

유전자 프로그래밍에 의한 로봇 행동 전략과 행동 제어기의 공진화

조동연, 장병탁
서울대학교 컴퓨터공학과
{dycho, btzhang}@scail.snu.ac.kr

Coevolution of Robot Action Strategies and Action Controllers Using Genetic Programming

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

요 약

유전자 프로그래밍을 사용하여 로봇을 제어하는 기존의 방법들은 명시적인 제어 전략을 진화시키거나 센서의 입력으로부터 직접 제어 신호를 생성하는 행동 제어기를 진화시키고자 하였다. 본 논문에서는 로봇의 행동 전략과 행동 제어기를 동시에 진화시킴으로써 두 가지 방법의 장점을 결합하는 방법을 제시하고, 탁자 운반 문제에 적용하여 기존의 두 방법과 비교한다. 실험 결과 제안된 방법이 더 적은 계산 시간을 사용하면서도 더 좋은 성능을 나타내었다.

1. 서 론

로봇은 주변 환경의 상태를 감지할 수 있는 다수의 센서들을 장착하고 있으며 이 상태에 따라 수행할 수 있는 일련의 행동 양식을 갖고 있다. 로봇을 제어하는 작업의 궁극적인 목적은 환경의 상태를 파악하여 주어진 목적을 달성하도록 적절한 행동을 수행하게 하는 것이다. 본 논문에서는 유전자 프로그래밍을 이용하여 로봇이 이러한 성공적인 제어 전략을 학습할 수 있는 방법을 제안한다.

유전자 프로그래밍(genetic programming)은 자연 선택과 유전학에 기반하여 주어진 상황에 가장 적합한 컴퓨터 프로그램을 자동적으로 발견하는 방법이다 [3]. 그러므로 유전자 프로그래밍은 사람이 프로그래밍하는 것만으로는 가능한 모든 상황에 대하여 충분히 고려할 수 없는 환경에서 로봇의 제어 프로그램을 개발할 수 있는 유용한 방법을 제공한다.

유전자 프로그래밍을 사용하여 로봇을 제어하는 기존의 방법들은 제어 프로그램의 특성에 따라 크게 두 가지 종류로 나눌 수 있다. 첫 번째는 로봇이 현재의 환경을 판단하여 그 상태에 따라 적절한 행동을 취하도록 하는 것이다. 이 방법에서는 로봇이 환경을 판단하는 기준과 로봇이 취할 수 있는 행동이 미리 정해져 있다. 기존의 많은 연구들이 이 방법을 사용하고 있으며 [1], [2], [4], [6], 명시적인 제어 전략을 생성할 수 있지만 세부적인 행동 제어를 고려하지 않기 때문

에 실 세계를 정확하게 반영하지 못하는 문제가 있다. 두 번째는 센서의 출력값에 따라 제어 신호를 직접 생성하여 로봇을 제어하는 것이다 [5]. 이 방법에서 로봇은 현재 상태에 대한 판단 없이 센서의 출력에 직접 반응한다. 이것은 로봇이 빨리 행동을 취하도록 할 수 있으나, 행동을 계획하는 부분이 없으므로 장기적인 행동을 하기 어렵다.

본 논문에서는 이 두 가지 방법의 장점을 결합하여 더 효과적인 제어 프로그램을 생성하는 방법을 제안한다. 이 방법에서는 두 개의 개체군이 존재하는데, 하나는 행동 전략을 위한 개체군이고 다른 하나는 행동 제어기를 위한 개체군이다. 이 두 개체군을 공진화시켜 로봇을 더 효과적으로 제어하는 프로그램을 생성하게 된다.

논문의 구성은 다음과 같다. 2절에서는 로봇이 수행하는 작업과 동작 환경 및 로봇의 구조를 설명하며, 3절에서는 행동 전략과 행동 제어기를 공진화시키는 방법을 기술한다. 4절에서는 실험 결과를 보이고, 5절에서 결론을 맺는다.

