Monte Carlo Tree Search control scheme for the inverted pendulum

Yixuan Tang, Grzegorz Orzechowski, Aki Mikkola

Department of Mechanical Engineering
LUT University
53850 Lappeenranta, Finland
Yixuan.Tang@lut.fi, Grzegorz.Orzechowski@lut.fi, Aki.Mikkola@lut.fi

EXTENDED ABSTRACT

1 Introduction

Monte Carlo Tree Search (MCTS), as one of the reinforcement learning (RL) techniques, combines a selective search tree with repeatedly applying a Monte Carlo simulation [1, 2]. It is a search and planning framework for finding a semi-optimal path by selectively evaluating and comparing the values of each decision at each leaf node of the tree. Monte Carlo simulations are used as the Markov decision process optimization method [3]. Markov decision process is a sequential decision-making framework primarily used in the reinforcement learning [2]. MCTS has already profoundly impacted artificial intelligence (AI) approaches due to its spectacular success in games and planning problems.

This study discusses available choices for MCTS parameters and provides insight into the example of AI-based control with the multibody framework. The multibody application introduced in this study is the inverted pendulum on the cart.

2 Method

Modeling a control task as a Markov decision process is a key concept in reinforcement learning. In a Markov decision process setting, the reinforcement learning problem is formulated by the tuple $M = (S, A, R, P, \gamma)$ [4], where S is the state space, A is the action space, $R(s,a)$ is the reward for being in state s after taking action a, and $P \in [0\ \ 1]$ is the transition probability distribution function. In the tuple, $\gamma \in [0\ \ 1]$ is the discount factor, which presents how much the future affections are discounted. The objective of the agent is to take actions that maximize the expected value (sum of future discounted rewards) G_t during an episode. It can be shown that $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} \cdots + \gamma^{T-t-1} R_T = R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} \cdots + \gamma^{T-t-2} R_T) = R_{t+1} + \gamma G_{t+1}$, where T is the final step of an episode.

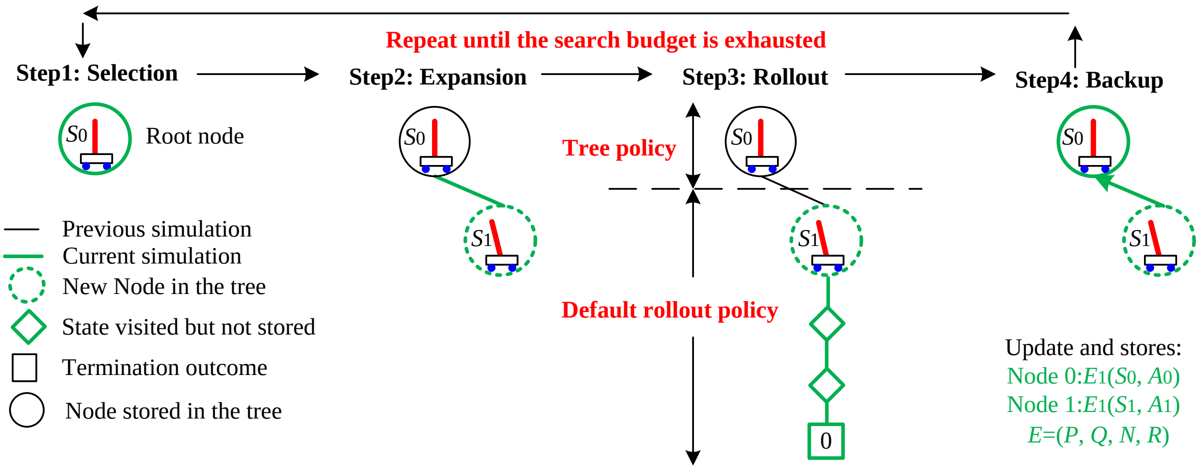


Figure 1: MCTS for a single inverted pendulum under two possible actions, *i.e.*, left and right, in four steps.

The core of MCTS is based on the search tree, where all possible step variants beginning from the state of the agent are mapped. The MCTS control agent tries to do a finite number of random simulations to find the best combination of moves in a look-ahead tree, and sample returns are observed between the initial and termination state during each a Monte Carlo simulation. The values estimated from Monte Carlo simulations provide the most promising combination of moves in the search tree. The core of MCTS contains the iterative process of four steps, please see Figure 1.

