Analysis of Quoridor by reusing the results of reduced version

Takuro Iwanaga
Graduate School of Engineering, Miyazaki University, Japan
E-mail: hm17008@student.miyazaki-u.ac.jp

Makoto Sakamoto
Faculty of Engineering, University of Miyazaki,
Miyazaki-City, Miyazaki, Japan

Takao Ito
Graduate School of Engineering, Hiroshima University
Higashi-Hiroshima, Hiroshima, Japan

Satoshi Ikeda*
Faculty of Engineering, University of Miyazaki, Miyazaki-City, Miyazaki, Japan,
E-mail: bisu@cs.miyazaki-u.ac.jp

**Corresponding Author*

Abstract

Retrograde analysis is a representative game analysis method that first lists all legal positions and searches for the best move by analyzing from the final positions to the initial position. This method is effective for analyzing the type of game in which the same position appears many times when the game is analyzed sequentially from the initial position. However, it also has the drawback of requiring a huge amount of memory to enumerate the legal solutions. In this paper, we take up the board game "Quoridor" released by Gigamic.

Keywords: Perfect play, Retrograde Analysis, Quoridor, combinatorial theory

1. Introduction

Game tree is a graph structure that represents a game as a directed graph with game positions as nodes and players' moves as edges [1]. The standard method for game analysis is to expand this game tree. Fig1 is a game tree of tic tac toe. A complete game tree will always find the best move. However, there are some problems with this method. One of the problem is the possibility of repeating the same positions. For Tic-Tac-Toe, the number of squares that can be placed on the board decreases with each move to an arbitrary position, and positions that have appeared before the current position will never appear again. However, in chess and shogi, once a move is made, it can be moved back to the previous position, so the same position can appear many times at different nodes of game tree.

Retrograde analysis was devised to solve this problem. Retrograde analysis enumerates all regal positions and propagates the win/loss information from the final position where the winner is decided to the initial position. When the win-loss information is no longer updated, the initial position is classified as either a must-win game, a must-lose game, or a tie for the first player.

Retrograde analysis avoids the loop in the game tree that occurs when the game continues to move through several positions. However, Retrograde analysis requires. that all regal positions be enumerated and used in the analysis, which creates a space-computing problem for keeping track of the game. Therefore, it is necessary to reduce the number of enumerated positions as much as possible.

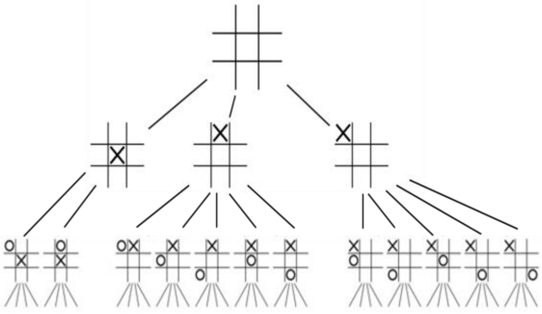

Fig 1The game tree of Tic-tac-toe. This figure shows all moves up to the second move. With a complete game tree, the best move can always be found.

The purpose of this study is to reduce the number of positions in the retrograde analysis by reusing the results of the reduced version. In this study, the phase set is partitioned as shown in Fig. 2. Here, the reduced version of the original game is the problem in which the number of items or board size has been reduced.

2. Quoridor

In this paper, we deal with a miniature board Quoridor[2]. This section describes the rules of a 5x5 board for two-player with one fence each.

2.1. *Object of the Game*

Object of the game is the same as the standard version [3], to be the first to reach the line opposite to one's base line.

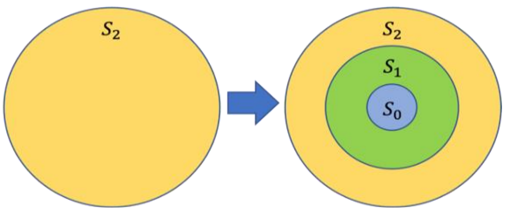

Fig 2 Before and after reuse. When the set of phases S_k(k=0,1,...) is $S_0 \subset S_1 \subset S_2$... as shown in the figure, and the only phase transitions are between phases in S_k and from S_k to S_{k+1}, the results analyzed in S_k(k=0,1,...,n-1) can be reused. This reduces the number of phases that need to be enumerated in S_n. Note that S_{k-1} is a reduced version of S_k in which the number of items is reduced by one step.

