Tree structure design for Connect6 opening

Jung-Kuei Yang[†1], Shi-Jim Yen[†2], Cheng-Wei Chou[†2], Jing Nie[†3], Xiao Bai[†3]

Abstract: Since Connect6 was introduced by Wu in 2005, many high-level computer program of Connect6 have also been developed. As the search space complexity in Connect6 is very high, computer must spend a large amount of time in searching most promising move. Recently, Monte Carlo Tree Search (MCTS) has become a well-known game search method, and has been successfully applied to many games. This study introduces how to design the tree structure in Connect6 opening. The opening is basis on the Board's position, and it is used to retrieve the relative information. In the study, Connect6 opening is a tree structure constructed by a lot of end games, but only the nodes from the final win position of the end game to the root node are saved in opening. The study is the first step of our research towards to build Connect6 opening. But the building is not constructed by the domain knowledge from Connect6 experts; it constructs openings by automated process of Connect6 opening systems. In other words, this research plan is to build Connect6 openings which are auto-generated ones by the program itself. It will combine with our previous experience in Bitboard knowledge base design, bitwise computing, and MCTS in Connect6, and apply it to the building of Connect6 Opening.

Keywords: Connect6, opening, tree structure

1. Introduction

This part will introduce the background and the purpose of the study. In this research, we want to develop a Connect6 opening which is effective for Connect6 program.

1.1 Connect6

Since Connect6 was introduced by Wu [2][3] in 2005, many high-level computer program of Connect6 have also been developed [6][8]. As the search space complexity in Connect6 is very high, computer must spend a large amount of time in searching most promising move.

Connect6 has two important features: numerous candidate moves and sudden-death property. Numerous candidate moves lead to complex search and the sudden-death characteristic increases the search complexity. The possibility of sudden-death should be considered in every game position[i]. The player who neglects this feature may lose the game.

1.2 The opening of Connect6

Opening plays an important role in most intelligent game design [4]. For Connect6, it can prevent the sudden-death in the beginning of a Connect6 game. Therefore, it is also one of the key factors in a computer game contest of Connect6.

This study is the first step of our research in constructing a Connect6 opening. The purpose of this study is to design the structure of saving in Connect6 opening. The study is helpful to Connect6 program, and it expects to retrieve the position saved in Connect6 opening efficiently. The opening design will combine with our previous experience in Bitboard knowledge base design and bitwise computing in Connect6 [5][9][10][11].

1.3 Searching in Connect6

Searching is both a method of solving problems and a means for programs to display their intelligence. When facing complex problems, computers must explore a vast number of states, which requires enormous computational time. Two means of tackling difficult problems exist in such situations. The first approach involves applying heuristic knowledge of relevant field to decrease the search states. This approach saves considerable time on problem solving. Currently, heuristic knowledge plays a significant role in branch elimination, but only effective evaluation can correctly evaluate different game states.

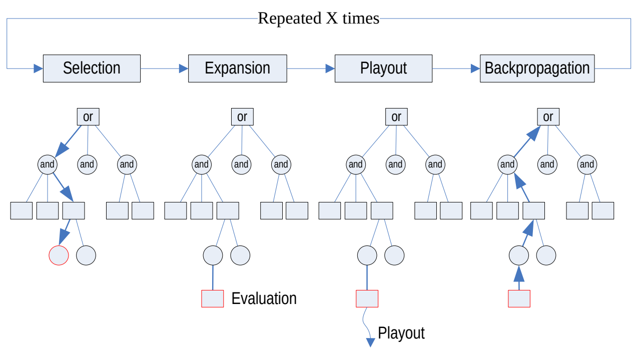


Fig. 1. Outline of the MCTS algorithm edited from [1]

The second approach involves selecting an efficient search algorithm. An effective search method can correctly guide search orientation and increase search efficiency. This can avoid unnecessary time wasting and focus the search on the optimal state space, significantly improving search performance. Recently, Monte Carlo Tree Search (MCTS) [1][7][9] has

---

†1 Dept. of Applied Foreign Languages, Lan Yang Institute of Technology, I Lan, Taiwan
†2 Dept. Of Computer Science and Information Engineering, National Dong Hwa University, Taiwan
†3 Software Engineering Department, Xiamen Institute of Software Technology, Xiamen, China

i The position is the state of Board, and it means the arrangement of all of the stones on a Board.

become a well-known game search method, and has been successfully applied to many games. Fig. 1 shows the outline of the MCTS algorithm.

