

# 유전자 알고리즘과 SOM 알고리즘을 이용한 효율적 경로 탐색

정지인 · 엄도성 · 김광백  
신라대학교 컴퓨터정보공학부

## Efficient Path Search Method using Genetic Algorithm and SOM Algorithm

Ji-in Jeong · Do-sung Eom · Kwang-beak Kim  
Division of Computer and Information Engineering, Silla University  
E-mail : jgi88@naver.com, alex00377@nate.com, gbkim@silla.ac.kr

### 요 약

본 논문에서는 유전자 알고리즘에 SOM 알고리즘을 적용하여 효율적으로 경로를 탐색할 수 있는 방법을 제안한다. 제안된 경로 탐색 방법은 효율적인 경로 탐색에 앞서 유전자 알고리즘에 의해 도출된 각각의 결과 좌표를 뉴런으로 설정하고 각 뉴런들의 모든 거리 값을 SOM 알고리즘에 적용하여 거리에 대한 가중치를 구한다. 뉴런 선택 조건(가장 적은 거리 가중치, 이전에 선택되지 않았던 뉴런)을 만족하는 뉴런 및 해당 뉴런의 이웃 반경 내에 존재하는 뉴런들의 연결 강도를 가우시안 분포(오차율 분포)에 적용하여 변경하고, 가장 강한 연결 강도를 가지는 승자 뉴런에 해당하는 경로를 선택한다. 이러한 과정을 뉴런의 개수만큼 반복하여 모든 뉴런들의 경로를 도출한다.

제안된 방법을 실험한 결과, 기존의 유전자 알고리즘을 이용한 방법보다 제안된 방법이 효율적인 경로를 탐색하는 것을 확인할 수 있었다.

### 키워드

유전자, SOM 알고리즘, 연결 강도, 가우시안 분포

## 1. 서 론

정보 공학에서 최적(화)란, 사용 중인 시스템을 변경 또는 수정하여 작업을 더 효과적으로, 또는 자원을 효율적으로 사용하도록 만드는 작업을 뜻한다. 이를테면, 컴퓨터 소프트웨어를 더 빠르게 실행시키거나 기억 장치나 자원을 덜 차지하게 하여 운영하거나, 전력을 덜 쓰게 함으로써 최적화할 수 있다[1]. 소프트웨어에서 사용되는 계산 작업은 여러 방식에 따라 효율성이 달라진다. 기존의 유전자 알고리즘을 이용한 경로 탐색 방법은 출발지 노드 및 도착지 노드와 중간 노드들의 유클리디안 거리를 계산한 후, 계산된 거리값을 이용하여 개체를 생성한다[1]. 생성된 개체를 유전자 알고리즘의 선택 연산과 교차 연산을 이용하여 새로운 개체를 생성한다. 생성된 개체를 적

합도 조건에 대입하여 중간 경로 노드를 도출하여 최적의 경로를 탐색하였다. 그러나 가장 우수한 노드를 다음 출발지 노드로 선정하는 유전자 알고리즘은 거리 값에 대한 오차율이 발생하여 효율적인 거리 값을 도출하지 못하는 경우가 발생한다.

따라서 본 논문에서는 유전자 알고리즘에 SOM 알고리즘을 적용하여 효율적으로 경로를 탐색할 수 있는 방법을 제안한다.

## II. 유전자 알고리즘을 이용한 좌표 도출

제안된 경로 탐색 방법은 중간 노드와 출발지 노드 및 도착지 노드 간의 유클리디안 거리 값을 가중치 값으로 설정하여 적합도 조건을 검사한

후, 도출된 결과 좌표를 뉴런화하여 각 뉴런들의 모든 거리 값을 SOM 알고리즘에 적용하여 뉴런들의 연결 강도를 이행하여 효율적인 경로를 탐색한다. SOM 알고리즘에 적용하기에 앞서 좌표를 뉴런화하기 위해 유전자 알고리즘을 이용하여 좌표를 도출한다. 좌표를 도출하기 위한 순서도는 그림 1과 같다.

