

최적경로 탐색을 위한 유전자 알고리즘과 A* 알고리즘의 적용

조원혁, 공창욱, 김인택
명지대학교 공과대학 제어계측공학과

Application of GA algorithm and A* algorithm
to optimal path finding problem

Wonhyuk Cho, Changwook Kong, Intaek Kim
Myong Ji Univ. Dept. of Control and Instrumentation

Abstract

In this paper, we applies two methods - Genetic Algorithm(GA) and A* Algorithm - to find the optimal path in route guidance system. Under the assumption that the traveling costs of each link are given, the task to find the optimal path becomes very complicated problem if the number of nodes or links increase. Two well-known algorithms are modified to resolve the problem and the preliminary demonstration show both optimistic result and needs to improvement.

1. 서론

자동차의 이용을 통한 육상교통의 급속한 증가는 교통의 혼잡으로 인한 경제적, 환경적인 문제점을 야기시키며, 미국, 일본, 그리고 유럽에서는 지능형교통체계(Intelligent Transport System)[1]를 도입하여 그 피해를 최소화하고 보다 쾌적한 교통환경을 조성하려고 노력하고 있다. 교통혼잡을 줄이는 방법으로 기존도로의 확장 보안을 우선적으로 생각할 수 있으나, 경제적으로 큰 부담이 되기 때문에, 현재의 전기, 전자, 통신 등의 첨단기술을 이용하여 운전자에게 적절한 정보를 제공하므로써 최적의 도로 사용을 유도하는 것이 현실적으로 바람직하다.

이와 관련하여 주행안내시스템(navigation system)은 운전자에게 자신의 위치와 전자지도를 제공하여 목적지로 갈 수 있는 경로를 알려주는 장치이다. 이러한 주행안내시스템의 가장 중요한 역할 중 하나가 운전자에게 실시간 최적경로(optimal path)의 제공이다. 이를 위해 본 논문에서는 유전자 알고리즘과 A* 알고리즘을 사용하여, 임의의 주어진 문제를 해결해 보았다. 유전자 알고리즘의 진화연산의 일종으로 주행경로를 유전자형으로 표현하여 최적의 유전자를 찾아, 최적 경로를 찾았으며, A* 알고리즘은 인공지능의 탐색 알고리즘으로 Dijkstra 알고리즘을 개선하여 얻은 것이다.

본 논문은 유전자 알고리즘, A* 알고리즘의 시뮬레이션 결과를 살펴보고, 향후 두 알고리즘의 성능의 개선을 위한 몇 가지 점들은 결론에서 지적한다.

2. 문제의 정의

주행안내시스템의 전자지도에서 최적경로를 찾기 위한 방법을 적용하기 위해 그래프 이론[2]을 사용하게 된다. 그래프는 일반적으로 비텍스(vertex)와 에지(edge)로 이루어져 있는데 이는 각각 도로의 교차로와 교차로 사이의 도로를 의미한다. 출발점과 도착점 모두 교차로이며, 본 논문에서는 그림 1과 같은 도로망이 있다고 가정하자.

본 논문에서는 유전자 알고리즘과 A* 알고리즘을 사용하여 그림의 비텍스 10에서 16까지 최적경로를 찾는 문제를 해결하고자 한다. 각 에지에 양방향으로 주어진 값은 비용을 의미하는 것으로 비용이 적을수록 통행시간이 적음을 의미한다.

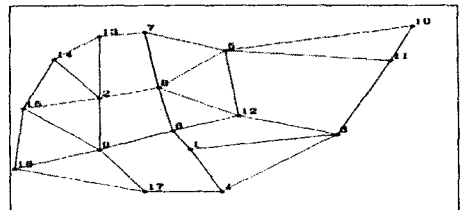


그림 1. 시뮬레이션 도로망 그래프

3. 유전자 알고리즘의 적용

주어진 문제에 유전자 알고리즘[2]을 적용하기 위해서는 다음과 같은 요소를 확정해야 한다.

- 염색체의 구성
- 유전연산자의 개발
- 적합도(fitness) 계산의 방법
- 각종 지배요소들의 최적화

하지만 본 논문의 최적경로 탐색문제는 먼저 시작점에서부터 종점까지 경로의 길이가 서로 다르기 때문에 고정된 염색체 구조의 적용이 어렵고 또한 그에 따른 유전연산자 적용도 어렵다. 이런 점들을 극복하기 위해 염색체는 그대로 경로 자체를 구조로 단 방향의 링크드 리스트(linked list)[3]로 표현하였다. 그리고 교차하는 그림 2.(a)와 같이 서로 만나는 지점(a,b,c)을 찾아 교차를 실시하고, 돌연변이는 그림 2.(b)와 같이 작은 부분에서의 우회로가 돌연변이가 되도록 하였다. 이는 교차의 넓은 영역의 탐색 그리고 돌연변이의 좁은 영역의 탐

색을 경로의 조합에서 만족해 준다. 높은 적합도를 가진 염색체들끼리 공통의 버텍스를 가지고 있을 때 그 염색체들이 구성하는 그 지점에 대한 경로의 조합의 경우 중에 더 좋은 경로가 있을 확률은 더 크다고 볼 수 있다.



