

추상 그래프를 활용한 경로 탐색 알고리즘의 구현 및 성능 평가

김지수, 이지완, 조대수
동서대학교

Implementation and Evaluation of Path-Finding Algorithm using Abstract Graphs

Ji-Soo Kim, Ji-wan Lee, Dae-Soo Cho
Dongseo University

E-mail : kimjisu29@gmail.com, wldhks85@dreamwiz.com, dscho@dongseo.ac.kr

요 약

최근 단말기 기반의 경로 탐색에서도 동적인 정보를 반영하기 위한 연구가 진행되고 있다. 그러나 제시하는 대부분의 알고리즘은 A*알고리즘을 기반으로 한다. 휴리스틱을 이용한 알고리즘에서는 다음과 같은 탐색 비용이 증가하는 문제가 발생할 수 있다. 휴리스틱에 의해 결정된 추정 경로에 실제 경로가 존재하지 않을 경우, 휴리스틱 가중치 값이 비슷한 2가지 이상의 경로가 존재할 경우 탐색 비용이 증가한다.

이 논문에서는 생성 방법이 다른 추상 그래프의 성능을 평가 하였다. 추상 그래프는 실제 도로 네트워크를 단순화한 그래프로, 휴리스틱의 의존성과 탐색 비용을 줄이기 위해 제안된 방법이다. 추상 그래프는 생성 방법에 따라 동일 특성 노드 합병을 통한 추상 그래프(AG^H)와 연결 노드 합병을 통한 추상 그래프(AG^C)로 구별된다. 성능 실험 결과 생성 비용 측면에서 AG^C 가 좋은 성능을 보였지만, 탐색 성능 측면에서는 AG^H 가 좋은 성능을 보였다.

ABSTRACT

Recently, Many studies have been progressing to path-finding with dynamic information on the Terminal Based Navigation System(TBNS). However, the most of existing algorithms are based on A* algorithm. Path-finding algorithms which use heuristic function may occur a problem of the increase of exploring cost in case of that there is no way determined by heuristic function or there are 2 way more which have almost same cost.

In this paper, two abstract graph(AG) that are different method of construction, Homogeneous Node merging(AG^H) and Connected Node Merging(AG^C), are implemented. The abstract graph is a simple graph of real road network. The method of using the abstract graph is proposed for reducing dependency of heuristic and exploring cost. In result of evaluation of performance, AG^C has better performance than AG^H at construction cost but AG^C has worse performance than AG^H exploring cost.

키워드

Dynamic Heuristic, Pruning Search Space, A* Algorithm, Abstract Graph

중요 색인어

동적 휴리스틱, 탐색 영역 가지치기, A* 알고리즘, 추상 그래프

1. 서 론

일반적으로 단말기 기반의 시스템에서는 하드웨어 성능을 고려하여 A* 또는 변형 A* 알고리즘 [1,2,3,4]을 사용한다. 최근 TPEG[5,6]와 같은 방송 채널을 통해 도로의 차량 속도, 실시간 유고 정보

등의 동적인 정보를 단말기 기반의 시스템에서도 활용할 수 있게 되었다.

현재까지 도로의 상황을 고려한 경로 탐색 알고리즘으로는 [2,3,4]가 있다. 그러나 제시하는 대부분의 알고리즘은 A*알고리즘을 변형한 형태이다. 일반적으로 휴리스틱을 이용한 알고리즘에서는 다음

과 같은 탐색 비용이 증가하는 문제가 발생할 수 있다. 휴리스틱에 의해 결정된 추정 경로에 실제 경로가 존재하지 않을 경우, 휴리스틱 가중치 값이 비슷한 2가지 이상의 경로가 존재할 경우 탐색 비용이 증가한다.

휴리스틱을 이용한 경로 탐색에서의 문제점을 해결하기 위해 추상 그래프를 이용한 방법을 제안하였다[7,8]. 추상 그래프는 실제 도로 네트워크를 간략한 그래프로 표현하고, 경로 탐색에 활용하여 휴리스틱 의존성과 탐색 비용을 줄이는 것이다.

이 논문에서는 생성 방법이 다른 두 추상 그래프에 이용하여 효율성에 대한 성능 평가를 수행하였다. 추상 그래프를 활용한 경로 탐색 방법은 동일하나 추상 그래프를 생성하는 방법론에 차이가 있다. 생성 방법에 따라 동일 특성 노드 병합을 통한 추상 그래프(AG_H)[7]와 연결 노드 병합을 통한 추상 그래프(AG^C)[8]로 분류되어진다.

II. 관련 연구

A* 알고리즘[1]은 가장 범용적으로 사용되고 있으며, 특히 AI 분야에서 많이 쓰이는 방법이다. A* 알고리즘은 노드 평가 함수 $F=G+H$ 를 사용하여 다음 탐색될 노드를 선택한다.

