

Evaluation functions for game trees.

The minimax algorithm generates the whole game tree. For a game like tic-tac-toe, this might be feasible to compute. In general, this is not feasible.

For chess and go, don't evaluate the whole game tree, instead expand tree to a certain depth then use a heuristic evaluation to estimate the "backed up" value of the position.

Then can use minimax and alpha beta pruning on levels above this again.

How to choose evaluation functions.

Typically look at various features of board state, and compute values for each of these features. And take a weighted sum of the values.

Example: chess

One feature might be based on material value of pieces in question.

1 pt Each pawn  
3 pts Bishop  
3.5 pts Knight  
5 pts Rook  
7 pts Queen

These values become negative when talking about the value of pieces belonging to opponent.

Example: CS156 checkers

f1 – total number of possible hops on board.

f2 –  $\sum$  (x-coordinate [piece I] ) from I=0, to 8  
(player 1's 9 pieces)

f3 –  $\sum$  (x-coordinate [piece I] ) from I=0, to 8  
(player 2's 9 pieces)

$w1*f1 + w2*f2 + w3*f3$