Monte Carlo Tree Search (MCTS) for HEX
Step-by-Step Visualization

Christophe Louargant

November 24, 2025

MCTS Algorithm Overview

The MCTS algorithm consists of four phases repeated for many iterations:

1. **Selection**: Traverse the tree from root to leaf using UCB1
2. **Expansion**: *the name is somehow confusing*
   - *Initial Phase: Add new child to root (tree grows wider)*
   - *Subsequent Phases: Reuse existing child (no growth)*
3. ***Simulation**: Run random playout from the new node*
4. ***Backpropagation**: Update statistics along the path*

UCB1 Formula

$$\text{UCB1}(i) = \frac{w_i}{n_i} + c\sqrt{\frac{\ln N}{n_i}}$$

Where:

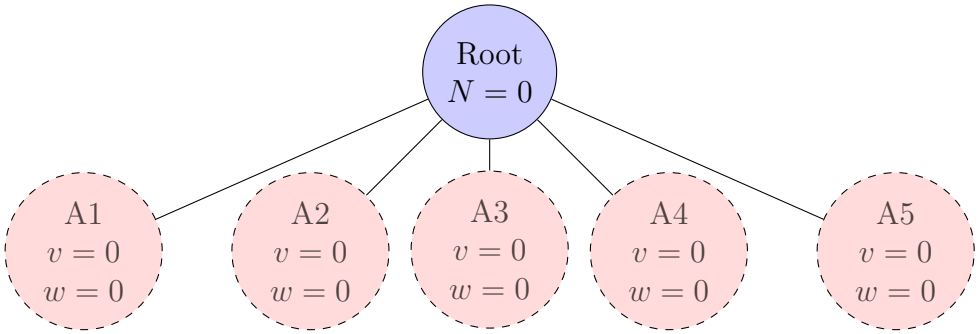
- w_i: Number of wins for node i
- n_i: Number of visits for node i
- N: Total number of visits to parent node
- c: Exploration constant (typically $\sqrt{2}$)

Step-by-Step Visualization

Suppose the board is in a state (represented by the root node of the tree) where there remains 5 positions available on the board for the current player (MCTS engine) to choose.

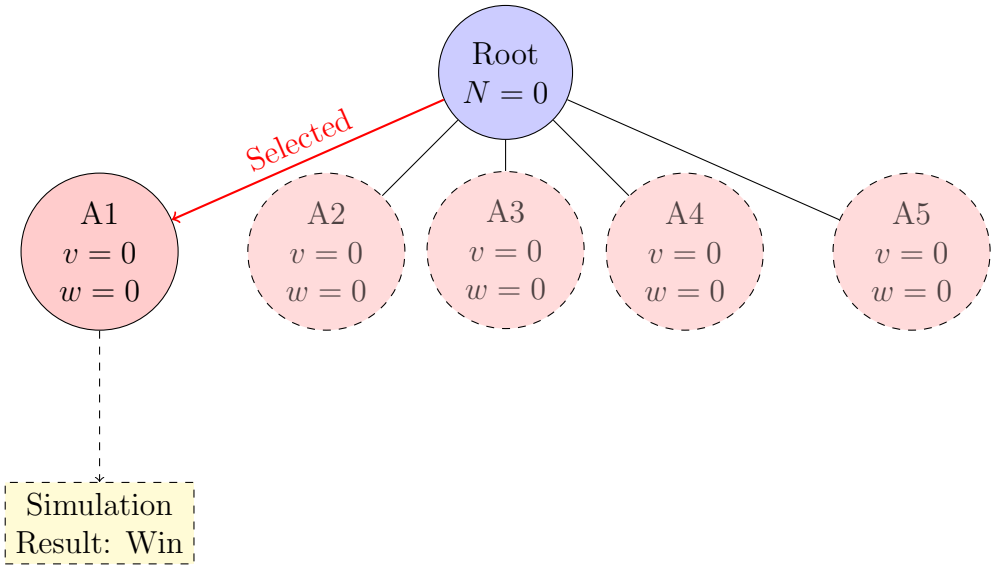
Initial State

Root node with 5 available moves: A_1, A_2, A_3, A_4, A_5. All unvisited.



