# Efficient Monte-Carlo Tree Search in Deterministic Environments

**Cedric Derstroff**
Hessian Center for Artificial Intelligence (hessian.AI)
Technische Universität Darmstadt
64289 Darmstadt
Germany
cedric.derstroff@tu-darmstadt.de

**Jannis Brugger**
Hessian Center for Artificial Intelligence (hessian.AI)
Technische Universität Darmstadt
jannis.brugger@tu-darmstadt.de

**Jannis Blüml**
Hessian Center for Artificial Intelligence (hessian.AI)
Technische Universität Darmstadt
jannis.blueml@tu-darmstadt.de

**Mira Mezini**
Hessian Center for Artificial Intelligence (hessian.AI)
National Research Center for Applied Cybersecurity ATHENE
Technische Universität Darmstadt
mira.mezini@tu-darmstadt.de

**Stefan Kramer**
Johannes Gutenberg-Universität Mainz
55122 Mainz
Germany
kramer@informatik.uni-mainz.de

**Kristian Kersting**
Hessian Center for Artificial Intelligence (hessian.AI)
Centre for Cognitive Science
German Research Center for Artificial Intelligence (DFKI)
Technische Universität Darmstadt
kristian.kersting@tu-darmstadt.de

## Abstract

Monte-Carlo Tree Search (MCTS) is a powerful algorithm for decision-making and planning, known for its ability to balance exploration and exploitation. However, in deterministic environments, MCTS often inefficiently revisits previously explored nodes, limiting its effectiveness and computational efficiency. This is particularly problematic in large and complex search spaces where broader exploration is critical. To address these limitations, we introduce AmEx-MCTS (Amplified Exploration Monte-Carlo Tree Search), an enhanced MCTS framework designed to avoid redundant exploration of completely explored subtrees systematically. AmEx-MCTS achieves this by decoupling three key components of the search process: value updates, visit count increments, and path selection. It redirects computational resources to unexplored regions, enhancing search coverage and precision without increasing computational costs by maintaining accurate visit counts and excluding terminal subtrees from further consideration. Theoretical analysis guarantees convergence properties equivalent to classical MCTS while demonstrating improved performance in deterministic environments. Empirical results validate these advantages across benchmark domains, including FrozenLake, compiler optimization tasks, and equation discovery. AmEx-MCTS consistently outperforms classical MCTS and other baselines, with up to 20 percent improved computational efficiency and superior discovery of optimal solutions. These results underscore AmEx-MCTS as a versatile tool for solving decision-making problems in deterministic settings, applicable to areas such as artificial intelligence, optimization, and complex systems modeling.

**Keywords:** Monte-Carlo Tree Search, Deterministic MDPs, Exploration

**Acknowledgements**

This research project was partly funded by the Hessian Ministry of Science and the Arts (HMWK) within the projects "The Third Wave of Artificial Intelligence - 3AI" and hessian.AI.