2. The Tree Structure Design

The purpose of this study is to build the tree structure of Connect6 opening. In this part, we introduce the basis concept in designing the tree structure of Connect6 opening. First, we introduce the positions in Connect6 opening. Then we define the terminology used in the study. Next, we introduce the searching in Connect6 opening. Finally, we introduce the tree structure.

2.1 The positions in opening

There are two ways to model states of a Connect6 Board: positions and Connection [5][10][11]. A position is the arrangement of Black and White stones on Board. Fig. 2 shows an example of positions on a Connect6 Board. A cell-array is an array of consecutive cells, and it is a general way to record the position of Board.

In Connect6 Board, there are 361 cells on a 19x19 board. For saving all states of 361 cells on Board, we need 3^{361} state spaces because there are three states for every cell: Empty, Black, and White. Fig. 2 shows one of the 3^{361} state spaces, and it is a pretty big number. Therefore, it is hard to represent a state of Connect6 Board by a variable.

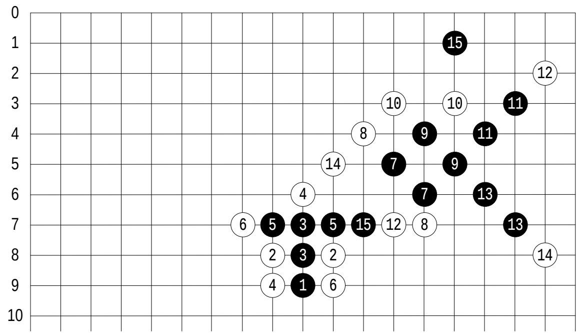


Fig. 2. An example of position on a Connect6 Board

The tree structure of a Connect6 opening must rely on positions as the index, and it can quickly retrieve the information by the position. Therefore, the data structure design of a position in Connect6 opening is a key point.

In this study, the data structure of position is divided into two parts: black string and white string. Therefore, combining black string and white string naturally form a Board's position. The main reason of the design is whenever a player plays move, only one string (black or white) changed. Take Table 1 as an example, when White plays M_4[ii], black string is stable (same as M_3). Only two stones add to white string. Therefore, when White plays move, the only thing we have to do is changing the white string, not the black string.

The location for all of the stones in Board is independent of the sequence to form it. In other words, a position is independent of the sequence how to form it. Therefore, before forming white string, the stones must be ordered. Take Table 1 as an example. When White plays M_4, there are four stones of White in the Board. First, four stones must ordered based on the index of cells[iii]. Then we can form the white string; otherwise, even if the same position, the string is different. Besides, a position in middlegame of Connect6 may come from several branches of the tree structure of Connect6 opening.

