JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 34 (2025) 76–80
Received: 17 October 2024
Accepted: 1 December 2024

Realization and discussion of selected artificial intelligence algorithms in computer games

Realizacja i porównanie wybranych algorytmów sztucznej inteligencji w grach komputerowych

Yurii Tyshchenko*

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

The study explores the usage of reinforcement learning algorithms in computer card games, such as Proximal Policy Optimization and Monte Carlo Tree Search. The aim is to evaluate the efficiency and learning ability across different scenarios, such as Blackjack and Poker Limit Hold'em. Comparative analysis focuses on key metrics: learning speed, stability, reward evaluation and win rate. The results highlight strengths and limitations of PPO and MCTS. Also, the potential of hybrid approaches is discussed, that combine the strategic depth of MCTS with PPO's computational efficiency to create versatile AI agents capable of excelling in diverse gaming environments. The findings underscore the importance of aligning algorithmic characteristics with task specifics and domain factors.

Keywords: games; reinforcement learning; artificial intelligence

Streszczenie

W artykule zbadano wykorzystanie algorytmów uczenia ze wzmocnieniem w komputerowych grach karcianych, takich jak Proximal Policy Optimization i Monte Carlo Tree Search. Celem jest ocena wydajności i zdolności uczenia się w różnych scenariuszach, takich jak Blackjack i Poker Limit Hold'em. Analiza porównawcza koncentruje się na kluczowych wskaźnikach: szybkości uczenia się, stabilności, ocenie nagród i współczynniku wygranych. Wyniki podkreślają mocne strony i ograniczenia PPO i MCTS. Omówiono również potencjał podejść hybrydowych, które łączą strategiczną głębię MCTS z wydajnością obliczeniową PPO w celu stworzenia uniwersalnych agentów AI zdolnych do doskonałości w różnych środowiskach gier. Wyniki badań podkreślają znaczenie dostosowania cech algorytmu do specyfiki zadania i czynników domeny.

Słowa kluczowe: gry komputerowe, uczenie ze wzmocnieniem, sztuczna inteligencja

*Corresponding author

Email address: **yurii.tyshchenko@pollub.edu.pl** (Y. Tyshchenko)

1. Introduction

The constantly changing AI landscape of computer games relentlessly fuels the search for algorithms that not only improve the players' experience but demonstrate superior adaptability and learning capabilities. This study provides a discussion on reinforcement learning and decision-making algorithms, more specifically the implementation and comparison of two critical AI algorithms—Proximal Policy Optimization (PPO) and Monte Carlo Tree Search (MCTS)—in the complicated setting of computer card games.

Card games play an excellent role in testing the efficiency of AI algorithms which proposes continuous state spaces and strategic decision making. The first goal of this study is to carefully investigate the productiveness, and learning abilities of PPO and MCTS for varying situations in card games.

The study aims to ascertain whether PPO will demonstrate superior performance in terms of learning speed and in-game stability in card games, while MCTS will excel in reward evaluation and overall win rate due to its exhaustive search approach. The comparative analysis centers around fundamental metrics: speed of learning, capacity for reward evaluation, win rate, and in-game stability.

Furthermore, this research has implications beyond gaming, as the principles could be applied to other fields requiring robust decision-making and adaptive learning in complex environments, such as autonomous driving, robotics, and financial modeling. The ultimate aim is to push the boundaries of what AI can achieve, paving the way for more intelligent and versatile AI systems.

2. Literature overview

There are many articles comparing different algorithms in card games, particularly focusing on reinforcement learning (RL) approaches. Several works have extensively compared such algorithms in card game environments.

M. Moravčík et al., in the article [1] applied MCTS in the game of Poker, developing a variant called DeepStack. They demonstrated that MCTS, combined with deep learning for evaluating game states, could outperform professional players in no-limit Texas Hold'em. This approach leverages MCTS for strategic decision-making, showing its effectiveness in handling

the complexities of Poker. Another work by N. Brown and T. Sandholm [2], introduced Libratus, an AI system for Poker that uses a combination of game theory and MCTS. Libratus employs a nested MCTS approach, where each iteration of the tree search refines the strategy, allowing it to effectively manage the vast state space and uncertainty inherent in Poker.

