Heuristic Mancala Players

Building a Series of Machines to Play Mancala

by Carrie Corcoran
SUNY Oswego, Spring 2023

Abstract

This project demonstrates the use of symbolic AI techniques in machine players of the game of Mancala. The associated project entailed creating infrastructure in CLISP to play Mancala, an interface for human vs machine play, and four machine players: a player that selects moves randomly, a heuristic player that selects a move resulting in an additional move, a heuristic player that selects the move with the single largest score increase, and a player that selects optimal moves using a Minimax tree. This paper describes that work in more depth and places it in the context of similar research on the topic.

1. Introduction

Mancala is a game that originated in East Africa. Archeological evidence indicates that the game has existed since at least 700 CE, if not substantially earlier.[I] There are many variants of Mancala, but the most common one is that of a two-player game in which each player attempts to maximize their score by getting the most stones into their Mancala, the long space at the end of the board. The players take turns picking up the stones from one of the smaller bowls on their side of the board and placing one stone in each subsequent bowl in a counterclockwise direction. Ending a move in that player's Mancala earns them an additional turn. This project uses a particular variant of Mancala known as continuous or multi-lap play with cross capture[II]. This means that if a player's last stone lands in a bowl that is not a Mancala that has stones in it, the player picks up the stones in the new bowl and continues playing. Cross-capture means that if the player's last stone lands in an empty bowl on their side of the board, they can pick up any stones in the opposite bowl and deposit them in their Mancala. This style of play can lead to long sequences of moves in a single turn, and a strong first-player advantage. The first player can change the state of the whole board with their opening move, and potentially even gain enough points to win the game with a particularly strong opening move sequence. This project simulates the game of Mancala using CLISP and creates multiple machines that can play the game with various levels of skill.

Mancala is an interesting game to use in developing heuristics and Minimax trees for move selection. Donkers, Uiterwijk and de Voogt pointed out that Mancala is deterministic.[III] There is no element of chance in Mancala. Everything that happens is the result of the players' actions. It is also a game that allowed for significant development of machine players within the project's time limit of a single semester. Mancala has relatively little structure when compared to other games like Chess. The game structure does not take a single undergraduate student months to build, which left ample time in the project for the development of machine players. This project involved building four machine players: a player that picks moves randomly, two heuristic players, and a player that selects moves using a Minimax tree.

2. Background

The game of Mancala has been used to explore topics in AI before. As Melanie Mitchell states, games are a useful microworld to test AI concepts without struggling with too many real-world variables[IV]. Some of the most famous AI game players are Deep Blue the chess player and AlphaGo the Go player. Several research projects have dealt directly with various forms of Mancala.

Gifford, Bley, Ajayi, and Thompson conducted work with Mancala using 7 different heuristics[V]. They used interesting metrics to examine the effectiveness of their heuristics, such as final score comparison and number of moves in a game. Ali, Gimo, and Saide worked with Minimax trees with a different variant of Mancala than this project uses.[VI] Mancala has also been used as a topic for undergraduate honor's theses. Hunter's project conducted very similar work to this project, using Minimax trees and heuristics.[VII] One of Hunter's heuristics is very similar to the Maximum Score Heuristic used in this project.

This project makes use of both heuristics and Minimax trees. Hjejj and Vilks define a heuristic as simple problem-solving methods that do not always result in an optimal solution, but work quickly and efficiently.[VIII] In contrast, Dem'yanov and Malozemov state that Minimax is "an important principle in optimal selection of parameters"[IX] These ideas are demonstrated in this project. The Minimax machine player takes substantially longer than either heuristic player to select a move, but this move will likely triumph against a heuristic player's move.

3. Program Description

This project was written entirely in Common Lisp, or CLISP. The four main aspects of the project are the development of game infrastructure to play Mancala including a machine player that can pick moves randomly, an interface for human-machine play, two heuristic machine players and a machine player that can select moves using a Minimax tree. The infrastructure was built primarily using property lists. Each space on the board corresponded to a property list that kept track of the number of stones in the space and the next space in the sequence.

The non-random machine players required a new set of infrastructure so that future moves could be examined without impacting the current game state. This was known in the project as the copy infrastructure. Each space had a corresponding copy space that could copy values from the main spaces and collectively simulate moves. This enabled the system to test sequences of moves without impacting the current game. This was very important as the heuristic players and the Minimax player both required the ability to look ahead at the result of potential moves.

One heuristic machine player looked for moves that would lead to an additional turn when possible. This was dubbed the Move Again Heuristic. Another heuristic machine player chose moves that would lead to the largest score increase in that turn. This was dubbed the Maximum Score Heuristic.

