A Perfect Solver for Connect-4

Melody Na and Palash A.
ln244, pa334

Game Introduction

- Modelling the game Connect 4
- Rules on the next slide
- Game is played based on strategy
- Our AI agents utilized a minimax algorithm
- Differences in the AI agents:
    - Speed and efficiency to solve

Game Rules

- Game is played on a vertical board
- We chose to utilize the standard 6 row by 7 column board
  - This is variable in our game
- Two players alternate dropping their pieces into the columns
- To win, a player needs to achieve four in a row
  - Can be vertical, diagonal, or horizontal
- This game can end in a draw as well

Goal of the Project

- To design, implement, and test different AI agents for Connect 4
- Evaluate the accuracy of a minimax algorithm for Connect4
- Evaluate how factors such as efficiency would affect the game outcome
    - Efficiency was increased by introducing:
        - Alpha beta pruning

Approach

- Wanted to build an agent that would perfectly solve Connect 4
    - Versus relying on heuristic to score non-final position
- All agents were built off of minimax algorithm
    - Explores game tree until the end for a move that leads to the best score/win
    - Perfectly solves the problem
    - Agent acted as the maximizer, opponent was the minimizer
- Scoring Function: Based on how many more moves are required to be made in order for the player to win
    - This is ideal since the agent wants to win as soon as possible

Agents

- Simple Agent:
  - Utilizes pure minimax algorithm strategy
- Advanced Agent:
  - Utilizes minimax algorithm + alpha beta pruning
- We expect that the agents will differ based on time to solve, not accuracy since alpha beta pruning simply increases efficiency and speed

Testing Protocol

- Drafted tests on our own
- 10 categories of tests, varying from 30 - 40 moves made initially
  - Individual tests vary based on number of moves left to win, easy to medium to hard, whether the agent plays first or second
- Goal was to compare different agents through traits such as:
  - Accuracy
  - Number of explored positions
  - Execution time
- We will list our hypotheses for the testing results on the next page
  - Hypotheses are listed simply as best, intermediate, and worst to indicate relative strength

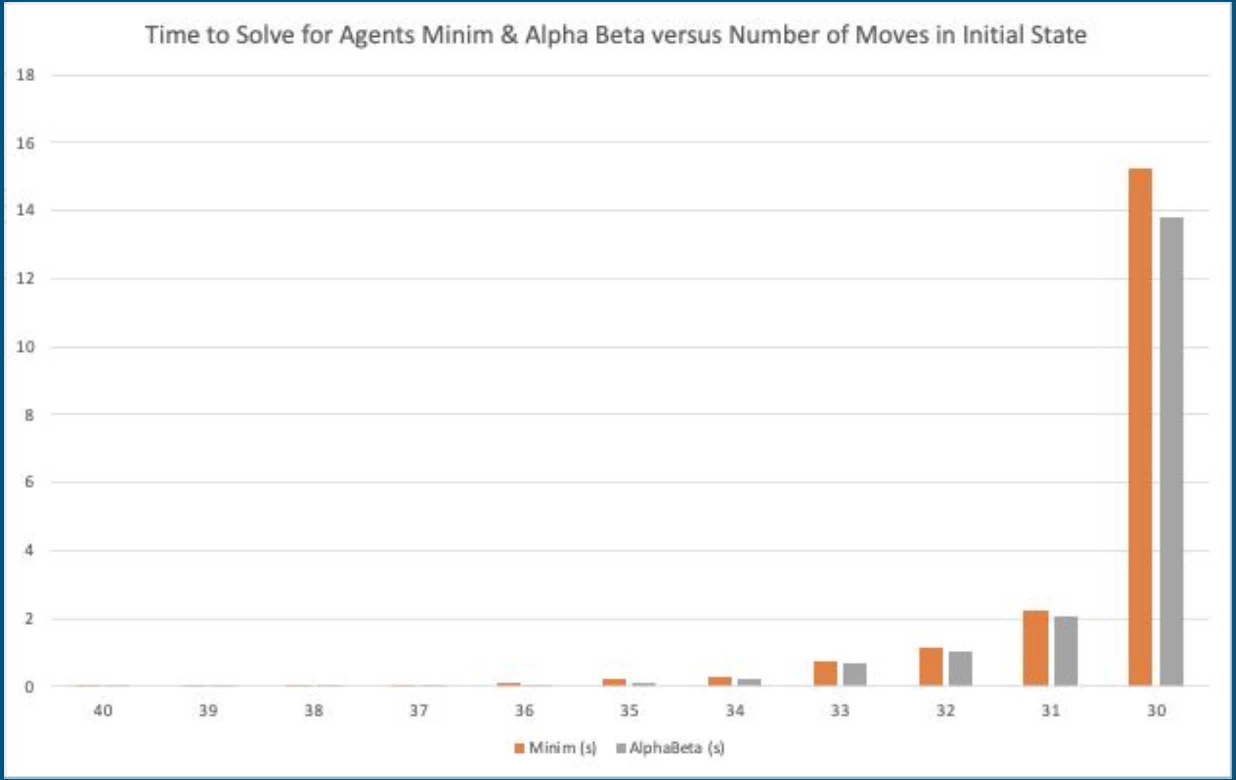
Testing Expectations

Robot Type	Accuracy	# of explored positions	Execution Time	# of explored positions/time
Simple	Best	Average	Worst	Worst
Advanced	Best	Average	Best	Best

