

# 효율적인 점증적 진화학습을 위한 씨앗개체의 자동생성

송 금범, 조 성배  
연세대학교 컴퓨터과학과

## Automatic Generation of Seed Individuals for Efficient Incremental Evolutionary Learning

Geum-Beom Song and Sung-Bae Cho  
Computer Science Department, Yonsei University

### 요 약

시뮬레이션 환경이나 실제 환경에서 이동 로봇 제어를 진화 알고리즘으로 만들어내는 연구가 최근 활발하다. 이전의 연구에서는 기존의 단순한 진화 알고리즘이 환경에 제한된 제어를 만들어 내는 문제점을 해결하기 위한 방법으로 셀룰라 오토마타 기반 신경망의 점증적 진화방법을 제시하였다. 점증적 진화 방법은 초기에 간단한 행동으로 해결할 수 있는 환경에 맞도록 제어를 진화시킨 다음, 점차 복잡한 행동이 요구되는 환경에서 제어를 점증적으로 진화시킨다. 실험결과, 점증적 진화의 방법이 좀더 효율적으로 로봇을 진화시키고 환경의 변화에 보다 강한 것을 알 수 있었다. 그러나 이전연구에서의 점증적 진화 방법은 한 단계에서 진화가 끝난 후 다음 단계로 넘어갈 개체를 사람이 선택해야 하는 문제가 있었다. 본 논문에서는 이러한 문제점을 해결하기 위한 다양한 방법을 제시하고 실험을 통해 그 유용성을 보이고자 한다.

### 1. 서론

이동 로봇의 제어를 진화방식으로 구축하고자 하는 연구가 최근 활발히 진행되고 있는데, 많은 경우에 주어진 환경에서 유전자 알고리즘을 적용하여 제어를 진화시킨다. 그러나, 이러한 방법은 환경에 제한된 제어를 만들어 내는 경향이 있다. 이러한 문제점을 해결하기 위해, 이전연구[1]에서는 셀룰라 오토마타 기반 신경망의 점증적 진화방법을 제시한 바 있다. 이것은 초기에 간단한 행동으로 해결할 수 있는 환경에 맞도록 제어를 진화시킨 다음, 점차 복잡한 행동이 요구되는 환경에서 제어를 점증적으로 진화시키는 방법이다.

이전연구에서의 점증적 진화 방법은 한 단계에서 진화가 끝난 후 적절한 개체를 선택하여 전체 집단으로 복제하고, 돌연변이시켜 새로운 집단을 생성하여 다음 단계의 진화에 사용하였다. 실험결과, 이러한 방법은 이전단계에서 성공한 개체의 좋은 성질을 그대로 갖고 있기 때문에 다음 단계에서의 진화가 빨리 이루어질 수 있었다. 그러나 이러한 방법은 한 개체에 너무 의존하는 문제점 때문에 어떤 개체를 선택하느냐에 따라 결과가 달라질 수 있다. 본 논문에서는 이러한 문제점을 해결하기 위해 한 단계가 끝난 후 새로운 집단을 자동적으로 생성하는 방법을 제시한다. 또한, 실험 결과를 통해 그 유용성을 입증하고자 한다.

### 2. 방법

#### 2.1 셀룰라 오토마타 기반 신경망

이 신경망은 간단한 규칙들의 결합으로 복잡한 현상을 표현할 수 있는 셀룰라 오토마타의 장점을 이용한 모델로, 셀룰라 오토마타의

구성요소(상태, 주변, 규칙)에 신경망의 구조를 결정하는 염색체를 적용시켜 하나의 신경망을 만들어낸다[1]. 이러한 방법은 몇 개의 규칙들의 조합으로 복잡한 신경망을 만들어낼 수 있도록 한다. 즉, 셀룰라 오토마타 공간에서 염색체로 표현된 신경망의 구조가 유전자 알고리즘을 통해서 진화되어 최적의 구조를 찾아낼 수 있다. 이 모델은 셀룰라 오토마타의 상태, 주변, 규칙과 각 셀에 대응되는 염색체의 정보에 따라 여백, 뉴런, 축삭돌기, 수상돌기로 이루어진 신경망을 만들어낸다[1].