Iteration 1: First Expansion

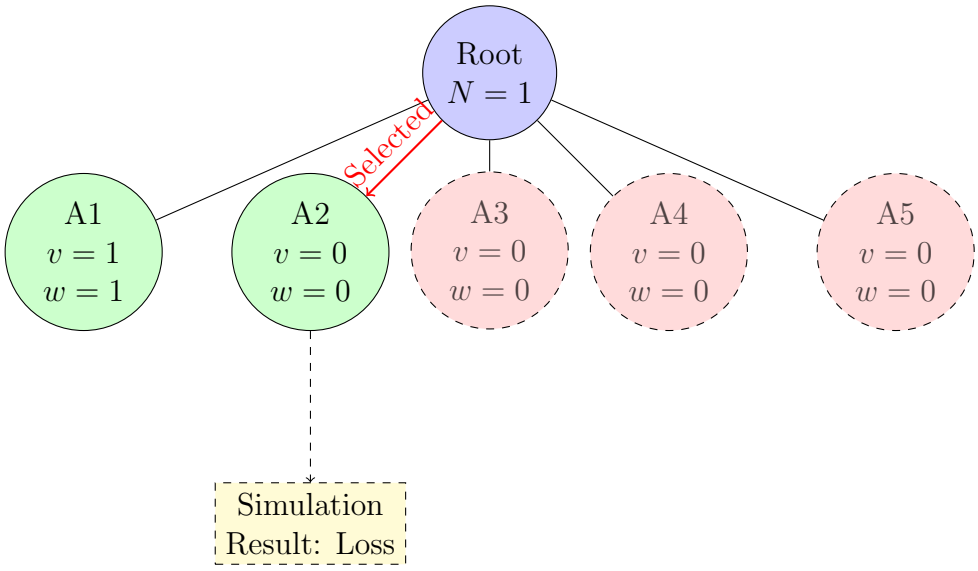
All nodes have UCB1 $= \infty$ (division by zero). Randomly select A1.



Backpropagation: A_1 wins $\Rightarrow v = 1$, $w = 1$, Root $N = 1$

Iteration 2: Second Expansion

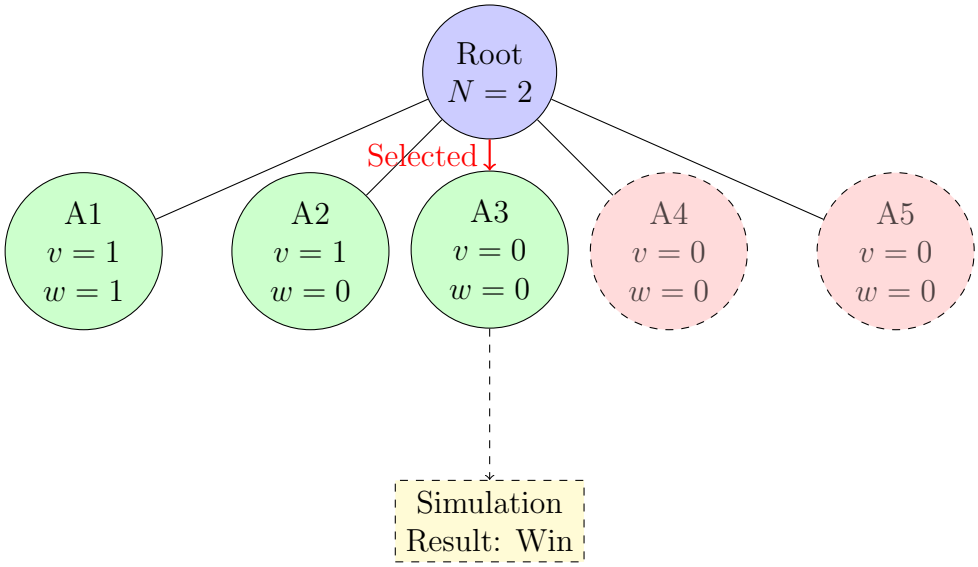
Unvisited nodes still have UCB1 $= \infty$. Randomly select A2.



