

이동로봇의 전역경로설정방법

정 현 박기두 최한수
조선대학교 제어계측공학과

The Grabal Path-planning for Mobil robot

Heon Jeong Ki-Du Park Han-Soo Choi
Dept. of control & instrumentation Eng. Chosun University

Abstract - There are a lot of paths which connect between the mobile robot and the goal point. To make a mobile robot arrive at the goal point fastly, The optimal path is needed and a path painning is necessary.

In this paper, we propose a new method of path planning to find a path for mobile robot. It is based on Ginetic Algorithm for serching the optimal grobal path planning. Simulations show the efficiency for the grobal path planning.

1. 서 론

로봇이 이동하기 위해서는 로봇의 현재 위치에서 목표점까지의 경로가 설정되어야 하며, 경로 계획은 로봇의 주변환경 정보에 따라 크게 두가지 형태로 구분된다. 활동 영역 안에 있는 장애물의 위치와 모양 등의 정보를 완벽하게 인지하고 경로를 계획하는 전역경로계획(Global Path Planning)과 장애물에 대한 사전지식이 없이 경로를 계획하는 지역경로계획(Local Path Planning)이 있다.

안정되면서 최적의 이동을 하기 위해서는 먼저 주어진 지도에서 최단 전역 경로를 얻고, 얻어진 경로를 이동하면서 예기치 못한 장애물을 만났을 경우 충돌회피를 통해 경로를 계속 유지하는 방법이 필요하다.

본 연구는 지도로부터 얻어진 주위환경에서 출발점과 목표점을 잇는 최단 전역경로를 찾는 새로운 알고리즘을 제시하는 것이다. 생성된 여러개의 경로 중 최적의 경로를 빠른시간에 안정적으로 찾을 수 있도록 선택과 교차 및 돌연 변이 과정을 통한 적자 생존의 원리가 적용되는 유전자 알고리즘(Ginetic Algorithm)을 적용하였다. 유전자 알고리즘은 자연의 생물학적인 유전자 정보와 그 진화과정의 기본원리에 바탕을 둔 기계적 학습방법이다. 또한, 유전자 알고리즘은 지금까지의 방법으로써는 해결이 곤란한 문제에 대해서 실용상의 최적해를 랜덤하고 빠르게 찾는 방법으로서 사용되어 왔다. 유전자 알고리즘은 종래의 검색 알고리즘과 달리 검색 공간 중 하나의 점뿐만 아니라 많은 점을 동시에 고려하기 때문에 국부 최소점(local minima)점에 수렴되는 현상을 극복할 수 있다.

2. 본 론

2.1 유전자알고리즘

유전자 알고리즘은 1970년대 초기에 Jone Holland에 의해 제안된 자연 도해의 원리를 바탕으로한 최적화 수법이다.¹⁾ 또한, 유전자 알고리즘은 지금까지의 방법으로써는 해결이 곤란한 문제에 대해서 실용상의 최적해를

랜덤하고 빠르게 찾는 방법으로서 사용되어 왔다. 유전자 알고리즘은 종래의 검색 알고리즘과 달리 검색 공간 중 하나의 점뿐만 아니라 많은 점을 동시에 고려하기 때문에 국부최소점(local minima)점에 수렴되는 현상을 극복할 수 있다.

본 논문에서는 정수 유전자 표현 방법을 사용하였다. 유전자 알고리즘은 입력 값의 집단을 랜덤하게 선택하는 것에서부터 시작한다. 선택된 모집단(population)보다 최적해에 가까운 집단을 만들어내는 세대교체(generations)를 반복하여 진화시킨다

2.2 로봇의 전역경로 탐색

본 논문에서는 유전자 알고리즘을 이용한 자율이동로봇의 전역 이동 경로 계획 방법을 제시하였다.

먼저 로봇이 이동하여야 할 전체 영역에 대한 지도가 주어지고 주어진 지도를 셀분해법²⁾에 의해 셀영역으로 분해 한다. 분해된 셀은 로봇과 장애물을 점유하는 지도로 재 구성되며, 장애물의 모서리영역을 탐색한다. 모서리영역들은 그래프의 형태를 가진 자료구조로서 재 구성되어지며, 모서리영역을 이동하는 최단 경로를 유전자 알고리즘을 통하여 탐색하게 된다.

