

위치정보를 이용한 최단경로탐색 알고리즘 개선

이 종 현* · 강 민 제** · 이 상 준***

An improvement of shortest path algorithm using the positioning information

Jong-Heon Lee*, Min-Je Kang** and Sang-Joon Lee***

ABSTRACT

The shortest path problem arises whenever we try to determine the shortest, cheapest, or most reliable path between one or many pairs of nodes in a network. The path search algorithm for the CNS(Car Navigation System) process the actual road map with complex data and should return the result within acceptable response time. In this paper, we studied the improvement of existing shortest path algorithm using the positioning information. The proposed algorithm was analyzed by the simulation with the real road map data in Jeju city. From the simulation result, it is known that the proposed algorithm required less memory usage and less computational time than the others.

Key words : Shortest path search, car navigation system, positioning information

1. 서론

차량경로계획 서비스는 차량항법시스템과 연계하여 운전자에게 목적지까지 빠르고 쾌적한 주행경로를 제공함으로써 도로망을 효율적으로 사용한다는 측면에서 매우 중요하다. 이런 차량경로계획을 수행하기 위한 알고리즘은 최단경로문제에 기초를 두고 있다. 경

로탐색은 운전자에게 현 위치에서 최종목적지까지 최단거리의 도로를 통하여 도달할 수 있는 경로를 알려주는 것이다.

대단히 규모가 크고 복잡한 조건을 갖는 도로망이 대상이므로 기존의 최단경로 알고리즘을 적용하는 것은 현실적으로 어렵다. 기존 최단경로 알고리즘을 실제 도로망에 적용할 때 많은 계산량과 메모리가 필요하기 때문에 빠른 응답시간을 요구하는 실제 시스템의 구현에는 적합하지 않다. 이러한 문제를 해결하기 위하여 보다 적은 계산시간과 적은 메모리를 이용하여 탐색을 할 수 있는 탐색알고리즘이 필요하다.

최단경로문제를 찾아내는 알고리즘은 Dijkstra 알고리즘^{1,2)}과, 휴리스틱함수를 이용한 A*알고리즘^{2,3,4)}이 대표적인 알고리즘이다. 또한 Dijkstra를 변형한

* 제주대학교 대학원

Graduated school, Cheju Nat'l Univ.

** 제주대학교 전기·전자공학부, 산업기술연구소

Faculty of Electrical & Electronic Eng., Res. Insti. Ind. Tech.,
Cheju Nat'l Univ.

*** 제주대학교 통신·컴퓨터공학부, 산업기술연구소

Faculty of Telecommunication & Computer Eng., Res. Insti. Ind. Tech.,
Cheju Nat'l Univ.

Bi-directional Dijkstra^{124,61}와 A*를 변형한 Bi-directional A*^{2,61} 등이 있다.

Dijkstra 알고리즘은 출발지에서 인접한 노드들까지의 비용을 계산하여 가장 적은 비용의 노드로 이동하며, 이러한 과정을 목적지에 도착할 때까지 반복 수행한다. Dijkstra 알고리즘은 확실히 최단경로를 탐색을 보장하지만 목적지의 방향에 관계없이 모든 방향의 노드를 검색하기 때문에 많은 탐색 소요시간과 메모리를 필요로 하는 단점을 가지고 있는데 이것은 시스템의 성능을 저하시키는 요인으로 작용하고 있다.

A*는 출발지 노드에서 현재노드의 인접한 노드들에 이르는 거리뿐만 아니라 현재 노드의 인접한 노드들에서 목적지까지의 거리도 고려하여 이동노드를 선택하는 알고리즘이다. 이 알고리즘은 탐색시간은 줄일 수 있으나 인접한 노드들에서 목적지까지의 거리 계산을 추가로 해야하는 부담을 안고 있다. 또한 선정된 경로가 출발지에서 목적지까지의 탐색경로가 최단경로라는 보장을 할 수 없다.

Bi-directional Dijkstra 알고리즘과 bi-directional Dijkstra A* 는 탐색을 출발지에서 목적지 방향으로 탐색하는 전방탐색과 목적지에서 출발지 방향으로 탐색하는 후방탐색을 병행하여 서로의 탐색이 중간에 교차될 때 탐색을 완료하게 된다. 이론적으로는 양방향 탐색은 단방향 탐색에 비해 전체 탐색노드 수의 절반을 줄일 수 있는 장점이 있다. 그러나 만일 양방향 탐색은 양단에서 탐색이 교차하지 않을 경우 오히려 단방향 탐색에 비해 최고 두 배의 노드를 탐색해야 하는 단점이 있다.