2. 탁자 운반 문제와 로봇의 구조

본 논문에서 다루는 탁자 운반 문제는 $n \times n$ 의 실수 공간에 놓여진 2×2 크기의 탁자와 4대의 로봇으로 구성된다. 여기에 역시 2×2 크기의 목표 지점이 설정된다. 이 구성 요소들의 초기 위치는 임의로 선택된다. 로봇의 이동을 방해하는 다양한 형태의 장애물들이 역시 임의의 위치에 놓여진다 (그림 1). 로봇의 임무는 탁자를 목표 지점으로

운반하는 것이다. 탁자는 너무 무겁고 커서 하나의 로봇이 운반할 수 없으므로 로봇은 무리를 지어 움직여야 한다.

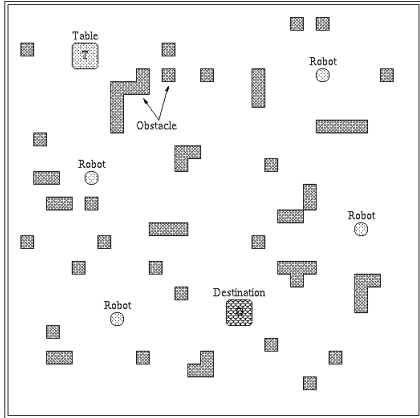


그림 1. 탁자 운반 문제의 환경

각 로봇은 4가지 종류(O : 장애물 감지, R : 다른 로봇 감지, T : 탁자와의 거리 측정, D : 목표 지점과의 거리 측정)의 센서에 대하여 각 8개씩 총 32개의 센서를 갖고 있다 (그림 2). 모든 로봇은 동일한 구조로 되어 있으며, 같은 행동 전략과 행동 제어기에 의하여 작동된다.

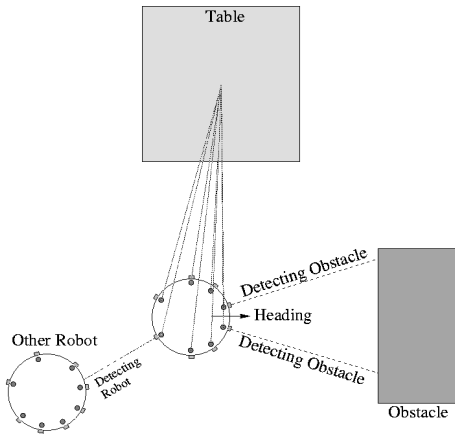


그림 2. 로봇에 장착된 센서의 기능

3. 행동 전략과 행동 제어기의 공진화

행동 전략의 개체군 A와 행동 제어기의 개체군 B가 존재하고 각 개체군의 크기는 각각 N_p , N_c 이다.

$$A = \{A_i\}, \quad (i=1,2,\dots,N^p)$$

$$B = \{B_j\}, \quad (j=1,2,\dots,N^c)$$

각 개체군은 일반적인 유전자 알고리즘(적합도 평가, 선택, 유전 연산자 적용)에 의하여 진화되나, 가장 큰 차이점은 행동 제어기의 적합도 평가가 행동 전략의 적합도 평가에 내포되어 있다는 것이다. 각 행동 전략 A_i 는 터미널 심볼에 의하여 정의된 기본 행동을 갖고 있다. 이 기본 행동을 수행하기 위해 개체군 B에 존재하는 모든 행동 제어기 B_j 의 출력을 평가하여 가장 좋은 적합도 값을 갖는 제어기의 출력을 실행한다 (그림 3).

탁자 운반 문제는 두 개의 세부 행동, 즉 탁자로의 귀환(homing)

과 목표 지점으로의 떼 짓기(herding)가 순차적으로 수행 되어야 하므로 기존의 유전자 알고리즘으로는 이 문제를 해결 하기 어렵다. 이전의 연구에서 [7], 우리는 이러한 문제를 해결하기 위해 적합도 교체 (fitness switching) 방법을 제안하였다. 이 방법에서는 각 세부 행동에 대한 적합도 함수가 정의되며 각 세부 행동에 대한 프로그램을 동시에 진화시키기 위해 한 세대 내에서 적합도의 교체가 이루어진다. 여기서 정의되는 행동 전략에 대한 적합도 함수는 로봇과 목표물과의 거리, 출발 지점과 도착 지점간의 거리, 로봇의 이동 회수와 충돌 회수, 다른 로봇과의 거리 등을 포함한다. 이에 대한 세부적인 내용은 이전 연구에 자세히 기술 되어 있다 [7].

