 시간 비용 생성 방법을 이용하여 노드 1 에서 노드 8 까지 경로를 탐색한다. 그리고,

현재 차량은 노드 1 에 위치한다. 목적지 노드 8 까지 최단 시간 경로로 이동하려면 차량이 지나는 구간의 거리와 속도로 그 구간을 통과하는 소요 시간을 구할 수 있다고 가정한다. 최단 경로 탐색 알고리즘은 Dijkstra 알고리즘을 사용한다. (그림 1)에 있는 a 부터 j 까지의 값은 현재 통행 소요 시간이다. 노드 1 에서 다음 노드를 탐색하면 노드 2 와 노드 6 가 탐색된다. 이때, 두 경로 중 더 작은 비용의 값을 선택하므로 6 번 노드를 선택한다. 노드 1 에서 노드 6 까지의 예상 통행 시간(*T*)은 3:20 이다. 노드 6 에서 다음 노드를 탐색하면 노드 7 만이 탐색된다. 노드 7 까지의 예상 통행 시간은 10:33 이다. 3 장에서의 도로 구간의 시간 비용 생성 방법을 통해 다른 경로의 값도 추정 할 수 있고 Dijkstra 알고리즘에 적용하여 경로를 탐색하면 (그림 3)과 같다. (그림 3)은 Dijkstra 알고리즘과 도로 구간의 비용 생성 방법을 적용하여 노드 탐색 과정을 보여준다. 이것은 구간의 시간 비용을 계산하고 노드를 선택하는 과정을 반복하여 경로를 탐색하는 것을 트리로 표현했다. (그림 3)을 보면 노드 1 에서 목적지 노드 8 까지 가는 경로는 7 가지인 것을 알 수 있다. 그 중에서 예상 소요 시간(*T*)가 가장 작은 12 분 43 초가 소요되는 경로가 최소 운행 시간을 가지는 경로가 된다. 계산된 최적의 경

로 순서는 노드 1 → 노드 6 → 노드 7 → 노드 8 이다.

4.2 최단 시간 경로 탐색 연산자 알고리즘

```
Procedure init(Graph g, Node s)
 begin
  for each vertex v in vertices(g) do
   begin
    g.t[v]=infinity; /*모든 노드의 시간비용 초기화*/
    g.S[v]=unreached; /*모든 노드의 상태 초기화*/
    g.pi[v]=nil; /* 부모 노드 초기화*/
   end
  g.t[s]=0; /*노드 s 의 시간비용 초기화*/
  g.S(s)= permanently labeled; /*s 의 시간비용 고정*/
 end
```

(그림 4) 그래프 초기화 함수

(그림 4)는 그래프에서 노드들을 초기화하는 함수이다. 그래프 g 에 있는 모든 노드들을 무한대로 표시하고 노드 상태는 노드들을 한 번도 지난 적이 없으므로 unreached 이며, 부모 노드는 없다. 이 함수는 최단 경로 탐색 함수에서 사용된다.

```
Procedure search(Node u, Node v, double w[][])
  begin
    if t[v] > t[u]+w[u,v] then /*기존의 시간비용이 크면*/
    begin
      t[v]=t[u]+w[u,v]; /* 작은 시간비용을 대신 저장*/
      t[v]=t[v]+dif[v]; /*현재비용에 변화된 시간비용 합*/
      S[v]=temporarily labeled; /*임시 저장*/
      pi[v]=u; /* 부모 노드 표시*/
    end
    S[u]=permanently labeled; /*이전 노드는 시간비용 고정*/
 end
```

(그림 5) 노드 탐색 함수

(그림 5)에서 노드들 간의 비용을 계산하는 함수를 보인다. 다음 노드인 v 로 이동할 때 기존의 시간 비용이 새로운 경로로 이동한 시간 비용보다 크면 기존의 비용 대신 새로운 비용을 저장한다. 또한, 3 장에서 제안한 방법처럼 저장된 비용에 과거 시간 변화 량만큼 더한다. 노드 상태는 탐색된 노드들에 한해서 temporarily labeled 되고 부모 노드는 현재 노드인 u 가 된다. 부모 노드인 u 의 상태는 최단 경로를 구성하는 노드들을 permanently labeled 한다. dif[v]는 3 장에서의 도로 구간 비용 생성 방법에 의해 계산된 시간 비용 변화 량이다.

(그림 6)은 구간의 시간 비용 생성 방법과 Dijkstra 알고리즘을 이용하여 구간의 통행 시간 예상하여 목적지까지 최단 시간에 도착할 수 있는 경로를 제공하는 연산자 알고리즘이다. 이 알고리즘은 (그림 4)와 (그림 5)를 이용하여 초기화하고 노드를 탐색한다.

```
Procedure shortest_path(Graph g, Node s)
 begin
  init(Graph g, Node s) /* 초기화 */
  P={0}; /* P 초기화 */
  Q=Vertices(g); /*Q 초기화*/
  while not empty(Q) do
   begin
    u=StartNode(Q); /*출발지을 u 로 지정*/
    AddNode(P,u); /*P 에 출발지 삽입*/
    for each vertex v in Adjacent(u) do /*다음 노드들 탐색*/
    begin
     search(Node u, Node v, double w[][]) /*다음 노드 비용 계산 후 선택*/
    end
   end
 end
```

(그림 6) 최단 경로 탐색 함수

그래프 g 의 모든 노드들은 Q 에 저장된다. 저장된 노드들 중 시간 비용이 가장 낮은 노드를 선택하여 P 에 추가하고 노드 u 와 이웃한 노드들을 탐색하고 시간 비용을 계산한다. 이와 같은 연산을 Q 에 저장된 노드가 없을 때까지 반복 수행한다.

5. 결론

이 논문에서는 도로 네트워크 환경에서 운행 시간을 고려한 최단 경로를 탐색하기 위해서 현재 구간의 시간 비용과 과거 정보로부터 같은 구간의 시간 변화량을 더하여 구간의 시간 비용을 추정하였다. 그 값을 바탕으로 Dijkstra 알고리즘에 적용하여 통행 시간을 고려한 최단 경로와 구간의 소요 시간과 도착 예상 시간까지 추정하는 연산자를 제안하였다. 향후 연구로는 제안한 연산자 구현 및 확장, 그리고 차량 관리 시스템[5]에서의 최단 경로 탐색 연산자를 적용하고 평가할 예정이다.

참고문헌

[1] R. H. Güting, V. T. Almeida, and Z. Ding, “Modeling and Querying Moving Objects in Network”, Informat-ik-Report, LGDatenbanksysteme fürneue Anwendun-gen, FernUniversität Hagen, D-58084 Hagen, Germa-ny, 2004.

[2] H. D. Chon, D. Agrawal, and A. El Abbadi, “FATES: Finding A Time dEpendent Shortest path”, In Proc. of the 4th Intl. Conf. on Mobile Data Manage-ment(MDM), pp. 165-180, 2003.

[3] F. B. Zhan, “Three Fastest Shortest Path Algorithms on Real Road Networks: Data Structures and Procedures”, Journal of Geographic Information and Decision Analysis(GIDA), JGIDA vol.1, no.1, pp. 69-82, 1997.

[4] A. V. Goldberg, C. Harrelson, “Computing the Shortest Path: A* Search Meets Graph Theory”, In Proc. 16th ACM-SIAM Symposium on Discrete Algorithms, pp. 156-165, 2005.

[5] Y. J. Jung, K. H. Ryu, “The Vehicle Tracking System for Analyzing Transportation Vehicle Information Effectively,” ICSE2006, will be published, January, 2005.