

게임 분석과 MCTS를 이용한 비디오 게임 일반 인공지능

General Artificial Intelligence for Video Games using Game Analysis and Monte-Carlo Tree Search

저자 (Authors)	김현태, 김도영, 김경중 HyunTae Kim, DoYoung Kim, KyungJoong Kim
출처 (Source)	한국정보과학회 학술발표논문집 , 2014.12, 565-567(3 pages)
발행처 (Publisher)	한국정보과학회 KOREA INFORMATION SCIENCE SOCIETY
URL	http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE06228715
APA Style	김현태, 김도영, 김경중 (2014). 게임 분석과 MCTS를 이용한 비디오 게임 일반 인공지능. 한국정보과학회 학술발표논문집, 565-567
이용정보 (Accessed)	한국외국어대학교 203.253.93.*** 2021/04/02 13:30 (KST)

저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s) and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s) for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

게임 분석과 MCTS를 이용한 비디오 게임 일반 인공지능

김현태[○], 김도영, 김경중

세종대학교 컴퓨터공학과

kimhyun0416@sju.ac.kr, kimdo489@naver.com, kimkj@sejong.ac.kr

General Artificial Intelligence for Video Games using Game Analysis and Monte-Carlo Tree Search

HyunTae Kim[○], DoYoung Kim, KyungJoong Kim

Dept. of Computer Engineering, Sejong Univ.

요 약

게임의 종류는 나날이 증가하고 있고 각 게임에 적합한 인공지능을 개발하는 것은 상당한 노력이 요구된다. 본 논문에서는 이러한 문제점을 해결하기 위해, 게임을 분석하고 Monte-Carlo Tree Search(MCTS)라는 기법을 이용하여 여러 종류의 게임을 동시에 수행이 가능한 인공지능을 개발하고 성능 분석을 수행한다. 특히, 일반적인 비디오 게임을 위한 인공지능을 만들기 위하여 MCTS를 효율적으로 탐색할 수 있도록 greedy-UCB1 함수와 확률적 모델을 제안한다. 제안된 방법으로 개발된 인공지능은 국제 인공지능 대회에 출전하여 4위의 성적을 거둬으로써 성능을 확인할 수 있었다.

1. 서 론

인공지능분야에서는 각기 다른 상황에 놓이게 되었을 때, 효과적인 판단을 내릴 수 있는 인공지능을 가진 에이전트를 개발하는 것이 하나의 중요한 연구 주제이다. 에이전트가 해당 시점에서 여러 가지 선택을 두고 하나를 결정해야 하는 문제는 잘 알려진 멀티암드 밴디트 문제(Multi-Armed Bandit Problem)와 연관이 깊은데, 에이전트가 이와 같은 문제에서 효과적인 결정을 내리는 것은 매우 어려운 문제이다. 왜냐하면, 이 에이전트가 매 순간 현 상황에서 최선의 선택을 위해 활용(Exploitation)을 할 것인지, 새로운 경우를 생각하여 탐사(Exploration)을 할 것인지의 비중을 어떻게 두느냐에 따라 에이전트의 전체적인 성능이 달라지기 때문이다.

이미 특정 보드 게임이나 비디오 게임에서는 멀티암드 밴디트 문제를 효과적으로 풀 수 있는 인공지능이 제안되었다[1]. 또한 최근에 게임 인공지능 분야에서는 멀티암드 밴디트 문제를 해결하기 위한 연구도 진행되고 있다. 하지만, 실제 최근 게임들은 다양하고 각기 다른 목적이 존재하고 이러한 다양한 목적을 인공지능이 파악하고 목적을 달성하기 위해 움직이는 것은 매우 어렵다. 실제로 IEEE 2014 Conference on Computational Intelligence in Games(CIG 2014)에서는 각기 다른 목적을 가진 게임들을 하나의 인공지능으로 플레이하여 서로 경쟁하는 GVG-AI 대회(The General Video Game AI Competition)가 주최되었다¹.

이 GVG-AI 대회는 Training, Validation, Test 세트로 각각 10개씩 총 30가지의 서로 다른 게임을 가지고 이루어진다. Training 세트는 미리 참가자에게 주어지며 자신의 인공지능을 로컬 컴퓨터에서 테스트해보기 위해 쓰인다. Validation 세트는 서버에서 인공지능을 업로드하여 테스트되며 10개의 숨겨진 게임들로 이루어져 있다. 마지막 Test 세트는 대회 당일에 발표되며, 모든 인공지능의 마지막 성능 결과로써 사용된다. 참가자는 모든 게임에 대해 전반적으로 좋은 성능을 내는 인공지능을 개발하여야 하며, 에이전트는 40ms 안에 행동을 결정하지 못하거나 제한된 시간을 초과하게 되면 해당 게임은 지게 된다.

