

## 유전자 프로그래밍을 이용한 자율 이동 로봇군의 협조행동 진화

조동연, 장병탁  
서울대학교 컴퓨터공학과

### Evolving Cooperative Behavior of Autonomous Mobile Robots Using Genetic Programming

Dong-Yeon Cho, Byoung-Tak Zhang  
Dept. of Computer Engineering, Seoul National University

**Abstract** - Many multiagents cooperative problems, such as table transport problem, require several emergent behaviors and a proper coordination of these is essential for successful accomplishment of the task. We study in this paper the genetic programming method, called fitness switching, to evolve cooperation strategies of robots in these kind of tasks and show simulation results to demonstrate its effectiveness.

#### 1. 서론

자율 이동 로봇 하나 하나는 단순 하지만 이들이 잘 협동하도록 한다면 혼자서 할 수 없는 일을 수행하도록 할 수 있고, 어떤 일들은 더 효율적으로 수행하도록 할 수 있다. 이러한 자율 이동 로봇군이 다양한 환경에서 잘 동작하려면 스스로 그 환경에 적응할 수 있는 능력이 있어야 한다. 따라서 자율 이동 로봇군의 협조 행동 또한 중앙 집중적 제어에 의하여 이루어지기 보다는 각 로봇들이 자신의 역할을 수행함으로써 창발적으로 이루어져야 한다.

이러한 자율 이동 로봇군의 협조 행동을 위한 제어 프로그램을 생성하기 위해 유전자 프로그래밍이 사용되어 왔다. 인공 개미가 먹이를 모으는 행동을 하도록 하는 연구도 있었고 [1, 3], 피식자와 포식자의 문제를 해결하는 시도도 있었다 [4]. 또한 정해진 영역을 협조적으로 탐사하기 위해 서로 다른 3개의 방법 (homogeneous, heterogeneous, and coevolution)이 제안되기도 하였다. [2]

이러한 작업들의 대부분은 기본적인 행동들로부터 직접 창발적인 집단 행동을 진화 시키는 것을 목표로 하였다. 그러나 더 실제적이고 복잡한 문제는 하나 이상의 창발적 행동을 필요로 하고 작업을 성공적으로 수행하기 위해서는 이러한 행동들의 적당한 조정이 필요하다.

예를 들면, 하나의 이동 로봇이 운반하기에는 너무 무거운 탁자를 여러대의 로봇이 협조하여 목표지점으로 운반하는 문제는 연속적으로 달성되어야 하는 두가지의 창발적 행동(귀환과 때 짓기)을 필요로 한다. 다시 말하면 처음에 로봇들은 운반해야 할 물체 주변으로 모여야 하고(귀환 - homing), 그 물체를 협력하여 목표 지점으로 운반해야 한다(때 짓기 - herding).

유전자 프로그래밍을 이용하여 이 문제를 성공적으로 해결하려면 제어 프로그램은 어떠한 방식으로든 귀환 행동과 때 짓기 행동이 협력하도록 해야 한다. 본 논문에서는 이러한 문제를 해결하기 위해 제시된 적합도 교체 (fitness switching) 방법을 살펴보고 [5], 시뮬레이션을 통해 장애물이 존재하는 환경에서 자율 이동 로봇군의 협조 행동이 이루어짐을 보인다.

논문의 구성은 다음과 같다. 2절에서는 탁자 운반 문제를 정의하며, 3절에서는 적합도 교체 방법을 설명한다. 4절에서는 실험 환경과 실험 결과를 보이고, 5절에

서 결론을 맺는다.

#### 2. 탁자 운반 문제

본 논문에서 다루는 탁자 운반 문제는 다음과 같이 정의된다.  $n \times n$ 의 격자 공간에 하나의 탁자와 4대의 로봇이 임의로 위치한다. 또한 특정한 부분이 목표 지점으로 지정된다. 그리고 로봇의 움직임을 방해하는 일정한 개수의 장애물들이 하나의 격자를 차지하면서 임의로 배치된다. 로봇들의 목표는 서로 협력하여 탁자를 목표 지점으로 운반하는 것이다. 탁자가 너무 무겁고 커서 하나의 로봇은 운반할 수 없으므로 로봇들은 무리를 지어 움직여야 한다.

