

CS 2710

FOUNDATIONS OF ARTIFICIAL INTELLIGENCE

ANURADHA KULKARNI (ANK163)

AN AUTOMATIC PLAYER FOR THE GAME OF THE AMAZONS REPORT

AN AUTOMATIC PLAYER FOR THE GAME OF THE AMAZONS:

This deals with developing an automatic player for the Game of the Amazons. The Rule of Game of Amazons:

- Each opponent will have their set of Queens.
- Each queen can be moved horizontal, vertical and diagonal.
- First the player should move his/her queen and then from that moved location, the player should choose the location to shoot the arrow.
- The player can shoot the arrow either horizontal, vertical or diagonal.
- The player cannot shoot the opponent queens.
- The player cannot cross the queen or arrow. The square where the arrow land's is marked to show that it can no longer be used.
- The last player to be able to make a move wins.

Implemented the automatic player using minimax alpha beta algorithm.

IMPLEMENTATION:

As seen minimax algorithm incorporates generating the Successor states, check if the state reached terminal and Utility of the state.

1. Determining the Successor of the States:
Every possible location of queen has many possible next moves. These next moves are generated first by considering all the possible valid moves.
2. Check the terminal:
When both wscore and bscore are not present then it reaches the terminal state. Wscore and bscore represents the count of each territory.
3. Define the Utility of that State:
Returning the count present in each territory.

DESIGN ISSUES:

1. Branching Factor:

Since the number of possibilities to move to a different state is huge. It's around 1848 states in the initial step. Thus giving many possible successor states. In order to handle this first, I considered just selecting random 10 branches. But this also took lot of time. Then I considered only those next states that possibly comes closer to the opponent Queen and those states that reduce the possible one step transaction. This reduced the number of branches to half.

2. Managing the Time Limit Factor:

The time limit is set to only 15 sec. But my player was initially taking 43 sec only for the first step. Then I limited this by incorporating depth limit. I am using a depth check to limit the minimax to traverse to a certain depth. This will just traverse through 3 depth level to give the idea of the next few levels of depth. So that the action is given based on this. It was observed that the time factor was reduced to a considerable level. It run close to 15 sec for the first step.

3. Heuristics:

The Heuristic function is to help in providing utility. The utility function implemented is to the count the area presents in each territory. I thought of considering the distance between the queen of one player and the other player. But that didn't turn out to be that efficient, rather gave absurd results. Thus I felt it valid to use the count of the area as the utility function.

4. Performance of the Automated Player:

The automated player performed better. Initially since I didn't know the rules of games, me as a human played very bad and performed badly. After getting accustomed to the rules, even then my automated computer version performed better than the player. The reason being better prediction of the next state and better choice of next step that would lead to a win position.