연속 행동공간에서의 몬테-카를로 트리 탐색에 관한 연구

이종민, 김건형, 김기응

한국과학기술원

jmlee@ai.kaist.ac.kr, ghkim@ai.kaist.ac.kr, keeeung.kim@kaist.edu

A Study on Monte-Carlo Tree Search in Continuous Action Spaces

Jongmin Lee, Geon-Hyeong Kim, Kee-Eung Kim

KAIST

요 약

몬테-카를로 트리 탐색 (Monte-Carlo Tree Search; MCTS)은 온라인 계획 알고리즘으로 다양한 이산 행동공간 문제에서 큰 성공을 거둔 바 있지만, 연속행동공간에서는 우선하여 고려되는 방법론은 아니었다. 이는 트리 탐색을 가능하게 만들기 위해 행동공간을 거칠게 이산화하는 작업이 불가피하기 때문이다. 본 논문에서는 연속 행동공간에서 UCT의 전역적 탐색과 환경의 미분 정보를 활용한 지역적 탐색을 결합하는 방법에 대해 다루고자 한다. 연속 행동공간의 제어문제의 벤치마크 실험 결과는 제안하는 방법론이 다양한 비교 대상 알고리즘의 성능을 능가함을 보여준다.

Ⅰ. 서론

로봇 조작을 위해 연속된 토크를 조절하거나 당구를 플레이하는 등 현실의 다양한 문제들은 연속 행동공간에서 행동을 차례로 선택하는 계획 (planning) 능력을 요구한다. 몬테-카를로 트리 탐색 (Monte-Carlo Tree Search; MCTS) [1] 은 큰 탐색 공간을 가진 문제를 풀기 위한 효율적인 온라인 계획 알고리즘으로, 바둑 [2]과 실시간 게임 [3] 등 행동공간이 이산적인 경우들에 매우 성공적으로 적용된 바 있다. 그러나, 이산 행동공간에서와는 달리 행동공간이 연속적인 문제들에 대해 MCTS는 최우선으로 고려되는 계획 알고리즘이 아니었다. 이는 선택 가능한 행동의 수가 무한히 많아 트리 탐색을 위해 행동공간을 거칠게 이산화하는 과정이 필요하며, 이 결과로 매우 정밀한 조작을 요구하는 현실 문제들에 실효성이 떨어지는 결과를 초래할 수 있기 때문이다.

현존하는 연속 행동공간을 다루는 MCTS 알고리즘들은 크게 두 가지 종류의 접근법을 취한다. 점진적 확장 (Progressive Widening) [4]은 트리 각 노드의 선택 가능한 행동의 가짓수를 노드의 방문 횟수에 기반하여 점진적으로 늘려나간다. 또 다른 접근법인 계층적 낙관 최적화 (Hierarchical Optimistic Optimization; HOO) [5]는 행동공간을 점진적으로 쪼개 나가며 선택될 행동의 정밀도를 높여나간다. 이러한 접근법들은 적당히 좋은 행동을 고르는 데에는 충분한 성능을 보여줄 수 있으나, 매우 정밀한 제어를 계획 결과로 얻기 위해서는 엄청난 양의 시뮬레이션이 필요하다. 반면, 기울기 (gradient) 기반의 최적화는 정확한 국소 최적점을 찾는 데에는 매우 효과적이지만 전역 최적점에 도달하기 매우 까다롭다.

본 논문에서는, MCTS와 기울기 기반의 행동 미세조정을 결합하는 방법론을 제안하고자 한다. 제안하는 알고리즘은 MCTS를 통한 전역적 (이지만 거친) 탐색과 행동 값 기울기 기반의 지역적 (이지만 세밀한) 탐색을 함께 수행한다. 제안하는 알고리즘은 연속 행동공간을 갖는 현실의 다양한 문제들이 지역적으로 상태와 행동에 대해 미분 가능한 환경 역학을 가지고 있다는 관찰에 기반을 두며, 이 경우 기울기 상승 기반의 최적화가 매우 효과적일 수 있음에 주목한다.

Ⅱ. 본론

연속 마코프 의사 결정 문제 (Markov Decision Process; MDP)는 상태의 집합 $S \subset \mathbb{R}^n$, 행동의 집합 $A \subset \mathbb{R}^m$, 전이 함수 $\rho : S \times A \to S$ $s_{t+1} = \rho(s_t, a_t)$, 그리고 보상함수 $r(s,a) \in \mathbb{R}$로 정의된다. MDP의 목표는 보상 합의 기댓값을 최대화하는 것이다. 온라인 계획 방법론은 매시간단계마다 제한된 지평 (horizon) T에 대해서 다음과 같이 최적 행동의 순서들 $a_0^*, \ldots, a_T^*$를 계산하는 문제를 풀게 된다.

$$\begin{aligned} \arg\max_{a_0,\ldots,a_T} \; & V(s_0) = \sum_{t=0}^{T} r(s_t, a_t) \\ \text{s.t.} \; & s_{t+1} = \rho(s_t, a_t) \quad \forall t = 0, \ldots T-1 \end{aligned} \tag{1}$$

제한된 탐색시간 동안 식 (1)을 푼 후, 첫 번째 행동 a_0^*가 실행되며, 다음 시간단계에서 다시 식 (1)을 풀어 계획을 수행하는 과정이 반복된다.

