On the Complexity of Dark Chinese Chess

Cong Wang, Tongwei Lu
School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430073, China
E-mail: xyfcw@qq.com

Abstract: Complexity is an important metric to measure the difficulty of the game and is widely used in AI research on board games. This paper provides a complexity analysis for the game of dark Chinese chess (a.k.a. “JieQi”), a variation of Chinese chess. Dark Chinese chess combines some of the most complicated aspects of board and card games, such as long-term strategy, large state space, stochastic, and imperfect information, which make it closer to the real world decision-making problem. Here we evaluate dark Chinese chess from a game theoretic perspective. We design a self-play program to calculate the game tree complexity and average information set size of the game and propose an algorithm to calculate the number of information sets.

Key Words: complexity analysis, dark Chinese chess, imperfect-information games, self-play

1 INTRODUCTION

In recent years, artificial intelligence (AI) has been competing with human beings in the field of games, and has made great progress in perfect-information games like Go [1, 2]. In some sense, two-player zero-sum perfect-information games have been solved by the learning algorithm AlphaZero [3]. Therefore, the focus of recent research turns to imperfect-information games, where the players compete with others in a partially observable environment [4-8].

Dark Chinese chess is a variation of Chinese chess, which is a stochastic and imperfect-information game. Perfect-information board games such as Chess, Go and Chinese Chess all have variants of imperfect-information [9-13]. For example, Banqi is a variant of Chinese chess. In the beginning of the game, all pieces are randomly placed on the board with piece icons facing down and can only be moved when they are turned over [13]. Dark Chinese chess can be viewed as an upgraded version of Banqi, which increases the size of state space on the basis of retaining the characteristics of imperfect information.

In the field of perfect-information games, general algorithms such as AlphaZero can not be applied to dark Chinese chess, because the algorithm does not have the ability to deal with random, imperfect information states. In the field of imperfect-information board games, we can explore the CFRL algorithm [14] and its variants combined with some abstract technologies have made a breakthrough in Texas Hold'em poker [4-6]. But these algorithms can not be directly applied to dark Chinese chess. First, dark Chinese chess not only contains uncertain factors, but also contains the huge action space common in board games. Second, different from card games, the value of the pieces in the game is related to the position, and the value of the pieces in different positions varies greatly. The above features of the game make these algorithms difficult to use.

Recent advancements in imperfect information games are also notable. In particular, DouZhu [15], a AI system for DouDizhu, combines enhanced traditional Monte Carlo methods with neural networks and parallel actor mechanism to avoid imperfect information difficulties and increase efficiency. This approach is considered to be applicable to similar imperfect-information games. But it still suffers from the high variance of the Monte Carlo method, which can be disastrous for game with long length (the game length represents the average number of plies of the game. A ply is a single move taken by a single player), such as dark Chinese chess.

As abstractions of the real world, games are an ideal platform for testing AI technology. In particular, board games are usually regarded as the epitome of war and have a special status. Dark Chinese chess can be regarded as the epitome of modern war and the duel between the two sides in an uncertain environment. The difficulties caused by the lack of perfect information make dark Chinese chess a challenge to AI technology.

In the research of game AI, complexity is an important metric, especially in board games. One of the reasons why Go is known as the most complex board game is the extremely high complexity (The state space complexity of Go is of the order of 172, and the game tree complexity is of the order of 360) [16].

In this work, we design a self-play program and propose a algorithm to measure the complexity of dark Chinese chess. The results show that it adds uncertainty similar to card games while maintaining the complexity of chess strategy.

2 DARK CHINESE CHESS

Dark Chinese chess is a variation of Chinese chess (a.k.a. “JieQi”), which has increased in popularity in China and Vietnam with tournaments held every year.

The pieces and board in dark Chinese chess are the same as those in Chinese chess. In the beginning of the game, all pieces except Kings will be flipped down and placed on the board out of order as shown in Figure 1. These pieces are now called dark pieces. Dark pieces move according to the Chinese chess rules of their positions. For example, the dark pieces in the original position of the pawn in Chinese chess must move as pawns (i.e. only one step forward). They will be flipped up randomly in the first move of them. The dark piece being flipped is called a revealed piece. Then, the revealed pieces move according to the rules of Chinese chess. Note that
different from Chinese chess, the Guard and Minister can cross the river in dark Chinese chess.