본 논문의 목적은 도로망탐색의 계산시간과 소요메모리를 감소시키며 타당한 주행경로를 제공하는 차량 경로계획 시스템의 구현이다. 기존의 도로망에는 위도와 경도에 대한 정보를 알 수 있기 때문에 출발지와 목적지간의 위치정보를 이용할 수 있는 여지가 있다. 따라서 위치정보를 이용한 경로탐색 알고리즘이 설계가 가능하다. 본 연구에서는 위치정보를 이용하여 계산시간과 소요메모리를 감소시킨 방향성을 갖는 알고리즘을 제안하였다. 제안된 경로탐색 알고리즘을 이용하여 구현한 차량경로계획 시스템을 제주시와 신제주의 도로망을 대상으로 모의실험을 수행하였다. 시물레이션은 실험 결과의 실용성을 위하여 실제 디지털 도

로 지도 데이터 베이스를 이용하여 진행하였다.

II. 최단경로탐색 알고리즘

이장에서는 기존의 대표적인 경로탐색 알고리즘들의 내용 및 특성들을 간략하게 기술하였다.

2.1. Dijkstra의 최단경로 알고리즘

Dijkstra 알고리즘은 이해가 쉽고 각종 최단경로 문제에 광범위하게 적용될 수 있으며, 사용이 간편함으로 가장 널리 사용되고 있는 알고리즘이다. Dijkstra 알고리즘은 우선 초기 출발노드를 제외한 모든 노드의 표지를 ∞ (무한대)로 설정한다. 그 다음 초기 출발노드에서 연결된 모든 노드의 표지에 각각의 거리를 입력한다. 이때 입력된 초기 출발노드로부터의 누적거리의 값이 가장 작은 노드를 그 다음 노드로 설정하고, 그 노드에 영구표지를 표시한다. 다시 다음 노드에서 연결된 모든 노드를 찾아내어, 각각의 노드표지에 초기 출발노드로부터의 거리의 누적값을 입력한다. 다시 그때까지 입력된 표지값 중 가장 작은 노드를 다음 노드로 하여 위의 과정을 모든 노드에 영구표지가 될 때까지 반복한다. 식 (1)과 같이 출발지 u 노드에서 인접한 모든 v 노드까지의 비용 g 를 두 노드간의 거리 $L(u, g)$ 로 계산하여 노드평가 함수 f 에 할당한 후 f 의 값이 가장 작은 노드로 이동하며 이를 목적지에 도착할 때까지 반복수행 한다.

$$f = g + L(u, v) \quad (1)$$

Dijkstra 알고리즘을 사용하면 노드의 검색 반경이 Fig. 1과 같이 출발지를 중심으로 동심원을 그리며 넓어지게 된다. Dijkstra 알고리즘은 확실하게 최단경로를 선정할 수 있는 장점이 있으나 목적지의 방향에 관계없이 모든 방향의 노드를 검색하게 되어 검색시간이 늦어지고 탐색공간이 많아지는 단점이 있다.

2.2. A* 알고리즘

A* 알고리즘은 목적지의 방향에 관계없이 모든 방

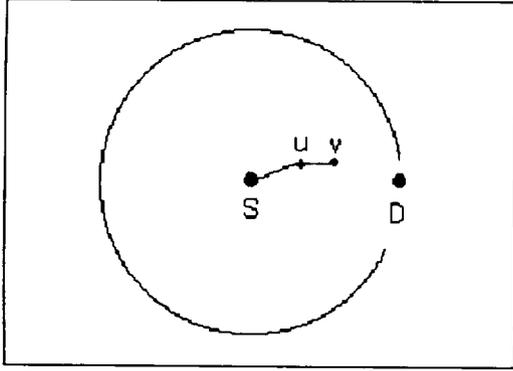


Fig. 1 Search area of Dijkstra's algorithm
(S: starting point. D: destination point
u: current node. v: adjacent node of u)