The final machine player used a Minimax tree to select an optimal move. The book *Minimax and Applications* notes that a full Minimax is an NP-Hard problem[X]. Running a full Minimax examination of all available moves would not be feasible both in terms of time and space. There are over 1200 opening moves for this variant of Mancala. Therefore, this project uses a Minimax tree to examine a subset of moves. A set of up to 500 moves is generated, then up to 50 moves are randomly selected from that set. The longest moves from this subset are selected in a set of up to 10 moves, and the minimax tree runs using those moves. The Minimax tree uses a static evaluation function to evaluate moves in a tree, and selects the optimal move. In this project, the static evaluation function calculates the difference between the players' scores. Player A is striving for positive values, while Player B is striving for negative values.

As each machine player was developed, it was added to the interface so a human player could play against it. The machine players also played against one another in testing to see which machine was the strongest player.

4. Demonstrations

This is the representation of the Mancala board, used in both human vs machine games and in visually representing machine vs machine games. Each space is labeled and the number of stones in each space changes throughout the game based on moves performed. The human player is always player A and goes first, for ease of reading the board. This variant of Mancala has a significant first player advantage. Gifford, Bley, Ajayi, and Thompson also found a first player advantage in their studies. It is likely that this variant exacerbates the advantage.

A human player enters a move by its space. For example, their move could be A3. This board is in the start game state.

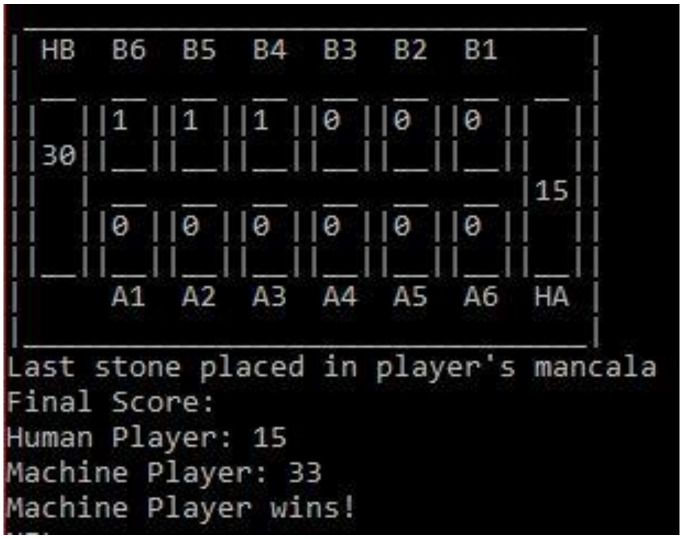
```
 ________________________________
| HB  B6  B5  B4  B3  B2  B1     |
| __  __  __  __  __  __  __  __ |
||  ||0 ||13||2 ||3 ||2 ||1 ||  ||
||13||__||__||__||__||__||__||  ||
||  |  __  __  __  __  __  __ |9 ||
||  ||1 ||1 ||0 ||0 ||0 ||3 ||  ||
||__||__||__||__||__||__||__||__||
|     A1  A2  A3  A4  A5  A6  HA |
|________________________________|

Machine player's turn!

Minimax Player's selected move: (B2 B6 B3 B5 B2 B4 B6 B1 B6 B4)
Next move: B2
```

Here is an example of the type of move the Minimax machine player selects, with the game state it was evaluating. The Minimax machine player favors longer moves.

It is very easy for a human player to win against the random machine player. The heuristic players require the human player to think more about their moves and play defensively in order to win. It is very difficult for a human player to win against the Minimax machine player. It requires deep knowledge of good starting moves.

```
 ________________________________
| HB  B6  B5  B4  B3  B2  B1     |
| __  __  __  __  __  __  __  __ |
||  ||1 ||1 ||0 ||0 ||1 ||0 ||  ||
||19||__||__||__||__||__||__||  ||
||  |  __  __  __  __  __  __ |26||
||  ||0 ||0 ||0 ||0 ||0 ||0 ||  ||
||__||__||__||__||__||__||__||__||
|     A1  A2  A3  A4  A5  A6  HA |
|________________________________|
Final Score:
Human Player: 26
Machine Player: 22
Human Player wins!
```

Results of two games of human vs Minimax machine player



The machine players were tested against one another in 100-game sets. Each player played both first player and second player in various tests, to examine the effect that first-player advantage has on the relative strength of the machine players.

Test Results

Player 1 Horizontal, Player 2 Vertical	Random Player	Move Again Heuristic Player	Maximum Score Heuristic Player	Minimax Player
Random Player	Player 1 wins: 49 Player 2 wins: 40 Ties: 11	Player 1 wins: 96 Player 2 wins: 2 Ties: 2	Player 1 wins: 85 Player 2 wins: 11 Ties: 4	Player 1 wins: 100 Player 2 wins: 0 Ties: 0
Move Again Heuristic Player	Player 1 wins: 17 Player 2 wins: 79 Ties: 4	Player 1 wins: 82 Player 2 wins: 12 Ties: 6	Player 1 wins: 28 Player 2 wins: 69 Ties: 3	Player 1 wins: 100 Player 2 wins: 0 Ties: 0
Maximum Score Heuristic Player	Player 1 wins: 19 Player 2 wins: 76 Ties: 5	Player 1 wins: 86 Player 2 wins: 11 Ties: 3	Player 1 wins: 82 Player 2 wins: 12 Ties: 6	Player 1 wins: 100 Player 2 wins: 0 Ties: 0
Minimax Player	Player 1 wins: 1 Player 2 wins: 97 Ties: 2	Player 1 wins: 48 Player 2 wins: 46 Ties: 6	Player 1 wins: 1 Player 2 wins: 98 Ties: 1	Player 1 wins: 100 Player 2 wins: 0 Ties: 0

