Game Trees, Minimax, Alpha-Beta Pruning

(Project2)

Nguyen Quoc Huy

23020082

CS 188

Method

Research Design

Objective: Understand the terminology of Game Trees, Minimax and Alpha-beta pruning.

Results

Fundamental Terminology

- Adversarial search problems, known as games.
- Type of games:
    + Deterministic or Stochastic (probabilistic).
    + Zero-sum.
    + Number of players.
    + Perfect information.
- Adversarial search returns a strategy or policy.
- Standard game consists of:
    + Initial state.
    + Players.
    + Actions.
    + Transition model.
    + Terminal test.
    + Terminal values.

Minimax

- Zero-sum-game algorithm.
- The opponent we face behaves optimally.
- Game tree:
    + State value: optimal score attainable by the agent that controls that state.

+ Terminal state: given end assumption.

+ Terminal utility: the value of a terminal state.

- We aim to maximize our score and minimize the margin of defeat:

  + $\forall$ agent-controlled states, $V(s) = \max V(s')$ ($s' \in$ successors(s)).

  + $\forall$ opponent-controlled states, $V(s) = \min V(s')$ ($s' \in$ successors(s)).

  + $\forall$ terminal states, V(s) = known.

Alpha-Beta Pruning

- An optimization for minimax.

- We can prune the search tree based on max's best option on the path to the root and min's best option on the path to the root.

Evaluation Function

- Functions take in a state and output an estimate of the true minimax value of that node.

- Depth-limited minimax: we give non-terminal nodes mock terminal utilities determined by a evaluation function.

- A common design for an evaluation function:

  $Eval(s) = w1\ f1(s) + w2\ f2(s) + ... + wn\ fn(s)$

  + Each feature fi is assigned a corresponding weight wi.

  + Features are some elements of a game state that we can extract and assign a numerical value.

Algorithm

I am currently revisiting my learning by implementing various search algorithms from previous lectures.