CSCI 3202 Course Project Report - Chenghao Xiong

Mancala AI: Minimax and Alpha-Beta Search with Utility Function Comparison

1. Algorithms Implemented

1.1 Random Player (Baseline)
The random player simply selects a non-empty pit from its side during its turn.

1.2 Minimax AI
- Explores all possible move sequences up to a predefined depth.
- Alternates between maximizing (AI) and minimizing (opponent) utility values.
- Uses recursive tree traversal and deepcopy() for state simulation.

1.3 Alpha-Beta AI
- Builds upon Minimax with alpha-beta pruning to eliminate irrelevant branches.
- Preserves optimality while significantly reducing the search space.
- Results in improved performance and shorter runtime, especially at higher depths.

2. Code Structure and Implementation Details

This project is implemented in Python 3, using a modular, object-oriented structure and common data science libraries.

Libraries Used:
- random: Generate legal random moves
- copy.deepcopy: Simulate board states in tree traversal
- multiprocessing: Run batch simulations in parallel
- datetime: Track runtime per game
- pandas: Store and export statistics
- matplotlib.pyplot: Plot win-rate comparisons
- tqdm: Display progress bars

Main Code Components:

1. Mancala Class
- Represents the board state and rules.
- Tracks player turns, captures, and winning conditions.

2. MinimaxPlayer Class
- Implements Minimax search to optimal depth.
- Tracks game tree nodes and transition states.

3. AlphaBetaPlayer Class
- Enhances Minimax with alpha-beta pruning.

4. Utility Functions
- utility: Default version uses Mancala score difference.
- utility_v2: Enhanced version with weighted pit stone control.

return 1.0 * (p2_mancala - p1_mancala) + 0.3 * (p2_pits - p1_pits)

5. Game Simulations
- do_one_minimax_game() / do_one_alphabeta_game(): simulate one full game.
- run_xxx_games_parallel(): performs bulk evaluation.

6. Main Runner Script
- Uses asyncio.run(main()) to execute experiments.
- Exports results to .csv and generates win-rate plots.

3. Experiment Design

Each AI played 100 games against a random player.
Tested depths: 1 to 5
Metrics collected:
- Win counts per player
- Draws
- Average number of turns per game
- Average runtime per game (in seconds)

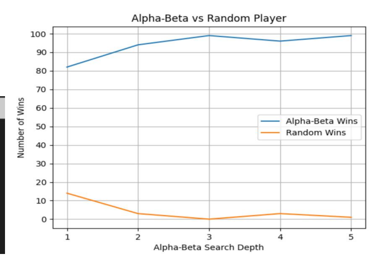
Both utility and utility_v2 were tested separately.

4. Results and Analysis

4.1 Alpha-Beta vs Random (utility_v2)

100%|

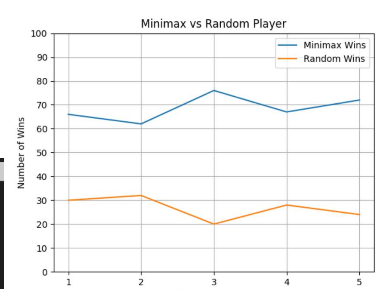
	depth	random_wins	alphabeta_wins	draws	avg_turns	avg_runtime
0	1	14	82	4	15.74	0.004597
1	2	3	94	3	18.31	0.014902
2	3	0	99	1	17.07	0.042342
3	4	3	96	1	15.87	0.101313
4	5	1	99	0	16.01	0.295559



4.2 Minimax vs Random (utility_v2)

100%|

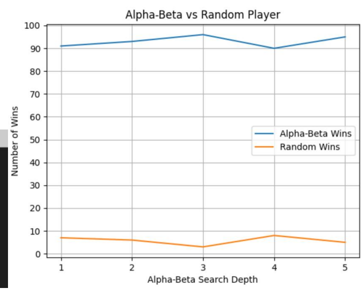
	minimax_depth	random_wins	minimax_wins	draws	avg_turns	avg_runtime
0	1	30	66	4	19.54	0.010014
1	2	32	62	6	20.40	0.032082
2	3	20	76	4	18.78	0.108151
3	4	28	67	5	18.66	0.447095
4	5	24	72	4	19.52	0.462324



4.3 Alpha-Beta vs Random (utility)

100%|

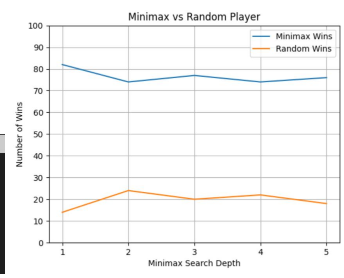
	depth	random_wins	alphabeta_wins	draws	avg_turns	avg_runtime
0	1	7	91	2	15.43	0.005318
1	2	6	93	1	18.43	0.016706
2	3	3	96	1	17.23	0.050176
3	4	8	90	2	17.19	0.139988
4	5	5	95	0	17.38	0.393070



4.4 Minimax vs Random (utility)

100%|

	minimax_depth	random_wins	minimax_wins	draws	avg_turns	avg_runtime
0	1	14	82	4	17.84	0.007974
1	2	24	74	2	22.04	0.036056
2	3	20	77	3	23.19	0.162472
3	4	22	74	4	20.23	0.521706
4	5	18	76	6	21.26	0.498063



5. Utility Function Comparison

Minimax: utility vs utility_v2

Depth	Wins (original)	Wins (v2)
1	82	66
3	77	76
5	76	72

Alpha-Beta: utility vs utility_v2

Depth	Wins (original)	Wins (v2)
1	91	82
3	96	99
5	95	99

6. Conclusion

This project demonstrates the successful implementation and evaluation of AI players for Mancala using Minimax and Alpha-Beta search. Alpha-Beta clearly offers improved performance over Minimax by reducing unnecessary evaluations, especially when paired with an enhanced utility function.

The extended utility function (utility_v2) enables more nuanced play, taking into account both immediate score and future potential. Its effectiveness becomes more visible at higher depths, where long-term planning is more relevant.