본 논문에서는 GVG-AI 대회에서 제공하는 플랫폼을 활용하여, 다양한 종류의 게임을 하나의 인공지능으로 해결하기 위한 방법을 제안한다. 제안하는 인공지능은 휴리스틱한 방법 일반 비디오 게임 플레이(General Video Game Playing)에서 멀티암드 밴디트 문제를 풀기 위해 주로 쓰이는 MCTS 알고리즘으로 이루어져 있다.

2. 관련 연구

2.1 MCTS(Monte-Carlo Tree Search)

MCTS란 하나의 최적해를 찾기 위해 효율적으로 트리를 탐색하는 알고리즘이다[1]. 기존의 탐색 알고리즘으로 탐색하기 어려웠던 문제들을 MCTS는 효율적으로 탐색할 수 있는 장점이 있다. 바둑과 같이 트리 전체를 탐색할 수 없는 문제에 주로 쓰이며, 기본적인 알고리즘은 먼저 가장 최적의 터미널 노드를 선택하여 트리를 확장하고 휴리스틱 함수를 통해 상황에 대한 평가를 받아 보상을 역전파시킨다.

¹ <http://gygai.net/index.php>

MCTS는 일반적으로 멀티암드 밴디트 문제를 위해, 활용과 탐사를 균형있게 선택하여 주는 upper confidence bound (UCB)가 사용되며 그 중 기본적인 UCB1이라는 수식이 주로 사용된다. UCB1 공식은 다음과 같다.

$$UCB1 = v_i + C \times \sqrt{\frac{\ln N}{n_i}}$$

v_i 과 n_i 는 각각 트리의 i 노드의 추정값과 방문 횟수를 의미한다. 또한, N 은 현재까지 부모 노드의 총 방문횟수를 뜻하며 C 는 조정 가능한 바이어스이다. UCB1으로 얻어진 값은 에이전트가 활용 또는 탐사를 결정할 때 사용된다. MCTS에서 가장 중요한 부분은 게임 혹은 다양한 상황을 시뮬레이션 할 수 있도록 하거나 이를 대체할 수 있는 휴리스틱 함수를 활용하여 각 상황에 대하여 평가를 받는 부분이다. 이는 게임에서 일어날 수 있는 다양한 상황에 대한 평가를 인간이 아닌 인공지능이 해야 하는 부분이기 때문에, 한가지가 아닌 다양한 게임에 대해 에이전트가 평가를 내리는 것은 쉽지 않다.

2.2 GVGP(General Video Game Playing)

GVGP는 하나의 에이전트로 여러 가지의 게임을 플레이하는 것을 의미한다. GVGP를 하기 위해서, 에이전트는 자신의 행동에 게임이 어떻게 반응할 것인지, 이로써 어떤 보상을 얻게 되는지, 결론적으로 어떻게 게임에서 이길 수 있는지를 찾아야만 한다[2]. 또 에이전트가 하나의 게임에 대한 도메인 지식을 이용하여 만들어지게 되면, 그 에이전트는 해당 게임을 제외한 다른 게임에 대해서는 좋지 않은 성능을 발휘하게 되므로 전반적인 측면에서의 개발이 필요하다.

