

윈도우 매커니즘을 이용한 실시간 최적경로 추출 알고리즘 제안

이우용, 하동문, 신준호, 김용득
아주대학교 전자공학과 컴퓨터 네트워크 연구실
wylee@comnet.ajou.ac.kr, yongdkim@comnet.ajou.ac.kr

A Proposal of the Real time Optimal Route Algorithm With Window mechanism

Woo Yong Lee, Dong Mun Ha, Shin Jun Ho, Yong Deak Kim

ABSTRACT

This paper deals with a real time optimization algorithm within real time for DRGS(Dynamic Route Guidance System) and evaluate the algorithm. A pre-developed system offers the optimal route in using only static traffic information. In using real-time traffic information, Dynamic route guidance algorithm is needed. The serious problem in implementing it is processing time increase as nodes increase and then the real time processing is impossible. Thus, in this paper we propose the optimal route algorithm with window mechanism for the real-time processing and then evaluate the algorithms.

1. 서론

산업화가 급속도로 진행되면서 교통 혼잡에 의해서 소비되어지는 비용이 날로 증가되어 가고 있다. 이에 한정된 교통 자원의 활용을 극대화시키기 위해 교통량의 편중을 막아 교통 정보를 수집하고 제공함으로써 교통의 흐름을 효과적으로 분산시키는 방안에 대한 연구가 절실하다.

이렇듯 교통 편중을 방지하기 위한 교통 정보의 수집 방안으로서 RFID(Radio Frequency Identification)을 사용하여 링크의 여행 시간을 실시간으로 수집할 수 있으며, 이와 같이 수집 되어진 교통 정보를 실시간으로 제공하기 위한 방법으로서 라디오 주파수 대역에 디지털 정보를 전송하는 기법인 DARC(Data Radio channel)를 사용함으로써 동

적 교통 정보를 제공할 수 있다.

기존의 주행 안내 시스템은 정적인 교통 정보(각 지점 사이의 거리, 차선의 크기)만을 사용하여 주행안내를 수행하였으므로 알고리즘의 실시간 처리가 불필요하였으나, 동적인 교통정보를 사용하여 실시간으로 목적지까지의 최적 경로를 안내해 주기 위한 주행 안내 시스템의 개발에 있어 동적 최적 경로 추출 알고리즘의 개발은 필수적이라고 할 수 있다. 그러나 기존의 최적 경로 알고리즘은 네트워크상의 탐색 지점이 증가함에 따라 계산량이 엄청난 증가로 인해 실시간으로 처리 및 전송이 불가능하다.

따라서 본 논문에서는 교통 정보를 실시간으로 처리 하기 위해 윈도우 매커니즘을 바탕으로 하여 발견적 접근(Heuristic Approach)을 사용한 최적 경로 추출 알고리즘을 제안하였으며 실제 주행 안내 시스템의 개발을 통하여 알고리즘의 성능을 평가하였다.

2. 최적 경로 추출 알고리즘

기존에 존재하는 주행 안내 시스템에서 널리 알려진 최적 경로 추출 알고리즘은 폭우선 탐색 알고리즘을 바탕으로 한 Dijkstra 의 최단 경로 추출 알고리즘이다. 하지만, 이 알고리즘은 네트워크의 노드의 수가 증가함에 따라 처리 시간이 기하급수적으로 증가하게 되어 이 알고리즘만으로는 실시

간으로 수집된 정보 처리가 불가능하므로 보정 알고리즘이 필요하다.

2.1 Dijkstra 알고리즘

Dijkstra 알고리즘은 최단 경로의 계산에 있어서 매우 효과적인 알고리즘으로 알려져 있다. 그러나 네트워크 상의 노드가 증가함에 따라 발생하는 계산량의 증가 $O(n^2)$ 로 인하여 실제 도로망에 적용할 때 기하급수적으로 발생하는 계산량의 증가로 인하여 동적 주행 안내 시스템의 구현에 있어서 실시간 처리의 문제점을 가진다.

따라서, 실시간 처리를 위해서는 Dijkstra 알고리즘을 바탕으로 실시간 처리를 위한 새로운 알고리즘의 개발을 필요로 한다.

2.2 교통망에 적용을 위한 알고리즘(ORCA-1)

주행 안내 시스템에 적용을 위한 최적 경로 추출 알고리즘에 적용에 있어서 앞서 설명한 Dijkstra 알고리즘만으로서 구현하기는 힘들다. 즉, 그래프로 표현되어진 네트워크 노드에 존재하지 않는 위치가 존재하기 때문이다. 따라서 주행 안내 시스템에서 현재의 위치로부터 목적지점까지의 주행 안내를 하고자 할 때, GPS 수신기를 통하여 얻어진 현재의 위치를 도로망의 그래프상의 적절한 노드에 매핑(Mapping)시키는 알고리즘을 필요로 한다.