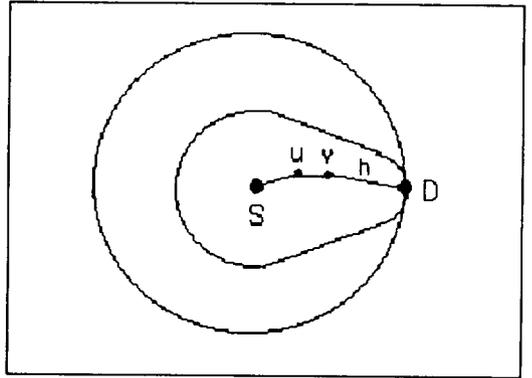


Fig. 2 Search area of A* algorithm
(S: starting point. D: destination point
u: current node. v: adjacent node of u)

향의 노드를 모두 검색하는 Dijkstra 알고리즘의 비효율성을 해결하기 위해 제시된 알고리즘들 중의 하나이다. A* 알고리즘에서는 탐색 노드의 선정에 있어서 Dijkstra 알고리즘에서 사용하는 출발지와 목적지와의 거리 g 이외에 식 (2)와 같이 목적지까지 직선 거리 h 를 추가하여 선정한다.

$$f = g + h \quad (2)$$

이로 인해 Fig. 2와 같이 목적지에서 멀어지는 노드에 대한 불필요한 계산을 생략하게 되어 전체적인 검색 속도를 증가시키고자 하는 것이다. 물론 A* 알고리즘은 Dijkstra 알고리즘에 비해 h 를 계산하기 위한 추가적인 계산의 부담을 가지고 있다. 또한 A* 알고리즘은 h 의 사용으로 인해 선정된 경로가 최적의 경로임을 보장할 수 없게 된다.

III. 위치정보를 이용한 최적 경로 탐색 알고리즘

최적 경로 탐색을 위한 디지털 지도에는 위도와 경도에 의한 위치정보가 수록되어 있다.

본 연구에서는 이에 착안하여 출발지로부터 목적지에 이르는 방향정보를 최적 경로 탐색 알고리즘의 설계에 이용하였다. 즉, 기존의 Dijkstra 알고리즘이 방

향에 상관없이 현재의 위치에서 인접한 모든 노드를 검사하는 하기 때문에 많은 계산량과 이에 따른 과도한 메모리의 소요 및 계산시간의 과다문제가 발생한다. 따라서 인접한 노드 중 목적지의 도달에 필요 없음을 노드에 대한 불필요한 탐색을 사전에 제거함으로써 계산시간과 메모리의 효율적 이용이 가능하게 된다.

본 연구에서 제안하는 알고리즘은 Fig. 3에서와 같이 출발지 u 에서 목적지 d 노드까지 인접한 모든 v 노드를 찾을 때 직선 u, d 와 u, v 의 방향각을 이용한 방향성을 이용한다. 이때 방향각을 설정하는 유클리드 함수를 이용한다. 이때 사용 거리에 따라 유클리드 함수를 조정함으로써 목적지의 방향과 일정 정도 이상의 일치하지 않는 노드는 탐색을 하지 않는다. 새로운 노드로의 확장을 위한 평가함수는 식 (3)과 같다.

$$f = g + h \quad (3)$$

h 값은 직선(u, d)와 직선(u, v)의 삼각함수 \cos 제곱을 이용하여 계산한다.

제안 알고리즘의 절차는 Fig. 3과 같다.

이로 인해 Fig. 4와 같이 목적지의 방향과는 일정 정도 이상 어긋나는 노드에 대한 불필요한 계산을 생략하게 되어 전체적인 검색 속도가 증가되리라고 예측할 수 있다. 식 (3)에서 최적의 성능을 보이는 h 의

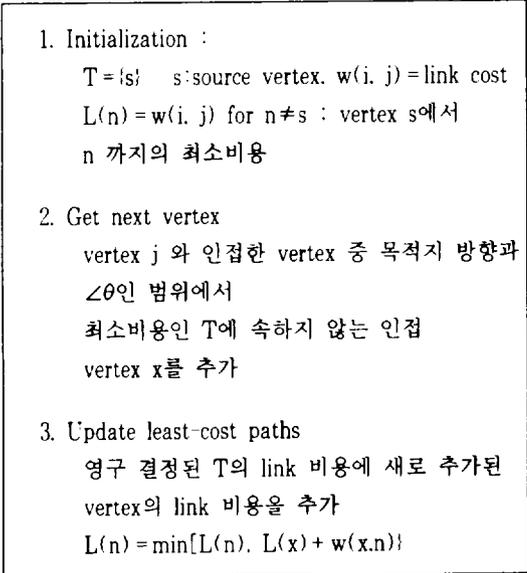


Fig. 3 Proposed Algorithm

값은 도로의 특성에 따라 가변적이며 거리가 멀어짐에 따라 각도도 적게 잡아도 됨을 알 수 있다. 본 논문에서는 시뮬레이션 결과 h의 값이 90도 일 경우에 도로의 특성에 관계없이 가장 안정적으로 경로 탐색을 진행함을 알고 h의 값으로 90도를 사용하였다.