각 로봇은 기본적인 행동 양식을 가지고 있으며, 8 방향(N, E, S, W, NE, SE, SW, NW)으로 이동이 가능하다. 로봇은 선택된 행동에 따라 현재의 방향으로 이동하거나, 위치는 유지하고 방향만 바꿀 수도 있으며, 아무 행동도 하지 않고 제자리에 머물러 있을 수 있다. 또한 이동 방향은 장애물이나 다른 로봇과의 충돌을 피하면서 선택될 수 있다. 로봇의 감지 범위는 이동 방향과 같은 방향의 격자 1 만큼으로 제한된다. 로봇들은 다른 로봇들을 인식할 수 있으며, 그것들과 장애물들을 구별할 수 있다.

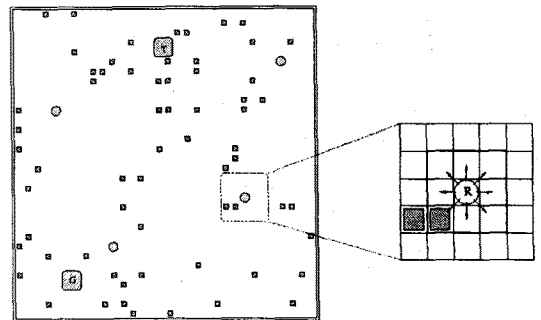


그림 1. 탁자 운반 문제의 환경과 로봇의 이동 및 감지 범위

#### 3. 적합도 교체

적합도 교체는 유전자 프로그래밍으로 복잡한 행동을 진화시키는 방법이다. 이것은 다음과 같은 점진적 학습 과정에 기반을 둔다.

1. 주어진 문제를 해결하는데 기여하는 세부 행동들을 정의한다.  
 $B = \{B_1, B_2, \dots, B_n\}$
2. 각 세부 행동들에 대한 적합도 함수를 정의한다.  
 $F = \{f_1, f_2, \dots, f_n\}$

3. 목표 행동을 달성하기 위한 세부 행동들의 순서를 정한다.

$$S_t = S_{t-1} \cup \{B_t\} \quad t=1, \dots, n, S_0 = \{\}$$

또한 이에 상응하는 적합도 함수는 다음과 같이 정의된다.

$$F_t = F_{t-1} + f_t = \sum_{i=1}^t f_i \quad t=1, \dots, n, F_0 = 0$$

4. 루트 노드 아래에  $n$ 개의 서브 트리를 갖는 유전자 프로그램의 구조를 정의한다.

5.  $S_t$ 를  $t=1, \dots, n$ 의 순서로 진화 시키기 위해 유전자 프로그래밍을 적용한다. 전체 트리 중 처음  $t$ 개의 서브 트리를 진화시키기 위해 각  $S_t$ 에 대하여 적합도 함수  $F_t$ 가 사용된다.

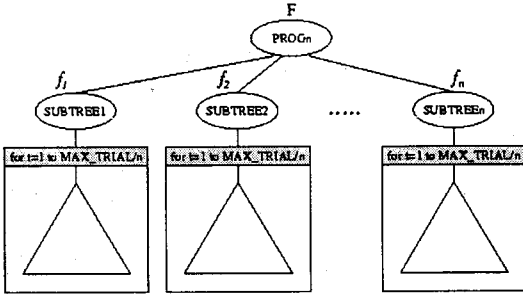


그림 2. 적합도 교체의 전체적인 구조

이 적합도 교체 방법을 탁자 운반 문제에 적용하면, 귀환 행동(homing)과 떼 짓기 행동(herding)의 연속적인 수행이 된다. 따라서 이 문제에 대한 세부 행동들의 집합은  $B = \{B_1, B_2\}$ 이고, 여기서  $B_1 = \text{homing}$ ,  $B_2 = \text{herding}$ 이 된다. 적합도 함수의 집합도  $F = \{f_1, f_2\}$ 가 되고,  $f_1$ 은 탁자로 모이는 행동에 대한 적합도 함수가 되고,  $f_2$ 는 탁자를 목표 지점으로 운반하는 행동에 대한 적합도 함수가 된다.