신경망을 실제 문제에 적용시키기 위해서는 주어진 신경망에 신호를 입력하여 결과를 얻어내야 한다. 입력과 출력은 격자형 셀룰라 오토마타 공간의 특정한 위치에 있는 셀을 통해 이루어지며, 입력과 출력이 정해지면 외부로부터 신호를 받아들이고 내보낸다. 이때 각 셀의 상태에 따라 역할이 달라지는데 수상돌기 셀은 주위의 셀로부터 신호를 모아 뉴런 셀에게 신호를 보내고, 뉴런 셀은 신호를 누적시켜 역치값보다 크면 축삭돌기 셀로 신호를 보낸다. 축삭돌기 셀은 뉴런으로부터 받은 신호를 자신의 주변 셀로 보내는 역할을 한다. 이러한 신호 입력력의 역할은 생물의 그것과 같다. 이에 대한 자세한 설명은 참고문헌 [1]을 참조하기 바란다.

#### 2.2 점증적 진화

기존의 진화 방법은 복잡하고 일반적인 문제를 해결하기 위해서는 많은 시간이 필요로 한다. 넓은 탐색공간에서 적절한 해결점을 찾는 것은 어려운 문제다. 하지만, 해결점을 찾기 위해 복잡한 문제를 좀더 쉬운 문제들로 나누어 탐색공간을 줄이면서 해결한다면 효율적인 탐색이 가능할 것이다. 즉, 우리가 해결하고자 하는 문제  $t$ 가 있을 때, 이 문제를 해결하기 위해  $t$ 보다 쉬운 문제  $t'$ 를 만들고, 다시  $t'$ 를 해

결하기 위해 좀더 쉬운 문제  $t'$ 를 만든다. 이제 문제  $t$ 를 해결하기 위해서는 먼저 문제  $t'$ 를 해결하기 위해 전체집단을 진화시키고, 이 집단을 다시 좀더 어려운 문제인  $t$ 에서 진화시키고, 마지막으로 문제  $t$ 를 해결해 나가는 방식으로 한다면, 좀더 효율적으로 문제를 해결할 수 있을 것이다.

점증적 진화의 방법을 정리하면, 해결하고자 하는 문제  $t$ 가 있을 때, 이를  $t$ 에서 파생된  $n$ 개의 문제 ( $t_1, t_2, t_3, \dots, t_n$ )로 나누어 해결한다. 이때  $t_i$ 는  $0 < i \leq n$ 인 모든  $i$ 에 대해서  $t_{i-1}$ 보다 쉬운 문제다. 문제  $t$ 를 해결하기 위해서는 먼저 집단을  $t_1$  문제에서 진화시킨 후, 다시  $t_2$  문제에서 진화시킨다. 이와 같은 방법으로 결국 해결하고자 하는 문제인  $t_n$ 을 해결할 수 있다[2, 3].

본 논문에서는 점증적 진화의 방법을 이용하여 여러 환경에서 장애물과 충돌하지 않고 잘 움직일 수 있는 신경망 제어기를 구축하고자 한다. 아무리 복잡한 환경이라도 직진(그림 1(a)와 (b)), 우회전(그림 1(c)와 (d), (e)), 좌회전(그림 1(f)와 (g)) 등의 움직임으로 벽과 충돌하지 않고 움직일 수 있기 때문에 본 논문에서는 이러한 기본 행동을 위한 환경에서 점증적으로 진화시키고자 한다.