Backpropagation: A2 loses $\Rightarrow v = 1$, $w = 0$, Root $N = 2$

Iteration 3: Third Expansion

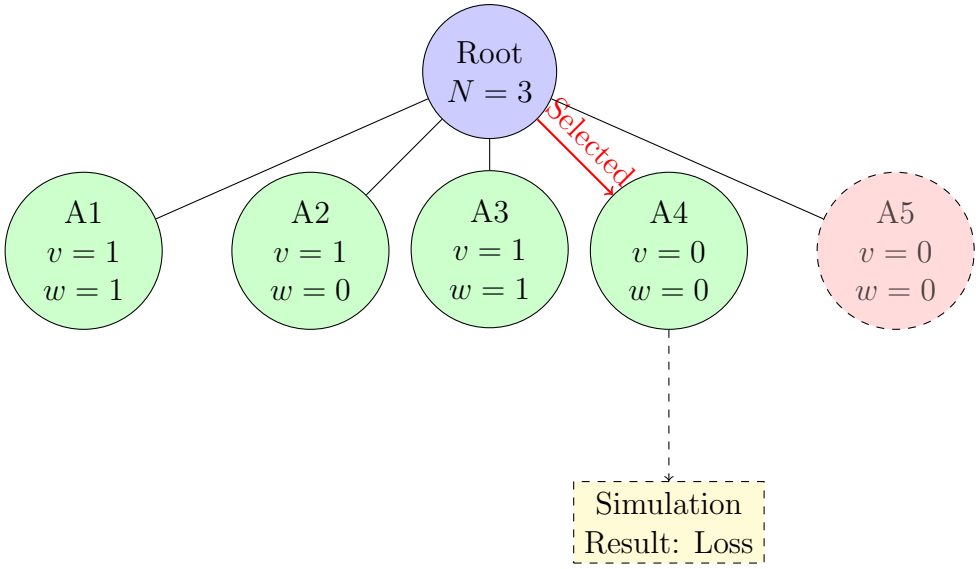
Select A3 (remaining unvisited node).



Backpropagation: A3 wins $\Rightarrow v = 1$, $w = 1$, Root $N = 3$

Iteration 4: Fourth Expansion

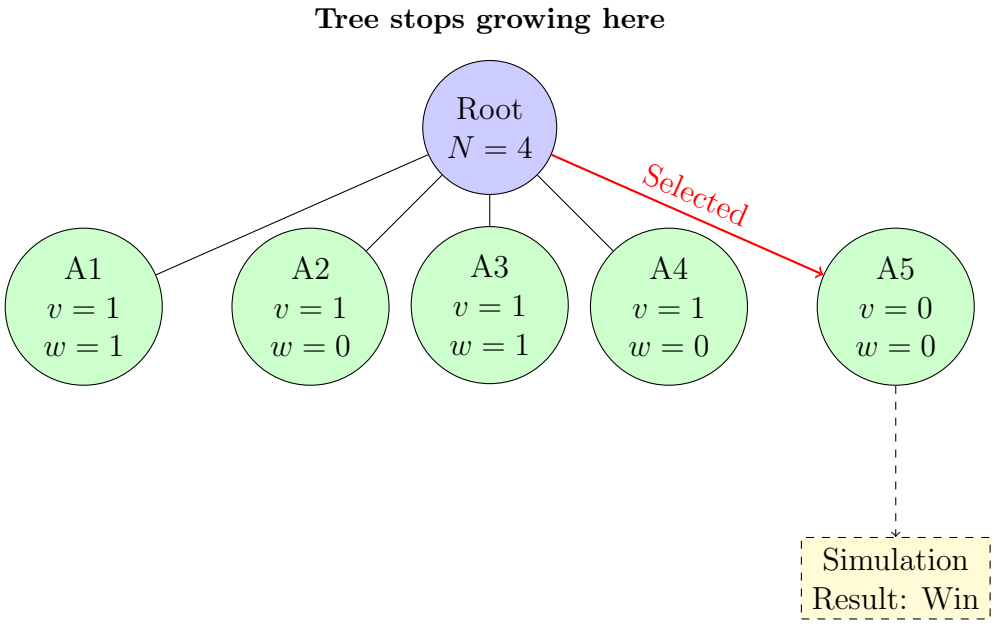
Select A4 (remaining unvisited node).



Backpropagation: A4 loses $\Rightarrow v = 1$, $w = 0$, Root $N = 4$

Iteration 5: Final Initial Expansion

Select A5 (last unvisited node).