2.2.1 셀분해법

Vision system 또는 미리 저장된 데이터로부터 지도(map)를 얻는다. 장애물이 있는 픽셀은 1로 그렇지 않은 경우는 0으로 설정되어 있다.

$$map(x, y) = \begin{cases} 1: obstacle \\ 0: otherwise \end{cases} \quad (1)$$

셀 분해법은 활동영역을 셀 단위로 처리하기 때문에 활동 영역에 대한 데이터가 줄어들기 때문에 데이터 계산량을 줄일 수 있는 장점이 있다.

한 셀 면적의 10%이상을 장애물이 차지 할 경우는 장애물이 존재하는 셀로 설정하도록 하였다.

$$M(i, j) = \begin{cases} 0: O_{PA} < 10 \\ 1: i=0 \text{ or } j=0 \text{ or } i=W_{MAP}-1 \\ \text{or } j=H_{MAP}-1 \\ 2: O_{PA} \geq 10 \end{cases} \quad (2)$$

2.2.2 장애물 모서리 셀 탐색

본 연구에서는 로봇의 이동 경로의 경우 수를 감소시키기 위하여 셀분해로부터 얻은 지도에 가시도표(Visibility Graph)방법³⁾을 적용하여 장애물의 모서리를 탐색하였다.

장애물 모서리 탐색알고리즘은 최단 경로 정의 집합의 수를 줄일 수 있는 방법의 하나로서 만약 장애물이 최단 경로 상에 놓여 있을 경우 장애물의 모서리를 따라 움직이는 것이 최단거리임을 알 수 있음을 이용한 것이다.

FMAP은 MAP에서 모서리의 셀을 찾아 FM(x,y)의 값을 3으로만 치환하는 효과를 얻는다.

$$FMAP = (FM : FM(x,y), x=0 \sim W_{MAP}, y=0 \sim H_{MAP}) \quad (3)$$

$$FM(x,y) = \begin{cases} 3 : FMC = TRUE \\ M(x,y) : Otherwise \end{cases}$$

장애물 모서리 탐색을 통하여 장애물을 회피하여 이동할 수 있는 지역 최단 경로 정보를 얻을 수 있었다. 본 논문에서는 얻어진 모서리 노드를 연결하여 임의의 경로를 설정한다. 그렇게 하기 위해서는 그래프 자료구조 형식을 가진 node정보군이 필요하다.

본 논문에서는 node 집합을 생성하기 위해서 각 node에서 5도 간격으로 360도를 회전시켜가면서 다른 node와 연결 가능 셀을 탐색한다. Node_i의 경로 그래프의 자료구조는 식(4)와 같은 구조체로서 정의가 되어 있다.

$$NODE = \{N : N_i = (x, y, Nnode, NC) \quad i : 0 \sim T_{node}, FM(x_N, y_N) = 3\} \quad (4)$$

$$NC_N = \{NC_0, NC_1, \dots, NC_n\}, n = Nnode - 1$$

노드 N_i는 FMAP에서의 임의의 셀 FM(x_{*u*}, y_{*u*})의 정보가 3일 때 생성되며, 그 때의 x_{*u*}, y_{*u*}는 N_i노드의 위치정보 x, y로서 저장된다. NC집합은 N_i노드에서 장애물을 통과하지 않고 선분을 연결할 수 있는 다른 노드들에 대한 집합이다. Nnode는 NC의 원소 개수이다.