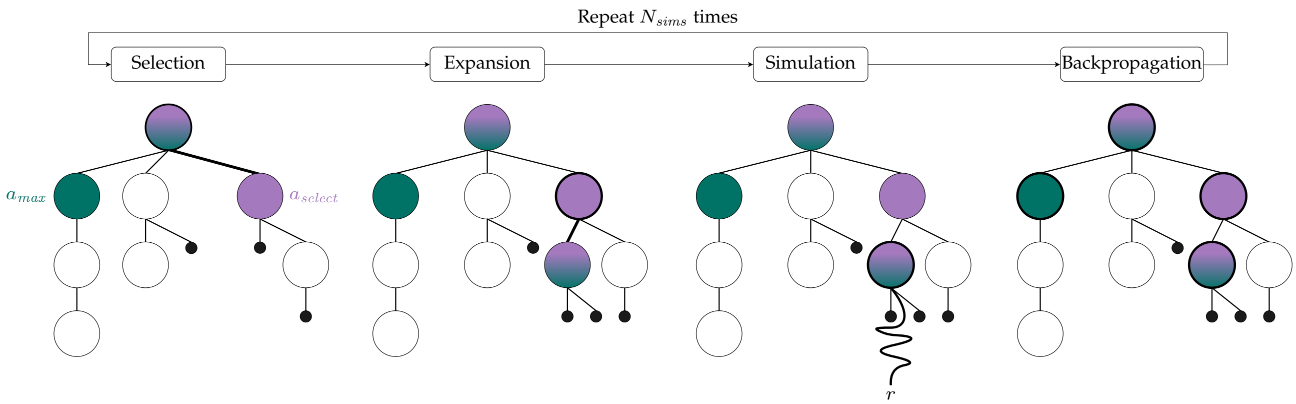


Figure 1: **Improving MCTS by ignoring already explored subtrees and leaves by focusing on the unknown.** Updating the search strategy within MCTS by separating "incrementing visit counts" (displayed in dark green) from the selected path (displayed in light purple) to explore more while keeping the number of iterations $N_{sims}$ the same. Nodes in the selected path are colored light purple, and nodes that classical MCTS would have chosen are marked dark green.

## 1 Introduction

Monte-Carlo Tree Search (MCTS) has emerged as one of the most prominent algorithms for decision-making and planning in AI. Its iterative approach to exploring large search spaces, balancing exploration of new possibilities with exploitation of promising paths, has made it a versatile tool across numerous domains, including board games, robotics, and optimization tasks (Coulom, 2007; Kocsis, Szepesvári, 2006). The strength of MCTS lies in its ability to adaptively allocate computational resources where they are most needed, constructing a search tree dynamically to approximate the optimal solution.

MCTS has seen extensive advancements, with numerous adaptations improving its performance across various domains. For instance, Score-Bounded MCTS and MCTS-Solver focus on pruning suboptimal paths, but these methods are often limited to specific settings such as two-player games (Cazenave, Saffidine, 2010; Winands et al., 2008). Techniques like black box planning and oversubscription planning emphasize resource allocation and uncertainty management but operate under fundamentally different assumptions (Benton et al., 2009; Hoffmann, Brafman, 2005). More directly related, modifications like uncertainty-driven exploration have been proposed to enhance the efficiency of MCTS, yet these approaches frequently alter the algorithm's exploration-exploitation tradeoff (Moerland et al., 2020).

Despite its success, classical MCTS suffers from inefficiencies in deterministic environments, where the outcome of each action is entirely predictable. In such cases, MCTS often revisits previously explored nodes and subtrees, wasting computational effort on areas that contribute no new information. This issue is especially pronounced in domains with large action spaces and numerous terminal nodes, where thorough exploration is critical but computationally expensive. These limitations can hinder the algorithm's ability to address increasingly complex, high-dimensional problems.

To address these challenges, we introduce AmEx-MCTS (Amplified Exploration Monte-Carlo Tree Search)—visualized in Fig. 1—an enhancement of the classical MCTS framework. AmEx-MCTS systematically avoids revisiting completely explored subtrees by decoupling value updates, visit counts, and path selection during the search process. This decoupling allows AmEx-MCTS to redirect computational resources toward unexplored areas of the decision space without increasing the overall computational budget. As a result, AmEx-MCTS achieves broader search coverage and more accurate evaluations while preserving the foundational exploration-exploitation tradeoff of classical MCTS.

The significance of AmEx-MCTS extends beyond the field of artificial intelligence. By improving search efficiency and decision-making accuracy, it has the potential to address interdisciplinary challenges in areas such as biology, where it can optimize experimental designs or analyze biological networks, and engineering, where it can assist in structural optimization or scheduling (Moerland et al., 2020; Zheng et al., 2021). Moreover, its compatibility with deterministic Markov decision processes (MDPs) makes it a powerful tool for tasks requiring precise solutions, such as compiler optimization or chemical synthesis (Cummins et al., 2021; Devata et al., 2024).