TABLE 1. THE NUMBER OF STONES IN DIFFERENT MOVES

MOVE	PLAYER	STONES	BLACK STRING	WHITE STRING
M_1	BLACK	1	$1(B_1)$	0
M_2	WHITE	3	$1(B_1)$	$2(W_1,W_2)$
M_3	BLACK	5	$3(B_1,B_2,B_3)$	$2(W_1,W_2)$
M_4	WHITE	7	$3(B_1,B_2,B_3)$	$4(W_1,W_2,W_3,W_4)$
M_5	BLACK	9	$5(B_1,B_2,B_3,B_4,B_5)$	$4(W_1,W_2,W_3,W_4)$
M_6	WHITE	11	$5(B_1,B_2,B_3,B_4,B_5)$	$6(W_1\sim W_4,W_5,W_6)$
M_7	BLACK	13	$7(B_1\sim B_5,B_6,B_7)$	$6(W_1\sim W_4,W_5,W_6)$
M_8	WHITE	15	$7(B_1\sim B_5,B_6,B_7)$	$8(W_1\sim W_6,W_7,W_8)$
M_9	BLACK	17	$9(B_1\sim B_7,B_8,B_9)$	$8(W_1\sim W_6,W_7,W_8)$
M_{10}	WHITE	19	$9(B_1\sim B_7,B_8,B_9)$	$10(W_1\sim W_8,W_9,W_{10})$

2.2 Searching in Connect6 opening

Connect6 opening is used for game search; therefore, the study will combine the opening with our previous experience in the searching of Connect6, and apply it to the building of Connect6 Opening. In our previous research, we have developed many search algorithms based on the two important features of Connect6: numerous candidate moves and sudden-death property, called 2-stage MCTS [7][9]. Fig. 3 shows the search architecture of 2-stage MCTS in Connect6.

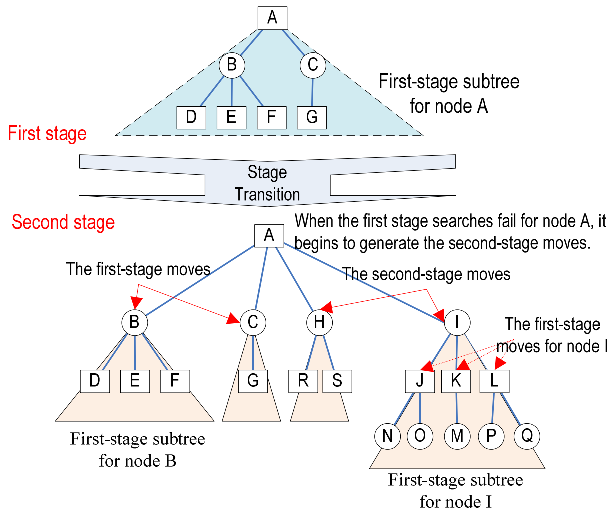


Fig. 3. Search architecture of 2-stage MCTS in Connect6

In 2-stage MCTS, the candidate moves is generated in two stages. The first stage focuses on Threat Space Search (TSS), which is designed to solve the sudden-death problem. For the

ii M_4 represents the fourth move of a Connect6 game, and it plays by White.

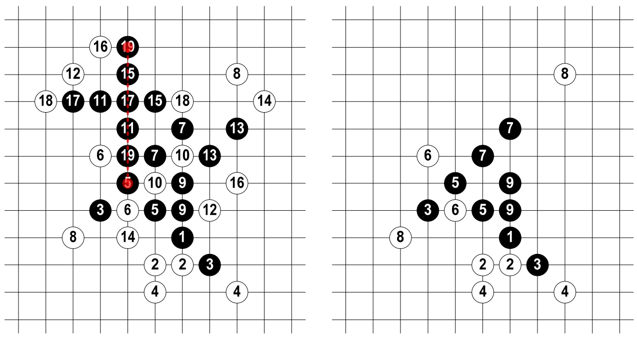
iii For the index of cells on Board, please refer to [9].

double-threat TSS in Connect6, 2-stage MCTS proposes an algorithm called Iterative Threat Space Search (ITSS) which combines general TSS with Conservative Threat Space Search (CTSS). The second stage uses MCTS to estimate the game-theoretic value of the initial position. This stage aims at finding the most promising move. The experiment proved that those search algorithms can play a good performance.

2.3 End position and win position

In Connect6 opening, position is the basis for retrieving the relative information. The end position is the position that a player (Black or White) gets six or more consecutive stones. Fig. 4(a) is an end position because Black gets six consecutive stones in M_{19} (marked by a red line in the figure). The win position is the position that a player (Black or White) is not yet gets six or more consecutive stones of its own, but can be found via the search algorithm to find the process to reach an end position. The search algorithm means 2-stage MCTS as discussed in "*2.2 Searching in Connect6 opening*".

