Relative Importance of Basic Strategy in Othello

Ben Shaman, Agastya Nashier, Josh Lee, Sumant Sharma
Department of Mathematics, Dartmouth College

Abstract

This paper attempts to elucidate the features, and advantages or disadvantages, of a few basic Othello principles. Therefore, we created player classes for an Othello simulation in Python to represent these principles in isolation. We ran simulations of every pairwise combination of these players against each other, playing both black and white, first (n = 10,000 through to the end of the game, and second (n = 320) with Egaroucid solver implemented starting at move 45. This data allowed us to reconstruct payoff matrices for the strategies played against each other and create dynamic visualizations. Further analysis provides explanations for certain elements of conventional Othello wisdom, including the topics of corners and parity. Our work demonstrates and analyzes the complex dynamics of a relatively simple game, from which new players can learn a lot.

Methods

Player Characteristics

To test various basic strategy principles, we created five different player classes, each intended to isolate an element of Othello's strategy.

1. Random Player
- Plays a random legal move on the board.
- Random player acts as our control class - a control to compare other strategies to

2. Greedy Player
- Plays the move that gains the most discs on every turn.
- This strategy is the intuitive choice for many beginning players; however, it only leads to short-term gains that actually weaken the player's position. The short-term gains of this strategy are easily lost later in the game.

3. Direction Minimizing Player
- Plays the move that flips in the fewest number of possible directions.
- A pitfall that some players fall into is making moves that are too “loud”. Moves that flip in many directions have far more hard-to-anticipate rippling consequences, so minimizing the number of directions can be beneficial in many situations, especially in the early game.

4. Defensive Player
- Plays the move that maximizes the difference between players' and opponents' number of legal moves.
- One of the most effective ways of controlling the outcome of Othello is to give your opponent a very limited number of legal possible moves. From there, you can force them to make moves that benefit your position.

5. Static Square Preference Player
- Plays based on a pre-set grid of preferences for certain squares. In our model, each square is assigned a value, and the move’s score is calculated by 2x the square being placed on plus the values of squares being flipped.
- Most squares have negative values to avoid the greedy player’s mistakes. The rationale is that some squares, such as corners, are extremely advantageous to possess, whereas others could benefit the opponent, so moves should be rewarded and penalized accordingly.

Simulation

To simulate these strategies and reconstruct payoff matrices, we:
- Created an Othello program in Python.
- Played every pairwise combination of the five players against each other through to the end of the game 10,000 times for each color.
- Since endgame strategy differs significantly from the rest of the game, we performed a second simulation using Egaroucid, an Othello solver, to play perfectly from move 45 to the end of the game. We were able to run this simulation 320 times for every combination.
- From these simulations we extracted and analyzed the win-loss percentage of every matchup as well as the margin of victory by percentage of discs held.

Simulation Results

Player/Player	Random	Greedy	Direction	Defensive	Preference
Random	0.5	0.3899	0.36415	0.2538	0.131675
Greedy	0.6101	0.5	0.555925	0.35415	0.164175
Direction	0.63585	0.444075	0.5	0.371575	0.153875
Defensive	0.7462	0.64585	0.628425	0.5	0.20805
Preference	0.868325	0.835825	0.846125	0.79195	0.5

Table 1: Payoff matrix for 5 strategies by win percentage, played through the end of the game.

Player/Player	Random	Greedy	Direction	Defensive	Preference
Random	0.5	0.70455	0.5666	0.17475	0.0729
Greedy	0.29545	0.5	0.3323	0.21785	0.02115
Direction	0.4334	0.6677	0.5	0.30485	0.0345
Defensive	0.82525	0.78215	0.69515	0.5	0.23275
Preference	0.9271	0.97885	0.9655	0.76725	0.5

Table 2: Payoff matrix for 5 strategies by win percentage, played through move 45, then Egaroucid for the endgame.

