Explanation Assignment1

Find all legal moves of a player and sort the legal moves by their board scores:

According to the rules of move, simple moves, simple hops, swaps and hop-chains are found for each peg of a player. After getting all legal moves, a score should be evaluated for each of them. The moves can be sorted according to their scores, this can decrease the problem space according to [1]. Cutting off the moves with a negative score can make the alpha-beta pruning easier and shorten the searching time. If all these moves are scored negative, just return them to use, even though negative. If scores are the same, use the move that changes the furthest-away peg higher.

Select the optimal move among these legal moves using Best-Reply Search (BRS) method:

This method regards the two opponents as one: It means that only the opponent with the strongest move against the root player may move. An opponent is allowed to make a move even if it is not its turn. When it’ s player A’ s turn, set the next player as B. When it’ s not A’ s turn, select the better optimal move between player B and C. After the player execute an optimal move, his peg positions should be updated.

A MIN node contains all players, all moves for all opponents are searched, at the following MAX node, it is again the turn of player A. When searching the child nodes of a move by Player 2 (player C), a regular alpha-beta pruning occurs.

For each one of a player’ s moves, we get a tuple that contains a value. This value is compared with alpha and beta. If it’ s bigger than alpha and the depth is equal to the initial depth, then this move is an optimal move, afterwards, assign the alpha with the bigger one between alpha and the value. The search ends when the depth is smaller than 1, or A has won or lost. A timeout is set in order to change the depth of the BRS function. If it has taken too much time to run the BRS function under the depth of 3, it turns to a smaller depth of 2, then it turns to 1 if it still exceeds the timeout threshold. The Pseudo Code and explanation of the BRS are explained in [2].

Heuristics for evaluation of the game state:

The board is divided into six directions. Every space has a coordinate corresponding to the physical rows and columns, which is used to calculate the following score.

When moving all our own pegs, it is better to keep a higher proximity. To quantify this attribute, the number of steps between two pegs is calculated.

Move(distance)-score is the change of the board score after a move.

The number of a player’s pegs that have already been in his home is likely the most important factor of score.

Player-score is the weighted sum of the distance-score, number of pegs in home, the distance of the furthest peg and the proximity.

Board-score is the average difference between the first-player and the other players’ scores.

The heuristics are based on the work of [3].

Resources

[1] URL: https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning

[2] M. P. D. Schadd and M. H. M. Winands, "Best Reply Search for Multiplayer Games," in *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 3, no. 1, pp. 57-66, March 2011, doi: 10.1109/TCIAIG.2011.2107323.

[3] URL: https://webdocs.cs.ualberta.ca/~nathanst/papers/multiplayergamesthesis.pdf, page number 117