가변적인 그리드 속력을 이용한 경로탐색 기법[2]은 실시간 교통정보를 활용할 경우를 대비한 경로 탐색 기법이다. 이 경로탐색 기법은 경로 탐색시 우선권을 가지는 노드를 탐색할 때 고정된 데이터인 거리만으로 평가하지 않고 가변적인 데이터인 각 도로의 속력정보를 고려한 탐색을 한다. 즉, 시간 $(T)=거리(S)/속력(V)$ 를 이용하여 이동 시간(T)을 노드 평가 시 가중치로 사용하였다.

최저속력 동적 휴리스틱[3]은 그리드 셀의 최저속력을 동적 휴리스틱으로 사용한다. 경로 탐색시 최저속력이 낮은 구역을 탐색범위에서 제외시킴으로 속력이 낮은 구간을 탐색할 경우의 수를 줄임으로서 경로의 질을 크게 향상 시켰다. 그러나 경로 탐색에서 제외된 그리드내에 주요 교차로나 강 또는 하천 등을 가로지르는 다리가 있는 경우 크게 돌아가는 경로를 탐색해야 하므로 탐색비용이 증가하는 단점이 있다.

최고 속력 동적 휴리스틱[4]을 이용한 경로 탐색 알고리즘은 A*알고리즘을 기반으로 하지만, 열린 목록에서 노드 추출 방법을 2단계 노드 추출로 변형하였다. 1차 노드 추출은 고정된 데이터인 거리 정보를 이용하여 우선순위가 정해진 노드 목록을 생성하고, 생성된 노드목록에서 우선순위가 높은 N개의 노드들을 대상으로 해당 노드를 포함하고 있는 그리드의 최고 속력 정보를 이용하여 최우선 순위 노드를 추출하게 된다. 동적 휴리스틱을 이용함으로써, 경로의 질은 높일 수 있었으나, 2단계 노드 추출을 이용하기 때문에 탐색 비용이 증가하는 단점이 있다.

III. 추상 그래프를 이용한 경로 탐색

3.1 추상 그래프 생성[7,8]

추상 그래프는 그리드 셀과 도로 정보를 기반으로 생성된다.

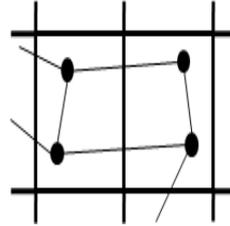


그림 1. 예제 그래프
Fig 1. Example Graph

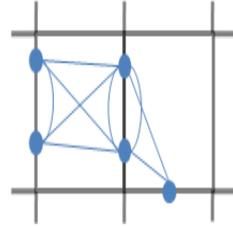


그림 2. 경계 노드와 간선
Fig 2. Boundary Node and Edge

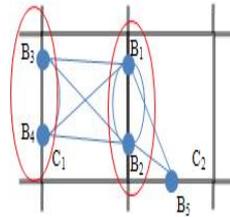


그림 3. 동일 노드 예제
Fig 3. Example of Homogeneous Nodes

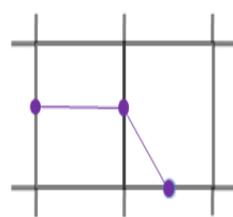


그림 4. 추상 그래프
Fig 4. Abstract Graph

동일 특성 노드 병합을 통한 추상 그래프 생성 (AG^H)[7]은 경계 노드 생성, 경계 간선 생성, 경계 노드 병합을 통한 추상 노드 생성 그리고 추상 노드 간의 연결을 통한 추상 간선 생성, 총 4단계로 생성된다.

경계 노드는 추상 노드를 생성하기 위한 기반이며, 경계 노드들 간의 병합을 통해 추상 노드가 생성 된다. 경계 노드와 경계 간선 생성은 다음과 같이 이루어진다. 첫째, 맵 전체를 고정된 크기의 그리드 셀로 구성된 하나의 그리드를 생성한다. 둘째, 각 셀 범위 내부에 존재하는 도로들 중에서 그리드 셀의 경계와 교차하는 위치에 경계 노드를 생성하게 된다. 셋째, 각 셀에 생성된 경계 노드들 간의 실제 경로 탐색을 통해 경계 간선을 생성한다. 그림 2는 그림 1을 기반으로 생성된 경계 노드와 경계 간선을 나타낸다. 각 셀에 생성된 경계 노드 중 동일한 특성을 가진 경계 노드간의 병합이 이루어진다. 추상 노드는 최소 하나의 경계 노드 또는 둘 이상의 경계 노드 병합으로 생성된다. 그림 3은 동일 특성을 가진 노드들을 표현하고 있다. 추상 간선 생성은 추상 노드를 구성하는 경계 노드의 연결 정보를 이용하여 생성하게 된다. 그림 4는 그림 1의 예

제 그래프를 기반으로 생성된 추상 그래프를 보여 준다.