According to this definition, there are many win positions when performing backtracks from an end position. Therefore, the win position means the final win position in this study. Take Fig. 4(a) as an example. M_{19} is end position, and the other moves of Black: M_{17}, M_{15}, M_{13}, M_{11}, M_9, M_7, and M_5 are win positions. But M_5 is the win position in this study as shown in Fig. 4(b).



(a) end position (b) win position

Fig. 4. (a) is an example of end position which Black gets six consecutive stones in M_{19}, and (b) is the win position got from (a).

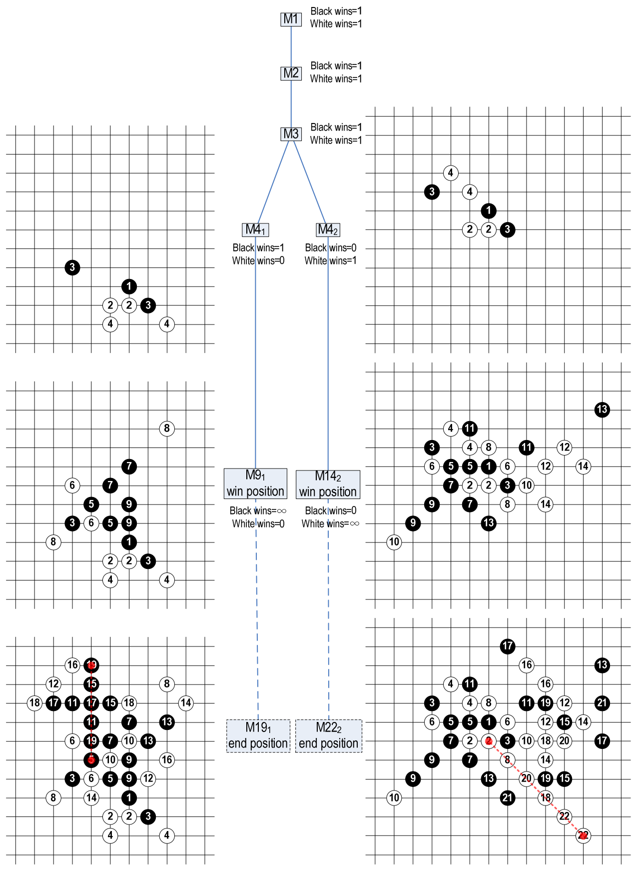
In the study, the design of Connect6 opening is based on an end position, backtracking to the win position, and finally backtracking to the first move. And the process forms a branch of the tree structure in Connect6 opening. Fig. 5(a) shows the branch based on the position of Fig. 4(a). The detail of tree structure will be further described in the next section.

From the above definition, win position is based on a search algorithm; therefore, it is related to the ability of a search algorithm. Fig. 4(b) is a win position, and it is backtracking from the end position of Fig. 4(a). In this study, we save the win position and all its backtracking positions until to the initial position. And all the positions save the number of wins in Black and White separately.

2.4 Black wins and white wins

When construct a branch based on an end position, all the nodes in the path from the leaf node (win position) to the root must record the win for the win position, it can be Black or White. When continue to construct the other branches based on other win positions, it will produce overlapping nodes near the root of the tree structure. For the overlapping nodes, it will not be just one win position under the nodes. Even there are different win positions under the node from different side (Black or White wins). Therefore, the number of Black (or White) wins must be recorded under those overlapping nodes.