These results demonstrate the first-player advantage in this variant of Mancala. When two machine players of the same type play, the first player is more likely to win. This difference becomes more prominent with stronger machine players. Two random players are relatively close in the number of wins, but two Maximum Score Heuristic players or two Move Again Heuristic players show a more marked difference. When two Minimax players play, the first player always wins. The only time the first player doesn't win the majority of the time is when a stronger player plays a weaker player and the weaker player goes first. An example of this is when a random player plays against a heuristic or Minimax player.

In terms of overall player strength, the Minimax player is the strongest. It wins every game in which it is the first player, and most games in which it is the second player. The one exception to this is when the Move Again Heuristic player went first against the Minimax player, which was almost evenly matched. This is likely because their move selection is similar. They both rely on long strings of moves for each turn.

The Move Again Heuristic Player is the stronger of the two heuristic players. It beats the Maximum Score Heuristic player the majority of the time even when the Maximum Score Heuristic player has first-player advantage. This may be because longer moves tend to result in higher scores overall.

5. Reflections and Conclusions

This project is successful in that the established goals were completed by the final deadline. The original project idea was to use a genetic algorithm for move selection, but after learning about Minimax trees, this seemed to fit the project better. The success of the project is largely due to successful planning. A game with relatively simple structure allowed more time to implement machine players. While there were some delays, the project selected was manageable in the time allotted. The delays mainly occurred in the development of the Minimax machine player. Some aspects of its development, such as recursive generation of moves, proved to be more difficult than originally thought. It is likely that the Minimax machine player would have been removed from the project had a breakthrough not occurred when it did. The finished product involved a good deal of flexibility and perseverance, and the developer is proud of the results.

While the goals of the project were accomplished, there are several ways to expand the project in the future. One would be to use another variant of Mancala, perhaps one that doesn't lead to such long sequences of moves. Another would be an additional heuristic which looks for the longest moves. The Minimax machine player uses a version of this in its preliminary move selection, but this could also be its own heuristic. It would also be interesting to gather additional data on the relative performance of the machines, such as relative end scores and the number of moves in a game. Finally, the current interface means that a human player always has first player advantage. A variation of the interface could be developed so that the human player can play either role, for further exploration.

Bibliography

[I] "Mancala"- Savannah African Art Museum- 2020- https://www.savannahafricanartmuseum.org/2020-workshops/05-2

[II] "Variants of Mancala" –UltraBoardGames- 2020- https://www.ultraboardgames.com/mancala/variations.php

[III] "Mancala games - Topics in Mathemathics and Artificial Intelligence" - Jeroen Donkers, Jos Uiterwijk and Alex de Voogt – 2001

[IV] *Artificial Intelligence- A Guide for Thinking Humans*- Melanie Mitchell- 2019

[V] "Searching and Game Playing: An Artificial Intelligence Approach to Mancala" Chris Gifford, James Bley, Dayo Ajayi, and Zach Thompson – 2008

[VI] A MiniMax Agent for Playing Ntxuva Game – The Mozambican Variant of Mancala - F. D. M. A. Ali, E. Gimo and S. M. Saide – 2020 - https://ieeexplore.ieee.org/abstract/document/9183848/metrics#metrics

[VII] "The Exploration and Analysis of Mancala from an AI Perspective" Trevon J Hunter- 2021- https://digitalcommons.andrews.edu/cgi/viewcontent.cgi?article=1259&context=honors

[VIII] "A brief history of heuristics: how did research on heuristics evolve?" –Mohamad Hjejj and Arnis Vilks- 2023 https://www.nature.com/articles/s41599-023-01542-z

[IX] *Introduction to Minimax* -V.F.Dem'yanov and V.N.Malozemov- Translation dated 1974, original work 1938 - https://books.google.com/books?hl=en&lr=&id=xRT-AgAAQBAJ&oi=fnd&pg=PP1&dq=minimax&ots=Bz5uljvT67&sig=UL8EeGM_5SIV4igJLf54iyHsTZM#v=onepage&q=minimax&f=false

[X] *Minimax and Applications*- edited by Ding-Zhu Du, Panos M. Pardalos-1995 https://books.google.com/books?hl=en&lr=&id=t2cMNX81U4oC&oi=fnd&pg=PR13&dq=minimax&ots=0LhQ-FNBBK&sig=iejJBTfldVoZhHRxDYj0vKWHiMg#v=onepage&q=minimax&f=false