2.3 유전자알고리즘 적용

초기의 모집단을 생성한다. 생성된 모집단에 대한 평가를 하여 생성된 4개의 경로집합(GO)을 적응도가 높은 순서로 정렬한다. 다음세대를 위하여 두 가지의 과정을 수행한다. 첫 번째 과정은 교차이고, 두 번째 과정은 돌연변이이다. 먼저 교차과정에 대하여 알아보면, 기존의 최 우수집단과 4개의 생성된 집단 중 첫번째 집단을 교차시켜 다음세대 집단 0.1을 만들고, 두번째 집단과 교차시켜 다음세대 집단 2.3을 생성한다. 다음으로 돌연변이의 과정을 보면 1/3확률값으로 랜덤하게 발생하는 돌연변이 플래그에 의해 동작되며, 방법으로는 다음 기존의 최우수집단과 4개의 우수집단 중 첫번째 집단에서 임의의 점을 택하고, 그 점부터 랜덤하게 경로를 설정하여 새로운 집단을 만든다. 또한 생성된 최 우수집단은 기존의 최 우수집단과 비교하여 적응도를 비교하여 최 우수집단을 갱신한다. 생성된 집단은 재 평가를 통하여 우수집단을 선택하여 교차하는 과정을 반복한다. 일정 횟수를 반복하여 가장 좋은 경로를 최단 경로로 선택한다.

2.3.1 초기 모집단 생성

유전자알고리즘을 적용하기 위하여 먼저 사각 경로 탐색에 의해 얻어진 가능경로의 그래프를 읽는다. 입력된 가능 경로를 바탕으로 로봇의 위치를 초기 node로 하여 랜덤하게 초기의 모집단을 설정한다. 식 10을 사용하여 초기 모집단을 얻는다. 초기의 모집단 갯수(k)는 4로 설정하였다.

$$GO = (G_i^k : \begin{cases} 0 : i=0 \\ NC_{i(N)} : 1 < i < NGO^k \end{cases}) \quad (5)$$

$$l = random(Npath \ G_i^k)$$

$$m = G_{i-1}^k, k=0 \sim 3$$

GO 집합은 Node 집합에서 원소들을 랜덤하게 선택하는 것과 같다. 즉 노드 N_i의 Npath사이의 값 중에 랜덤한 수 l를 생성하여 N_i와 연결가능한 노드에 대한 정보집합(NC)에서의 l번째 노드번호를 참조하여 GO^k 집합의 i번째 원소로 설정한다. 다음 GO^k집합의 i+1번째 원소는 GO^k집합의 i번째 원소에 대한 노드 정보를 참조하여 설정한다. 두 node간 무한 루프에 빠지는 것을 방지하기 위하여 p node에서 q node이동하였을 경우에 p와 q의 연결을 끊는다. 즉 p node에서 이동 가능 node 번호에서 q번을 삭제하고, q node에서 이동 가능 node번호 중에서 p번을 삭제한다.

G의 원소인 노드번호가 목표점의 노드번호와 같거나 Npath_{G^k}의 값이 0일 경우에는 GO^k 집합에 대한 경로 설정을 종료하고 그때의 i값을 NGO(G^k)로 설정한다.

2.3.2 평가

얻어진 경로들에 대한 평가를 실시한다. 본 논문에서 사용한 평가 함수는 식 6, 7과 같다.

평가함수 fit는 거리함수에 반비례하며, 최종 노드값이 목표점의 노드값인지를 판별하여 아닐 경우는 거리함수에 가중치를 더하였다. 가중치의 추가로 마지막 노드가 목표점인 경우에는 노드들의 거리합이 작은 것이 좋은 경로이며, 목표점이 아닌 경우는 가중치의 추가과정을 거쳐 가장 많은 경로를 거친 경로가 좋은 경로이다. PDist는 백분율 거리값이다.

$$Dist^k = \begin{cases} \sum_{i=0}^{Tnode-1} \sqrt{(x_{N_i} - x_{N_{i+1}})^2 + (y_{N_i} - y_{N_{i+1}})^2} \\ \text{if } (x_{N_{Tnode-1}} = x_G \text{ and } y_{N_{Tnode-1}} = y_G) \\ \sum_{i=0}^{Tnode-1} \sqrt{(x_{N_i} - x_{N_{i+1}})^2 + (y_{N_i} - y_{N_{i+1}})^2} \\ + (NGBEST - Nnode_n) * 20 \\ \text{if } (x_{N_{Tnode-1}} \neq x_G \text{ or } y_{N_{Tnode-1}} \neq y_G) \end{cases} \quad (6)$$

$$fit^k = \frac{100}{PDist^k} \quad (7)$$

2.3.3 선택, 교차, 돌연변이

평가함수의 값을 키값으로 하여 내림차순으로 정렬한다. 가장 fit값이 높은 경로가 SG⁰에 위치하게 되고, 그 다음으로 SG¹, SG², SG³으로 위치하게 된다.

