8th International Conference on Advanced Computational Intelligence
Chiang Mai, Thailand; February 14-16, 2016

Building Connect6 Opening by Using the Monte Carlo Tree Search

Jung-Kuei Yang
Dept. of Applied Foreign Languages
Lan Yang Institute of Technology
I Lan, Taiwan
Email: kueiy@mail.fit.edu.tw

Ping-Jung Tseng
Dept. of Computer Application Engineering
Lan Yang Institute of Technology
I Lan, Taiwan
Email: james@mail.fit.edu.tw

***Abstract*—An opening book is an important part in most game-playing computer programs. The purpose of the research aims to construct an Opening-Book system of Connect6 game. In this study, it develops a real system that can apply Opening-Book system of Connect6 to Kavalan. However, the method in which constructed the system is not by profession's domain knowledge, and it is constructed by numerous end games. The study collects various end games to construct positions of opening book from public competitions, including self-games.**

The study incorporates the previous results into the design of the tree structure of Connect6 opening: the research of Bitboard design and bitwise computing of Connect6, revised algorithm of MCTS to fit the property of sudden-death, and the experience of software development of Connect6 game. It plays an important role of developing Connect6 opening to combine the building of Opening-Book system and the search algorithm of Connect6. In addition, it can show the overall efficiency only if the search algorithm and Connect6 opening are perfect match. Hence, two methods balance the advantage and disadvantage to achieve the greatest accomplishment.

The study has finished the analysis of Connect6 board and the design of Connect6 opening. Besides, it also finished the development of the Opening-Book system of Connect6 game, and attached it to the MCTS of Connect6. With the increasing of positions saving in the Connect6 opening, Kavalan already greatly reducing the time spent on opening-game. Therefore, the results of the research greatly enhance the search efficiency of Kavalan.

Keywords—Connect6, Monte Carlo tree search, opening

I. 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.

A. Connect6

Since Connect6 was introduced by Wu [2][3] in 2005, many high-level computer program of Connect6 have also been developed [4][9]. 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[1]. The player who neglects this feature may lose the game.

B. The opening of Connect6

Opening plays an important role in most intelligent game design [1][10]. 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 contest of computer game of Connect6.

The purpose of the research aims to construct an Opening-Book system of Connect6 game. In this study, we develop a real system that can apply Opening-Book system of Connect6 to Kavalan. However, the method in which constructed the system is not by profession's domain knowledge, and it is constructed by many end games. The study collects various end games from public competitions, including self-games.

The study is helpful to Connect6 program, and it expects to retrieve the position saving in Connect6 opening efficiently. The opening design will combine with our previous experience in Bitboard knowledge base design, bitwise computing, and Monte Carlo Tree Search (MCTS) in Connect6 [4][5][6][8][10].

C. 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.

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

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

optimal state space, significantly improving search performance. Recently, Monte Carlo Tree Search (MCTS) [4][8] has become a well-known game search method, and has been successfully applied to many games. Fig. 1 shows the outline of the MCTS algorithm.

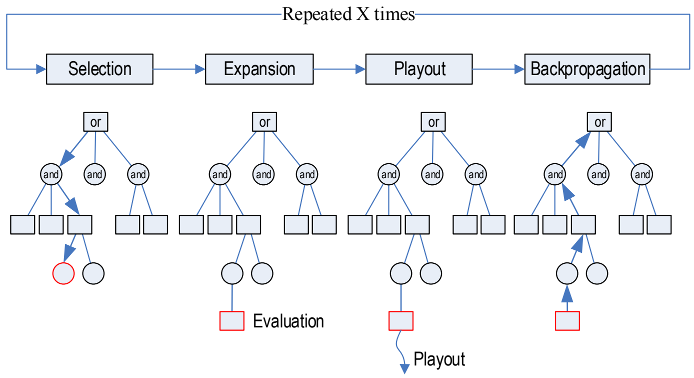


Fig. 1. Outline of the MCTS algorithm edited from [4][8]

II. THE TREE STRUCTURE DESIGN

The purpose of this study is to build 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. Finally, we introduce the tree structure.

A. The positions in opening

There are two ways to model states of a Connect6 Board: positions and Connection [5][6][10]. 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, it needs 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 in programming.

