동적 환경에서의 적응을 위한 로봇 에이전트 제어: 조건별 개체 유지를 이용한 LCS기반 행동 선택 네트워크 학습

Robot agent control for the adaptation to dynamic environment: Learning behavior network based on LCS with keeping population by conditions

박문희, 박한샘, 조성배
연세대학교 컴퓨터과학과

Moon-Hee Park, Han-Saem Park, Sung-Bae Cho
Department of Computer Science
Yonsei University
E-mail: {moonypark,sammy}@sclab.yonsei.ac.kr ,sbcho@cs.yonsei.ac.kr

요 약

로봇 에이전트는 변화하는 환경에서 센서정보를 바탕으로 적절한 행동을 선택하며 동작하는 것이 중요하다. 행동 선택 네트워크는 이러한 환경에서 변화하는 센서정보에 따라 실시간으로 행동을 선택할 수 있다는 점에서, 장시간에 걸친 최적화보다 단시간 내 개선된 효율성에 초점을 맞추어 사용되어 왔다. 하지만 행동 선택 네트워크는 초기 문제에 의존적으로 설계되어 변화하는 환경에 유연하게 대처하지 못한다는 맹점을 가지고 있다. 본 논문에서는 행동 선택 네트워크의 연결을 LCS를 기반으로 진화 학습시켰다. LCS는 유전자 알고리즘을 통해 만들어진 규칙들을 강화학습을 통해 평가하며, 이를 통해 변화하는 환경에 적합한 규칙을 생성한다. 제안하는 모델에서는 LCS의 규칙이 센서정보를 포함한다. 진화가 진행되는 도중 이 규칙들이 모든 센서 정보를 포함하지 못하기 때문에 현재의 센서 정보를 반영하지 못하는 경우가 발생할 수 있다. 본 논문에서는 이를 해결하기 위해 센서정보 별로 개체를 따로 유지하는 방법을 제안한다. 제안하는 방법의 검증을 위해 Webots 시뮬레이터에서 케페라 로봇을 이용해 실험을 하여, 변화하는 환경에서 로봇 에이전트가 학습을 통해 올바른 행동을 선택함을 보였고, 일반 LCS를 사용한 것보다 조건별 개체 유지를 통해 더 나은 결과를 보이는 것 또한 확인하였다.

1. 서론

로봇 에이전트의 학습은 입력되는 센서 정보와 출력을 위한 동작기의 불확실성, 동적으로 변하는 환경과 그에 대한 불완전한 관찰 등으로 인해 매우 어려운 문제로 알려져 있다[1]. 학습은 패턴 인식 분야에서 오랫동안 다루어져 온 문제로, 크게 교사학습(supervised learning)과 비교사 학습(unsupervised learning)으로 나눌 수 있다. 일반적으로 학습에 사용할 수 있는 충분한 양의 데이터를 가지고 있다면 교사학습이 더 효율적인 선택이다[2]. 그러나 로봇 에이전트의 경우 환경을 비롯한 여러 요소가 많은 불확실성을 포함하기 때문에 예측하지 못한 상황이 발생할 가능성이 커서, 미리 데이터를 이용해 학습을 하는 것이 거의 불가능하다[3]. 따라서 로봇 에이전트의 제어를 위한 방법으로는 진화 연산(evolutionary computation)이나 강화학습(reinforcement learning)과 같은 방법이 연구되어왔다[4].

한편 에이전트의 설계를 위한 방법론 중 하나인 행동 기반(behavior-based) 시스템은 센서입력이 바로 행동으로 연결되는 반응형(reactive) 시스템에 목적 및 행동의 시퀀스가 더해져 상위 수준의 행동 제어가 가

능한 방식으로, Maes의 행동 선택 메커니즘(action selection mechanism)이 대표적인 예이다. 이 방법은 반응형 시스템에 가까우면서도 필요한 경우 행동의 시퀀스를 통해 계획을 세울 수 있는 장점을 갖기 때문에 많은 응용연구가 이루어져 왔다. Nicolescu와 Mataric은 로봇 에이전트가 사람 및 다른 에이전트들 사이의 상호작용을 통해 학습이 가능하도록 하기 위해 행동선택 네트워크를 변형하여 사용하였고[5], Khoo와 Zubek은 게임 캐릭터의 행동을 생성하기 위해 행동선택 네트워크를 사용하였다[6].