We demonstrate its superior performance through empirical evaluations on benchmark tasks, including deterministic planning problems, equation discovery, and FrozenLake. The results show that AmEx-MCTS significantly outperforms classical MCTS and related approaches, achieving up to 20 percent improvements in search efficiency and solution quality. These findings underscore the potential of AmEx-MCTS as a robust framework for solving complex decision-making problems across diverse fields.

## 2 AmEx-MCTS

AmEx-MCTS (Amplified Exploration Monte-Carlo Tree Search) builds on the foundational structure of classical MCTS while addressing its inefficiencies in deterministic environments by focusing computational resources on unexplored regions of the search space. A central feature of the algorithm is its ability to systematically track and exclude completely explored subtrees from further consideration, ensuring that every simulation contributes new information to the search process. This is achieved without disrupting the foundational balance between exploration and exploitation that characterizes classical MCTS. By maintaining a dynamically updated list of not completely explored subtrees (NCES) for each node, AmEx-MCTS redirects computational effort toward areas of the search space that remain uncertain. This focus enhances search coverage and accelerates convergence while preserving the algorithm's compatibility with existing MCTS frameworks. Additionally, the lightweight modifications in AmEx-MCTS ensure minimal computational overhead, making it an efficient and versatile solution for deterministic decision-making problems.

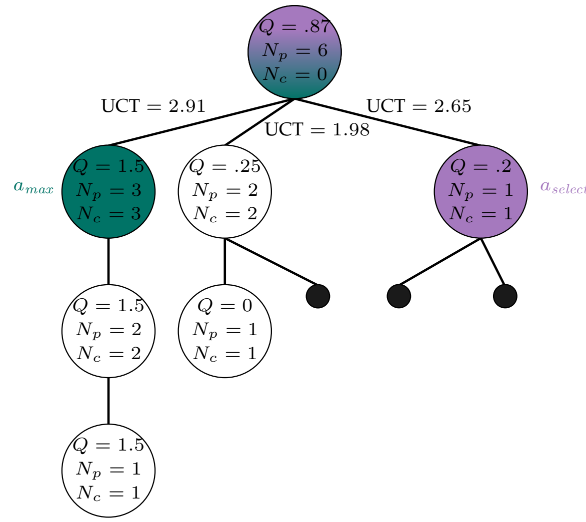


Figure 2: **Ignoring completely explored subtrees within the search.** AmEx-MCTS introduces two new variables $a_{max}$ and $a_{select}$ to differentiate between completely explored subtrees and such that are not. Selecting $a_{select}$ ignores the already completely explored left subtree, while $a_{max}$ would only lead to already known states. $N_p$ describes how often the node was visited within the search, $N_c$ the number of how often it had the highest UCT value among its siblings, and $Q$ the value of the state. The selected path is light purple, and the nodes classical MCTS would have chosen are dark green.

The modifications in AmEx-MCTS are implemented across the four standard phases of MCTS: selection, expansion, simulation, and backpropagation. These changes allow AmEx-MCTS to focus computational resources on unexplored regions of the decision space, enhancing its performance in deterministic decision-making tasks.

### 2.1 Algorithmic Enhancements

The key innovation in AmEx-MCTS lies in decoupling three components of the search process: value updates, visit counts, and path selection. These modifications are integrated into each phase of the algorithm:

**Selection Phase** AmEx-MCTS introduces a distinction between actions leading to unexplored subtrees and those associated with already completely explored subtrees (cf. Fig. 2). During the selection phase, the algorithm prioritizes unexplored actions, ensuring that computational resources are directed toward gathering new information. This is achieved by defining a new selection candidate, $a_{select}$, which considers only the actions leading to partially or unexplored subtrees while still updating visit counts for $a_{max}$, the action with the highest utility under classical MCTS.