2.2. *Game Play (2 players)*

Each player in turn, chooses to move his pawn or to put up one of his fence. When he has run out of fences, the player must move his pawn.

At the beginning the board is empty. Choose and place your pawn in the center of the first line of your side of the board, your opponent takes another pawn and places it in the center of the first line of his side of the board (the one facing yours). Then take one fence each.

2.3. *Pawn moves*

As shown in Fig 3, the pawns are moved one square at a time, horizontally or vertically, forwards or backwards, never diagonally. The pawns must bypass the fences. If, while you move, you face your opponent's pawn you can jump over.

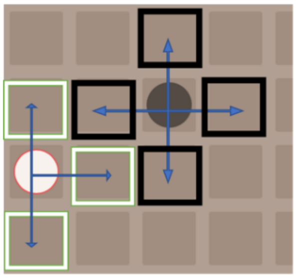

Fig 3 How to move pawn. The white square is where the white pawn can move and the black square is where the black pawn can move.

2.4. *Positioning of the fences*

The fences must be placed between 2 sets of 2 squares. By placing fences, you force your opponent to move around it and increase the number of moves they need to make. However, you are not allowed to lock up to lock up your opponents pawn, it must always be able to reach it's goal by at least one square.

2.5. *Face to face*

As shown in Fig 4, when two pawns face each other on neighboring squares which are not separated by a fence, the player whose turn it is can jump the opponent's pawn (and place himself behind him), thus advancing an extra square.

If there is a fence behind the said pawn, the player can place his pawn to the left or the right of the other pawn.

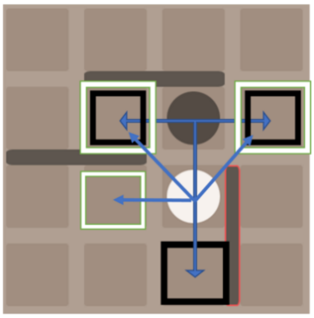

Fig 4 When two pawns are next to each other or when the path is blocked by a fence.

2.6. *End of the game*

The first player who reaches one of the 5 squares opposite his base line is the winner.

3. Retrograde Analysis

In this study, we conducted an experiment using retrograde analysis[4][5]. This method goes back one step at a time from the final stage where the victory or defeat is decided toward the initial board. In the process, if the previous move is connected to the victory phase, the victory information is received, and if all are connected to the defeat phase, the defeat information is received and the flow is repeated, so that the victory or defeat of the first phase can be known. The update of win/loss information is performed as shown in Fig 5. The advantage of this method is that you can also consider the case of a tie, which involves repeating the same move with each other, which is called "Sennichite".

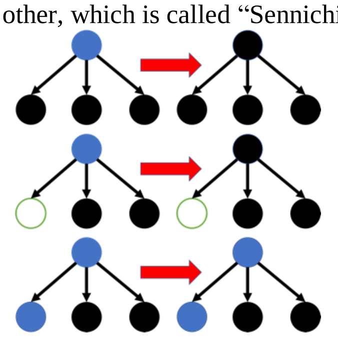

Fig 5 Blue represents an undecided game, white represents a game won by the white player, and black represents a game won by the black player.

4. Research Methods

In order to reduce the number of games to enumerate, we reuse the results of the reduced version of the game. The reuse of the results is done by using the results of the inclusion groups when performing backtracking analysis on each of the games that produce irreversible moves. In this paper, the set relation is defined as $S'_k=Sk\setminus S_{k-1}$ (k=1,2,3,...) for $S_0 \subset S_1 \subset S_2$.... S_0 is the part that includes the initial phase.

When the value of S_k is unchanged, the entire analysis in S_k can be shortened because it does not affect the final result. When the value of S_k changes, it is updated until the phase information of the changed part is no longer propagated.

In addition, the search for all possible fronts is necessary for the regression analysis, so the regular 9x9 board size with a total of 20 fences is too large to be handled. Therefore, in this experiment, a reduced board with a smaller board size and fewer fences is used.