행동선택 네트워크는 센서 입력과 선행 조건을 고려해 현재 상황에서 가장 우선순위가 높은 행동을 선택하는 방식으로, 반사적으로 환경에 적절한 행동을 생성할 수 있고, 선행조건, 목적, 그리고 행동의 모델링에 따라 환경에 독립적으로 적용할 수 있다[7]. 하지만 로봇 에이전트가 처할 수 있는 불확실한 상황에서 적응적인 행동을 생성하지는 못하며, 네트워크의 초기 설계는 주어진 문제에 의존적으로 구성되기 때문에 이후 변화하는 환경에 대해 부적절한 행동을 보일 수 있다[8]. 이런 단점을 보완하고자 본 논문은 행동선택 네트워크의 연결을 LCS(learning classifier system)를 이용해 진화 학습시켰다. LCS는 강화학습을 통한 피드백을 이용해 규칙을 진화시켜 환경조건에 적합한 규칙을 학습하는 방식으로, 제안하는 모델은 고유한 LCS에 다양한 환경 조건별로 개체를 유지하는 아이디어를 추가함으로써 진화 도중 현재의 환경정보와 매치되는 규칙이 없어 학습된 개체가 사라지는 단점을 개선하여 보다 효율적인 학습이 가능하도록 하였다.

제안하는 방법의 유용성을 보이기 위해서 Webots 시뮬레이터를 이용한 실험을 수행하였다. 제안하는 모델을 이용해 행동선택 네트워크의 노드 간 연결을 학습함으로써 변화하는 환경에 적응할 수 있는 네트워크를 구성하였으며, 일반 LCS를 이용한 학습 방법과의 비교 실험을 통해 조건별로 개체 유지를 함으로써 더 나은 결과를 보이는 것 또한 확인하였다.

2. 행동 선택 네트워크

P. Maes가 제안한 행동 선택 네트워크는 센서로부터 얻어진 외부환경 정보, 내부목표, 그리고 하위레벨의 행동을 의미하는 행동 노드들로 구성된다[9]. 그림 1. A에서와 같이 노드들은 내부연결과 외부연결을 갖는다. 내부연결은 행동 노드 사이의 연결을 의미하며, 선행자, 후임자, 억제자의 3가지 종류의 연결이 있다. 외부연결은 센서 또는 환경과 행동 노드 사이의 연결, 내부목표와 행동 노드 사이의 연결의 2가지 종류가 있다.

그림 1. B는 행동노드와 그 구성 요소를 보여준다. 행동선택은 그림1.C의 과정을 거쳐 로봇이 속한 환경과 상호작용을 한다. 환경과 목표로부터 활성도를 계산하고, 그 값이 내부링크들을 따라 확산된다. 활성도가 한 없이 커지거나 작아지는 것을 막기 위해서 정규화 과정을 거친 후, 활성도가 가장 높은 행동을 선택하여 실행한다. 만약 활성도가 임계값 보다 적으면 임계값을 감소시키고 앞의 과정을 반복한다.

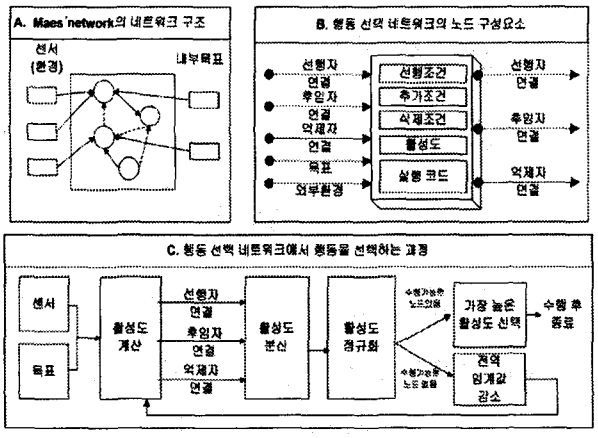


그림 1. 행동 선택 네트워크의 구성과 행동선택 절차

3. 조건별 개체유지를 이용한 LCS기반 행동 선택 네트워크 학습

LCS를 이용한 학습에서 개체들이 오랜 시간동안 활성화 되지 않으면, 기존에 학습되었던 개체들이 소멸되는 경향을 보인다[10]. 이는 학습의 효율성을 떨어뜨려, 행동 선택 네트워크에서 올바른 행동을 선택하는 비율을 낮춘다. 또한 일반 LCS는 매치 되는 개체 집합의 편차가 큰 반면에, 제안하는 방법은 매치 된