Testing Results

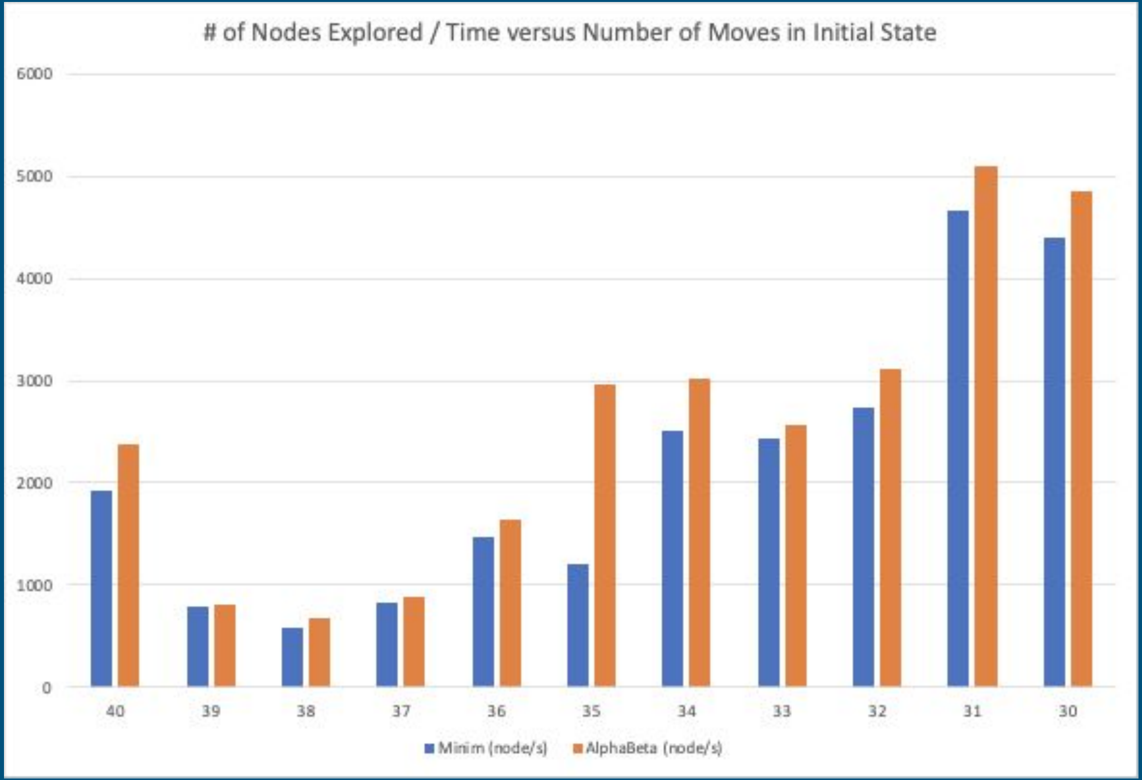
State	Minim (s)	AlphaBeta (s)	Minim (node/s)	AlphaBeta (node/s)
40	0.011950075	0.0096949	1924.674113	2372.381355
39	0.030551219	0.029426966	785.5660358	815.5784732
38	0.04438954	0.03850214	585.7235736	675.2871399
37	0.060735039	0.056429085	823.2480101	886.067885
36	0.0821908	0.07346526	1472.184235	1647.03698
35	0.205433775	0.0836232	1202.333939	2953.72576
34	0.258075356	0.214338467	2510.894535	3023.255742
33	0.75268706	0.71250588	2427.303586	2564.189365
32	1.17360224	1.02589552	2726.647829	3119.22602
31	2.266278025	2.0747436	4658.29871	5087.857603
30	15.2413429	13.8332996	4395.413215	4842.80699

Graphs



Time to Solve for Agents Minim & Alpha Beta versus Number of Moves in Initial State
18
16
14
12
10
8
6
4
2
0
40
39
38
37
36
35
34
33
32
31
30
Minim (s)
AlphaBeta (s)

Graphs



# of Nodes Explored / Time versus Number of Moves in Initial State
6000
5000
4000
3000
2000
1000
0
40
39
38
37
36
35
34
33
32
31
30
Minim (node/s)
AlphaBeta (node/s)

Testing Analysis

- Time to solve increases when number of moves in initial state decreases
- Time to solve is lower for AlphaBeta in almost all of the cases
- Rate of nodes explored is generally higher when number of moves in initial state decreases
- Rate of nodes explored is better for AlphaBeta in almost all of the cases
- Our hypothesis that AlphaBeta would perform better than Minim in terms of efficiency was correct

Notes for Future

- Add a randomness factor to the agents to see how that affects the overall accuracy of the solvers
- Try to utilize more techniques such as iterative deepening to improve the efficiency of the agent so it is playable