COMP300024 Artificial Intelligence - Project Part B (Playing the Game) Report

Sammi Li 1271851 & Gonzalo Molina 1085253

Game-Playing Strategy / Algorithm

Our game-playing agent utilises a decision making approach based on the Alpha-Beta pruning variant of the mini-max algorithm. The objective of this agent is to selectively play the most optimal move on the board by exploring multiple game playing decisions and their effectiveness on the board. We chose to use the mini-max algorithm over other adversarial game-playing strategies as Inflexion is a deterministic and perfect-information game, which allows the mini-max algorithm to perform perfect play and thus guarantee optimal decisions, whilst Monte Carlo Tree Search is more appropriate for games that have uncertainty. While the standard implementation of the mini-max algorithm would have given similar results, we integrated alpha-beta pruning and set the depth to 3 to expedite the search process, aligning it with the time constraints imposed by the agent and specifications of Infexion.

The minimax algorithm makes use of an evaluation function which evaluates and assigns a value to the many moves that are up for consideration. The role of the minimax algorithm is to essentially maximise the current players utility and minimise that of the opponents. In other words, the algorithm tries to make the best move for the current player which would cause the most difficulty for the opponent.

Evaluation Function

The evaluation function plays a critical role in deciding which moves are the most advantageous for the Agent, and where most of the game strategy is held. The evaluation function essentially calculates a score that reflects the desirability of the current game state for the agent. This evaluation score is obtained by considering various in-game metrics and assigning weights to them, components that play a more critical role are assigned a higher weight versus components that provide little strategic advantages for the agent. The in-game metrics used to calculate the evaluation value were *cell power*, *cell dominance*, *mobility* and *vulnerability*.

- *Cell power* compares the amount of power the player and the opponent holds by counting the total power of their respective cells on the board. The difference between the player's power and the enemy's power is multiplied by the assigned power weight to obtain a power evaluation score.
- *Board dominance* measures the dominance of the player and the opponent on the board. It calculates the proportion of cells occupied by each player relative to the total number of cells (including empty cells). This dominance ratio is then multiplied by the assigned cell dominance weight to obtain a dominance evaluation score.
- *Mobility* takes into account the potential spread power from current players perspective to any reachable enemy cell. It calculates the amount of opponent cells which can be reached on the board proving the strategic advantage.
- Conversely, *Vulnerability* takes into account the spread of the opponent cells from the current player's perspective. It calculates the number of enemy cells that can potentially spread and affect the player's positions.

Metric Weightings

Metric	Weight	Importance	Motivation
Cell power	0.5	Medium	Accumulate as much power as possible
Cell dominance	0.5	Medium	Expanding to as many cells as possible
Mobility	1	High	Spread cells to near-by enemy cell to accumulate power and territory
Vulnerability	-1	High	Punish states which leave the player vulnerable to getting captured.

After numerous play throughs, observing the agents behaviour challenged by simple heuristic based opponents and fine tuning the each metric weight.

Initially a simple evaluation function was created that was only utilising the first two metrics of the game strategy, *cell power* and *cell dominance*. However it was found that the agent would prioritise cell dominance throughout the game and would ultimately lose the majority of the time. The evaluation state would encourage spawning cells all over the board in order to quickly accumulate territories, almost blindly. So when it was challenged by a simple heuristic based agent, most of the time the opponent would simply spread to these exposed areas on the board and gradually build up power and territory. And to combat the opponent expansion the player would simply just find other cells spots to spawn on to make up for it.

It was obvious that early expansion of territory was a much better strategy than simply accumulating cells by spawning on them. Not only that, the spreading of cells would need to be made in a manner that would not leave territory exposed to the opponent and easy to capture. In order to address this problem the addition of the *mobility* and *vulnerability* metric was created and incorporated to the evaluation function. These new metrics were assigned higher importance than the previous metrics and it was found that after numerous playthroughs the agent performed much better, often dominating the opponent early in the game and maintaining the advantage until all opponent cells were captured to the amount of turns was met.