표 1. 진화 학습을 위한 행동 선택 네트워크 표현

	비트	의 미	
조건부 (6 bit)	C1	목적지의 위치	위
	C2		아래
	C3		왼쪽
	C4		오른쪽
	C5	장애물 유무	단거리장애물
	C6		장애물위치
행동부 (17 bit)	A1	"장애물 없음"→ 행동노드 11: 직진 10: 좌회전 01: 우회전	외부연결 (환경 센서로부터)
	A2		
	A3	"장애물 근처"→"장애물피하기"	
	A4	"직진"→"장애물피하기"	내부연결 (00:연결없음, 01: 활성연결 10:억제자 연결)
	A5		
	A6	"장애물피하기"→"직진"	
	A7		
	A8	"좌회전"→"장애물피하기"	
	A9		
	A10	"장애물피하기"→"좌회전"	
	A11		
	A12	"우회전"→"장애물피하기"	
	A13		
	A14	"장애물피하기"→"우회전"	
	A15		
	A16	"목적지"→ 행동노드 11: 직진 10: 좌회전 01: 우회전	외부연결 (내부 목표로부터)
	A17		

개체집합이 상수 값으로 일정하다는 점에서 더 안정적이고, 효율적인 학습이 가능하다. 본 논문에서는 주어진 환경에서 요구되는 모든 센서 정보를 고려하여 개체집합을 유지시킴으로써 학습된 개체가 사라지는 단점을 극복하여 효율적인 학습이 가능하도록 하였다.

진화학습을 위한 로봇 에이전트의 센서 정보와 네트워크 연결은 표1과 같이 표현되었다. 행동부는 행동선택 네트워크로 모델링 하였다. 조건부 6비트 중 C1~C4, 로봇 에이전트와 목적지 사이의 상대적인 위치, C5, 로봇 에이전트를 중심으로 가장 가까운 위치의 장애물 유무, C6은 모든 센서에 대한 장애물의 위치를 나타낸다.

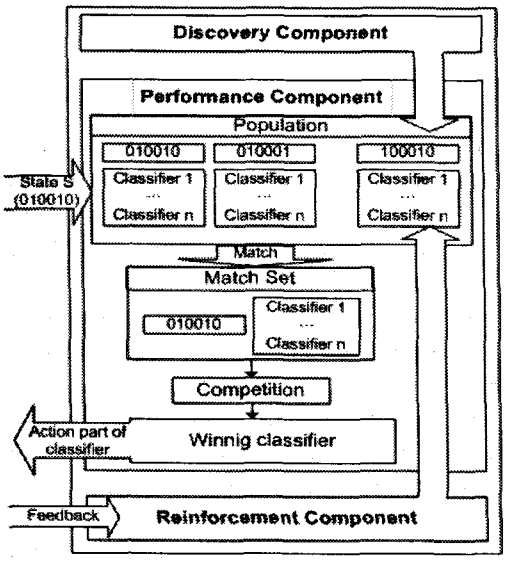


그림 2. 조건별 개체 유지를 이용한 LCS구조

그림 2는 본 논문에서 제안하는 조건별로 개체를 유지하여 효율적인 학습이 가능한 LCS 구조이다. 개체집합은 환경으로부터 얻을 수 있는 모든 센서정보별로 일정수의 개체를 유지한다. 로봇 에이전트의 이동에 따라서 지속적으로 센서정보가 변경되고, 그에 따라 전체 개체집합 내에서 적절한 부분 개체집합을 선택한다. 선택된 부분 개체집합 내에서 개체간의 경쟁을 통해 현재의 상황에서 가장 적절한 개체를 선택하여 행동부를 환경으로 피드백하는 과정을 반복한다.

4. 시뮬레이션 및 결과 고찰

4.1 실험환경

본 논문에서는 학습을 통해 로봇 에이전트가 적절한 행동을 선택하여 빠른 시간 내에 목적지 도달하도록 하는 문제를 다룬다. 실험은 Webots 시뮬레이션 환경에서 케페라 로봇을 이용하였다. Webots 시뮬레이션은 그림 4에서 보듯이 사방을 벽으로 구성하여 장애물로 인식하는 환경으로 설정하였다. 케페라 로봇은 그림 3에서 보듯이 8개의 빛과 장애물에 대한 7방향의 입력센서를 가지고 있다.