**Expansion Phase** Nodes are expanded as in classical MCTS but with additional checks to avoid redundancy. A transposition table is used to identify whether the current state has already been visited elsewhere in the tree. If the state is already present, the expansion step skips the simulation and directly proceeds to backpropagation, reusing the stored value for the node. This prevents duplicate evaluations of identical states reached through different paths.

**Simulation Phase** The simulation phase remains unchanged from classical MCTS.

**Backpropagation Phase** The backpropagation phase in AmEx-MCTS introduces separate updates for two sets of visit counts: those associated with the actual selected path ($a_{select}$) and those that classical MCTS would have chosen ($a_{max}$). This ensures that the algorithm maintains accurate statistics for guiding future exploration while avoiding the propagation of redundant information from completely explored subtrees. Additionally, rewards from terminal nodes are used to label subtrees as "completely explored," preventing further revisits.

### 2.2 Theoretical Guarantees

AmEx-MCTS preserves the convergence and concentration properties of classical MCTS while accelerating convergence in deterministic environments. The algorithm achieves faster and more accurate estimates of the optimal policy by reducing redundant evaluations. These improvements are particularly pronounced in domains with high branching factors and numerous terminal nodes, where efficient resource allocation is critical.

## 3 Results

The performance of AmEx-MCTS was evaluated across a diverse set of deterministic benchmark tasks, demonstrating noteworthy improvements over classical MCTS and related variants. In the FrozenLake environment, AmEx-MCTS consistently achieved broader search coverage and higher solution quality than all baselines. Figure 3 shows that it reached optimal solutions with significantly fewer simulations by efficiently avoiding revisits to completely explored subtrees. In chain-like environments, including Chain and ChainLoop, AmEx-MCTS outperformed state-of-the-art approaches such as MCTS-T and MCTS-T+, particularly in cases with large state-action spaces, where the broader exploration afforded by AmEx-MCTS proved critical (cf. Fig. 4).

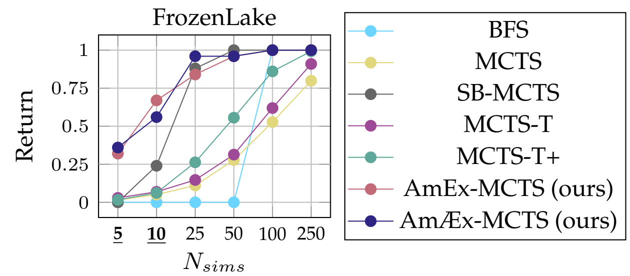


Figure 3: **Our approach dominates on the FrozenLake environment** as used by Moerland et al. (2020). Higher values are better. Baseline results are taken from the original paper. As the outcome of one run is binary, this represents complete information about the 25 runs of the comparison methods. The x-label is **highlighted** when the lower-performing version of our approach is significantly better ($\alpha = 0.05$).

In the CompilerGym domain, AmEx-MCTS excelled in optimizing program size, achieving up to 20 percent higher computational efficiency compared to Score-Bounded MCTS (SB-MCTS), a variant designed for deterministic settings. These improvements were statistically significant across all tested compiler tasks, highlighting AmEx-MCTS's ability to allocate computational resources in large-scale, complex environments effectively.

The algorithm's versatility was further validated in an equation discovery task, where AmEx-MCTS explored the search space more comprehensively than classical MCTS. Unlike its counterparts, which often revisited already evaluated regions, AmEx-MCTS consistently discovered the globally optimal solution with fewer simulations. Even in high-branching-factor environments like Crazyhouse chess, the algorithm expanded approximately 10 percent more nodes in the search tree, maintaining nearly identical computational speed compared to classical MCTS.

These results underline its broad applicability to deterministic problems, making it a robust and efficient framework for tackling complex decision-making tasks in various fields.