Fig. 5 shows the tree structure forming by two end positions ((a) is the one for Black and (b) is the other for White). In Fig. 5(a), M_{19} is the end position of Black. In the study, end position is not record in Connect6 opening. The first position recorded in opening is the win position backtracking to the final win position from the leaf node. In Fig. 5(a), the Black move M_9 is the final win-position and it is the leaf node of the tree structure in Connect6 opening. In the study, the leaf node means the win player has been identified. The win position (M_9) in Fig. 5(a) means Black wins; therefore, the positions from M_1 to M_9 are recorded in Connect6 opening.



(a) black wins (b) white wins

Fig. 5. An example of the tree structure in Connect6

Fig. 5 shows a small part of the tree structure in Connect6 opening. (a) is the end position of Black wins, and (b) is the end position of White wins. In Fig. 5(a), the final win position is M_9, and M_8 to M_5 are omitted. In Fig. 5(b), the final win position is

M_{14}, and M_{13} to M_5 are omitted. In the tree structure, although there are only two branches under M_3, every node can develop other branches except for the end position (M_9 in (a) and M_{14} in (b)). This approach is combined Connect6 opening with the search algorithm, and it is more efficient in reducing the storage space.

3. The strategy of selecting a candidate move

In this part, we introduce the strategy of selecting a candidate move from Connect6 opening. First, we introduce the positions in Connect6 opening. Then we define the terminology used in the study. Finally, we introduce the tree structure.

For the game search, there are two purposes about saving the tree structure of positions in Connect6 opening. First, if there are search solutions in some position, this information must be fully controlled when it is in searching. It is important because of the feature of sudden death. Second, if there is not search solution in a position, the most promising moves must be recorded in Connect6 opening.

3.1 The algorithm of selecting a candidate move

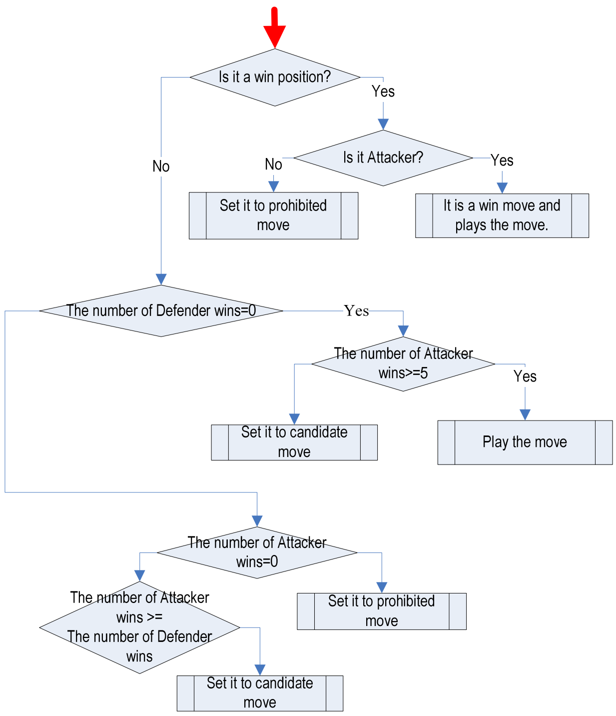


Fig. 6. The flow chart of selecting a candidate move from Connect6 opening

According to the two important features, the algorithm of selecting a candidate move from Connect6 opening is shown in below.

(1) **If the position of candidate move is a win position, it must be the next move. Otherwise, the candidate move is set to prohibited move.**

(2) **If there is not any wins in the position of Defender's move and the number of win more than 5 in the position of Attacker's move, it must be the next move. Otherwise, the position is set to candidate move.**

(3) **If there is not any wins in the position of Attacker's move, the position is set to prohibited move.**

(4) **If the number of wins of Attacker is bigger than Defender, the position is set to candidate move.**

According to the discussion of algorithm in selecting a candidate move, the first step uses the candidate move from Connect6 opening as the next move because it is the win position. The second step considers two reasons. First, there is not any number of Defender' wins under the position, and the threat to Attacker is lower based on the situation. The third step is the inverse of the second step, and it is bad to Attacker. The fourth step describes the equal situation; the move does not prefer to which side. It chooses the most promising move after performing MCTS search. The process of the algorithm is shown in Fig. 6.