본 논문에서 다루는 문제는 에이전트의 초기 출발지점에 종속적으로 소수의 개체로 학습이 집중화 되는 현상이 발생한다. 그러므로 다양한 개체를 골고루 학습시켜 학습의 유용성을 입증하고자, 다양한 센서 정보 수집을 위해서 에이전트를 그림4와 같이 여러 위치에서 출발시켰다.

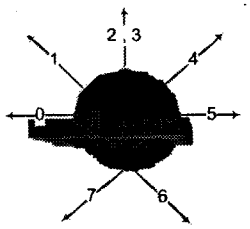


그림 3. 케페라 로봇의 센서 및 이동 방향

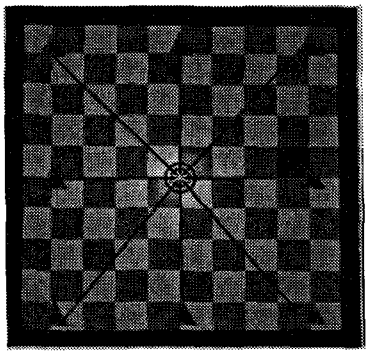


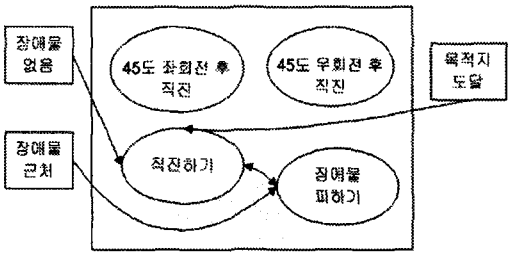
그림 4. Webots 시뮬레이션 환경

그림 4는 본 논문의 실험을 진행한 Webots 시뮬레이션 환경이다. 그림의 중앙에 위치한 중심원은 로봇 에이전트가 도달해야하는 목적지이고, 각 모서리와 사방에 자리 잡은 알파벳순으로 레이블링 된 삼각형은 로봇 에이전트의 출발지점을 나타낸다. 출발위치에 따라 센서정보의 차별성을 주기위해 모든 에이전트는 위쪽을 향해 출발하도록 하였다.

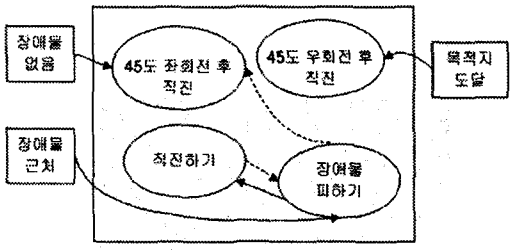
4.2 실험결과 및 분석

실험의 목표는 학습이 진행되면서 소멸되는 개체를 감소시켜, 각 조건에 대한 지속적인 학습을 가능하게 하여 로봇의 초기에 주어진 위치와 관계없이 변화하는 환경 속에서도 학습을 통해서 가장 적절한 행동을 선택하도록 하는 것이다.

일반적으로 행동 선택 네트워크에는 행동 노드 간 연결이 세 종류였으나, 본 논문에서는 유사한 기능을 갖는 선행자 연결과 후임자 연결을 흥분성 연결로 정의하고 억제자 연결은 그대로 억제성 연결로 정의하여 두 종류만을 사용한다. 그림 5. (a)는 학습이전에 설계된 행동 선택네트워크의 구조이다. 이를 바탕으로 로봇 에이전트는 초기에 직진하기와 장애물피하기의 행동만으로 목적지를 찾아 이동한다. 그림 5. (b)는 로봇이 학습을 거치면서 초기 행동 선택 네트워크의 구조(그림 5. (a))에서 각 노드 간의 연결구조가 변경되었음을 확인할 수 있는 한 예이다. 또한 표 2는 제안하는 방법이 학습을 전후로 로봇 에이전트가 목적지 도달하는데 소요되는 스텝수가 감소했음을 보여준다. A에서 H로 학습량은 누적되고 있으며, 지역적 편차가 나타나는 이유는 목적지의 상대적 차이로 인해서 목적지에 도달하는데 요구되는 스텝수가 다르기 때문이다.