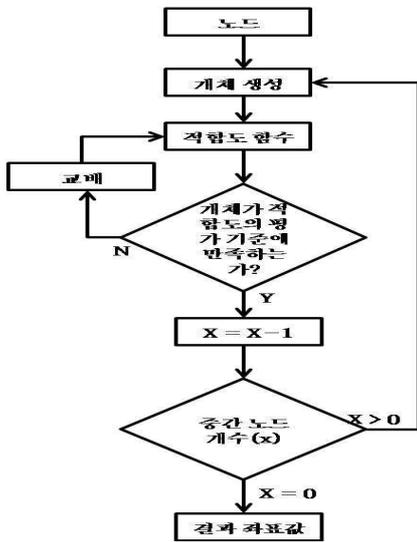


그림 1. 좌표값 도출 흐름도

그림 1과 같이 좌표값 도출을 위해 각 노드 간에 유클리디안 거리를 가중치 값으로 계산하여 유전자 알고리즘의 개체를 생성한다. 이때 개체를 생성할 때에는 다음과 같은 식(1)에 적용하여 가중치 값, 즉 개체를 생성한다.

$$f(x) = A - SRp + ERp \quad (1)$$

식(1)에서 A는 최대 거리 값이고 SRp는 출발지 노드와 중간 노드의 유클리디안 거리이다. 그리고 ERp는 도착지 노드와 중간 노드의 유클리디안 거리를 의미한다. 식(1)의 A는 모든 노드들을 포함하는 직사각형의 대각선 거리이다. 식(1)은 최적의 경로를 탐색하기 위한 개체를 생성하기 위한 식으로 출발지 노드로부터 중간 노드까지의 거리가 작을수록, 중간 노드와 도착지 노드간의 거리가 멀수록 최적의 경로를 의미하므로 제안된 방법에서는 식(1)과 같은 식을 적용하여 개체를 생성한다.

현재 생성된 개체들을 이용하여 다음 세대 개체들의 적합도를 평가하는 기준을 식(2)와 같이 설정한다.

$$f(x) = Max * Percent \quad (2)$$

식(2)의 Max는 생성된 개체들의 값 중에서 최대값이고, Percent의 초기값은 2.0으로 설정하였

다. 개체가 적합도의 평가 기준을 만족하지 못한 상태에서 SOM 알고리즘 적용하면 적합도의 기준을 감소시킨다. 따라서 평가 기준을 개체들에 맞추도록 Percent의 값을 0.1씩 감소시켜 적합도의 평가 기준을 설정한다. 제안된 방법에서는 가장 적합한 개체의 순으로 선택될 확률을 높게 부여하는 방법을 적용한다. 식(3)과 같이 적합도 함수는 특정 해가 얼마나 적합한지 나타내는 함수로 개체가 유전자에 따라 특정 환경에서 살아남을 수 있는 확률이 존재하는 것과 같다. 적합도 함수는 주어진 해가 다음 세대를 생산할 수 있을지 혹은 세대를 남기지 못하고 도태되어 제거될지를 결정한다[2].

유전자 알고리즘에서는 개체들의 값이 클수록 최적의 해를 구하는 환경에서 유리하다. 따라서 식(3)에 개체들을 적용하여 적합도를 계산한다.

$$f(x) = x^2 \quad (3)$$

식(3)의 x는 개체들의 값이다. 식(3)을 적용하여 계산된 값들은 다음 세대를 생산할 수 있을지, 혹은 세대를 남기지 못하고 도태되어 사라질지를 다음과 같은 식(4)를 적용하여 결정한다.

$$f(x) = N / TN \quad (4)$$

식(4)의 N은 식(3)을 계산한 값이고, TN은 식(3)을 계산한 값들의 총합이다. N / TN의 값에 반올림하여 다음 세대를 생산할 수 있는 개체와 도태되어질 개체를 선택하여 제거한다. 그리고 적합도에 만족할 경우에는 중간 경로 노드의 좌표로 설정한다.

