Project Work Written Report
Category 8

Solutions and Tactics in Winning Chinese Checkers

Lebron Mah Kai Ze (1i3-15) [Leader]
Brandon Ong Jing Jie (1i3-01)
Xin Qingyang (1i3-25)
Zachary Phua Yu Tze (1i3-27)

Hwa Chong Institution (High School)

Contents

1 Introduction

1.1 Background Information

Chinese Checkers was invented in Germany in 1892 under the name "Stern-Halma" as a variation of the older American game Halma. The "Stern" (German for "star") refers to the board's star shape unlike Halma's square shape. Despite its name, Chinese Checkers does not have any relations to Chinese or Checkers but was named "Chinese Checker" as a marketing scheme.

1.2 Rationale

Since, it is a relatively simple game that is easy to understand and we are passionate to learn and play more, we would naturally want to find strategies on how to win Chinese Checkers.

1.3 Objective

The objective of our project is to find strategies that will allow a player to win a game of Chinese Checkers within the least possible number of moves.

1.4 Research Questions

We have come up with 3 research questions. Firstly, "How does "building bridges" throughout the game affect the chance of winning?". Secondly, "Does using "brute force" guarantee that a player will win?" Keywords like "building bridges" and "brute force" will be defined and explained later. Finally, "Does starting first affect the chance of winning and how is the chance of winning affected?" The keywords in the research questions will be defined later in the literature review.

2 Literature Review

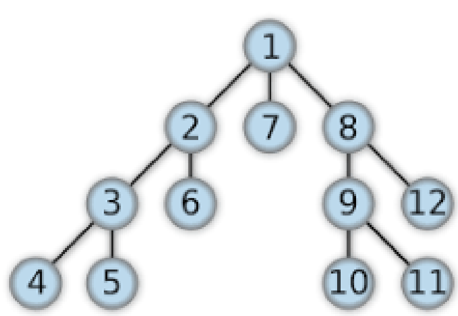
2.1 Introduction to the game

Chinese Checkers is a multiplayer game that can be played by 2, 3, 4 or 6 players. Each player chooses a colour and places his 10 pieces of his colour into the appropriately coloured triangle. The players then take turns in moving their pieces into the opposite triangle and the first player to do so wins. In each turn, the player can either move his piece into an adjacent space or move his piece over other single pieces.

2.2 Definitions of keywords used

As mentioned earlier, we would define keywords we will be using. Firstly, "hopping" is to jump over another adjacent piece into a hole/grid right behind it. "Hopping" can be done continuously in 1 turn until the player wishes to stop. Another way to move one's piece is to shift it to an empty adjacent hole (grid). However, an adjacent move cannot be done continuously in 1 turn. Secondly, a "combo" is a move in a turn in which the player "hops" continuously multiple times. Thirdly, "building bridges" is the action of placing one's pieces so that his other pieces can hop over them, making it easier for the entire movement of all pieces. Fourth, the "hopping distance" is how many holes (grids) a piece moves forward in one turn. Lastly, "brute force" is constantly "hopping" as far as possible without any contemplation.

2.3 Information & Strategies found



A strategy that we found online was created using a depth-first search algorithm written by Cedric Jules, a teacher with experience in the video game industry. He explains the algorithm as a system that uses depth-first search and tries all possible starting 5 moves. This way, the algorithm can compare all first 5 moves and deduce the best 5 starting moves. This allows the player using the strategy to, minimally in 5 turns, reach the center of the board, which is where their pieces will start meeting their opponent's. We found this strategy interesting as every move was decided based on its "hopping distance". Other sources that we found were from various sources like "WikiHow". They mainly focused on talking about how the opening moves affect the tide of the game and most of them suggested keeping your own pieces close together. If not, it would be very hard to move around the board because it is harder to do combos and hop multiple times. Thus, moving the pieces to the end zone would be slower. Another source from "Semantic Scholar" proved that the least number of moves to win a game of chinese checkers is where both players collaborate to achieve a win. Each player uses only 15 moves in this scenario.

3 Methodology

3.1 Strategies to develop project

To answer our research questions, we played games against each other while using strategies in the first 2 research questions. We also created an artificial intelligence (AI) programme that tests these 2 strategies as well. Whereas for the third research question, we extracted data from the manual games and AI programme to calculate our findings.

3.2 Applications used to help with results and findings

As we gave research questions that required data collection and analysis, we decided to use Google Sheets to help us with formulating, formatting and tabulation of results for manual games. As for the AI, we used C++ as the coding language and cocos2d-x game engine as the application's framework.

4 Study

4.1 Results and findings (Manual Games)

For the games we played, we experimented on strategies that we were thinking about in our 2 research questions, namely "building bridges" and "brute force". The following is our results for the 2-player games.