The game ends under three conditions. First, the game ends when the king of one player is captured. Second, when the "meet the marshals" occurs, the player who makes the "meet the marshals" situation loses. Finally, the game ends in a draw when both players do not capture a piece within 40 plies.

It is worth mentioning that if a player captures an opponent’s dark piece, the opponent will learn that he or she loses a dark piece, but does not know what it is. This means that the information known to both players goes from symmetric to asymmetric, which not only increases the fun of the game, but also obviously makes the game more difficult.

3 GAME COMPLEXITY

In general, the complexity of a game can be measured by state space complexity (SSC) and game tree complexity (GTC) [16]. Game tree complexity represents the number of all different game paths for a game. The state space complexity of a game is the total number of states that can be reached in accordance with the rules since the initial state of the game. However, in imperfect-information games, a reasonable game strategy should be based on the information set rather than the game state, due to the information is imperfect and asymmetric. Accordingly, when we measure the complexity of imperfect-information games, we should measure the number of information sets, not the number of the game's states.

3.1 Game Tree Complexity

Obviously, the exact value of game tree complexity is difficult to calculate. The common method is to estimate its reasonable lower bound. We choose to use Shannon's method of calculating the size of the game tree [17] to estimate the game tree complexity of dark Chinese chess. The approach is illustrated by formula (1), where $b$ represents branching factor which means the average number of legal moves per turn by the player, and $p$ represents the average length of the game. In the calculations, the length of the game is measured in plies. A ply is a single turn completed by a single player, which is significantly different from a round.

$$GTC \geq b^p$$  \hspace{1cm} (1)

3.2 Number of Information Sets

We use the number of information sets to measure the size of imperfect-information games. An information set is a game state that is indistinguishable from a particular player's perspective. For example, in the beginning state of dark Chinese chess, we cannot distinguish what the dark piece is, so all possible states will be grouped into the same information set. The size of a finite game is a fixed number, but often difficult to calculate. So a upper bound can be calculated instead. Here, we assume that all pieces are of different types and have two states (dark or revealed) in any position. There are 32 different types of pieces, and the size of the chessboard is 90. Therefore, we can calculate the upper bound of the number of information sets according to the number of pieces on the chessboard.

$$\sum_{n=0}^{32} C_{32}^n \cdot A_{40}^{15} \cdot 2^n \approx 10^{66}$$  \hspace{1cm} (2)

The detailed calculation is given by formula (2), where $n$ represents the number of pieces on the chessboard. We can get that the upper bound of the number of information sets is $10^{66}$.

3.3 Average Size of Information Sets

Another term used to measure imperfect-information games is the average size of the information set, which is the average number of indistinguishable game states in the information set. This metric reflects how difficult it is to evaluate the quality of a state of the game.

The average size of information sets of some card games can be deduced mathematical method, but the pieces of dark Chinese chess are related to position, so the average size of the information set is difficult to be deduced in this way. But it is easy to figure out the information set size of the beginning state of the game is $10^{17}$ roughly.

4 Experiment

4.1 Monte Carlo Simulation

Game tree complexity (GTC) and mean information set size are usually calculated with the data of human games. This method is easy to operate but not reasonable. Complexity is a preparation for search algorithms, but human data has a large number of stereotypes, which will incorrectly evaluate it. Therefore, we use the Monte Carlo simulation to calculate GTC and mean information set size of dark Chinese chess through a large number of samples. Monte Carlo simulation is not biased against the game, so it can better evaluate the complexity. We designed a self-play program, in each ply of game, the program computers and stores all legal moves and choose one from it randomly to play game until it is over. We played 10000 self-play games, saved the length of the game and calculated the average legal moves and mean information.

