2014 PRISM Summer Research Project

Computer Science and Engineering

"Mancala: Playing is Easy - Winning is Computational"

Collin Martin

Piérre Earls Miguel Saldivar

Helena Nunez Miguel Leiva

Daniel Ghossein Oswaldo Loya

Research Advisor: Dr. Kerstin Voigt

Special thanks to: Juan Nevares, Srinivas Bonthu

Getting to Know the Mancala Game

The initial stage of our research involved playing a traditional game of Mancala. There are numerous variants of Mancala, but the original game without modifications seemed best suited for our research goals. The variation of Mancala that we played involved a few rules that players must abide by. Players are also only allowed to move counter-clockwise, and they are only allowed to scoop from their side of the board. If a player scoops a pit and places their counters across the board with their final counter landing in their store, the player is granted an extra turn. The player can choose to accept or skip the extra turn. In order for a player to capture an opponent's counters, the player must have an empty pit corresponding to the opponent's pit. They then must scoop another pit, and have the final counter of that pit land in the empty pit aforementioned. The player is then allowed to claim an opponent's counters. While Mancala does not have many rules and seems relatively simple to play, it is a rather complex game. While players can make any move they wish, it is more appropriate to take a strategic approach by carefully planning and thinking about current and future moves. In addition, Mancala is also a two-player zero-sum game with perfect information.[7] We adjusted the counters and pits of the board in a later stage of our research in hopes of possibly recognizing certain strategies which we could apply to the original sized board. The reason for this is because the possible number of game states increases almost exponentially as the number of pits and counters are increased on a board. Simplifying the board provided us a more simplified look as to how Mancala operates. We also used computational strategies to aid us in our research, specifically implementing tools using the language Python to run scenarios.

A Beginning Advantage

As we continued to play Mancala, certain moves on the board stood out in terms of providing a player with an advantage. “After four days of processing, the computers had tested some 500 billion games and had shown that the CF opening was indeed a sure-win strategy for most responses by Yo.”[1] According to Dooley's research, the optimal starting move was made by choosing pit three – which would grant a free turn – followed by pit six. A visual representation of this starting move is displayed in *Figure 1* below.

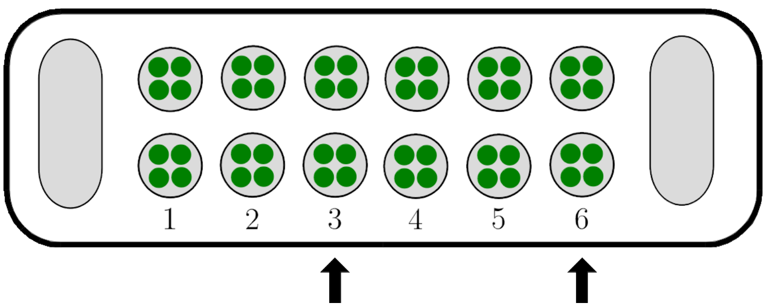

Figure 1.

Our own testing of Dooley's strategy tended to grant the first player an advantage throughout the game, often netting them a win.[1] Dooley's additional research indicated that there are sets of strategies that players can use. After initially learning and testing Mancala without accessing other research, we found that many of our team members had been applying these strategies all along. Reading Dooley's descriptions of said strategies helped clarify the differences and how we can apply them in each game.[1] The descriptions of these strategies also helped us identify them, which lead to the analysis of strategies on different sized boards.

The Tchoukaillon 'Strategy'

A strategy that was applicable to almost all variations of the boards we tested were 'Tchoukaillon Positions'. While this is technically a variation of Mancala, our team decided to also consider it as a strategy. Tchoukaillon positions occur when we are able to secure an extra turn multiple times. An example of a Tchoukaillon position would be two counters placed in pit five, and a single counter placed in pit six. In order to perform this strategy correctly, a player would need to move the single counter into the store, granting an extra turn. This extra turn allows the player to move the two counters in pit five, which would grant an additional turn to the player. After distributing the two counters from pit five, there is now a counter residing in pit six which the player can then move into the store as the final move. Tchoukaillon positions can provide a player with a significant advantage in terms of securing counters.[3] The sequence depicted in this section is represented in order by Figure 2.