### III. SOM 알고리즘을 이용한 경로 탐색

SOM 알고리즘에서 각 뉴런은 연결강도 백터와 입력백터가 얼마나 가까운가를 계산한다. 그리고 각 뉴런들은 학습할 수 있는 특권을 부여받으려고 서로 경쟁하려는데 거리가 가장 가까운 뉴런이 승리하게 된다. 이 승자 뉴런이 출력신호를 보낼 수 있는 유일한 뉴런이다. 또한 이 뉴런과 이와 인접한 이웃 뉴런들만이 제시된 입력백터에 대하여 학습이 허용된다. 이것은 학습에 있어서 전혀 새로운 접근 방식이다. 이 모델이 있기 이전에는 network 에 있는 모든 뉴런들이 반복되는 훈련 과정에서 연결강도를 조정하였다[3].

따라서 본 논문에서는 효율적으로 경로 탐색의 속도를 줄이기 위해 유전자 알고리즘을 통해서 도출된 노드들의 좌표를 뉴런화하여 SOM 알고리즘에 적용한다. 제안된 경로 탐색 과정은 그림 2와 같다.

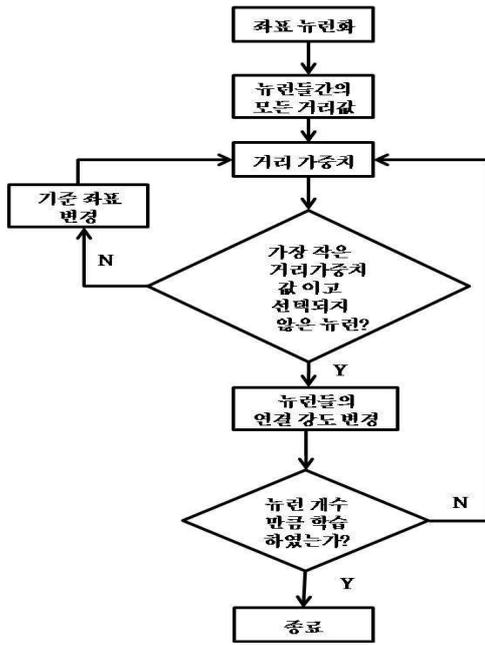


그림 2. 제안된 경로 탐색 방법

효율적인 경로를 탐색하기 위해 유전자 알고리즘을 적용하여 도출된 결과 노드들의 x와 y 좌표 쌍을 입력 벡터로 구성하고 가중치를 부여하여 뉴런화 한다. 그리고 뉴런들 간의 연관성은 가중치와 연관된 모든 입력 벡터과의 거리를 계산하여 인식한다. 따라서 다음과 같이 식(5)를 적용하여 거리를 계산한다.

$$D = (X - N.W_0)^2 + (Y - N.W_1)^2 \quad (5)$$

여기서 X와 Y는 기준이 되는 뉴런의 좌표 x와 y 값이고 N.W(0)와 N.W(1)은 뉴런이 가지는 좌표의 가중치 벡터이다.

식(5)를 통해서 계산된 모든 뉴런들 간에 거리 가중치를 이용하여 효율적으로 경로를 탐색한다. 이 때 가장 거리가 적은 가중치 값을 선택한다. 즉 거리가 가까운 뉴런이 강한 연결강도를 가지게 된다는 것을 의미한다. 그리고 모든 뉴런들의 거리 가중치를 계산하기 때문에 탐색하는 시간이 많이 소요되는 문제점을 개선하기 위해 이전에 선택되지 않은 뉴런이 선택되게 한다. 하지만 이 조건을 만족하지 못한 경우에는 기준이 되는 뉴런을 변경하여 변경된 뉴런의 좌표 쌍을 이용하여 뉴런들 간의 거리 가중치를 다시 계산하여 조건에 만족하도록 한다[4,5].

