Analyzing Nine Men's Morris For a Optimal Strategy

Martin Boyd and Christopher Hirunthanakorn

August 18, 2012

Abstract

While analysis of Nine Men's Morris has been done, an optimal strategy for either player has yet to be presented. We utilize an adaptive learning program and AI in order to find out more about the optimal strategy. The tatics that we discuss have been verfied to ensure either player an advantage during gameplay. We hope to be able to develop the optimal strategy in the future.

1 Introduction

Over the years, computers have been utilized to find the solution for numerous games of varying complexity. Even Nine Mens Morris, the oldest game still played today, has been solved for. [1] However, that solution only focuses on the outcome of perfect play during the latter phases of the game. An optimal strategy for either player has yet to be found and is the main focus of our research. Due to factors such as complexity and the large number of game states, we decided to look at Five Mens Morris and later prove that our findings apply to Nine Mens Morris as well. In our search for the best strategy, we developed an adaptive computer player that would provide insight on the optimal moves for each situation after countless simulations as well as an artificial intelligence (AI) that could test the validity of our suggested tactics for the player.

In this paper, we describe the different strategies for the three phases of gameplay that will ensure the best outcome for either player. Details on our adaptive computer player, the data it generated from playing thousands of times, and how we analyzed it to develop the general strategy for each phase are presented as well. Additionally, we list all patterns and good moves that were verified from the results of our artificial intelligence that may not pertain to a specific strategy. While it may be impossible for a human to play the entire game perfectly, the strategies discussed here will hopefully improve their chances of winning.

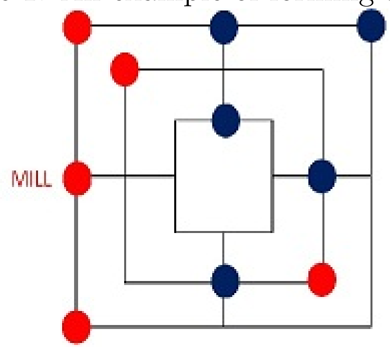
2 Game Description

Nine Mens Morris is a game that has many components. The game consists of a board with 24 playable spots and nine pieces for each player. The board is composed of three square rings with four lines connecting the midpoints of each respective side of the square and a spot is where two or more lines intersect. Gameplay can be broken down into three parts: the opening phase, midgame phase, and endgame phase. The opening phase consists of each player taking turns placing their respective pieces on the empty board. Whenever a player gets three pieces along the same line, they have formed a mill and may remove one of his or her opponents pieces. The mill only counts for the turn it was formed on. See figure 1. When removing a piece, the player must remove a piece that is not part of his or her opponents mills unless their opponent only has mills, in which case, the player may take any piece. Once a piece has been removed, it cannot come back into play. After all 18 pieces have been placed, the game moves to the midgame phase and players take turns moving their pieces to adjoining spots. Just like in the opening phase, mills can be formed during the midgame phase as well. Players keep moving pieces until one player is reduced to only three and that starts the endgame phase. The following rule is considered a variation that can be used for the endgame phase but we decided to exclude it due to the difficulty of its implementation. On their turn, the player with three pieces can move one of their pieces to any vacant spot on the board. The game ends when a player is reduced to less than three pieces or they are unable to make a legal move.

3 Our Approach

Our plan of action for tackling the task of figuring out the optimal strategy for a certain player had three main parts. First, we needed to familiarize ourselves with the game mechanics and observe any patterns that arose from multiple game plays. This was achieved with the development of a graphical user interface in conjunction with the basic game program. The next step was to create the adaptive learning

Figure 1: An example of forming a mill



player to speed up the process of finding the optimal moves a player should make in a given game state. And lastly, we designed an AI to test the general patterns that we discovered and prove that it gives an advantage for the player. With this three part plan, we got started on analyzing the game.

To help as a stepping stone in our research, we used Five Mens Morris as a simpler game to conduct experiments and hopefully gain insight into Nine Mens Morris. Both games follow the same rules. The only difference between the two games is that Five Mens Morris has 10 total pieces instead of 18, two rings instead of three, and there are 16 spots instead of 24.

4 General Strategy(GUI)

To start off, we created a graphical user interface (GUI) program to help visually analyze Nine Mens Morris and gain knowledge on what some of the strategy may be. It helped us identify actions that were beneficial to the player and form the notation that we would use for describing the positions on the board later on. From playing the game multiple times, we observed many tactics for the general strategy of both players. For example, in the opening phase, each player should attempt to capture spots that have the most connections and also capture spots on multiple rings. It prevents the opponent from being able to trap the players pieces when the midgame phase is reached and forcing a win. When a player is blocking his or her opponent from forming a mill, that player should make sure to do so in a way that does not trap that players pieces for the same reason. A great way to ensure a victory is to align two potential mills parallel to each other and have a piece swing back and forth to form a mill each turn.

After those preliminary findings, we wrote a program for Five Mens Morris with no graphics for research purposes. With this code, we obtained results from playing the game millions of times with each player playing a random strategy and observing the results to hopefully give us an approximation as to who may be the winner. Results from looping the opening phase a million times revealed that Player 1 ended with more pieces for 127,123 games. The same happened for Player 2 for 120,720 games and 752,157 games ended in a draw. Results from looping the full game a hundred thousand times revealed that Player 1 won 50,982 games and Player 2 won 49,018 games. Thus, we concluded from this experiment that both players have an even chance of winning with random strategy. With this data in mind, we moved onto developing and testing our methods for finding the optimal strategy.