Number of Players	Games & Strategies Used	Research Question 1: How does "building bridges" throughout the game affect the chance of winning?	Research Question 2: Does using "brute force" guarantee that the player will win?
2 Players	**Game 1:** Player 1 & 2: Building Bridges	Player 1: 47 Player 2: 50	
	Game 2: Player 1: Building Bridges Player 2: Brute Force	Player 1: 50	Player 2: 50
	Game 3: Player 1 & 2: Brute Force		Player 1: 37 Player 2: 62

We have also played 3-player games. The following is our results.

Number of Players	Games & Strategies Used	Research Question 1: How does "building bridges" throughout the game affect the chance of winning?	Research Question 2: Does using "brute force" guarantee that the player will win?
3 Players	**Game 1:** Players 1,2,3: Building Bridges	Player 1: 39 Player 2: 51 Player 3: 35	
	Game 2: Players 1,2: Building Bridges Player 3: Brute Force	Player 1: 47 Player 2: 40	Player 3: 54
	Game 3: Player 1: Building Bridges Players 2,3: Brute force	Player 1: 39	Player 2: 46 Player 3: 49
	Players 1,2,3: Brute Force		Player 1: 41 Player 2: 41 Player 3: 44

As for the 3rd research question, we extracted the information from the results of our manual games and compiled all the information into a table. The following is our results for the 2 player games:

Number of Players	Games & Strategies Used	Player 1 Wins	Player 1 Loses/Draws
2 Players	Game 1: Players 1 and 2: Building Bridges	✔	NIL
	Game 3: Players 1 and 2: Brute Force	✔	NIL

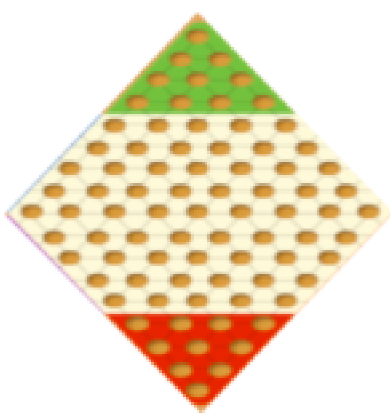
We have also played 3-player games, the following are our results:

Number of Players	Games & Strategies Used	Player 1 Wins	Player 1 Loses/Draws
3 Players	Game 4: Players 1, 2, 3: Building Bridges	NIL	✔
	Game 7: Players 1, 2, 3: Brute Force	NIL	✔

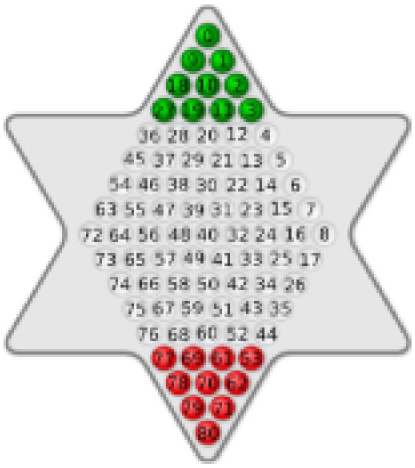
Next, the analysation of data. For Research Questions 1 & 2, the results show that building bridges is better than brute force. For 2-player games, the results for building bridges and brute force are almost equal but the total number of moves needed for 2 players using building bridges is 2 less than the total number of moves needed for 2 players using brute force. For 3-player games, it is quite evident that building bridges is better than brute force. The total number of moves needed for 3 players using building bridges is 1 less than the total number of moves needed for 3 players using brute force. In games 2 and 3 for 3-player games, the players using building bridges also needed less moves than those using brute force.

4.2 Results and Findings (AI Simulation)

Traditional chinese checkers supports up to 6 players but in the AI application, we only support 2 players for simplicity. When the board is cleared of the other 4 players, we can simplify the board to a 9 by 9 table.



Then, we assign a Grid ID to every hole (grid) from 0 to 80 as such:



In order for the computer to identify where the pieces are, we use an internal representation of a game state where 0 = empty grid, 1 = first player and 2 = second player. When the game initially starts, the game state looks like this:

1	1	1	1	0	0	0	0	0
1	1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	2
0	0	0	0	0	0	0	2	2
0	0	0	0	0	0	2	2	2
0	0	0	0	0	2	2	2	2

To calculate the forward distance of each move, we use the following formula:

Current Location = Row1, Column1
Target Location = Row2, Column2
Forward Distance = (Row2 + Column2) + (Row1 + Column1)

However, for the second player, we have to reverse the calculation formula to:

Forward Distance = (Row1 + Column1) + (Row2 + Column2)

because the second player is going from a region of high Grid ID to a region of lower Grid ID.

Now, to apply the 2 strategies, we need to find the furthest move. First, we find all moves, whether it is an adjacent step or a hop. For adjacent steps, we need to find all empty holes surrounding the chosen piece. For hops, we will find all the moves using Depth First Search. Then we:

1. Calculate the forward distance for every move
2. Choose the one with the greatest forward distance
3. If there is a tie, choose from them randomly

Now, to apply the strategies. For "Brute Force", we just need to find the furthest move. However for Building Bridges, if the furthest move has a Forward Distance of less than 6, we try to find the 2 furthest consecutive moves. When the AI builds the bridge, it makes sure that it will not benefit the opponent.