그림 2. 최적경로 탐색을 위한 유전자 연산자:
(a)교차, (b)돌연변이

유전자 알고리즘의 적용에 있어서 또 중요한 것은 '적자생존의 원리'가 들어간 적합도의 계산이다. 적합도는 염색체의 성능을 수로 표현한 것인데 더 좋은 성능의 개체가 더 높은 점수를 가질 수 있게 해야 한다. 이는 부모 선발의 기준이 된다. 본 논문의 제시된 문제는 최소의 비용을 소요하는 경우이므로 다음과 같이 적합도(fitness)를 구하였다.

$$fitness = FF / \sum_i cost_i$$

위의 수식은 각 경로 에지의 비용(cost)의 합에 반비례한다. 따라서 적은 비용은 높은 적합도를 갖게 되므로 적은 비용의 경로들이 살아남게 하는 것이다. 여기서 FF는 상수로써 분수 함수의 굴곡의 정도를 지배하게 되어 부모의 선발패턴을 조절 하는데 쓰여진다.

앞서 제시된 알고리즘의 요소들은 각기 확률, 수식의 상수 계수 등으로 성능을 조절할 수 있는데 이는 각각의 문제마다 다를 수 있으므로 실험적으로 구하게 된다.

4. A* 알고리즘의 적용.

A* 알고리즘의 적용을 위해 다음과 같은 정의가 필요하다. 쌍방향 그래프 $G=(V,E)$, $k(v,u)$ 는 $e(v,u)$ 의 비용, 시작점 s , 목표점 t , 점(vertex)의 개수 m , 에지(edge)의 개수 n .

A* 알고리즘은 최적경로 탐색기법으로 잘 알려진 Dijkstra 알고리즘[4]을 확장한 보다 효율적인 탐색기법이다. Dijkstra 알고리즘은 한정된 탐색구간에서 완벽한 최적 경로 발생을 보장해 주지만 최적 경로와 무관한 구간까지도 탐색구간에 포함시키는 비효율성을 가지고 있다[5][6].

A* 알고리즘은 Dijkstra 알고리즘의 시작점으로부터의 최소 비용경로 우선 선택 탐색기법($g(n)$)과 그리디(Greedy) 알고리즘의 목표점까지의 최단거리경로 우선 선택 탐색기법($h(n)$)을 결합한 탐색기법이다. 탐색기법에 포함되어 있는 점 n 으로부터 목표점까지의 최소경로비용의 비교로 단일방향 탐색

알고리즘의 문제점을 해결하였다. 목표점까지의 최적경로 탐색구간과 무관한 탐색구간으로의 탐색 방향성을 억제하고, 목표점을 포함하는 방향으로 탐색이 진행되도록 유도하여 탐색 시간을 개선하였다.

시작점에서 목표점까지 점 n 을 반드시 지나는 비용, $f(n)$ 은

$$f(n) = g(n) + h(n)$$

$g(n)$: 시작점부터 점 n 까지의 비용
 $h(n)$: 점 n 부터 목표점까지의 비용

이며, 점 n 부터 목표점까지의 최소거리, $h(n)$ 의 실제거리는 휴리스틱(heuristic) 방법을 사용하여 추측되어지며, 휴리스틱(heuristic)함수, $h(n)$ 은 $\forall n \in G$ 대하여 $0 \leq h(n) \leq h'(n)$, ($h'(n)$ 은 실제 거리) 과 양방향 선택 가능성을 만족해야 한다.

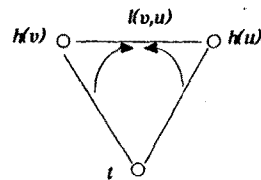


그림 3. Euclidean distance의 양방향가능성.

본 논문에서 $h(n)$ 의 휴리스틱 추측방법으로 양방향 선택 가능성이 있는 Euclidean distance를 사용하였다.

$\forall(v,u) \in E$ 에 대하여 $h(u) \leq h(v) + k(v,u)$
목표점 t 에 인접하여 연결된 점 n 의 t 까지의 최소비용 $h(n)$ 와 n 에 연결된 점 n' 간의 비용 $k(n,n')$ 을 Euclidean distance에 적용하여 $h(n')$ 을 구하였고 동일한 방법으로 G 의 모든 점들의 $h(n)$ 을 유도해 내었다.

$$\forall n, n' \in G \text{에 대하여 } 0 \leq h(n) \leq h(n, n')$$

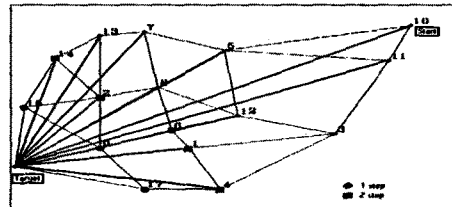


그림 4. $h(n)$ 의 유도

$h(n)$ 을 구하고 나면 Dijkstra 알고리즘과 동일한 조건이 갖춰지게 된다.

지금까지 설명한 A* 알고리즘은 다음과 같은 과정을 거쳐 수행한다.