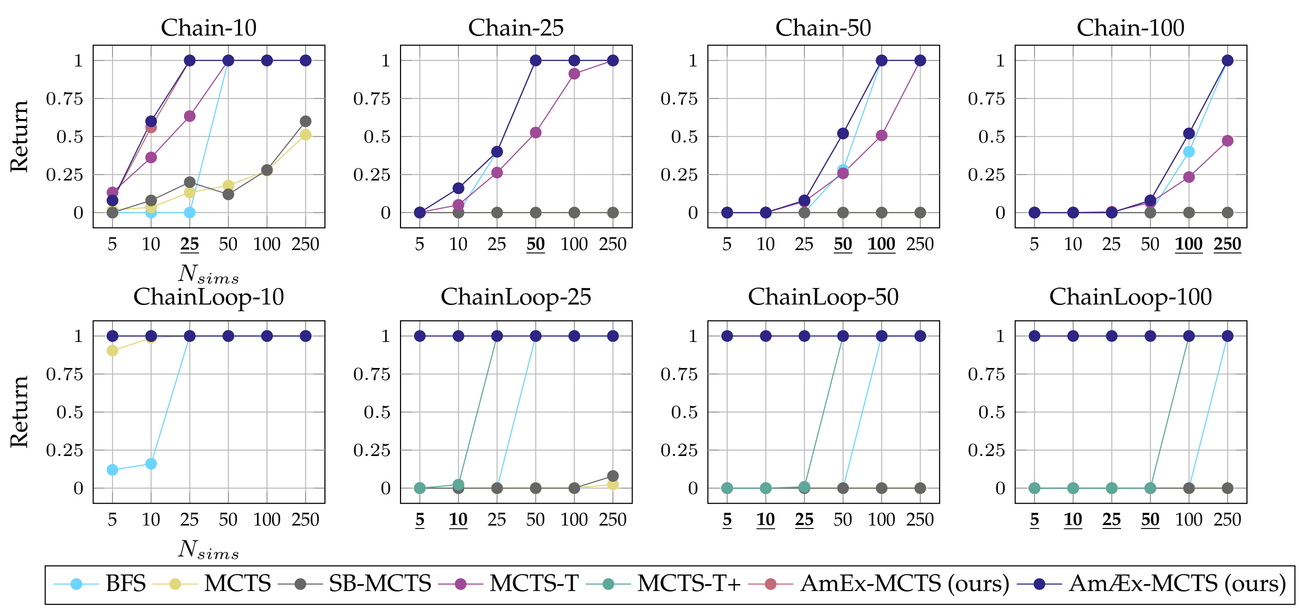


Figure 4: **Our approach strongly outperforms the baselines on the Chain environment and achieves optimal results on the ChainLoop environment** as used by Moerland et al. (2020). Higher values are better. Baseline results are taken from the original paper. As the outcome of one run is binary, this represents complete information about the 25 runs of the comparison methods. The x-label is **highlighted** when the lower-performing version of our approach is significantly better ($\alpha = 0.05$) than the MCTS baselines.

## 4 Broader Impact

AmEx-MCTS offers significant potential for advancing RL and other fields by addressing inefficiencies in classical MCTS. Its robust framework for decision-making has applications in RL tasks, deterministic MDPs, and diverse domains like biology, where it can optimize experimental designs, and engineering, where it can aid in structural optimization and scheduling. Its flexibility also supports interdisciplinary uses, including economic modeling and computational social science. The algorithm's efficiency reduces computational costs, potentially lowering environmental impact and enabling safer, more accurate decision-making in areas like healthcare and autonomous systems. However, ethical considerations must be addressed to ensure fairness, transparency, and accountability, particularly in sensitive fields like finance or surveillance. AmEx-MCTS provides a foundation for advancing scientific discovery and solving complex problems but requires responsible deployment to maximize benefits and minimize risks.