M. Lanctot et al. [3], conducted a comprehensive study comparing various deep reinforcement learning algorithms, including PPO, in the context of safe navigation in challenging environments. Although their primary focus was not on card games, the insights into PPO's robustness and adaptability are highly relevant. In the articles [4, 5] Barros et al., investigated the application of Proximal Policy Optimization (PPO) within the context of the Chef's Hat card game. Their studies showcased how PPO can be effectively utilized to enhance competitive performance in this specific environment. By incorporating rivalry dynamics and adapting reinforcement learning agents to learn from their opponents, they demonstrated the algorithm's capability to handle the strategic complexities of Chef's Hat. These findings highlight PPO's potential in competitive game settings, underscoring its adaptability and effectiveness in achieving high-level performance.

In the paper [6], T. Anthony, Z. Tian, and D. Barber introduced a hybrid approach. Authors combined MCTS with deep neural networks, applying this hybrid method to various strategic games. Their study demonstrated that integrating MCTS with neural networks significantly enhances performance, leveraging the strategic depth of MCTS and the learning efficiency of neural networks, similar to PPO. Article [7] by D. Silver et al., also showcased the power of combining MCTS with deep learning. Although the primary focus was on the game of Go, the principles and methodologies apply to other strategic games, including Poker and Blackjack. This work illustrated how MCTS can be augmented with deep learning techniques to achieve superhuman performance in complex games. Another article [8] by J. Popic, B. Boskovic, and J. Brest, demonstrated the efficiency of combining MCTS with deep learning in the domain of Checkers. Their study highlighted how integrating deep learning techniques with MCTS can significantly enhance performance by enabling the algorithm to better evaluate game states and make informed decisions. While their primary focus was on Checkers, the principles and methodologies they presented are applicable to any other strategic games, including Poker and Blackjack, illustrating the potential for achieving high-level performance in various complex games.

Article [9] by Larsen et al., compared various RL algorithms, including PPO and MCTS, in different game scenarios. It was discovered that while MCTS excels in strategic planning due to its thorough exploration capabilities, PPO provides superior learning speed and adaptability in dynamic environments. Their analysis in card games like Poker emphasized PPO's ability to quickly adapt to opponent strategies and changing game conditions.

These studies collectively illustrate the strengths and limitations of MCTS and PPO in card game environments. MCTS's strategic planning and thorough exploration make it ideal for complex, strategic games like Poker, whereas PPO's rapid learning and adaptability are well-suited for more straightforward, rule-based games like Blackjack. The choice of algorithm depends on the specific requirements of the game environment, including the complexity of strategies, computational resources, and the need for adaptability to changing conditions.

3. Algorithms

This section discusses the approach, employed to implement and conduct a comparative analysis of two artificial intelligence algorithms—Proximal Policy Optimization (PPO) and Monte Carlo Tree Search (MCTS)—in the context of computer card games.

The reasoning for particular algorithms being selected for this comparison is due to their different approaches, distinctive strengths, and widespread use in different areas of reinforcement learning and game AI.

By comparing these two algorithms, the article aims to highlight their respective advantages and limitations, providing a deeper understanding of their performance in different card game scenarios. This may also shed light on the potential benefits of hybrid approaches that combine their strengths.

3.1. Proximal Policy Optimization

Proximal Policy Optimization [10] is a sophisticated state-of-the-art reinforcement learning algorithm that was designed specifically to solve the problems involved in training artificial intelligence agents in complex and continuous action spaces. Introduced as an evolution of earlier policy optimization methods, PPO utilizes a new objective function that limits policy updates to avoid large policy changes, therefore contributing to the algorithm's stability during the learning process.

The PPO algorithm operates by defining a surrogate objective function, which is used to maximize the expected rewards. The specific formula for the objective function is:

$$E_t\left[min\left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \quad (1)$$

where:

- $r_t(\theta)$ represents the probability ratio between the new and old policies,
- $\hat{A}_t$ is the advantage estimate at the time step t,
- ϵ is a small hyperparameter that defines the clipping range ensuring that $r_t(\theta)$ stays within $[1-\epsilon, 1+\epsilon]$.

By clipping the probability ratio, PPO ensures that the new policy does not deviate too much from the old policy, thereby maintaining stability and improving convergence rates.

PPO was chosen due to its effectiveness in handling continuous action spaces and maintaining stability during

the learning process. PPO was set up with $\gamma = 0.99$, $1e4$ epochs and $lr = 0.0001$.

The training process for the PPO involves continuously updating the policy based on the collected experiences from the environment. The agent interacts with the environment, collects rewards, and updates its policy using the PPO objective function. This iterative process allows the agent to improve its performance over time by learning from its past experiences.