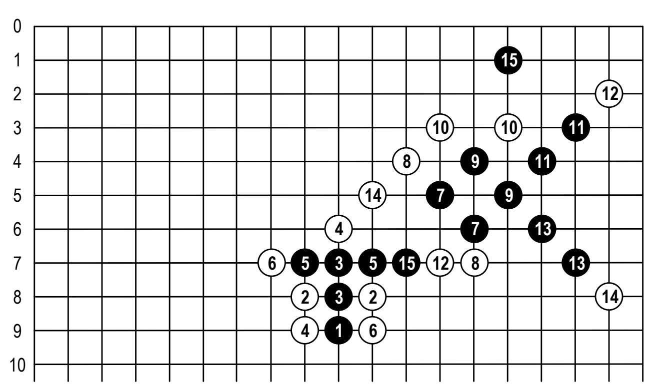


Fig. 2. An example of position on a Connect6 Board

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[2], black string is stable. Because the Black stones of M_3 and M_4 are the same, 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.

TABLE I. 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	$9(B_1,{\sim}B_7,B_8,B_9)$	$10(W_1{\sim}W_8,W_9,W_{10})$

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[3]. 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.

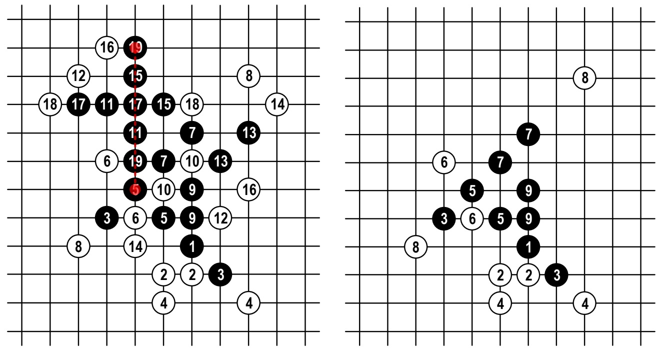
B. End position and win position

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. 3 (a) is an end position because Black gets six consecutive stones in M19 (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 "III. A 2-stage MCTS in Connect6".

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. 3 (a) as an example. M19 is an end position, and the other moves of Black: M17, M15, M13, M11, M9, M7, and M5 are win positions. But M5 is the win position in this study as shown in Fig. 3 (b).

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

[3] For the index of cells on Board, please refer to [4].



(a) end position (b) win position

Fig. 3. (a) is an example of end position which Black gets six consecutive stones in M19, 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. 4(a) shows the branch based on the position of Fig. 3(a). The detail of tree structure will be further described in the next section.

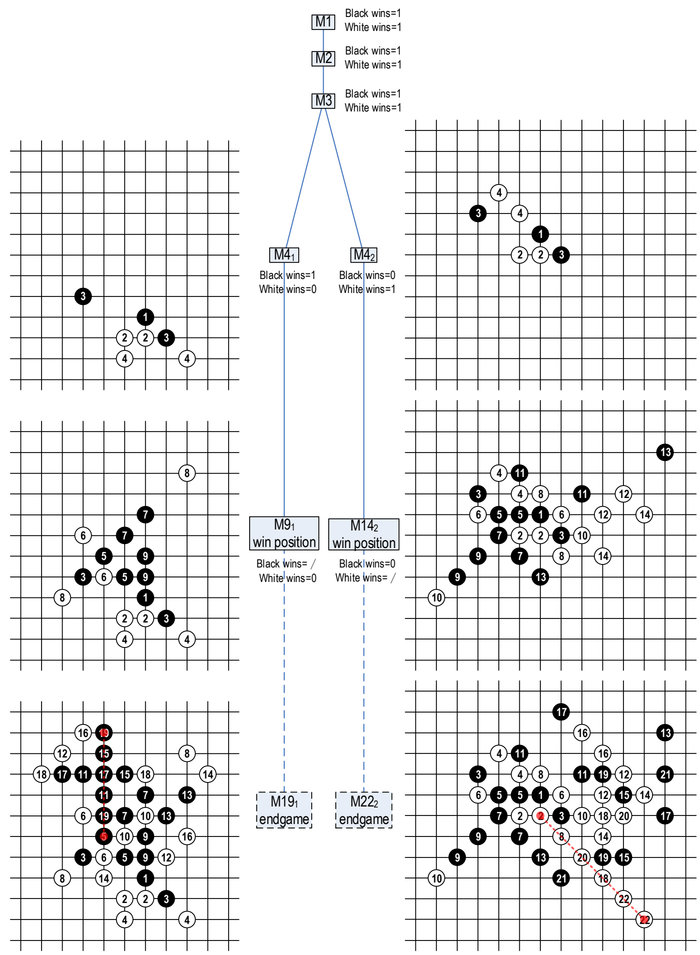
From the above definition, win position is based on a search algorithm; therefore, it is related to the ability of a search algorithm. Fig. 3(b) is a win position, and it is backtracking from the end position of Fig. 3(a). In this study, we save the win position and all its backtracking positions until to the initial position. And all the positions save the number of wins in Black and White separately.