조건에 만족되는 경우에는 다음과 같은 식(6)을 이용하여 연결 강도를 변경한다.

$$W_{jnew} = W_{jold} + (I - W_{jold})g(\lambda, d_j),$$

$$g(\lambda, d_j) = \exp(-d_j^2/\lambda) / \sqrt{2} \quad (6)$$

식(6)에서  $W_{jold}$ 는 기존의 연결 강도이고  $I$ 는 기준이 되는 입력 벡터의 좌표이다.  $g(\lambda, d_j)$ 는 x 좌표의 연결 강도 오차와 y좌표의 연결 강도 오차를 계산하여 가중치 벡터를 조정한다. 즉 연결 강도를 재조정한다. 식(6)을 통하여 조건에 만족되는 뉴런이 선택되어진 경우에는 입력 벡터에 가장 가까운 가중치 벡터를 가지는 뉴런을 승자 뉴런으로 선정하고, 그 값이 입력 벡터에 더 가까워지도록 하기 위해 가우시안 함수를 이용하여 선택되어진 뉴런과 그 이웃들의 연결 강도를 재조정한다. 이러한 과정을 뉴런의 개수만큼 학습시키게 되는데 이때 학습률의 초기 값을 0.95로 설정하고 이것에 0.08을 곱셈하는 연산을 가우시안 함수에 적용하여 지역 최소점에 위치할 수 있는 오류를 최소화하여 효율적으로 경로를 탐색하도록 한다.

#### IV. 실험 및 결과분석

본 논문에서 제안한 방법을 Intel Core(TM)2 Duo 2.66GHz CPU와 2GB RAM이 장착된 IBM 호환 PC 상에서 Java - NetBeans를 이용해 구현하여 실험하였다. 그림 3은 제안된 방법을 이용한 경로 탐색 결과이다.

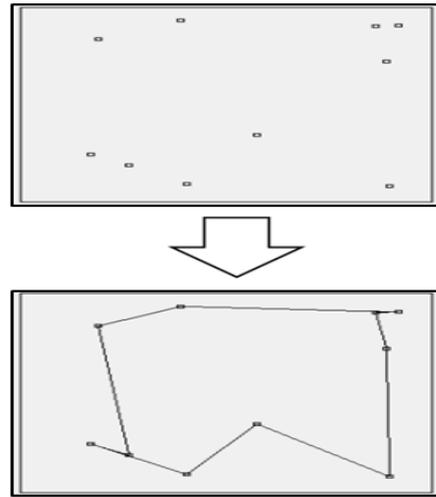
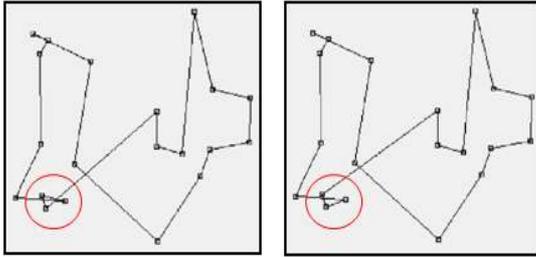


그림 3. 경로 탐색 결과

유전자 알고리즘을 이용한 기존의 경로 탐색 방법과 본 논문에서 제안된 경로 탐색 방법 간의 결과를 비교하여 그림 4로 나타내었다.

표 1은 기존의 경로 탐색 방법[1]과 제안된 경로 탐색 방법에서 탐색된 노드들의 좌표와 좌표가 가지는 index를 나타내었다. 표 2에서 알 수 있듯이 제안한 방법이 기존의 유전자 알고리즘을 이용한 경로탐색 방법보다 총 탐색 거리 값이 적어서 경로 탐색에 있어서 효율적인 것을 확인할 수 있다.