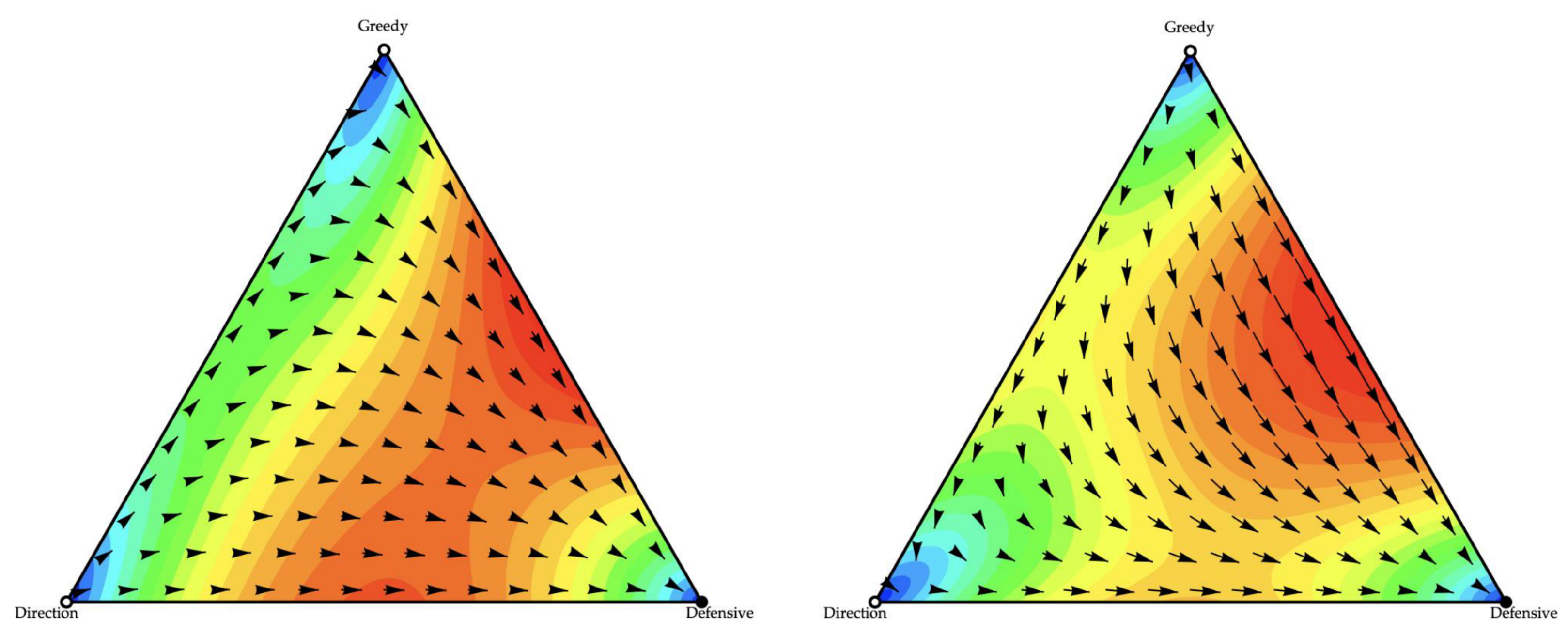

Figure 1: Dynamo 3S simulation without endgame solver (n=10000, left) and with endgame solver (n=320, right)

Results

Both with and without Egaroucid implemented, the static square preference player is dominant and represents an evolutionarily stable strategy. Without Egaroucid implemented, all four substantive strategies outperform random. However, with Egaroucid, the greedy and direction minimizing players both perform worse than random. The critical flaw of the greedy player is that it does exactly the opposite of the defensive player. By taking as many pieces as possible on every move, it decreases its own options, allowing the opponent to play more forcing moves. The random player is not able to take advantage of this when played through, but the endgame solver shows random's advantage. Direction minimizing, on the other hand, is a generally effective strategy, and is fairly similar to the defensive player in the early game. However, in the middle game, it develops a tendency to play diagonally-flipping moves on the perimeter of the position. This includes a lot of X-squares, which some other players, but particularly the endgame solver, are able to exploit. In a three-strategy comparison, removing random and static square preference, we see that the defensive player dominates. However, a notable change is that the direction minimizing player beats the greedy player when the perfect endgame solver is in place. Additionally, the direction minimizing player performs much than the greedy better against the defensive player when the endgame solver is in place.

