

Project Proposal

Backgammon Minimax Algorithm

Tejit Pabari (tvp2107), Archit Choudhury (ac4385)

The Game

Backgammon is a two-player game where each player has fifteen pieces that move between twenty-four triangles based on dice roll. The objective of the game is to be the first to move all fifteen checkers off the board.



Rules: See <https://en.wikipedia.org/wiki/Backgammon#Rules> for the rules

Backgammon involves a combination of strategy and luck (because of the dice). There are a couple of strategies that a player can follow to move their checker and anticipate possible counter-moves by the opponent. It also has some good opening moves (like chess) that have been analysed by players over the years.

Algorithm

The game is similar to a 2 player zero-sum game. The possible moves at each step can be represented on a tree, based on the dice role and the current state of the board. The tree can be searched using an expecti minimax algorithm to find the best possible next move. A couple of heuristics can also be applied to improve calculations and optimize the algorithm, such as alpha-beta pruning and forward pruning. All these could be parallized, which would improve the performance of the algorithm.

Since the search tree is big, we can't find an exact answer. Again, quite similar to chess. So we could experiment with different tree depths, and the time it takes to perform the operations on 1 core. Following this, we can gradually increase the core count, and see how that affects the time taken to perform the same operation. Overall, it would be quite a challenge to implement this using Haskell, but we do believe that it would be a fun project!