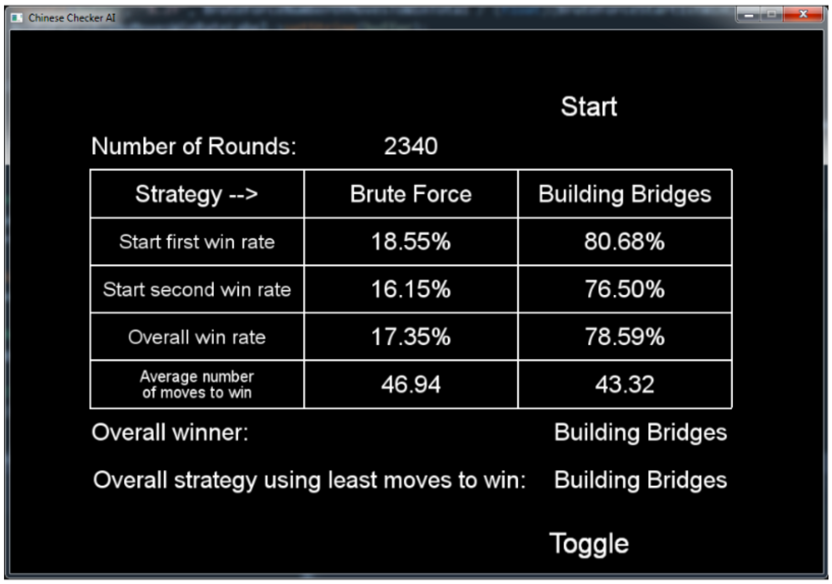
Here is a sample of the code:

```
CPawn* CBoard::PickPawn(const cocos2d::Point& Location)
{
    // Find the picked grid by mouse location
    int PickedGrid = PickGrid(Location);

    if (PickedGrid == INVALID_GRID_INDEX)
    {
        // No valid grid picked, no valid piece also
        return nullptr;
    }

    // give the piece on picked grid
    return FindPawnOnGrid(PickedGrid);
}

int CBoard::PickGrid(const cocos2d::Point& Location)
{
    // Iterate every grid to find the one which is close to the mouse location (within the GridRadius)
    for (int GridID = 0; GridID < GRID_COUNT;   ++GridID)
    {
        const cocos2d::Point& GridPosition = Grids[GridID];

        if (GridPosition.distance(Location) < GridRadius)
        {
            return GridID;
        }
    }

    return INVALID_GRID_INDEX;
}
```

To create the statistics table, the application draws the table first, then runs the simulation and collects data. It takes around 1.5 seconds to run 1 round.

Then, the data is presented in the table and the application makes its conclusion.

For example, after about 1 hour of calculation, here are the results: Here is a sample of the code:



It is evident that “Building Bridges” is much better than “Brute Force” and “Building Bridges” also requires less moves to win. In response to our third research question, starting first slightly increases the chance of winning.

5 Conclusions

From the manual games, we can conclude that building bridges is a better strategy overall than brute force and that is shown more prominently in 3-player games than 2-player games. From the AI application, we can conclude that “Building Bridges” has a higher win rate than “Brute Force” and also requires less moves to win. It also proves that starting first increases the chance of winning, but only slightly by about 3.29%.

6 References

1. Bell, G. I. (1970, January 01). The shortest game of Chinese Checkers and related problems. Retrieved August 12, 2018, from https://www.semanticscholar.org/paper/The-shortest-game-of-Chinese-Checkers-and-related-Bell/cb1a4719a273900053eded2444db1d9ac00b7e7a

2. Chinese checkers. (2018, March 12). Retrieved March/April, 2018, from https://en.wikipedia.org/wiki/Chinese_checkers

3. C. J. (2013, June 10). Chinese checkers best opening moves. Retrieved March/April, 2018, from http://totologic.blogspot.sg/2013/06/chinese-checkers-best-opening-moves.html

4. Chinese Checkers Rules and Gameplay. (n.d.). Retrieved March/April, 2018, from http://www.ducksters.com/games/chinese_checkers.php

5. Depth-first search. (2018, March 12). Retrieved March/April, 2018, from https://en.wikipedia.org/wiki/Depth-first_search

6. How to Play Chinese Checkers. (2018, February 19). Retrieved March/April, 2018, from https://www.wikihow.com/Play-Chinese-Checkers

7. M. (n.d.). The Rules of Chinese Checkers. Retrieved March/April, 2018, from https://www.mastersofgames/rules/chinese-checkers-rules.html

8. Online Chinese Checkers - Free Brain Game. (n.d.). Retrieved March/April, 2018, from https://www.memory-improvement-tips.com/online-chinese-checkers.html