CS 404 - HW 3

Görkem Yar 27970, Murat Demiraslan 27869

April 2024

1 Game Tree Search Problem Model of Hashi Game

Hashi game is a zero-sum game played with two players. One is human, and the other is an AI agent. As an extensive form game, in the Hashi game, players have multiple moves before the game ends. minimax method is used by the AI agent as the main strategy against the human opponent.

Input: A n x n matrix whose cells can be integers from 0 to 4 and characters ".". 0 represents the unlabeled cells which means empty islands that can be labeled by the players. Integers 1 to 4 represent the labeled island's value. "." represents the candidate cells for bridges

1.1 Players

Players = {Human, AI agent}

From the perspective of the AI, while the human opponent is MIN, the AI is MAX itself. Therefore, AI is trying to maximize its utility which is minimized by the human at each possible state.

1.2 States

States = {n x n grids consisting of labeled or unlabeled islands and bridges between islands} Furthermore, the states have the current scores of the players (ai.score and player.score) and a turn identification that shows whose turn it is. AI's turn is given as 0 and the player's turn is 1.

1.3 Initial State

The initial state (I) is the input state at the beginning of the game. It consists of a grid where a number or "." represents each cell. Additionally, the initial state consists of the current scores of the players as 0 and the game's turn. If AI starts the game the turn is initialized as 0 otherwise 1.

$I \in States$

1.4 Terminal State

If a state has no successor states then it is a terminal state (T). The successor states can be generated either by labeling an unlabeled island or creating a horizontal or vertical bridge between two islands. If none of these options are possible. Then the current state is terminal.

$T \subseteq States$

1.5 State Transition Function

Actions= {Place a horizontal bridge to connect two labeled islands, Place a vertical bridge to connect two labeled islands, label an unlabeled empty island with 3 or 4}

State Transition Function: From the current state, creating new states by one of the actions by following the rules. The first rule is that a new bridge cannot cross any other bridge or island. The second rule is that if a pair of islands have 2 bridges already, then a new bridge cannot be added between this pair. The third rule is that the total bridge number of an island cannot be greater than its value. By following these rules, for a new state, any of the actions is used.
& is the transition function symbol.

&: $s \in S \rightarrow D \subseteq S$; $\&(CurrentState) = \{NewStates\}$

Additionally, if the successor state is generated via creating a new bridge and the islands that are connected to the new bridge have completed their last bridge (# of bridges that are connected to the island = label of island), we need to update the player scores.

- if the player connects the bridge

  player.score += labels of the newly completed islands

  ai.score -= labels of the newly completed islands

- if the ai connects the bridge

  player.score -= labels of the newly completed islands

  ai.score += labels of the newly completed islands

1.6 Payoff Function

By the rule of the game, if the number of bridges connected to the island is equal to its label n, the player gets n points and the opponent loses n points.

$P : T \rightarrow R$ (how good is each terminal state for player 1 (AI Agent))

$P(T)$ = the total score of the agent at terminal node T

A payoff function for an AI agent is adequate for the solution of the model since the Hashi game is a zero-sum game, the human payoff is -P(T)

$P(T)$ is used for the mini-max method. At the turn of the human, the minimum of P(T)s of each terminal node is assigned to the predecessor node of these terminal nodes. In other words, from AI perspective, human tries to minimize the payoff of AI agent.

In the AI's turn, among the minimum payoffs assigned by the human, the maximum payoff is assigned to the AI agent's node. Similarly, AI calculates the Human's turn as the minimum payoff of the maximum scores that are assigned by the AI's next turn.

In our implementation, we used the minimax method with alpha-beta pruning to get rid of unnecessary calculations.