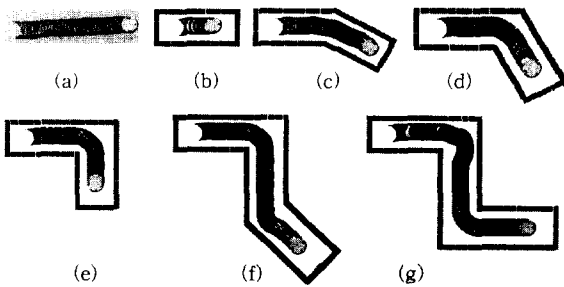


그림 1. 각 환경에서 성공한 로봇의 경로.

### 2.3 확률분포를 이용한 집단생성

점증적 진화방법을 이용할 때 한 단계에서 진화가 끝난 후 어떻게 다음 단계에서 사용할 집단을 만드는가 하는 문제는 중요하다. 이전의 방법은 성공한 개체중 적절한 것을 선택하여 전체집단으로 복제하고 돌연변이시켜 새로운 집단을 생성했다. 그러나, 이러한 방법은 이전 단계에서 성공한 개체의 좋은 성질을 그대로 가질 수 있는 반면에 전체 집단이 한 개체에 너무 의존하는 문제점이 있다.

F. Gomez[3]는 점증적 진화에 유전자 알고리즘의 지역극소 문제를 해결하기 위한 델타코딩[4] 방법을 도입하여 문제를 해결하였다. 본 논문에서는 이러한 방법을 도입, 변형시켜 각 단계에서 성공한 개체를 하나만 선택하지 않고, 적합도의 크기에 따라 여러 개를 선택한 후, 이들로부터 새로운 집단을 생성한다. 새로운 개체를 만들 때는 선택한 개체를 코오시 확률분포에 따라 돌연변이 횟수를 조정함으로써 공간상에서 염색체들이 적절히 분포되도록 하였다. 코오시 확률분포는 다음과 같다.

$$f(x) = \frac{a}{\pi(a^2 + x^2)} \quad (1)$$

여기에서  $\pi$ 는 원주율이고,  $f$ 는  $\pm a$  구간안에 50%의 확률로 값을 갖는다. 코오시 확률분포는 가운데에 집중되어 있으면서도 양끝에도 값을 갖고 있기 때문에 공간상에서 염색체들이 적절한 분포를 갖도록 한다. 즉, 공간상에 최고 적합도를 갖는 근처에 많은 개체를 갖게 하

여 좀더 나은 개체를 탐색하도록 하고, 또 약간 벗어난 곳에서도 분포하도록 하여 개체의 다양성을 더해준다. 그림 2는 식(1)에서  $a$ 값이 0.5일 때의 확률분포를 보여주고 있다. 본 실험에서는 그림 2와 같은 확률 분포를 이용하여 돌연변이의 횟수를 결정한다.

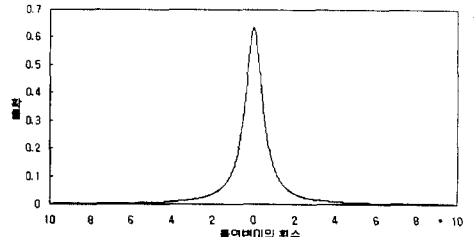


그림 2.  $a=0.5$ 일 때의 코오시 확률분포에 의한 돌연변이의 수.

## 3. 실험 결과

### 3.1 환경

5x5 크기의 셀룰라 오토마타 공간을 주어진 문제에 적용시켰으며 로봇의 입력센서 8개중 4개만을 사용했다. 각각의 입력은 정육면체 셀룰라 오토마타공간의 위와 아래의 면을 제외한 네 옆면을 자신의 입력 영역으로 하며, 출력 셀은 정육면체 셀룰라 오토마타 공간의 윗면과 아랫면에 있는 가운데 셀로 하였다. 먼저 염색체에 의해 만들어진 신경망에 시뮬레이터로부터 센서 값을 입력한다. 입력받은 센서 값은 신경망의 입력이 되고 신호의 전달을 통해 얻어진 출력 값은 로봇의 바퀴 속도가 된다. 로봇의 학습에 중요한 적합도 측정은 로봇의 위치가 목표지점과 얼마나 가까운가와 로봇이 직진 행동을 얼마나 많이 하였는가로 측정하였다.