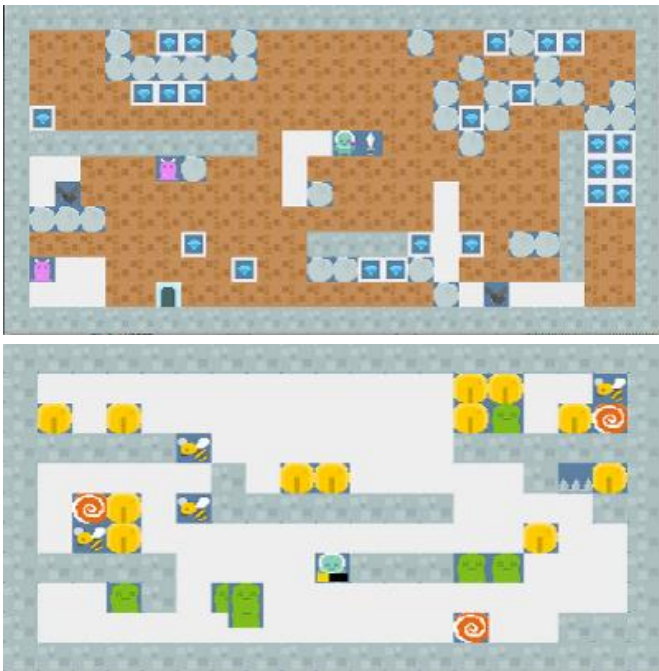


그림 1. VGDL로 만든 GVGP-AI 게임 예시

Tom Schaul은 GGP(General Game Playing)을 위한 환경을 제공하기 위해 VGDL(Video Game Description Language)을 제안하였다[3]. VGDL은 2차원 비디오 게임을 간단히 제작할 수 있는 개발언어으로써, 이미지만 있다면 간단하게 여러 가지 컨셉의 게임을 제작할 수 있는 것이 장점이다. 실제로 GVG-AI 대회 of 모든 게임은 VGDL을 이용하여 만들어졌으며, 그림 1은 GVG-AI 대회의 Training 세트중 하나이다. VGDL를 사용하면 게임의 간단한 승리조건(예를 들면, 단순 길찾기 문제)부터 복잡한 로직이 요구되는 승리조건(예를 들면, 일정개수의 다이아몬드를 획득한 후 출구로 나가야 하는 문제)까지 다양한 승리조건을 몇 줄의 언어로 정의가 가능하다. 그렇기 때문에 GVGP 문제를 풀기 위한 환경을 손쉽게 구축하는 것이 가능하다.

3. 제안하는 방법

해당 장에서는 GVGP를 위한 인공지능을 개발하기 위해 MCTS라는 알고리즘을 사용한 이유를 설명한다. 또한, 이 알고리즘을 사용한 인공지능이 여러 게임에서도 전반적으로 좋은 성능을 낼 수 있도록 UCB1을 변형한 greedy한 UCB1과 확률적 모델을 사용하여 인공지능을 개발하는 방법에 대해 제안한다.

GVG-AI 대회에서는 40ms라는 짧은 시간 안에 에이전트가 행동을 결정하고 움직여야 한다. 짧은 시간안에 결정을 내리기 위해서는 탐색 알고리즘을 활용하여 현재 상황에서 다음 상황을 예측하여 효율적으로 탐색하는 것이 무엇보다 중요하다. 탐색이 제대로 이루어졌다면, 탐색을 통해 얻은 정보를 바탕으로 에이전트는 다음 행동 중 가장 효과적인 움직임을 선택한다.

탐색 알고리즘으로는 깊이 우선 탐색, 너비 우선 탐색, 무작위 탐색과 같은 기본적인 방법들이 존재한다. 만약 이 탐색 방법들로 게임의 다음 상황을 탐색 한다면, 제한된 시간안에 에이전트의 의사결정을 위한 정보를 제대로 얻을 수 없다. 하지만, MCTS같은 경우 UCB1을 적용하여 활용 또는 탐사를 균형있게 수행할 수 있어 보다 의미있는 정보를 얻을 수 있다.

GVGP에서는 탐색으로 얻어지는 보상은 크지만 그 보상이 숨겨져 있는 경우가 많다. 만약 에이전트의 탐색 알고리즘이 숨겨진 보상이 존재하는 특정 트리 노드를 탐색을 통해 찾게 되면, 그 특정 노드의 주변의 노드들은 다른 상황의 노드보다 더 좋은 평가를 받을 확률이 높다. 하지만 UCB1을 이용하게 되면 균형적으로 탐색을 하기 때문에 자칫 다른 쪽으로의 탐사를 진행할 수 있다. 그러므로, 본 논문에서는 greedy-UCB1을 이용하여 효과적인 선택을 할 수 있도록 한다.

$$\text{Greedy-UCB1} = v_i + C \times \sqrt{\frac{\ln N}{n_i}} + t_i$$

위 수식에서 t_i 은 노드 i 의 총 누적값이다. 이 값은 MCTS를 반복하면서 노드 i 를 포함한 자식 노드가 받은 보상이 역전파 되면서 누적된다. Greedy-UCB1의 t_i 의 값만 제외하고 나머지는 UCB1과 같다. Greedy-UCB1은 처음에는 t_i 의 값이 작아 다른 노드와 비슷한 비율로 탐색을 할 수 있지만 t_i 의 값이 커지게 되면, greedy-UCB1의 값이 상대적으로 커져서 높은 확률로 t_i 가 큰 방향으로 탐색을 하게 된다. 그러므로, 제안하는 방법을 이용하면 트리 탐색 초반에는 활용과 탐사가 균형적으로 이루어지게 되지만, 보상을 많이 해주는 상황을 발견하게 되면 그 방향으로의 탐욕적인 탐색을 하게 된다.

제안하는 방법을 이용한 인공지능의 MCTS는 위와 같이 다음 노드를 확장하는 선택 단계에 greedy-UCB1를 이용하여 트리의 터미널 노드까지 선택을 한다. 확장 단계 후, MCTS는 휴리스틱 함수 또는 시뮬레이션을 거쳐 확장 단계에 대한 보상을 획득하여 역전파를 수행한다. GVG-AI 대회 플랫폼은 시뮬레이션이 제공되는데, 게임 전체의 현재상태를 복사하고 에이전트가 다음 행동을 하였을 때, 게임 전체의 다음상태를 알 수 있다. 그러므로, 제안하는 방법으로 설계된 에이전트는 제한된 깊이까지 계속 다음 상황을 시뮬레이션을 하여 확장된 노드에 대한 보상을 받는다.