Fig 1. The beginning state of the game.
set size in each game. The specific process is shown in Figure 2. Then we calculated the average size of the data, which means adding the data of each round and dividing it by the number of rounds of the game. The results are shown in Figure 3.

As can be seen from Figure 3, the data fluctuated a little in the early stage and then stabilized. Results show that the branching factor is about 34, the game length is about 133 and the average of information set is $|0|^{15}$. Therefore, it is easy to calculate that the game tree complexity of the game is $10^{205}$.

4.2 Calculate Number of Information Sets

We calculate the number of information sets from the perspective of the red player because of the symmetry of the board and the pieces (the type and number of pieces on both sides are the same). The kings exist in all statistical states, so we can only consider the 30 pieces left and the 88 positions on the 9×10 chessboard (the kings need to occupy two positions). For the convenience of calculation, we assume that the 15 pieces on each side are different.

We determine the state (dark or revealed) and number of pieces inside and outside the chessboard to calculate the number of information sets in the state. In order to avoid repeated calculation, we use the three-dimensional array $R$ to store all the combined numbers when the state (dark or revealed) and number of red pieces inside and outside the chessboard are determined. Specifically, each dimension represents the number of red pieces on the chessboard, the number of red dark pieces on the chessboard and the number of red dark pieces outside the chessboard respectively. From the perspective of the red side, the captured black chess, whether revealed or dark, does not affect the red side's judgment of the black chess in the chessboard. Therefore, we use the two-dimensional array $B$ to store all the combined numbers when the number of black pieces and black dark pieces in the chessboard are determined. It is worth noting that when all the red chess pieces on the chessboard are revealed, the red chess outside the chessboard need not be considered. See Algorithm 1 for the specific implementation.

The number of dark Chinese chess information sets we calculate is about $10^{37}$.

Algorithm 1 Calculate Number of Information Sets

Output: $num$

1: Initialize $R[i][j][k] \leftarrow C_{15}^i \times C_{15}^{i-1} \times C_{15}^{i-1}$.
2: $B[i][j] \leftarrow C_{15}^i \times C_{15}^{i-1}$ and $num \leftarrow 0$
3: For $R_{on} = 0, 1, \ldots$ until 15 do
4: \hspace{1em} For $B_{on} = 0, 1, \ldots$ until 15 do
5: \hspace{2em} $A_{on} \leftarrow R_{on} + B_{on}$
6: \hspace{1em} For $B_{dk} = 0, 1, \ldots$ until $B_{on}$ do
7: \hspace{2em} $A_{dk} \leftarrow R_{dk} + B_{dk}$
8: \hspace{2em} if $R_{dk} = 0$ then
9: \hspace{3em} $B_1 \leftarrow C_{15}^{R_{on}} \times B[B_{on}] [B_{dk}]$
10: \hspace{3em} $\times B[|A_{on}|, |A_{dk}|, C_{15}^{R_{dk}} \times C_{15}^{R_{dk}}$
11: \hspace{3em} $num \leftarrow num + B_1$
12: \hspace{2em} Else
13: \hspace{3em} For $R_{fdk} = 0, 1, \ldots$ until $15 - R_{on}$ do
14: \hspace{4em} $R_{dk} \leftarrow R_{fon} + R_{fdk}$
15: \hspace{3em} $B_2 \leftarrow R[R_{on}][R_{fon}][R_{fdk}] \times B[B_{on}] [B_{dk}]$
16: \hspace{4em} $\times B[|A_{on}|, |A_{dk}|, C_{15}^{R_{dk}} \times C_{15}^{R_{dk}}$
17: \hspace{3em} $num \leftarrow num + B_2$
18: \hspace{2em} End for
19: \hspace{2em} End if
20: \hspace{2em} End for
21: \hspace{1em} End for
22: \hspace{1em} End for
23: \hspace{1em} End for

Fig 3. The average size of branching factor, game length and information set with the increase of game rounds.
4.3 Comparison to Other Games

We compare the complexity of dark Chinese chess with some popular games, and the results are shown in Table 1 and Figure 4. The figure and table clearly show that, on the basis of Chinese chess, the difficulty of the game is greatly increased simply by flipping down all pieces except Kings. Dark Chinese chess not only increases the average game length, but also contains the difficulties of card games.