실험에서  $f_1$ 과  $f_2$ 는 다음과 같이 정의된다.

$$f_1 = \sum_{r=1}^n \{c_1 \max(X_r, Y_r) + c_2 S_r + c_3 C_r - c_4 M_r + K\}$$

$$f_2 = \sum_{r=1}^n \{c_1 \max(X_r, Y_r) + c_2 S_r + c_3 C_r - c_4 M_r + c_5 A_r + K\}$$

여기서  $X_r, Y_r$ 은 각각 목표 지점과 로봇  $r$ 의 출발점간의  $x$ 축,  $y$ 축 상의 거리를 나타낸다. ( $f_1$ 에서의 목표지점은 탁자이고,  $f_2$ 에서의 목표 지점은 탁자가 놓여질 지점이다.)  $S_r$ 은 로봇  $r$ 이 이동한 회수이고,  $C_r$ 은 로봇  $r$ 이 장애물, 벽 또는 동료 로봇과 충돌한 회수이다. 또한  $M_r$ 은 로봇  $r$ 의 처음 위치와 최종 위치 사이의 거리이며,  $A_r$ 은 다른 로봇으로부터 멀어지는 경우에 받은 벌점이다. 그리고  $c_i$ 는 실험적으로 결정된 계수이고  $K$ 는 양의 상수이다.

이러한 적합도 교체를 구현하는데는 세가지 방법이 있을 수 있다. 첫째는 기존의 유전자 프로그래밍 연구에서 대부분 사용하고 있는 기본적인 방법(naive evolution)이다. 여기서는 세부 행동에 대한 서브 트리의 구별이 없이 전체 트리가 여러 세부 행동에 의하여 공유된다. 즉, 전체 트리에 대하여  $f_1$ 과  $f_2$ 가 모두 적용된다. 따라서 기억 장소를 줄일 수 있다는 장점은 있으나 각 세부 행동을 정확하게 표현할 수 없게 된다.

적합도 교체의 두 번째 구현 방법은 순차적인 진화이다(sequential evolution). 이 방법에서는 왼쪽의 서브 트리가 첫 번째 행동을 책임지고, 오른쪽 서브 트리가 두 번째 행동을 책임진다. 유전자 프로그래밍을 한번 실행하여 처음 행동에 대한 왼쪽 서브 트리를 진화시킨 후, 이 중 가장 좋은 프로그램을 두 번째 행동을 진화 시키기 위한 유전자 프로그래밍의 실행에 사용한다.

인자	값
함수 집합	IF-OBSTACLE, IF-ROBOT, IF-TABLE, IF-GOAL, PROG2, PROG3
터미널 집합	FORWARD, AVOID, STOP, TURN-TABLE, TURN-GOAL, RANDOM-MOVE
실험 대상 실험 공간	훈련 대상 20개, 검사 대상 20개 32 × 32 격자 공간, 1개의 탁자, 64개의 장애물
개체군의 크기	100
최대 세대수	200
교차 확률	1.0
돌연 변이 확률	0.1
트리의 최대 깊이	10

표 1. 탁자 운반 문제의 실험에 대한 설정

다. 이러한 접근법은 독립적인 세부 행동들의 연속으로 나누어 질 수 있는 작업을 해결하기 위한 가장 분명한 방법이지만 세부 작업의 수만큼 유전자 프로그래밍을 실행해야하고 앞 단계의 수행 결과를 계속 유지해야 하므로 시간이 오래 걸리고 절차가 복잡하다.

세 번째 방법으로, 공진화(coevolution)에 의한 적합도 교체를 생각할 수 있다. 공진화적인 교체는 서브 트리 서로 다른 세부 행동들을 책임진다는 점에서는 순차적인 진화와 비슷하다. 그러나 적합도의 교체가 한 세대에서 일어난다는 것이 가장 큰 차이점이다. 즉 왼쪽 서브 트리의 적합도를  $f_1$ 에 의하여 계산하고, 오른쪽 서브 트리의 적합도를  $f_2$ 에 의하여 계산하여 전체 프로그램의 적합도는  $f_1$ 과  $f_2$ 의 합으로서 정의한다. 이 방법의 장점은 기본이 되는 세부 행동들의 진화가 동시에 이루어진다는 것이다.