C. *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. 4 shows the tree structure forming by two end positions ((a) is the one for Black and (b) is the other for White). In Fig. 4(a), M19 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. 4(a), the Black move M9 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 (M9) in Fig. 4(a) means Black wins; therefore, the positions from M1 to M9 are recorded in Connect6 opening.

Fig. 4 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. 4(a), the final win position is M9, and M8 to M5 are omitted. In Fig. 4(b), the final win position is M14, and M13 to M5 are omitted. In the tree structure, although there are only two branches under M3, every node can develop other branches except for the end position (M9 in (a) and M14 in (b)). This approach is combined Connect6 opening with the search algorithm, and it is more efficient in reducing the storage space.



(a) black wins (b) white wins

Fig. 4. An example of the tree structure in Connect6

III. The Auto-Generated System of Connect6 Opening Which Integates MCTS

The purpose of this study is to build Connect6 opening. But the building is not constructed by the domain knowledge from Connect6 experts; it constructs opening 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 chapter introduce the opening book design for Connect6. First, we introduce the searching in Connect6 opening. Then how to select a candidate move was introduced. In this part, we introduce the strategy of selecting a candidate move from Connect6 opening. Finally, the source of positions saving in Connect6 opening will be introduced.

A. 2-stage MCTS in Connect6

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 [4][8]. Fig. 5 shows the search architecture of 2-stage MCTS in Connect6 [4][8].

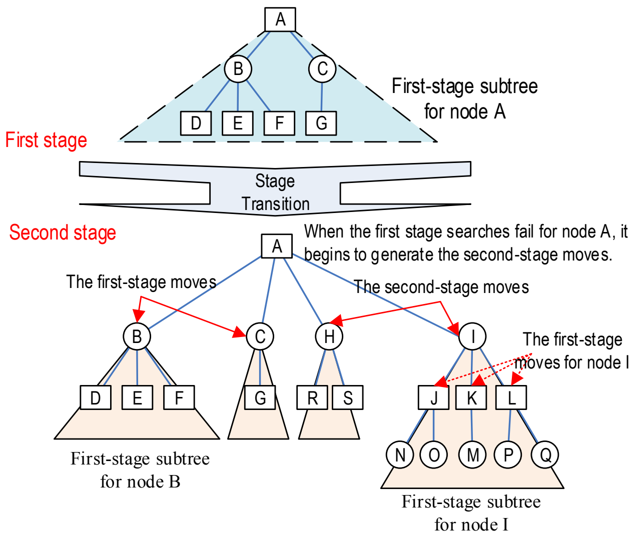


Fig. 5. Search architecture of 2-stage MCTS in Connect6

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 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 a position especially from the initial position. This stage aims at finding the most promising move. The experiment proved that those search algorithms can play a good performance.

Furthermore, this study take opening-book system as an auxiliary tool in MCTS. Therefore, opening book must be combined with Monte Carlo Tree Search during the searching process. In other words, the candidate moves generated by MCTS must consider the position saving in opening book, and the result of 2-stage MCTS must also be saved into opening book.

B. The strategy of selecting a candidate move

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 positions, 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.

1) The win position between candidate moves

There is a feature in the win position existed in the tree structure of positions in Connect6 opening. It is different about the saving of candidate moves under a position if there is a different side of end position under the position. There are two conditions, and we separately discuss them as follows.

a) The win move of candidate moves

For the candidate moves under a position of stone (for example: Black stone of the root node in Fig. 6), if it exists an end position of candidate move (White win), the win position must be saved as the only candidate move. It is not necessary to save the other candidate moves because the win move has been found.

Take the Fig. 6(a) as an example. Under the Black position, if it exists the end position of White under the candidate position of White (for example: W_3), the other candidate moves (W_1, W_2, W_4 in Fig. 6(a)) need not be saved in Connect6 opening. In the situation, White must play W_3 to win the game, it cannot be other move. Therefore, it is not necessary to save W_1, W_2, or W_4.

