

Monte Carlo Tree Search

It does thing based on exploration probability

- 1- Simulating random outcomes of choices.
- 2- Building a tree of possibilities.
- 3- Choosing the best move based on these simulations.

1- Tree Traversal

This is where we decide which node (move) to explore using a formula

$$UCB1(S_i) = \bar{V}_i + C \cdot \sqrt{\frac{\log N}{n_i}} \quad C=2$$

Upper
confidence
Bound 1

\bar{V}_i = Average value (win rate) of move i

N = Total Times parent node was visited (Total games we've played)

How good
the move
so far.

n_i = Times this move i was visited

\log = Natural log (helps balance

exploration vs exploitation

How many
times we
tried this
move

C = A constant that controls how much we explore new moves.

So, UCB1 helps pick a move that:

- 1 - Performed well before (high V_i)
- 2 - Hasn't been tried too often.

2- Node Expansion

If we find a move not explored before, add it to the tree.

3- Rollout (Random Simulation)

From the new node, we play a game randomly until the end.

∴ Just like flipping a coin and playing a pretend game to see if this move leads to win or loss.

4- Backpropagation

We go back up the tree and update the tree.