Fig. 4는 본 연구에서 제안한 알고리즘의 줄어드는 검색 반경의 모습을 보여준다.

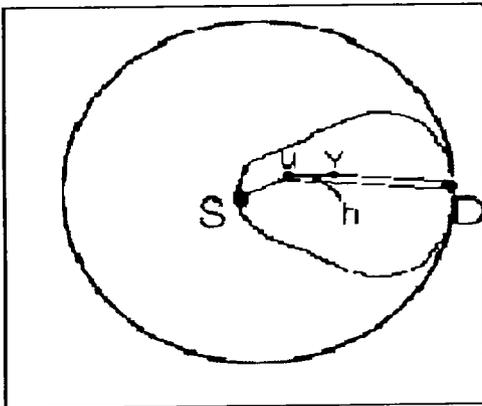


Fig. 4 Search area of proposed algorithm
(S: starting point. D: destination point
u: current node. v: adjacent node of u
h: line(u, v) and line(u, D) of angle)

IV. 시뮬레이션

본 장에서는 다양한 최단 경로 탐색 알고리즘들의 성능을 비교하기 위하여 실제의 디지털 도로 지도 데이터 베이스를 가지고 시뮬레이션한 결과와 그 결과를 바탕으로 분석한 내용을 정리한다.

4.1. 시스템 환경

구현한 시스템 환경은 다음과 같다.

- 하드웨어 : Pentium II
- 운영체제 : Windows98
- 소프트웨어 :

- 1) 개발언어 : Delphi 4.0
- 2) GIS 엔진 : Mapobject 2.0
- 3) DataBase : DbaseIII . MSACCESS
- 4) Data : 제주시와 신제주의 디지털 도로망도

4.2. 시뮬레이션 내용

차량 항법 시스템에서 경로 탐색 알고리즘의 성능을 평가하는 기준은 탐색된 최단 경로의 정확성과 탐색에 소요된 시간이다. 본 논문에서는 탐색 알고리즘들이 선정한 경로의 정확성을 측정하기 위하여 각 알고리즘을 이용하여 탐색된 최단 경로의 길이를 측정하였으며 탐색 속도를 측정하기 위해서 탐색 과정에서 방문했던 전체 노드 수와 각 경로 탐색 알고리즘 모듈의 호출에 소요된 시간을 측정하였다. 탐색에 소요된 시간의 정확성을 위하여 동일한 경로의 실험을 10회 반복 실시하여 소요 시간의 평균을 해당 경로의 탐색에 소요된 시간으로 설정하였다.

4.3. 시뮬레이션 결과 및 분석

Fig. 5는 실험에 사용된 맵(map)의 한 예와 각 알고리즘에 대한 최단 경로 탐색 결과이다. 맵은 차량 항법 시스템에서 제주시와 신제주 지역을 표현한 것인데 노드 수가 70개이며 링크 수는 110개를 포함하고 있다. 노드는 내부적으로 번호로 구분하였다. 또한 실험의 편의를 위하여 편도 2차선 이상의 도로만을

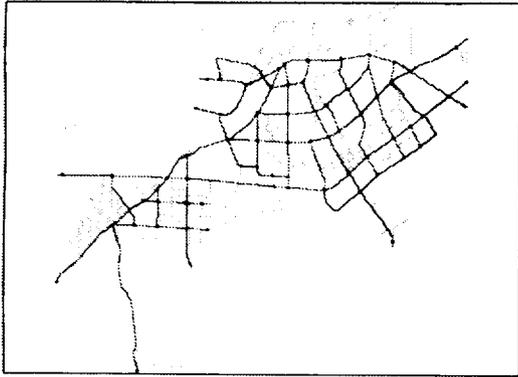


Fig. 5 Target digital map



Fig. 6 The results of shortest path search

대상으로 하였다.