The entire process consists of 20000 episodes, where the agent undergoes training across all episodes. Every 500 episodes, the agent participates in 100 matches in a tournament setting for evaluation purposes, which results in a total of 4000 evaluation games played.

3.2. Monte Carlo Tree Search

As a robust and widely used decision-making algorithm, Monte Carlo Tree Search (MCTS) was designed to solve complex problems in environments with uncertainty. The algorithm is renowned for its strategic planning capabilities and flexibility, making it particularly effective in dynamic environments with evolving strategies. Its ability to traverse extensive decision spaces and make informed choices based on simulated outcomes makes it a strong choice for games that require deep strategic thinking and adaptability.

The algorithm uses a standard Upper Confidence Bound $UCB1 = \frac{\omega_i}{n_i} + c\sqrt{\frac{2lnN}{n_i}}$ for tree policy expansion, where:

- ω_i is the number of wins for the child i,
- n_i is the number of visits to the child i,
- N is the total number of visits to the parent node,
- $c = 0.1$ is the exploration parameter.

Because of the nature of the algorithm, training is not required in tree search, so only 4000 episodes instead of 20000 and thus games are played. The episode number for MCTS is later tweaked, for ease of graphical showcase, but it accurately describes actual evaluation for the point in episodes.

4. Game Environments

In this section of the study, two distinct card games were selected to evaluate the performance of PPO and MCTS: Blackjack and Poker (specifically Limit Texas Hold'em).

These games were chosen due to their contrasting characteristics: Blackjack offers a high-dimensional state space and requires quick adaptation and efficient learning curves, making it well-suited for evaluating PPO. In contrast, Poker, with its complex strategic depth and dynamic environment, provides an ideal setting to assess the strategic planning and adaptability strengths of MCTS.

To create a consistent and controlled environment for experiments, the "rlcard" library was utilized, a Python toolkit specifically designed for reinforcement learning [12]. This library offers standardized implementations of various card games, as well as helpful data logging, tournaments, helpful random agents, and other useful utilities to ensure a reliable platform for comparative analysis.

4.1. Blackjack

Blackjack, also known as 21, is a classic card game where players compete against the dealer, aiming to have a hand value as close to 21 as possible without exceeding it. To make the job a little easier, the game is set to have two possible actions: hit and stand, without doubling down. The simplicity of rules, combined with the strategic decisions, makes Blackjack an ideal environment for testing and comparing AI algorithms, specifically challenges in decision-making and risk assessment.

Reward is calculated by the simplest logic. Awarding 1 point when a dealer busts or the player's score is higher than the dealer's yet less than or equal to 21. An opposite outcome results in -1 for the player. A tie does not reward or punish contender, resulting in 0 points.

4.2. Poker (Limit Hold'em)

Poker [11], particularly the variant of Limit Texas Hold'em, introduces a higher level of complexity compared to Blackjack. Players are dealt two private cards and share five public ones, dealt in multiple rounds. The game requires adaptive strategies to navigate through betting, card dealing uncertainties, and the behavior of the opponents. Thus, Limit Hold'em provides a rich and dynamic environment for evaluating the strategic capabilities of artificial intelligence (AI) algorithms.

For the reward, the standard calculation of the "rlcard" library was taken. The payoff is generated based on the hands of each player, and the amount of chips that were bet by a particular player. If both players have the same combination level, the reward is split considering the bets.

5. Results analysis

For the analysis, Blackjack games are played 1 on 1 between the MCTS or PPO agent and the dealer, while Poker Limit Hold'em games are played between the selected agent and random agent available in the "rlcard" library.

Results consist of mean, median, and standard deviation for each algorithm's performance in every card game. Additionally, the comparison includes an analysis of mean and median decision time for the used methods, providing insights into their computational efficiency in given environments.

Figure 1 illustrates the variation in rewards obtained by two different algorithms, Proximal Policy Optimization and Monte Carlo Tree Search, over a series of 20000 episodes in the Blackjack environment. The x-axis represents the episode number, while the y-axis shows the obtained reward. The graph demonstrates a fluctuating reward pattern for both algorithms, indicating the inherent variability in the Blackjack game.