시뮬레이션 방법은 먼저 직진을 유도하는 환경에서 제어기를 진화시킨 후 성공한 개체의 염색체를 전체 집단으로 복제시킨 다음, 직진과 우회전을 필요로 하는 환경에서 진화시킨다. 이 환경에서 성공한 개체는 다시 직진, 우회전, 좌회전을 필요로 하는 환경에서 진화시킨다. 이러한 방법은 유전자 알고리즘에서 탐색할 공간을 축소시켜 가며 제어기를 만들어 내는 것으로 좀더 효율적인 진화를 이루어 낼 수 있으리라 기대된다.

### 3.2 결과분석

한 단계에서 다음단계로 넘어갈 때는 2.3에서 언급한 방법으로 새로운 집단을 생성한다. 그러나 이전 단계에서 성공한 개체 중에서 몇 개의 개체를 선택할 것인가의 문제는 여전히 남아있다. 각 단계에서 성공한 개체의 수는 일정치 않은데 너무 많으면 공간상에서 성공한 개체를 적절히 분포시켜 가장 좋은 개체를 찾는다는 초기 목적에 맞지 않으므로 최대 수를 10개와 20개로 제한하였다.

그림 3은 점증적 진화방법을 사용하지 않고, 그림 1(g)환경에서 진화시켰을 때의 적합도의 변화를 보여주며, 그림 4는 이전연구와 같은 수동방법을 사용했을 경우 적합도의 변화를 보여주고 있다. 그림 5는 선택하는 개체의 최대수를 10개로 제한하였을 경우 각 단계별 적합도의 변화를 보여주고 있다. 그림 6은 선택하는 개체의 최대수를 20개로 제한하였을 경우 적합도의 변화를 보여주고 있다. 그림 7은 위의 두 실험에서 성공한 로봇의 신경망의 구조를 보여주고 있다.

실험결과, 점증적 진화의 방법을 사용하지 않았을 경우(그림 3) 100세대 가까이 지나도 성공한 개체를 찾을 수 없었지만, 점증적 진

화의 방법을 사용하였을 경우(그림 4, 5, 6) 성공한 개체를 얻을 수 있었다. 또한 수동방법(그림 4)보다 새로 제안한 자동방법(그림 5, 6)으로 더 빨리 성공한 개체를 찾을 수 있었다. 그림 8은 성공적으로 진화된 개체를 좀더 어려운 환경에 적용시켰을 때의 결과를 보여주고 있다. 본 논문에서 제시된 점증적 진화의 방법으로 생성된 개체가 직선과 좌회전 및 우회전의 행동이 필요한 환경에서는 어느 정도 적응성이 있음을 알 수 있다.

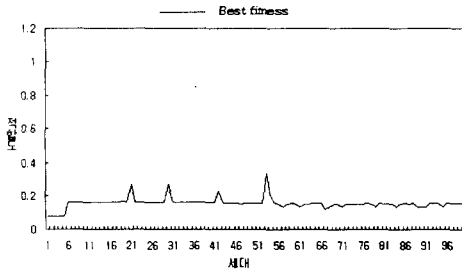


그림 3. 점증적 진화방법을 사용하지 않았을 때의 적합도 변화.

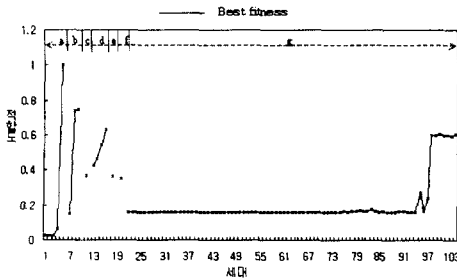


그림 4. 수동방법에서의 적합도 변화.