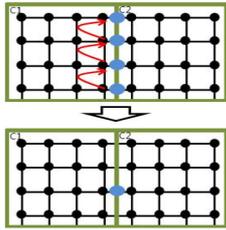


그림 5. 추상 노드 병합
Fig 5. Abstract Node Merging

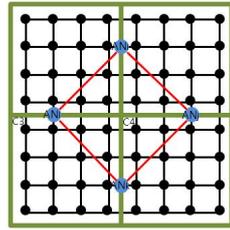


그림 6. 추상 노드 그래프
Fig 6. Abstract Graph

연결 노드 제거를 통한 추상 그래프 생성(AG^C)[8]은 추상 노드 생성, 추상 노드 병합, 추상 간선 생성, 총 3단계로 생성된다. 추상 그래프 생성은 다음과 같다. 첫째, 그리드를 생성하고 그리드 셀의 경계와 도로 간선이 교차하는 곳에 추상 노드를 생성한다. 둘째, 같은 그리드 셀 경계에 존재하는 추상 노드 간에 실제 경로 탐색을 통해 경로 존재 여부를 판단한다. 단, 탐색이 수행되는 범위는 전체 영역이 아닌 추상 노드가 생성된 그리드 셀 경계를 가진 셀들 범위 내에서 경로 존재 여부를 판단한다. 추상 노드의 수를 최소화하기 위해 경로가 존재하는 추상 노드 간에 병합이 이루어진다. 그림 5는 추상 노드 병합을 보여준다. 셋째, 추상 노드들에 대해 경로의 존재 여부를 판별하여 추상 간선을 생성하게 된다. 그림 6은 생성된 추상 그래프를 보여준다.

3.2 추상 그래프를 활용한 경로 탐색

추상 그래프를 활용한 탐색 방법은 추상 그래프 경로 탐색과 실제 경로 탐색 2단계로 이루어진다. 1단계 탐색은 실제 네트워크에 비해 노드의 수와 간선의 수가 적은 추상 그래프에서 Dijkstra알고리즘[9] 탐색을 수행하며, 2단계 탐색은 1단계 탐색의 결과를 기반으로 실제 네트워크에서 A*알고리즘 탐색으로 수행된다. 노드 평가 방법은 시간(T)=거리(S)/속력(V)를 이용하여 이동 시간(T)을 노드 가중치로 사용하며, 해당 노드를 포함하고 있는 그리드의 속력 정보를 적용한다. A* 알고리즘의 휴리스틱 설정은 [10]에서 제안한 고정 그리드 기반의 추정치 간소화 기법을 이용한다.

추상 그래프 탐색을 통해 경로의 존재여부, 탐색 범위를 추출 할 수 있다. 실제 경로 탐색은 추상 탐색의 결과인 추상 경로를 포함하는 유효 셀 범위 내에서 수행된다. 추상 경로를 포함하는 셀들을 유효 셀이라 한다.

IV. 성능 실험 및 비교 분석

생성 방법이 다른 두 추상 그래프에 대해 생성 비용 및 탐색 비용을 평가하기 위해 성능 실험을 하였다. 성능 비교를 위해 사용된 데이터는 106,254 개의 간선으로 이루어진 부산 시내 도로데이터를 사용하였다.

표 1. 추상 그래프 비교

Table 1. Comparison Abstract Graphs

	AG^H	AG^C
추상 노드 수	4,224	2,016
추상 간선 수	24,861	13,411
생성 시간	1시간 42분	2분 21초

추상 그래프는 셀의 크기에 따라 추상 노드와 추상 간선의 수가 다르며, 추상 그래프 생성 비용에 영향을 미친다. 표 1은 크기가 0.5km인 그리드 셀을 기반으로 생성된 AG^H 와 AG^C 를 비교해 보았다. 생성 비용 측면에서 AG^C 가 AG^H 에 비해 모든 면에서 좋은 성능을 보이고 있다. 생성 시간에서 많은 차이를 보이는 것은 AG^H 에서 경계 간선 생성 비용이 크기 때문이다.

AG^H 와 AG^C 를 활용하여 경로 탐색 성능을 실험해 보았다. 성능 실험은 동일한 출발지와 목적지를 사용하여 각 추상 그래프에서 탐색을 수행했으며, 출발지와 목적지까지의 직선거리별로 5가지로 구별하여 구간별로 각 200회 실험하였다. 또한 그리드 셀의 크기에 따라 탐색 비용에 영향을 미치기 때문에 0.5km와 1.0km의 셀에서 생성된 추상 그래프를 활용하여 실험해 보았다.