The size of the reduced board size is increased in each experiment, with 5x5 as the standard size. However, only odd numbers of horizontal lengths are used. This is to fix the rule that the initial placement starts from the center in front of the player. The set of games with zero fences is denoted by S_0, and the set of games with one fence is denoted by S_1. The fence is denoted by S_{10} if it was placed by the first player and by S_{01} if it was placed by the second player. The number of fences is based on each player having one fence. When two fences are placed, S_{20}, S_{11}, and S_{02} exist, but only S_{11} is used when each player has one fence available.

5. Results

The results of the experiment are shown in the Table 1.

Table 1 Number of games won or lost when the number of fences set up in the game decreased.

Number of fences installed	Number of all phases	Number of phases in which win/loss information changed
0 fence	960	764
1 fence	61440	49273
2 fence	436364	

When the game with two fences was not solved and the player moved from a game with one fence to a game with zero fences, the win/loss information for 764 games was swapped. When moving from a phase with two fences to a phase with one fence, the win/loss information for 49273 stations was swapped. When the number of fences held by both players is zero or one each, both players must win the game [6]. When there are zero fences(S_0), the late player has the advantage, but when the one fence is placed(S_1), the game changes to the first player's advantage in almost all situations, regardless of which player places the fence. Next, when the game transitions from a game with one fence(S_1) to a game with two fences(S_2), the game changes from first move advantage to second move advantage. This means that when there are an odd number of fences, the game is biased in favor of the first player, and when there are an even number of fences, the game is biased in favor of the second player. This made it almost impossible to shorten the analysis. However, by using the results of the reduced version in enumerating the phases, we were able to analyze only the added phases. In addition, we were able to reduce the number of phases to be handled at one time by separating the phases with the same number of fences that did not interfere with each other.

6. Conclusion

The way the set of phases was divided in this case, the win/loss information was almost completely changed because of the large change in advantage/disadvantage. Therefore, in addition to irreversible transitions to the next phase, it is necessary to make the division of the phase set into segments where the advantage/disadvantage does not change significantly. In the Quoridor used in this study, the results for the combined set of S_2 and S_1 can be compared with S_0 to further reduce the portion that must be reanalyzed.

References

1. Hu, Te Chiang; Shing, Man-tak, *Combinatorial Algorithms*. Courier Dover Publications. 2002.
2. https://quoridor.jp/#
3. "Quoridor Game Rules" https://www.ultraboardgames.com/quoridor/game-rules.php
4. Teturo Tanaka An Analysis of a Board Game "Doubutsu Shogi" Journal of Information Processing, Vol. 48, No. 11, pp. 3470–3476(2007).
5. J. Romein and H. Bal: Solving the Game of Awari using Parallel Retrograde Analysis, IEEE Computer, Vol. 36, No. 10, pp. 26 – 33(2003).
6. Takuro IWANAGA, Makoto SAKAMOTO, Takeo ITO, Satoshi IKEDA, Analysis of Quoridor, 2022 Information Processing Society of Japan.

Authors Introduction

Mr.Takuro Iwanaga

He received his B.S. degree in Information Systems Engineering from the University of Miyazaki in 2021. He is currently enrolled in the master's program at the University of Miyazaki.

Prof. Makoto Sakamoto

Makoto Sakamoto received the Ph.D. degree in computer science and systems engineering from Yamaguchi University. He is presently an associate professor in the Faculty of Engineering, University of Miyazaki. He is a theoretical computer scientist, and his current main research interests are automata theory, languages and computation. He is also interested in digital geometry, digital image processing, computer vision, computer graphics, virtual reality, augmented reality, entertainment computing, complex systems and so on.

Prof. Takao Ito

He is Professor of Management of Technology (MOT) in Graduate School of Engineering at Hiroshima University. He is serving concurrently as Professor of Harbin Institute of Technology (Weihai) China. He has published numerous papers in refereed journals and proceedings, particularly in the area of management science, and computer science. He has published more than eight academic books including a book on Network Organizations and Information (Japanese Edition). His current research interests include automata theory, artificial intelligence, systems control, quantitative analysis of inter-firm relationships using graph theory, and engineering approach of organizational structures using complex systems theory.

Prof. Satoshi Ikeda

He received PhD degree from Hiroshima University. He is an associate professor in the Faculty of Engineering, University of Miyazaki. His research interest includes graph theory, probabilistic algorithm, fractal geometry and measure theory.