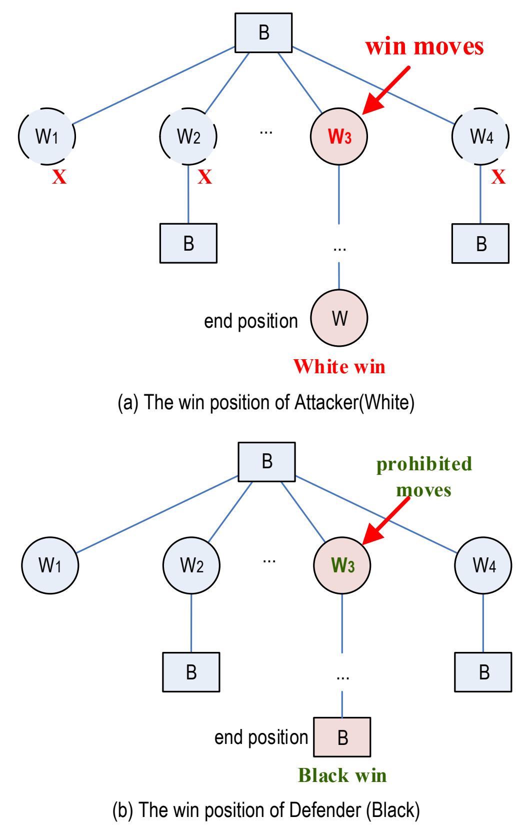


Fig. 6. The flow chart of selecting a candidate move from Connect6 opening

b) The prohibited move of candidate moves

For the candidate moves under a position of stone (for example: Black stone of the root node in Fig. 6), if it exists an end position which is not the candidate move (Black win), the win position must be saved as the prohibited move. Absolutely the move must not be considered in the candidate moves from the initial position.

Take the Fig. 6(b) as an example. Under the Black position, if it exists the end position of Black under the candidate position of White (for example: W_3), the move must be prohibited. Other moves (W_1, W_2, and W_4 in Fig. 6(b)) are the candidate moves from the initial position in this situation. In other words, White must consider W_1, W_2, or W_4 and not W_3 from the initial position.

2) The algorithm of selecting a candidate move

According to the two important features of Connect6, the algorithm of selecting a candidate move from Connect6 opening must consider the sudden-death feature. Therefore, there are three steps for generated candidate moves from Connect6 opening as shown in Fig. 7.

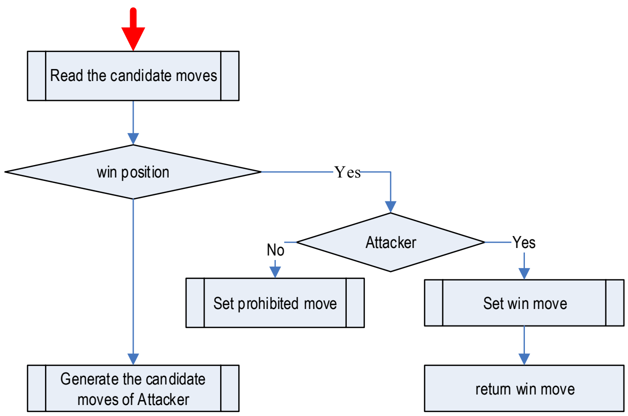


Fig. 7. The algorithm of selecting candidate moves.

Step 1. Read the candidate moves

First, it reads all candidate moves from opening book based on the position. Then it sequentially checks all the candidate moves according to the next step.

Step 2. Judge whether it is a win position

Step 2 checks all the candidate moves for whether the move is a win position. If it is a win position of Attacker, it sets the move as a win move, and plays it. Otherwise, it sets the move as a prohibited move. The purpose of this step is judging whether the move is a sudden-death position. And the win position is a sudden-death position.

Step 3. Generate the candidate moves of Attacker

Excluding the sudden-death position, the rest of the moves are candidate moves of Attacker. No matter whether the move is good for Attacker, it should be candidate moves of Attacker.

The purpose of the algorithm that generating candidate moves is to yield Attacker's moves from the opening book. Then it checks them by MCTS to select the most promising move. If the criterion of selecting candidate moves focus on those moves that the threat is lower, it is easy to find the local optimal of candidate moves. Therefore, the first principle is that it is not the sudden-death move, and it can be the candidate moves of Attacker. Whether there are any other promising moves is checked by MCTS.

C. *The source of position saving in the Connect6 opening*

The purpose of this study is to build Connect6 opening by an auto-generated process, and it constructs the tree structure of positions into opening book by the end game.

There are two main sources of the end game. The first is the end game generated from the Connect6 contest. Those end games are the results generated from the competition of superior program around the world, and it is valuable. The second one is generated from the contest by the program itself. It is good for normal distribution of opening book, and it can be implemented via experimental control of program.

