Playing Breakthrough

Ryan Zappone

I. Problem Formulation

A. State Representation

$$\mathbf{state} = (\mathbf{to\_move}, \mathbf{utility}, \mathbf{board}, \mathbf{moves})$$

A state is a snapshot of the game at any point of time. For the game we tested, Breakthrough, the things that change from action to action include which player's turn it is, the utility of the game, the board including piece positions as well as each player's amount of captures, and finally the possible moves each player can make.

B. Actions

From any given state, each player can move in one of three possible directions:

$$\mathbf{Forward}, \mathbf{Forward\ Left}, \mathbf{Forward\ Right}$$

These actions are only valid when:

- The piece's new position is within the borders of board.
- Piece moving forward move into a empty cell.
- Piece moving diagonally move into empty cell or one with the opposite piece

C. Transition Model

$$\mathbf{result}(\mathbf{state}, \mathbf{action}) \ \rightarrow \ \mathbf{new\_state}$$

The transition model is what describes what the state will look like after taking an action. For Breakthrough:

- If piece moves into an empty cell $\rightarrow$ update the piece's position
- If piece moves diagonally into a cell with an opposite piece $\rightarrow$ capture and update the piece's position

D. Goal Test

$$\mathbf{utility}(\mathbf{state},\ \mathbf{state.to\_move})\ != \ \mathbf{0}$$
$$\mathbf{white\_pieces} = \mathbf{0}\ \mathbf{or}\ \mathbf{black\_pieces}\ =\ \mathbf{0}$$

The goal for this game is reached once either any one player's piece reaches the opposite side of the board or when either player has no remaining pieces.

II. Heuristic Functions

A. Offensive Heuristic 2

The design of my offensive heuristic 2 revolved around being better than the original. To do this, in addition with the original heuristic's enemy counts, the heuristic implements two more rewards for the pieces. The first, rewards the pieces for advancing up the board where each additional row advancement is added to its total. For the second, the pieces are rewarded for capturing the enemies pieces which are another defining part of an offensive strategy. After testing multiple values of weights the final resulted in a 1:1:2 ratio of enemy's count, piece advancement, and captures, respectively.

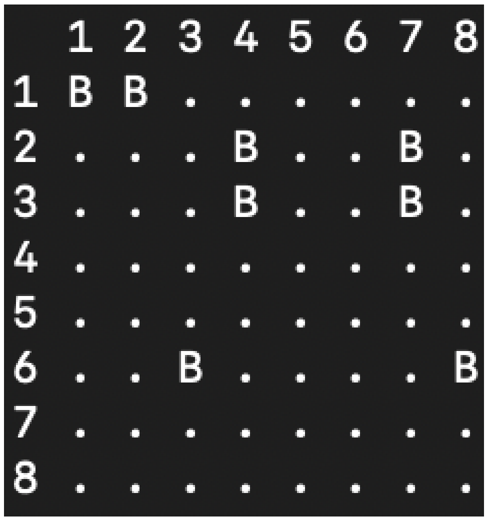
B. Defensive Heuristic 2

Similar to my offensive heuristic 2, this version 2 heuristic was designed to out perform the original defensive heuristic. Again in addition to the original reward for the amount of own pieces remaining this function rewarded and punished the pieces in 4 more different ways. After many, many adjustments to what is important, the list of rewards includes the following: A piece is rewarded if it has a protection piece, e.g if this piece were to get captured, another piece on our team would be there to then capture that piece. The final reward is whether the pieces are defending the back couple lines, trying to counter act enemy advances. In addition, the pieces are punished if the opposite pieces get too close to our opposing goal as well as if there are too many enemy threats surrounding a single piece. These weights were adjusted for several hours but the final layout of weights went with a ratio of 4:5:-7:-5:10 of pieces remaining, pieces protected, enemy's near goal, enemy's near pieces, and back line defense, respectively. For a final note, both of these heuristics also included a very small amount of noise just to keep the results slightly less deterministic.