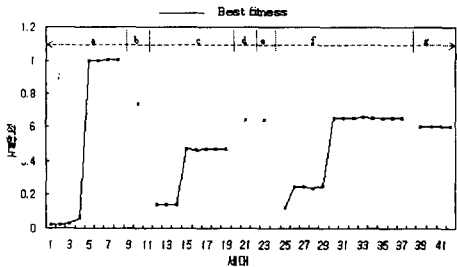


그림 5. 최대 개체수를 10개로 제한한 실험의 적합도 변화.

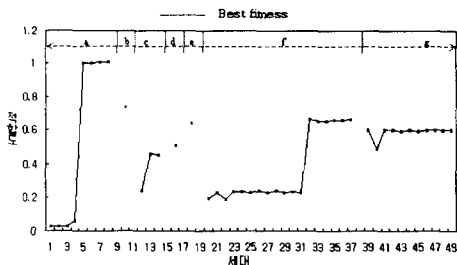


그림 6. 최대 개체수를 20개로 제한한 실험의 적합도 변화.

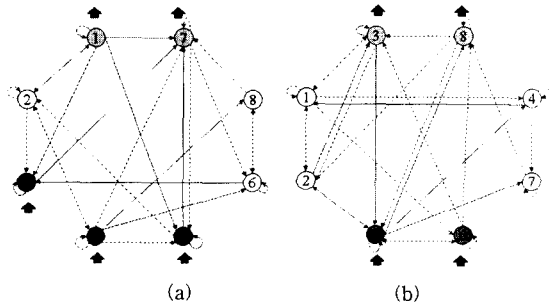


그림 7. 성공한 로봇의 신경망 구조. 실선: 흥분성 연결, 점선: 억제성 연결. (a) 최대 개체수를 10개로 제한한 실험. (b) 최대 개체수를 20개로 제한한 실험.

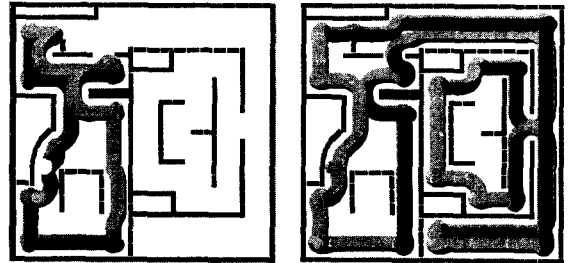


그림 8. 다른 환경에서의 적용. (a) 최대 개체수를 10개로 제한한 실험. (b) 최대 개체수를 20개로 제한한 실험.

#### 4. 결론 및 토의

이전 연구는 기존의 진화방법을 보완할 수 있는 점증적 진화의 방법에 대한 연구였다. 본 논문에서는 이를 좀더 보완하여 다양한 방법의 점증적 진화방법을 제시하였다. 또한 셀룰라 오토마타 기반의 진화형 신경망을 제시한 방법으로 실험하여 그 결과를 보여주었다. 그 결과 기존의 진화방법보다 주어진 문제를 빨리 해결할 수 있음을 보았고 이를 다른 환경에 적용시켜서 좋은 결과를 얻을 수 있었다.

#### 참고 문헌

- [1] 송 금범, 조 성배, "셀룰라 오토마타 기반 신경망의 점증적 진화에 관한 연구," 한국 정보과학회 추계 학술발표 논문집(B), Vol. 25, No. 2, pp. 348-350, 1998.
- [2] I. Harvey, P. Husbands and D. Cliff, "Seeing the light: Artificial evolution, real vision," *Proc. of 3rd Int. Conf. on SAB94*, pp. 392-401. MIT Press/Bradford Books, 1994.
- [3] F. Gomez and R. Miikkulainen, "Incremental evolution of complex general behavior," *Adaptive Behavior*, Vol. 5 Issue 3-4, pp 317-342, 1997.
- [4] K. E. Mathias and L. D. Whitley, "Initial performance comparisons for the delta coding algorithm," *Proc. IEEE Conf. Evolutionary Computation*, Vol. 1, pp. 433-438, 1994.