#### 4. 실험 방법 및 결과

유전자 프로그래밍에서 사용한 함수 집합은 6개의 원소로 구성된다. IF-OBSTACLE은 감지 범위 내에 장애물의 존재 여부를, IF-ROBOT은 다른 로봇의 존재 여부를 판단한다. IF-TABLE과 IF-GOAL은 각각 탁자와 목표 지점을 감지하기 위해 사용된다. PROG2 (PROG3)는 두(세) 개의 서브 트리를 연속적으로 수행한다. 터미널의 집합도 역시 6개의 원소로 구성된다. FORWARD는 현재의 방향으로 한 스텝 이동한다. 이것은 충돌이 일어날 가능성이 있다. 여기서 탁자를 들어 올리는 행동은 움직이는 동작에 포함되었다고 가정한다. AVOID는 주변의 영역을 검사하여 충돌을 피하는 첫 번째 방향으로 움직인다. 검사는 현재 방향으로부터 시계방향으로 이루어진다. RANDOM-MOVE는 임의의 방향으로 움직인다. 이것도 역시 충돌이 일어날 가능성이 있다. TRUN-TABLE과 TRUN-GOAL은 탁자 또는 목표 지점과 가장 가까운 지점으로 시계방향으로 회전한다. STOP은 움직이지 않고, 현재 위치에 머물러 있다.

위와 같이 정의된 함수와 터미널의 집합으로부터 초기 개체군은 임의로 생성된다. 각 개체를 로봇의 제어 프로그램으로 사용하여 탁자 운반 문제에 적용한 후 3절에서 설명한 방법으로 적합도를 계산한다. 개체군에서 상위 50%가 부모로서 선택될 가능성이 있다. 서브 트리 교차와 노드 돌연변이에 의하여 두 자손을 생성하기 위해 부모 두 개를 선택한다. 각 연산자가 정해진 확률에 따라 적용된다. 이 과정을 새로운 개체군이 만들어질 때까지 반복한다. 모든 자손들이 다 만들어진 후에 부모 중에서 가장 좋은 두 개체로 임의의 두 개체를 대체한다(elitism). 이것이 완전한 한 세대이고 이 과정을 정해

진 세대 수만큼 반복한다. 다른 실험의 설정은 표 1에 나와 있다.

그림 3은 공진화 방법으로 구현한 적합도 교체를 적용한 경우에 세대에 따른 적합도의 변화를 보인다. 적합도 값이 급격하게 감소 하는 것으로 보아 협조 행동이 빠르게 향상되고 있음을 알 수 있다.

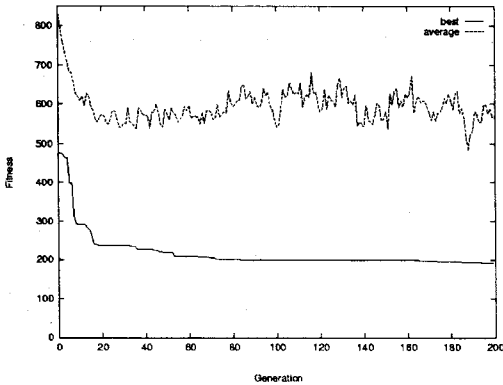


그림 3. 공진화 방법을 적용한 경우의 세대에 따른 적합도의 변화

공진화에 의한 적합도 교체를 사용한 유전자 프로그래밍은 하나 이상의 환경에 대한 탁자 운송 문제를 해결할 수 있다. 그림 4는 20개의 훈련 대상 중 2개에 대한 가장 좋은 적합도 값을 갖는 제어 프로그램을 사용한 로봇의 움직임 모습을 보여준다.

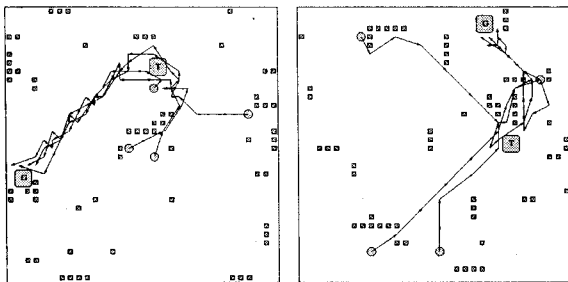


그림 4. 훈련 환경에서의 로봇의 이동 경로

진화된 프로그램의 일반성을 검증하기 위해 훈련 대상과는 다른 검사 대상에서 로봇을 작동시켜 본다. 그림 5는 20개의 검사 대상 중 2개에 대한 로봇의 움직임을 보여준다.