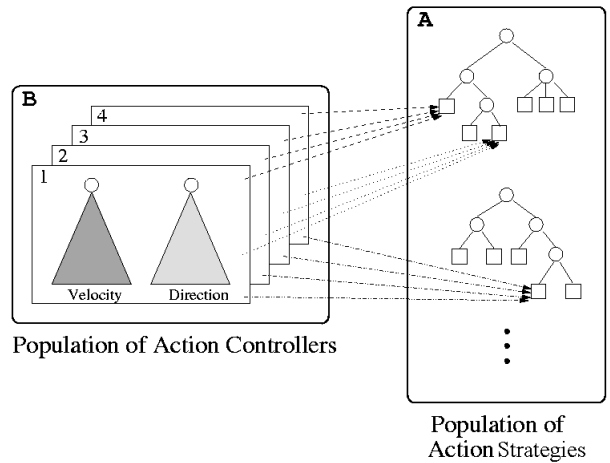


그림 3. 행동 전략과 행동 제어기의 공진화

행동 제어기에 대한 적합도 함수는 다음과 같이 정의된다.

$$F_j^a = f_j^v + f_j^d$$

$$f_j^v = \frac{\sum_{i=1}^{N^p} \text{가장 좋은 속도로 평가된 총 회수}}{A_i \text{에서 } B_j^v \text{를 실행한 총 회수}}$$

$$f_j^d = \frac{\sum_{i=1}^{N^p} \text{가장 좋은 각도로 평가된 총 회수}}{A_i \text{에서 } B_j^d \text{를 실행한 총 회수}}$$

여기서 f_j^v 는 속도 제어기에 대한 적합도이며 f_j^d 는 회전각 제어기에 대한 적합도이다.

4. 실험 결과

실험에서 사용된 행동 전략과 행동 제어기를 위한 함수 집합과 터미널 집합이 각각 표 1과 표 2에 제시되어 있다. 행동 전략을 위한 함수 집합은 현재 상황을 판단하여 수행되어야 하는 기본 행동을 결정하며, 터미널 집합은 기본 행동을 나타낸다. 행동 제어기를 위한 함수 집합은 센서의 출력에 의해 속도와 회전각을 계산하는 함수를 생성하기 위해 사용되며, 터미널 집합은 각 센서의 출력값과 현재 수행 중인 세부 행동을 나타낸다.

집합	심볼
함수 집합	IF-OBSTACLE, IF-ROBOT, IF-TABLE, IF-GOAL, PROG2, PROG3
터미널 집합	FORWARD, AVOID, RANDOM-MOVE, TURN-TABLE, TURN-GOAL, STOP

표 1. 행동 전략을 위한 함수, 터미널 집합

집합	심볼
함수 집합	+, -, *, %, SIN, COS, EXP, RLOG, TAND, TOR, TNOI
터미널 집합	O ₀ , O ₁ , ..., O ₇ , R ₀ , R ₁ , ..., R ₇ , T ₀ , T ₁ , ..., T ₇ , G ₀ , G ₁ , ..., G ₇ , MODE

표 2. 행동 제어를 위한 함수, 터미널 집합

본 논문에서 제안한 방법과 기존의 두 방법을 탁자 이동 문제의 시뮬레이션을 통하여 비교하였다. 실험에 사용된 인자값들이 표 3에 제시되어 있다.

인자	공진화 방법		기존 방법
	행동 전략	행동 제어기	
개체군의 크기	100	4	100
교차 확률	0.9	0.5	0.9
돌연변이 확률	0.1	0.5	0.1
환경	32×32 크기의 실수 공간		
최대 속도	2 unit/step		
회전각	[0°, 360°)		
실험대상	훈련 대상 10개, 시험 대상 10개		

표 3. 실험에서 사용된 인자 값

그림 4는 기존의 두 방법과 공진화 방법의 성능을 비교하고 있다. 공진화 방법이 행동 전략이나 행동 제어기 하나만을 진화시키는 것보다 더 빠르게 더 좋은 결과를 얻을 수 있다. 기존의 방법에 대해서는 행동 전략만을 진화시키는 것보다 행동 제어기만을 진화시키는 것이 더 좋은 성능을 보인다. 이것은 실수 공간의 문제에서 상위 레벨의 전략만으로는 효과적인 제어를 할 수 없음을 의미한다. 그림 5는 개체군 수와 훈련 데이터 크기의 곱으로 계산된 적합도 평가 회수에 따른 적합도 값의 변화를 보이고 있는데, 제안된 방법이 더 좋은 성능을 더 효율적으로 얻을 수 있음을 훨씬 명확하게 나타낸다.