1) The end game gotten from Connect6 contest

Since Wu [1] investigated online board games like k-in-a-row or Connect(m, n, k, p, q) in 2005, Connect6 has been a very popular research topic. Connect6 contest has been held every year since 2005; therefore, there is a considerable number of end games are generated.

The study analysis those end games and checks the win position by the searching algorithm, and saves those positions into opening book. It is a good way to avoid making the same mistakes from these end games.

2) The end game gotten from the program itself

Kavalan already have the ability to self-battle; therefore, the study further develops a system to analysis the end games and to add the branch of end game to opening book. In the process of automated generated system through the self-battle, Kavalan enhances its level of playing move by increasing the number of positions saving in opening book.

However, if the opening of the game saving in opening book only focus on some important positions, the opening book will lose the function when the position break away from the tree structure. Therefore, the allocation of positions in the beginning of tree structure uses relatively comprehensive configuration. So the biased configuration will not happen on the issue.

3) The positions gotten from flipping, mirroring and rotating of Board

There are two solutions about flipping, mirroring and rotating of Board. First, one position is saved in opening book. This is a way to save storage space, but it will waste computing time when the program inquiries positions from opening book.

Second, the positions gotten from flipping, mirroring and rotating of Board are saved in opening book when handling the position of end game. The key point of the two methods is the speed of inquiring opening book. In this study, the second method is used because inquiring opening book spends more time than computing the flipping, mirroring and rotating of Board. So, the second method is used to reduce the number of reading opening book.

IV. CONCLUSION

This study developed an opening book system of Connect6 which can be used in the program of "Kavalan". However, the method of constructing opening book is not built by the domain knowledge from Connect6 experts; this study constructed it from the end games gotten from competition games and the result of self-battle games. The research incorporated the previous results from the design of Connect6's Bitboard design and bitwise computing, experiences of relating search algorithm, and experience of developed program into constructing the opening book of Connect6.

A. Finishing the analysis of Connect6 board and the design of Connect6 opening

Depended on the results from the analysis of Bitboard and searching 2-stage MCTS, the study hoped to use it as the basis of constructing opening book of Connect6. As for the design of tree structure in opening book, it became the basis for designing the tree structure of Connect6 opening. The advantages of tree structure design are shown as below.

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.

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.

Besides, the use of Bitboard and bitwise computing improved the searching efficiency. While its efficiency increased, the entire 2-stage MCTS would be followed up, and the design of tree structure in opening book would influence in the beginning of a game competition. Thus, these two systems have been considered indispensable foundations to advance the level of Connect6 program.

B. Finishing the development of the Opening-Book system of Connect6 game, and attached it to the MCTS of Connect6

In this study, to develop an opening book system of Connect6 which can be used in the program of "Kavalan" is the most important work. Therefore, this study developed the auto-generated system of Connect6 opening which integrates 2-stage MCTS of Connect6.

1) The auto-generated system of Connect6 opening

Connect6 is a game that have characteristic of sudden-death. A game with characteristic sudden-death is inappropriate to use probability search algorithm, which use prediction to decide the move. However, MCTS is a good search algorithm when facing numerous candidate moves. Hence, the study proposed an auto-generated system of Connect6 program to support MCTS, combining opening book into 2-stage MCTS to enhance the search efficiency.

2) The development of opening system could reduce the time spent on searching and at the same time, increasing its effectiveness and efficiency.

With the increasing number saving in opening book with tree structure, the program would saving the time spent on the beginning of a game competition; hence, the capability of searching enhance relatively. This works in which plays a crucial role during the development of opening book system. During the game time, the perfect combination of search algorithm and opening book show an entire efficiency. Hence, only applying both of weaknesses and strengths could achieve the great function.

ACKNOWLEDGMENT

The authors thank anonymous reviewers for their valuable comments, and thank the Ministry of Science and Technology, R.O.C. for financial support of this research under contract numbers MOST 104-2221-E-267-001 -.

REFERENCES

[1] M. Buro, "Toward opening book learning," Icca Journal 22.2 (1999): 98-102.

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

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

[5] 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.

[6] 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.

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

[8] 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.

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

[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] Ting-Han Wei, I-Chen Wu, Chao-Chin Liang, Bing-Tsung Chiang, Wen-Jie Tseng, Shi-Jim Yen, and Chang-Shing Lee, "Job-Level Algorithms for Connect6 Opening Position Analysis," ECAI Computer Games Workshop 2014, Prague, Czech Republic, August 2014.