Optimal Route Computation Algorithm

1. Initialization

- N list of intersections in the transit network
- E list of streets in the transit network
- L_{ij} adjacency list (weight from i to j)
- $D[i]$ contains the length of the current shortest special route to node i
- R_{ij} list of shortest - route segments
- S list of processed nodes

2. If A_S and A_D are not intersection nodes

Find the nearest intersection node and assign to N_S and N_D

Else

$N_S \leftarrow A_S$ and $N_D \leftarrow A_D$

3. Search for a node x in $N - S$ that ensures $D[x]$ is minimum

$S \leftarrow S \cup \{x\}$

5. For each node y in $N - S$, perform

$D[y] \leftarrow \text{Min}\{ D[y], D[x] + L_{xy} \}$

$R_{ij} \leftarrow y$

6. Repeat stps 3 through 5 until V_D is processed

대부분의 알고리즘의 흐름은 Dijkstra 의 알고리즘과 같다. 즉 1 번째 과정은 그래프에 의해서 표현되어진 요소들의 초기화 부분이며, 2 번째 과정에서 네트워크상에 존재하지 않는 지점을 네트워크상의 인접한 노드로 할당하는 부분이 추가 되어져 있다. 실질적인 알고리즘 수행 부분은 2 번째에서 5 번째 과정까지이고 2 번째 처리부분을 제외하고 나머지는 Dijkstra 와 동일하다. 원래의 Dijkstra 알고리즘이 네트워크상에 시작 노드로부터 다른 모든 노드까지의 최단 경로를 추출 한다. 이때, 알고리즘의 종료 시점은 $(n-1)$ 개의 노드가 모두 집합 S 의 원소가 되었을 때 종료하나, 목적지가 결정되어져 있을 경우에 위의 과정에서는 V_D 가 집합 S 의 원소가 되었을 때 처리 루틴을 종료하게 된다.

2.3 윈도우 매커니즘을 이용한 발견적 접근 알고리즘

본 논문에서 사용할 발견적 접근에 의한 최적 경로 추출 알고리즘은 실시간 처리를 가능케 하기 위한 어느 정도의 최적 경로를 추출해낼 수 있으면서, 네트워크의 크기에 무관하게 결론을 도출할 수 있는 알고리즘이다. 이를 위해서 발견적 접근 기법을 사용한 것이다.

1. Get N_S and N_D
2. Compute the Euclidean distance between the two points(d)
3. Create a window
- 4 Extract the subnetwork nodes within the window
5. Create and adjacency list of the subnetwork
6. Call ORCA -1

ORCA-W 알고리즘의 처리 과정 중 2 번째 과정에서 N_S 와 N_D 로 부터 두 지점 사이의 거리 d 를 계산하고, 계산된 d 에 따라 그림 4 와 같이 윈도우를 설정한다. 설정되어진 윈도우는 그림 5 와 같이 전체 네트워크 중에서 계산 할 노드의 리스트를 추출하게 된다. 즉, 그림 5 에서 전체 네트워크는 a 부터 n 까지의 노드가 존재하게 되나, 윈도우에 의해서 추출된 노드는 b, c, h, i, j, k, l 뿐이다.

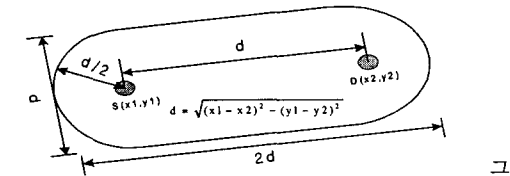


그림 1. 윈도우의 예 (width=0.5d, d=시작지점과 목적 지점사이의 거리)

위와 같은 알고리즘을 사용함으로써 실제 계산에 사용되어지는 네트워크의 크기는 실제 존재하는 네트워크의 크기와 무관하게 된다.

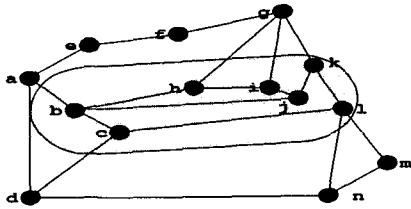


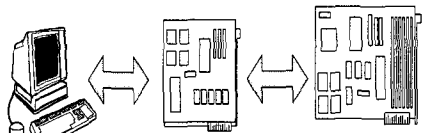
그림 2. 윈도우 매커니즘에 의한 서브네트워크

따라서, 추출 되어진 서브네트워크만을 가지고 ORCA-1 알고리즘에 의해서 최적 경로를 추출할 수 있게 된다. 하지만 직관적으로도 알 수 있듯이 윈도우의 크기와 네트워크의 크기에 따라 알고리즘의 효율성이 결정되어진다.

