Techniques of artificial intelligence

PROJ-H418

Project report : *Monte-Carlo* tree-search for Checkers

Sami Abdul Sater
Alexandre Flachs
Diego Rubas
Jeanne Szpirer

Academic year 2021-2022

Contents

1 Introduction : *Monte-Carlo* tree-search

Tree search is an intuitive way to solve a game with a limited number of possible moves. A *Monte-Carlo* tree-search (MCTS) is a tree-search algorithm that exploits **randomness** and **evaluation of simulated games** to decide the next move. The tree is built according to a policy that we hereby define.

Repeat n_{iter} times :

1. **Selection** of the **best** node according to policy
   - **Expansion** of nodes if needed
2. **Simulation** of the rest of the game, starting from the selected node. This simulation ends with a **reward** that takes into account if the game has been won or not.
3. This reward is **backpropagated** to the selected node.

Once all the simulations have been done, the tree is considered to be computed (though not necessarily fully expanded) : we then select the **best child**

1.1 Parameters

Are variable :

- The selection policy
- The best-child selection policy
- The number of iterations

1.2 Optimization and constraints

There are no particular mathematical constraints to ensure for this project. However, constraints are to be imposed to make it sure it runs in a **realistic time**, e.g. 15 seconds by move.

Under this time, the parameters of the search (n_{iter}, the policies, and more) must be tuned to **optimize the win rate**.

This report presents the implementation of a MCTS on top of a Checkers game. Explaining first the rules, very briefly, we then explain the implementation itself before presenting results of our AI agains a **deterministic** AI (minimax).

1.3 Our contribution

We took the implementation of a Checkers game with a minimax AI on top of it from an Open Source repository. Implementing MCTS required a huge refactor, at the game level and thus also at the minimax level. After implementing MCTS and refactoring, a benchmark was run for different parameters, which lead to an optimization of the win rate over the parameters of the search.

2 Rules of the Checkers game

2.1 Beginning of the game

2.2 Movements

2.3 Endgame

3 Implementation of MCTS to Checkers

4 Genetic Algorithm to tune parameters

5 Results