MCTS는 범용적인 온라인 계획 알고리즘으로서 랜덤 샘플링과 트리 탐색을 결합하며, 빈 트리에서 시작하여 유한 지평의 시뮬레이션과 잎 노드가 확장되는 과정을 반복한다. UCT [1] 는 가장 널리 채택되는 MCTS 알고리즘으로 탐색 트리의 중간 노드들에서 행동을 고르기 위해 UCB 정책을 사용하며 식 (1)을 애니타임 (anytime) 방식으로 풀게 된다.

$$a_t \sim \begin{cases} \arg\max_{a \in A(h_t)} \left[V(h_t a) + c\sqrt{\frac{\log N(h_t)}{N(h_t a)}} \right] & \text{(inside the tree)} \\ \pi_{\text{rollout}}(s_t) & \text{(outside the tree)} \end{cases} \tag{2}$$

UCT는 유한 MDP의 경우 시뮬레이션 횟수가 무한히 많아지면 최적의 행동 순서들 $a_0^*, \ldots, a_T^*$을 확률 1로 선택함이 이론적으로 보장된다.

무한한 행동공간을 다루기 위해 본 논문에서는, 고려하는 유한한 행동의 집합을 점차 늘려나가는 점진적 확장 [4]을 채택하였다. 점진적 확장은 선형 이하의 확장 속도를 조절하는 상수 $\alpha \in (0,1)$와 트리의 각 중간

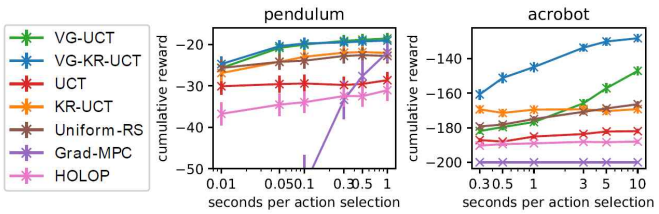

그림 1. 연속 행동 제어 벤치마크 실험 결과

노드 h의 방문 횟수 $N(h)$, 노드에서 고려하는 행동 입자들의 집합 $A(h)$에 대하여 다음의 부등식 (3)이 만족 될 때마다, 새로운 행동 입자를 생성하여 노드 h에서 고려하는 유한한 행동의 집합에 추가한다.

$$\lfloor N(h)^{\alpha} \rfloor \geq |A(h)| \qquad (3)$$

한편, 식 (1)을 푸는 온라인 계획 상황에서, 행동 순서들 $a_0, \ldots, a_T$이 주어졌을 때 전이 함수 $s_{t+1} = \rho(s_t, a_t)$와 보상함수 $r_t = r(s_t, a_t)$를 통해 중간 상태들 $s_{0:T}$와 보상 $r_{0:T}$이 계산되는데 이때 식 (1)을 행동 a_t에 대해 미분하면 재귀적으로 계산되는 값-기울기 (value-gradient) 식을 얻을 수 있다.

$$\frac{\partial V(s_0)}{\partial a_t} = \frac{\partial r_t}{\partial a_t} + \frac{\partial s_{t+1}}{\partial a_t}\frac{\partial V(s_{t+1})}{\partial s_{t+1}} \qquad (4)$$

$$\frac{\partial V(s_t)}{\partial s_t} = \frac{\partial r_t}{\partial s_t} + \frac{\partial s_{t+1}}{\partial s_t}\frac{\partial V(s_{t+1})}{\partial s_{t+1}} \qquad (5)$$

따라서, 만약 전이 함수와 보상함수의 자코비안(Jacobian)을 매시간단계마다 얻을 수 있다면 이로부터 값-기울기를 재귀적으로 계산할 수 있으며 값-기울기 상승을 이용해 행동 입자들 $a_{0:T}$를 지역적으로 개선할 수 있다.

$$a_t \leftarrow a_t + \eta \frac{\partial V(s_t)}{\partial a_t} \text{ for } t = 0, \ldots T \qquad (6)$$

본 논문에서는 UCT의 거칠게 이산화된 행동에 대한 전역적 탐색과 값-기울기 상승을 통한 지역적 탐색을 결합하는 알고리즘을 제안하며, 이는 다음의 두 절차를 매 시뮬레이션마다 반복한다.