4. Conclusion

Finally, we conclude our study in the design of tree structure for Connect6 opening.

4.1 More efficient storage space

According to the aforementioned tree structure in Connect6 opening, the win position is the leaf node, and once reached the position, the outcome has been determined. In other words, all the positions under the win position will not need to save in opening because the outcome has been identified. In addition to reduce the size of tree structure, it can reduce the storage space and the complexity to retrieve the information in opening.

4.2 More accurate prediction value

The Black wins of a position is the position which it is not a win position and it records the number of Black's win-positions under the node in the tree. From the above definition, Black wins (or White wins) is related to the number of win positions in opening, but the prediction of theoretical wins value. Therefore, Black (or white) wins is the predicted outcome value. When the opening tree is complete, the wins value is correct.

4.3 Future research

The next step of this study is to build a Connect6 opening. But the building is not constructed by the domain knowledge from Connect6 experts; it constructs openings by automated process of Connect6 opening systems. In other words, the purpose is to build Connect6 openings which are auto-generated ones by the program itself. This project will research innovative technologies, and it will combine our previous results into Connect6 openings. The study is helpful to Connect6 program.

Acknowledgments The authors thank anonymous reviewers for their valuable comments, and thank the National Science Council of the Republic of China (Taiwan) for financial support of this research under contract numbers MOST 103-2221-E-267-001-.

Reference

[1] B. Bouzy and G.M.J-B. Chaslot, "Monte-Carlo Go Reinforcement Learning Experiments," In IEEE 2006 Symposium on Computational Intelligence in Games, Reno, USA, pp. 187-194, 2006.

[2] I-C. Wu, D.-Y. Huang and H.-C. Chang, "Connect6," ICGA Journal, Vol. 28, No. 4, pp. 234-241, 2005.

[3] I-C. Wu and D.-Y. Huang, A New Family of k-in-a-row Games. the 11th Advances in Computer Games Conference (CG 2005), Proceedings of the 11th *Computers and Games*, in: H. Jaap van den Herik, Shun-Chin Hsu, Tsan-sheng Hsu and H.H.L.M. Donkers (Eds.), *Lecture Notes in Computer Science* (*LNCS 4250*), 2007, pp. 180–194.

[4] M. Buro, Toward Opening Book Learning , NECI Technical Note #2, 1997.

[5] S.-J. Yen and J.-K. Yang, The Bitboard Design and Bitwise Computing in Connect6. *Proceedings of the 14th Game Programming Workshop*, 2009, pp. 95–98.

[6] S.-J. Yen, T.-C. Su and I-C. Wu, The TCGA 2011 Computer-Games Tournament. *ICGA Journal*, **34** 2 (2011), pp. 108–110.

[7] S.-J. Yen and J.-K. Yang, Two-Stage Monte Carlo Tree Search for Connect6. *IEEE Transactions on Computational Intelligence and AI in Games*, **3** 2 (2011), pp. 100–118.

[8] S.-J. Yen, T.-C. Su and I-C. Wu, The TCGA 2011 Computer-Games Tournament. *ICGA Journal*, **34** 2 (2011), pp. 108–110.

[9] Jung-Kuei Yang, "MCTS design for Connect6," Ph.D. Thesis, National Dong Hwa University, Taiwan, 2011.

[10] Shi-Jim Yen, Jung-Kuei Yang, Kuo-Yuan Kao, and Tai-Ning Yang, "Bitboard Knowledge Base System and Elegant Search Architectures for Connect6," Knowledge-Based Systems, 0950-7051, vol. 34, 2012, pp. 43-54.

[11] Jung-Kuei Yang, "Bitboard Connection Code Design for Connect6," the 2013 Conference on Technologies and Applications of Artificial Intelligence (TAAI 2013), Taipei, Taiwan, December 2013.