III. Reports and Matchups

On the following pages, 6 unique matchups are executed and reported:

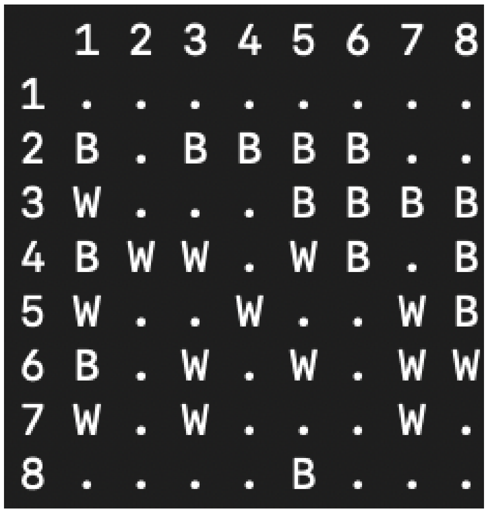
Matchup 1: Minimax Off1 vs AlphaBeta Off1 — Winner: Black



```
  1 2 3 4 5 6 7 8
1 B B . . . . . .
2 . . . B . . B .
3 . . . B . . B .
4 . . . . . . . .
5 . . . . . . . .
6 . . B . . . . B
7 . . . . . . . .
8 . . . . . . . .
```

Total moves: 82
White nodes: 8,929,245, Black nodes: 42,475,379
Nodes per move: White 217,786, Black 1,035,985
Time per move (s): White 0.575, Black 3.078
Captures: White 16, Black 8

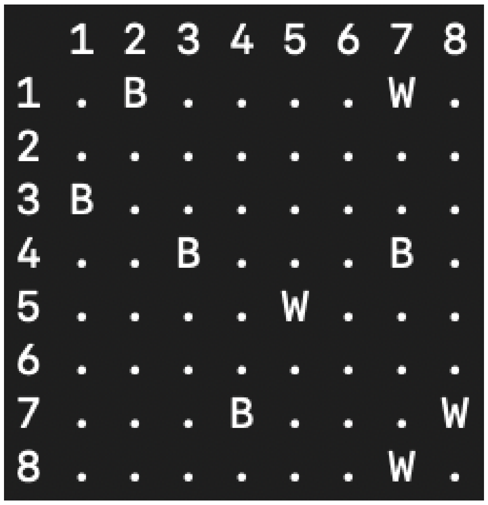
Matchup 2: AlphaBeta Off2 vs AlphaBeta Def1 — Winner: Black



```
  1 2 3 4 5 6 7 8
1 . . . . . . . .
2 B . B B B B . .
3 W . . . B B B B
4 B W W . W B . B
5 W . . W . . W B
6 B . W . W . W W
7 W . W . . . W .
8 . . . . B . . .
```

Total moves: 68
White nodes: 30,004,920, Black nodes: 18,021,428
Nodes per move: White 882,498, Black 530,042
Time per move (s): White 5.529, Black 2.427
Captures: White 2, Black 1

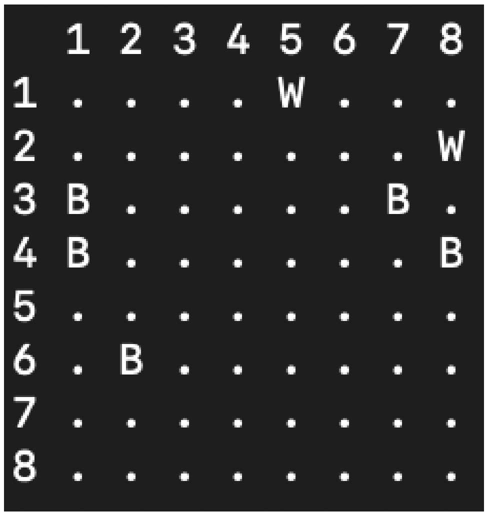
Matchup 3: AlphaBeta Def2 vs AlphaBeta Off1 — Winner: White



```
  1 2 3 4 5 6 7 8
1 . B . . . . W .
2 . . . . . . . .
3 B . . . . . . .
4 . . B . . . B .
5 . . . . W . . .
6 . . . . . . . .
7 . . . B . . . W
8 . . . . . . W .
```

Total moves: 105
White nodes: 30,656,587, Black nodes: 33,325,054
Nodes per move: White 578,426, Black 640,866
Time per move (s): White 4.852, Black 1.899
Captures: White 12, Black 11