그림 7과 그림 8은 셀의 크기가 각 0.5km와 1.0km인 추상 그래프를 활용하여 성능 비교한 것이다. 노드 접근 횟수와 탐색시간, 경로의 길에 따른 성능을 $[AG^C]/[AG^H]$ 으로 표현한 그래프이다. 모든 탐색 성능 측면에서 AG^H 가 AG^C 에 비해 좋은 성능을 보여주고 있다. AG^H 의 경우 경계 간선 생성시 실제 탐색을 통해 경로의 길이를 가중치로 사용하며, 이를 기반으로 추상 간선의 가중치 값이 설정된다. 하지만 AG^C 의 경우 추상 간선의 가중치를 각 셀별로 동일한 값을 사용하기 때문에 AG^H 에 비해 가중치의 정확성이 떨어져 좋지 못한 탐색 성능을 보여주고 있다.

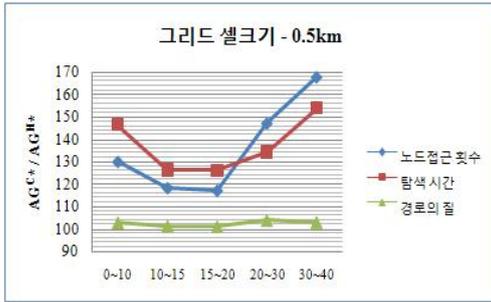


그림 7 0.5km 셀에서의 성능비교
Fig 7. Performance Evaluation(0.5km)

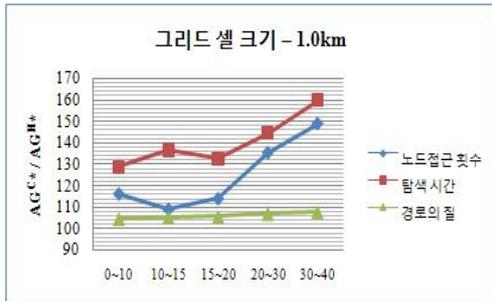


그림 8. 1.0km 셀에서의 성능비교
Fig 8. Performance Evaluation(1.0km)

V. 결론 및 향후 과제

이 논문에서는 생성 방법이 다른 두 추상 그래프를 이용하여 성능 실험을 해 보았다. 생성 비용 측면에서는 AG^C가 AG^H에 비해 좋은 성능을 보였지만, 탐색 비용 측면에서는 AG^H가 AG^C에 비해 좋은 성능을 보였다.

향후, 동적 정보를 이용한 A*알고리즘과 탐색 성능 평가를 수행하고, 최적의 추상 그래프를 생성하기 위한 연구가 진행되어야 할 것이다.

참고문헌

[1] Peter E. Hart, Nils J. Nilsson and Bertram Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths in Graphs," IEEE Trans. on Systems Science and Cybernetics, Vol. SSC-4, No. 2, pp 100-107, 1968
 [2] 이현섭, 김진덕, "고정 그리드 기반 가변 휴리스틱을 이용한 최적경로 탐색," 한국해양정보통신학회 2005 추계 종합학술대회, Vol.9, No.2, pp.137-141, 2005
 [3] 문대진, 조대수, "최저 속력 동적 휴리스틱을 이용한 경로 탐색," 한국공간정보시스템학회 2008 Vol.19, No2, 2008
 [4] 문대진, 조대수, "실시간 도로 정보를 이용한 최고속력 동적 휴리스틱의 설계," 한국해양정보통신학회 2008 추계 종합학술대회, Vol.12, No.1,

pp.827-830, 2008

[5] <http://en.wikipedia.org/wiki/TPEG>

[6] EBU B/TPEG, "Transport Protocol Experts Group (TPEG) TPEG specifications - Part 1: Introduction, Numbering and Versions", TPEG-INV/002, draft, October 2002.

[7] 김지수, 이지완, 조대수 "동적 교통 정보를 적용하기 위한 도로망 추상화 기법의 설계", 2009 춘계 종합학술대회, Vol.13, No.1, pp.199-202, 2009

[8] 이지완, 조대수 "동적정보를 이용하는 환경에서 탐색영역 최소화를 위한 이웃노드 병합 추상그래프 설계", 한국공간정보시스템학회 2009 춘계학술대회, pp.220-226, 2009

[9] Dijkstra, E. W., "A note on two problems in connection with graphs," Numerische Mathematic, Vol. 1, pp260-271, 1959

[10] 김진덕, "가변 휴리스틱 기반 추정치 간소화를 통한 경로 탐색 기법의 설계 및 성능평가", 한국해양정보통신학회논문지 Vol 10 , No 11, pp2002-2007, 2006