Table 1. Compared to the game tree complexity of other games

| Game            | Branching factor | Average game length | Game-tree complexity (as log to base 10) |
|-----------------|------------------|---------------------|------------------------------------------|
| Gomoku (15*15)  | 210              | 30                  | 70                                       |
| Chess           | 35               | 70                  | 123                                      |
| Chinese chess   | 35               | 95                  | 150                                      |
| Dark Chinese chess | 35         | 133                 | 205                                      |
| Go              | 250              | 150                 | 360                                      |

5 Conclusion

The main goal of this paper is to calculate the complexity of dark Chinese chess. The complexity of Chinese chess can be measured from the game tree complexity, the number of information sets and the size of information sets. We design a self-play program and propose an algorithm to obtain these quantities, which provides a basis for the search algorithm.

This paper is the first attempt to study the interesting and complicated game. We hope it can promote future researches for dark Chinese chess.

ACKNOWLEDGMENT

We would like to thank Xiaoyong Mao. Our program is based on his work.

REFERENCES

[1] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ..., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. nature, 529(7587), 484-489.
[2] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ..., & Hassabis, D. (2017). Mastering the game of go without human knowledge. nature, 550(7676), 354-359.
[3] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ..., & Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science, 362(6419), 1140-1144.
[4] Brown N, Sandholm T. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals[J]. Science, 2018, 359(6374): 418-424.
[5] Moravěk, M., Schmid, M., Burch, N., Lisié, V., Morrill, D., Bard, N., ..., & Bowling, M. (2017). Deepspade: Expert-level artificial intelligence in heads-up no-limit poker. Science, 356(6337), 508-513.
[6] Brown N, Sandholm T. Superhuman AI for multiplayer poker[J]. Science, 2019, 365(6456): 885-890.
[7] Li, J.; Koyama, S.; Ye, Q.; Liu, G.; Wang, C.; Yang, R.; Zhao, L.; Qin, T.; Liu, T.-Y.; and Hon, H.-W. 2020. SuphX: Mastering Mahjong with deep reinforcement learning. arXiv preprint arXiv:2003.13590.
[8] Zha, D., Xie, J., Ma, W., Zhang, S., Lian, X., Hu, X., & Liu, J. (2021). DouZero: Mastering DouDizhu with Self-Play Deep Reinforcement Learning. arXiv preprint arXiv:2106.06135.
[9] Cazanave, T. (2005, September). A phantom-go program. In Advances in Computer Games (pp. 120-125). Springer, Berlin, Heidelberg.
[10] Russell, S., and Wolfe, J. 2005.Efficient belief-state AND-OR search, with application to Kriegspiel. In Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI-05, 279-285.San Francisco, CA, USA: Morgan Kaufmann Publishers
[11] Markowitz J, Gardner R W, Llorens A J. On the complexity of reconnaissance blind chess[J]. arXiv preprint arXiv:1811.03119, 2018.
[12] Inc.Cicarini, P., and Favini, G. P. 2010. Monte Carlo tree search in Kriegspiel. Artificial Intelligence 174(11):670 – 684.
[13] Chen B N, Shen B J, Hsu T. Chinese dark chess[J]. Icga Journal, 2010, 33(2): 93-106.
[14] Zinkevich, M., Johanson, M., Bowling, M., and Piccione.C. Regret minimization in games with incomplete information. In Advances in Neural Information Processing Systems, 2008.
[15] Zha D, Xie J, Ma W, et al. DouZero: Mastering DouDizhu with Self-Play Deep Reinforcement Learning[J]. arXiv preprint arXiv:2106.06135, 2021.
[16] Allis, L. V. (1994). Searching for solutions in games and artificial intelligence (pp. 21-152). Wagenmeng: Ponsen & Looijen.
[17] Shannon, C. E. 1950. Programming a computer for playing chess. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 41(314):256-275.