Game Playing with Monte-Carlo Tree Search

Project Plan

Alexandra Danciu

1. Abstract

The goal of this project is to design and implement a game-playing program using Monte-Carlo Tree Search. The game I have decided to implement using the MCTS is Connect4; Connect4 is a two-player strategy board game where players take turns dropping coloured discs into a grid to be the first to get four in a row.

Monte-Carlo Tree Search is an innovative heuristic search method that has revolutionized decision-making in complex game trees. MCTS has demonstrated remarkable success in challenging domains, such as Go, where traditional algorithms like alpha-beta pruning struggle due to high branching factors.

The main motivation behind this project is to create a game-playing program that is able to successfully win a game of Connect4. The algorithm for this program should play more efficiently than if using alpha-beta pruning. I will complete some experimental evaluations of the performance of this game to back up the previous statement. I will also be creating a user-friendly interface, the program will allow players to experience the capabilities of the AI firsthand, enhancing their understanding of MCTS and its practical implications in game strategy.

These are some of the individual project goals I have for this project:

1. Implementation of Bandit Problem Solver:
   * The project will commence with the development of proof-of-concept programs to tackle bandit problems, assessing the effectiveness of Upper Confidence Bound (UCB) methods against naive strategies.
2. Human-Only Gameplay:
   * Create a functional program allowing two players to play Connect4 without AI involvement. This will lay out the foundations for understanding game mechanics and state transitions, and for creating an easy-to-use and user-friendly GUI.
3. Game Rules Documentation:
   * Formulate semi-formal documentation detailing the rules of Connect4, including state representations and allowable transitions. This ensures clarity in the game design.
4. Development of MCTS Algorithm:
   * Implement the MCTS algorithm, encapsulating its core components, selection, expansion, simulation, and backpropagation, within a well-structured, object-oriented design.
5. User Interface Design:
   * Design and implement a GUI that supports both human VS human and human VS AI gameplay. The interface will focus on user experience, providing intuitive controls and visual feedback.
6. Experimental Evaluation:
   * Conduct extensive experimental evaluations to analyse the performance of MCTS based on various parameters, such as search depth and the number of rollouts.
7. Software Engineering Documentation:
   * Compile a comprehensive report detailing the software engineering process, design decisions, and challenges faced during development. This will include discussions on the theoretical foundations of bandit processes and MCTS, as well as justifications for the GUI design.
8. Timeline
   1. Term 1

Weeks 1 & 2,

30/09/2024 – 13/10/2024:

I will spend the first two weeks of the term doing the project plan, ensuring I have a good understanding of the project and having basic research done about some of the more complex topics.

Weeks 3 – 6,

14/10/2024 – 10/11/2024:

I will spend these four weeks working on the bandit problem solver proof of concept program. I will spend the first week doing in depth research and planning out my program. The following three weeks will be spent coding and creating the concept program and testing it. I will also be working on the report for the UCB method for bandit problems during this section.

Weeks 7 – 10,

11/11/2024 – 08/12/2024:

I will spend these four weeks working on the Connect4 proof of concept program. The first week will be spent doing some more research into how to code a Connect4 game. The following three weeks will be spent coding the concept program and then testing it. I will also be working on the report for the rules of the game, explaining the states and allowable transitions and how I used these to create the proof of concept.

Weeks 10 & 11,

02/12/2024 – 13/12/2024:

These last two weeks will be spent working on the report for the algorithm for Monte-Carlo Tree Search and preparation for the deadline, ensuring that everything is complete.

* 1. Term 2

Weeks 1 – 6,

13/01/2025 – 23/02/2025:

These six weeks will be spent working on the programming aspect of the final project. Using the reports and proofs of concept from term 1, I will set out a plan on how to complete the final program deliverable. Ensuring the final project deliverable has a full object-oriented design and has a GUI that is easy to use for users. The final project will also allow both player vs player gameplay and player vs AI gameplay in the same program.

