MONTE CARLO METHOD

... NOT THE ADMINISTRATIVE AREA OF THE PRINCIPALITY OF MONACO

Christian Treffs

MONTE CARLO METHODS OR EXPERIMENTS

> *... broad class of computational algorithms that rely on repeated random sampling to obtain numerical results.*

- Uses randomness to solve problems that might be deterministic in principle.
- Monte Carlo methods can be used to solve any problem having a probabilistic interpretation
- Method of **Reinforcement Learning**

MONTE CARLO TREE SEARCH (MCTS)

- Is a probabilistic search algorithm.
- Effective in open-ended environments with an enormous amount of possibilities
- MCTS applies Monte Carlo method to the game tree search
- Does not need to brute force its way out of each possibility
- Does not necessarily require an evaluation or good heuristic function

MCTS: BENEFITS

- Explores all potential options a that time
- Quickly identify the "best" one
- Keeps looking for other "good" options whilst validating how good the current best is
- Answer will typically improve the longer it has to "think" about it

MCTS: PHASES

1. Selection
2. Expansion
3. Simulation
4. Backpropagation

1. SELECTION

- Start with a root node and select a child node such that the node has maximum win rate/score.
- Make sure that each node is given a fair chance
- Keep selecting optimal child nodes until a leaf node of the tree is reached

SELECTION HEURISTIC: UPPER CONFIDENCE BOUND APPLIED TO TREES (UCT)

$$\frac{w_i}{n_i} + c\sqrt{\frac{\ln t}{n_i}}$$

- w_i = number of wins after the i-th move
- n_i = number of simulations after the i-th move
- c = exploration parameter (theoretically equal to $\sqrt{2}$)
- t = total number of simulations for the parent node

Formula ensures that no state will be a victim of starvation and it also plays promising branches more often than their counterparts

2. EXPANSION

- When it can no longer apply UCT to find the successor node, it expands the game tree by appending all possible states from the leaf node
- Move one step down to expose a new state in the tree, provided the state did not end the game either win or loss or exceeds the tree depth

3. SIMULATION

- Algorithm picks a child node arbitrarily, and it simulates a randomized game from selected node until it reaches the resulting state of the game.
- If nodes are picked randomly or semi-randomly during the play out, it is called light play out.
- You can also opt for heavy play out by writing quality heuristics or evaluation functions.
- Playout provides a score.

4. BACKPROPAGATION

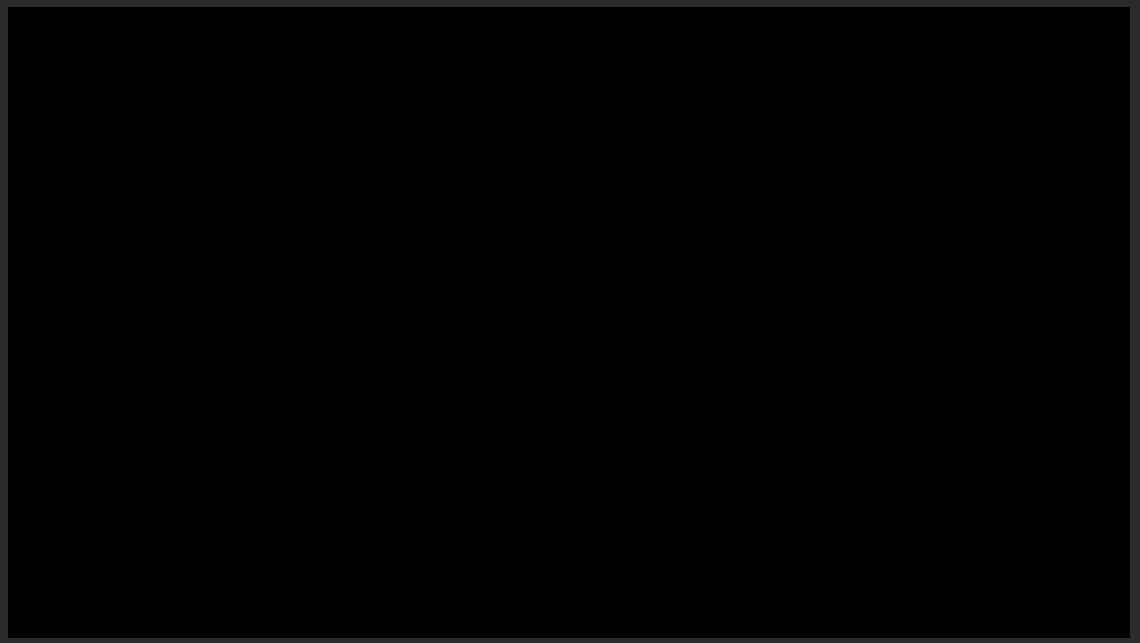
- Also known as an update phase
- Once the algorithm reaches the end of the game, it evaluates the state to figure out which player has won
- It traverses upwards to the root and increments visit score for all visited nodes
- It also updates win score for each node if the player for that position has won the playout.

MCTS keeps repeating these four phases until some fixed number of iterations or some fixed amount of time.

EXPLORATION VS. EXPLOITATION TRAIT-OFF

Keep searching for new strategies while exploring the best strategiers found thus far

VIDEO: AI OF TOTAL WAR: ROME II



https://youtu.be/1m9-7ZrpbBo?t=246

DEMO: TIC TAC TOE

SOURCES

- https://en.wikipedia.org/wiki/Monte_Carlo_method
- https://www.baeldung.com/java-monte-carlo-tree-sea
- https://github.com/eugenp/tutorials/blob/master/algo miscellaneous-1/
- https://www.analyticsvidhya.com/blog/2019/01/mont carlo-tree-search-introduction-algorithm-deepmind-alphago/
- https://jeffbradberry.com/posts/2015/09/intro-to-mo carlo-tree-search/
- https://github.com/brilee/python_uct