$$SGO = sort(GO) \quad (8)$$

본 논문에서는 새로운 세대에 대한 경로를 위하여 SG⁰ 경로와 SG¹의 경로를 선택하고, N_Best와 교차 또는 돌연변이를 실시한다.

$$\begin{cases} NGO(0,1) = cross(SGO^0, BEST) : CrossFlag = 0 \\ NGO(2,3) = cross(SGO^1, BEST) : CrossFlag = 1 \\ NGO(0,1) = RanGO(SGO^0, GO^1) : CrossFlag = 2 \\ NGO(2,3) = RanGO(SGO^2, GO^3) : CrossFlag = 3 \end{cases} \quad (9)$$

교차는 CrossFlag에 의해 작업이 분리된다. Crossflag 0.1은 교차를 수행하여 집합 NG⁰를 얻을 경우에 설정되고, 2.3은 교차 실패 또는 돌연 변이를 수행할 경우의 NG⁰를 얻고자 할때 설정된다. Cross(SG¹, SG²)함수는 경로 SG¹에서 1이상의 랜덤한 node N1을 선택하고 SG²에서 같은 노드 N2를 탐색한다. 만약 N2가 SG²에 존재한다면 SG¹에서 N1노드 이하 노드와 SG²경로에서 N2노드 이하의 노드 중에 다음

의 같은 노드 N3와 N4를 탐색한다. 얻어진 N1에서 N3까지의 경로를 N2에서 N4까지 서로 교차 시킨다. 만약 N1과 N2는 존재하였으나 N3, N4가 존재하지 않을 경우는 N1이하와 N2이하의 경로 모두가 교차될 것이다.

2.3.4 저장 및 진화

SG(0)와 Best의 적응도를 비교하여 좋은 것을 Best로 저장, 갱신한다.

$$\text{if } \text{fit}_{GO(0)} > \text{fit}_{\text{Best}} \text{ then } \text{Best} = \text{copy}(\text{SGO}(0)) \quad (10)$$

다음 세대로 얻어진 NGO의 path를 GO로 이동시키고 max_time까지 평가, 선택, 교차, 돌연변이 과정을 반복 수행하여 최적의 경로를 도출해 나간다.

$$\text{REGENERATE : } GO_i = \text{NGO}_i \quad (11)$$

2.4. 모의실험

본 논문에서 제안한 유전자 알고리즘을 이용한 최단 경로 탐색방법을 여러가지 지도에 대하여 적용하여 그 효과를 검증하였다. 모의실험의 과정은 임의의 Vision data를 생성하여 BMP file로 저장하고, 저장된 지도 내용을 읽어들이며 셀 분해법을 이용한 새로운 셀단위의 지도를 새로이 얻는다. 얻어진 지도에서 장애물의 사각 모서리 정보를 얻고 가능 경로 그래프를 도출하였다. 그

표 1. 모의실험 1의 거리함수값 결과

time	1	4	6	41
dist	80.584	59.879	58.862	55.093
fit	0.517	0.920	0.567	0.412

래프를 랜덤하게 발생하여 초기 모집단을 설정하고 평가, 선택 및 교차, 돌연변이의 과정을 반복하였다.

1차 모의실험 지도로서는 그림 1(a)과 같이 셀분해된 지도를 통하여 모의실험을 하였다. 셀분해 때의 폭과 높이의 크기는 15pixel로 설정하였으며, 전체 이동로봇의 초기위치는 M(2,2), 골의 위치는 M(3,21)으로 설정하였다. max_time은 100으로 설정하였다. 그림 1의 (b)는 장애물의 모서리를 탐색하여 사각 표시를 하고 그래프를 형성한 후 유전자알고리즘 적용을 하여 초기 모집단에서의 best경로를 표현한 것이며, 그림 1의 (c)는 4번의 진화 후, 즉 time=4일 경우의 Best경로를 사선으로 표현한 것이며, 그림 1의 (d)는 time=6일 경우에 대한 최적 경로이다. 그림 1의 (e)는 time=41경우의 최적경로이며, max_time까지 같은 경로를 얻었다.