3. 시스템의 구현

본 논문에서 알고리즘의 성능을 평가하기 위해 구현한 주행 안내시스템은 크게 3부분으로 나뉜다. 처리 결과의 확인과 무작위의 교통 정보의 발생을 위한 PC 환경의 프로그램과 알고리즘 처리부와 통신을 위한 ISA Interface 부 그리고, 적용 알고리즘의 처리부로 구성 되어진다.

3.1 전체 시스템의 구성



ISA Interface부 알고리즘 처리부

그림 3. 전체 시스템의 구성

3.2 알고리즘 처리부

알고리즘 처리부는 DSP, Memory, 통신 모듈로 구성되어진다. 메모리는 전자 지도 정보와 프로그램의 다운 로딩을 위해서 사용되어진다. 주행 안내 시스템에서 전자 지도 정보는 중요한 역할을 수행하나 본 논문에서 이에 대한 언급은 하지 않는 것으로 한다. 그리고, 통신 모듈은 ISA 인터페이스 보드의 통신 모듈과 통신을 수행하며 같은 모듈을 사용하였다.

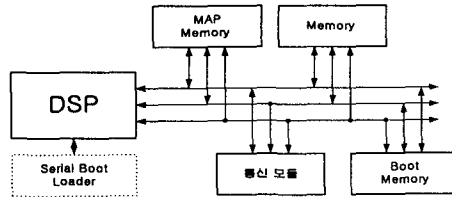


그림 4. 알고리즘 처리부

3.3 ISA 인터페이스부

ISA 인터페이스부는 통신 모듈과 EPLD 부로 구성되어 되어진다. EPLD는 통신 모듈의 메모리 매핑을 위하여 사용되어진다. 그리고, 통신 모듈은 알고리즘 처리부와 PC 쪽 프로그램과의 통신을 위하여 사용 되어진다.

4. 시뮬레이션 결과

본 논문에서 알고리즘의 성능을 평가 하기 위해서 사용한 교통망 네트워크의 크기는 노드의 개수가 70개인 네트워크 데이터를 가지고 시뮬레이션을 수행하였다.

알고리즘의 수행도를 평가하기 위해서 사용한 것은 시간[sec]이다. 알고리즘 처리부에서는 교통정보의 제공 후 처리 결과를 수신 ISA 인터페이스부를 통해 PC 상의 프로그램에서 확인하였다. 표 1의 알고리즘 처리 결과에서 (a) 와 (c) 는 ORCA-W 알고리즘을 사용한 결과이며 (b) 와 (d) 는 ORCA-I 를 사용하여 얻은 결과이다.

Destination : 12	Destination : 12
Cost : 15.00000	Cost : 9.000000
Path : 12 4	Path : 12 4
Path : 4 3	Path : 4 5
Path : 3 2	Start Node 1
Start Node 1	Start Time
Start Time	: 1.107000
Processing Time	Processing Time
: 0.989000	: 2.001000
: 0.453000	
(a)	(b)

Destination : 27	Destination : 27
Cost : 27.20000	Cost : 21.00000
Path : 27 26	Path : 27 17
Path : 26 25	Path : 17 13
Path : 25 23	Path : 13 12
Path : 23 12	Path : 12 14
Path : 12 4	Start Node : 1
Path : 4 3	Start Time
Path : 3 2	: 1.231000
Start Node 1	Processing Time
Start Time	: 7.638000
Processing Time	
: 1.207000	
: 1.616000	
(c)	(d)

표 1. 알고리즘 처리 결과

표 1의 결과로부터 (b)보다 (a)가, (d)보다 (c)가 처리시간이 짧음을 확인 할 수 있다. 하지만, 그림 9에서 볼 수 있듯이 윈도우에 의한 서브 네트워크의 크기가 증가 하게 되면 계산량이 기하급수적으로 증가함을 확인 할 수 있다. 즉, 비례 상수 $P = N / M$ (N :전체네트워크의 크기, M :서브네트워크의 크기)으로 정의 할 때, 그림 5의 그래프에서 알 수 있듯이 P 의 값이 증가 할수록 ORCA-1보다 ORCA-W 장점이 점점 증가하며, P 의 값이 감소 할수록 장점은 점점 감소한다.

