Comp 424 - Final Project

Aidon Lebar - 260668812

April 8, 2018

1 Approach and Motivation

The agent mainly relies on a minimax search of all legal moves from the current board state for move selection. Using soft-fail alpha-beta pruning[1] to speed up the search, the agent will look to a depth of three to evaluate which move is the most advantageous. Due to the branching factor of the game and the overhead of cloning board states, the search could not look any deeper, as it took too long to evaluate such a large number of moves. To further ensure that the move would not exceed the time limit, at the start of each move the time would be logged and passed down recursively to the alpha-beta runing. If the move took longer than a presepecified amount of time, the search would break and return the best move it had seen so far. Through experimentation, 1995 milliseconds was determined to be an appropriate time limit, as it was usually enough time to complete the minimax search, while giving enough time for the final operations after the search and a reasonable buffer.

Upon reaching the maximum depth, each move is evaluated using a weighted linear function instead of being expanded. As a design choice, positive scores correspond to the move favouring the Swedes and negative score favouring the Muscovites. This allowed the same evaluation function to be used for both sides, and gave a clear deliniation between a good move and a bad one for the agent. This fuction is based on six heuritsics for how advantageous or disadvantageous a move seems given the current board

[1] A modified and unified version of the pseudocode found in Russell and Norvig, Artificial Intelligence: A Modern Approach (2nd ed.)

state. The simplest factor was whether or not the move ended the game. If it did, and the Muscovites won, the move would be weighed overpoweringly in the direction of the Moscivites, and correspondingly, would weigh it heavily in favour of the Swedes if it caused them to win. If a move clearly caused the agent to win, that move would be taken immediately, and if it clearly caused the agent to lose it would be discounted.

The other factors do not immediately discount or choose a move. They give smaller penalties and bonuses used for deciding whether or not a move is worth taking. The foundation of the evaluation function is a move value established through random simulation. Since it was too slow to run the simulations in game, two million random games were run in advance using a faster autoplay class that played two random players against each other. These games were then preprocessed, giving +1 to every move in a game where the Swedes won, -1 to every move where the Muscovites won and 0 to the moves in a drawn game. These totals were then divided by the number of games the move appeared in to get a standardized value for each move. Moves that appeared rarely (less than 25 times) were given the average value among all the moves, to avoid the misleading values caused by a small sample size. These values were the stored in a HashMap, which was serialized and can be read in from data during the first move of the game.

Several heuristics were then added to this base value to take the board cofiguaration into account when considering a move. Each of these factors was given a hand-selected value that was determined through expertimentation in games against random, greedy and fellow student players.

2 Theoretical Basis for Approach

3 Advantages and Disadvantages

4 Other Approaches Tried

5 Improvements