그림에서 볼 수 있는 것과 같이 초기에는 복잡한 경로를 설정하였으나, 여러회의 진화를 통하여 최적경로로 수렴되는 것을 알 수 있다. 각각에 대한 dist 값은 다음 표 1과 같이 얻었다. dist값이 time=1일 경우에 비해 time=41일 경우의 값이 작음을 볼 수 있다. 같은 장애물을 15번 반복을 실험을 하였으며, 그 결과 평균 최단 경로의 수렴 time은 34.2이었다.

2차 모의실험은 좀더 많은 셀로 지도를 분해하였을

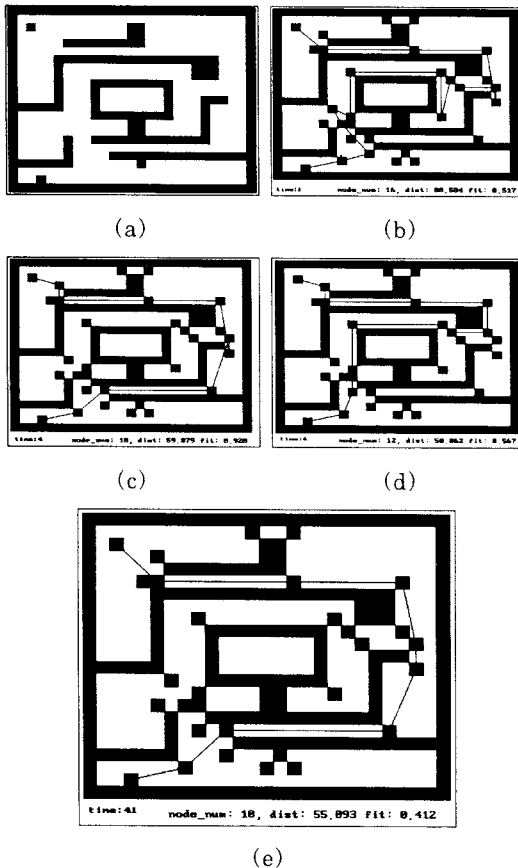


그림 1. 모의 실험 1 결과

Fig 1. The result of experimental simulation 1

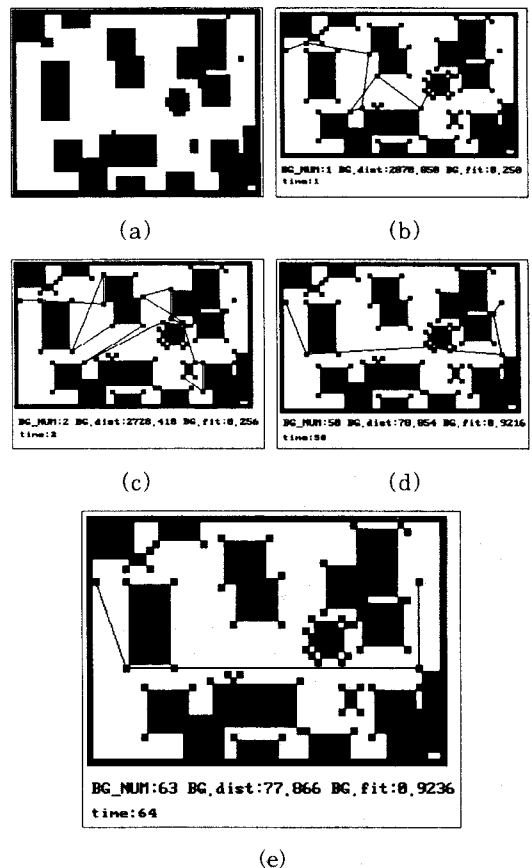


그림 2. 모의 실험 2의 결과

Fig 2. The result of Experimental simulation 2

표 2. 모의실험 2의 거리합수값 결과

time	1	2	50	64
dist	2070.8	2728.4	78.854	77.866
fit	0.250	0.256	0.9216	0.9236

경우에 대하여 최단경로 탐색 실험을 실시하였다. 셀분해 때의 폭과 높이의 크기는 5pixel로 설정하였으며, 전체 이동로봇의 초기위치는 M(1,10), 골의 위치는 M(55,10)으로 설정하였다. max_time은 150으로 설정하였다. 셀분해된 지도는 그림 2의 (a)와 같고, 그림 2의 (b)는 그래프 생성후의 유전자알고리즘 적용하여 얻은 초기의 최적경로이다. 그림 2의 (c)는 두 번째 진화시의 최적경로이다. 그림에서 볼 수 있는 것과 같이 초기의 경로와 두 번째 경로는 목표점에 도달하지 않은 경로가 최적임을 볼 수 있다. 목표점에 도달 하지 않은 경로의 비교시에는 가장 장거리 경로가 최적 경로로 판단하도록 평가함수를 설정하였다. 그림 2의 (d)는 time=50일 경우의 최적경로 이며, 그림 2의 (e)는 time=64일 경우의 것이다. max.time까지 진화를 하여도 time=64의 경로를 유지하였다.

10번의 실험을 반복수행하였으며 평균 70번에 수렴함을 보였다. 모의 실험1에서와 거의 비슷한 100번 이하의 탐색으로도 최단 경로를 설정할 수 있음을 모의실험을 통하여 알 수 있었다.

3. 결 론

본 논문에서는 유전자 알고리즘을 이용한 이동로봇의 전역 경로 계획 방법을 제안하였다. 유전자 알고리즘을 사용함으로써 목표점의 변동과는 무관하게 최단의 전역 경로에 안정적 수렴을 하는 것을 확인 하였으며, 수행가능한 여러가지 경로 수에 비해 빠른 탐색이 가능함을 확인하였다.

