Reversi Agent Report

Elya Renom
McGill ID: 261094604

James Kidd
McGill ID: 260276236

Executive Summary

The strongest algorithm we discovered to play Reversi is a combination of iterative deepening, minimax search, and alpha-beta pruning, all designed to work within a strict 2-second time constraint. This approach provides a balance of strategic depth, computational efficiency, and adaptability to different board sizes, making it effective in competitive play. Our goal was to create an agent capable of not only making optimal moves in immediate situations but also strategically planning ahead, while maintaining adaptability for dynamic board states.

The crux of the agent's strength is the iterative deepening framework. This method progressively increases the search depth, ensuring that even if time constraints limit the exploration of the entire search space, the agent will always return the best move found so far. This feature was critical in handling the strict 2-second per move constraint, allowing our agent to explore deeper layers of decision-making when the computational budget allowed. The minimax algorithm forms the decision-making backbone, simulating all possible moves for both the player and the opponent to find the optimal outcome under the assumption that the opponent plays perfectly. Alpha-beta pruning enhances this process by eliminating branches of the search tree that cannot influence the final decision, significantly improving efficiency.

Together, these techniques allowed our agent to calculate optimal strategies while adhering to computational constraints. A key contributor to the agent's success is the heuristic evaluation function, which assesses board states at leaf nodes of the search tree. The evaluation function prioritizes critical factors like coin parity (difference in the number of discs controlled by the player and opponent), mobility (difference in valid moves available to each player), and corner control (possession of stable and strategically advantageous corner positions). This heuristic balances short-term gains with long-term strategic objectives, ensuring the agent maintains dominance on the board while setting up favorable endgame conditions.

Detailed Explanation of Agent Design

Core Algorithms

The agent is built on three foundational techniques:

1. **Iterative Deepening Search (IDS):** This technique dynamically adjusts the search depth, progressively increasing it within a 2-second per-move time limit. IDS ensures that the agent always returns the best move from the deepest completed depth.

2. **Minimax Algorithm:** Minimax simulates all possible moves for both the agent and opponent, assuming perfect play. The agent seeks to maximize its score while minimizing the opponent's gains.

3. **Alpha-Beta Pruning:** This optimization reduces the number of evaluated states by eliminating branches that cannot affect the decision, enabling deeper searches within the time limit.

Iterative deepening search is the backbone of the agent's decision-making process. This technique progressively deepens the search tree, evaluating board states at increasing depths until the 2-second per-move time limit is reached. The primary advantage of this approach is that it ensures the agent always returns the best move discovered within the time constraint. For example, if the agent cannot fully explore the search tree due to the time limit, it will still provide a move based on the most recent completed depth. This method is particularly effective for time-critical environments like competitive Reversi matches. In our implementation, the agent starts with a depth of 1 and incrementally increases it. Each iteration evaluates possible moves using minimax search and alpha-beta pruning. If time expires before completing the current depth, the agent uses the best move identified in previous iterations.

The minimax algorithm drives the agent's strategic decision-making. Minimax simulates all possible moves for the player and the opponent, assuming both play optimally. The algorithm evaluates the outcomes of these moves using the heuristic function, seeking to maximize the agent's score while minimizing the opponent's potential gains. In Reversi, this process maps directly to exploring valid moves, flipping opponent discs, and recursively evaluating the resulting board states. Minimax ensures that the agent considers not only its immediate gains but also the opponent's potential responses, making it highly effective for planning several moves ahead. To improve efficiency, we integrate alpha-beta pruning into the minimax search. This optimization reduces the number of board states the agent evaluates by pruning branches of the search tree that cannot affect the final decision. If a move is worse than a previously evaluated move along the same branch, further exploration of that branch is unnecessary. We consulted class slides for the alpha beta pruning code. In our agent, alpha and beta values track the minimum and maximum scores that the agent and opponent can respectively achieve. As the agent explores potential moves, these values are updated, and branches are pruned when the current evaluation falls outside the alpha-beta range. This technique significantly reduces computational overhead, allowing the agent to explore deeper levels within the 2-second limit.

