Report

Heuristic function:

This was the evaluation(utility) function that was used to determine the value of a state in the game tree. This function took into account the number of red and black coins in the board, and decided the value of going to that state. So every red was -1( because Red is our min) and black was +1 ( because Black is our AI and the max), and every king coin was double its color’s value.

Increasing search Depth:

Initially the algorithm was designed to search only a depth of 3 in the game tree. The AI was able to predict upcoming threats to it. For example, if a Red coin was placed in a configuration that would remove a Black in the next move, the AI would recognize and find a way to take counter action. The other benefit of having such a low depth was that it determine an action to take in a short period of time.

Later, to better increase the evaluation, the tree was searched to a depth of 6. Like before, the AI was able t predict upcoming threats. An additional thing the AI did was that if there was no option for coin at a specific point in the game, it would ignore that coin and focus on a different coin that would reach the goal.

After this, I tried increasing the search depth to 15. This definitely improved the decisions the AI made for a certain move. It was hard to beat the AI, but it took a lot of time and space to compute, which lead to a slow game.

Drawbacks:

The biggest the most important drawback of this function was that when the game was nearing the end with very few opponent coins left. The AI performed safe approach and tried to stay around the same region instead of going closer to the opponent and attacking. The game was basically playing defense when it was about to win extending the game. The other drawback is that this method values a jump or a regular move as the same making it hard for the AI to goal for more Jump moves.

Heuristic Function(Version 2):

This function not only evaluates the value of going to state, but also tries to find states with more jumps. If there are more jumps possible for Black, then that state is picked over a regular state and if there are more jumps available for Red, then that state is least picked. This function ensures that the AI takes complete advantage of jumps, making it slightly efficient.

Drawback: This algorithm sometime picks jumping over being protective on a move, which might lose its coins.