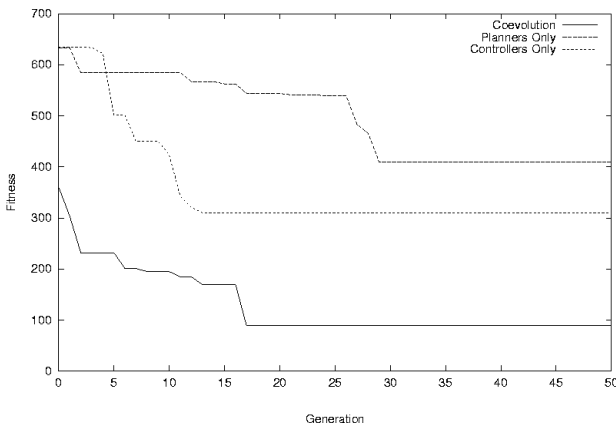


그림 4. 세대에 따른 적합도 값의 변화

5. 결 론

유전자 알고리즘을 사용하여 로봇의 행동 전략과 행동 제어기를 공진화시키는 방법을 제안하고 탁자 운반 문제에 적용하여 그 효과를 검증하였다. 이 방법은 두 방법의 장점을 통합할 수 있으므로 기존의 방법보다 더 효과적인 결과를 얻게 되며, 행동 전략이 적은 수의 행동 제어기를 공유하므로 더 효율적으로 진화를 시킬 수 있다.

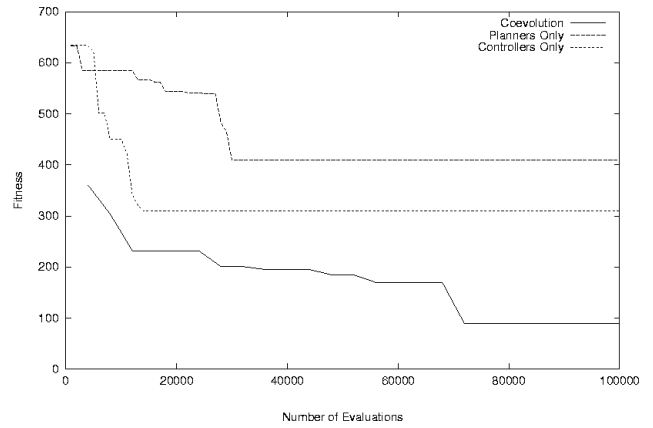


그림 5. 적합도 평가 회수에 따른 적합도 값의 변화

감사의 글

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

참고문헌

- [1] Bennett III, F. H. 1996. Automatic creation of an efficient multi-agent architecture using genetic programming with architecture-altering operations. In Koza, John R. *et al* (editors), *Proceedings of the First Annual Genetic Programming Conference*. MA: MIT Press. pp. 30-38.
- [2] Iba, Hitoshi. 1996. Robustness of robot programs generated by genetic programming. In Koza, John R. *et al* (editors), *Proceedings of the First Annual Genetic Programming Conference*. MIT Press. pp. 321-326.
- [3] Koza, John R. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press.
- [4] Koza, John R. 1994. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press.
- [5] McNutt, Greg. 1997. Using co-evolution to produce robust robot control. In Koza, John R. (editor), *Late Breaking Papers at the 1997 Genetic Programming Conference*. Stanford Bookstore. pp. 141-149.
- [6] Rosca, Justinian P. 1995. Toward automatic discovery of building blocks in genetic programming. In Rosca, Justinian P. (editor), *Proceedings of the 1995 AAAI Fall Symposium on Genetic Programming*. AAAI Press. pp 78-85.
- [7] Zhang, B. T. and Cho, D. Y. 1998. Fitness switching: Evolving Complex Group Behavior Using Genetic Programming. In Koza, John R. *et al* (editors), *Proceedings of the Third Annual Genetic Programming Conference*. Morgan Kaufmann. pp. 431-438.