1. 행동의 순서 $a_{0:T}$를 UCT를 통해 선택한다. (식 (2))
2. 선택된 행동 $a_{0:T}$을 값-기울기 상승으로 업데이트한다. (식 (4-6))

실험 결과

제안하는 알고리즘을 전통적인 연속 행동공간 제어 문제인 Pendulum과 Acrobot에 대하여 다양한 비교 대상 알고리즘과 비교하였다. 기존의 MCTS 알고리즘인 (점진적 확장이 결합된) UCT, 연속 행동공간에서의 정보 공유를 통해 탐색 성능을 높인 KR-UCT [6], 제한된 탐색 시간 동안 행동들을 무작위로 생성한 후 그중 가장 높은 보상을 받았던 궤적의 첫 번째 행동을 선택하는 Random Shooting (Uniform-RS) [7], HOO를 통해 행동공간을 쪼개 나가며 행동 순서들을 최적화하는 HOLOP [8], 랜덤한 행동 순서를 초기값으로 하여 반복된 값-기울기 상승을 통해 행동을 제한된 탐색시간 동안 개선하는 Grad-MPC와 비교하였다. VG-KR-UCT는 KR-UCT을 제안하는 핵심 아이디어 (트리 탐색과 선택된 행동에 대한 값-기울기 상승 수행)을 사용하여 확장한 알고리즘이다.

그림 1의 실험 결과는 제안하는 방법론 (VG-UCT 및 VG-KR-UCT)이 다른 비교 알고리즘의 성능을 크게 능가하고 있음을 보여준다. 이는 전역적 탐색과 지역적 탐색을 결합하는 방식의 효과를 잘 드러내고 있다. UCT, Uniform-RS, 그리고 HOLOP과 같은 알고리즘은 거친 이산화의 문제로 계획의 최종 결과로 수행되는 행동이 충분히 정밀하지 못하여 큰 보상을 받지 못한다. 반면, Grad-MPC의 경우 행동 순서의 초기값에 큰 영향을 받는다. 특히 Acrobot과 같이 유의미한 보상 신호를 받기 위해서는 충분히 좋은 위치에 도달해야 하는 문제에서는 기울기 신호의 부재로 성능 향상이 거의 일어나지 않는 것을 확인할 수 있었다. 마지막으로, UCT에서 커널 회귀 (kernel regression)을 사용한 비슷한 행동간 정보 공유는 성능 향상에 도움이 되었으나 그 향상 정도는 제한적이었다.

Ⅲ. 결론

본 논문에서는 연속 행동공간을 다루는 MCTS 알고리즘을 제안한다. 제안하는 방법론은 전통적인 MCTS의 전역적 탐색과 값-기울기 상승을 이용한 행동 입자 최적화의 지역적 탐색을 결합하며, 실험 결과는 제안하는 방법론이 다양한 비교 대상 접근법들의 성능을 능가함을 보여준다. 향후 연구 방향으로 제안하는 방법론의 이론적 최적 수렴 등에 관한 보장, 블랙-박스 시뮬레이터만 주어진 상황에서의 알고리즘 효율화 및 다양한 벤치마크 제어문제들에 대한 추가 실험 등을 고려할 수 있다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터지원사업의 연구결과로 수행되었음 (IITP-2019-2016-0-00464)

참 고 문 헌

[1] L. Kocsis and C. Szepesvari, “Bandit based Monte-Carlo planning”, Proceedings of the Seventeenth European Conference on Machine Learning (ECML), pp. 282 - 293, 2006.

[2] D. Silver et al. “Mastering the game of go without human knowledge”, Nature, 550:354 - 359, 2017.

[3] T. Pepels et al. “Real-time Monte Carlo tree search in Ms Pac-Man”. IEEE Transactions on Computational Intelligence and AI in Games, 6(3):245 - 257, 2014.

[4] R. Coulom, “Computing Elo Ratings of Move Patterns in the Game of Go“. In Computer Games Workshop, 2007.

[5] S. Bubeck et al. “Online optimization in X-armed bandits”. In Advances in Neural Information Processing Systems (NIPS) 21, pp. 201 - 208, 2009.

[6] T. Yee et al. “Monte Carlo tree search in continuous action spaces with execution uncertainty”, In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), pp. 690 - 696, 2016.

[7] A. Nagabandi et al. “Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning”, 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 7559 - 7566, 2018.

[8] A. Weinstein and M. Littman, “Bandit-based planning and learning in continuous-action Markov decision processes“, In Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS), 2012.