**Limitations** AmEx-MCTS is designed explicitly for deterministic environments, limiting its direct applicability to stochastic or highly uncertain domains without significant adaptation. Its reliance on a transposition table for reusing states may become memory-intensive in tasks with large state spaces, requiring optimization for scalability. Additionally, the algorithm assumes non-negative rewards for non-terminal states, which may not align with all problem settings. While these enhancements add minimal computational overhead, they may not provide significant advantages in more straightforward problems where classical MCTS suffices. Despite these limitations, AmEx-MCTS offers substantial benefits in deterministic tasks and presents opportunities for future extensions to more complex settings.

## 5 Conclusion

AmEx-MCTS addresses key inefficiencies of classical MCTS in deterministic domains by systematically avoiding redundant revisits to completely explored subtrees. Its innovative design improves search efficiency and solution quality while preserving the foundational strengths of MCTS. Through extensive evaluation on diverse benchmarks, AmEx-MCTS demonstrates its potential as a robust and scalable tool for solving complex decision-making tasks across various disciplines. Enhancing computational efficiency and enabling broader exploration sets the stage for advancements in reinforcement learning, optimization, and interdisciplinary applications.

**Future Work** Future research will explore extending AmEx-MCTS to stochastic and partially observable environments, where uncertainty plays a central role. Incorporating mechanisms to handle probabilistic transitions and outcomes will be a key focus. Additionally, scaling the algorithm for large state-action spaces using memory-efficient strategies, such as compact state representations, will enhance its applicability to real-world problems. Integration with neural networks, akin to AlphaZero-like frameworks, could further strengthen its utility in domains like autonomous systems, game-playing, and complex scientific modeling. Finally, ethical considerations, particularly in high-stakes applications, will remain essential for ensuring responsible and transparent use of the algorithm.

## References


*Benton J., Do Minh, Kambhampati Subbarao*. Anytime heuristic search for partial satisfaction planning // Artificial Intelligence. 2009. 562–592.

*Cazenave Tristan, Saffidine Abdallah*. Score bounded Monte-Carlo tree search // International conference on computers and games. 2010. 93–104.

*Coulom Rémi*. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search // Computers and Games. Berlin, Heidelberg: Springer, 2007. 72–83. (Lecture Notes in Computer Science).

*Cummins Chris, Wasti Bram, Guo Jiadong, Cui Brandon, Ansel Jason, Gomez Sahir, Jain Somya, Liu Jia, Teytaud Olivier, Steiner Benoit, Tian Yuandong, Leather Hugh*. CompilerGym: Robust, Performant Compiler Optimization Environments for AI Research. 2021.

*Devata Sriram, Sridharan Bhuvanesh, Mehta Sarvesh, Pathak Yashaswi, Laghuvarapu Siddhartha, Varma Girish, Priyakumar U. Deva*. Deep-SPInN – deep reinforcement learning for molecular structure prediction from infrared and 13C NMR spectra // Digital Discovery. 2024. 3. 818–829.

*Hoffmann Jörg, Brafman Ronen*. Contingent planning via heuristic forward search with implicit belief states // Proc. ICAPS. 2005.

*Kocsis Levente, Szepesvári Csaba*. Bandit Based Monte-Carlo Planning // European conference on machine learning. 2006. IX 2006. 282–293.

*Moerland Thomas M., Broekens Joost, Plaat Aske, Jonker Catholijn M.* The Second Type of Uncertainty in Monte Carlo Tree Search. V 2020.

*Winands Mark H. M., Björnsson Yngvi, Saito Jahn-Takeshi*. Monte-Carlo Tree Search Solver // Computers and Games. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. 25–36.

*Zheng Guangcong, Wang Cong, Shao Weijie, Yuan Ying, Tian Zejie, Peng Sancheng, Bashir Ali Kashif, Mumtaz Shahid*. A single-player Monte Carlo tree search method combined with node importance for virtual network embedding // Annals of Telecommunications. VI 2021. 76, 5-6. 297–312.