VGDL로 만들어진 게임들은 각 게임마다 에이전트가 가능한 행동은 다르다. 좌우로만 움직일 수 있는 게임이 있는 반면, 상하좌우와 공격까지 가능한 게임도 존재한다. 본 논문에서는 에이전트가 공격이 가능할 시, 공격 쪽의 확률을 조금 더 주어 결과적으로 공격위주의 선택이 중심이 되게 설계하였다. 왜냐하면, 보통의 경우 공격이 가능한 게임에서는 제한된 시간 안에 Non-Player Character를 해치우거나 목표물을 없애야 한다. 하지만 에이전트에게 '바로 앞에 적이 있을 때 공격을 하라'라고 가르치는 것은 GVG-P에서 벗어날 수 있다. 이러한 이유로 공격을 선택할 확률을 높여서 공격을 하면 좋은 성능을 보여준다.

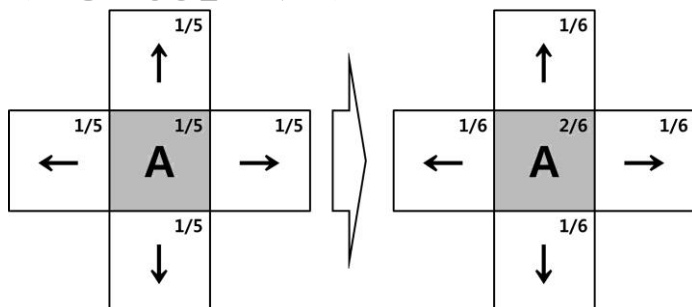


그림 2. 확률적 모델의 적용 예시

그림 2는 제안하는 확률적 모델의 예이다. 그림 2의 왼쪽 그림은 회색의 에이전트가 상하좌우, 공격(A)이 가능함을 나타내고 있다. 각 셀의 오른쪽 위의 숫자는

에이전트가 행동을 할 확률을 나타낸다. 제안하는 확률적 모델을 인공지능에 적용하게 되면 그림 2와 같이 에이전트는 특정 다음 행동(그림 2에서는 공격 A)을 할 확률이 높아지게 된다. 이 확률적 모델을 적용하면, 게임 분석을 통해 현재 게임의 상황을 파악하여 에이전트의 행동비중의 변화를 준다.

4. 결론 및 향후 연구

본 논문에서는, 게임을 분석하여 MCTS의 선택단계에서 초반 탐색에는 활용과 탐사를 균형 있게 수행하고 후반 탐색에서는 탐욕적인 greedy-UCB1을 제안하였다. 또한, 에이전트의 행동을 게임 분석을 통해 무작위적인 선택이 아닌 확률적으로 선택 함으로써, 좀더 효율적인 움직임이 가능한 모델을 제안하였다. 제안된 방법으로 개발된 인공지능은 CIG2014에서 열린 GVG-AI 대회에서 19팀 중 4위의 성적을 거두었다(표 1). 참가팀들 중 인공지능을 제출한 평균 획득 점수는 60.71점이며 평균적으로 500게임 중 114.64번 이기는 것으로 나타났다. 향후 개발에서는 확률적 모델을 발전시켜 MCTS의 확장단계에도 적용 가능한 인공지능을 개발할 예정이다.

표 1. CIG2014 GVG-AI 대회 결과

순위	에이전트	점수	이긴횟수
1	Adrienctx	158	256 / 500
2	JinJerry	148	216 / 500
3	Shmokin	77	158 / 500
4	제안하는 방법	68	127 / 500
:	:	:	:

6. 감사의 글

이 논문은 2013년도 정부(미래창조과학부)의 재원으로 한국 연구재단의 지원을 받아 수행된 중견연구자지원사업(2013 R1A2A2A01016589) 및 뇌과학 원천기술개발사업임(2010-0018950)

REFERENCES

- [1] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A Survey of Monte Carlo Tree Search Methods," IEEE Trans. Comput. Intell. AI Games, vol. 4, no. 1, pp. 1-43, 2014.
- [2] J. Levine, C. B. Congdon, M. Ebner, G. Kendall, S. M. Lucas, R. Mikkulainen, T. Schaul, and T. Thompson, "General Video Game Playing," Dagstuhl Follow-Ups, vol. 6, pp. 77-83, 2013.
- [3] T. Schaul, "A video game description language for model-based or interactive learning," in Computational Intelligence in Games (CIG), 2013 IEEE Conference on, pp. 1-8, 2013.