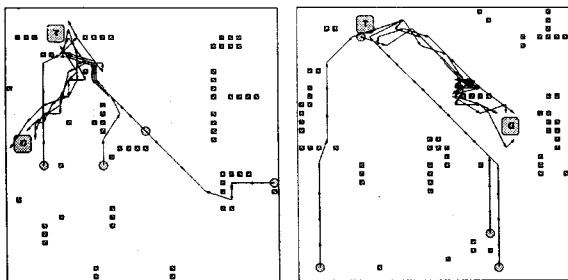


그림 5. 검사 환경에서의 로봇의 이동 경로

유전자 프로그램의 성능은 모든 환경에 대하여 목표 지점에 성공적으로 도달한 회수로서 측정될 수도 있다. 그림 6은 공진화 방법을 사용하는 경우에 세대에 따른 성공 회수의 변화를 보이고 있다.

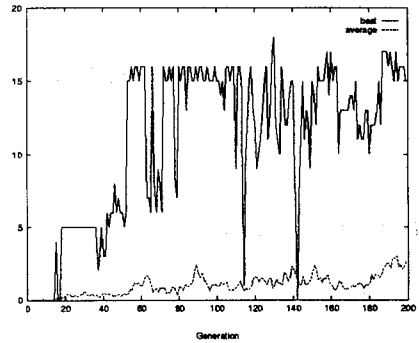


그림 6. 공진화 방법을 적용한 경우의 세대에 따른 성공 회수의 변화

표 2는 3절에서 설명한 세가지의 적합도 교체 방법에 대한 성공 비의 비교를 나타낸다. 기본적인 방법은 이 문제를 거의 해결하지 못했다. 순차적 진화에 의한 적합도 교체는 생각했던 대로 가장 높은 성공 비를 보인다. 공진화에 의한 적합도 교체는 순차적 진화에 의한 적합도 교체와 견줄만한 성공비를 보인다.

방법	성공 비	
	훈련 (20)	검사 (20)
Naive	0.05 (1)	0.15 (3)
Sequential	0.90 (18)	0.75 (15)
Coevolution	0.85 (17)	0.65 (13)

표 2. 성공 비의 비교

## 5. 결 론

무거운 탁자를 운반하는 자율 이동 로봇군의 협조 행동 진화를 위해 적합도 교체를 적용한 유전자 프로그래밍을 사용하여 제어 프로그램을 자동으로 생성하였다. 실험 결과 공진화에 의한 적합도 교체는 훈련 환경과 검사 환경에서 잘 정의된 순차적인 적합도 교체와 견줄 만한 성능을 보였다.

감사의 글 - 본 연구는 한국과학재단의 연구비에 의하여 일부 지원 되었음(과제 번호 96-0102-13-01-3, 981-0920-350-2)

## (참 고 문 헌)

- [1] F.H. Bennett III, "Automatic creation of an efficient multi-agent architecture using genetic programming with architecture-altering operations", *Proc. of First Int. Conf. on Genetic Programming*, PP. 30-38, 1996
- [2] H. Iba, "Multi-agent learning for a robot navigation task by genetic programming", *Proc. of Second Int. Conf. on Genetic Programming*, PP. 195-200, 1997
- [3] J.R. Koza, "Genetic Programming: On the Programming of Computers by Means of Natural Selection", 1992
- [4] T. Haynes, S. Sen, D. Schoenfeld, and R. Wainwright, "Evolving a team", *Proc. AAAI-95 Fall symposium on Genetic Programming*, PP. 23-30, 1995
- [5] B.T. Zhang and D.Y. Cho, "Fitness switching: Evolving complex group behaviors using genetic programming", *Proc. of Third Int. Conf. on Genetic Programming*, PP. 431-438, 1998