Backpropagation: A5 wins $\Rightarrow$ $v = 1$, $w = 1$, Root $N = 5$

Iteration 6: First UCB1 Selection

Now all nodes have been visited. Calculate UCB1 ($c = \sqrt{2}$):

$$\text{UCB1}(A1) = \frac{1}{1} + \sqrt{2}\sqrt{\frac{\ln 5}{1}} = 1 + 1.87 = 2.87$$

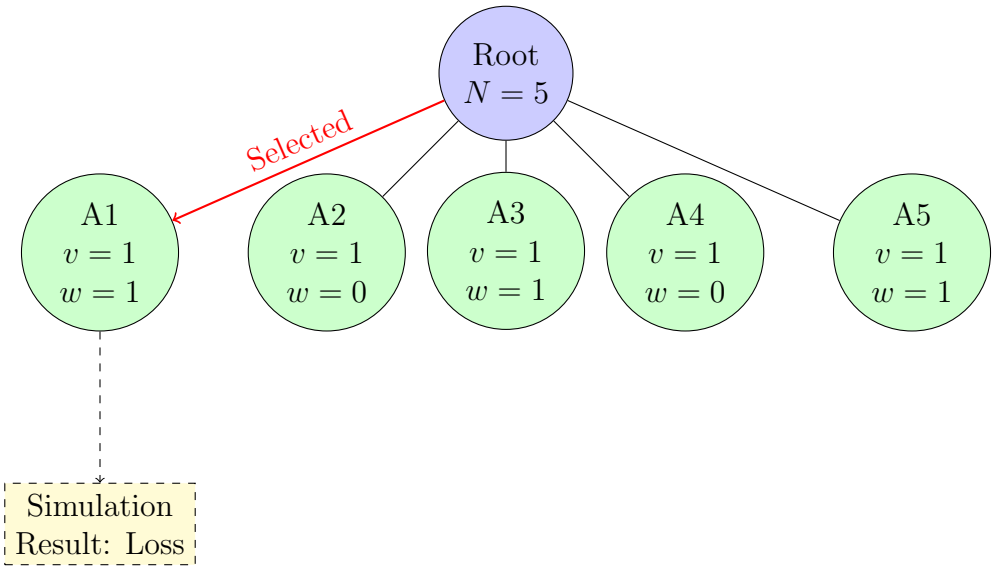
$$\text{UCB1}(A2) = \frac{0}{1} + \sqrt{2}\sqrt{\frac{\ln 5}{1}} = 0 + 1.87 = 1.87$$

$$\text{UCB1}(A3) = \frac{1}{1} + \sqrt{2}\sqrt{\frac{\ln 5}{1}} = 1 + 1.87 = 2.87$$

$$\text{UCB1}(A4) = \frac{0}{1} + \sqrt{2}\sqrt{\frac{\ln 5}{1}} = 0 + 1.87 = 1.87$$

$$\text{UCB1}(A5) = \frac{1}{1} + \sqrt{2}\sqrt{\frac{\ln 5}{1}} = 1 + 1.87 = 2.87$$

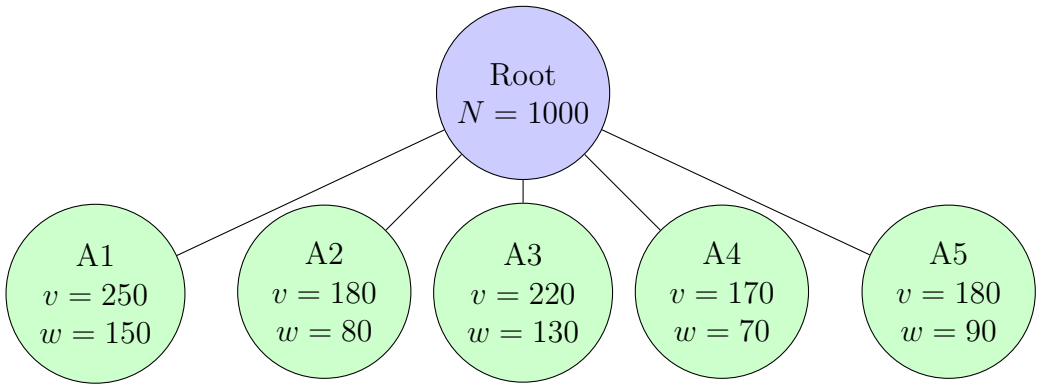
Tie between A1, A3, A5. Randomly select A1.



Backpropagation: Only updates selected node A_1: $v = 2$, $w = 1$, Root $N = 6$

Final Tree State

After many iterations, the tree might look like:



Final Decision: Choose move with highest win rate: A1 (150/250 = 60%)