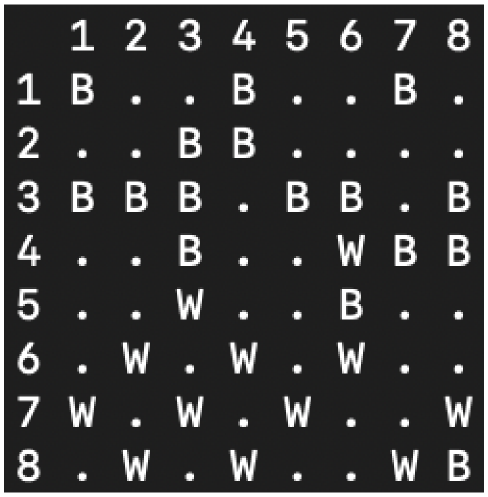
Matchup 4: AlphaBeta Off2 vs AlphaBeta Off1 — Winner: White



```
  1 2 3 4 5 6 7 8
1 . . . . W . . .
2 . . . . . . . W
3 B . . . . . B .
4 B . . . . . . B
5 . . . . . . . .
6 . B . . . . . .
7 . . . . . . . .
8 . . . . . . . .
```

Total moves: 95
White nodes: 23,280,048, Black nodes: 20,534,058
Nodes per move: White 485,001, Black 436,895
Time per move (s): White 2.750, Black 1.285
Captures: White 14, Black 11

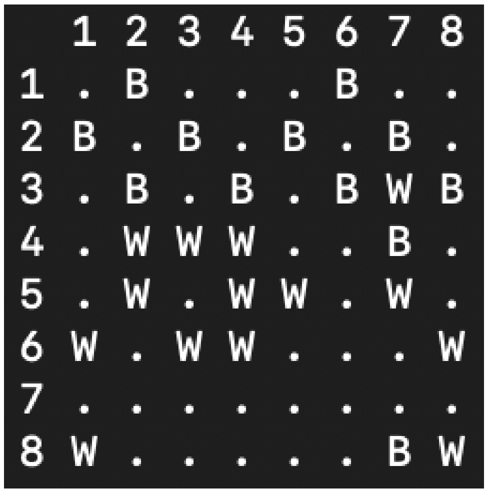
Matchup 5: AlphaBeta Def2 vs AlphaBeta Def1 — Winner: Black



```
  1 2 3 4 5 6 7 8
1 B . . B . . B .
2 . . B B . . . .
3 B B B . B B . B
4 . . B . . W B B
5 . . W . . B . .
6 . W . W . W . .
7 W . W . W . . W
8 . W . W . . W B
```

Total moves: 52
White nodes: 21,308,572, Black nodes: 28,695,363
Nodes per move: White 819,560, Black 1,103,668
Time per move (s): White 7.766, Black 4.253
Captures: White 4, Black 0

Matchup 6: AlphaBeta Off2 vs AlphaBeta Def2 — Winner: Black



```
  1 2 3 4 5 6 7 8
1 . B . . . B . .
2 B . B . B . B .
3 . B . B . B W B
4 . W W W . . B .
5 . W . W W . W .
6 W . W W . . . W
7 . . . . . . . .
8 W . . . . . B W
```

Total moves: 68
White nodes: 22,495,548, Black nodes: 30,931,322
Nodes per move: White 661,634, Black 909,745
Time per move (s): White 4.006, Black 8.469
Captures: White 2, Black 4

IV. Conclusion

Overall, the results had some degree of rationality. To start, all the alpha betas were at a depth of 5 and the mini max was at a depth of 4. The offensive heuristics generally performed better when they were paired against weaker defenses but it wasn't as consistent as expected. Especially defensive heuristic 2. It often won against offensive agents because it was in great positioning to exploit any offensive mistakes. It's notable that games with both defensive heuristics had very few captures but as seen in this report as well as other experiments, the defensive heuristic 2 does not in-fact consistently beat the original, despite its complexity. I mostly acquaint this to the randomness in both functions related to the fact mini max search, a known opposition to randomness, was used. Despite this, I was thrilled that the offensive heuristic 2 in-fact did consistently beat its original counter part. I was lucky enough to perform these tests on a pretty exceptional cpu so the times for depths of 5 weren't too extreme. Going off this, however, if deeper depths, possibly of 8 to 10 were utilized, I think the probability of my improved heuristics beating their counter parts would be greatly increased. For a future experiment I might want to further figure out optimizing my alpha prune function, possibly with transposition tables as well as efficient action sorting to test deeper experiments.