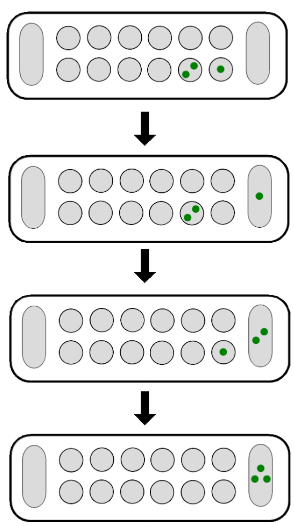

Figure 2.

A program created by Dr. Kerstin Voigt which will be discussed further in this paper has been known to recommend this specific strategy to a player quite often and would often net said player a win because this strategy tends to be able to secure a large number of counters.

Combinatorics Relating to the Mancala Game

The application of combinatorics to our research allowed us to approximate the amount of possible moves that could be made in a single game. We can determine the number of all theoretically possible boards, given that we we have N pits (N/2 on each side), and R counters (R/N in each pit at the starting state). The question of how many different board constellations can be produced from N pits and R counters is analogous to the questions of how many combinations exist for R selections from N items where repetitions are allowed and the order of the R selections does not matter. The combinatorics formula for this is

$$\frac{(N + R - 1)!}{R!(N-1)!}$$

We used a quickly written Python program in order to calculate the values of this formula for N and R varying from N = 4 pits (2 pits per side plus 2 stores) and a total number of R = 2 counters (1 counter per pits excluding stores) to N = 14 (12 pits per side plus 2 stores) and R = 72 counters.

#cnts per pit	#pits per side (= (N-2)/2)					
(= R/(N-2))	1	2	3	4	5	6
1	10	35	84	165	286	455
2	126	1287	6188	20349	53130	118755
3	1716	50388	480700	2629575	10295472	32224114
4	24310	2042975	38567100	3.5E+08	2.05E+09	9E+09
5	352716	84672315	3.16E+09	4.76E+10	4.18E+11	2.56E+12

What we found was that the number outputted by the combinatorial function that we created was not completely accurate, but it gave us a representation of how the possibilities increased as the counters and pits of a board were altered. We say it is not completely accurate because there are certain states of the game that are impossible to obtain. For example, playing Mancala will never result in a board where all counters are in a single pit. Due to the number of possibilities that are created from the normal board of four counters and six pits per side, we decided to look into ways we could simplify the game while still creating an accurate representation of the normal board. In order to recreate an accurate representation of the board while simplifying it, we looked into ratios. We decided to test a board that would translate to exactly half of a normal board, meaning we used two counters per pit and three pits on each side. This would reduce the number of possible game states that occur, so if we found that a 1:2 ratio board yielded results pertaining to winning strategies, then we would be able to find it much more quickly. By first applying combinatorics to a game of Mancala and seeing how we can reduce the number of possibilities, the likelihood of finding a winning strategy became much more possibly. Unfortunately, this was not the case and certain strategies became more or less relevant as the number of counters and pits were adjusted.