Further Analysis

Looking beyond these player strategies, there is the game theoretical question of whether it is better to play black or white. Figure 2 shows an example of a typical early endgame position in Othello. A common pattern is the occurrence of sectioned off regions of open squares. This leads to a parity advantage for white, who goes second. In other words, if they play intelligently, they will play the last move in each region, gaining the maximum possible number of discs. When played through to the end, the greedy strategy, by nature of constantly trying to maximize piece count in the short term, gives away this parity advantage. That is why in Table 3a, greedy plays better against itself when playing black. With the endgame solver, all players have an advantage playing white.

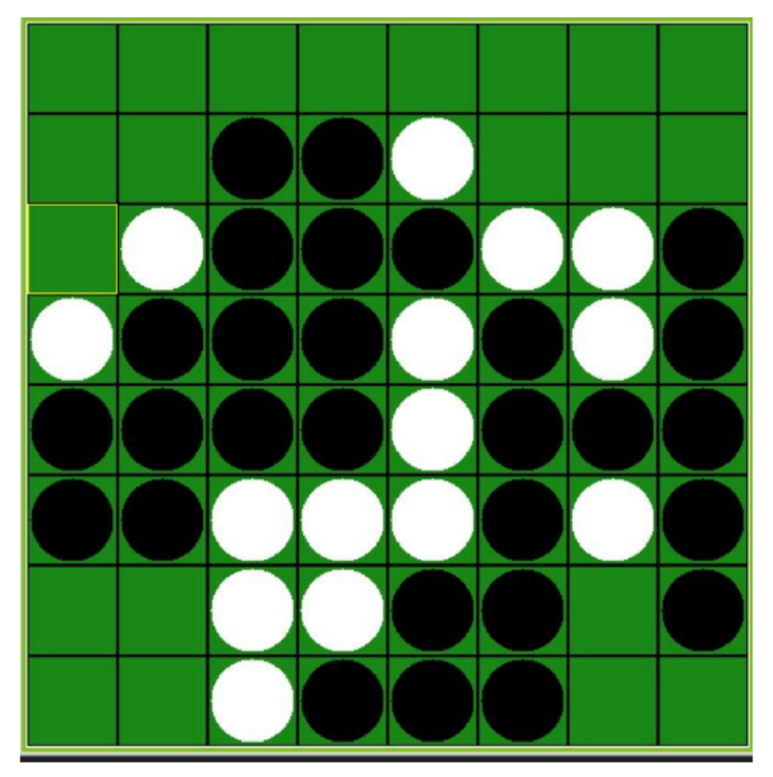

Figure 1: Hypothetical Othello position

	White advantage (%wins)	White advantage (%discs)
Random	0.0391	0.0251656
Greedy	-0.0502	-0.0400374
Direction	0.1246	0.0740062
Defensive	0.0811	0.1061438
Preference	0.0553	0.0501688

	White advantage (%wins)	White advantage (%discs)
Random	0.2194358	0.105957
Greedy	0.2664576	0.1179688
Direction	0.1159874	0.0589356
Defensive	0.1191222	0.1073242
Preference	0.1755486	0.08208

Table 3: (a) White players' advantage without endgame solver, (b) with endgame solver

Limitations and Future Research

One limitation we face is that our analysis only features n = 320 iterations for our endgame solver, so the results are relevant, but maybe not quite as robust as we want. Another is that we have not yet implemented any look-ahead for our players. The static square preference player dominates in this environment because its preloaded information contains far more game knowledge than any of the others could possibly access. It is preprogrammed to avoid some of the most egregious blunders, whereas the others are not. We hypothesize that implementing even basic 3-ply look-ahead could reduce this advantage, swinging in favor of the defensive player. Alternatively, we could add a level of "idiot-proofing" to the other strategies. Future research could include determining which strategies benefit most from look-ahead as well as combining elements of multiple strategies. We could also tweak the criteria by which players such as defensive and direction minimizing calculate their optimal moves.

Conclusion

The main implication of our findings is that this type of analysis can be very helpful for the understanding of beginning players. Despite Takizawa’s claim to have weakly solved Othello (2024), this analysis shows that there is still much we can learn and confirm from a more rudimentary analysis of the game's strategies.

References

[1] Egaroucid for Console (endgame solver) by Takuto Yamana (updated 2024)
[2] Takizawa, H. (2024). Othello Is Solved. https://arxiv.org/html/2310.19387v3

Thank you to Jared Heath ‘25 for his help implementing and debugging Egaroucid