The heuristic evaluation function for our Reversi agent was designed to estimate the quality of a board state by considering key strategic factors: coin parity, mobility, and corner control. Each of these components plays a crucial role in determining the agent's ability to dominate different phases of the game. Coin parity measures the difference in the number of discs controlled by the agent compared to its opponent. This factor provides an immediate indication of positional advantage. However, early testing revealed that prioritizing coin parity too heavily could lead to suboptimal play in the mid and late game phases, as it focuses too much on short-term gains at the expense of long-term stability. Mobility, another critical factor, evaluates the difference in the number of valid moves available to the agent and its opponent. Greater mobility allows the agent to maintain strategic flexibility while limiting the opponent's options. This is particularly valuable in the midgame when positioning for future turns becomes crucial. Corner control, the third component, focuses on the ownership of stable discs in the four corners of the board. Corners are uniquely important because discs placed there cannot be flipped for the remainder of the game. This stability provides a long-term strategic advantage, particularly on larger boards where corner dominance has a ripple effect on controlling edge positions and surrounding areas. These components were combined into a weighted sum to produce the final heuristic value. The weights for each factor—2.0 for coin parity, 3.0 for mobility, and 5.0 for corner control—were determined through an extensive tuning process. Using a scipy-learn, we systematically explored different weight combinations through grid search and random sampling techniques. This process involved running numerous matches against various opponents and across different board sizes, from 6x6 to 12x12. Each combination was evaluated based on win rate, average discs captured, and mobility advantage, ensuring that the agent's performance was robust under diverse conditions. The results of the tuning process highlighted the importance of balancing short-term gains with long-term strategic priorities. While coin parity was effective for immediate positional gains, placing greater emphasis on mobility and corner control improved the agent's ability to adapt to dynamic board states and dominate the late game. This iterative refinement ensured that the agent's evaluation function consistently prioritized the most critical elements of strong Reversi play, resulting in an agent capable of achieving high performance against a wide range of opponents.

Our agent utilizes iterative deepening search to dynamically adjust its depth of exploration based on the strict two-second time limit per move. This approach ensures that the agent always returns the best move from the deepest completed level, regardless of when the time expires. On smaller boards, such as 6x6, the agent typically reaches a depth of 3 to 5, while on larger boards like 12x12, the increased branching factor limits the depth to 2 or 3. Depth variation occurs across branches due to alpha-beta pruning, which effectively eliminates irrelevant branches when a better move has already been found. This method ensures robustness, enabling the agent to adapt to varying game states and consistently make high-quality moves within the given constraints. The breadth of the search is determined by the number of valid moves available in each game state. On smaller boards during the midgame, the agent usually considers 5 to 10 moves, whereas on larger boards, the number can exceed 20 due to increased mobility and disc positioning options. The breadth is slightly higher for the maximizing player (the agent) because it seeks to maximize its opportunities, while the minimizing player (the opponent) may have fewer valid moves due to restricted options created by alpha-beta pruning. By evaluating moves in decreasing order of their heuristic scores—prioritizing corner captures or maximizing mobility—the agent enhances pruning efficiency, reducing unnecessary computations and improving overall performance. Board size has a significant impact on the agent's performance, particularly in terms of look-ahead depth and breadth. Larger boards increase the number of valid moves and the complexity of the branching factor, limiting the achievable depth within the fixed time limit. Conversely, smaller boards allow for deeper searches. To address these challenges, our agent incorporates board-size-specific adjustments to its heuristic evaluation function. On larger boards, corner control is weighted more heavily, as securing corners provides long-term stability and reduces the opponent's ability to dominate edge positions. This customization ensures that the agent efficiently balances computational demands with strategic decision-making across different board sizes. The agent's heuristic evaluation function combines coin parity, mobility, and corner control to estimate board desirability. Corner control is weighted most heavily due to its critical impact on securing stable positions that cannot be flipped. Mobility, which reflects the number of valid moves available, ensures the agent maintains flexibility and restricts the opponent's options. Coin parity measures the disc difference between the agent and its opponent, ensuring competitiveness in disc count. Alpha-beta pruning is employed to significantly reduce the computational load by eliminating branches that do not influence the final decision. Furthermore, move ordering prioritizes actions that maximize corner control and mobility, increasing the pruning efficiency and enabling deeper searches within the time constraints.

Initially, our approach to developing a competitive Reversi agent involved segmenting the game into three distinct phases—opening, midgame, and endgame—based on the percentage of empty spaces on the board. Each phase was to be governed by a tailored strategy that reflected the unique priorities and dynamics of that stage of the game. The opening phase prioritized mobility and strategic positioning, particularly emphasizing corners and avoiding risky moves near unstable edges. The midgame was designed to focus on balancing mobility and stability while limiting the opponent's options. Finally, the endgame sought to maximize disc captures and secure stable regions of the board, where moves are less likely to be flipped. We implemented these strategies by defining threshold percentages of empty cells to determine the transition between phases. For each phase, we designed evaluation heuristics and decision-making logic tailored to its objectives. During testing, this phase-based design demonstrated reasonable success in addressing the specific challenges of each part of the game. For example, it effectively avoided early pitfalls like corner-adjacent moves in the opening phase and navigated the more tactical decisions of the midgame. However, while this phase-based system provided a clear framework, it introduced significant complexity. Transitioning between phases required additional computational overhead, and in some cases, the boundaries between phases were too rigid. These transitions often overlooked opportunities or risks unique to specific game states, particularly when the board's configuration didn't neatly align with predefined thresholds. Furthermore, this approach lacked the flexibility to dynamically adapt across varying board sizes and opponent strategies, leading to inconsistent performance during testing. Through iterative design and testing, we realized that a unified strategy, driven by a single evaluation function and adaptable heuristics, could achieve better results without the added complexity. By incorporating heuristics that naturally accounted for phase-specific priorities—such as corner control, stability, and mobility—we captured the essence of the phase-based approach while maintaining simplicity and computational efficiency. For example, dynamic weighting of heuristics based on board size and game state allowed the agent to adapt organically, reducing the need for explicit phase transitions.

