0.1 Monte Carlo Tree Search (MCTS) Bot

To further improve decision quality beyond local heuristics, we implemented a bot based on Monte Carlo Tree Search (MCTS). The `MCTS_Bot` explores multiple future scenarios to select the best initial move.

For each valid pair of cells in the current grid, the bot performs a fixed number of complete simulations of the game. Each simulation starts by playing the selected pair and proceeds by alternating between players until no more valid moves remain. At each turn, moves are chosen using an ε-greedy strategy: with probability ε, the player selects a random move among the 5 lowest-cost options; otherwise, the best available move (i.e., the one with minimal cost) is selected.

We chose this ε-greedy strategy because it reflects a plausible human behavior when playing this game: players generally act greedily, but not perfectly. In approximately ε percent of cases, they may fail to find the optimal pair and instead choose a suboptimal, yet still reasonable, move.

Complexity Let n and m be the grid's dimensions. Assuming there are approximately nm valid initial pairs, and S simulations are performed per pair, each simulation consists of up to nm moves. Since each move requires selecting the best pair using a greedy or noisy greedy approach, which takes $\mathcal{O}(nm)$ time, the cost of a single simulation is $\mathcal{O}((nm)^2)$. Therefore, the total complexity per decision is:

$$\mathcal{O}(nm \cdot S \cdot (nm)^2) = \mathcal{O}(S \cdot n^3 m^3)$$