(a) 기존의 방법 (b) 제안된 방법  
 그림 4. 기존의 방법과 제안된 방법간의 경로 탐색 결과 비교

표 1. 기존 방법과 제안된 방법간의 경로 탐색 좌표와 index

기존의 방법		제안된 방법	
Index	좌표(x, y)	Index	좌표(x, y)
1	65, 221	1	65, 221
16	61, 209	13	84, 214
13	84, 214	14	35, 211
14	35, 211	10	60, 160
10	60, 160	7	59, 74
7	59, 74	4	67, 61
4	67, 61	9	52, 55
9	52, 55	8	109, 81
8	109, 81	12	93, 179
12	93, 179	11	175, 252
11	175, 252	15	217, 190
15	217, 190	6	226, 165
6	226, 165	3	265, 158
3	265, 158	18	266, 116
18	266, 116	17	229, 108
17	229, 108	5	211, 34
5	211, 34	19	199, 169
19	199, 169	2	174, 162
2	174, 162	20	174, 129
20	174, 129	16	61, 209

표 2. 유전자 방법과 제안된 방법 간의 총 탐색 거리 비교

유전자를 이용한 방법	제안한 방법
Index	
1 - 16 - 13 - 14 - 10 - 7 - 4 - 9 - 8 - 12 - 11 - 15 - 6 - 3 - 18 - 17 - 5 - 19 - 2 - 20 - 1	1 - 13 - 14 - 10 - 7 - 4 - 9 - 8 - 12 - 11 - 15 - 6 - 3 - 18 - 17 - 5 - 19 - 2 - 20 - 16 - 1
총 탐색 거리 값	
1233.2689958	1203.6391858

### V. 결 론

가장 우수한 노드를 다음 출발지 노드로 선정하는 기존의 유전자 알고리즘을 이용한 경로 탐색 방법은 거리 값에 대한 오차율이 발생하여 효율적인 거리 값을 도출하지 못하는 경우가 발생하였다. 따라서 기존의 방법보다 효율적인 경로를 탐색하기 위해 자율 학습방법인 SOM 알고리즘을 적용하였다. 제안하는 방법은 효율적인 경로를 탐색하기 위해 기존의 방법인 유전자 알고리즘을 적용하여 도출된 좌표를 뉴런화 한다. 그리고 뉴런화된 좌표에 가중치 벡터를 적용하여 모든 뉴런 간의 거리 가중치를 계산하였다. 계산된 거리 가중치를 이용하여 경로를 선택하는 조건을 검사 후, 조건에 만족하는 뉴런을 선택하였다. 선택된 뉴런을 기준으로 가우시안 함수를 이용하여 이웃 뉴런 간의 연결강도를 조정하였다. 이러한 과정을 뉴런의 개수만큼 학습하여 효율적인 경로를 도출하였다. 실험에서 알 수 있듯이 제안된 방법의 경로 탐색 방법이 기존의 방법보다 효율적인 경로를 탐색하는 것을 확인하였다.

향후 연구 방향은 제안된 방법과 최적 경로 탐색 기법으로 적용되는 A\* 알고리즘 및 볼츠만 머신과 비교 분석하고 최적의 경로를 탐색할 수 있도록 확장할 것이다.

### 참고문헌

- [1] 조현학, 조재현, 김광백, "유전자 알고리즘을 이용한 효율적인 경로 탐색," 한국컴퓨터정보학회 하계학술대회논문집, 18권, 2호, pp.417-419, 2010.
- [2] 조재훈, 이대중, 송창규, 김용삼, 전명근, "유전자 알고리즘과 정보이론을 이용한 속성 선택," 한국지능시스템학회 논문지, 18권, 1호, pp.94-99, 2008.
- [3] [http://www.aistudy.co.kr/neural/self\\_organizing\\_map.htm/](http://www.aistudy.co.kr/neural/self_organizing_map.htm/)
- [4] T. Kohonen, "The Self-organizing Map." Proc. of IEEE, Vol. 78, pp.1464-1480, 1990.
- [5] B. Angeniol, et al, "Self-organizing Feature Maps and the Travelling Salesman Problem," Neural Networks, Vol.1, pp.289-293, 1988.