How Ratios Affect Strategies in Mancala

While analyzing the boards consisting of a variety of different ratios of counters and pits we also recorded data based off of the wins and losses for each player, number of captures, and number of moves. This gave us an insight as to how the game behaved when we altered the original size of the board. There were some noticeable differences depending on the size of the board. For smaller boards with a larger number of counters, the strategy known as ‘looping’[2] appeared quite often. Smaller boards also tended to have games which did not last long. Depending on the number of counters in play, different strategies became more apparent. For example, 'looping'[2] was quite common if there was a high amount of counters in each pit. In smaller sized boards it is quite difficult to employ strategies within the game. Every strategy that could be used on a smaller board seemed as if it were a weakened version of said strategy, or it was just simply not applicable. For boards with a larger number of pits, we tended to see much more use of the strategy ‘capturing’[2]. The beginning states of the game in the larger board tends to be mostly captures. As the game progresses, we begin to see more of the two strategies known as ‘starving’[2] and ‘stalling’[2]. These strategies proved effective for a larger board because there were more pits that each player could use to effectively apply the aforementioned strategies. The larger game boards proved to be very time consuming, which made it a non-viable area for us to possibly find a winning strategy. So in order to find a possible winning strategy or technique, we tended to focus more on the smaller sized boards in hopes that it may provide us with more information on the typical setup for a game of Mancala. The smaller sized boards also had less possible game states, making it much easier on the computers we used to produce these states.

Generation of States of Game Trees Using Python Programming

A program was created in Python by our research adviser, Dr. Kerstin Voigt. The program outputted all game states with parameters that were given by the user. The program’s parameters were not N and R as above; instead, we specified each game by the number of pits on each side (equivalent to (N-2)/2) and the initial number of counters in each pit (equivalent to R/(N-2)). This program allowed us to analyze and review all possible states of the game (including a text-based display of the board), the wins and losses of player A, number of captures, and any duplicate states. Duplicate states indicated that a certain state was reproduced somewhere else within all of the game states. The most reviewed game state by our research team was a board with two counters per pit, and two pits per side. The games were reasonably short, but the board size still provided enough depth for strategy to play a part. Since the games involving two counters and two pits on each side produced a relatively small number of game states, we were able to find ways that a player could react appropriately to their opponent's moves in order to gain an advantage. Through this analysis, game trees were also generated using the aforementioned program. These trees displayed a node and all possible moves from this node, otherwise called the ‘parent’ and the ‘children’ nodes. With these nodes generated by the program we were able to determine which branches led player A into a win, loss, or a draw. We further simplified these game trees by selecting the parent nodes which led to the safest options. Another issue presented itself – how can we tell which branch is the safer choice for player A? This led us to the development of an algorithm.

An Algorithmic Approach to Winning Mancala

This issue prompted us to develop an algorithm for determining the path that led to the greatest chance of winning for player A. The original development of the algorithm was difficult because we had to make a choice between two foundations: should player A go for a win only, or should player A go for a win while trying not to lose? These two options would affect the algorithm drastically. In the end, we chose that player A should go for the win no matter what. The original basis of the research was to provide a winning solution to Mancala, so this means that player A should attempt to secure a win at all costs. The algorithm itself determines the win percentage first. If the win percentages are the same, we also calculate the chance player A has to lose. If the chance for player A to lose is higher, then the algorithm determines that player A should choose the branch in which it has a lesser chance to lose. First the algorithm adds the number of wins, draws, and losses. The number of wins is used as the numerator while the sum of wins, losses, and draws is used as the denominator. This resulting number is represented as a percentage, which is used as a comparison for each branch. The algorithm calculates the losing percentage in a similar manner, except the number of losses is used as the numerator. The losing percentage will only be used if the winning percentages happen to be the same. If this occurs, the losing percentage with the lowest value will be chosen. Using this method we are still able to secure a higher chance at winning. We felt that these percentages provide an accurate portrayal of the chance of winning a game because we determined that losses and draws would be detrimental to a player's chance at winning. While losing obviously takes away from the chance of a player securing a win, declaring draws as a detriment yielded more debate. The main arguments presented for draws was that they were beneficial because it prevented the player from losing, but they were detrimental because it prevented the player from winning. Again, this decision would drastically affect how the algorithm works. Our original goal was to find a winning strategy for Mancala regardless of the costs, so we declared that draws should be considered as a negative. Therefore, an accurate percentage can be produced by dividing the number of wins by a sum of all three statistics – number of wins, losses, and draws. We also believe that the losing percentage would be accurate as well, because it follows the same formula and concept.