5 Adaptive Program

The main component to aid in our research of Nine Mens Morris is an adaptive learning program written in python. Again, we wrote it for Five Mens Morris because its simplicity will help us obtain results quicker and gain insight into Nine Mens Morris. We made two adaptive programs: one for the opening phase and the other for the midgame and endgame phases. The adaptive programs work by playing the game many times and thus hopefully covering every game state and all possible moves that can be made at that game phase. When a certain player loses, the program goes to the last game state and eliminates the move that led to that loss. As the program continues, it essentially keeps cutting different branches in the game tree that are not beneficial to the players strategy and hopefully leaves the paths that lead to a certain player winning all the time.

To accomplish the tasks required of the adaptive program, we used a data structure that we refer to as matchboxes in our code. A matchbox consists of a game state a player may encounter and the possible moves that player can make at that game state. Each move is also given a value of one, which represents a bead. For a game state, the corresponding matchbox is looked up and a move is chosen at random from the list of possible moves that have a bead. We compiled a list of all the matchboxes and stored this data in a dictionary to make searching for game states quicker. To make the program work with the dictionary, we converted the game state notation into string form. The notation for this string board is an n for open spots, R for a spot occupied by player one, and a B for a spot occupied by player two. To punish a players losing move, the program searches through the dictionary for the matchbox with the losing game state and then takes the bead from the losing move. Taking away a bead means changing the value of that move to zero. Without the bead, the program does not repeat that same move again. If the matchbox for the certain game state the program is looking at does not have any beads for its moves, then the program calls that game a loss and punishes the previous move. Here is an example of a matchbox with some of its moves already punished. matchbox = [RRnnnBnnnnBnnnnn,0,0,1,1,1,0,1,1,1,1,0,1,1,1,1,1] The opening phase adaptive program uses one dictionary of matchboxes but the midgame and endgame phase version required two dictionaries due to the fact that the same game state can be reached by moving or removing different pieces and are therefore the results of different moves.

With our matchboxes set up, we began our simulations by running the program millions of times and keeping track of how many times a player won. After 10 million runs, the opening phase adaptive program gave us similar results as the basic Five Mens Morris program did. Player 1 had more pieces on the board in 72,544 games while it was the opposite in 44,095 games. The remaining 9,883,397 games ended in a tie in the number of pieces. From this data, we were able to suggest that player 1 has a 20

6 AI Program

The final piece in our analysis of Nine Mens Morris is the creation and use of an AI program written in python to confirm the validity of our general strategy. We applied it first to Five Mens Morris and plan on moving up to Nine Mens Morris after making improvements to the overall logic of the program. The basis of our AI comes from the game theory decision rule algorithm called Minimax, which works by determining a value for all possible game states using a set of conditions. Utilizing that information on the game states, the algorithm cycles through the list of game states and selects the one with the best value for the current board. Our AI applies the same methodology in order to determine what the next best move from the set of possible moves for its current state is. It will go through and evaluate the worth of each move in the set based on the resultant game board before selecting the most valuable one. The worth of a move is formulated by checking if certain conditions are met that would correlate to the actions a player would take if they were following the general strategy we mentioned earlier. The criteria includes placing or moving a piece next to open spots, setting up a potential mill, blocking an opponents mill, and making a mill with that move. They each have varying values assigned to them to ensure that certain conditions such as making a mill takes precedence over placing a piece with more connections. For example, setting up a potential mill and blocking an opponents mill both have the same value of 2 associated with them. The AI program currently only looks forward one move to choose the best move in all three phases but we plan to expand the search depth even further to look up to possibly 3 or 4 moves ahead.

Running simulations with the AI produced results that agree with the general strategy of taking spots with the most connections and blocking mills in such a way that you are not trapped that we proposed. After 10,000 runs with the AI playing against a computer player using a random strategy, roughly 7,000 games ended as a victory for the AI and 3,000 games are losses against the random computer player. Although this data does not prove that the AI has a winning strategy due to the fact that it is not a 100 percent win rate, this does support the validity of the advantage of our general strategy

7 Future Work

In our allotted time, we were able to formulate a general strategy for gameplay with information from the adaptive program and it was verified to give an advantage to the player as seen from the results of the AI. The next step in finding the optimal strategy would be to improve both the adaptive program and the AI to run faster and more efficiently. It would allow us to run longer and more in depth tests to obtain conclusive results on the best move for each player. Another possible route is to update our algorithms to analyze the full Nine Mens Morris game with the endgame rule of moving a piece to any vacant spot if said player has three pieces. If we are able to change the program to handle the higher level of complexity, we would not need to prove that our results are applicable to Nine Mens Morris as was the case with our Five Mens analysis. From there, the only thing remaining would be to examine the data for patterns and hopefully develop the optimal strategy.

References

[1] Ralph Gasser. Solving nine men's morris. *Games of No Chance: Combinatorial Games at MSRI*, 29:101–13, 1996.