Evaluation

The heuristic function evaluates board states using a weighted sum of the following:

- **Coin Parity:** Difference in the number of discs controlled by the agent and the opponent.
- **Mobility:** Difference in the number of valid moves available to each player.
- **Corner Control:** Ownership of stable, non-flippable corner positions.

Weights were determined through systematic tuning using a grid search, ensuring optimal balance between short-term and long-term priorities.

To evaluate the quality of our agent's play, we conducted rigorous testing across multiple dimensions. First, we performed extensive head-to-head simulations against a variety of opponent agents that we constructed ourselves using different approaches (some with the three-phase approach, some with just a heuristic, some with monte carlo, some with minimax but different parameters), as well as the provided agents. These simulations were conducted on even boards of varying sizes, ranging from 6x6 to 12x12, to ensure the agent's adaptability to different game scenarios. We also engaged in iterative design, tweaking weights in the heuristic evaluation function and observing the impact on performance through controlled experiments.

Beyond automated testing, we played directly against the agent as humans to assess its strategic depth and adaptability in real-time scenarios. This hands-on approach gave us unique insights into the agent's strengths and weaknesses, revealing its ability to effectively capitalize on human errors while maintaining a consistent and strategic playstyle. Additionally, we consulted existing research on Reversi strategies and algorithms to refine our approach, incorporating insights from both academic sources and prior implementations of game-playing AI.

Based on its strategic design, the agent demonstrates strong quantitative performance. Against the Random Agent, we predict a win rate of approximately 98

Quantitative Analysis

Depth of Search

On smaller boards (6x6), the agent typically reaches depths of 3 to 5. On larger boards (12x12), the depth is limited to 2 to 3 due to the increased branching factor. Alpha-beta pruning ensures depth variation across branches.

Search Breadth

The agent evaluates 5-10 moves per state on smaller boards and over 20 on larger ones. Move ordering and pruning reduce unnecessary computations, prioritizing high-value actions such as corner captures.

Impact of Board Size

Larger boards increase computational complexity, requiring customized heuristics emphasizing corner control. Smaller boards allow deeper searches due to reduced branching factors.

Predicted Win Rates

- **Random Agent:** 98%
- **Average Human (e.g., "Dave"):** 80-90%
- **Classmates' Agents:** 70-80%

Advantages

The principal advantage of our agent is the integration of Iterative Deepening Search (IDS) with Minimax and Alpha-Beta Pruning. This hybrid approach allows for efficient exploration of the game tree within the strict computational constraints imposed by the single-threaded Mimi-Server paired with the tournament's two-second decision time limit. During midgame scenarios, when the branching factor is at its peak, this strategy allows the agent to focus computational resources on promising branches, effectively reducing the branching factor and maximizing search depth.

IDS ensures that the agent always has a feasible move prepared by incrementally deepening the search, even if the search does not reach a terminal state. This dynamic adjustment prevents suboptimal decisions, such as random moves, which would inevitably occur under a static Minimax approach constrained by Alpha-Beta pruning alone.

To further optimize the search, we incorporated a transposition table to store previously evaluated game states. This mechanism minimizes redundant computations by leveraging memoization principles. Caching is particularly effective in a game like Reversi, where many states recur due to overlapping gameplay scenarios. In the context of our Iterative Deepening Search (IDS) strategy, the transposition table is particularly valuable as it avoids recomputing state evaluations across successive iterations of increasing depth.

Disadvantages

Our agent's heuristic design is limited by the absence of advanced evaluation metrics, such as coin parity and stability heuristics. These metrics provide critical insights into midgame and endgame dynamics, potentially enhancing the agent's ability to make nuanced strategic decisions. [**SOURCE**].

The fixed size and hashing approach of the transposition table introduce significant inefficiencies. On smaller boards, the sparsity of the table results in underutilization. Conversely, on larger boards, hash collisions and limited capacity lead to frequent evictions of valuable state information, compromising the caching mechanism's effectiveness. These issues reduce the consistency of the agent's performance across varying board sizes, as the transposition table struggles to adapt to the dynamic demands of different game configurations.

Finally, the agent's performance is notably influenced by board dimensions. On larger boards, where complex evaluations and deeper searches are required, the agent performs well. However, on smaller boards with simpler states, the computational resources are not fully utilized, leading to diminished efficiency. This disparity underscores the need for a more adaptive approach to handling diverse board sizes effectively.

Future Improvements

1. **Dynamic Heuristic Weights:** Adapt weights based on game phase for better decision-making.
2. **Advanced Stability Metrics:** Incorporate piece stability to enhance heuristic evaluation.
3. **Dynamic Transposition Tables:** Preconfigure table sizes for different board dimensions to optimize memory usage and consistency.

Acknowledgements

This project was informed by class discussions, academic resources, and testing against various opponents. We consulted example code for alpha-beta pruning and utilized grid search for heuristic tuning.