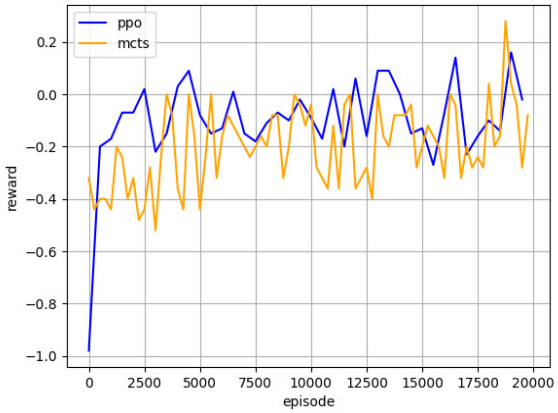

Figure 1: Change of calculated reward for MCTS and PPO in Blackjack over episodes.

Table 1: Mean, median, and standard deviation of rewards for Blackjack

Algorithm	Mean	Median	σ
PPO	-0.1	-0.1	0.18
MCTS	-0.2	-0.2	0.15

Table 1 provides statistical insights into the performance of PPO and MCTS. It includes the mean, median, and standard deviation (σ) of rewards across all episodes:

- The mean reward for PPO is -0.1, compared to -0.2 for MCTS, suggesting that PPO generally performs slightly better on average,
- Both algorithms have the same median reward of -0.1, indicating that the central tendency of their rewards is similar,
- The standard deviation of rewards is higher for PPO (0.18) compared to MCTS (0.15), which implies that PPO experiences more variability in rewards.

Win rates shown in Table 2 provide further evidence that PPO is generally more effective than MCTS in this particular environment. The close win rate of PPO to 50% also indicates a near-balanced performance, whereas the lower win rate of MCTS suggests it struggles to find the correct action.

Table 2: Win statistics for algorithms in Blackjack

Algorithm	Wins	Loses	Win %
PPO	1979	2021	49.48%
MCTS	1597	2403	39.93%

Figure 2, shows similar information on the Limit Hold'em episodes. The reward line representing PPO is visibly balanced in terms of rewards, compared to chaotic fluctuations of the MCTS evaluation line, which is supported by Table 3. The higher standard deviation for MCTS indicates significant variability in its performance. This suggests that while MCTS can achieve high rewards, it does so inconsistently.

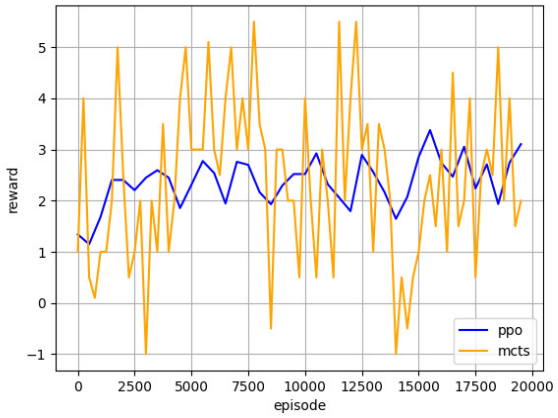

Figure 2: Change of calculated reward for MCTS and PPO in Limit Hold'em Poker over episodes.

Table 3: Mean, median, and standard deviation of rewards for Limit Hold'em

Algorithm	Mean	Median	σ
PPO	2.34	2.41	0.48
MCTS	2.31	2.5	1.79

Statistics presented in Table 4, show that despite having greater variability in performance (reflected by a greater standard deviation), similar mean results with a slightly larger median, the percentage of wins is 14.58% greater for the MCTS algorithm. While PPO offers more stable and balanced performance in terms of rewards, MCTS is more successful in terms of overall win rate despite its higher variability. It suggests that the algorithm has a strong capability to win games, with less predictability in reward outcomes.

Table 4: Win statistics for algorithms in Limit Hold'em Poker

Algorithm	Wins	Loses	Win %
PPO	2006	1994	50.15%
MCTS	2589	1411	64.73%

Efficiency in decision-making is crucial for real-time applications. Because of the algorithmic differences between MCTS and PPO, where the latter would first train and then almost instantly output the best action, two PPO time evaluations were presented: decision only and a sum of decision-making and learning.

As presented in Table 5, decision-making time is fast, varying from only 2 to 14 milliseconds depending on the

game and algorithm. For a Limit Hold'em environment, the PPO makes decisions around 3 times faster compared to MCTS. Time to train during the episode flips the results, as now PPO is around 2.5 times slower with training, making 86% of the time. As for Blackjack, Monte Carlo Tree Search has no changes in time compared to Poker, yet PPO spends 3 milliseconds less training.