Weeks 7 – 11,

24/02/2025 – 04/04/2025:

These last five weeks will be spent working on the report. Ensuring that it meets all the requirements for the final deliverables in the project specification:

1. The report should include experimental assessments of effects on program performance of parameters of MCTS (depth of search, number of rollouts used, etc).
2. The report will describe the software engineering process involved in generating your software.
3. The report will have sections on: the theory of bandit processes, and the algorithm for MCTS; the game used, the software engineering process in development of the program; an explanation and justification of design decisions for the GUI; and an experimental analysis of the performance of MCTS.
4. Risk assessment

* Complexity of MCTS Implementation
  + Risk: MCTS is a complex algorithm with various components (selection, expansion, simulation, backpropagation) that may lead to implementation challenges and the final project not being to the full quality standards.
  + Mitigation: Start with a simplified version of MCTS and iteratively enhance its complexity. Utilize modular coding practices to isolate and test components. Do research on how MCTS works and how it goes through the processes. Research on coded implementations.
* Game Complexity
  + Risk: The chosen game, Connect4, has rules that complicate state representation and transitions. Planning how the algorithm will decide what the next best move will be.
  + Mitigation: Create a thorough understanding of the game rules and design a clear state transition model before coding. Do research on the best game winning strategy for the game to understand what steps the AI should be following.
* Timeline Underestimation
  + Risk: The project may take longer than anticipated due to unforeseen challenges. Human error in the estimation of tasks that can lead to some taking longer than expected.
  + Mitigation: Break the project into phases with clear milestones and allow buffer time for each phase.
* Hardware Malfunction
  + Risk: I will be working on this project mainly on my home PC, which means if any malfunctions happen, I will be left without access to my work and perhaps loss of certain work.
  + Mitigation: Use the provided Git Repo to back up all my work including text documentation. In the case of loosing access to work from home I can use any PC provided by the university and pull from Git Repo to re-gain access to my work and files.
* Skill Gaps
  + Risk: I personally may lack the skills in advanced algorithm design or GUI development that are necessary to complete the project.
  + Mitigation: Giving myself buffer time in the project timeline to be able to research and ask for help in solving the relevant problems while still staying on track with the planning.
* User Interface Usability
  + Risk: The GUI may not be user-friendly, leading to poor user experience.
  + Mitigation: Conduct user testing and gather feedback early in the design phase to refine the interface.
* Inadequate Experimental Design
  + Risk: The experiments to assess MCTS performance might not be well-structured, leading to inconclusive results.
  + Mitigation: Develop a clear experimental framework and review it with peers before execution.
* Documentation Gaps
  + Risk: Poor documentation of the development process and decisions may impact the final documentation and report for the project.
  + Mitigation: Maintain thorough documentation practices throughout the project lifecycle, such as consistently using the diary.

1. Bibliography

Key for references:

Name of person posting video (Year video posted) Title of film or programme. Available at: URL

Website author (Year published/Last updated) *Title of internet site*. Available at: URL

Bandit Solver Problem:

I have used a combination of websites and YouTube videos to explain the bandit solver problem to me, so I have a wide range of understanding on it, from simple and vague to more in-depth explanations.

[ritvikmath](https://www.youtube.com/@ritvikmath)(2020) Multi-Armed Bandit: Data Science Concepts. Available at: <https://www.youtube.com/watch?v=e3L4VocZnnQ>

[Nikolay Manchev](https://domino.ai/blog/author/nikolay-manchev)(2023) Reinforcement learning: The K-armed bandit problem. Available at: <https://domino.ai/blog/k-armed-bandit-problem>

Monte-Carlo Tree Search Algorithm:

I have used a combination of websites and YouTube videos to understand the Monte-Carlo Tree Search algorithm, so I have a wide range of understanding on it, from simple and vague to more in-depth explanations. These resources cover topics such as: what the algorithm is, how it works, the different stages of the algorithms explained, advantages and disadvantages, examples with iterations to further assist understanding.

[Parag Radke](https://builtin.com/authors/parag-radke)(2023) Monte-Carlo Tree Search: A Guide. Available at: <https://builtin.com/machine-learning/monte-carlo-tree-search>

[AI and Games](https://www.youtube.com/@AIandGames)(2018) AI 101: Monte Carlo Tree Search. Available at:

<https://www.youtube.com/watch?v=lhFXKNyA0QA>

[Udacity](https://www.youtube.com/@Udacity)(2016) Monte Carlo Tree Search p1. Available at: <https://www.youtube.com/watch?v=onBYsen2_eA>

Connect4: Rules and Strategies:

I will be using these resources to help understanding of the game and its rules. This will aid in the report I have to do in term one for the rules of the game, explaining the states and allowable transitions.

Buffington (2016) How to Play Connect 4. Available at: <https://www.youtube.com/watch?v=utXzIFEVPjA>

[Keith Galli](https://www.youtube.com/@KeithGalli)(2018) The Best strategy to Win at Connect 4! (Odd Even Strategy). Available at:

<https://www.youtube.com/watch?v=YqqcNjQMX18&list=PLFCB5Dp81iNWTTfsXb_ZQwXtPYvaOIpbF>