A healthy balance was found which mostly prioritises early expansion on the board whilst still maintaining a dominant territorial foundation.

Ultimately, the evaluation function takes into account all of the above metrics and sums them up into a numeric value. This value represents the move's strategic value to the current game state, where the higher the value, the more advantageous to the current player and a more favourable position to make.

Effectiveness

We chose to judge effectiveness through the means of the end game result, appropriate decision making between SPAWN and SPREAD moves, and the total moves taken to end the game. These metrics were tested amongst the three agents we had developed: a greedy opponent, mini-max opponent with consideration of difference in powers, and our final version of the mini-max algorithm with evaluation of multiple in-game metrics.

Agents were tested against the greedy heuristic algorithm we developed in Part A, which gave insight into how they performed against a greedy opponent. Multiple versions of our evaluation function were compared against this greedy opponent. Initially, we found that the heuristic agent effectively overperformed against the mini-max algorithm with consideration of only the differences in power. This was because the mini-max agent would prefer SPAWN moves over SPREAD moves as SPAWN moves would result in an immediate positive effect in the differences in power. This was ineffective as the majority of the player's cells would sustain low powers, and thus rarely have the opportunity to capture opposing cells. Additionally, continuous spawning of cells would result in more opportunities for the opposing heuristic agent to accumulate power. Thus, the evaluation function was tinkered to prioritise potential captures and control of the board, arising our final agent.

Therefore, from the performance of the agents, we determined that the mini-max algorithm with alpha-beta pruning and consideration of multiple in-game metrics was the most time-efficient and effective agent. This is due to its ability to consistently win against the heuristic agent and other mini-max implementations as both RED and BLUE, to appropriately pick between SPAWN and SPREAD options, and being able to win without reaching the maximum amount of turns.

Creative or Technical Advancements

One significant idea we incorporated from our independent research into our strategy is the choice of the starting SPAWN moves to be in the centre of the board and adjacent to other player cells. This decision was made to limit the amount of isolated cells on the board. By grouping cells next to each other, the player is able to gain control of a certain part of the board. Thus, if a player cell is captured, there is a higher chance for the player to immediately recapture the enemy cell with a cell nearby. There will be an immediate reaction and punishment to capturing one of the player's cells.

The first two cells will spawn in the centre and then expand out to capture enemy cells. This choice was motivated by the desire to gain control of the centre of the board. By doing so, it allows the player to reach other parts of the board more easily. As the game progresses, spreading cells will be more important, thus the choice between spawning or spreading is under the control of the mini-max algorithm.

Report questions to answer:

- Describe your approach: How does your game-playing program select actions throughout the game?
    - Example questions: What search algorithm have you chosen, and why? Have you made any modifications to an existing algorithm? What are the features of your evaluation function, and what are their strategic motivations? If you have applied machine learning, how does this fit into your overall approach? What learning methodology have you followed, and why? (Note that it is not essential to use machine learning to design a strong player)

1. Performance evaluation: How effective is your game-playing program?
    a. Example questions: How have you judged your program's performance? Have you compared multiple programs based on different approaches, and, if so, how have you selected which is the most effective?

2. Other aspects: Are there any other important creative or technical aspects of your work?

    a. Examples: algorithmic optimisations, specialised data structures, any other significant efficiency optimisations, alternative or enhanced algorithms beyond those discussed in class, or any other significant ideas you have incorporated from your independent research.

3. Supporting work: Have you completed any other work to assist you in the process of developing your game-playing program?
    a. Examples: developing additional programs or tools to help you understand the game or your program's behaviour, or scripts or modifications to the provided driver program to help you more thoroughly compare different versions of your program or strategy

10–11 marks: Work that demonstrates a highly successful application of important techniques discussed in class for playing adversarial games, along with many significant theoretical, strategic, or algorithmic enhancements to those techniques, based on independent research into algorithmic game-playing or original strategic insights into the game, leading to excellent player agent performance.