Fig. 6과 Table 1은 각각 Dijkstra 알고리즘과 A* 알고리즘, 제안 알고리즘으로 찾아낸 최단거리 경로 및 그에 따른 각 알고리즘의 탐색 노드 수이다. 세 알고리즘 모두 동일한 경로를 최단거리로 선택하였다. Table 1에서 나타난 바와 같이 제안한 알고리즘의 탐색노드수가 가장 적은 것을 볼 수 있다.

4.3.1. 정확성

이 절에서는 각 경로 탐색 알고리즘들을 이용하여 선정된 탐색 경로의 정확성을 비교 분석한다. 이론적으로 Dijkstra 알고리즘이 최단의 경로를 선정하기 때문에 알고리즘들 간의 성능 차이를 알아보기 위하여 Dijkstra 알고리즘을 기준으로 다른 알고리즘들이

선정한 경로를 관찰하였다.

Fig. 7은 5회에 걸친 실험을 통해 각 알고리즘 별로 선정된 최단 경로의 비용의 합을 나타낸 그림이다. 실험 결과에 의하면 알고리즘모두 Dijkstra 알고리즘에 의해 선정된 최단 경로의 길이와 거의 비슷하다는 것을 알 수 있다.

4.3.2. 탐색 노드 수

일반적으로 경로 탐색 알고리즘의 특성을 파악하기 위해 가장 많이 사용하는 방법이 알고리즘의 수행 과정에서 최단 경로를 선정하기 전까지 방문하게 되는 전체 노드의 수를 관찰하는 것이다. 본 논문에서도 각 알고리즘 별 전체 방문 노드 수를 관찰해 보았는데 그 결과를 Fig. 8에 나타냈다. Fig. 8은 각 알고리

Table 1. The visited node order and count

Nodes Alg.	Visited order of nodes	Total nodes count(including duplicates)
Dijkstra	51 45 45 45 38 38 46 37 37 39 39 29 29 29 47 30 40 40 36 36 27 27 23 23 31 31 48 35 41 32 32 26 26 42 22 16 15 15 54 54 54 18 18 25 43 43 21 17 14 14 53 53 11 58 58 58 20 20 13 13 55 6 64 64 64 52 52 56 4 3 63 60 61 61 2	75
A*	51 45 45 45 38 38 46 37 37 39 39 29 29 29 47 30 40 40 36 36 27 27 23 23 31 31 35 32 32 26 26 22 16 15 15 54 54 25 21 14 14 58 20 20 13 13 6 4 3 2	50
Proposed	51 45 45 38 37 37 29 29 36 36 27 27 23 35 26 26 22 15 54 25 21 14 14 20 20 13 6 4 3 2	30