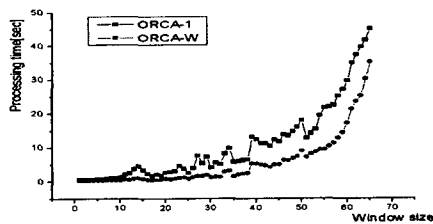


그림 5. 알고리즘의 처리 시간

즉, 본 논문에서 제안한 알고리즘은 비례 상수 P 에 따라서 성능이 결정된다. 비례 상수 P 에 따른 처리 효율 $Pe = Pt_{ORCA-1} / Pt_{ORCA-W}$ 는 그림 6에서 보여 주고 있다. 제안한 알고리즘을 사용했을 때의 효율은 비례상수가 0.2와 0.6 사이에서 최대의 효율을 나타내고 있다.

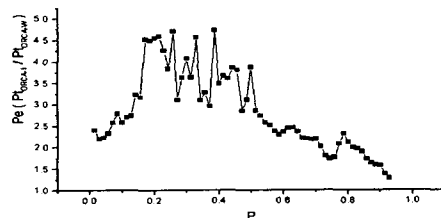


그림 6. 알고리즘의 처리 효율

그림 6에서 P 의 값이 1로 가까워짐에 따라 처리 효율도 1로 접근하고 있다. 시뮬레이션 결과 처리 효율은 평균 약 3 배의 속도의 향상을 가져온 것으로 나타났다.

즉, 전체 네트워크의 크기에 비해 현재의 지점과 목적 지점의 거리가 가깝다면 위의 결과에서와 같이 우수한 처리 속도를 가진다.

7. 결론

본 논문에서 실시간의 교통 정보를 수신하여 운전자에게 최적의 경로를 제공하는 시스템을 구현하

기 위한 최적 경로 추출 알고리즘을 제안하고, 알고리즘의 성능 평가를 위하여 TMS320c31(50Mhz) DSP를 사용하여 시스템을 구현하였다. 최적 경로를 추출하기 위하여 사용된 알고리즘은 Dijkstra의 최단 경로 추출 알고리즘을 바탕으로 하여, 시작지점과 목적지점을 축으로 하는 윈도우 매커니즘을 사용하여 전체의 네트워크를 처리할 때의 계산량의 증가를 줄이기 위한 알고리즘이다. 시뮬레이션의 결과를 통하여 윈도우의 크기가 크게 되면 기존의 Dijkstra 알고리즘을 사용하는 것과 같은 문제점을 확인할 수 있지만, 전체 네트워크의 크기에 비해 윈도우의 크기가 작아 질수록 알고리즘의 처리 속도 측면에서 우수함을 확인할 수 있었다.

실제 주행 안내 시스템의 구현에 있어서 알고리즘의 성능은 전체 시스템의 성능을 좌우한다. 여기서 알고리즘의 성능이라는 것은 최소한의 비용을 가지고 최대의 효과를 가져오는 알고리즘 이어야 하며, 이를 위해서 윈도우 매커니즘을 이용한 발견적 접근 방법에 의한 알고리즘(Heuristic Algorithm)을 사용하는 것이 주행 안내 시스템의 적용 알고리즘으로서 적합하리라 본다.

참고 문헌

- [1] 김용득, 서재홍, “무선인식기법을 사용한 구간별 교통 정보 수집 장치 구현”, 대한 전자공학회 추계 종합학술대회 논문집 제 20 권 제 2 호, pp.699~702, 97.11
- [2] Thomas h.Cormen, “Introduction to Algorithms”, McGraw-Hill, 1996, pp469~547
- [3] Dijkstra, E. W. “ A Note on Two Problems in Connexion with Graphs”, Numerische Mathematik 1, 1959, pp.269~271
- [4] James L. Bander , “ A New Route Optimization Algorithm for Rapid Decision Support”, Pennsylvania Society of automotive engineer, 1991, pp.709~728
- [5] Hassan A.Karimi, “Real-Time Optimal-route Comutaiton:A Heuristic approach”, ITS journal 1996 vol 3(2), pp.111~127
- [6] Ellis horowitz, “ Fundamentals of Data Structures in C”, Computer Science Press,1993 , pp.257~318
- [7] Nilsson, N.J. , “Principles Artificial Intelligence’Palo Alto,CA, Tioga Publishing Co., 1980, pp. 310~332
- [8] Karimi, H.A. and Krakiwsky, E.J.,”Design concept for Knowledge-Based route guidance system”,IEEE Position Location and Navigation Symposium , Florida, U.S.A, 1988, pp.95~103