본 논문에서 제안한 전역경로 계획 방법은 먼저 지도를 입력받고, 이것을 셀단위로 분해하여, 모서리 탐색에 의한 가시 그래프를 생성하는 과정을 거친다. 여기에서 얻어진 가시그래프의 정보로부터 최적의 전역경로를 탐색하기 위해 유전자 알고리즘 적용하였다. 유전자 알고리즘에서 초기의 랜덤한 경로 모집단에서 출발하여 fit 값을 증가시키는 최적의 경로를 찾는다. 제안한 방법을 검증하기위한 모의실험에서는 전체 27×23와 80×60 셀에서의 장애물에 대한 지도에서 장애물의 갯수와 관계없이 약 100회 이내의 반복 탐색으로 최단 경로가 설정됨을 확인하였다. 본 논문에서 제안한 전역경로 설정방법은 지도상에 존재하는 경의 수가 기하급수적임에도 불구하고, 또한 수식적인 전개없이 유전적 적자생존의 원리를 이용하여 효과적으로 최단 경로가 설정됨을 확인하였다.

본 논문은 전역 경로 계획에 대한 연구를 하였으며, 본 연구에서 얻어진 전역경로를 기준 이동 경로로 사용하여 향후 여러 알고리즘을 적용한 지역경로계획에 의해 장애물을 효율적으로 회피하면서 전역경로를 추종할 수 있도록 구현할 계획이다.

(참 고 문 헌)

[1] Holland J., "Adaptation in Natural and Artificial System", The University of Michigan, 1975

[2] Moravec H.P., "Certainty grids for mobil robots", Technical Report Preprint, The Robotics Inst., Carnegie-Mellon Univ., 1986

[3] Oommen B. J. et al, "Robot Navigation in Unknown Terrains Using Learned Visibility Graphs. Part I: The Disjoint Convex Obstacle Case", IEEE Journal of Robotics and Automation, Vol. RA-3, NO. 6, pp. 672-681, Dec. 1987.

[4] Arkin R. C., "Motor schema based navigation for a mobile robot. An approach to programming by behavior", Proc. 1988 IEEE Conf. of Robotics and Automation, 264-271, 1988.

[5] Lozano-Perez, T. and Wesley, M. A., "An Alogorithm for Planning Collision-Free Path Among Polyhedral Obstacles", Comm. of ACM, Vol. 22, NO. 10, October 1979.

[6] David E. Goldberg., "Genetic Algorithms in Serch, Optimazation and Machine Learning", Addison-Wesley, 1989.