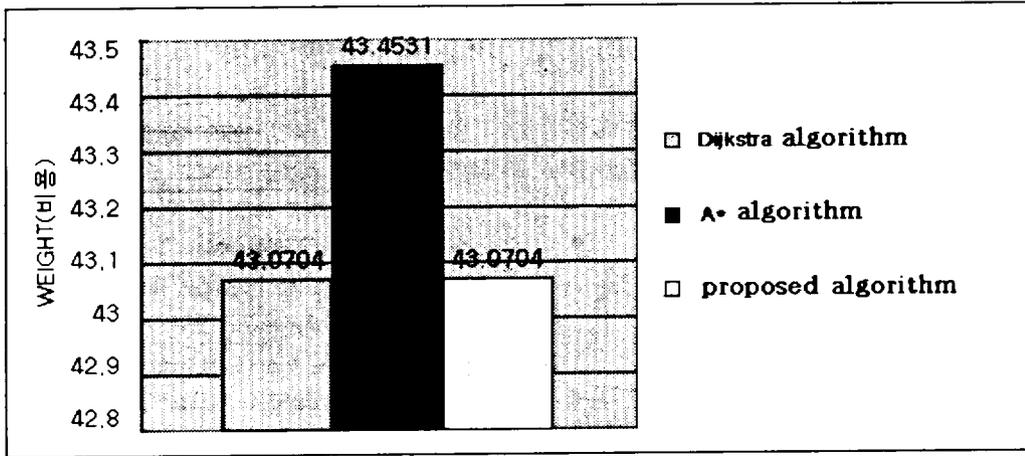


Fig. 7 Weights sum of results paths

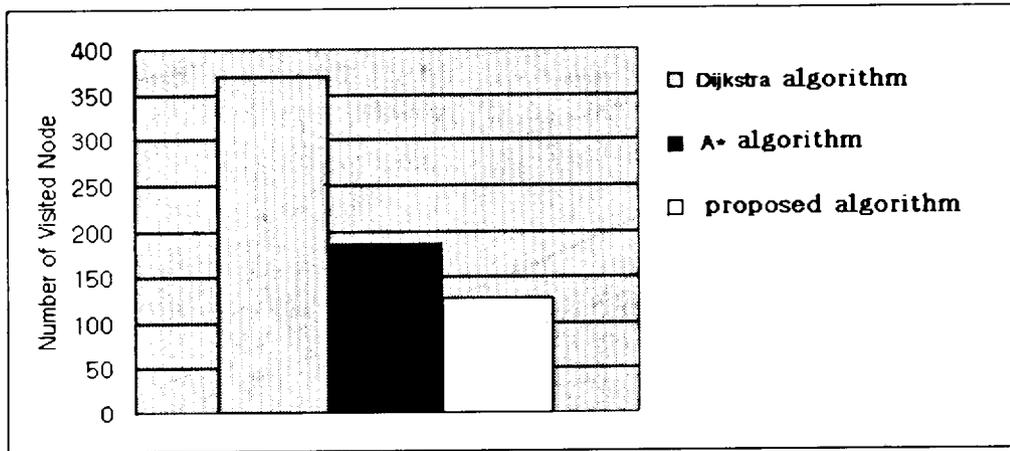


Fig. 8 Sum of searched nodes

증들이 전체 실험을 통해 방문한 노드 수의 총 합을 도표로 나타낸 것이다.

V. 결론 및 향후 연구

본 논문에서는 차량 항법 시스템에 적용하기 위한 빠른 경로 탐색을 할 수 있는 알고리즘을 제안하고 시뮬레이션을 통하여 그 성능을 평가하였다. 제안된 알고리즘은 디지털 도로 맵상의 위치정보를 이용하도록 고안되었으며 탐색 범위를 줄이기 위하여 출발지

와 목적지간에 만나는 각 노드에서 방향각 정보를 이용하였다. 제안된 알고리즘은 경로 탐색을 위한 다양한 경로 탐색 알고리즘들과 시뮬레이션을 통해 비교 측정하였다. 측정결과 정확성과 빠른 탐색 시간 모두를 요구하는 차량 항법 시스템에서의 경로 탐색에서의 사용이 가능하다는 평가를 할 수 있었다.

본 연구에서는 평가의 편의를 위해 지도상의 단순 거리를 이용하였다. 현실적으로는 도로의 넓이나 상태, 시간대에 따른 교통량의 변화 등 동적인 교통 상황 정보를 고려해야 할 것이다. 추후 알고리즘의 좀 더 세밀한 평가를 위해 제주도의 전지역 지도를 대상

으로 적용하여 평가를 할 계획이며 거리 이외의 다른 매개변수들을 고려한 알고리즘의 개선도 지속적으로 수행해 나갈 것이다. 제안한 시스템은 향후 차량 소통 정보 및 GPS(Global Positioning System) 등과 연계를 통하여 ITS(Intelligent Transportation System) 등의 응용 서비스에 활용할 수 있으리라 여겨진다.

참고문헌

- 1) 김기섭. 1999. 실시간 차량경로결정을 위한 동적 최단경로 알고리즘. 홍익대학교 산업공학과 석사논문
- 2) 전홍석. 1998. 차량 항법 시스템의 경로 탐색을 위한 탐색 알고리즘들의 성능 비교. 한국정보교육학회논문지, 제2권 2호, pp.252-259
- 3) 남상우, 박문성. 1997. 방향성을 고려한 우편 경로 최적화 시스템의 최단 경로 생성 알고리즘 연구. 한국정보처리학회, 정보처리 논문지, 제4권 2호, pp.491-498.
- 4) Dennis de Champeaux. 1983. Bidirectional Heuristic Search Again. JACM 30(1), pp.22-32.
- 5) E. van Dijk. 1982. Heuristic Search with Partial Node Expansion and Bi-Directional Search in Product Space. pp.180-182.
- 6) Leni Sint. 1977. An Improved Bidirectional Heuristic Search Algorithm. JACM 24(2), pp.177-191.
- 7) Andrea Borella, Franco Chiaraluce. 1998. Multicast routing in BMSNs through greedy, weighted greedy, and Dijkstra algorithms: a comparative analysis. SYBEN, pp.444-454.
- 8) P. Hart, N. Nilsson, and B. Raphael. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems, Science and Cybernetics, SCC-4(2), pp.100-107.