Table 5: Mean and median for MCTS decision time and PPO decision time + learning time

	Blackjack		Limit Hold'em	
	Mean (ms)	Median (ms)	Mean (ms)	Median (ms)
PPO (decide)	2	2	2	2
PPO (learn + decide)	11	11	14	14
MCTS	5	5	6	5

6. Conclusion

The choice between MCTS and PPO depends on the specific requirements and constraints of the application. MCTS excels in strategic planning and adaptability, making it ideal for complex, strategic games like Poker, where it can effectively handle dynamic environments and evolving strategies. On the other hand, PPO offers computational efficiency and a more desirable learning curve as well as stability in decision-making, being suitable for tasks with high-dimensional state spaces and continuous action spaces, such as Blackjack.

A hybrid approach that combines the strengths of both MCTS and PPO may provide a promising direction for future research, allowing for flexible AI agents capable of addressing a wide range of game scenarios and challenges. Such an approach could leverage MCTS's strategic depth and adaptability alongside PPO's rapid learning and computational efficiency, potentially enhancing overall performance in diverse gaming environments.

In summary, the comparative analysis underscores the importance of considering algorithmic characteristics, task requirements, and domain-specific factors when selecting an artificial intelligence approach for card game environments. While MCTS and PPO are not the newest approaches they have laid the foundation for newer methods and technologies. They continue providing insights into modern AI development and their operation. Further exploration and experimentation alongside the ongoing development in this field promises exciting advancements in AI's ability to understand and master complex strategic environments.

References

[1] M. Moravčík, M. Schmid, N. Burch, V. Lisý, D. Morrill, N. Bard, T. Davis, K. Waugh, M. Johanson, M. Bowling, Deepstack: Expert-level artificial intelligence in heads-up no-limit poker, Science 356 (2017) 508–513, https://doi.org/10.1126/science.aam6960.

[2] N. Brown, T. Sandholm, Superhuman AI for heads-up no-limit poker: Libratus beats top professionals, Science 359 (2018) 418-424, https://doi.org/10.1126/science.aao1733.

[3] M. Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Perolat, D. Silver, T. Graepel, A Unified Game-Theoretic Approach to multiagent Reinforcement learning, Neural Information Processing Systems 30 (2017) 4190–4203, https://doi.org/10.48550/arXiv.1711.00832.

[4] P. Barros, O. Yalçın, A. Tanevska, A. Sciutti, Incorporating rivalry in Reinforcement Learning for a competitive game, Neural Computing and Applications 35 (2022) 16739-16752, https://doi.org/10.1007/s00521-022-07746-9.

[5] P. Barros, A. Tanevska, A. Sciutti, Learning from learners: Adapting reinforcement learning agents to be competitive in a card game, 2020 25th International Conference on Pattern Recognition (ICPR) (2020) 2716-2723, https://doi.org/10.1109/icpr48806.2021.9412807.

[6] T. Anthony, Z. Tian, D. Barber, Thinking Fast and Slow with Deep Learning and Tree Search, NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems (2017) 5366-5376, https://doi.org/10.48550/arXiv.1705.08439.

[7] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of go with deep neural networks and Tree Search, Nature 529 (2016) 484–489, https://doi.org/10.1038/nature16961.

[8] J. Popic, B. Boskovic, J. Brest, Deep learning and the game of Checkers, MENDEL 27 (2021) 1–6, https://doi.org/10.13164/mendel.2021.2.001.

[9] T. Larsen, H. Teigen, T. Laache, D. Varagnolo, A. Rasheed, Comparing deep reinforcement learning algorithms' Ability to safely navigate challenging waters, Frontiers in Robotics and AI 8 (2021) 738113, https://doi.org/10.3389/frobt.2021.738113.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, CoRR (2017), https://arxiv.org/abs/1707.06347.

[11] J. Rubin, I. Watson, Computer poker: A Review, Artificial Intelligence 175 (2011) 958-987, https://doi.org/10.1016/j.artint.2010.12.005.

[12] D. Zha, K. Lai, S. Huang, Y. Cao, K. Reddy, J. Vargas, A. Nguyen, R. Wei, J. Guo, X. Hu, RLCard: A platform for reinforcement learning in Card Games, Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (2020) 5264-5266, https://doi.org/10.24963/ijcai.2020.