

This game is highly customizable, the user can change the hole(s) of the board, the number of stone(s), frame fate, and the frame rate for solving the solution.

The search was implemented using MAX-MIN with a toggle for alpha-beta pruning. The search used a priority queue to return the best heuristic out of the children nodes.

The state space consists for the following: the current player of the game, a 2d array of pits and player dimension, number of holes, and if the state is a terminal state. The pit in the state space contain the number of stone, the pits position on the board and the player the pit belong to. The state is stored in the Node space along with the heuristic value as well as the action and pit to get to it. The number of moves possible is twice the number of holes with stones. In order to get the final result a tree is formed recursively the total nodes in this tree can we used to represent the size of the state space which as an upper bound of: $(\text{NumberOfPits} * \text{NumberOfPlayers} * \text{NumberOfDirections})^{\text{Depth}}$. The reasoning behind the formula for the complexity in because: in a iteration of mini max we can be looking at any of the players, be choosing any direction to go in, choose any of the pits for the player and be moving the stones in any of the directions and since we do this for the depth of the search we raise the number by the depth.

As the number of stones and/or number of holes increased the search got slightly slower and the number of nodes expanded increases. However as the ply/depth increases the search got slower and a lot more nodes where expanded then there where when of stones and/or number of holes increased by the same amount. The nodes expanded/the time taken also decreased as the game started to progress; comparing the time for the first moves to the last couple of moves there was a difference by a factor of 10. With pruning enable the number nodes expanded where decreased quite a lot.

For my heuristics I choose:

1) To get the sum of all stones in the holes of ai player

This was the second worst heuristic. It tries to maximize the stones in the player's holes in the hopes that the opponent will get all his holes emptied and it will be able to take the maximized stones in its holes and put them all in the mancala hence getting the better score. But the chance of such an event happening are low and hence this heuristic does not work too well.

2) To get the sum of all stones in mancala of ai player

This for the second best heuristic. I think this is because it tries to maximize the player's mancala, however it fails against the best heuristic and it does not see that its maximizing move can give the opponents a small opportunity to make a better move.

3) To get the sum of all stones in the holes of ai player - the sum of all stones in the holes opponent

This was the worst heuristic. It tries to maximize the stones in the player's holes while minimizing stones in the opponent's holes in the hopes that the opponent will get all his holes emptied and it will be able to take the maximized stones in its holes and put them all in the mancala hence getting the better score. But the chance of such an event happening are low and hence this heuristic does not work too well.

4) To get the sum of all stones in mancala of ai player - the sum of all stones in mancala of opponent

This was the best heuristic. I think it was the best as it fist tries to maximize stones in the player's mancala keeping in mind to minimize stones in the opponent's mancala. This way the AI get the highest score.

Some statistics for Depth=4, Number of Hole(s)=6, Stone(s) Per Hole=4:

Using Heuristic #1 without Pruning Took 0.145 seconds Expanded 60732 nodes	Using Heuristic #2 without Pruning Took 0.084 seconds Expanded 60732 nodes	Using Heuristic #3 without Pruning Took 0.09 seconds Expanded 60732 nodes	Using Heuristic #4 without Pruning Took 0.125 seconds Expanded 60732 nodes
Using Heuristic#1 with Pruning Took 0.026 seconds Expanded 17728 nodes	Using Heuristic#2 with Pruning Took 0.024 seconds Expanded 14666 nodes	Using Heuristic#3 with Pruning Took 0.021 seconds Expanded 18716 nodes	Using Heuristic#4 with Pruning Took 0.02 seconds Expanded 15398 nodes

Winning statistics for Depth=4, Number of Hole(s)=6, Stone(s) Per Hole=4:

Player 0 → Player 1 ↓	Human	H #1	H #2	H #3	H #4	H #1 - Prune	H #2 - Prune	H #3 - Prune	H #4 - Prune
Human	N\A	↑	↑	↑	↑	↑	↑	↑	↑
H #1	←	N\A	Tie	←	↑	N\A	N\A	N\A	N\A
H #2	←	Tie	N\A	←	↑	N\A	N\A	N\A	N\A
H #3	←	↑	↑	N\A	↑	N\A	N\A	N\A	N\A
H #4	←	←	←	←	N\A	N\A	N\A	N\A	N\A
H #1 - Prune	←	N\A	N\A	N\A	N\A	N\A	Tie	←	↑
H #2 - Prune	←	N\A	N\A	N\A	N\A	Tie	N\A	←	↑
H #3 - Prune	←	N\A	N\A	N\A	N\A	↑	↑	N\A	↑
H #4 - Prune	←	N\A	N\A	N\A	N\A	←	←	←	N\A