

Selected Optimization Methods

final project documentation

Damian Woźniak

Computer Science (sem. V, extramural studies)

1 Theoretical description.

The purpose of this project is to present an algorithm that can be used to find an approximation of an optimal move in a game. It is presented on a specific game, but the general idea applies to any turn based game with any number of players.

1.1 Building search tree.

In mathematics tree is a type of graph that has no cycles. We are going to use a tree to create a model of possible game states. Each node of the tree is going to represent game state. Root represents current situation, first level represents states after each possible move for the current player, second level represent states after moves of the second player. We can observe that number of nodes grows exponentially.

In a game with two players where each player gets to make only one move per turn nodes at odd levels represent game state after moves of one player and nodes at even levels represent game state after moves of the other player. Leaves represent game states where the game ended.

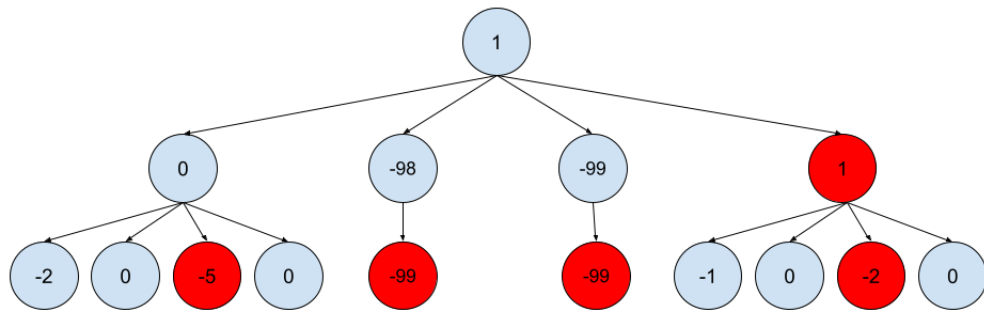
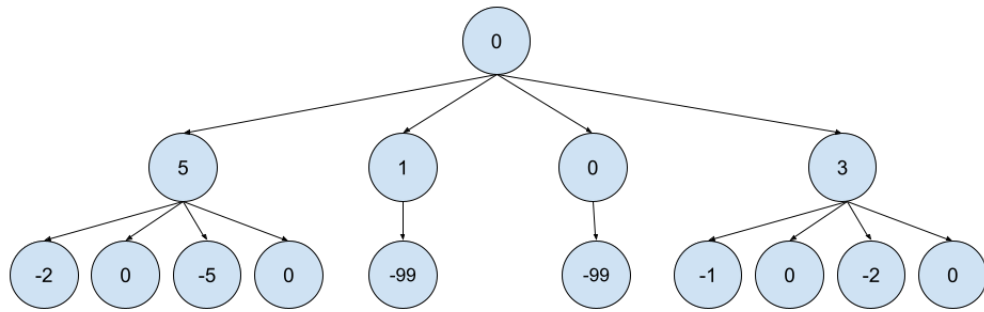
For many games creating full tree is impossible. It might contain infinite branches where game state changes back and forth, or it might be just simply too big to compute. As we noted before, number of nodes grows exponentially with each move.

Simplest heuristic is to cut the tree at a given level. However doing so often leaves us with a tree with branches that are all the same size, and gives us no basis to choose one move over another.

1.2 Searching

To search the tree we could use one of the recursion algorithms.

2 Exemplary calculations



3 Supporting computer program

<https://github.com/wutek/chess>
<https://wutek.github.io/chess/>