A Game Tree Based Recommender Function

We were able to work with a program that generated and displayed the game trees for smaller-sized games. A game tree is rooted in the initial game; for each state in the tree, all possible next states are generated as they would result from making the various possible moves. Winning and losing states, and states that are draws for both players make up the “leaves” (= end nodes) of the game tree. Working backwards from these known (since generated) endstates of the game makes it possible to determine which nodes are the best choice for player A. Dr. Kerstin Voigt provided our research team with an additional function included in the original program which recommended a move when the current game state was given as a parameter. Our thorough testing of this feature yielded a single logic flaw residing within the new function. In boards of certain ratios, the recommender function would choose an inappropriate game state based off of the calculations the program performed. In some instances our researchers could see a game-winning move that could be performed by the recommender, but the recommender opted out of these specific moves in some cases. The issue was located and fixed in a function that determined the best move for player B. The recommender was then performing as expected, and played very well. Of the various games we observed among our team, the player using the recommender as an aid tended to win most games. While we were not able to discover a reliable defined strategy for winning Mancala during the period of our research, we were able to find ways to react to an opponent’s move in order to increase our chances of winning.

Conclusion

Before we announce that Mancala has no set strategy for winning, we feel that we must explore possibilities with a normal sized board and possibly even larger sized boards. Within the scope of our research, a winning strategy for Mancala seems to be reacting to an opponent's moves with the best possible move available. The best move is determined by comparing the wins to losses and draws. This will result in several possible game states and a player will be able to choose the most optimal move by comparing percentages. This was the basis for the algorithm we designed for our research. This means that the winning strategy for Mancala varies depending on the move that the opponent will make. This conclusion seems to hold true for smaller sized boards as well. While a player may be able to skew a game in their favor, it is possible for the opponent to completely change the course of a game with certain moves. Therefore, the way that the first player reacts to these moves will determine the outcome of the game. There are simply too many possibilities for Mancala in order to identify a set strategy for winning.

Suggestions for Future Research

Additional areas of research that we encourage others to explore are the possibilities with larger boards. Many of the simulations we ran were limited to boards of smaller sizes due to the amount of computing power it took to process the possible game states. Our analysis indicated that as the number of pits increased on the variants of the boards we considered, the amount of combinations also increased drastically. As aforementioned, the ratio of certain boards can affect the relevance of certain strategies and in some cases it can even grant an advantage to a certain player. For those who have the appropriate computing power to handle the original board or even larger boards, it would be an interesting and informative area for additional research. Another possible area of research would be to incorporate an alternate algorithm.

References

[1] F. Dooley. (2008). *Best Opening Move in Mancala* [Online]. Available: http://fritzdooley.com/mancala/6_mancala_best_opening_move.html

[2] F. Dooley. (2008). *Mancala Strategies* [Online]. http://fritzdooley.com/mancala/5_online_guide_mancala_strategy.html

[3] J. Donkers, U. Jos, and A. Voogt. (2001). *Mancala Games - Topics in Mathematics* [Online]. Available:http://www.academia.edu/2543117/Mancala_games_Topics_in_Mathemathics_and_Artificial_Intelligence

[4] R. Pierce. (2011, January 28). *Combinations and Permutations* [Online]. Available: http://www.mathsisfun.com/combinatorics/combinations-permutations.html

[5] F. Dooley. (2008). *Structural Analysis of Mancala* [Online]. Available: http://fritzdooley.com/mancala/3_structural_analysis_of_macala_game.html

[6] F. Dooley. (2008). *Conclusion* [Online]. Available: http://fritzdooley.com/mancala/7_mancala_backward_induction_results.html

[7] Nils, Nilsson J. “Artifical Intelligence: A New Synthesis.” Morgan Kaufmann Publishers, Inc. 1998. 1st edition. Text. p.p 195.