(a) 학습 초기



(b) 학습 후기

그림 5. 행동 선택네트워크: 흥분성 연결은 실선, 억제성 연결은 점선표현

표 2. 위치에 따른 학습 전후의 목적지 도달 스텝 수

	목적지 도달에 필요한 스텝 수	
	학습 전	학습 후
A	435	280
B	401	221
C	3106	1693
D	1622	285
E	2573	928
F	82	80
G	1977	1290
H	1382	1337

그림 6은 로봇 에이전트의 위치별 학습 진행에 따른 목적지 도달에 필요한 누적 스텝수이다. 일반 LCS 방법을 적용한 실험과 비교해 보았을 때, 제안하는 방법을 사용했을 때 목적지 도달에 필요한 스텝수가 적으며 학습이 더 빨리 이루어짐을 확인할 수 있다.

5. 결론 및 향후연구

본 논문에서는 실시간 환경에 적응하기 위해 조건별로 개체를 유지하는 LCS 기반으로 행동네트워크를 학습시켰다. 제안하는 방법의 학습 전후를 비교했을 때, 평균 약 39%의 스텝수가 감소하여 학습이 이루어짐을 확인하였으며, 일반 LCS 기반의 학습과 비교했을 때, 에이전트의 목적지 도달 스텝수가 약 80% 감소함을 확인하여 일반 LCS와 비교해 제안하는 방법이 더 나은 결과를 보임을 확인하였다.

향후 연구로는 좀 더 실제적인 환경과 가깝게 움직이는 장애물이 추가된 문제에서도 학습이 이루어짐을 확인해야 할 것이다. 또한 지금의 연구와 같이 간단하게 행동 노드를 구성하는 것이 아니라, 다양한 행동 노드를 갖춘 복잡한 행동 선택 네트워크를 구성하여 학습을 통해서 변화하는 환경에 적응 가능한 행동 선택 네트워크 구축에 대한 연구가 진행되어야 할 것이다.

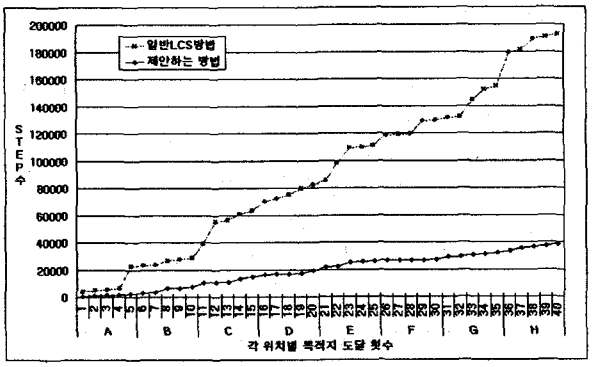


그림 6. 학습 진행에 따른 누적 스텝 수

6. 참고문헌

[1] M. J. Mataric, "Learning in behavior-based multi-robot systems: policies, models, and other agents," *Cognitive System Research*, vol. 2, pp. 81-93, 2000.

[2] R. S. Sutton, A. G. Bafto, *Reinforcement learning*: An introduction, MIT Press, Cambridge, MA, 1998.

[3] C. Zhou and Q. Meng, "Dynamic balance of a biped robot using fuzzy reinforcmenet learning agents," *Fuzzy Sets and Systems*, vol. 134, pp. 169-187, 2003.

[4] T. Kondo and K. Ito, "A reinforcement learning with evolutionary state recruitment strategy for autonomous mobile robots control," *Robotics and Autonomous Systems*, vol. 46, pp. 111-124, 2004.

[5] M. N. Nicolescu and M. J. Mataric, "Learning and interacting in human-robot domains," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 313, no. 5, 2001.

[6] A. Khoo and R. Zubek, "Applying inexpensive AI techniques to computer games," *IEEE Intelligent* Systems, vol. 17, no. 4, pp. 48-53, 2002.

[7] D. Singleton, "An Evolvable approach to the Maes action selection mechanism," *M. S. Thesis of University of Sussex*, 2002.

[8] H.-J. Min, "Generating behaviors of an autonomous robot using goal-oriented planning of behavior network modules," *M. S. Thesis of Yonsei University*, 2005.

[9] T. Tyrrell, "An evaluation of Maes's bottom-up mechanism for behavior selection," *Adaptive Behavior*, vol.2, no.4, pp.307-348, 1994.

[10] Philippe Preux, Samuel Delepoulle, Jean-Claude Darcheville, "A generic architecture for adaptive agents based on reinforcement learning," *Information Sciences*, vol. 161, pp. 37 - 55, 2004.