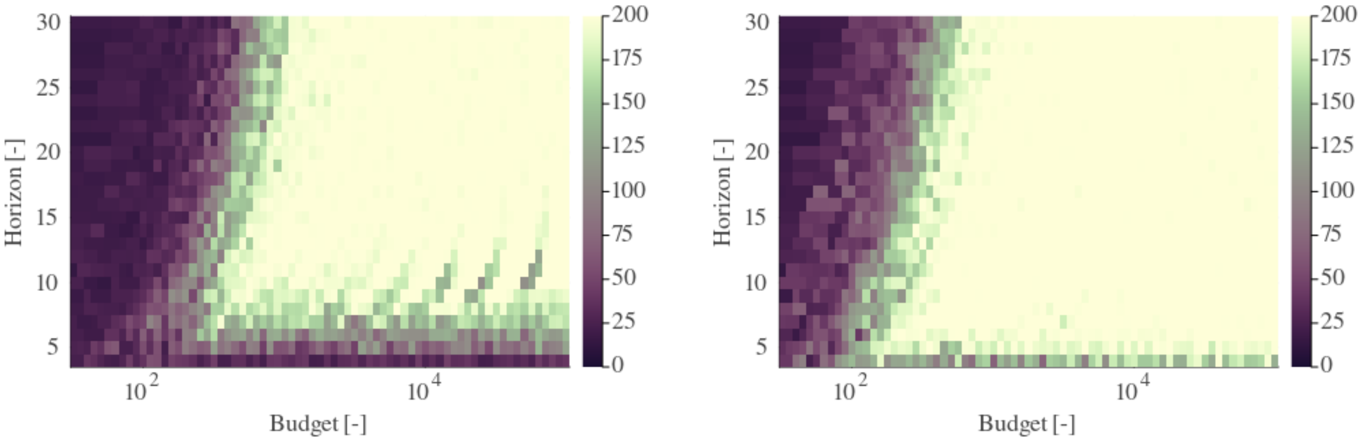
(1) Selection – In this step, the agent has to exploit what it has already experienced to obtain a reward and explore new paths with the potential to find a better move in the future. This trade-off between exploitation and exploration represents the real-life problem-solving approaches in the reinforcement learning. The algorithm starts at the root node and traverses the tree according to the so-called tree policy, *i.e.*, PUCT [5], whose main goal is to select the best node in terms of

maximizing the estimated value and reaching a child node. For child node i, the PUCT formula can be expressed as follows, $\pi_{puct}(a|s) = argmax\,[Q(S_i,A_i)+U(S_i,A_i)]$,where the action-value Q represents the exploitation, and U represents the exploration of the un-visited paths.

(2) Expansion – Add a single child node to the search tree under the node selected in the previous step.
(3) Rollout – Based on random simulations, the default rollout policy is used with a termination outcome.
(4) Backup – After reaching the terminal state, the values of the current simulation are updated through all nodes.

The iteration of steps (1)-(4) in MCTS will continue until the search budget is exhausted. The PUCT formula strengthens the convergence of known reward nodes and encourages the exploration of those nodes that have not been visited.

3 Results & Conclusions



(a) Default reward, $\gamma = 0.85$, $C_p = 2$, maximum 200 steps

(b) Modified reward, $\gamma = 0.85$, $C_p = 2$, maximum 200 steps

Figure 2: Simulation results for single inverted pendulum on cart environment for two different rewards. Plots present the mean number of time steps for which the MCTS agent successfully controls the environment. At most, 200 steps were allowed (resulting in a total control time of 4 s). For each data point, 10 separate simulations with control were run.

The control of an inverted pendulum is a complex problem due to various physical phenomena that make it unstable, nonlinear, and under-actuated. The results shown in Figure 2 present the influence of planning horizon, available simulation budget, discount parameter γ, exploration parameter C_p, and reward type on control effectiveness. The default reward is equal to 1 when the pendulum is stabilized, see Figure 2(a). The modified reward is 1 when the pendulum is halfway or closer to a prescribed stable position and 0.5 otherwise, see Figure 2(b). Therefore, the modified reward provides an additional hint when the pendulum diverges from an upright position. Only the combination of γ and C_p giving the best outcome is presented. On both plots, a large bright and mostly continuous area of high control effectiveness is visible (the brightest color denotes episodes where the pendulum remained stable for 200 steps in every simulation for the given horizon and budget settings). It is worth noting that the modified reword contributed better to control effectiveness for shorter prediction horizons and smaller budgets.

Numerical examples show that the MCTS control agent is performing well in multibody dynamics applications. Better results can be obtained with a well-determined reward function, which reveals that significant variability in the performance of the models is firmly attributed to the value of the parameter settings. Therefore, future studies should focus on analyzing the parameter sensitivity to various multibody system dynamics applications via different AI-based algorithms.

Acknowledgments The authors wish to acknowledge CSC – IT Center for Science, Finland, for computational resources. The authors gratefully acknowledge funding provided by Business Finland's "Kumppanuusmalli – SANTTU – LUT" project under grant 8859/31/2021.

References

[1] R. Coulom. Efficient selectivity and backup operators in Monte-Carlo Tree Search. In the 5th international conference on computer and games, page 72–83, 2006.

[2] M. Morales. Grokking deep reinforcement learning. Manning publication, 2020.

[3] M.L. Puterman. Markov decision processes: discrete stochastic dynamic programming. Wiley, New York, 2013.

[4] R. Sutton and A. Barto. Reinforcement learning: an introduction, the 2nd edition. MIT press, 1998.

[5] C. D. Rosin. Multi-armed bandits with episode context. Ann. Math. Artif. Intell., 61:203–230, 2011.