Evaluation Functions

* A heuristic evaluation function EVAL(s, p) returns an estimate of the expected utility of state s to player p

Evaluation Functions-Chess

A black and white chess board

Description automatically generated

For chess, typically linear weighted sum of features

Eval(s) = w\_1f\_1(s) + w\_2f\_2(s) + … + w\_nf\_n(s)

e.g., w\_1 = 9 with

f\_1(s) = (number of white queens) – (number of black queens), etc.

Monte Carlo Tree Search (MCTS)

* The basic MCTS Strategy dies not use a heuristic evaluation function
* The value of a state is estimated as the average utility over a number of simulations of complete games starting from the state.
* A simulation chooses moves first for one player, than for the other, repeating until a terminal position is reached.
* The get useful information from the simulation, we need playout policy that biases the moves towards the good ones
* Selection is starting at the root of the search tree and choosing a move guided by the selection policy
* Expansion is growing the search tree by a new child of the selected node
* Simulation is performing a playout from the newly generated child node, choosing move for both players according to playout policy
* Back-propagation is using the result of the simulation to update all the search tree nodes going up to the root

MCTS Algorithm

A white background with text

Description automatically generated

Stochastic Games in General

In nondeterministic games, chance introduced by dice, card-shuffling

Simplified example with coin-flipping:

A diagram of a tree

Description automatically generated

Algorithm for Stochastic Games

We can only calculate expected value of a position: the average over all possible outcomes of the chance nodes

Experctiminimax gives perfect play

Just like Minimax, expect we must also handle chance nodes:

A white background with black and green text

Description automatically generated

Summary

Games are fun to work on! (and dangerous)

They illustrate important points about AI

* Game theory and types of games (zero-sum, non-zero-sum)
* Optimal decisions in games using the Minimax algorithm
* Heuristic improvements with Alpha-Beta pruning
* Monte Carlo Tree Search for large and complex search spaces
* Stochastic games and decision-making under uncertainty