1. list OPEN,CLOSED를 초기화하고, OPEN에 시작점 s 를 보관한다.
2. OPEN중 $f(N)$ 이 최소인 점 N 을 찾고, CLOSED로 이동시킨다. 만일 OPEN이 비어 있다면 탐색은 실패이다.
3. 만일 N 과 목표점 t 가 일치한다면 탐색은 성공이다. 그렇지 않을 경우, N 에 연결된 점 N' 의 $f(N')$ 을 구하고 OPEN에 보관한다.
4. 2번으로 이동하여 탐색을 계속한다.

A*알고리즘 수행 소요시간은 list OPEN이 정렬되어 있지 않을 경우, O(n) 회수 동안 O(n)번의 최소값 검색회수와 O(n) 번의 자식 점(vertex)으로의 확장이 이뤄지므로 $O(n^2)$ 이 소요 되고, 이번 실험에서 사용과 같이 list OPEN이 정렬되어 있을 경우 $O(m \log n)$ 이 소요된다.

5. 시뮬레이션 결과

5.1 유전자 알고리즘

유전자 알고리즘의 시뮬레이션 결과는 최적 경로와 그와 준최적경로로 신속한 수렴을 보였다. 돌연변이율에 관계없이 약 10세대 이전에 이미 최적경로와 그와 준한 경로로 수렴하였다.

돌연변이율 세대	0.002	0.003	0.004	0.02	0.03	0.04	0.2	0.3	0.4
1	78	227	49	24	1	28	(91)	2	2
2	2	(64)	5	(91)	12	(64)	3	1	4
3	1	3	(91)	(91)	18	5	(64)	6	1
4	(64)	5	4	3	4	2	2	2	1
5	41	(64)	8	7	12	1	10	2	1
6	340	(64)	93	3	5	23	6	1	4
7	(64)	(91)	(91)	4	(91)	2	(64)	1	2
8	6	79	(64)	2	4	(91)	(64)	2	(91)
9	41	18	54	1	(91)	(91)	3	(91)	1
10	(64)	2	(64)	7	(91)	3	(91)	3	4

표 1. 각 돌연변이율에 따른 최적 경로 탐색에 걸린 세대수 (괄호 안의 숫자는 1000세대 이후의 최적경로의 비용)

돌연변이율이 작을 때(0.2이하)에는 하나의 경로에 빨리 수렴되고 변화가 없는 양상을 보인 반면 돌연변이율이 커짐(0.3 이상)에 따라 빠른 수렴이 억제되고 공평한 탐색이 이루어져 최적해를 찾음을 알 수 있다.

5.2 A* 알고리즘

본 논문에는 최대 가지(branch)의 수가 5개 이고, 버텍스의 수가 각각 17개, 100개를 가진 네트워크 2개를 시뮬레이션 대상으로 사용하였고, 네트워크의 양 끝점에 고정되어 있는 임의의 두 점의 경우와 예지와 연결된 점들로 둘러싸인 임의의 두 점의 경우 2가지에 대해 시뮬레이션 하였다.

표1과 2는 각각의 문제에 A* 알고리즘과 Dijkstra 알고리즘을 적용하여 (최적경로비용, 방문한 점의 수)로 비교한 결과이다.

s→t	17	
	A*	Dijkstra
4→7	(22,28)	(22,53)
7→4	(19,39)	(19,53)
3→14	(38,46)	(38,56)
14→3	(14,34)	(14,38)

표 2. 점 17의 네트워크

s→t	100	
	A*	Dijkstra
100→10	(8,168)	(8,357)
10→100	(8,193)	(8,383)
80→57	(7,129)	(7,258)
57→85	(7,157)	(7,312)

표 3. 점 100개의 네트워크

실험결과를 살펴보면 A* 알고리즘은 최소비용경로 선택 탐색 기법에 의해 방문점의 개수가 현저히 줄어들어 탐색시간이 줄어들음을 볼 수 있다.

6. 결론

본 논문에서는 최적경로 탐색을 위해 유전자 알고리즘과 A* 알고리즘의 적용을 살펴보았다. 시뮬레이션 결과는 17 개의 교차점을 가진 경로에서 실시하여, 두 알고리즘이 최적경로를 찾는데 성공적이었다. 교차점의 증가에 따른 성능의 평가와 두 알고리즘의 직접적인 비교가 향후 과제로 남게되었다.

참고문헌

1. Richard Whelan, *Smart Highways, Smart Cars*, Boston, London, Artech House, 1995
2. Goldberg, David E. *Genetic algorithms in search, optimization, and machine learning*, Addison Wesley, 1989
3. Mark Allen Weiss, *Datastructure and Algorithm analysis in C*, The Benjamin Cummings Publish company, 1993
4. Tetsuo Shibuya, et al., "Finding a Realistic Detour by AI Search Techniques" Proc. of the second world congress on ITS '95 YOKOHAMA, Vol4, 1995, pp.2037-2044.
5. S. J. Russell and Peter Norvig, "Artificial Intelligence a Modern Approach", Prentice-Hall, 1995.
6. Thomas Dean, et al., "Artificial Intelligence Theory and Practice", Addison-Wesley, 1995.